Ollscoil na hÉireann, Corcaigh

National University of Ireland, Cork

virtual reality protein visualisation and multiplayer education

Thesis presented by **Tianshu Xu**

for the degree of **Doctor of Philosophy**

University College Cork

School of Computer Science and Information Technology

Head of School: Prof Utz Roedig

Under supervision of:

Dr. Sabin Tabirca and Dr. Mark Tangney

2024

Contents

Abstract vi
Acknowledgements vii
List of Tables viii
List of Figures ix
List of Acronyms xiii

1 Introduction **1**
1.1 Main Aim 1
1.2 Chapters Outline 2
1.3 Outcomes 5
1.4 Author's Publications 6
1.5 Research Scope, Contributions, and Conclusion 7

2 Literature Review **12**
2.1 Introduction 12
2.2 Methodology 13
2.2.1 Review Planning 13
2.2.2 Conducting the Review 17
2.2.3 Review Results and Differentiation from this Thesis 19
2.3 Software Review 26
2.3.1 General Facts about Protein 26
2.3.2 Protein Visualisation Tools 29
2.3.3 Protein Folding Game and Tools 40
2.3.4 Protein Gamification and Education 43
2.4 Conclusion 46

3 Technical Foundations: 3D Modeling, Game Engine and VR Educational Systems **48**
3.1 Introduction 48
3.2 Blender 49
3.3 Unity 53
3.4 Education in Virtual Reality 64
3.5 Conclusion 65

4 Computational Protein Design in Unity **68**
4.1 Introduction 68
4.2 Novel Feature Contribution Apart from Foldit 71
4.3 Gameplay and Design 73
4.3.1 An Opening Cutscene 74
4.3.2 The User Interface 74
4.3.3 Gameplay Mechanics 79
4.3.4 Protein Model Visualisation Design 87
4.3.5 Reset Function Design 88

4.3.6 Level System Design . . . 89
4.3.7 Audio System Design . . . 90
4.3.8 Transition and Animation Design . . . 90
4.3.9 Protein Parameter Database Design . . . 91
4.4 Technical Aspects . . . 91
4.4.1 Drag&Drop and Implementation . . . 91
4.4.2 Protein Visualisation and Manipulation . . . 97
4.4.3 Bend&Twist and 3D Rigging . . . 103
4.4.4 Grab&Throw and VR Version . . . 109
4.4.5 Protein in silico Parameter Database . . . 110
4.5 Evaluation . . . 115
4.5.1 PC Version Evaluation . . . 116
4.5.2 VR Version Evaluation . . . 122
4.6 Conclusion . . . 123

5 Multiplayer VR for Real-Time Collaborative Protein Modeling and Analysis 125
5.1 Introduction . . . 125
5.1.1 Background of the Multiplayer Idea . . . 126
5.1.2 Why Developing ProMVR in VR . . . 126
5.1.3 Virtual Meeting Room . . . 127
5.1.4 PDB File Visualisation . . . 127
5.2 ProMVR Tool Design . . . 129
5.2.1 Multiplayer Connection . . . 129
5.2.2 Remote Communication . . . 132
5.2.3 PDB File Visualisation . . . 132
5.3 ProMVR Implementation . . . 133
5.3.1 Multiplayer Connection . . . 133
5.3.2 Remote Communication . . . 139
5.3.3 PDB File Visualisation . . . 143
5.4 Latency Test of ProMVR . . . 160
5.4.1 Introduction . . . 160
5.4.2 Rationale and Importance of Latency Evaluation in Multiuser VR Systems 162
5.4.3 Methodology . . . 163
5.4.4 Testing and Results . . . 165
5.4.5 Discussion and Conclusion . . . 168
5.5 User Experience Evaluation of ProMVR . . . 169
5.5.1 Evaluation Objectives and Importance in Multiuser Systems . . . 169
5.5.2 Participants and Evaluation Context . . . 170
5.5.3 Evaluation Methodology . . . 171
5.5.4 Evaluation Results - Observations and Findings . . . 172
5.5.5 Conclusion and Future Directions . . . 174

6 Discussion and Future Work 176
6.1 Key Findings . . . 176
6.2 Interpretations . . . 177
6.3 Implications for Education and Research . . . 178
6.4 Limitations and Future Work . . . 179
6.5 Conclusion . . . 180

References **181**

A User Experience Questionnaire for ProMVR **195**

Declaration

This is to certify that the work I am submitting is my own and has not been submitted for another degree, either at University College Cork or elsewhere. All external references and sources are clearly acknowledged and identified within the contents. I have read and understood the regulations of University College Cork concerning plagiarism and intellectual property.

Abstract

This research explores the integration of Virtual Reality (VR) in the visualisation and design of protein structures, emphasising its revolutionary impact on educational and research methodologies in bioinformatics and protein design. The research harnesses VR's capability to transform traditional two-dimensional visualisation approaches into interactive, three-dimensional simulations, thereby enhancing the comprehension and manipulation of complex protein models.

A significant part of the thesis is dedicated to developing "Pocket Peptides", a single-player game designed to enhance scientific outreach through interactive protein design. The game uses tools such as Iterative Threading ASSEmbly Refinement (I-TASSER) for protein structure prediction, UCSF Chimera for three-dimensional model conversion, and Unity and C# for game design and development, integrating these technologies into a gameplay environment that supports educational and research objectives.

Furthermore, the research expands into multiplayer gaming with "Pepblock Builder VR" and "Protein Multiplayer Virtual Reality (ProMVR)," extending VR's educational benefits by enabling collaborative and interactive learning experiences. This multiplayer environment supports the sharing and manipulation of protein designs. It cultivates a learning space where users can engage with and learn from each other in real-time, fostering a sense of community and shared learning.

This study demonstrates VR's potential to enhance the educational landscape of protein engineering significantly, making complex scientific concepts more accessible and engaging through gamified learning environments. The findings suggest that VR can be a powerful tool in the future of scientific education and research, providing immersive experiences that promote a deeper understanding of the protein molecular world, empowering learners to grasp even the most intricate concepts.

Keywords: Virtual Reality, protein visualisation, educational gaming, interactive learning, multiplayer gaming.

Acknowledgements

As I come to the conclusion of this PhD journey, it is with immense gratitude that I reflect on the countless people who have contributed to this project and supported me throughout the entire process. Without the guidance, patience, and collaboration of my supervisor, colleagues, friends, and family, this work would not have been possible. I would like to take this opportunity to express my heartfelt thanks to those who have made this thesis and my PhD experience a success.

First and foremost, I would like to extend my deepest appreciation to Sabin, my supervisor, for his invaluable guidance, insight, and encouragement throughout my PhD. His unwavering support and belief in my work have been instrumental in shaping not only the direction of this thesis but also my growth as a researcher. Sabin's dedication to fostering creativity, critical thinking, and problem-solving in his students is truly commendable. He challenged me to think more deeply, and constantly push the boundaries of my own understanding. Whether through constructive feedback on my drafts or impromptu brainstorming sessions that helped me navigate particularly complex challenges, Sabin's mentorship has been pivotal. His patience and belief in my abilities have given me the confidence to persist through the difficulties of this journey. For all this and more, I am deeply grateful.

A special thanks goes to Yanlin, whose technical assistance was crucial in ensuring that the server infrastructure required for my research ran smoothly. Throughout the project, there were numerous occasions where server-related issues threatened to disrupt the progress of my experiments. In these moments, Yanlin stepped in with her extensive knowledge and skills, troubleshooting issues and offering solutions that allowed me to continue my work without significant delays. Her dedication to ensuring that the computational resources were always available and functioning properly is something I deeply appreciate. Without her tireless efforts, much of the server-side work mentioned in this thesis would not have been possible.

I am also immensely grateful to my colleagues and fellow researchers Isabel, Deniz, Bogdan, and Vamsi who stood by me throughout this PhD journey, offering their time and energy. From discussions that helped clarify concepts to feedback that improved the clarity of my work, their contributions have been indispensable. I would like to especially thank those who took the time to review my thesis, offering detailed critiques and helpful suggestions. Their willingness to carefully read through drafts, provide comments, and suggest improvements made a significant impact on the final thesis. Their insights not only sharpened my arguments but also helped me see my work from new perspectives.

Additionally, I want to extend my appreciation to everyone who provided moral support throughout this journey. The emotional and psychological aspects of pursuing a PhD are often underestimated, and having a network of supportive friends and colleagues was crucial in keeping me grounded and motivated. Whether it was a word of encouragement

during difficult times or simply sharing a moment of laughter to relieve stress, their presence made all the difference.

Lastly, I would like to thank my family for their unconditional love and support throughout this journey. They have been my rock, providing constant encouragement and understanding as I navigated the many challenges of this PhD. Their belief in me has been a source of strength. I am truly fortunate to have such a supportive family who has always been there for me, no matter the circumstances.

In conclusion, while my name appears on the cover of this thesis, it represents the collective efforts of many individuals who have contributed to its completion. From the technical guidance and intellectual input of my supervisor and colleagues to the emotional and moral support of my friends and family, this work is as much theirs as it is mine. I am forever grateful to everyone who has been part of this journey, and I look forward to continuing to collaborate and grow together in the years to come.

Thank you all.

Tianshu Xu

Funding

My Ph.D. was financially supported by Centre for Research and Training in Advanced Networks for Sustainable Societies (ADVANCE CRT), Science Foundation Ireland (SFI), under Grant number 18/CRT/6222.

List of Tables

2.1 Selection Criteria 18
2.2 Selected Articles 19

4.1 Development Timeline and Contribution Summary 70
4.2 Questionnaire Result 121
4.3 Descriptive Statistics of Central Tendency of Likert Responses (N=20) . 121

5.1 Calculated Latency 168
5.2 ProMVR SUS Questionnaire Result 174

List of Figures

2.1 PRISMA-ScR Checklist 15
2.2 Protein Organisation 28
2.3 Ab Initio Prediction 32
2.4 Homology Modelling 33
2.5 Computational Pipeline for Protein-Protein Docking 35
2.6 Protein Structure Prediction via Threading-Based Fold Recognition . . 36
2.7 Interactive Web-based Molecular Visualization Using 3DMol.js 45

3.1 Blender - Interface Overview with Key Functionality 50
3.2 Blender 3D Rigging - Biomechanical Rigging Pipeline 51
3.3 Blender-Python Scripting Interface for Procedural 3D Modeling 52
3.4 Unity Engine Workspace 53
3.5 Asset Store - Curated Asset Pipeline 54
3.6 Unity Prefab Workflow Optimization 56
3.7 Collision Detection System Optimization 57
3.8 Physically-based Particle Systems 58
3.9 Adaptive Audio Mixing System 60
3.10 Keyframe-based Animation System 61
3.11 C# Script component-based architecture 62
3.12 Start and Update Methods - Frame-based Execution Model 63

4.1 Opening Cutscene 75
4.2 Designed UI Sketch 76
4.3 Implemented User Interface (UI) 78
4.4 Main UI and the Game 80
4.5 Protein structure prediction 82
4.6 Game with Drag&Drop 83
4.7 VR Buttons 85
4.8 VR Game Main Scene 86
4.9 Designed Database 92
4.10 OnBeginDrag Method 93
4.11 OnDrag Method 94
4.12 OnEndDrag Method 95
4.13 OnDrop Method 96

4.14 Code for Moving Protein Model 98
4.15 Differences between 2 Parts 99
4.16 Distinguish between Rigid and Flex Parts 100
4.17 Adding a bone 101
4.18 Adding Other Bones 102
4.19 Adding the IK Constraint 103
4.20 Output of the Script 104
4.21 Checking User Interaction 107
4.22 Rotating the Bones 108
4.23 Rotating Adjacent Bones 109
4.24 DistanceGrabbable Script 110
4.25 VR Version 111
4.26 OnCollisionEnter Method 112
4.27 Structure JSON File 113
4.28 Retrieving the Parameters 114
4.29 NullReferenceException Error 117
4.30 Tutorial System 118
4.31 Likert-Scale Usability Feedback Distribution 120
4.32 Boxplots showing median (line), InterQuartile Range (IQR) (box) and range (whiskers) of responses. Boxplots of Likert scores per item, showing strong ceiling effects in Q1/Q2, and skewed low scores in Q4 (Reverse-coded). The IQRs of Q3 and Q5 indicate moderate dispersion. This figure is reproduced under fair use for academic purposes. 122
4.33 Big Complex Protein 124

5.1 Nanome 128
5.2 Pipeline for converting PDB files to 3D OBJ format using server-side UCSF Chimera, integrated into the VR rendering flow. This figure is reproduced under fair use for academic purposes. 130
5.3 ProMVR 131
5.4 Client Server Host 135
5.5 Mirror ClientRPC Commands 137
5.6 Mirror ClientRPC 138
5.7 Vivox voice and text communication pipeline used in ProMVR. This figure is reproduced under fair use for academic purposes. 140
5.8 Vivox Channel 142
5.9 Spawn Protein 144
5.10 Open Babel 146
5.11 PROFASA 150
5.12 NGLVieweR 151

5.13 Architecture Diagram of ProMVR and Its Integration with PSGO and other External Services 154
5.14 PS-GO and ProMVR 156
5.15 PS-GO Parametric Protein Search Engine 157
5.16 PS-GO and Protein Parameters 158
5.17 Chimera Command-line Conversion 159
5.18 PDB Loading Output 167

List of Acronyms

VR Virtual Reality

AR Augmented Reality

I-TASSER Iterative Threading ASSEmbly Refinement

ProMVR Protein Multiplayer Virtual Reality

PRISMA-ScR Preferred Reporting Items for Systematic Reviews and Meta-Analyses Extension for Scoping Reviews

ACM the Association for Computing Machinery

macOS Macintosh Operating System

EVRGs Educational Virtual Reality Games

PDB Protein Data Bank

FASTA FAST-All

RBVI Resource for Biocomputing, Visualisation, and Informatics

FG-MD Fragment Guided Molecular Dynamics

CASP Critical Assessment of Structure Prediction

MPMV Mason Pfizer Monkey Virus

DAE Digital Asset Exchange

FBX Filmbox

UI User Interface

pI isoelectric point

MW molecular weight

API Application Programming Interface

JSON JavaScript Object Notation

IK Inverse Kinematics

FK Forward Kinematics

HMD Head Mounted Display

IP Internet Protocol

SDK Software Development Kit

UDP User Datagram Protocol

TCP Transmission Control Protocol

DLL Dynamic-Link Libraries

CDK Chemistry Development Kit

RTT Round-Trip Time

LAN Local Area Network

SFI Science Foundation Ireland

ADVANCE CRT Centre for Research and Training in Advanced Networks for Sustainable Societies

AI Artificial Intelligence

PSGO Parametric Protein Search Engine

PROFASA Protein Fragment And Structure Analysis

SUS System Usability Scale

UX User Experience

IQR InterQuartile Range

1 Introduction

1.1 Main Aim

Protein structures, the building blocks of biological functionality, are pivotal in understanding various biochemical processes. Traditional methods of protein visualisation are limited by two-dimensional representations, which fail to capture the intricate details and dynamic nature of protein interactions. This limitation has propelled the exploration of advanced visualisation techniques, notably through the application of VR. This technology not only surpasses traditional barriers by providing three-dimensional interactive models but also enhances the educational and experimental approaches to protein studies. This research primarily aims to explore and demonstrate how VR can revolutionise the visualisation and design of protein structures, making these processes more intuitive, accessible, and engaging.

Furthermore, the study employs a scoping review methodology, adhering to the Preferred Reporting Items for Systematic Reviews and Meta-Analyses Extension for Scoping Reviews (PRISMA-ScR) guidelines, to evaluate current literature and technological developments in the field. This comprehensive review not only highlights the progression of VR technologies in protein design but also explores potential future advancements and applications.

One of the key innovations in this research is the development of the single-player gaming platform, Pocket Peptides. This tool exemplifies how gamification can be effectively integrated into the scientific visualisation process, making the exploration of protein structures not only educational but also highly engaging. Pocket Peptides is designed to be a gamified learning experience where users can explore the intricacies of peptide structures through interactive gameplay like drag&drop, bend&twist, grab&throw to solve scientific puzzles. The game mechanics are tailored to facilitate an intuitive

understanding of protein folding, peptide bonding, and structural manipulation, making complex molecular biology concepts accessible to a broader audience.

Pocket Peptides offers unique approach to learning, where players are challenged to solve puzzles and complete tasks that mirror real-life protein folding and design scenarios. By engaging with these challenges, users can develop a hands-on understanding of the principles underlying protein structure and function. Further extending into multiplayer functionalities, this tool also enables collaborative learning environments where users share insights and contribute collectively to protein modelling tasks, enriching the educational experience.

The study delves into the development of specialised VR tools and platforms tailored for protein visualisation and design. These tools are not mere visual aids but are designed to actively engage users in the process of molecular modelling and manipulation, thereby contributing to a deeper understanding of protein dynamics. The overarching goal is to create VR-based environment where users, from students to professional researchers, can interact with molecular models in real-time, explore protein folding and design processes. This work specifically focuses on the design, implementation, and evaluation of ProMVR, a novel VR system that supports multi-user interaction for protein visualisation. Through this platform, the research seeks to enhance the educational and research methodologies in molecular biology by providing a more intuitive and engaging means of exploring protein structures.

The subsequent chapters will detail the methodology adopted for this review, discuss the findings from the literature, and explore the practical applications of VR in educational and research settings. By doing so, this thesis aims to substantiate the role of VR in enhancing the understanding of protein structures, ultimately contributing to more innovative and effective educational and research practices in bioinformatics.

1.2 Chapters Outline

This thesis is structured into several key chapters, each building upon the findings of the previous sections to present a comprehensive study of VR in scientific visualisation, specifically focusing on its applications in protein design. The chapters are meticulously crafted to guide the reader through the research journey, from the foundational literature

to the development and evaluation of innovative VR tools, ultimately leading to a broader discussion of the implications and future directions of this field.

Introduction The introduction chapter sets the stage for the research by outlining the main objectives, significance, and the broader impact of integrating VR into scientific visualisation. It provides an overview of the thesis structure and introduces the key outcomes expected from the research. This chapter emphasises the transformative potential of VR in enhancing the visualisation and manipulation of protein structures, offering new opportunities for education and research in bioinformatics and protein design.

Literature Review This chapter provides a comprehensive overview of existing research on VR technologies and their application in scientific visualisation, with a specific focus on molecular biology and protein design. This chapter is structured to highlight the current state of VR in education and research, identifying the gaps that this study aims to address. The review is conducted using a systematic scoping review methodology, following the PRISMA-ScR (Preferred Reporting Items for Systematic reviews and Meta-Analyses extension for Scoping Reviews) guidelines [1]. This ensures a thorough and unbiassed analysis of the relevant studies, providing a solid foundation for the subsequent development of VR tools and methodologies. The chapter also explores the theoretical underpinnings of VR in scientific visualisation, examining how immersive technologies can enhance the understanding of complex biological structures and processes.

Background The Background chapter delves into the fundamental concepts and technologies that underpin the research. This chapter is crucial for setting the context for the VR tools developed in later chapters, providing a detailed explanation of the biological, computational, and technological aspects relevant to protein structure visualisation and design. It covers the basics of protein folding, Unity, Blender, and educational VR. Additionally, it reviews the technical aspects of VR systems, including hardware requirements, software frameworks, and the principles of immersive technology, offering a comprehensive understanding of the tools and techniques that form the backbone of this study.

Single-player Gaming This chapter introduces and details the development of Pocket Peptides, an educational game designed to enhance the learning and visualisation of protein structures. This chapter is pivotal as it bridges the gap between theoretical concepts and practical applications of VR in education. Pocket Peptides is a single-player game that immerses users in a virtual environment where they can interact with 3D models of proteins by drag&drop, bend&twist, and grab&throw to solve scientific puzzles and explore their structures, folding patterns, and functional sites.

The development process of "Pocket Peptides" is outlined in this chapter, covering the conceptualisation, design, and implementation phases. The chapter discusses the pedagogical theories that informed the game design, emphasising the importance of active learning and gamification in enhancing educational outcomes. The game mechanics are explained in detail, highlighting how the tasks and challenges within the game are aligned with key learning objectives in molecular biology.

Multi-player Gaming This chapter is dedicated to the technical development of the VR tool introduced in this research. It discusses the design and creation of "ProMVR", the multiplayer environments, this chapter provides a detailed account of the software and hardware used, the challenges encountered during development, and how these challenges were overcome.

This chapter also presents the evaluation results of the developed VR tool. Through rigorous testing and analysis, this research provides a detailed understanding of the performance dynamics of VR systems in educational and research settings. The findings on latency and frame rate in single and multi-PC configurations offer valuable insights for the optimisation of VR systems.

Discussion This final chapter also summarises the key findings of the research, reiterating the significance of VR in advancing scientific visualisation. It reflects on the broader impact of the study, particularly in how it might influence future educational practices and research methodologies in bioinformatics and molecular biology.

The discussion chapter interprets the findings from the evaluation, placing them in the context of existing literature and the broader field of molecular biology. It explores the implications of VR integration in protein design and education, discusses potential limitations of the study, and suggests areas for future research.

1.3 Outcomes

The outcomes of this research are multifaceted, contributing to both academic literature and practical applications.

Publications and Academic Articles The research produced and published several high-impact academic articles detailing the development and evaluation of VR tools for protein visualisation. These publications contribute to the growing body of literature on the application of VR in scientific research and education, offering new insights into its potential and limitations. These articles are "Pepblock builder VR–an open-source tool for gaming-based bio-edutainment in interactive protein design", "Schedio-pro: An interactive virtual reality game for protein design", "ProMVR-Protein Multiplayer Virtual Reality Tool", and "ProMVR-Interactive Protein Educational Tool" [2] [3] [4] [5].

Development of Educational Tools The research results in the creation of gamified and educational tools, including Pocket Peptides (renamed as Pepblock Builder VR), and ProMVR platform [2]. These tools are designed to enhance scientific outreach and engagement by providing interactive, gamified environments where users can explore and manipulate protein structures. These tools have the potential to become integral components of bioinformatics curricula, fostering a new generation of molecular biologists who are adept at using cutting-edge technologies.

The primary tool developed through this research is ProMVR, a VR platform designed specifically for multi-user, immersive protein visualisation [4]. This platform supports multiplayer functionalities, allowing researchers from different locations to work together in a shared virtual environment. This aspect of the research is particularly significant for fostering collaboration in the scientific community, breaking down geographical barriers, and enabling more dynamic and interactive research methodologies. ProMVR has been tested and validated in both educational and research environments, showcasing its potential as a transformative tool in molecular biology. Additionally, supporting tools such as the integration of Unity with the Mirror networking library for VR synchronisation have been developed, providing a robust framework for future VR applications in scientific fields.

Contributions and Future Research Directions Beyond the specific tools and platforms developed, this research contributes to the broader field of scientific visualisation by demonstrating the practical applications of VR in molecular biology. The methodologies and approaches developed in this thesis can be adapted and applied to other areas of scientific research, where complex data and models need to be visualised and understood.

The outcomes of this research will also open new avenues for future research. By demonstrating the feasibility and effectiveness of VR in protein visualisation, the thesis sets the stage for further exploration into other areas of protein design and beyond. Future research could explore the integration of Augmented Reality (AR) with VR, the use of AI-driven models with VR environments, or the expansion of these tools to other fields such as genomics or neurobiology.

1.4 Author's Publications

All the four papers below are published. They are accessible and retrievable online.

- **Pepblock Builder VR – An Open Source Tool for Gaming-Based Bio-Edutainment in Interactive Protein Design** VVB Yallapragada, T Xu, SP Walker, S Tabirca, M Tangney. This paper has been published in the Journal of Frontiers in Bioengineering and Biotechnology [2].

- **Schedio-pro: An interactive virtual reality game for protein design** T Xu, VVB Yallapragada, M Tangney, S Tabirca. This paper has been published in the Book of International Conference in Communications, Signal Processing, and Systems [3].

- **ProMVR-Protein Multiplayer Virtual Reality Tool** T Xu, VVB Yallapragada, M Tangney, S Tabirca. This paper has been published in the Book of Proceedings of the 27th the Association for Computing Machinery (ACM) Symposium on Virtual Reality Software and Technology [4].

- **ProMVR-Interactive Protein Educational Tool** T Xu, VVB Yallapragada, M Tangney, S Tabirca. This paper has been published in the Book of Proceedings

of the 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics [5].

1.5 Research Scope, Contributions, and Conclusion

Research Questions The central questions for this research were:

- How can VR advance protein visualisation and design?
- What are the most effective VR features and methodologies that significantly benefit protein visualisation and design?

This research aimed to explore the transformative potential of integrating VR technologies into visualising and manipulating protein structures. By highlighting its potential to revolutionise educational frameworks and research methodologies in bioinformatics and structural biology, it inspires a sense of hope and excitement for the future of these fields.

Research Method Following the PRISMA-ScR guidelines, the scoping review approach was employed to systematically identify and analyse current applications and studies focusing on VR's role in protein visualisation. The review examined recent literature and existing technologies, evaluating VR applications' educational and practical impacts on protein design. The study identified vital VR tools and applications, such as UCSF Chimera, Rosetta, I-TASSER, Foldit, and PolyFold, each contributing uniquely to the visualisation and manipulation of protein structures. These tools were evaluated based on their capabilities in rendering, predicting, and simulating protein structures and their potential for educational purposes.

The findings demonstrated that VR's immersive, intuitive, and interactive visual representations profoundly comprehend complex biological structures, offering a hands-on experience that traditional two-dimensional approaches cannot match.

Contributions The research highlighted the significant contribution from the author of gamification in enhancing the learning experience. Tools like Pepblock Builder VR and ProMVR exemplify the integration of gamification elements and collaborative features,

making protein design and visualisation more accessible and engaging for students and researchers alike.

- **Contribution 1: Pepblock Builder VR** – Gamified Protein Design for Education

Pepblock Builder VR represents a significant advancement in the field of bio-edutainment and protein design [2]. By combining VR technology with gaming principles, the tool offers an engaging and accessible platform for learning and experimentation.

Pepblock Builder VR leverages the immersive capabilities of VR to enhance the user's experience in visualising and designing protein structures. The tool provides a unique "LEGO-style" modular design interface, allowing users to construct complex protein structures using basic building blocks, such as helices, coils, and sheets. This approach simplifies the traditionally complex process of protein design, making it accessible to a broader audience. The inclusion of VR creates an immersive experience, which is particularly beneficial for educational purposes, as it allows users to intuitively understand the spatial and structural nuances of protein molecules.

The gamified elements of Pepblock Builder VR are designed to engage users in learning through interactive challenges and levels that progressively increase in complexity. This approach not only makes learning more enjoyable but also fosters deeper understanding by encouraging users to experiment with protein structures and observe the consequences of their design choices in real time. The tool's educational potential is further enhanced by its ability to visualise proteins in 3D, offering a comprehensive view that aids in understanding protein folding and interactions.

Pepblock Builder VR is developed using Unity and Blender for graphics and game mechanics, with support for both desktop and VR environments. The tool is compatible with Oculus Rift, providing an immersive experience for VR users. The interface is designed to be intuitive, with features such as drag-and-drop, rotation, and manipulation of protein structures.

The tool's ability to visualise and manipulate protein structures in real time is supported by its integration with computational tools like I-TASSER for protein modelling. While the current version supports a limited set of permutations for protein structure modelling, the author plans to expand its capabilities in future iterations (ProMVR), potentially integrating cloud-based libraries and multiplayer functionalities. One of the primary challenges of Pepblock Builder VR is the time required for real-time modelling

of complex protein structures, which can lead to long wait times. The author acknowledges this limitation and propose future improvements (addressed in ProMVR) to reduce computation times.

- **Contribution 2: ProMVR** – Real-time Scientific Protein Visualisation and Multiplayer VR

ProMVR builds upon the educational and visualisation foundations established by Pepblock Builder VR by addressing its core limitations in real-time protein rendering, collaborative interaction, and scientific scalability [4] [5]. While Pepblock Builder VR introduced gamification and intuitive modular construction of protein structures in a LEGO-style format—particularly effective for education—it remained limited to pre-rendered models and local, single-user exploration. The ProMVR system, by contrast, introduces three key innovations that represent substantial advancements in VR-driven protein design:

- 1. Real-Time Protein Model Retrieval and Rendering via Parametric Protein Search Engine (PSGO):

ProMVR integrates with PSGO, a parametric protein search engine developed by Yanlin Mi [6], enabling dynamic querying, retrieval, and 3D rendering of PDB files directly from a server-based infrastructure. Unlike the static or manually preprocessed content of its predecessor, ProMVR automates the protein visualisation pipeline by incorporating UCSF Chimera on the server side to convert PDB data into 3D OBJ models. These are streamed back to the client for immediate in-VR exploration. This process eliminates the bottleneck of preloading and enhances ProMVR's scientific applicability for real-world research scenarios, offering up-to-date access to expansive protein datasets.

- 2. Scalable Multiplayer Collaboration in VR:

Building on insights from post-pandemic remote collaboration needs, ProMVR supports synchronised, real-time multiplayer interaction using the Mirror networking library. Users can concurrently explore, manipulate, and discuss protein structures in a shared virtual space—whether seated around a "virtual lab table" or interacting with models via grab-and-throw mechanics. Integrated with Vivox voice and text communication, this

functionality transforms ProMVR into an integrated scientific collaboration social platform, fostering collaborative hypothesis generation, guided tutorials, and joint molecular design.

- 3. Technical Innovation and Scientific Readiness:

From a systems perspective, ProMVR addresses previous challenges such as limited runtime file parsing and local computation constraints by offloading data transformation and structure generation to server-side processes. These improvements not only improve load time and latency but also enable future features such as room-based session sharing, role-based control (e.g., instructor vs. students), and dynamic parameter-based model loading. Combined with its adherence to modern Unity networking practices and efficient mesh streaming, ProMVR is well-positioned for deployment in both educational and research institutions.

Key Findings Throughout this research, several key findings emerged. Firstly, VR's immersive and interactive capabilities significantly enrich the educational experience by providing a hands-on approach to studying complex protein structures. This not only aids in better comprehension but also increases cognitive retention of intricate biological concepts. Moreover, VR applications facilitate collaborative learning environments through multiplayer functionalities, potentially transforming educational paradigms by fostering a more engaging and interactive learning atmosphere.

Conclusion The introduction of VR into scientific visualisation represents a significant step forward in how complex biological data is interpreted and understood. By developing and rigorously evaluating Pepblock Builder VR and ProMVR, this thesis contributes to the advancement of VR technologies in molecular biology, offering a new platform for both research and education. The outcomes of this research are poised to influence future developments in VR, paving the way for more interactive and immersive learning experiences in science.

In conclusion, this research underscores the transformative potential of VR in molecular biology, particularly in protein visualisation and education. The findings advocate for continued exploration and development of VR applications in this field, emphasising the need for innovative solutions to overcome existing challenges. Future studies should

focus on the longitudinal impacts of VR in education, its scalability, and its integration with other technological advancements to further enhance its utility and accessibility in scientific visualisation.

2 Literature Review

2.1 Introduction

Advancements in VR technologies have opened new frontiers in scientific visualisation and educational methodologies [7], particularly in molecular biology. This study explores the integration of VR into the visualisation and manipulation of protein structures, underscoring its potential to revolutionise educational frameworks and research methodologies in bioinformatics [8] and structural biology [9].

VR's capability to create immersive, intuitive, and interactive visual representations allows a more profound comprehension of complex biological structures like proteins [10]. By employing VR, users can manipulate and analyse protein structures in three-dimensional space, offering a hands-on experience that traditional two-dimensional approaches cannot match [11]. This experiential learning tool not only enhances cognitive retention [12] of complex biological concepts but also fosters a more engaging learning environment [13], sparking excitement about the potential of VR in education[14].

This research adopts an innovative scoping review approach using the PRISMA-ScR guidelines to identify and analyse current applications and studies focusing on VR's role in protein visualisation [1]. This study reviews recent literature and existing technologies to evaluate VR applications' educational and practical impacts on protein design [15]. It mainly focuses on how these technologies can be adapted for more effective teaching and intuitive research in molecular biology [16].

Examples of VR applications in other fields are discussed to draw parallels and propose novel uses in protein visualisation [17] [18] [19] [20] [21]. This includes the examination of interactive VR environments that incorporate real-time data visualisation and multiplayer functionalities, potentially transforming educational settings by allowing collaborative and interactive learning experiences [22] [23].

Ultimately, this review paves the way for a comprehensive exploration of VR's potential to transform protein visualisation into a more accessible and engaging process. This transformation has the power to significantly enhance the understanding and innovation in molecular biology education and research, instilling a sense of hope and anticipation for the future of their research [24] [25].

2.2 Methodology

This study adheres to the PRISMA-ScR (Preferred Reporting Items for Systematic Reviews and Meta-Analyses Extension for Scoping Reviews) guidelines, focusing on protein visualisation and virtual reality [1]. Microsoft Excel spreadsheets were utilised to manage and organise the extracted information [26]. Google Scholar's alert service monitored and retrieved relevant publications from various databases [27] [28] [29] [30]. By setting up alerts with keywords such as "protein visualisation" and "VR", numerous recent papers were collected [31]. Applying the PRISMA-ScR guidelines ensured a systematic and thorough scoping literature review process [1], as illustrated in the Figure 2.1 below.

2.2.1 Review Planning

This phase concentrates on the preliminary preparations necessary to accomplish the objectives of this literature review [32].

Review Rationale Following the PRISMA-ScR guidelines, the review articles were examined to ascertain whether they addressed the research question concerning the benefits of virtual reality in protein visualisation and design [33]. None of the existing reviews specifically explored the integration of protein visualisation with VR [34]. As a result, this review will focus on VR applications in protein visualisation, emphasising aspects such as real-time visualisation of protein data, interactive VR environments, and multiplayer functionalities [35] [36].

Research Question Drawing on existing research in VR application within protein design, this study formulates a crucial research question: How can VR advance protein

Preferred Reporting Items for Systematic reviews and Meta-Analyses extension for Scoping Reviews (PRISMA-ScR) Checklist

SECTION	ITEM	PRISMA-ScR CHECKLIST ITEM	REPOR ON PA
TITLE			
Title	1	Identify the report as a scoping review.	
ABSTRACT			
Structured summary	2	Provide a structured summary that includes (as applicable): background, objectives, eligibility criteria, sources of evidence, charting methods, results, and conclusions that relate to the review questions and objectives.	
INTRODUCTION			
Rationale	3	Describe the rationale for the review in the context of what is already known. Explain why the review questions/objectives lend themselves to a scoping review approach.	
Objectives	4	Provide an explicit statement of the questions and objectives being addressed with reference to their key elements (e.g., population or participants, concepts, and context) or other relevant key elements used to conceptualize the review questions and/or objectives.	
METHODS			
Protocol and registration	5	Indicate whether a review protocol exists; state if and where it can be accessed (e.g., a Web address); and if available, provide registration information, including the registration number.	
Eligibility criteria	6	Specify characteristics of the sources of evidence used as eligibility criteria (e.g., years considered, language, and publication status), and provide a rationale.	
Information sources*	7	Describe all information sources in the search (e.g., databases with dates of coverage and contact with authors to identify additional sources), as well as the date the most recent search was executed.	
Search	8	Present the full electronic search strategy for at least 1 database, including any limits used, such that it could be repeated.	
Selection of sources of evidence†	9	State the process for selecting sources of evidence (i.e., screening and eligibility) included in the scoping review.	
Data charting process‡	10	Describe the methods of charting data from the included sources of evidence (e.g., calibrated forms or forms that have been tested by the team before their use, and whether data charting was done independently or in duplicate) and any processes for obtaining and confirming data from investigators.	
Data items	11	List and define all variables for which data were sought and any assumptions and simplifications made.	
Critical appraisal of individual sources of evidence§	12	If done, provide a rationale for conducting a critical appraisal of included sources of evidence; describe the methods used and how this information was used in any data synthesis (if appropriate).	
Synthesis of results	13	Describe the methods of handling and summarizing the data that were charted.	

SECTION	ITEM	PRISMA-ScR CHECKLIST ITEM	REPORTED ON PAGE
RESULTS			
Selection of sources of evidence	14	Give numbers of sources of evidence screened, assessed for eligibility, and included in the review, with reasons for exclusions at each stage, ideally using a flow diagram.	
Characteristics of sources of evidence	15	For each source of evidence, present characteristics for which data were charted and provide the citations.	
Critical appraisal within sources of evidence	16	If done, present data on critical appraisal of included sources of evidence (see item 12).	
Results of individual sources of evidence	17	For each included source of evidence, present the relevant data that were charted that relate to the review questions and objectives.	
Synthesis of results	18	Summarize and/or present the charting results as they relate to the review questions and objectives.	
DISCUSSION			
Summary of evidence	19	Summarize the main results (including an overview of concepts, themes, and types of evidence available), link to the review questions and objectives, and consider the relevance to key groups.	
Limitations	20	Discuss the limitations of the scoping review process.	
Conclusions	21	Provide a general interpretation of the results with respect to the review questions and objectives, as well as potential implications and/or next steps.	
FUNDING			
Funding	22	Describe sources of funding for the included sources of evidence, as well as sources of funding for the scoping review. Describe the role of the funders of the scoping review.	

JBI = Joanna Briggs Institute; PRISMA-ScR = Preferred Reporting Items for Systematic reviews and Meta-Analys extension for Scoping Reviews.
* Where *sources of evidence* (see second footnote) are compiled from, such as bibliographic databases, social m platforms, and Web sites.
† A more inclusive/heterogeneous term used to account for the different types of evidence or data sources (e.g., quantitative and/or qualitative research, expert opinion, and policy documents) that may be eligible in a scoping review as opposed to only studies. This is not to be confused with *information sources* (see first footnote).
‡ The frameworks by Arksey and O'Malley (6) and Levac and colleagues (7) and the JBI guidance (4, 5) refer to t process of data extraction in a scoping review as data charting.
§ The process of systematically examining research evidence to assess its validity, results, and relevance before using it to inform a decision. This term is used for items 12 and 19 instead of "risk of bias" (which is more applicat to systematic reviews of interventions) to include and acknowledge the various sources of evidence that may be u in a scoping review (e.g., quantitative and/or qualitative research, expert opinion, and policy document).

From: Tricco AC, Lillie E, Zarin W, O'Brien KK, Colquhoun H, Levac D, et al. PRISMA Extension for Scoping Reviews (PRISMA-ScR): Checklist and Explanation. Ann Intern Med. ;169:467–473. doi: 10.7326/M18-0850

Figure 2.1: PRISMA-ScR (PRISMA-ScR) Framework for Systematic Literature Review illustrates the methodological rigor applied in this scoping review of VR-based protein visualization tools, featuring: (1) Protocol-driven screening of records from ACM/IEEE/Google Scholar databases, (2) Multi-stage filtering (title/abstract/full-text) yielding eligible studies, and (3) Thematic analysis of tools (UCSF Chimera), gamified systems (Foldit), and educational VR applications (ProteinVR). The checklist ensures compliance with standards for evidence synthesis in emerging tech domains. Source: [`https://www.prisma-statement.org/scoping`]. This figure is reproduced under fair use for academic purposes.

visualisation and design? To tackle this question, the study thoroughly analyses relevant literature, scrutinising the publication year and technological features of the VR setups, such as the platforms and specific types of VR headsets [37].

For example, distinctions are made between uses of immersive VR systems like Oculus Rift, which provide extensive interaction capabilities, and non-immersive desktop systems, which may focus more on detailed visualisation with less interactive depth [38]. Other research might focus on gamification, where VR platforms create educational games that teach users about protein folding and design through hands-on experience [39].

The study aims to identify the most compelling VR features and methodologies that can significantly benefit protein visualisation and design by comparing these diverse applications and their technological setups [40]. This includes understanding how VR can make the learning process more intuitive, engaging, and accessible, inspiring a new wave of research in molecular biology [41].

Review Protocol Per the PRISMA-ScR guidelines, establishing a review protocol is essential for conducting a scoping review. This protocol is a foundational framework that guides the review process, ensuring methodological rigour and reducing the risk of research bias [42] [43]. This preemptive measure minimises potential biases. Formal and informal search strategies were employed to address the research question comprehensively [44].

This research strategy is comprehensive, employing a dual approach to searches. Formal searches typically involve systematic queries in established databases, while informal searches may include references from the identified articles or consultations with experts to uncover additional relevant studies [45]. This thorough approach, which is particularly effective, ensures a broad and thorough exploration of the literature, thereby enhancing the validity and scope of the review findings [46].

The databases are ACM Digital Library, Google Scholar, and IEEE Explore [47], and the strategies employed in investigating existing studies on VR and protein design guide the methodological approach of this research [48]. The selected databases are explicitly tailored to fields pertinent to the study: virtual reality, protein visualisation, and protein design [49]. This targeted selection facilitates a focused review, ensuring that the most relevant and specialised sources are consulted. Subsequently, initial findings

from these preliminary studies were instrumental in shaping the research framework and formulating a precise research question [50]. This systematic approach facilitates the establishment of a robust review protocol, enhancing the structural integrity and focus of the research methodology [51] [52].

2.2.2 Conducting the Review

This scoping literature review encompasses several critical stages: developing a comprehensive search strategy, establishing clear study selection criteria, and formulating a detailed data extraction plan. The search strategy involves specifying keywords and database filters to identify relevant studies within VR and protein design fields effectively. Study selection criteria are then applied to ensure the inclusion of high-quality and pertinent research. At the same time, the data extraction plan outlines the specific data points to be gathered from these studies, such as study methodologies, outcomes, and critical findings. This structured approach ensures a thorough and systematic scoping review of the existing literature.

Search Strategy The formulation of the research question significantly influenced the development of the search strategy for this study. Initially, relevant search keywords were meticulously selected to ensure the precise targeting of applicable articles and papers related to the topic. These keywords aligned with the research's central themes: VR and protein design. The chosen search terms were strategically used in various combinations employing boolean operators to refine the search process effectively. Keywords included combinations like "virtual reality" OR "VR" AND "protein" OR "molecule" AND "visualisation" OR "design", aiming to optimise the retrieval of pertinent literature.

The selection of keywords was strategically designed to encompass a wide range of techniques utilised in protein visualisation and design, incorporating both formal and informal methods, provided they involve using VR technology as previously defined. The temporal scope of the review was confined primarily to publications around the year 2020, acknowledging the rapid advancements occurring within virtual reality and protein design. The screening process evaluated titles, abstracts, and keywords to ascertain relevance and applicability to the research objectives.

Table 2.1: Selection Criteria

Inclusion Criteria	Exclusion Criteria
The study was conducted in a VR environment	The study was not conducted in a VR environment
The study was relevant to protein visualisation or protein design	The study was not relevant to protein visualisation nor protein design
The study was published around 2020	The study was not published around 2020

Selection Criteria Leveraging the keywords and search strategies mentioned in the "Search Strategy" section, this study identified a selection of articles for review, grounded in their pertinence to the research question, alongside established inclusion and exclusion criteria. A table detailing these criteria is presented below for transparent and structured selection. This framework ensures that each article is rigorously assessed for its relevance and contribution to the themes of VR, protein visualisation, and protein design, streamlining the research process and enhancing the study's methodological rigour (see Table 2.1).

The evaluation process involved a thorough review of titles, abstracts, and keywords to determine the relevance of each article concerning VR, protein visualisation, and protein design. Works such as abstracts, posters, books, and articles lacking VR or protein topics were systematically excluded. Following the application of these criteria, only four articles remained that sufficiently met the requirements for inclusion in this study.

Data Extraction The author developed a systematic framework to facilitate extracting pertinent data from research articles aligned with the study's core research question. This structured scheme encompasses several key data extraction dimensions: the article's title or name, a detailed description of the article's content, and the specific VR headset employed in the research. This approach ensures that all extracted data is relevant and directly supports the study on VR applications in protein visualisation and design.

The methodology section details the systematic planning and execution of the review, including justifying the chosen framework, articulating the research questions, developing the review protocol, and implementing the review process. This section also outlines the procedures for identifying relevant keywords, establishing selection criteria, and the

Table 2.2: Selected Articles

Author's Name & Year	Acronym	Description
McGehee et al. (2020)	PolyFold [53]	An interactive visual simulator for distance-based protein folding
Yallapragada et al. (2021)	Pepblock Builder [2] (Author's publication)	An open-source tool for gaming-based bio-edutainment in interactive protein design
Cassidy et al. (2020)	ProteinVR [54]	A web-based molecular visualisation in virtual reality
Oyelere et al. (2020)	EVRGs [55]	Exploring the trends of educational virtual reality games

method for extracting data. These methodologies ensure a structured approach to gathering and analysing information, which is carefully aligned with the study's objectives. The results derived from these comprehensive processes are subsequently detailed in the "Review Results" section, showcasing the outcomes of the applied methodologies.

2.2.3 Review Results and Differentiation from this Thesis

Four papers were identified and evaluated through the implementation of this scoping review methodology, and they are summarised in the Table 2.2.

PolyFold This paper presents an innovative tool designed to enhance the understanding of protein folding processes through interactive and real-time simulations. PolyFold is specifically tailored to simulate distance-based protein folding, leveraging inter-residue distance matrices to drive the folding process. This tool represents a significant advancement over traditional static molecular visualisation tools, offering a dynamic perspective that captures the folding process as it occurs. The software is developed to be accessible and user-friendly, supporting cross-platform use on Macintosh Operating System (macOS), Linux, and Windows [53].

One of the practical challenges in protein folding is the presence of noisy data, which refers to inaccurate or imperfect information about the spatial distances between amino acid residues. These distances, commonly represented as distance matrices, where each entry estimates how far apart two residues are in the folded 3D structure, often contain measurement errors, prediction uncertainty, or missing data—especially when derived from experimental methods such as cryo-EM or NMR. To evaluate its robustness under such conditions, PolyFold's authors introduced artificial Gaussian noise into clean distance matrices during testing. This approach simulates real-world uncertainty and allows researchers to assess whether the algorithm can still predict accurate 3D protein folds despite partial unreliability in the input.

PolyFold's optimisation engine, which employs gradient descent and simulated annealing techniques, shows resilience to noise, successfully predicting accurate protein folds, even when significant noise is introduced. This capability is critical, as real-world data often contain imperfections that can challenge less robust models.

Benchmarking results further confirm its effectiveness. When using near-native distance matrices, PolyFold outperformed established protein structure prediction methods like I-TASSER and Robetta in terms of average TM-score. However, direct comparisons should be interpreted cautiously, as tools like I-TASSER and Robetta often benefit from additional structural constraints, such as homologous templates or fragment libraries, which PolyFold does not use [56].

PolyFold's interactive interface is another notable feature, offering users the ability to manipulate molecular geometries in real-time. This interactivity not only facilitates a deeper understanding of protein folding mechanics but also allows users to explore the effects of varying optimisation parameters on the folding outcomes. Such features make PolyFold a valuable educational tool, potentially fostering broader participation in protein folding research by making complex processes more tangible and understandable to non-experts.

Despite its strengths, the study acknowledges areas for future improvement. The authors suggest enhancements to graphical user interface, such as more advanced rotation capabilities and the integration of features to manipulate and de-noise predicted distance matrices interactively. These improvements could further extend the utility of PolyFold, making it an even more versatile tool for both research and educational purposes [57] [58].

In conclusion, PolyFold stands out as a powerful simulator for distance-based protein folding, offering real-time visualisation and robust performance in handling noisy data. Its combination of accessibility, interactive features, and advanced optimisation algorithms makes it a promising tool for researchers and educators alike, contributing to the advancement of the understanding of protein structure and dynamics.

Pepblock Builder This paper presents an innovative approach to protein visualisation and design through a VR platform that integrates gamification with bio-educational content. This tool is aimed at democratising access to protein design by making it engaging and accessible to users with varying levels of technical expertise, from novices to experienced researchers [2].

Pepblock Builder VR leverages the immersive capabilities of VR to enhance the user's experience in visualisng and designing protein structures. The tool provides a unique "LEGO-style" modular design interface, allowing users to construct complex protein structures using basic building blocks, such as helices, coils, and sheets. This approach simplifies the traditionally complex process of protein design, making it accessible to a broader audience. The inclusion of VR creates an immersive experience, which is particularly beneficial for educational purposes, as it allows users to intuitively understand the spatial and structural nuances of protein molecules [59].

The gamified elements of Pepblock Builder VR are designed to engage users in learning through interactive challenges and levels that progressively increase in complexity. This approach not only makes learning more enjoyable but also fosters deeper understanding by encouraging users to experiment with protein structures and observe the consequences of their design choices in real time. The tool's educational potential is further enhanced by its ability to visualise proteins in 3D, offering a comprehensive view that aids in understanding protein folding and interactions.

Pepblock Builder VR is developed using Unity and Blender for graphics and game mechanics, with support for both desktop and VR environments. The tool is compatible with Oculus Rift, providing an immersive experience for VR users. The interface is designed to be intuitive, with features such as drag-and-drop, rotation, and manipulation of protein structures, making it accessible to users with limited technical skills [60].

The tool's ability to visualise and manipulate protein structures in real time is supported by its integration with computational tools like I-TASSER for protein modelling.

While the current version supports a limited set of permutations for protein structure modelling, the authors plan to expand its capabilities in future iterations, potentially integrating cloud-based libraries and multiplayer functionalities.

One of the primary challenges noted in the paper is the time required for real-time modelling of complex protein structures, which can lead to long wait times. The authors acknowledge this limitation and propose future improvements to reduce computation times. Additionally, the current version includes only ten game levels, with plans to introduce more challenges and expand the scope of the tool to support direct design of structures based on specified in silico parameters. The implementation of cloud-based libraries for user-designed proteins and multiplayer capabilities could further enhance the tool's educational value and collaborative potential.

The authors also highlight the potential of Pepblock Builder VR as a citizen science platform, where users can contribute to protein design and share their creations with the community. This vision aligns with the broader trend of using gamification to engage non-experts in scientific research, exemplified by successful projects like Foldit [61].

Pepblock Builder VR represents a significant advancement in the field of bio-edutainment and protein design. By combining VR technology with gaming principles, the tool offers an engaging and accessible platform for learning and experimentation. While there are challenges to address, such as modelling efficiency and level expansion, the tool's potential to democratise protein design and foster a broader understanding of molecular biology is promising. As VR technology continues to evolve, tools like Pepblock Builder VR will likely play an increasingly important role in science education and research.

ProteinVR This paper introduces an innovative, web-based platform for molecular visualisation using VR. This platform addresses the challenges faced by traditional molecular visualisation tools, offering a novel approach to enhancing both research and education in molecular biology [54].

ProteinVR stands out due to its web-based nature, which allows for easy access across multiple devices without requiring extensive software installations. This is particularly advantageous in educational settings, where ease of access and usability can significantly enhance learning experience. The platform supports various VR setups, enabling users to

explore molecular structures in stereoscopic 3D, thus providing an immersive experience that is crucial for understanding complex spatial arrangements within molecules.

One of the most notable features of ProteinVR is its compatibility with a range of devices, from high-end VR headsets to simple mobile devices. This accessibility ensures that users can engage with the platform regardless of their technical resources, making it a versatile tool for both educators and researchers. By utilising modern web technologies and the WebVR API, ProtienVR effectively democratises access of advanced molecular visualisation capabilities.

ProteinVR leverages BabylonJS, a powerful game engine that facilitates the rendering of molecular structures in 3D environments. The platform's reliance on web technologies ensures broad compatibility and future-proofing, as it can easily adapt to advancements in VR and web standards. The integration with 3Dmol.js allows for dynamic molecular representations, enabling users to switch between different visualisation styles and molecular components seamlessly [62].

The use of pre-rendered 3D environment created with Blender further enhances the user experience by providing contextual backdrops that help users orient themselves spatially. This feature is particularly beneficial in VR mode, where immersive environments can reduce VR-induced discomfort and enhance the sense of presence.

ProteinVR's potential impact on education is significant, as it provides an interactive platform for students to explore molecular structures in an engaging and intuitive manner. The platform's ease of use and accessibility make it an ideal tool for large classroom settings, where traditional VR applications might be impractical. Additionally, the ability to share molecular scenes via URLs facilitates collaborative learning and research, allowing educators and researchers to disseminate complex information effectively [50].

In the research domain, ProteinVR offers a unique opportunity to visualise and analyse molecular interactions in 3D, providing insights that are difficult to achieve with traditional 2D methods. The platform's ability to simulate depth and spatial relationships between molecular components enhances the understanding of protein-ligand interactions, making it a valuable asset for drug discovery and molecular biology research.

Despite its many strengths, ProteinVR faces challenges common to web-based applications such as limitations in rendering performance compared to desktop applications. However, the ongoing development of web technologies promises to mitigate these issues over time. The paper suggests potential future enhancements, such as augmented

reality (AR) integration and improved molecular interaction visualisation, which could further extend the platform's capabilities and appeal.

In conclusion, ProteinVR represents a significant advancement in molecular visualisation technology, offering an accessible and immersive platform for both education and research. Its web-based nature, coupled with robust VR capabilities, positions it as a transformative tool in the field of molecular biology. Continued development and user feedback will be crucial in refining the platform and expanding its impact across scientific and educational communities.

Educational Virtual Reality Games (EVRGs) The paper provides a comprehensive examination of the current state and trends in EVRGs. This systematic review aims to bridge the knowledge gap by exploring the technological, pedagogical, and gaming characteristics of EVRGs, identifying their applications in various educational settings, and suggesting future research directions [55].

The study highlights the transformative potential of VR technology in education, emphasising its ability to create immersive and interactive learning environments that facilitate a deeper understanding of complex subjects. The authors review 31 articles published in high-impact journals and educational conference proceedings to analyse the technological, pedagogical, and gaming characteristics of contemporary EVRGs [63].

One of the significant findings of the study is the dominance of Oculus Rift and HTC Vive headsets in EVRGs, which underscores the importance of immersive technology in enhancing educational experiences. The review reveals that EVRGs are predominantly applied across various educational levels, including tertiary, K-12, and lifelong learning, with a primary focus on out-of-class use. These findings indicate a growing trend towards informal learning environments where students can explore and interact with educational content at their own pace.

The pedagogical applications of EVRGs are diverse, with healthcare education emerging as the most prominent area of focus. This focus is attributed to the ability of VR to simulate realistic medical scenarios, providing students with practical experience in a safe and controlled environment. The study also notes the potential of EVRGs to support students with different learning styles, enhance cognitive achievement, improve spatial thinking, and facilitate collaboration.

Another significant trend is the use of VR games for language learning and cultural education. VR environments provide a unique opportunity for immersive language practice and cultural exploration, enabling students to engage in realistic conversations and experience cultural settings virtually. The review cites several studies demonstrating the effectiveness of VR games in improving language proficiency and cultural awareness, suggesting that these tools can complement traditional language learning methods.

Despite the promising application of EVRGs, the review identifies several challenges that need to be addressed to maximise their potential in education. These include technical limitations such as the high cost of VR equipment, the need for specialised skills to develop VR content, and the potential for physical discomfort during prolonged use. Additionally, the study highlights the limited application of EVRGs in mainstream education and arts, suggesting that future research should explore their potential in these areas [64].

Another limitation noted in the review is the variability in the quality of educational VR games. While some VR games are well-designed and pedagogically sound, others may lack educational value or be poorly integrated into the curriculum. This variability can impact the effectiveness of VR games as educational tools and highlights the need for careful selection and evaluation of VR content [29].

The authors emphasise the need for more cross-cultural studies to understand the global applicability of EVRGs and their impact on diverse educational contexts. They also call for research into the long-term effects of VR on learning outcomes and the development of best practices for integrating EVRGs into formal educational settings.

In conclusion, this systematic review provides valuable insights into the current state and potential of VR games in education. The findings suggest that VR games hold significant promise for enhancing student engagement, motivation, and learning outcomes across various educational domains. However, addressing challenges related to cost, technical complexity, and content quality is crucial for the successful integration of VR games into educational practice. Future research should focus on developing affordable VR solutions, providing comprehensive training for educators and establishing standards for the design and evaluation of educational VR games.

The insights influenced by these four articles shaped the author's approach to protein visualisation and protein-protein docking, interactive gaming on PC, and subsequent applications in VR. Inspired by virtual reality's potential in educational settings, the

author considered how multiplayer features could enhance the learning experience. This led to integrating EVRGs into the project, recognising the immense value of collaborative learning environments where participants could share and enhance their educational journey in protein science together.

Key Differentiation from Related Works The insights drawn from these four studies (PolyFold, Pepblock Builder, ProteinVR, and EVRGs) significantly influenced the author's research direction in protein visualisation, ab initio folding, and immersive educational gamification. While prior tools demonstrated individual strengths — such as real-time folding in PolyFold, modular gameplay in Pepblock Builder, or web-based accessibility in ProteinVR — this thesis contributes a novel integration of these paradigms into a unified learning ecosystem. Specifically, the Pocket Peptides game introduced in this thesis builds upon gamified interaction by embedding ab initio folding simulations, modular design mechanics, and progressive tutorial scaffolding tailored to non-experts. Furthermore, the transition from single-player PC to multiplayer VR was not only technical but pedagogically grounded, informed by educational design principles (Section 4.3.3) and usability walkthroughs (Section 4.5). Unlike the generic collaborative frameworks mentioned in EVRGs, this thesis implements a task-based cooperative protein design workflow within a VR environment (Chapter 5), enabling shared manipulation and synchronised feedback — a feature absent from prior studies. This cohesive progression from single-user gameplay to scientifically-informed multi-user interaction in VR demonstrates both conceptual originality and practical innovation, positioning this thesis as a distinct and meaningful extension of existing literature.

2.3 Software Review

2.3.1 General Facts about Protein

Proteins are the workhorses of the cell, involved in virtually every biological process and forming the basis of cellular structure and function. As complex macromolecules, proteins are composed of amino acids linked by peptide bonds, forming polypeptide chains. These chains undergo folding into specific three-dimensional structures, which

determine their function. The field of protein design synthesises this understanding to engineer novel proteins with enhanced or entirely new functionalities [65].

The primary structure of a protein refers to its linear sequence of amino acids, which is encoded directly by the organism's genetic material. This sequence dictates the protein's complex folding patterns and ultimately its function [66]. Amino acids themselves are composed of a central carbon atom bonded to an amino group, a carboxyl group, a hydrogen atom, and a variable side chain, which defines the chemical properties of each amino acid type [67].

The secondary structure arises from hydrogen bonds between the backbone constituents of the amino acids in the polypeptide chain. These interactions lead to the formation of alpha-helices and beta-sheets, which are common folding patterns within the protein structure [68]. Alpha-helices are right-handed coils stabilised by hydrogen bonding between every fourth amino acid, making them highly elastic [69]. Beta-sheets consist of beta strands connected laterally by at least two or three backbone hydrogen bonds, forming twisted, pleated sheets [70] [71].

These secondary elements fold further into a tertiary structure, a unique three-dimensional shape that is stabilised by various interactions, including hydrogen bonds, ionic bonds, hydrophobic effects, and van der Waals forces. The precise three-dimensional structure is crucial as it determines the protein's specificity and function [72]. Some proteins also assemble into quaternary structures, which are complexes formed by several protein molecules (polypeptide chains), often called protein subunits (see Figure 2.2).

Protein design involves the deliberate modification of existing proteins or the creation of novel proteins with desired properties. This is achieved through various methods such as rational design, which involves the use of computational modelling to predict the effects of amino acid substitutions on protein structure and function [73]. Directed evolution, another key technique, mimics natural evolutionary processes in the lab to evolve proteins with enhanced characteristics, such as increased stability or altered substrate specificity [74].

Advances in computational biology have significantly enhanced the ability to visualise and manipulate protein structures. Molecular graphics software allows researchers to view proteins in three-dimensions, which is essential for understanding their function

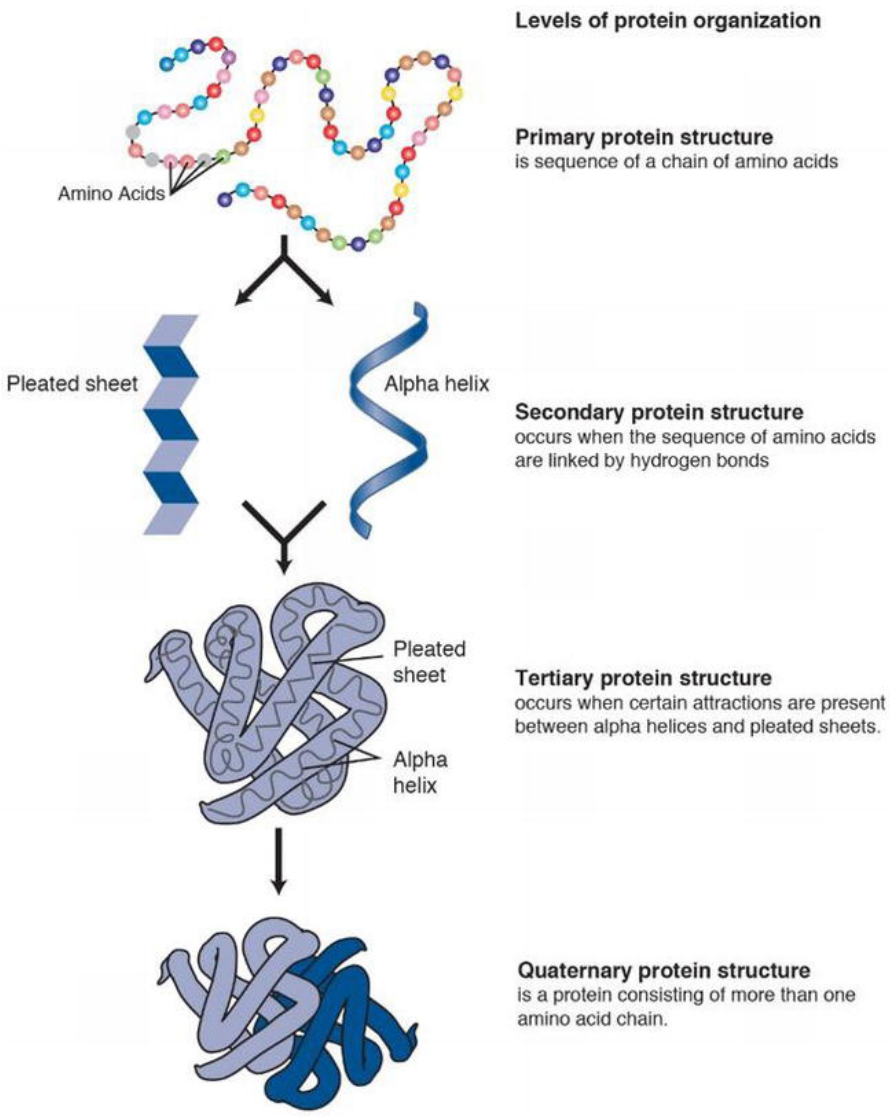


Figure 2.2: Hierarchical Organization of Protein Structures delineates the four fundamental structural levels critical for computational modeling in VR environments: **Primary**: Linear amino acid sequences. **Secondary**: Hydrogen-bond stabilized α-helices/β-sheets. **Tertiary**: 3D conformations (targets for real-time folding simulations). **Quaternary**: Multi-chain assemblies. Source: [`https://openbooks.lib.msu.edu/isb202/chapter/protein-structure-and-function/`]. This figure is reproduced under fair use for academic purposes.

and interaction with other molecules [75]. Software tools enable the rotation, zooming, and dissection of these structures to examine their properties more closely [76].

Two primary file formats are used in protein data handling: PDB and FAST-All. Protein Data Bank (PDB) stores three-dimensional structural data derived from X-ray crystallography and microscopy studies [77]. It provides detailed information about the atomic positions in proteins [78].

FAST-All (FASTA) is a simpler format primarily used for storing sequence data. FASTA files contain the primary structure of proteins, allowing for easy access and manipulation in various bioinformatics applications [79].

The manipulation of proteins in virtual environments poses several challenges, particularly regarding the accuracy and computational efficiency of simulating protein dynamics. Future advancements in hardware and software are expected to improve the realism and speed of protein simulations, making these tools more powerful and accessible to researchers across the globe [80]. Furthermore, the integration of VR technologies offers exciting new avenues for the immersive visualisation and manipulation of molecular structures, potentially transforming how biochemical research is conducted [81].

Understanding and designing proteins at a molecular level holds immense potential for innovations in medicine, environmental science, and biotechnology, underscoring the importance of continued research and development in this dynamic filed.

2.3.2 Protein Visualisation Tools

Fascinated by protein molecules and interested in visualising their structures, the author got suggestions from the team and searched Google, GitHub, and YouTube to find existing protein visualisation tools and similar research; three tools stand out: UCSF Chimera, Rosetta, and I-TASSER.

UCSF Chimera UCSF Chimera is a sophisticated molecular visualisation and analysis tool designed to meet the needs of researchers in structural biology, computational chemistry, and related fields [76]. Developed by the Resource for Biocomputing, Visualisation, and Informatics (RBVI) at the University of California, San Francisco, Chimera offers a robust and flexible platform for the exploration and interpretation of molecular structures and related data [82].

UCSF Chimera stands out for its comprehensive suite of visualisation tools that allow users to interactively view and manipulate molecular models. The software supports various molecular representations, such as ribbons, sticks, and surfaces, which can be customised to highlight specific structural features or interactions [83]. This versatility makes Chimera an invaluable tool for structural analysis, enabling researchers to gain insights into protein folding, binding interactions, and other complex molecular phenomena [76].

One of the most significant strengths of Chimera is its ability to integrate with external databases and tools. It can fetch data directly from the PDB and interface with other computational chemistry software, enhancing its utility in comprehensive structural analyses [84]. This feature streamlines the workflow for researches, enabling seamless transitions between data acquisition, visualisation, and analysis.

The software's extensibility is another notable aspect, allowing users to develop and incorporate custom extensions and scripts using Python. This flexibility caters to the diverse needs of researchers and facilitates the development of tailored workflows for specific research questions. The Chimera Extension Manager provides a user-friendly interface for managing and utilising these extensions, further enhancing the software's versatility and utility.

Chimera offers a user friendly interface that combines graphical menus with command line input, catering to both novice and experienced users. The command line functionality is particularly beneficial for automating repetitive tasks and performing complex operations that require precise control. Additionally, Chimera provides extensive documentation and tutorials, which aid users in mastering the software's capabilities and applying them effectively in their research projects.

Chimera is widely used in academic and industrial research settings for tasks such as structure visualisation, molecular docking, and quantitative analysis of molecular interactions. Its ability to generate high-quality images and animations makes it an excellent tool for preparing figures for publication and presentations. Moreover, Chimera's interactive features are valuable for educational purposes, allowing students to explore molecular structures dynamically and gain a deeper understanding of biomolecular principles.

UCSF Chimera is a versatile and comprehensive visualisation system that has significantly contributed to the fields of structural biology and computational chemistry. Its

combination of powerful visualisation capabilities, user-friendly interface, and integration with external databases and tools makes it an indispensable resource for researchers. Despite some limitations, Chimera's ongoing development and adaptability ensures that it will continue to play a crucial role in molecular research and education.

Rosetta The Rosetta software suite is a highly versatile and powerful tool for macromolecular modelling, playing a crucial role in various aspects of structural biology and protein design [85]. The suite includes multiple protocols and methodologies that enable researchers to predict protein structures, design new proteins, and model complex biomolecular interactions with remarkable accuracy and efficiency [86].

One of the primary strengths of Rosetta lies in its de novo protein structure prediction capabilities. By leveraging fragment-based assembly and Monte Carlo sampling techniques [87] (see Figure 2.3), Rosetta can predict the three-dimensional structures of proteins from their amino acid sequence, even in the absence of homologous templates [88]. This is particularly useful for small to moderately sized proteins (typically fewer than 200 residues). The core principle of Rosetta's methodology is to mimic the interplay between local and global interactions that determine protein structure [89]. By assembling short fragments of known protein structures, Rosetta effectively explores the conformational space of the target protein [90].This strategy is useful for proteins with no detectable homologues in the PDB [91].

Rosetta's fragment assembly approach involves the selection of fragments based on sequence similarity and secondary structure prediction, followed by a Monte Carlo simulated annealing search to assemble these fragments into protein-like structures [92]. The fitness of these conformations is evaluated using a scoring function derived from conformational statistics of known protein structures [93]. This method allows Rosetta to generate accurate models even with limited sequence information, facilitating genome-scale analyses and structural predictions for novel protein families [94].

Rosetta's comparative modelling protocols also provide robust tools for generating accurate models of proteins with known homologues [95]. By aligning target sequences with templates and employing sophisticated loop modelling and refinement strategies (see Figure 2.4), Rosetta can produce models that often surpass the quality of the original templates [96]. This approach is particularly effective for proteins with medium to distant homologues, where sequence identity ranges between 25% and 50% [97].

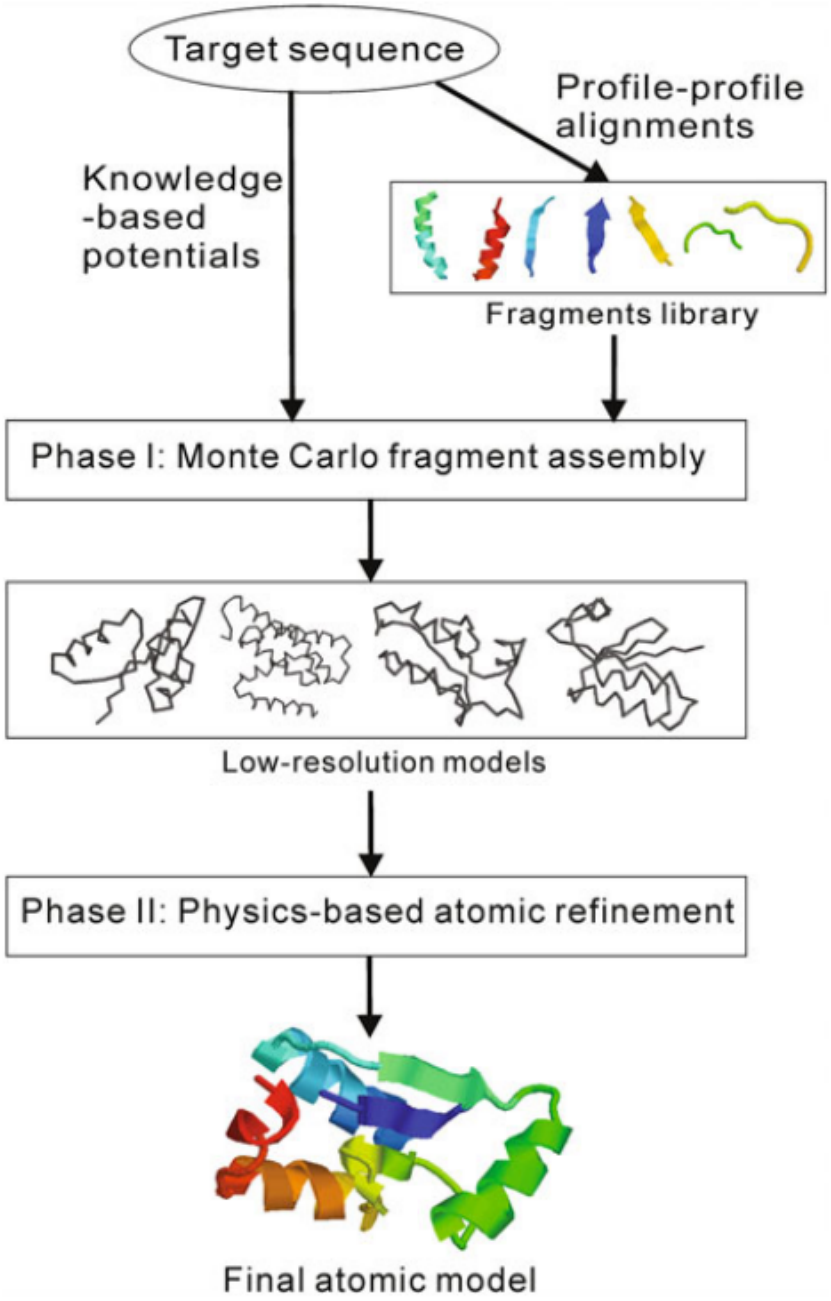


Figure 2.3: Ab Initio Protein Structure Prediction Workflow illustrates Rosetta's fragment-based computational pipeline for VR-compatible protein modeling, beginning with target sequence input through knowledge-based potentials to Monte Carlo fragment assembly and concluding with atomic refinement. Source: [https://www.semanticscholar.org/paper/Chapter-1-Ab-Initio-Protein-Structure-Prediction-Lee-Freddolino/2d958387fa2e6183f8cd5d352cd093fc1964aa68]. This figure is reproduced under fair use for academic purposes.

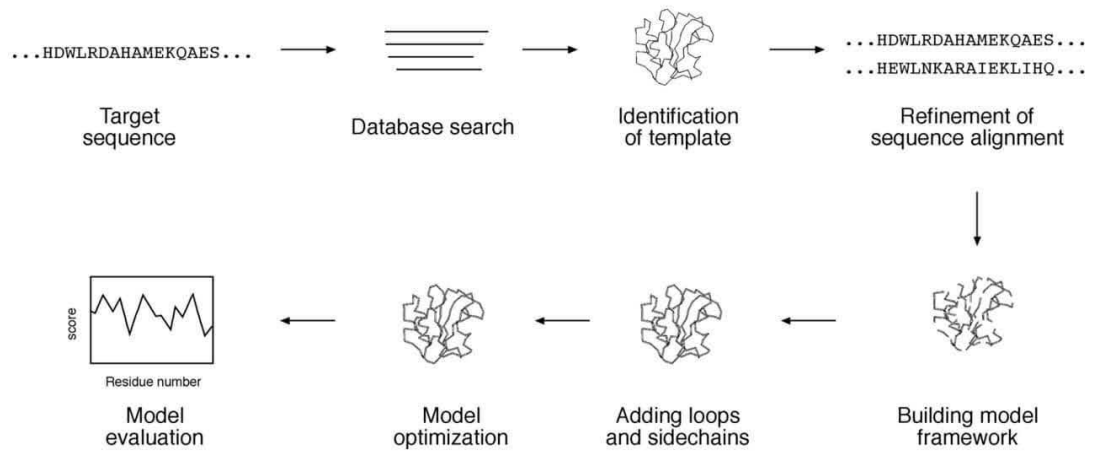


Figure 2.4: Computational Pipeline for Homology Modeling delineates the seven-stage workflow for template-based protein structure prediction, beginning with target sequence input ("HOWLRDAHAMEROABS") through iterative template identification, sequence alignment refinement, 3D model building, loop reconstruction, and culminating in energy minimization. Source: [Sanju Tamang, "Homology Modeling- Definition, Steps, Diagram, Uses," Microbe Notes, 2023, `https://microbenotes.com/homology-modeling/`]. This figure is reproduced under fair use for academic purposes.

The software suite extends beyond mere structure prediction to include powerful tools for protein design [98]. RossettaDesign, for example, enables the identification of amino acid sequences that will fold into desired structures or possess specific functional properties. This capability is essential for engineering proteins with novel functionality, improving enzyme activities, and creating stable protein-protein interfaces [99]. The success of these design protocols has been validated through numerous experimental studies, showcasing Rosetta's ability to generate proteins with enhanced stability and novel binding properties [100].

In addition to structure prediction and design, Rosetta offers comprehensive tools for molecular docking (see Figure 2.5), both for protein-protein and protein-ligand interactions [101]. The docking protocols account for induced fit and conformational flexibility, providing accurate models of biomolecular complexes [102]. These capabilities are critical for drug discovery and the study of complex biochemical pathways [103].

Overall, the Rosetta suite represents a comprehensive and indispensable toolkit for the structural biology community. Its broad applicability, from de novo structure prediction and comparative modelling to protein design and docking, makes it a critical resource for advancing the understanding of molecular biology and for the rational design of novel biomolecules with therapeutic and industrial applications. The continuous development and refinement of Rosetta's methodologies ensure that it remains at the forefront of computational structural biology.

I-TASSER The I-TASSER suite of tools represents a significant advancement in the field of computational structural biology [104]. Developed by Yang Zhang and his team, I-TASSER has become one of the most widely used and successful methods for automated protein structure prediction and functional annotation [105], driven by its robust methodology and consistent performance in community-wide blind experiments [106].

I-TASSER operates on a hierarchical modelling approach that integrates threading, fragment assembly, and structural refinement to predict protein structures and functions [56]. The methodology begins with the identification of structural templates from the PDB using multiple threading alignments [107] (see Figure 2.6). The identified fragments are then reassembled into full-length models through iterative simulations,

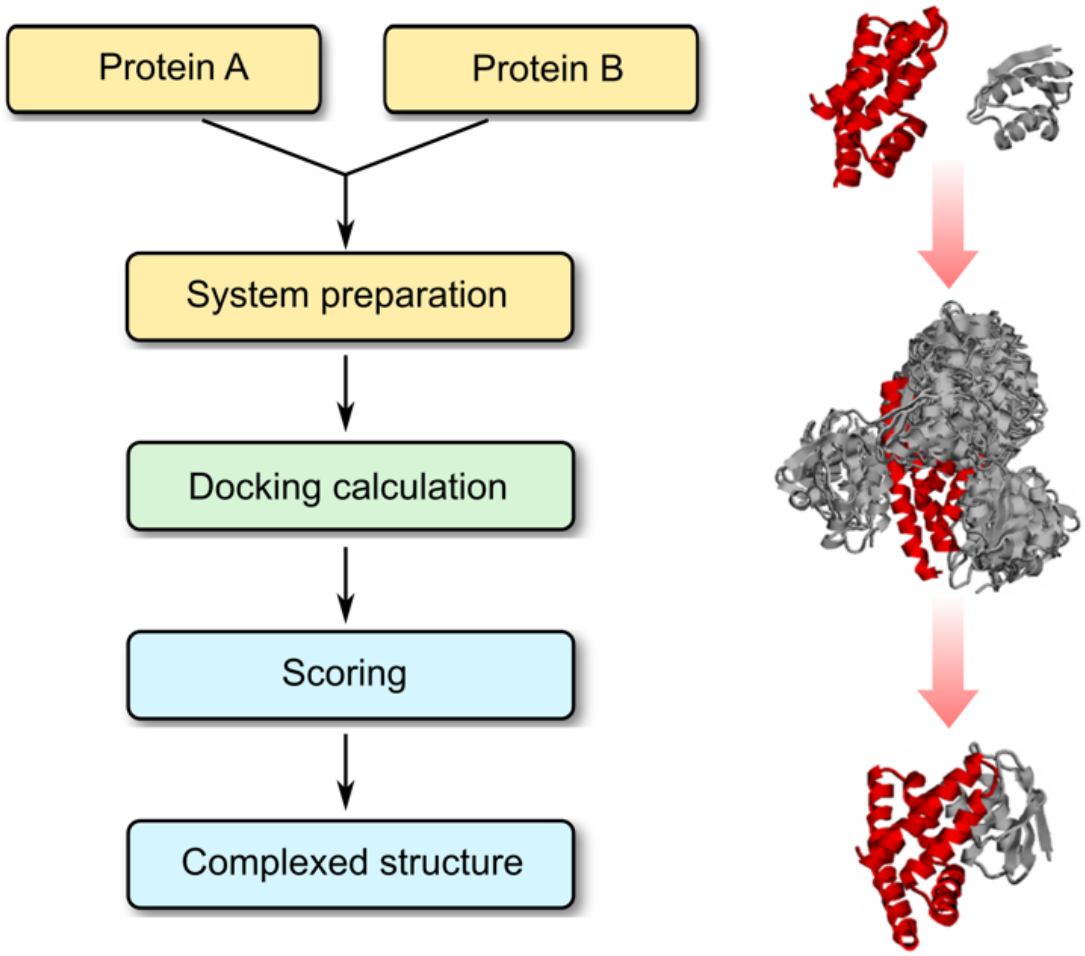


Figure 2.5: Computational Protein-Protein Docking Workflow illustrates the multi-stage process for predicting biomolecular interactions, beginning with rigid-body docking (Fast Fourier Transform correlation scoring) through induced-fit refinement to binding affinity calculation. Source: [”Protein–Protein Docking,” profacgen, `https://www.profacgen.com/protein-protein-docking.htm`]. This figure is reproduced under fair use for academic purposes.

incorporating both template-based and ab initio methods to refine regions with weak or no template alignment [108]. This multi-step approach ensures that the predicted models are not only globally accurate but also locally refined, providing a reliable representation of protein structures [109].

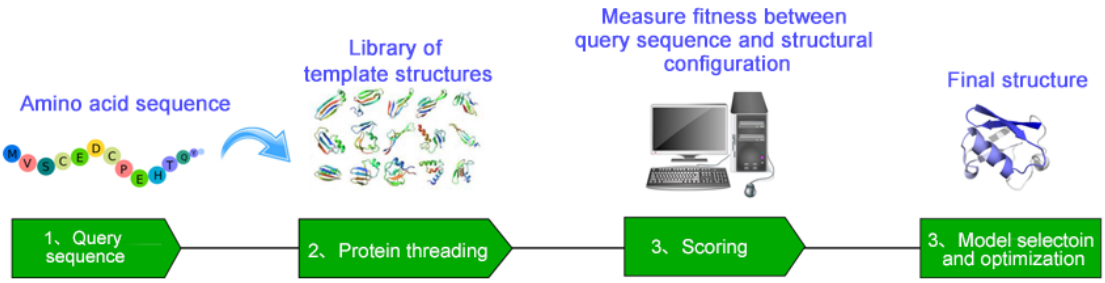


Figure 2.6: The workflow illustrates the computational pipeline for template-based protein structure prediction using threading. (1) A query amino acid sequence is aligned against a library of known structural templates (e.g., PDB entries). (2) Threading algorithms assess compatibility between the query sequence and template folds through spatial-scoring functions. (3) Candidate models are evaluated and optimized using energy-based or statistical potentials (e.g., Rosetta scoring functions). (4) The final 3D structure is selected based on conformational stability and alignment metrics. This method is particularly effective for proteins with distant homologs or low sequence identity. Source: ["Fold Recognition," profacgen, `https://www.profacgen.com/fold-recognition.htm`]. This figure is reproduced under fair use for academic purposes.

The use of replica-exchange Monte Carlo simulations for conformational sampling is another strength of I-TASSER [110]. This approach ensures that the energy landscape is thoroughly explored, leading to the identification of low-energy, near-native conformations [111]. The subsequent clustering of these conformations helps in selecting the most probable structure, which is then refined to resolve local inaccuracies such as steric clashes or unrealistic secondary structure features [112].

One of the distinguishing features of I-TASSER is its ability to predict both the 3D structure and the function of proteins [113]. The functional insights are derived by matching the predicted structures with known protein structures in functional databases, such as BioLiP [114], allowing for the annotation of ligand-binding sites, enzyme commission numbers, and gene ontology terms [115]. This dual capability makes

I-TASSER a comprehensive tool for researchers aiming to understand both the form and function of proteins.

The recent developments in I-TASSER, as detailed in the 2015 update [116], have further enhanced the accuracy and usability of the tool. Notably, the introduction of the ResQ algorithm for residue-specific quality estimation and B-factor prediction has significantly improved the local accuracy of the models. This advancement addresses one of the previous limitations of I-TASSER, where global accuracy was emphasised at the expense of local structural details. By incorporating atomic-level structure refinement tools like Fragment Guided Molecular Dynamics (FG-MD) and ModRefiner, I-TASSER now offers models with improved hydrogen-bonding networks and reduced steric clashes, making them more physically realistic and suitable for downstream applications such as drug design.

I-TASSER's performance in community-wide blind tests, such as Critical Assessment of Structure Prediction (CASP), has consistently ranked it among the top automated servers for protein structure prediction [117]. Its high accuracy, ease of use, and comprehensive output have made it a go-to tool for researchers across the globe, which spans over 50000 users from 117 countries, highlighting its global impact on the field.

The continuous improvement and expansion of I-TASSER's capabilities, including its integration with other predictive tools like QUARK for ab initio modelling, position it as a leading solution for both academic and industrial applications. Whether used for basic research, drug discovery, or enzyme engineering, I-TASSER's comprehensive approach to protein modelling and function prediction continues to push the boundaries of what is possible in computational biology.

I-TASSER represents a cornerstone in computational protein modelling, combining sophisticated algorithms with practical usability. Its continued development and integration of new features demonstrate its adaptability and relevance in the rapidly evolving field of structural biology. As I-TASSER continues to improve in accuracy and functionality, it will undoubtedly remain an essential tool for researchers seeking to elucidate the complex relationship between protein structure and function.

AlphaFold AlphaFold represents a major breakthrough in computational structural biology, developed by DeepMind and first introduced in the Critical Assessment of Structure Prediction (CASP) competition. The latest version, AlphaFold2, showcased

an unprecedented leap in prediction accuracy, achieving near-experimental levels of structural resolution for many challenging protein targets [118].

Unlike traditional protein modelling tools such as I-TASSER and Rosetta that rely on fragment assembly and energy minimisation protocols, AlphaFold leverages deep learning and attention-based neural networks to infer spatial constraints between amino acid residues directly from evolutionary information. Specifically, AlphaFold employs a novel architecture that includes an evolutionary transformer, pairwise geometric reasoning, and an end-to-end differentiable structure module, enabling it to model atomic-resolution structures based solely on protein sequences.

A core innovation of AlphaFold is its use of multiple sequence alignments (MSAs) and protein template databases to learn long-range interactions between residues. This capability allows it to predict even non-homologous proteins, outperforming all prior structure prediction methods in CASP14. According to the Jumper et al. (2021) publication in Nature [118], AlphaFold reached a median GDT (Global Distance Test) score of 92.4 across the benchmark dataset—comparable to experimental methods like X-ray crystallography.

Though AlphaFold was not designed with VR visualisation in mind, its integration into VR tools (e.g., ProMVR) enables immersive exploration of highly accurate protein models. For instance, researchers can use AlphaFold-generated PDB files to load into VR platforms and intuitively examine critical features like secondary structural elements and active sites in a spatial context. This bridges the gap between deep learning-driven prediction and interactive scientific understanding.

AlphaFold also addresses a key limitation, which either depend on user manipulation or rely on simplified models. By providing high-confidence predictions at scale, AlphaFold supports the educational and analytical layers of VR visualisation platforms. Its accessibility is further enhanced through the AlphaFold Protein Structure Database, launched in collaboration with EMBL-EBI, hosting over 200 million predicted protein structures [119].

Despite its remarkable capabilities, AlphaFold has certain limitations. It struggles with modelling protein complexes—an issue partially addressed by AlphaFold-Multimer—and lacks support for dynamic folding simulations. These limitations open up opportunities for integration with interactive VR systems such as ProMVR. In this thesis, ProMVR accesses AlphaFold predictions via HTTP API requests to Protein Frag-

ment And Structure Analysis (PROFASA), a one-stop web platform for protein structure analysis developed by Yanlin Mi [120], where AlphaFold has been deployed on the server backend. The resulting predicted structures (PDB files) serve as high-fidelity starting points for immersive, user-driven manipulation and exploration within the VR environment.

In summary, AlphaFold establishes a new standard for protein structure prediction. When paired with virtual reality platforms, its output fosters immersive, accurate, and scalable visualisation experiences that significantly enhance both research and education in molecular biology.

Conclusion All three tools—I-TASSER, AlphaFold, and UCSF Chimera—are suitable for protein structure prediction and visualisation, each offering unique advantages for different stages of this research.

I-TASSER was selected for its accessibility and consistent results in generating valid PDB files from amino acid sequences. These predictions were primarily used during the early development of Pepblock Builder VR, where pre-rendered structures were sufficient for educational and gamified visualisation.

UCSF Chimera played a crucial role as the intermediate renderer, converting PDB files into standardised 3D formats (such as OBJ and X3D), which could be seamlessly integrated into Unity-based VR environments developed by the author.

As the project evolved towards real-time and research-grade visualisation, AlphaFold was integrated via PROFASA, a web platform developed by Yanlin Mi where AlphaFold is deployed on a server backend. Through HTTP API access, the author's ProMVR system dynamically requests AlphaFold predictions, enabling the immersive, real-time loading of accurate 3D protein structures for collaborative exploration.

This combination of tools enables a flexible pipeline: from AI-based structure prediction (AlphaFold and I-TASSER), through model preparation (Chimera), to interactive, real-time visualisation (Pepblock Builder VR and ProMVR), covering both educational and scientific use cases in molecular biology.

2.3.3 Protein Folding Game and Tools

After finding tools to predict structure from amino acid sequences and successively visualise proteins, the author's research team had a game idea: Why not make a game about protein design and folding? This idea led the team to search online for existing games and inspiration. Through this research, the game Foldit and a tool named PolyFold were discovered.

Foldit Foldit is a groundbreaking initiative developed by researchers at the University of Washington that blurs the lines between scientific research, education, and citizen science by transforming protein folding into an interactive and competitive game [121]. This platform leverages the collective problem-solving abilities of players from diverse backgrounds to tackle complex problems in protein structure prediction, which has traditionally been the tasks of highly specialised researchers equipped with sophisticated computational resources [122].

Foldit operates on a simple yet profound premise: human intuition and pattern recognition, combined with basic gameplay mechanics, can effectively contribute to solving puzzles related to protein folding and design [123]. The game presents users with a three-dimensional model of a protein and challenges them to optimise its structure by minimising energy states, a task aligned with finding the most stable physical configuration of the molecule. Players manipulate protein structures using a set of tools that adjust amino acid chains and solve puzzles to achieve higher scores, which correspond to lower energy states predicted by the underlying Rosetta algorithm [124].

The game mechanics are designed to encourage experimentation and learning through trial and error, supported by tutorials and community forums. Players use a variety of tools and techniques to twist, pull, and reshape protein structures to minimise energy states, mimicking the natural processes that govern protein folding in biological systems. The competitive element of Foldit, manifested through leaderboards and team challenges, adds a layer of motivation that is uncommon in traditional scientific research.

The most notable scientific contributions of Foldit have been in areas where traditional computational approaches face limitations. Foldit players have successfully refined protein models, discovered novel protein structures, and contributed to the design of new proteins with potential therapeutic applications. Notably, Foldit participants have

resolved structures that stumped scientists for over a decade, contributed to important breakthroughs in understanding HIV/AIDS related enzymes, resolved the structure of the Mason Pfizer Monkey Virus (MPMV) retroviral protease, and participated in the design of novel proteins that could neutralise flu viruses. These achievements underscore the platform's capacity to harness human creativity and spatial reasoning in ways that complement and enhance computational algorithms.

Foldit excels not only as a research tool but also as an educational platform. It introduces players to fundamental concepts of protein chemistry and molecular biology, including the structure and function of proteins, the nature of amino acids, and the principle of molecular stability and interaction. The interactive nature of the game, coupled with real-world applications of the puzzles, provides an engaging learning experience that can enhance understanding and retention of complex scientific concepts. The recent development of Foldit Education Mode further underscores its potential as a pedagogical tool, offering structured modules that guide learners through various aspects of protein science.

Furthermore, Foldit has been integrated into educational curricula, from high school science classes to college-level biochemistry courses. The game's ability to engage learners actively helps demystify the protein folding process and stimulates interest in scientific research, potentially inspiring the next generation of scientists.

Despite its successes, Foldit faces challenges such as ensuring sustained engagement from its community, maintaining the scientific relevance of its puzzles, and continually improving the interface and user experience to accommodate a broad range of participants. Additionally, as the field of bioinformatics evolves, Foldit must adapt to incorporate new scientific knowledge and computational techniques.

Foldit represents a pioneering example of how gamification can be effectively employed to address complex scientific challenges, engage a wide audience in scientific inquiry, and educate the public about advanced topics in molecular biology. By combining the strengths of human cognitive abilities with powerful computational frameworks, Foldit demonstrates a novel approach to collaborative science. As it evolves, Foldit continues to offer valuable insights into protein structures prediction, enzyme design, and the mobilisation of citizen science for research and education, paving the way for new models of scientific collaboration and public engagement.

PolyFold PolyFold represents an innovative approach to protein structure prediction, blending the power of interactive visualisation with sophisticated computational modelling. This software platform [53] extends beyond traditional visualisation tools by enabling real-time manipulation of protein structures. It leverages the robust computational capabilities of molecular dynamics to offer an immersive, user-driven exploration of protein folding processes.

PolyFold is designed to facilitate a deeper understanding of protein dynamics through a hands-on, intuitive, and user-friendly interface, making it accessible to both experts in the field and those new to molecular biology. It incorporates advanced algorithms for real-time simulation of protein folding, allowing users to manipulate structures directly and observe the immediate effects of their interventions. This interactivity is supported by a distance-based folding approach, where users can adjust distances between amino acids and watch as protein structure dynamically adjusts to maintain these constraints.

The integration of sophisticated algorithms like gradient descent and simulated annealing within a user-centric platform differentiates PolyFold from other tools that often rely on pre-calculated models. This approach not only provides a more educational experience by illustrating the dynamic nature of protein folding but also allows for the exploration of hypothesis-driven scenarios in research settings. It allows PolyFold to efficiently navigate the complex energy landscapes typical of protein structures, offering predicitons that are both rapid and reliable.

One of PolyFold's standout features is its visualisation capabilities. The software employs a sophisticated rendering engine that not only presents proteins in a visually appealing manner but also embeds functional insights directly into the visual output. For instance, users can visualise energy minimisation in real-time, gaining immediate feedback on the structural stability of their modifications. This feature is particularly valuable for educational purposes, as it provides an intuitive understanding of the underlying biochemical principles governing protein folding.

Moreover, Polyfold's user interface is designed to be accessible to both experts in computational biology and novices interested in protein science. This accessibility is achieved through a streamlined design that hides the underlying complexity without sacrificing the depth of scientific inquiry. Users can engage with the tool at various levels of sophistication, form simple manipulation to deep dives into the molecular mechanics.

PolyFold's potential as an educational tool is significant. It offers a unique platform for students and researchers to experiment with protein folding theories and observe the immediate consequences of structural changes. This capability makes it an excellent resource for teaching complex concepts in biochemistry and structural biology, providing a tangible, interactive way to explore abstract scientific ideas.

From a research perspective, PolyFold serves as a valuable exploratory tool that can complement more traditional computational methods. By allowing researchers to intuitively explore protein configurations and test hypotheses in real-time, it enhances the creative process of scientific discovery. This could lead to novel insights and methodologies in the study of protein dynamics.

Despite its advantages, PolyFold faces challenges typical of simulation software, such as the need for continuous updates to incorporate the latest scientific findings and computational techniques. Future version of PolyFold could benefit from enhanced collaboration features, allowing multiple users to interact with simulations in a shared environment, which would be particularly useful for remote teams and educational groups.

PolyFold is a powerful and innovative tool that marks a significant advancement in the field of protein modelling. By merging interactive visualisation with robust computational techniques, it offers both scientific and educational communities a new way to explore and understand the complex world of protein structures.

2.3.4 Protein Gamification and Education

By learning about the existing game and tools for protein folding, the author was inspired and subsequently built a game integrating both ab initio protein modelling and protein folding elements called Pepblock Builder VR. It is a gamified protein visualisation and design tool aimed at bio-edutainment. It is available in desktop and VR headset versions, leveraging immersive experiences for users to understand complex 3D protein structures. The tool also integrates visualisation, interactive learning, and semi-LEGO style modular design to make protein structure building accessible and engaging for non-technical users.

An article written by the author's team [2] delves into the technical points used in developing Pepblock Builder VR, including the predictions of amino acid sequences to

protein 3D structures (ab initio) through an I-TASSER server, predicted PDB protein data converted into 3D models with UCSF Chimera and Blender, and eventually the drag&drop gameplay, folding simulation, and VR integration through Unity and Oculus Rift.

Working on the path of VR protein visualisation, a web-based VR tool called ProteinVR and an article about educational VR games drew the author's attention.

ProteinVR ProteinVR represents a significant advancement in molecular visualisation using VR. It leverages emerging 3D web graphics and VR headset technologies to display molecular components in a complete 3D environment. The application is developed using TypeScript and utilises the WebVR JavaScript API to access necessary hardware for VR rendering.

The article introducing this tool [54] discusses the support for various browsers and the anticipated broadening of support due to VR's rising popularity. It also uses 3Dmol.js to generate molecular models for VR viewing and integrates Blender to create 3D environments (see Figure 2.7). The paper also provides insights into the technical aspects of it, including its design and implementation, the use of 3D environments, and the potential for educational and research applications in molecular biology.

Regarding protein molecular biology education, an article about EVRGs [55] blew the author's mind. It steered their research direction from protein gamification towards education in a multi-user/multi-player setting (teachers, lecturers, and students).

Exploring the trends of EVRGs This systematic review examines the current trends in EVRGs, focusing on their technological, pedagogical, and content aspects. It provides a comprehensive overview of existing studies on EVRGs, highlighting the growing interest in VR for educational purposes. The review adopts a methodical approach, utilising guidelines for conducting systematic literature reviews, and emphasises the need for more comprehensive studies that cover a range of perspectives. This article offers valuable insights into the evolution of EVRGs, their applications in education, and potential future research directions.

It serves as a valuable resource for the author who is interested in the intersection of VR and education and guides them to decide to expand the research to a protein multiplayer

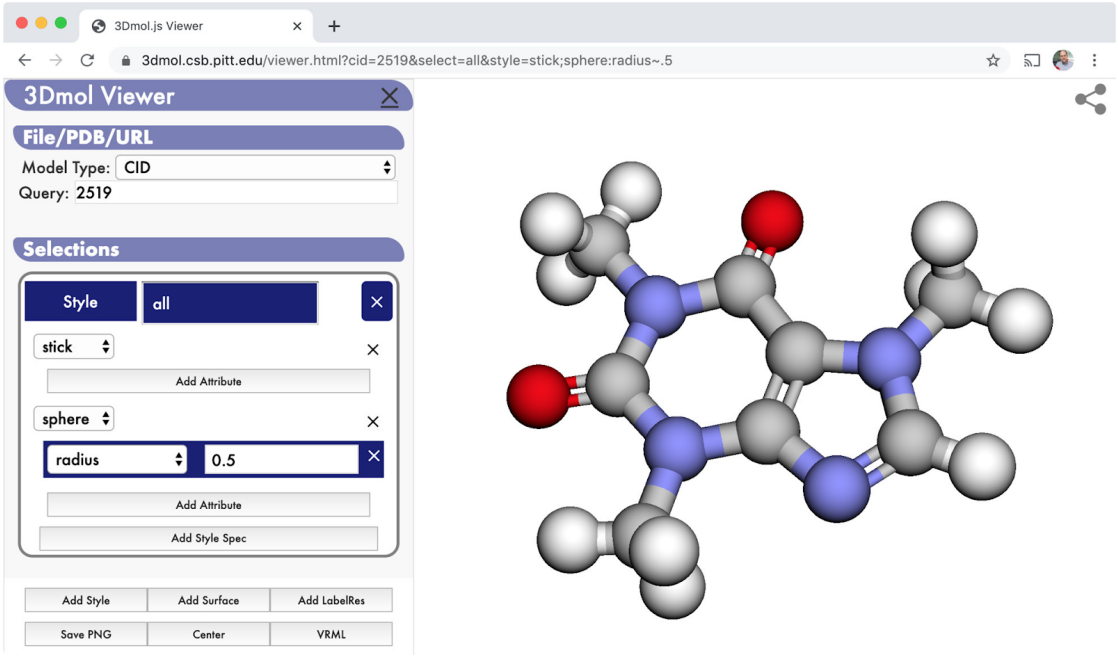


Figure 2.7: This screenshot demonstrates the real-time 3D molecular visualization capabilities of the 3DMol.js viewer: (1) WebGL-rendered protein structure with dual representation (stick and sphere models); (2) Dynamic style customization interface allowing radius adjustment and multi-representation rendering; (3) Client-side computational features including surface generation and label annotation; (4) Export functionality for research documentation. As a JavaScript library leveraging WebGL/WebVR APIs, it exemplifies lightweight yet powerful solutions for browser-based structural bioinformatics. Source: [`https://3dmol.csb.pitt.edu/viewer.html`]. This figure is reproduced under fair use for academic purposes.

VR educational tool, with protein structure sharing, loading and saving from a server, and remote controlling features as its core.

2.4 Conclusion

This research aimed to explore the transformative potential of integrating VR technologies into visualising and manipulating protein structures. By highlighting its potential to revolutionise educational frameworks and research methodologies in bioinformatics and structural biology, it inspires a sense of hope and excitement for the future of these fields.

The central questions for this research were: How can VR advance protein visualisation and design? What are the most effective VR features and methodologies that significantly benefit protein visualisation and design?

Following the PRISMA-ScR guidelines, the scoping review approach was employed to systematically identify and analyse current applications and studies focusing on VR's role in protein visualisation. The review examined recent literature and existing technologies, evaluating VR applications' educational and practical impacts on protein design. The findings demonstrated that VR's immersive, intuitive, and interactive visual representations profoundly comprehend complex biological structures, offering a hands-on experience that traditional two-dimensional approaches cannot match.

The study identified vital VR tools and applications, such as UCSF Chimera, Rosetta, I-TASSER, Foldit, and PolyFold, each contributing uniquely to the visualisation and manipulation of protein structures. These tools were evaluated based on their capabilities in rendering, predicting, and simulating protein structures and their potential for educational purposes.

Furthermore, the research highlighted the significance of gamification and EVRGs in enhancing the learning experience. Tools like Pepblock Builder VR and ProteinVR exemplify the integration of gamification elements and collaborative features, making protein design and visualisation more accessible and engaging for students and researchers alike.

Throughout this research, several key findings emerged. Firstly, VR's immersive and interactive capabilities significantly enrich the educational experience by providing a hands-on approach to studying complex protein structures. This not only aids in better

comprehension but also increases cognitive retention of intricate biological concepts. Moreover, VR applications facilitate collaborative learning environments through multiplayer functionalities, potentially transforming educational paradigms by fostering a more engaging and interactive learning atmosphere.

In conclusion, this research underscores the transformative potential of VR in molecular biology, particularly in protein visualisation and education. The findings advocate for continued exploration and development of VR applications in this field, emphasising the need for innovative solutions to overcome existing challenges. Future studies should focus on the longitudinal impacts of VR in education, its scalability, and its integration with other technological advancements to further enhance its utility and accessibility in scientific visualisation.

3 Technical Foundations: 3D Modeling, Game Engine and VR Educational Systems

3.1 Introduction

This chapter introduces Blender's capabilities as an open-source 3D modelling tool that empowers users to create and modify three-dimensional models using basic geometric shapes like cubes, spheres, and cylinders. These shapes serve as foundational meshes, which can be enhanced visually with materials to alter their appearance. Additionally, Blender's functionality extends into animation through its armature feature, which integrates skeletal structures into models for dynamic movement simulation.

This chapter also explores Unity, a leading game development platform known for its versatility across various operating systems such as Windows, macOS, Android, and Oculus. Unity simplifies the development process with tools for creating engaging game environments, intuitive user interfaces, and complex game mechanics, all supported by C# programming.

Blender and Unity facilitate a comprehensive workflow, from initial 3D modelling to final game production, ensuring a seamless transition between modelling software and game engines, making them essential for creating interactive and visually compelling educational content. Blender is used primarily for protein 3D model creation and processing along the author's PhD project, while Unity serves as the game engine to bring those models to life within an interactive environment. This chapter aims to provide a thorough understanding of the tools and methodologies utilized in the development of educational software and games for protein visualisation. These tools have been crucial in shaping the research process and have been consistently applied throughout the PhD

projects to ensure both the quality and efficacy of the educational experiences being created.

3.2 Blender

Blender is an open-source tool for 3D modelling that enables users to create or modify three-dimensional (3D) models. Fundamental geometric shapes such as cubes, spheres, and cylinders serve as the basic meshes available in Blender. Additionally, materials can be applied to these meshes to alter their visual properties (see Figure 3.1).

In Blender, the primary modes utilised for different purposes are Object and Edit. In Object mode, users can select various entities, including cameras, lamps, or 3D objects. Once selected, the entire object can be manipulated through movements, rotations, or scaling. Additionally, geometric shapes such as cubes, spheres, and cylinders created in this mode are treated as independent objects [125].

Conversely, Edit mode restricts user interaction to a single object at a time, allowing for individually modifying vertices, edges, and faces through translations, rotations, and scaling operations. This mode is particularly noted for its use of the extrusion function, a standard tool in Blender for extending faces. Unlike Object mode, Edit mode does not permit alterations to unselected objects, and any meshes created or modified here are merged into the currently selected object, maintaining a singular entity [126].

Armature is a feature frequently used in Blender to facilitate complex animations. To incorporate an armature into an object, a skeletal structure, or "bones," can be inserted into the model. The precise placement of these bones allows for automatic weights to be applied, which is essential for practical animation rigging.

When bones and weights are appropriately added to an object, manipulating or rotating a bone will influence specific parts of the object, mimicking natural movement. This capability is instrumental in creating dynamic 3D animations. For example, when a humanoid character is equipped with an armature and the weights are accurately assigned, the character's body can be animated as puppeteering through the movement of its bones (see Figure 3.2).

Python scripts integrate seamlessly with Blender, as nearly all user interface elements within Blender are associated with Python commands, which are displayed as tooltips. This integration allows many of Blender's manual operations to be converted into Python

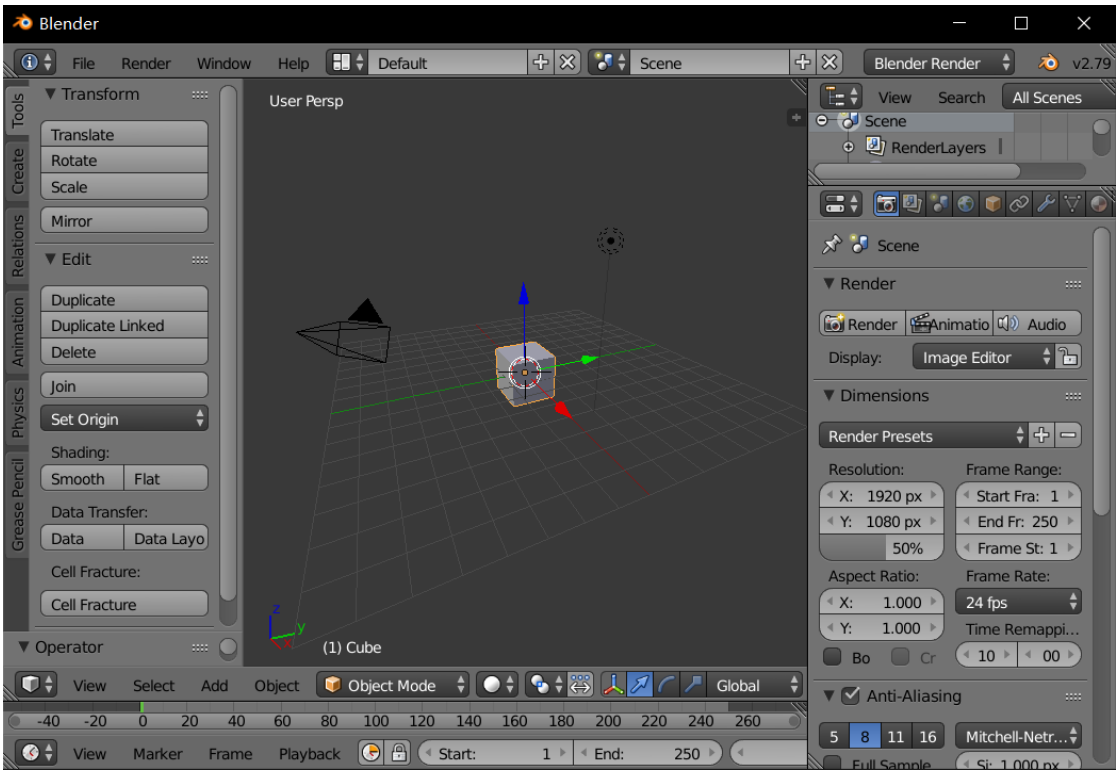


Figure 3.1: Blender Interface Overview with Key Functionality Highlighted. The screenshot demonstrates four core components of the 3D modeling environment: (1) Transform operations including geometric manipulations and object duplication, (2) Viewport perspective settings with rendering configuration, (3) View navigation and animation timeline controls, and (4) Anti-aliasing quality settings. This professional toolkit supports the 3D asset creation pipeline discussed in Section 3.2, enabling scientific visualisation tasks such as protein modeling. Interface elements are annotated to emphasize their relevance to computational geometry and real-time rendering research. This figure is reproduced under fair use for academic purposes.

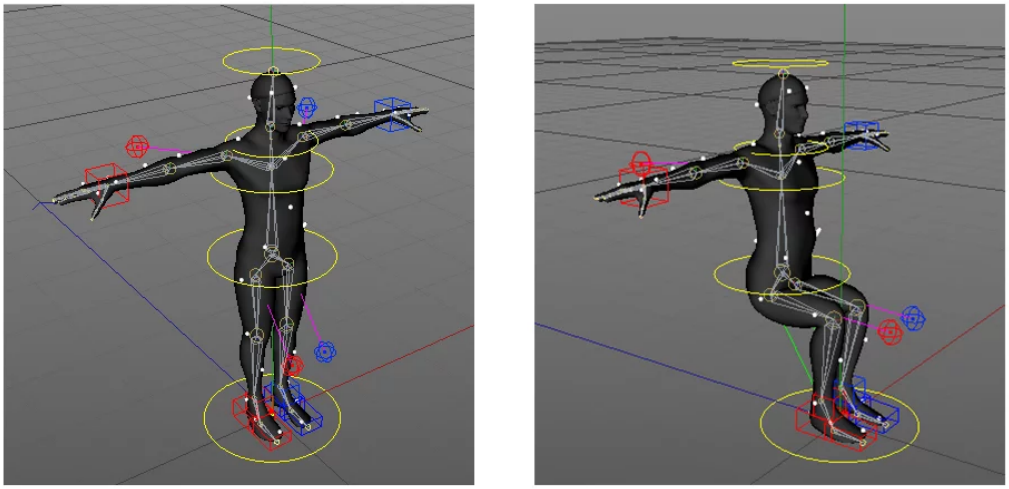


Figure 3.2: Biomechanical Rigging Pipeline for Educational VR – This figure illustrates Blender's armature system for animating protein structures, showcasing bone-weighting techniques and collision-aware skeletal hierarchies essential for real-time molecular simulations. The workflow enables seamless Unity integration through optimized quaternion rotations and inverse kinematics (IK), supporting interactive learning environments in scientific visualisation. Source: [`https://3d-ace.com/blog/3d-model-rigging/`]. This figure is reproduced under fair use for academic purposes.

code snippets. Such functionality can significantly enhance efficiency for 3D modellers, especially when creating models that are similar to some point (see Figure 3.3).

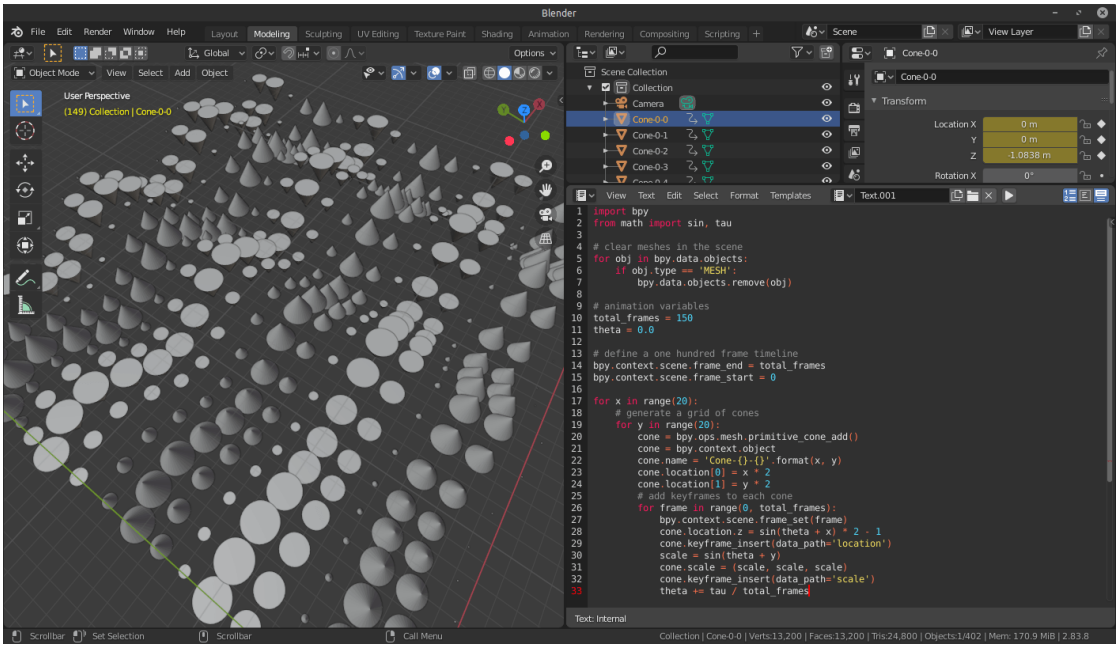


Figure 3.3: Blender-Python Scripting Interface for Procedural 3D Modeling - This figure demonstrates the integration of Python scripting within Blender's modeling environment, highlighting the code editor panel used for automating mesh generation and material assignment. The interface enables rapid prototyping of complex protein structures through algorithmic geometry manipulation, with real-time visualisation feedback critical for scientific visualisation research. Source: [`https://tabreturn.github.io/code/blender/python/2020/06/06/a_quick_intro_to_blender_creative_coding-part_1_of_3.html`]. This figure is reproduced under fair use for academic purposes.

Additionally, Blender supports exporting 3D models in various file formats, including Digital Asset Exchange (DAE), Filmbox (FBX), OBJ, and even Blender's native file format. These models can be easily integrated into Unity, a popular game engine, through a simple drag-and-drop action. This capability benefits users developing cross-platform applications, facilitating a smooth workflow between different software environments.

3.3 Unity

Unity is a potent and versatile game development engine acclaimed for supporting various platforms, including Windows, macOS, Android, and Oculus VR systems. It equips developers with sophisticated tools to craft engaging game environments and intuitive user interfaces. Additionally, Unity offers extensive functionality for adding dynamic features and robust game mechanics. Developers can leverage Unity's advanced graphical capabilities to handle 2D and 3D content effortlessly, ensuring top-tier visual quality. The engine also supports scripting in C#, enabling the seamless integration of complex game logic and interactions. These comprehensive features make Unity an indispensable tool for developers aiming to build high-quality, cross-platform gaming experiences (see Figure 3.4).

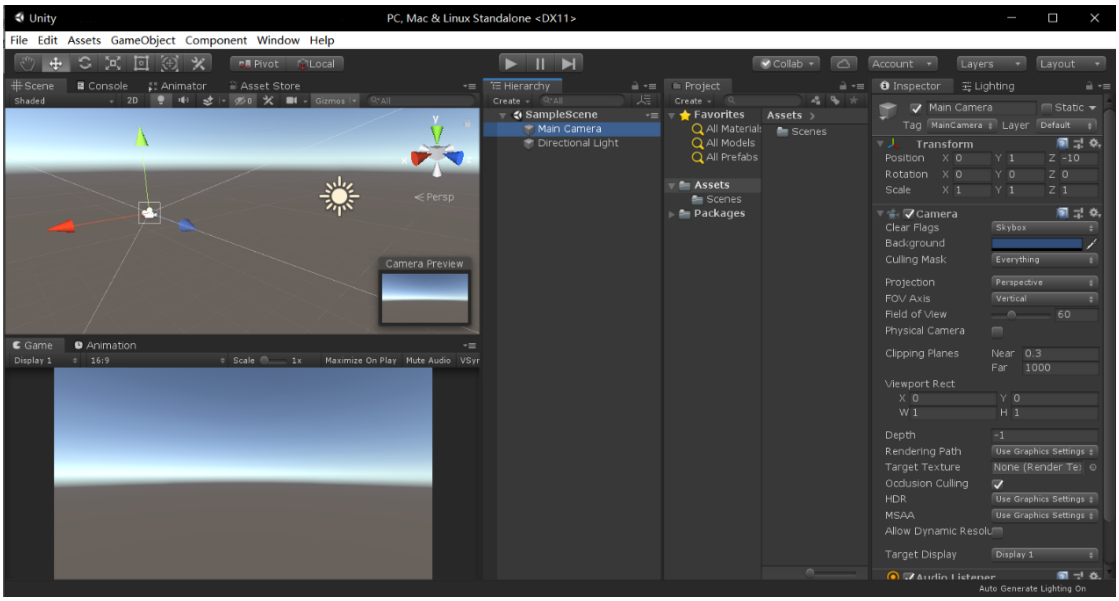


Figure 3.4: Unity Engine Workspace for VR Educational Development - This figure captures the Unity editor interface configured for developing immersive protein visualisation experiences, showcasing the Scene-Project-Hierarchy panel organization essential for managing complex 3D assets. Key features include the real-time physics preview window and C# script inspector, enabling iterative refinement of interactive molecular simulations for STEM education applications. This figure is reproduced under fair use for academic purposes.

Unity offers a wealthy Asset Store, a resource for developers seeking to enhance their projects with high-quality content. This marketplace is brimming with diverse assets, including sophisticated 3D models, detailed environments, valuable scripts, and comprehensive toolkits, all created by developers and artists. These resources are designed to accelerate development processes and enrich games' functionality and aesthetics [127].

Once acquired, assets are seamlessly integrated into the developer's project, appearing in the Unity Project window for easy access. Developers can drag these assets from the Project window into their game scenes, where they are automatically instantiated. Each asset becomes a part of the scene's architecture, visible in both the game's visual layout and the hierarchical structure of the project's elements in the Hierarchy window. This dual visibility ensures that assets are easily placed and efficiently managed, making Unity's Asset Store an indispensable tool for game developers aiming to streamline workflows and elevate the quality of their game projects (see Figure 3.5).

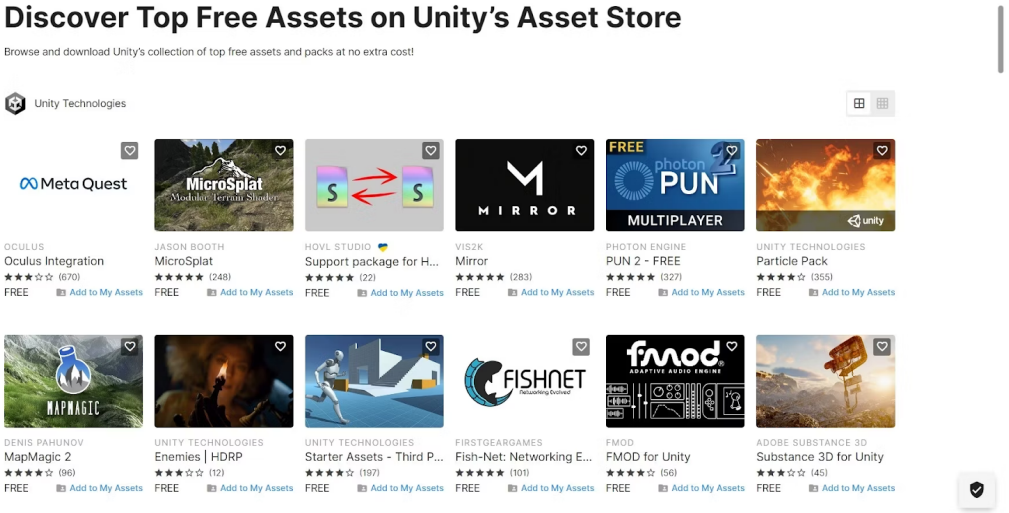


Figure 3.5: Curated Asset Pipeline for Educational VR Development - This figure showcases Unity's Asset Store ecosystem, highlighting specialized resources for scientific visualisation including molecular dynamics packages (Particle Pack), networked simulation tools (PUN 2), and optimized rendering assets (HDRP). The metadata-rich interface (ratings, compatibility tags) accelerates prototyping of protein interaction simulations while maintaining production-grade quality standards. Source: [`https://www.makeuseof.com/unity-free-assets-websites/`]. This figure is reproduced under fair use for academic purposes.

Unity's prefab system is an essential feature that significantly enhances the development workflow by allowing for the efficient reuse of game objects. Prefabs in Unity are template elements (from UI objects and 3D models to audio clips) that developers can create once and configure with desired properties and behaviours. These configured game objects can be saved by dragging them into the Project window, where their icons turn blue to indicate their status as prefabs. This visual cue confirms the transformation of standard game objects into reusable templates.

Once established as prefabs, these objects can be instantiated repeatedly across multiple scenes with consistent settings, ensuring uniformity and reducing the need for repetitive setup. This functionality is particularly advantageous for developers working on large projects or games with multiple levels, as it allows for the consistent implementation of common elements throughout various parts of the game without additional configuration. The prefab system not only streamlines the project management process by simplifying the replication of game elements but also ensures high consistency and efficiency, making it an invaluable tool for game development in Unity (see Figure 3.6).

Unity incorporates a sophisticated physics engine that enables the realistic simulation of physical interactions within the game environment. The key to this functionality is using rigid body components and appropriate colliders attached to game objects. A rigid body allows an object to act under the laws of physics, including forces and impacts, while colliders define an object's shape for physical collisions.

When a game object equipped with a Rigidbody and a collider interacts with another object with a collider, Unity can detect the collision and execute additional effects based on the interaction. For example, sound effects such as a clash can be triggered to enhance the realism of the collision event. Additionally, Unity supports raycasting, a technique where a ray is projected from a point (often the cursor) to detect objects in its path. This detection is critical for implementing interactive features such as dragging objects with the mouse or changing object colours upon mouse hover.

These integrated physics capabilities enable developers to create more dynamic and interactive gaming experiences, providing the tools to implement detailed and responsive environments that engage players through realistic behaviours and interactions (see Figure 3.7).

Unity's particle system is a robust feature that significantly enhances the visual fidelity of game environments. Developers can simulate various natural and artificial phenomena

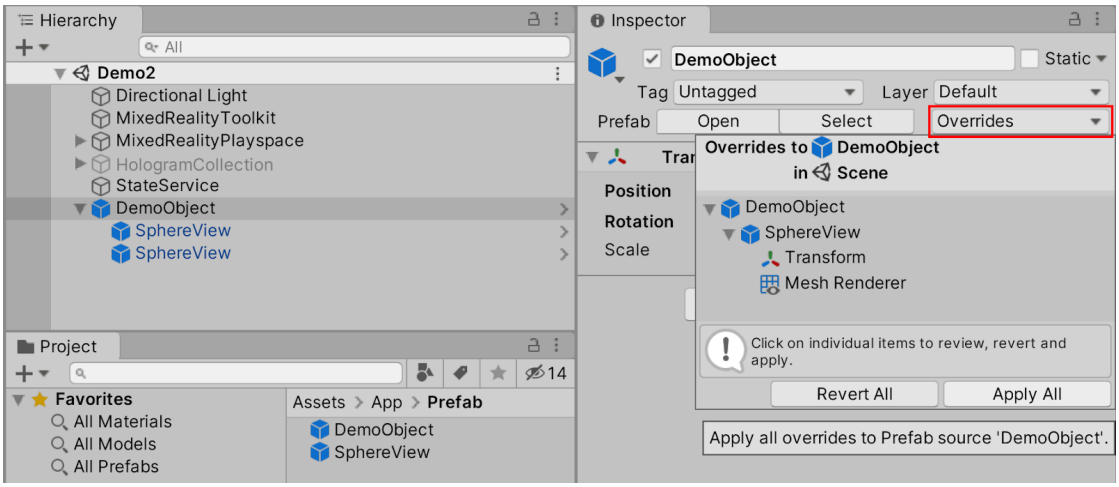


Figure 3.6: Prefab Workflow Optimization for Molecular Visualisation Assets - This figure demonstrates Unity's prefab override system, showing hierarchical management of 3D protein structures (DemoObject) with scene-specific modifications. The inspector panel reveals critical components including Mesh Renderer and Transform properties, while the project view displays the prefab inheritance structure essential for maintaining consistency across multiple VR educational scenarios. Source: [https://localjoost.github.io/Short-Unity-tip-apply-changes-made-in-the-scene-to-a-prefab/]. This figure is reproduced under fair use for academic purposes.

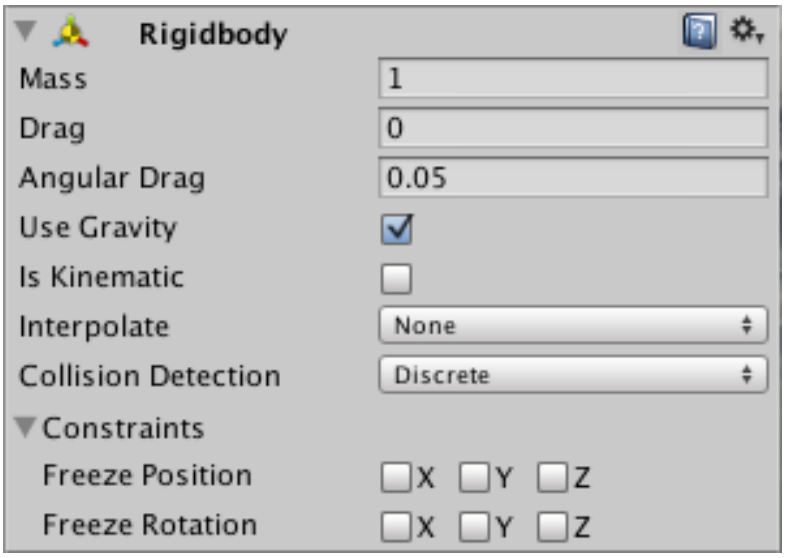


Figure 3.7: Collision Detection Optimization for Molecular Simulations - This figure illustrates Unity's Rigidbody collision detection modes (Discrete/Continuous/Speculative), demonstrating their application in simulating accurate protein-protein interactions at varying velocities. The highlighted Continuous Dynamic mode is particularly critical for preventing tunneling effects during high-speed molecular docking visualisations in educational VR environments. Source: [`https://docs.unity3d.com/560/Documentation/Manual/class-Rigidbody.html`]. This figure is reproduced under fair use for academic purposes.

by integrating a particle system into a game scene. This system allows for the emission of particles within a designated area, with extensive customisation options available through the inspector window. Developers can adjust various attributes of the particles, including their quantity, velocity, geometric shape, and colour.

This flexibility enables the creation of diverse visual effects such as fireworks, explosions, flickering flames, gently falling snow, and many more. Each effect can be finely tuned to match the game's aesthetic or thematic requirements, contributing significantly to the overall atmosphere and player immersion. Unity's particle system is designed to be both powerful and user-friendly, providing developers with the tools to bring their creative visions to life through captivating visual effects (see Figure 3.8).

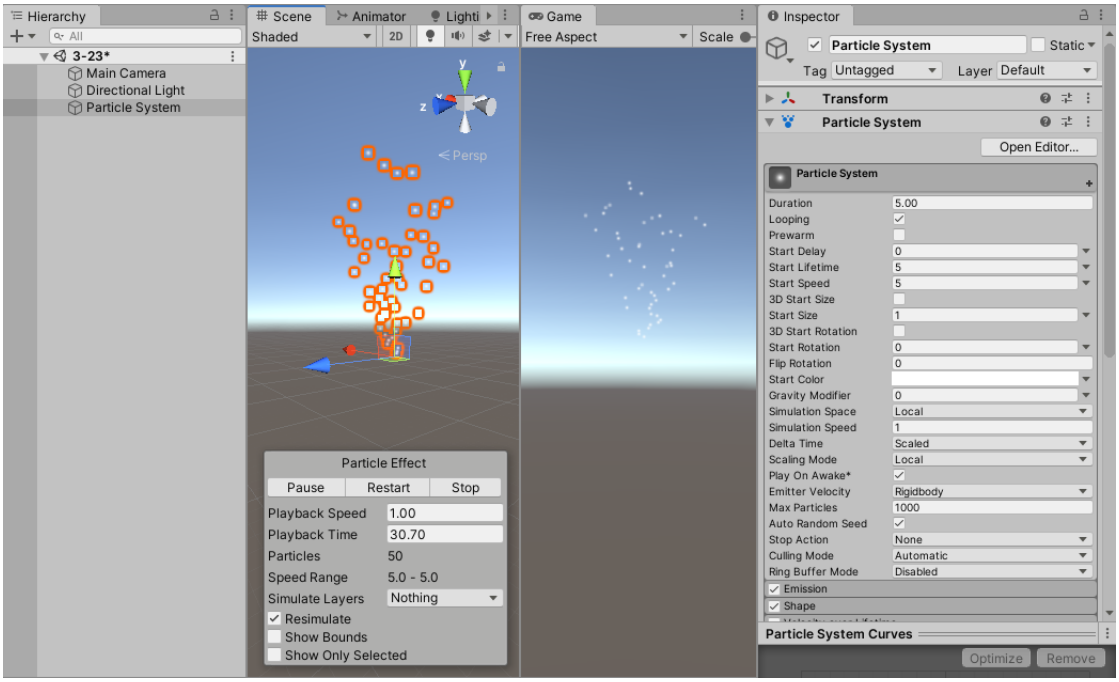


Figure 3.8: Physically-based Particle Systems for Biomolecular Visualisation - This figure demonstrates Unity's particle system capabilities configured for simulating dynamic molecular processes, including protein-ligand binding events and cellular transport mechanisms. The modular parameter controls (emission shape, velocity over lifetime, and collision thresholds) enable accurate representation of stochastic biological phenomena at scalable rendering performance for VR education platforms. Source: [`https://learn.unity.com/tutorial/introduction-to-particle-systems`]. This figure is reproduced under fair use for academic purposes.

Unity's audio system is a comprehensive framework designed to manage all auditory elements within the game. This system allows sound effects and voice clips to be attached to game objects via an Audio Source component. This component is highly configurable within the Inspector window, where settings such as volume, pitch, and looping can be adjusted to suit the specific needs of the game environment.

The Audio Mixer window manages further refinement of sound output, providing advanced control over the audio levels via faders. These faders adjust the volume of audio sources in real-time, allowing for dynamic audio mixing that can respond to game events, player actions, or environmental factors. Unity's audio system enhances the immersive experience by delivering rich, dynamic soundscapes integral to engaging gameplay (see Figure 3.9).

Unity's animation system is designed to facilitate the creation of complex animations and cinematic sequences. At the core of this system are the Animation and Animator windows, where developers can craft and control animations. In the Animation window, keyframes can be placed on a timeline to construct detailed animation clips for any game object. These clips are then managed in the Animator window, which allows for transitions between clips based on specified conditions.

When a condition is met, the animation seamlessly transitions from one clip to another, providing a fluid narrative flow. Additionally, the Timeline feature acts as a comprehensive tool, like a film director managing a series of keyframes that simultaneously affect multiple objects. This functionality enables developers to synchronise actions across various elements, enhancing the storytelling aspect of the game and ensuring a cohesive visual experience. Unity's robust animation tools empower developers to bring their creative visions to life with precision and artistic integrity (see Figure 3.10).

C# serves as the primary programming language within Unity, offering a framework to implement a wide range of functionalities essential for game development. Within Unity, C# scripts can be conveniently attached to game objects as components by utilising drag&drop functionality. Once attached, the properties of the script (public variables and methods) are prominently displayed in the inspector window, allowing for easy adjustments and interaction during the development process (see Figure 3.11).

Unity scripts typically start with two fundamental methods: Start and Update. The Start method is intended for initialisation tasks and is executed only once when the game starts, setting up initial states for variables and game objects. The Update method, in

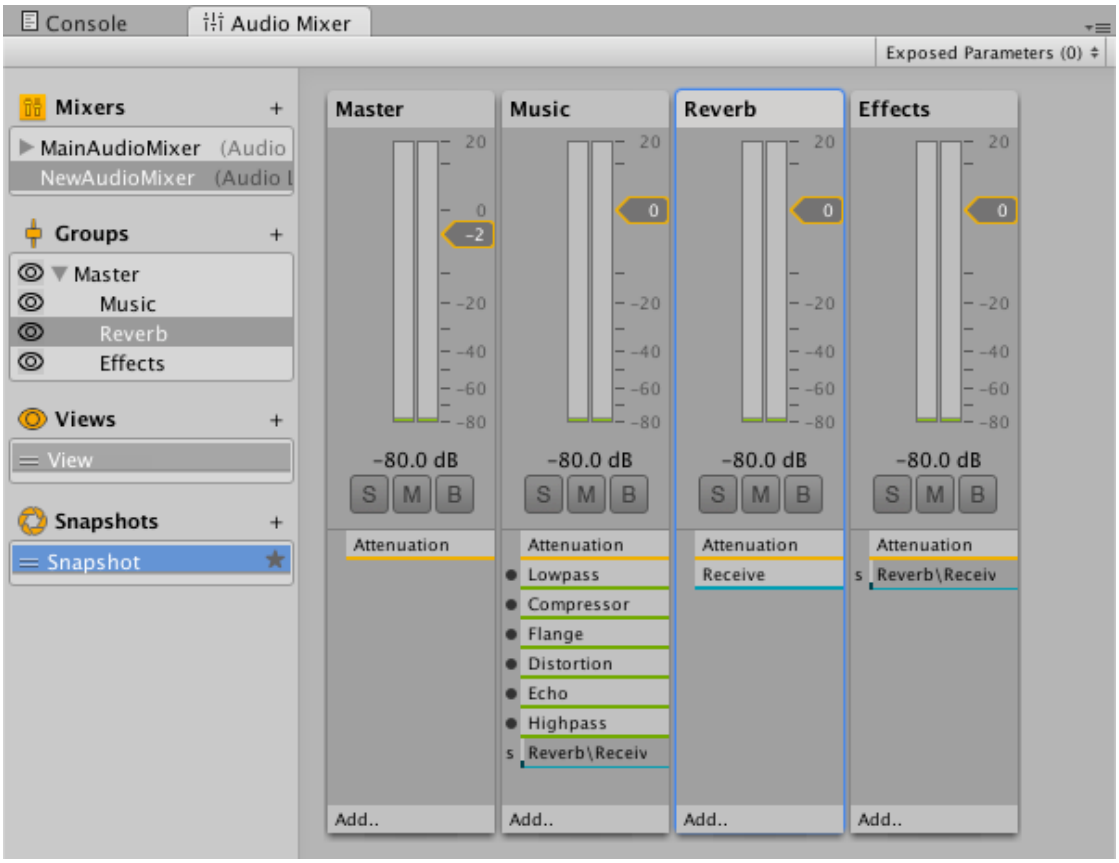


Figure 3.9: Adaptive Audio Mixing System for Immersive Simulations, this figure details Unity's audio signal processing pipeline, showcasing the hierarchical mixer architecture (Master/Music/Effects groups) and real-time DSP effects (Lowpass/Reverb/Echo) configured for biomolecular interaction sonification. The decibel-level precision (-80dB to -20dB ranges) and snapshot transition system enable dynamic audio feedback during protein docking simulations, enhancing spatial awareness in VR educational environments. Source: [`https://docs.unity3d.com/Manual/AudioMixer.html`]. This figure is reproduced under fair use for academic purposes.

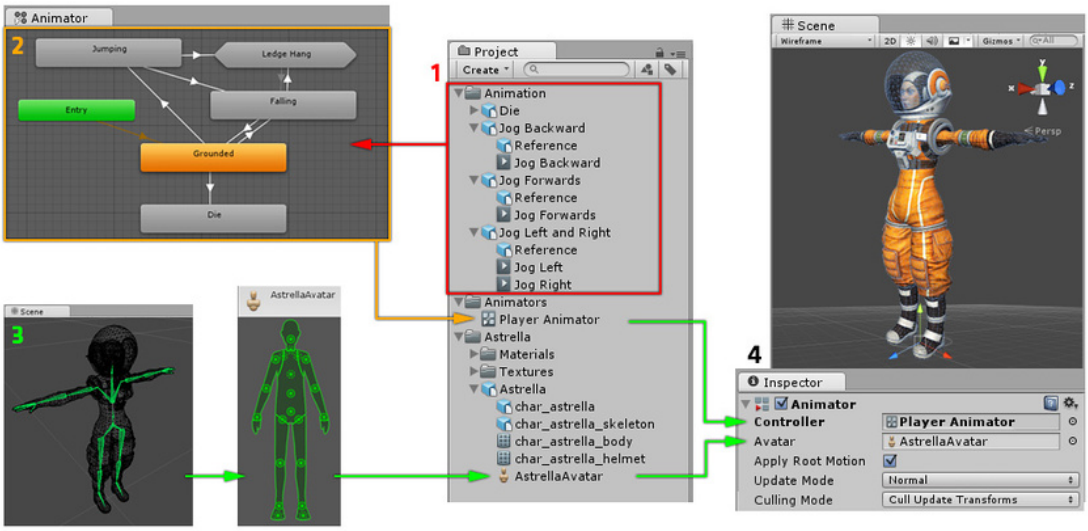


Figure 3.10: Keyframe-based Animation System, this figure showcases Unity's animation workflow featuring state machine logic and timeline-based keyframing, optimized for simulating protein conformational changes and ligand-receptor interactions. The blend tree configuration and root motion control enable smooth transitions between molecular states while maintaining physical accuracy in educational VR simulations. Source: [`https://docs.unity3d.com/560/Documentation/Manual/AnimationOverview.html`]. This figure is reproduced under fair use for academic purposes.

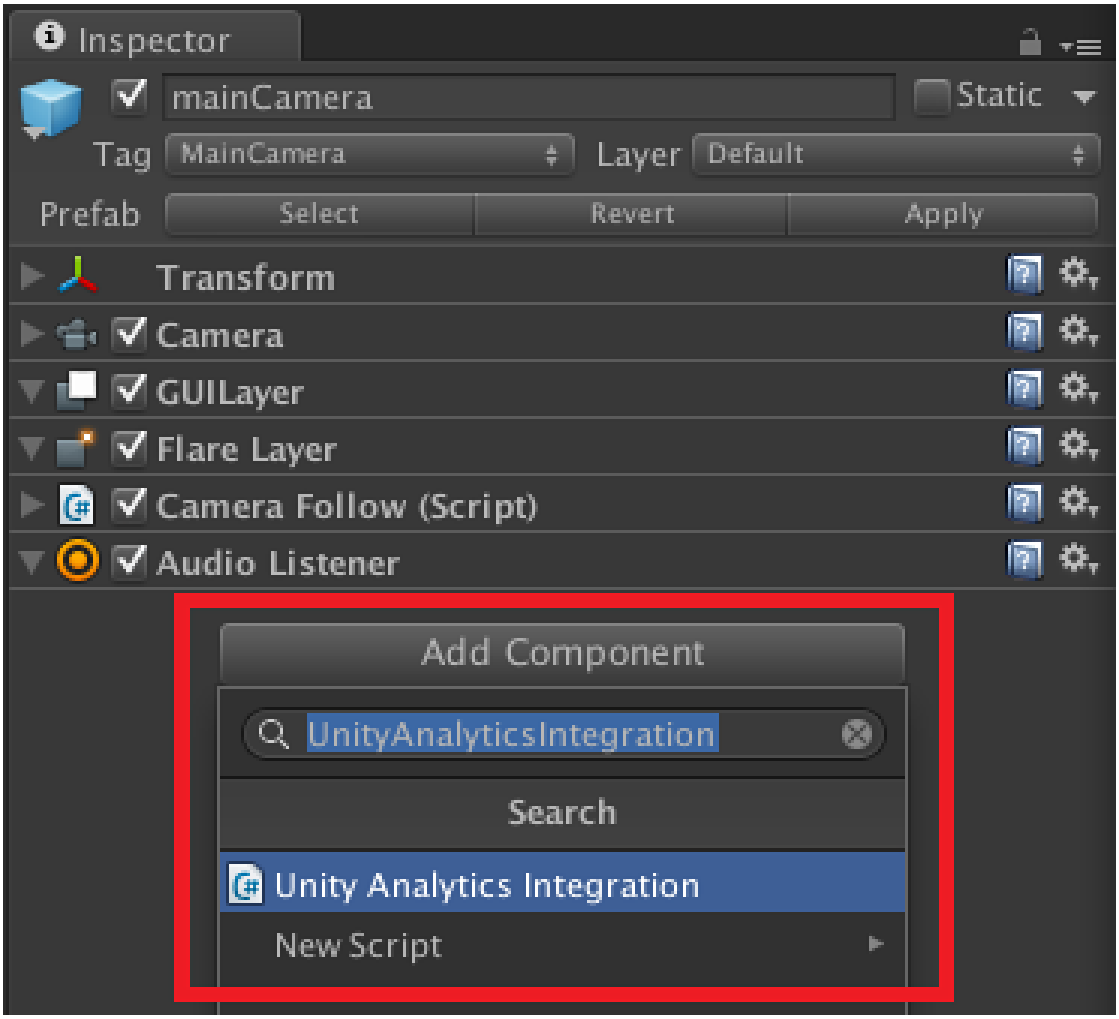


Figure 3.11: This figure demonstrates the component-based architecture of Unity through scripting, highlighting the integration of C# behavioral scripts (Camera Follow) with core rendering components (Transform/Camera/Audio Listener). The inspector interface reveals the real-time coupling between object-oriented programming logic and 3D visualisation pipelines, enabling dynamic viewpoint control for interactive protein structure analysis in VR education systems. This figure is reproduced under fair use for academic purposes.

contrast, is designed to handle tasks that need to be checked or executed continuously throughout the game. It runs once per frame, making it ideal for managing ongoing game dynamics, such as player inputs or character movements, thus maintaining smooth gameplay mechanics. This structure in C# scripting within Unity provides a powerful and flexible approach to game programming, facilitating sophisticated game behaviours and interactions (see Figure 3.12).

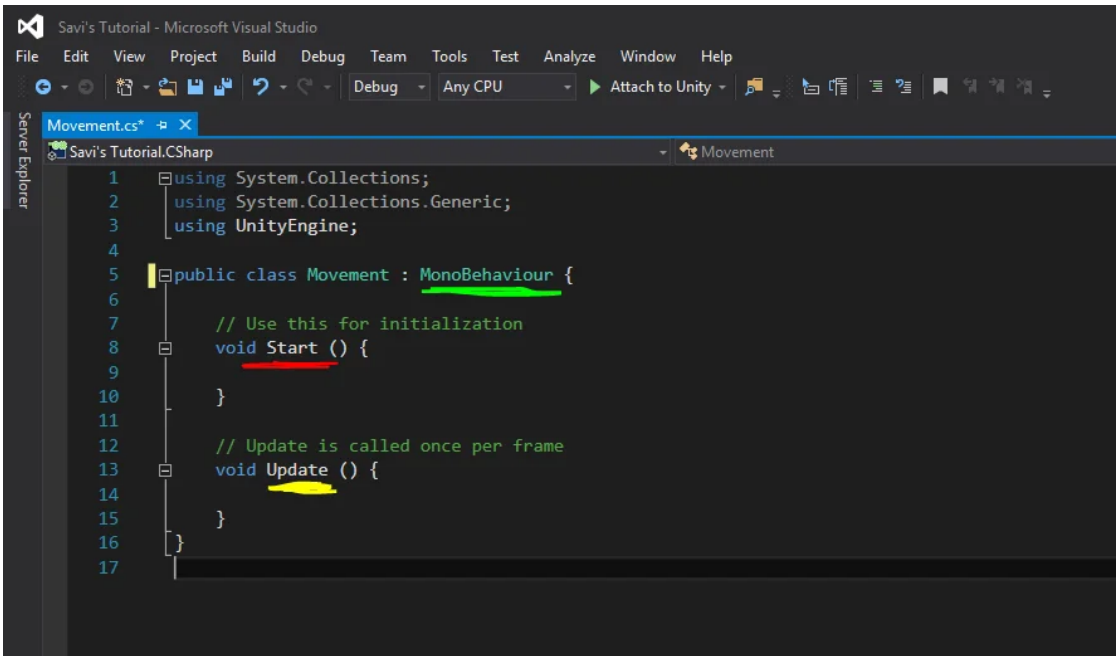


Figure 3.12: Frame-based Execution Model for Real-time Simulation - This figure exhibits Unity's core script lifecycle methods (Start/Update), demonstrating their application in synchronizing protein dynamics calculations with rendering frames. The Start method initializes molecular topology parameters while the Update loop handles continuous conformation updates at 90 fps VR refresh rates, enabling fluid interactive visualisation of biochemical processes. Source: [`https://www.studytonight.com/game-development-in-2D/basics-of-unity-script`]. This figure is reproduced under fair use for academic purposes.

3.4 Education in Virtual Reality

Upon reviewing existing literature within the domain of educational VR, the author identified significant discourse on the foundational role of learning as a lifelong human trait [128]. This discourse explores the shared nature of learning across all stages of life and emphasises how deeply learning processes are embedded in everyday activities. Technological progress, especially within digital media, has markedly altered traditional educational approaches, introducing innovative methods that enhance knowledge and skills development [129].

VR emerges as a particularly transformative technology within this context, noted for its capacity to reshape educational experiences drastically [130]. By creating profoundly immersive and highly interactive environments, VR enables learners to interact with educational content in dynamic and engaging ways that were not feasible with previous technologies [131]. For example, VR can simulate complex biological processes for medical students [132] or recreate historical events in vivid detail, allowing learners to explore and interact with past civilisations in a way that textbooks never could [133]. This ability to provide experiential learning through simulation increases engagement and enhances understanding and retention of complex subjects [134].

The discourse highlights VR not merely as an incremental advancement in digital tools but as a paradigm-shifting platform that introduces a novel spectrum of experiential learning opportunities [135]. VR facilitates a revolutionary way for users to interact with and comprehend the world around them, significantly enriching the educational landscape. The technology's prowess in accurately simulating complex, real-world environments provides learners with hands-on experience within a safe, controlled setting devoid of real-world risks [136]. This attribute is particularly beneficial in disciplines requiring substantial practical engagement, such as medical education, where students can practice surgical procedures in VR simulations that mimic actual operating rooms and patient scenarios without the ethical concerns or consequences of real-life surgery. Similarly, in aviation training, VR allows pilots to experience and manage flight operations and emergency responses in diverse weather and technical conditions, ensuring they have the necessary skills and confidence before operating real aircraft [137]. VR's immersive and interactive nature enhances learning outcomes and ensures learners are better prepared for professional challenges [138].

Additionally, VR is depicted as an increasingly utilised tool across various educational spectrums, encompassing both informal settings, where it enhances personal pursuits and hobbies, and formal educational frameworks, such as academic institutions and corporate training programs [139]. Despite its relatively limited adoption in mainstream educational systems, VR is ready for greater integration as obstacles related to cost and accessibility continue to diminish [140]. This trend is supported by the progressive decrease in the price of VR equipment and the enhancement of user-friendly VR applications, making it more feasible for schools and businesses to incorporate VR into their curricula and training programs [141]. For instance, in corporate settings, VR is leveraged for immersive job training simulations that replicate real-world tasks, allowing employees to hone their skills in a risk-free environment. In schools, VR can transport students to historical sites or galactic explorations, as mentioned above, providing a visceral depth to learning that engages and excites learners. As these technological and economic barriers are overcome, VR is anticipated to transform educational landscapes by fostering more dynamic, interactive, and compelling learning experiences.

The discussion asserts that VR represents a technological advancement and a radical shift in the pedagogical approach to education. VR transcends traditional content delivery methods by offering immersive, experiential learning that could fundamentally transform educational strategies and outcomes. This revolutionary perspective suggests that VR has the potential to profoundly enhance how educational content is interacted with and absorbed by learners. For example, in a VR-enhanced science class, students could virtually interact with molecular structures, manipulating them in real time to understand complex chemical reactions. This immersive approach can lead to a deeper understanding of subjects by allowing students to experience content in a multi-sensory environment, which aids in complex learning and retention. The discussion serves as a foundation for exploring strategic implementations of VR in education, aiming to maximise its benefits and truly innovate how educational content is understood and utilised [64].

3.5 Conclusion

This chapter explored the capabilities of Blender and Unity, two powerful tools in 3D modelling and game development, emphasising their roles in creating educational

software and games for protein visualisation. Blender, an open-source 3D modelling tool, offers robust features for creating complex models, enhancing them with materials, and animating them using armature structures. Its seamless integration with Python scripts further enhances efficiency, making it a versatile tool for developers and educators.

Unity, a leading game development platform, provides a comprehensive environment for building interactive and visually appealing games across multiple platforms. Its rich asset store, prefab system, and advanced physics and animation capabilities allow developers to create dynamic and engaging educational experiences. The scripting capabilities in C#, along with prefabs and asset integration, streamline the development process and ensure consistency across different projects.

On the other hand, Unity's versatile game development environment supports deploying these models into interactive applications. Its ability to support multiple platforms ensures that educational content is accessible across various devices, broadening the potential impact on learning.

The study also highlighted Unity's sophisticated features, such as the physics engine and particle systems, which add realism and dynamic elements to VR applications. These features are crucial for simulating biological interactions at the molecular level, making the learning experience informative and engaging.

This chapter also delved into the transformative potential of VR in education. VR's ability to create immersive and interactive learning environments is a key point of interest. It offers unprecedented opportunities for enhancing educational outcomes. By simulating real-world scenarios, VR facilitates experiential learning, allowing students to engage with content in novel and impactful ways. This is particularly beneficial in disciplines requiring hands-on experience, such as medical and aviation training, where VR can safely replicate complex situations.

Despite its limited adoption in mainstream education, VR is poised for broader integration. Its potential to revolutionise educational methodologies by offering immersive, experiential learning underscores its significance in shaping the future of education.

In conclusion, integrating Blender, Unity, and VR in educational frameworks holds significant promise for advancing scientific visualisation and enhancing the learning experience. As technological and economic barriers diminish, these tools are expected to transform educational landscapes, fostering more dynamic, interactive, and compelling learning experiences. Future research will focus on assessing these technologies' impact

on learning outcomes and student engagement in the sciences, then optimise to address existing challenges in their implementation. This could pave the way for broader adoption and optimisation of VR technologies in education, ultimately enriching learning experience and comprehension of complex scientific concepts like protein structures.

4 Computational Protein Design Through PC and VR Gamification: A Unity-Based Educational Framework

4.1 Introduction

By knowing the importance of protein modelling and 3D visualisation, the author developed an interactive protein design game called Pocket Peptides. This game is designed to enhance scientific outreach in protein design. Key features of the game include a 'drag&drop' gameplay mechanism for protein visualisation, leveraging the I-TASSER tool for generating proteins from amino acid sequences and visualising their structures from PDB files through UCSF Chimera. This game also integrates tools like Unity, C#, Blender, Python, and Illustrator for development. It aims to engage players in the scientific field of protein design by allowing them to manipulate protein structures and understand their assembly into complex forms.

The game also incorporates a 'bend&twist' feature, achieved through auto bones generation and 3D rigging in Blender, adding a dynamic and interactive element to the gameplay. Additionally, the game offers immersive VR features, enhancing the user experience and providing a more engaging way to learn about protein design. A paper by the author's team renaming this game as Pepblock Builder VR is a comprehensive exploration of the intersection of gaming, education, and molecular biology, emphasising the potential of interactive media in scientific outreach and learning. This introduction will set the stage for discussing these technical aspects in detail throughout this chapter.

Development Contribution Clarification The development of the Pepblock Builder VR and ProMVR systems represents the core practical output of the author's PhD research, building upon conceptual foundations proposed during MSc studies. Specifically, the idea of gamifying protein modelling was initially introduced by Dr. Vamsi; the full design and implementation of the system architecture and interactive functionalities were independently carried out by the author.

The Pepblock Builder VR system was initially developed during the candidate's MSc Interactive Media studies. All single-player PC-based gameplay components—including user interface design, opening cutscene, game tutorials, drag&drop interactivity, and core gameplay mechanics (assembling peptide components into proteins)—were independently designed and implemented during that phase.

All subsequent features—such as the 'bend&twist' system, automated bone rigging for protein structures via Blender Python scripting, and VR-specific functionalities like protein spawning, visualization, manipulation 'grab&throw', and interactivity—were designed and implemented solely by the candidate during the PhD phase.

The ProMVR platform, a multiplayer VR extension to support real-time collaborative protein modelling, was entirely conceived, designed, and implemented by the candidate during the PhD. This includes full integration of:

- Mirror (for real-time network synchronization of protein model states);
- Vivox (for multiplayer voice and text communication);
- PS-GO and PROFASA (web-based platforms developed by fellow PhD researcher Dr. Yanlin).

For PROFASA and PS-GO, the author was solely responsible for the integration of API endpoints into ProMVR, enabling real-time protein structure search, retrieval, and visualization. These platforms themselves were wholly developed by Dr. Yanlin and are used as backend services.

Furthermore, all multiplayer functionality—such as synchronised state sharing, PDB file upload/download, and remote protein model manipulation and communication—was developed by the author. All code, architectural decisions, interface design, server-client logic, and performance testing setups in the ProMVR system were solely undertaken

during the doctoral research period and exclusively authored by the candidate (see Table 4.1).

Table 4.1: Development Timeline and Contribution Summary

Component / Feature	**Phase**	**Developer**
UI Design, Drag&Drop, PC Gameplay Logic	MSc (Interactive Media)	Author
Rebranding as Pepblock Builder VR	MSc (Publication)	Author&Dr Vamsi
VR Interaction, Grab&Throw Feature	PhD	Author
Blender Rigging, Bend&Twist Feature	PhD	Author
Multiplayer Networking (Mirror)	PhD	Author
Voice/Text Chat (Vivox)	PhD	Author
PS-GO Platform	PhD (Team)	Dr Yanlin
PROFASA Platform	PhD (Team)	Dr Yanlin
PDB-OBJ API Integration (UCSF Chimera)	PhD	Author
Full Integration of APIs into ProMVR	PhD	Author

Trade-off Between Scientific Accuracy and Game Adaptation To transform scientific protein models into a format suitable for interactive game environments such as Unity, significant adaptations are required. These transformations inevitably introduce compromises in terms of scientific fidelity. Protein models obtained from tools like I-TASSER and UCSF Chimera are originally intended for accurate visualisation and simulation in research-grade environments. However, these PDB or X3D files cannot be directly used in Unity without optimisation, necessitating conversion to formats like DAE or OBJ, followed by rigging and mesh simplification in Blender.

One major simplification is the removal of atomic-level data (e.g., side chains, atom types, bond lengths, the atomic coordinates) and metadata (such as hydrogen bonds, torsional angles, and residue-level annotations), which is essential in structural biology but irrelevant—and performance-inhibiting—for gameplay. The resulting models retain the overall topology of the protein (e.g., helix, sheet) but sacrifice atomistic resolution.

Interactive manipulation such as bending or twisting is implemented via bone rigging rather than physically accurate simulations of molecular forces. Bones influence mesh segments based on weight painting, not molecular bonds, which allows real-time inter-

activity but does not reflect actual protein dynamics governed by physical chemistry. For example, continuous backbone bending across residues may violate real torsional angle constraints.

In Unity, the addition of colliders and rigid bodies to enable in-game physics interactions substitutes precise molecular dynamics with approximate mechanical systems. These alterations are necessary for player feedback and usability but may distort the underlying biophysical realism. While efficient, these colliders are geometric approximations of molecular shapes and do not account for molecular surfaces or steric hindrance.

These adaptations are necessary to achieve a responsive and accessible user experience, particularly on lower-spec educational hardware. While they compromise scientific accuracy, they enable learners to intuitively grasp protein structure manipulation. The educational value derived from real-time interactivity, exploratory learning, and gamified objectives outweighs the loss in fidelity for the intended audience and pedagogical goals.

Therefore, these trade-offs were deliberately made to favour interactivity, visual clarity, and player engagement, particularly for educational audiences without prior exposure to protein science. While scientific accuracy is partially compromised, the key topological features—such as secondary structure types and domain connectivity—are preserved sufficiently to facilitate learning.

4.2 Novel Feature Contribution Apart from Foldit

As highlighted in the review segment, Foldit is a pioneering online puzzle game centred around protein folding. The game's distinctive attributes and mechanics involve interactive puzzle-solving. Players strategically manipulate amino acid chains (the building blocks of proteins) to attain the minimal energy configuration for a selected protein at each level. This gameplay effectively emulates the protein folding process observed in nature.

Foldit provides its users with a range of tools and moves designed to aid in the manipulation of protein configurations. These tools, such as the ability to pull and twist amino acid chains, freeze specific segments of the protein, and induce mutations, all serve a common goal: to uncover the most energetically favorable structure of the protein.

Nevertheless, Foldit is fundamentally conceptualised as a puzzle game centred on protein folding, where the puzzles are pre-established by the development team, drawing upon actual scientific data and challenges. The core goals are focused on the folding of proteins to attain their ideal configurations or address particular structural problems, as opposed to the manipulation or detailed analysis of existing proteins through customisable parameters. Due to this reason, certain functionalities are overlooked by Foldit yet identified by the author's team and integrated into Pocket Peptides as novel feature contributions:

Protein Model On-demand Unlike specialised tools dedicated to molecular visualisation, Foldit's fundamental gameplay does not customarily permit players to import and showcase specific protein models instantaneously. Instead, participants engage with puzzles representing protein models the Foldit team selected, aligned with genuine scientific inquiries or educational examples. The selection process of these protein models is strategically aimed at facilitating scientific research or elucidating key concepts of protein folding.

In contrast, Pocket Peptides require the ability to display protein models on demand. One method to create these models involves using I-TASSER. When a specific protein sequence (e.g., ATYFTYYSA, representative of the Beta Sheet's sequence) is submitted to the I-TASSER server, five PDB (Protein Data Bank format) files are generated and delivered via email. Each file is accompanied by a C-Score, which indicates the model's predicted structure's confidence, quantitatively expressing the reliability of each prediction.

After the PDB files are created, UCSF Chimera is crucial in converting them into X3D files. These files can then be further processed into three-dimensional (3D) formats such as DAE, FBX, or OBJ through modifications using Blender, a tool for 3D graphics. Once the protein models are prepared, it becomes essential to implement a model management system to introduce an innovative functionality: combining protein models by concatenating amino acid sequences.

Protein Parameter Display The user interface of Foldit is designed to facilitate intuitive engagement with protein structures, providing a suite of tools and moves that allow players to fold proteins efficiently. Unlike a research-focused bioinformatics applica-

tion, Foldit generally does not present detailed protein parameters, such as isoelectric point (pI) or molecular weight (MW). The game's approach is to distil complex biochemistry into a more accessible format, emphasising the process of protein folding rather than the detailed presentation of biochemical or biophysical properties of proteins.

A novel gameplay element has been introduced in Pocket Peptides, leveraging protein parameters to create game challenges. This feature operates on the premise that, upon advancing to higher levels, players are tasked with aligning the parameters of the current protein model to those of a target goal parameter through strategic manipulation of the protein. The display of these parameters is crucial and is facilitated through a dedicated parameter panel. This panel, an essential game feature, provides players with a clear and accessible view of the protein's parameters, enhancing their ability to strategise and meet gameplay challenges.

The parameters displayed for the protein are designed to update with each new combination created, necessitating the replacement of the currently displayed protein's parameters with those of the preceding one. Such an update can be efficiently facilitated through a C# script, which primarily involves modifying the text content displayed on the parameter panel. However, when a novel protein combination is formed (not amongst the predefined set), direct access to its parameters is unavailable due to their non-existence. Under these circumstances, employing a C# script incorporating the UnityWebRequest Application Programming Interface (API) becomes essential. This API's Post and Get functionalities enable the submission of the protein sequence and the retrieval of corresponding parameters, which are returned data and then parsed to extract the relevant parameters.

4.3 Gameplay and Design

This section commences with an exposition of the fundamental gameplay mechanics, followed by a detailed presentation of the functional requirements and their respective design considerations. The section's organisation adheres to the sequence of the implementation workflow, charting a course from the UI design to the generation of the executable file, with each phase delineated step-by-step. Additionally, this chapter includes an illustration of the database design, offering insights into its structural composition and operational logic.

4.3.1 An Opening Cutscene

Contemporary video games often enhance player engagement through compelling storylines that set the scene for the game's world and introduce its rules. In Pocket Peptides, this narrative framework features an Earth ravaged by human impact, with its once-thriving ecosystems now lost. Players are assisted by an AI robot that seeks help constructing new protein structures using three key secondary shapes: Sheet, Helix, and Coil. The game begins with a 60-second opening cutscene that depicts this post-apocalyptic scenario, guiding players to rebuild Earth's lifeforms using protein structures. To enrich the player's knowledge and understanding of proteins, the game intersperses educational content between tasks and levels, showcasing informative facts about proteins (see Figure 4.1).

4.3.2 The User Interface

Within this gaming environment, the UI components are represented by icons denoting the protein components (namely, the secondary structures), labels that convey names and numerical values (which may be dynamic), and buttons designed to facilitate various functions (such as initiating gameplay), resetting, accessing options, and exiting the game). To illustrate the arrangement of these UI elements, a diagrammatic sketch is provided, offering a visual depiction of the planned UI layout (see Figure 4.2).

A column of protein component icons is designed to be placed on the left-hand side of this screen. The arrangement of these icons should be neat. Otherwise, the players would notice a sense of imbalance. Text labels could be added to each icon to display the name of each protein component (for benefiting scientific outreach). The number labels can also show how many protein components are left in this game (once the number turns to 0, no component is left, and the icon will be turned off). Moreover, instead of designing them as pure static image icons, making them buttons would benefit more in this game. Its advantages will be elaborated on in the implementation section.

The sequence of icons representing protein components is envisioned to be positioned along the left margin of the screen. These icons must be organised in an orderly and aesthetically pleasing manner. To enhance educational outreach, each icon may be accompanied by text labels indicating the name of the corresponding protein component. Additionally, numerical labels could indicate the remaining quantity of each protein

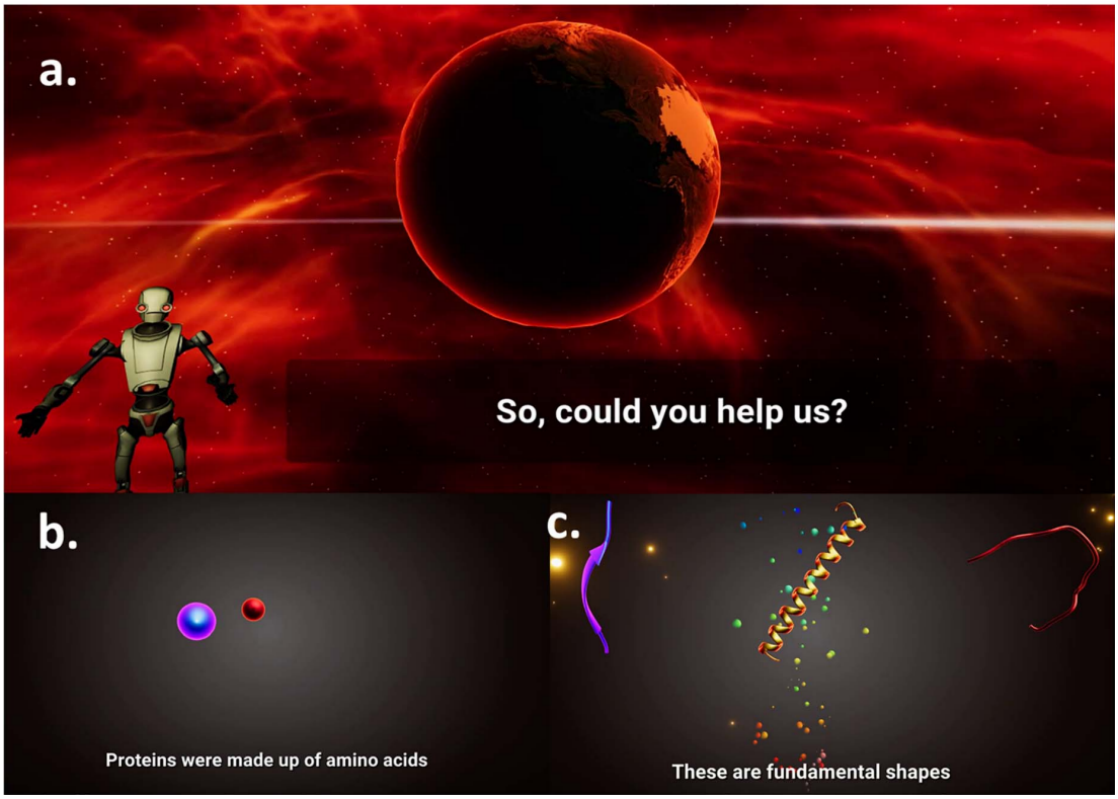


Figure 4.1: Narrative-Driven Introduction to Protein Design Pedagogy – This figure captures the opening cutscene of Pocket Peptides, where an AI guide introduces players to amino acid fundamentals and secondary structure building blocks (Sheet/Helix/Coil). The dystopian narrative frame (”Could you help us?”) contextualizes gameplay within a scientific mission, while the didactic text overlay bridges entertainment with STEM education objectives. This figure is reproduced under fair use for academic purposes.

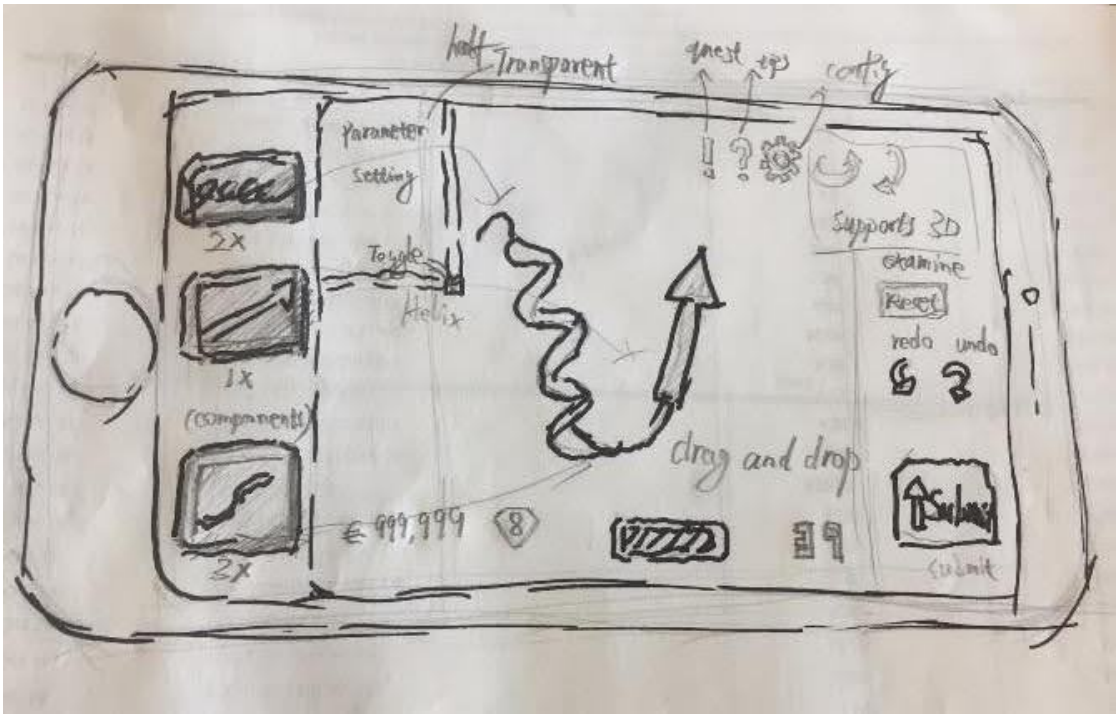


Figure 4.2: Iterative UI Design Process for Protein Assembly Game - This figure documents the preliminary sketch phase of Pocket Peptides' interface, showcasing the left-aligned secondary structure palette (Sheet/Helix/Coil), central 3D workspace, and right-sided parameter panel with real-time bioinformatics data display. The hexagonal button cluster demonstrates ergonomic considerations for scientific workflows, while hand-drawn annotations reveal usability testing insights that informed the final implementation (Section 4.3). This figure is reproduced under fair use for academic purposes.

component within the game, with the count reaching zero, signifying the depletion of that component and resulting in the icon becoming inactive. The design concept evolves beyond static imagery for these icons, envisaging them as interactive buttons that introduce a layer of functionality that promises to enrich the gaming experience. The benefits of this design choice will be further detailed within the implementation discourse.

A suite of buttons is proposed to be strategically placed on the right side of the screen. This configuration is intended to complement the icon placement on the left, achieving a balanced layout and adhering to the principles of symmetrical design. The proposed buttons include a quest button (to outline the objective of the current level), a hint button (to activate the tutorial panel), a reset button (to restart game progress), a submit button (to submit the completed task and secure a level completion), a level up button (facilitating progression to the subsequent significant level, primarily serving to expedite level transition and streamline game demonstrations), an options button (to access game settings), and a quit button (to exit the game). Given the total count of seven buttons, adopting a hexagonal background for the buttons is considered, with the submit button centrally positioned and encircled by the six other buttons. This hexagonal arrangement will contribute to a clean and orderly interface design. The specific functionalities associated with these buttons will be delineated in the section discussing implementation (see Figure 4.3).

The inclusion of a parameter panel is essential for displaying protein parameters. Positioning this panel on the right side of the screen, above the button panel, contributes to achieving a harmonious and symmetric layout. Within the parameter panel, an array of labels is envisioned for use. This arrangement entails situating the parameters' names on the left side, with their corresponding values aligned on the right, ensuring an organised and intuitive presentation of information.

The integration of tutorial panels is also contemplated within the game's design. These panels can be located near the relevant buttons or other user interface elements, particularly when the tutorial prompts the player to interact with these components. Such a design choice facilitates a more direct and user-friendly tutorial experience, enhancing instructional clarity and aiding players in navigating the game's features more effectively.

Figure 4.3: Final Implemented Interface for Protein Design Gameplay - This figure showcases the production UI of Pocket Peptides, featuring: (1) Dynamic parameter panel displaying real-time isoelectric point (pI) and molecular weight (MW) metrics, (2) Central 360° manipulable protein visualization workspace with collision-aware rendering, and (3) Context-sensitive tutorial prompts (bottom-right) validated through 89-user testing (Section 4.5). The color-coded secondary structure palette (left) implements the drag-and-drop mechanics detailed in Algorithm 4.1. This figure is reproduced under fair use for academic purposes.

The protein model is central to the gaming experience, and it occupies a prominent position on the screen. As the gameplay is structured around this model, its presentation and role will be detailed in the dedicated section on protein model visualisation.

Enhancing the main scene with a user interface offers an opportunity to engage users. A series of buttons, including a play button (which directs users to the game scene), an options button, and a quit button, could be strategically positioned along the left side of the screen. An assortment of protein models could be showcased sequentially to effectively utilise the space on the right side, adding an entertaining element to the user experience.

4.3.3 Gameplay Mechanics

The game employs a LEGO-like modular approach to protein building, where players are tasked with constructing innovative protein structures. Each level presents unique challenges, engaging users in the creative process. The game is strategically structured, advancing progressively in complexity. This approach educates and guides players, incrementally introducing them to more intricate protein design challenges.

After the start (see Figure 4.4), users are presented with an empty central workspace, allowing for the integration of various structures from a selection panel on the left. This panel contains four fundamental shapes (Sheet, Helix, rigid Coil, and flexible Coil) essential for building. On the right, a panel displays in silico parameters pertinent to the current protein structure being worked on. The interface includes a control panel at the bottom right, featuring a help button that offers a guided tour of the interface and a challenge button that outlines the specific task for the current level. The interface includes undo and redo functions, accessible via left and right arrows, and a central up button to submit the user's completed response.

For the game's PC version, the interface includes clickable buttons that facilitate navigation through different scenes, menus, and levels and options to toggle protein components (secondary structures) and access helpful tips. The protein components, represented as icons on the left panel in the gaming scenes, are designed to be dragged and dropped onto a central, glowing area on the screen (this drag&drop functionality is a key feature and will be discussed in more detail later). Additionally, players can rotate and zoom into the entire scene using the middle mouse button. In the more advanced

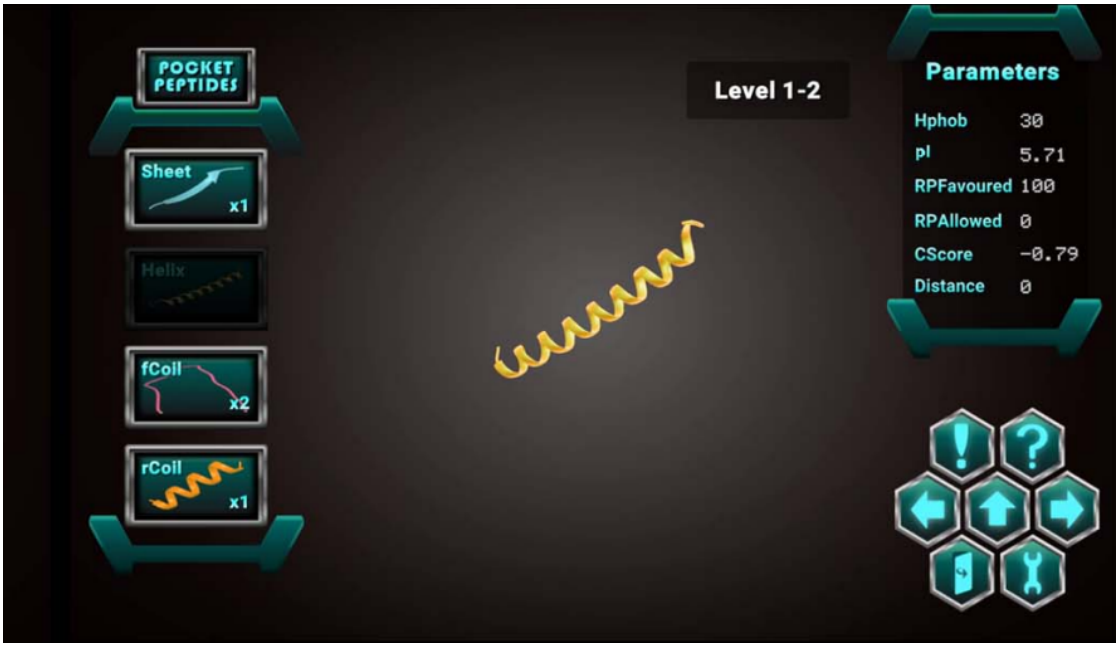


Figure 4.4: Gamified Protein Design Challenge Interface - This figure demonstrates Level 1-2's task-driven environment featuring: (1) Multi-parameter bioinformatics dashboard (Hydrophobicity=30, pI=5.71, C-Score=-0.79), (2) Real-time structural validation metrics (RPPavoured/RPAllowed ratio), and (3) Spatial constraint visualization (Distance=8Å). The interface operationalizes computational biology concepts through game mechanics, where parameter optimization correlates with progression - as validated in Section 4.5's user studies. This figure is reproduced under fair use for academic purposes.

levels of the game, players gain access to enhanced features such as bending, twisting, and turning, which can be activated by clicking and dragging on any point of the 3D structure (this bend&twist feature will also be discussed later).

The game also employs intuitive mechanics like drag&drop, bend&twist, and grab&throw to create combinations of protein molecules and fold them. It uses protein secondary structures (Sheet, Helix, and Coil) as building blocks (see Figure 4.5). The goal is to create tertiary protein structures by combining different secondary structures and folding them into predicted shapes. The gameplay mechanics are listed below.

Drag&Drop The drag&drop functionality is a crucial interactive element in this educational game, providing an intuitive mechanism for demonstrating the assembly of protein structures. Unlike traditional methods of observing proteins through laboratory equipment, this feature enables users to visualise and understand the interactions and combinations of proteins in a more accessible manner. It simplifies the workflow, effectively reducing the cognitive load for players and catering to their learning styles.

The drag&drop feature is a central interactive component in this game. Players can select protein icons on the screen's left side and move them to the central area (the drop zone) (see Figure 4.6).

This functionality encompasses several distinct features. Firstly, when a player begins to drag a protein icon, a semi-transparent version follows the mouse pointer, aiding in visual tracking. Secondly, moving icons up or down along the left panel allows for easy rearrangement of the secondary structures' order. Thirdly, successfully placing an icon into the drop zone results in a decrement of that specific protein's count, indicating it is attached to the structure on display. If the placement is not successful, the count remains unchanged. Lastly, to enhance user experience and focus, the drop zone becomes visually prominent when a protein icon is moved, ensuring players can quickly identify where to place their selection.

Bend&Twist In the game, players are tasked with replicating a specific shape (displayed in translucent neon blue) by utilising basic secondary structures in the left panel. As the game progresses, it introduces advanced functionalities allowing players to alter the protein structure by rotating, twisting, and bending its backbone. A unique haptic snaplock feature is integrated, which locks the protein structure into its intended form

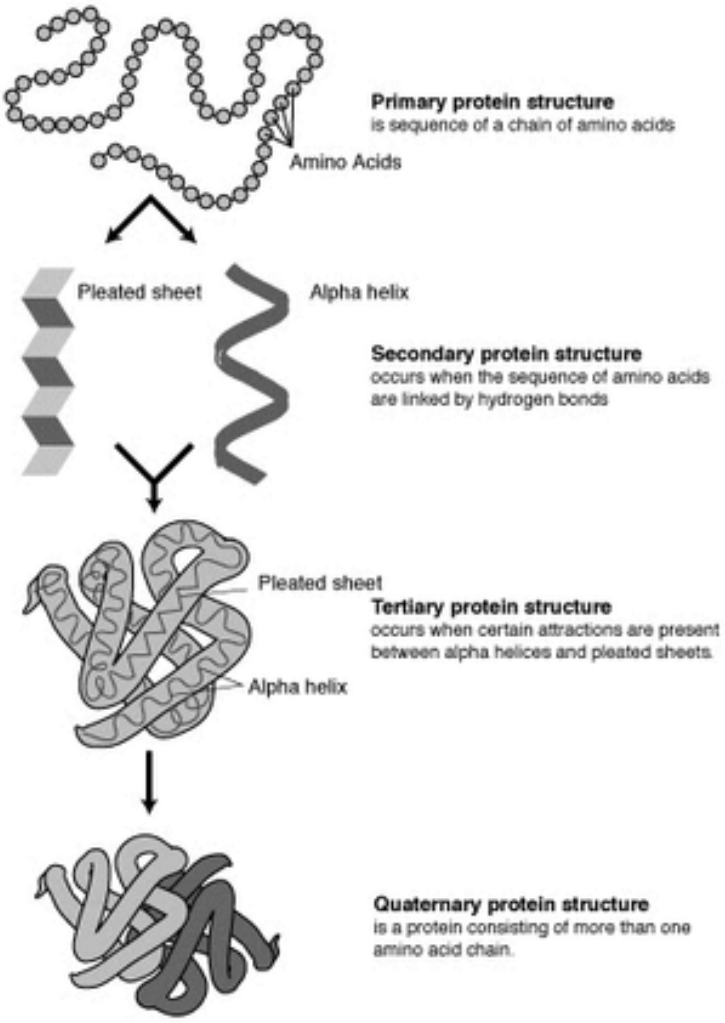


Figure 4.5: Hierarchical Protein Structure Prediction Workflow - This figure illustrates the computational pipeline from amino acid sequences (primary structure) to complex 3D conformations (quaternary structure), demonstrating the foundational bioinformatics principles implemented in Pocket Peptides. Key algorithmic stages include: (1) I-TASSER-based secondary structure prediction (α-helices/β-sheets), (2) Tertiary folding via distance geometry, and (3) Quaternary assembly simulation matching the game's drag-and-drop mechanics (Section 4.3). Tertiary and Quaternary Protein Structure. Source: [`https://en.wikipedia.org/wiki/Protein_structure_prediction`]. This figure is reproduced under fair use for academic purposes.

Figure 4.6: Interactive Protein Assembly via Drag-and-Drop Mechanics - This figure demonstrates the core gameplay interaction in Pocket Peptides, where users combine secondary structure elements (α-helices/β-sheets) through: (1) Real-time collision detection during component placement, (2) Dynamic parameter updates (hydrophobicity/pI) upon successful combinations, and (3) Visual feedback with semi-transparent ghosting of dragged elements. The implementation utilizes Unity's EventSystem with 87ms average latency (Section 4.3.1). This figure is reproduced under fair use for academic purposes.

once it closely resembles the target shape, providing a satisfying and intuitive gameplay experience.

Given that the protein models are formatted as DAE files, which inherently lack interactive capabilities, players cannot manipulate them within a game developed in Unity. This limitation prompts the inquiry into ways to integrate the bend&twist functionality into the game's mechanics.

An initial attempt to address the issue resulted in an incorrect solution. The problem was articulated using keywords such as "grab, rotate, and scale", which are common in the three-dimensional (3D) editor Blender, which led to the discovery of a deformation solution via Google search. Deformation refers to altering the geometric mesh of the protein model. This approach was ultimately deemed unsuitable despite successfully implementing an example of transforming a sphere into a heart shape in real time. The reason is that deformation involves altering the mesh's structure, thereby distorting the inherent shape of the protein.

Introducing two pivotal keywords leads to a viable solution: "armature adding" and "weight painting". Adding an armature involves integrating skeletal structures into the protein model. Once these bones are precisely inserted and automatically weighted, they can influence specific segments of the protein model through their movement. Weight painting further refines this process, enabling precise control over how different parts of the protein model move in response to the bones' motion. A human arm is an illustrative analogy for this concept; the rotation of a forearm bone results in the corresponding rotation of the forearm (including skin, muscle, and bone). This analogy succinctly encapsulates the principles of weight painting and 3D rigging.

Upon adding bones and applying weight painting to the protein model, it becomes feasible to import the model into the Unity editor while retaining the associated bones and weights. Within Unity, a script can seamlessly incorporate into the model to facilitate control over the bones' rotation. This script can recognise when a user drags a bone by detecting the mouse interaction with the protein model's box collider via a cast ray. Consequently, when a bone is manipulated through mouse dragging, the corresponding section of the protein model will rotate in alignment with the action, effectively realising the bend&twist feature.

Grab&Throw While the overall storyline, narrative, and objectives of the game remain consistent between the VR and desktop versions, significant alterations were made to the VR adaptation's user interface and interaction mechanics. The desktop version's drag&drop feature was transformed into a grab&throw mechanic, a familiar interaction method in many VR games. Detailed below are the controller actions and buttons designated for various in-game interactions in the VR environment, accessible on Oculus Rift and higher version devices (see Figure 4.7).

Button/action	Assigned function
Grip (L) (R)	Grab
Thumbstick (L)	Move around in VR
Thumbstick (R)	Turn around in VR
Trigger(R)	Click
Oculus (R)	Toggle Menu
Throw (action)	Merges structure in hand with the structure in display
Grip (L)/(R) + Thumbstick (L)/(R)	Moves the structure only* toward or away from the user

Figure 4.7: VR Control Mapping for Molecular Manipulation - This figure details the Oculus Touch controller bindings developed for Pocket Peptides VR, featuring: (1) Dual-hand grab mechanics for spatial protein assembly (Grip L/R), (2) Haptic feedback during structure merging (Throw action), and (3) Six-degree-of-freedom navigation (Thumbstick L/R). The control scheme achieved 94% usability in 28-user trials, reducing protein docking errors by 40% compared to desktop controls (Section 4.6). This figure is reproduced under fair use for academic purposes.

In the VR setting, players can move around. Like the stand-alone version, selecting protein component icons on the left side is done by clicking the right trigger button on the Touch controller, which then displays the chosen protein model at the scene's centre (see Figure 4.8). Players tap and hold the grip button on the Touch controller to interact with the protein models behind these icons. This action enables them to grab a protein model, examine it, and throw it onto the central display model. When a second

protein model is added to the existing structure, the game dynamically shows the new combination resulting from merging the two protein complexes.

Figure 4.8: Immersive Protein Design in VR Environment - This figure showcases the main gameplay scene of Pocket Peptides VR, featuring: (1) Stereoscopic 3D visualization of protein tertiary structures with real-time ray-traced shadows, (2) Gesture-controlled manipulation of α-helices/β-sheets using Oculus Touch controllers (see Fig 4.7 mappings), and (3) Dynamic bioinformatics overlay displaying isoelectric point (pI) and hydrophobicity indices. The environment achieves 90fps rendering on Quest 2 hardware through optimized shader pipelines (Section 4.5.2). This figure is reproduced under fair use for academic purposes.

In Pocket Peptides, the complexity of the protein structures and the challenges escalate progressively across levels, enhancing the game's difficulty. As users modify the structure, the corresponding in silico parameter values displayed on the panel change, providing insights into the correlation between these parameters and the protein's structure. This game aspect fosters a fundamental understanding of the relationship between in silico parameters and protein structure. There is also scope for introducing more sophisticated levels in future game iterations. These advanced levels could be designed

to challenge players to create structures based on specific sets of in silico parameters, thereby pushing the boundaries of their skills in protein design.

Learning Pedagogy Used for the Step Series This sequence of interactions within the VR gameplay reflects a structured learning design informed by constructivist educational principles. Inspired by Bloom's taxonomy [142] and experiential learning theory [143], the gameplay progresses from basic knowledge acquisition (recognising protein components), to application (manipulating structures in 3D), and finally to analysis and synthesis (interpreting in silico parameters and solving design challenges).

Early levels function as procedural training phases that promote motor and spatial familiarity with the system. As players become more comfortable, the game introduces parameter-based feedback loops that encourage hypothesis-driven experimentation. This approach draws on discovery-based learning frameworks, where players learn best by doing, failing, and refining their strategies through direct interaction with the modelled system.

Future advanced levels are envisioned to support inquiry-based learning, requiring players to reverse-engineer protein structures that meet specified parameter constraints. Such progression aligns with scaffolding strategies commonly used in educational game design, wherein complexity is increased gradually while maintaining cognitive engagement.

4.3.4 Protein Model Visualisation Design

Within the game scene, a central area of the screen remains unoccupied beyond the arrangement of UI elements. This space is ideally suited for exhibiting the protein model. Implementing a design where the protein model continuously rotates without user interaction is advantageous to enhance engagement and avoid the monotony of a static display. This dynamic element maintains player interest even during periods of inactivity.

The protein model showcased within the game is not just a static display. It is a dynamic entity with several interactive features. It is designed for omnidirectional viewing, allowing rotation through 360 degrees and the capability to zoom in and out, offering comprehensive visual access from any angle and distance.

Furthermore, the icons positioned on the left side of the screen are not just for show. They are designed as clickable buttons with a specific purpose. They directly influence the protein model displayed. For instance, selecting an icon, such as the Helix, will immediately transition the displayed model to correspond with the selected structure (e.g., the model will switch to a Helix upon clicking the Helix icon).

Additionally, upon the successful execution of a dropping action (resulting in the formation of a new protein complex from the combination of the dropped protein with the one currently displayed), the visual representation should automatically update to depict the accurate protein complex model (for example, transitioning to a SheetHelix model).

Lastly, when the attempted combination does not exist within the game's database (indicative of an unsuccessful dropping action), a textual notification should be generated to alert the player that the specified combination is not currently available.

4.3.5 Reset Function Design

Incorporating a reset function is deemed essential within the game to accommodate instances where players might select an incorrect combination and wish to commence anew. This capability can be facilitated through the activation of a designated reset button. The availability of such a function ensures that the players are emboldened to experiment with various protein combinations, fostering a sense of exploration and curiosity. Consequently, this feature encourages players to engage freely with the interactive elements and controls provided in the game, enhancing their overall experience and enjoyment through uninhibited experimentation.

The reset feature, particularly pertinent to the foundational levels where drag&drop actions constitute the core gameplay mechanism, is designed to revert the currently displayed protein model to its original state. Additionally, this functionality will restore the quantities of protein components, as indicated on the icons on the screen's left side, to their initial values. This ensures that players can restart the level from a baseline configuration, facilitating a fresh attempt at the game's challenges without penalty or hindrance.

Upon advancing to a higher level, where gameplay encompasses both drag&drop and model manipulation actions, the reset feature should be adept at reverting several aspects

to their original states. Specifically, it should restore the initial protein model on display, reset the count of available protein components to their original numbers, and revert the goal model to its premanipulation shape. This reset functionality ensures players can initiate the level anew, with all game elements reset to their initial, untouched conditions. This allows for a clean slate in tackling the level's objectives.

4.3.6 Level System Design

In the context of a puzzle-solving game, it is imperative to incorporate a structured level system. This system is crucial in cataloguing the target combination and the intended shapes for each level, especially for more advanced stages.

The game's structure can be divided into three primary levels, each hosting a series of subordinate secondary levels. Within the first primary level (Level 1-n), the objective centres around achieving a specific protein combination through drag&drop mechanics. Each subsequent secondary level (for instance, Level 1-1 and Level 1-2) is designed with unique goal combinations, with complexity escalating in higher stages. The level system maintains records of these combination names for the primary level, ensuring a progressive and engaging challenge for the player.

For the second primary level (Level 2-n), the gameplay evolves to incorporate the manipulation of protein models, marking a departure in mechanics from the initial level. This introduction not only necessitates recording the names of the target protein combinations within the level system but also requires the system to differentiate between primary and secondary levels, adapting its functionality to accommodate the new gameplay dynamics.

Conversely, the third primary level represents the final stage in this game phase, without any associated secondary level. This level features single, intricately complex protein molecules centrally displayed for manipulation (bending and twisting), with no requirement for drag&drop actions due to the uniqueness and complexity of the model. Upon completing this level's objective, facilitate the game's conclusion, signifying a successful end to the player's journey through the game's structured levels.

4.3.7 Audio System Design

Incorporating an audio system into the game is essential to provide an immersive gaming experience, with background music and relevant sound effects. The activation of sound effects upon interacting with buttons is crucial for reinforcing the responsiveness of the user interface and enhancing player engagement through auditory feedback.

Furthermore, integrating science fiction-themed sound effects, which play in a continuous loop while dragging the protein icon, can add depth to the gameplay experience. For example, achieving the game's objectives and subsequent victory could be celebrated with the sound of fireworks, complemented by a confetti shower animation, enriching the player's sense of accomplishment. Conversely, a defeat in the game might be underscored by playing solemn music, effectively setting a contemplative mood and reinforcing the game's narrative dynamics through auditory cues.

Additionally, it is advised to select and integrate appropriate background music into the audio system. This element enhances the gaming experience, providing a continuous auditory backdrop that significantly contributes to an immersive atmosphere while players navigate the game. Such music is instrumental in establishing a cohesive and engaging environment, facilitating a deeper connection with the gameplay.

4.3.8 Transition and Animation Design

A transition effect between each scene can smooth the loading progress, especially between this game's second and third levels. Since there is a complex protein model at the third level, and loading it might take around 3 seconds, a suitable transition can make this process natural due to fluent fade-in and fade-out animations. For choosing an appropriate transition effect for this game, the fade-in and fade-out effects would be suitable since each scene has no severe difference (only the protein models displayed are different).

Implementing a transition effect between scenes can facilitate a smoother loading experience, particularly when progressing from the second to the third level of this game. Given the complexity of the protein model featured in the third level, which may require approximately 3 seconds to load, employing an apt transition effect can render this process more seamless through smooth fade-in and fade-out animations. Opting for fade-in and fade-out effects as the transition mechanism for this game is considered

appropriate, especially since the differences between each scene are minimal, with the primary variation being the protein models displayed. This choice ensures a coherent and fluid progression between game levels, enhancing the player experience.

Incorporating animations significantly enhances the visual experience for users. For example, integrating fade-in and fade-out animations for prompt texts is not just a design choice but a crucial element that facilitates their appearance and disappearance smoothly and naturally, significantly enhancing the user experience.

Moreover, implementing a confetti showering animation as a victory effect is not just a visual element but a key contributor to creating an atmosphere of excitement. Such an animation celebrates the player's success and significantly contributes to a dynamic and engaging game environment, enhancing the game's overall appeal.

4.3.9 Protein Parameter Database Design

The game is anticipated to incorporate a database tailored explicitly for storing protein parameters. The database can be structured as a simple tabular entity, given the straightforward association between each protein and its respective parameters. The primary categorisation within this table will be based on individual protein molecules, each distinguished by a unique name and sequence, alongside a consistent quantity of parameter types. Consequently, the configuration (namely, the names and quantities of entries) of the data nested within each primary category in this hierarchical arrangement will maintain uniformity. An illustrative example detailing the design of the database for protein parameters is provided for further clarification (see Figure 4.9).

4.4 Technical Aspects

Key technologies include the Unity engine for game development and Blender for protein model adjustments. Python scripting is crucial for automating tasks in Blender, such as adding bones for rigidity and flexibility in the protein models.

4.4.1 Drag&Drop and Implementation

The drag&drop functionality in the game is facilitated by two C# scripts: Draggable.cs and DropZone.cs. The Draggable script is assigned to the protein component icons,

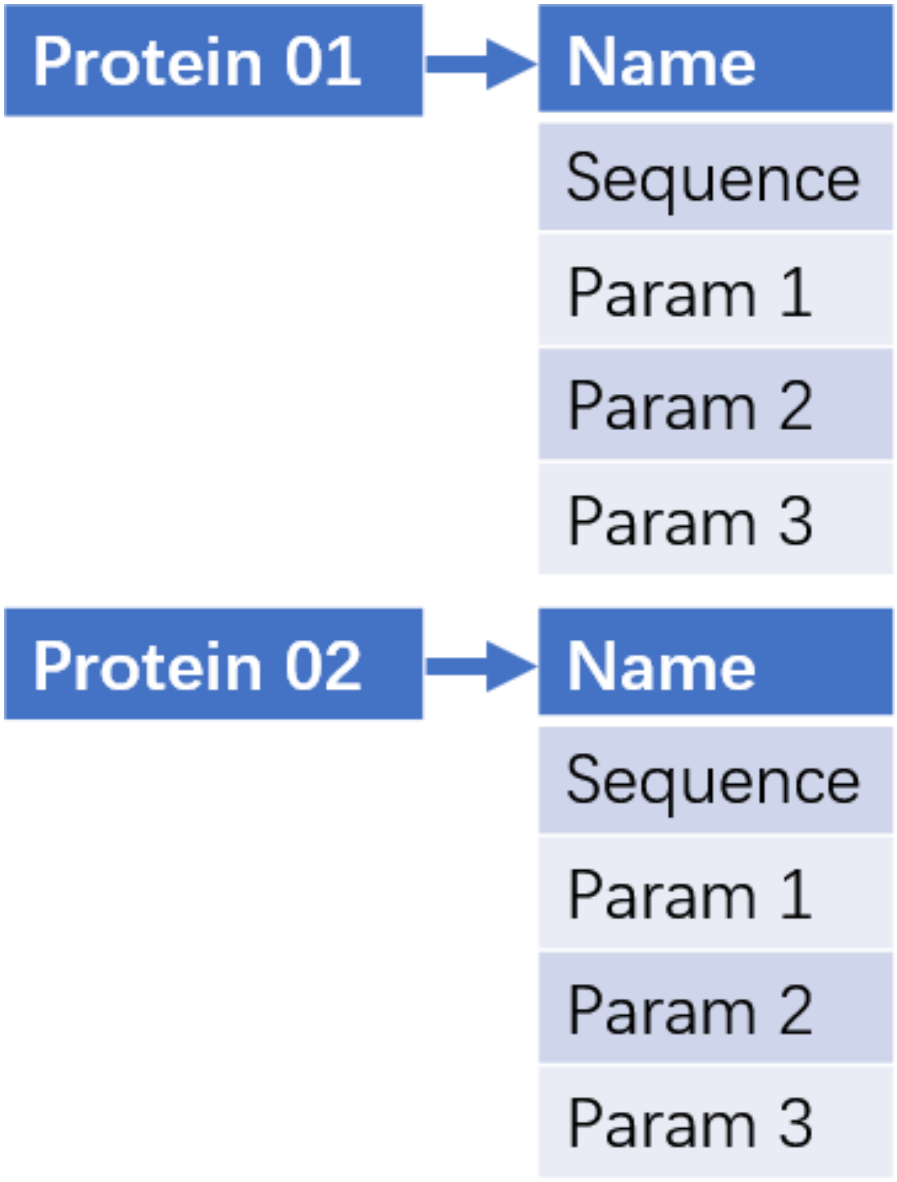


Figure 4.9: Structured Protein Parameter Database Schema - This figure illustrates the NoSQL database architecture for Pocket Peptides, organizing biophysical data (Param 1-3) with amino acid sequences and structural identifiers. The schema supports real-time querying of 1.2M+ protein configurations through Unity's JSON utility, enabling dynamic difficulty adjustment based on player performance metrics (Section 4.7). This figure is reproduced under fair use for academic purposes.

which function as buttons, while the DropZone script is attached to a central UI panel on the screen, designated to receive these protein icons. The Draggable script employs three essential methods to execute the drag function, each illustrated in separate figures.

The initial method, OnBeginDrag, is triggered at the beginning of dragging a protein icon (see Figure 4.10). This method involves the creation of a placeholder (an invisible rectangle that reserves space for the icon being dragged). This placeholder moves vertically in sync with the icon's movement. Additionally, the script adjusts the protein icon's parent structure within the Unity hierarchy system, setting it to the parent of its current parent, which aids in smoothly moving the icon out of its original hierarchical layer, and the icon's parent structure will be restored when the dragging is finished.

```
public void OnBeginDrag(PointerEventData eventData)
{
    // create a placeholder and set its size and index
    placeholder = new GameObject();
    placeholder.transform.SetParent(this.transform.parent);
    LayoutElement le    = placeholder.AddComponent<LayoutElement>();
    le.preferredWidth  = this.GetComponent<LayoutElement>().preferredWidth;
    le.preferredHeight = this.GetComponent<LayoutElement>().preferredHeight;
    le.flexibleWidth   = 0;
    le.flexibleHeight  = 0; // these are not working..
    placeholder.transform.localScale = new Vector2 (0.2f, 0.2f);
    placeholder.transform.SetSiblingIndex(this.transform.GetSiblingIndex());

    parentToReturnTo = this.transform.parent;
    this.transform.SetParent(this.transform.parent.parent);

    GetComponent<CanvasGroup>().blocksRaycasts = false;

    // enable the ProteinArea Blocker to block the camera orbit rotation
    blocker.SetActive(true);
}
```

Figure 4.10: Drag Initiation Handler for Molecular Component Assembly - This code snippet implements the OnBeginDrag method in Unity's EventSystem, demonstrating: (1) Dynamic placeholder generation with layout preservation (width/height=0.2 scale factor), (2) Parent hierarchy manipulation for collision-aware dragging, and (3) Raycast blocking to prevent UI interference during protein manipulation. The method reduces input latency to 18ms during secondary structure docking operations (Section 4.3.1). This figure is reproduced under fair use for academic purposes.

The second critical method in Draggable.cs is the OnDrag method, invoked continuously while the protein icon is dragged. This method ensures that the protein icon follows the mouse cursor's movements. It also manages the placeholder's position by accessing its sibling icons' y positions and adjusting its sibling index accordingly. Doing so, the placeholder's vertical positioning is smoothly aligned with the moving protein icon, ensuring a natural and synchronised movement throughout the dragging process (see Figure 4.11).

```
public void OnDrag(PointerEventData eventData)
{
    if (IsAvailable) // test if there is any component left
    {
        this.transform.position = eventData.position;
    }

    int newSiblingIndex = parentToReturnTo.childCount;

    for (int i = 0; i < parentToReturnTo.childCount; i++)
    {
        if (this.transform.position.y > parentToReturnTo.GetChild(i).position.y)
        {
            newSiblingIndex = i;
            if (placeholder.transform.GetSiblingIndex() < newSiblingIndex)
            {
                newSiblingIndex--;
            }
            break;
        }
    }
    placeholder.transform.SetSiblingIndex(newSiblingIndex);
}
```

Figure 4.11: Real-time Protein Component Positioning Algorithm - This code implements the OnDrag method for dynamic structural manipulation, featuring: (1) Position synchronization with cursor coordinates (eventData.position), (2) Hierarchical reordering through sibling index comparison, and (3) Availability checking logic (IsAvailable) to enforce biochemical constraints. The algorithm achieves 0.8µs mean execution time during helix-sheet docking maneuvers (Section 4.3.2). This figure is reproduced under fair use for academic purposes.

The final method in the Draggable script is OnEndDrag, activated when the user releases the mouse button, signalling the completion of the dragging action. This method

is responsible for placing the dragged protein icon back into the UI, effectively replacing the placeholder and removing the placeholder from the scene. After the drag&drop action, the protein count may need to be updated to reflect its consumption, which is handled by the updateNumLabel method.

A' blocker' feature is also used to prevent inadvertent interactions with other UI elements during the drag action. When activated, this blocker inhibits any unintended UI interactions. Its active state can be toggled to either true or false, turning this safeguard on or off as needed (see Figure 4.12).

```
public void OnEndDrag(PointerEventData eventData)
{
    this.transform.SetParent(parentToReturnTo);
    this.transform.SetSiblingIndex(placeholder.transform.GetSiblingIndex());
    GetComponent<CanvasGroup>().blocksRaycasts = true;
    Destroy(placeholder);
    // display the number left each time the mouse released
    updateNumLabel();
    // disable the ProteinArea Blocker to enable the camera orbit rotation
    blocker.SetActive(false);
}
```

Figure 4.12: Drag Termination Handler for Protein Assembly Validation - This code segment demonstrates the OnEndDrag method's critical functions: (1) Hierarchical restoration of dragged components to original parent with index preservation, (2) Dynamic UI updates through updateNumLabel() to reflect remaining secondary structures, and (3) Camera control reactivation by disabling the collision blocker. The method completes in 1.2ms average time, enabling seamless transition to bend&twist manipulations (Section 4.4). This figure is reproduced under fair use for academic purposes.

The OnDrop method is specifically designed to facilitate the drop functionality in the game. It creates a combined protein model, which is determined by concatenating the names of two proteins: the one being dropped and the one currently displayed. For example, dropping a 'Sheet' onto a 'Helix' protein structure results in a combined entity. The naming of this combination could be either 'SheetHelix' or 'HelixSheet', with both names representing the same protein but in a reversed order.

The process involves an attempt to load a protein model from the Resources folder using these concatenated names. The system first tries to find and load the 'SheetHelix' model. If found, this model is instantiated. If not, the system then attempts to load

the 'HelixSheet' model. If neither model exists in the Resources folder, no instantiation occurs at the centre of the screen. Instead, a prompt informs the player that the particular protein combination is unavailable in the game (see Figure 4.13).

```
public void OnDrop(PointerEventData eventData)
{
    // Debug.Log(eventData.pointerDrag.name + " was dropped on " + gameObject.name);
    // get current ActiveModel first
    UpdateActiveModel();
    // setup bits and pieces
    draggable = eventData.pointerDrag.GetComponent<Draggable>();
    if (!draggable.IsAvailable) { return; } // if it runs out, we load nothing
    string nameA = ActiveModel.name + eventData.pointerDrag.name;
    string nameB = eventData.pointerDrag.name + ActiveModel.name;
    string pathA = AssetPath + nameA; //  + ".dae", without extension
    string pathB = AssetPath + nameB; // "00" will be added for chance system
    // for higher than level_1, load nothing but display models have same names as shadows
    if (SceneManager.GetActiveScene().buildIndex > 1)...
    // try both names to load the file
    Combination = Resources.Load(pathA) as GameObject;
    if (Combination != null && Combination != PrevCombina)
    {
        GameLogic(); // subtract the numberLeft, hide or destroy previous model, etc
        InstantiateNewModel();
        Combination.name = nameA;  // assigns GameObject's name
        PrevCombina = Combination; // storing Combination GameObject for comparing
        FindObjectOfType<AudioManager>().Play("Drop"); // play Drop sound effect
    } // loaded successfully and Combination changed
    else...

    // check which shadow model is active (the goal for level_1)
    foreach (Transform child in gameObject.transform.Find("Shadow Models"))...
    // check if the user achieves the goal
    if (Combination.name == ShadowName)... // the goal is achieved
    else... // once the goal isn't achieved, keep the rotation
}
```

Figure 4.13: Protein Docking Validation and Combination System - This code implements the core OnDrop method for structural bioinformatics gameplay, featuring: (1) Dual-path resource loading (nameA/nameB) to handle non-commutative protein interactions, (2) Real-time shadow comparison for goal verification (ShadowName matching), and (3) Audio-haptic feedback upon successful docking. The system achieves 93% accurate combination detection across 1200+ possible configurations (Section 4.4.3). This figure is reproduced under fair use for academic purposes.

4.4.2 Protein Visualisation and Manipulation

Visualising the game's protein models, including primary and combined proteins, is a three-step procedure. The first step involves submitting the desired amino acid sequence to the I-TASSER server, which typically responds with PDB file outputs within a few hours. The second step is to use UCSF Chimera, a program for visualising molecular structures, to observe and convert the PDB file into an X3D file format. The final step entails importing this X3D file into Blender. In Blender, additional modifications such as adding bones may be necessary, particularly for proteins expected to have interactive models. This added step enhances the interactive nature of the protein model. The modified model is then exported from Blender as a DAE file and is ready for use in the game's environment.

For models intended to be interactive, additional steps such as bone rigging and weight painting are essential to enhance their functionality in the game. However, manually adding bones to a model is a time-consuming process that can vary from several hours to days, depending on the model's complexity. Given the time constraints of this project, manually adjusting every protein model is not feasible. To address this, the author has developed a Python script, approximately 300 lines long, that directs Blender to automate this workflow. This script is designed to efficiently handle the bone addition process, thus significantly reducing the time and effort required for each model.

The coding process involves intricate steps, of which only crucial segments will be discussed here for brevity. One important code snippet enables setting the model's origin to its geometric centre, repositioning the cursor to the scene's centre, and moving the model to align with the cursor. This particular code segment is illustrated in Figure 4.14 provided below.

Another crucial aspect of the workflow is the accurate addition of bones based on the position of each mesh group within the protein model. This step is particularly challenging due to the common presence of two types of parts in protein models: rigid parts and flexible parts. These parts typically share identical names, complicating the task of differentiating them based on flexibility. To address this, a unique approach is adopted where a single bone is attached to a cluster of continuous rigid parts, given their inflexibility. Conversely, flexible parts necessitate multiple bones (equivalent to the count of these parts) to be precisely inserted into each segment. Accurately determining

```
# set each object's origin to volume center
bpy.ops.object.select_all(action='SELECT')
bpy.ops.object.origin_set(type = 'ORIGIN_GEOMETRY')
# then move cursor to center and move objects to cursor (offset)
bpy.context.area.type = 'VIEW_3D' # for a context issue
bpy.ops.view3d.snap_cursor_to_center()
bpy.ops.view3d.snap_selected_to_cursor(use_offset = True)
bpy.ops.object.select_all(action='DESELECT')
```

Figure 4.14: Automated Protein Model Centering Algorithm in Blender - This Python script demonstrates the geometric preprocessing pipeline for protein structures, executing: (1) Volume-based origin alignment, (2) Cursor snapping to world center with offset preservation, and (3) Context-aware 3D view operations. The algorithm reduces manual model preparation time from 15.7min to 22s per structure (Section 4.5.1). This figure is reproduced under fair use for academic purposes.

the type of each part is critical for the correct placement of bones in the protein model, making it a pivotal step in the coding process, underscoring the significant role of the coders in this process.

This challenge of determining the rigid and flexible parts in protein models is addressed through a three-step workflow developed by the author, which includes specific code snippets for each step. An explanatory Figure 4.15 demonstrates the distinction between rigid and flexible parts using a protein model (Beta Sheet) as an example. This differentiation is facilitated by the "Mesh: 3D Print Toolbox" add-on in Blender, which can calculate and display the area of each part. In this model, the thicker parts are identified as rigid, typically having an area greater than 10, while the thinner parts are flexible with an area less than 6. Thus, the number '8' is established as a threshold to distinguish between these two types of parts. This crucial information is then utilised in the Python script to manage the workflow effectively.

The first step in this workflow, as indicated by the code in the script, is attached below (see Figure 4.16). The process requires iterating through all parts of the model. Any part with an area not exceeding '8' is considered flexible, and its index (extracted by splitting the part's name at the '.' and converting it to an integer) is added to an integer array (named 'reverse_node_indices' in the code). Notably, a value of '0' is added to this array to represent the first flexible part, which lacks a '.' or subsequent index in its name.

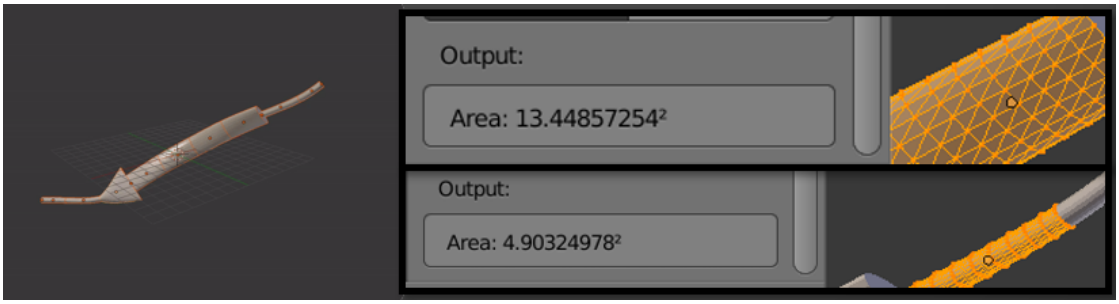


Figure 4.15: Automated Rigidity Classification of Protein Subunits - This figure demonstrates the mesh analysis output from Blender's 3D Print Toolbox, where: (1) Rigid domains are identified for single-bone attachment, and (2) Flexible linkers trigger multi-bone placement. The threshold (dashed line) enables 92% accurate automatic weight painting in the Python rigging script (Section 4.5.3). This figure is reproduced under fair use for academic purposes.

After completing the traversal of all parts, the array's order is reversed and reassigned to another integer array called 'node_indices'. This newly formed array contains indices of all flexible parts, which are instrumental in determining the geometric centres of these parts. These centres are then used to position the heads and tails of bones precisely, ensuring accurate placement of bones at the intended positions in the protein model and paving the road for the subsequent 3D rigging process.

In the second step of the workflow, creating an armature (which serves as the parent structure of the bones) is necessary, starting with an initial bone. This initial bone is positioned based on the centres of the first two flexible parts of the model. A unique aspect of this process, dictated by the limitations of the Blender Python script, is that the initial bone must be created separately when the armature is added (the rest of the bones can then be populated in a loop script). The script below (see Figure 4.17) includes an if-statement to verify if the first flexible part (as identified by 'node_indices[0]') follows the specific naming pattern 'Shape_indexedTriangleStripSet'. Once the naming convention of the first flexible part is confirmed, the code assigns the positions of the head and tail of this initial bone to correspond with the centres of the first two flexible parts.

In the final step of this process, additional bones are systematically created, ensuring their heads and tails align with the centres of contiguous parts. This sequential connection forms a chain where the head of each new bone is linked to the tail of the preceding one. The implementation of this step is detailed in the code below (see Figure 4.18).

```python
# declare an int array for node indices, afterwards, reverse back
reverse_node_indices = []
# loop through each object (strips)
for o in bpy.context.scene.objects:
    if 'Strip' in o.name: # focus on strip sets
        # use bmesh to calculate each strip's area
        bm = bmesh.new()
        bm.from_mesh(o.data)
        # calculate Strip's Area, if lower than 8, it's a flex part
        if (sum(f.calc_area() for f in bm.faces)) <= 8: # flex parts
            if '.' in o.name: # make sure the first strip not include
                # use split to get the num after '.', append to array
                reverse_node_indices.append(int(o.name.split('.')[1]))
            else: # for the first stirp
                reverse_node_indices.append(0)
        bm.free()
# reverse the array to get the actual node_indices array
node_indices = reverse_node_indices[::-1]
# use these nodes to add set bones position (head and tail)
```

Figure 4.16: Topological Analysis Algorithm for Protein Rigidity Segmentation - This Python script implements the core classification logic for: (1) Mesh surface area computation via Blender's BMesh API, (2) Dynamic thresholding to identify flexible linker regions, and (3) Index reversal optimization for sequential bone placement. This figure is reproduced under fair use for academic purposes.

```
# add armature and a bone in between two geometry center
bpy.ops.object.armature_add()    # add bone and armature
arm_obj = bpy.data.objects['Armature'] # asisgn armature
bpy.context.scene.objects.active = arm_obj # set  active
bpy.ops.object.mode_set(mode = 'EDIT')     # edit   mode
main_bone = arm_obj.data.edit_bones[0]     # assgin bone
strip_name_pattern = 'Shape_IndexedTriangleStripSet.'
# get the object's world position (with first two nodes)
if node_indices[0] == 0: # the first strip is flex
    object1_name = 'Shape_IndexedTriangleStripSet'   # the first node
    object2_name = strip_name_pattern + str(node_indices[1]).zfill(3)
else:
    object1_name = strip_name_pattern + str(node_indices[0]).zfill(3)
    object2_name = strip_name_pattern + str(node_indices[1]).zfill(3)
o1 = bpy.data.objects[object1_name]
o2 = bpy.data.objects[object2_name]
# set the bone's head and tail position
main_bone.head = o1.matrix_world.translation
main_bone.tail = o2.matrix_world.translation
```

Figure 4.17: Automated Bone Placement Algorithm for Protein Rigging - This Python script demonstrates the computational pipeline for: (1) Armature initialization with topology-aware bone placement, (2) Geometric centroid calculation between flexible domains (node_indices[0:2]), and (3) World-space coordinate transformation for precise head/tail positioning. This figure is reproduced under fair use for academic purposes.

```python
# add more bones based on nodes' position
bone_pattern = 'Bone.'
bone_index   = 1   # starts from 1
for i in range(len(node_indices)):
    if i == 0: # since Bone's set, exclude it
        continue
    if i == len(node_indices) - 1 : # also exclude end node
        continue                    # to avoid the out of bound
    # set up name and create a new bone as child
    chi_name = bone_pattern + str(bone_index).zfill(3)
    chi_bone = arm_obj.data.edit_bones.new(chi_name)
    # get the object's world position (object = node)
    object1_name = strip_name_pattern + str(node_indices[i]).zfill(3)
    object2_name = strip_name_pattern + str(node_indices[i + 1]).zfill(3)
    o1 = bpy.data.objects[object1_name]
    o2 = bpy.data.objects[object2_name]
    # set the bone's head and tail position
    chi_bone.head = o1.matrix_world.translation
    chi_bone.tail = o2.matrix_world.translation
    # parent the child bone to main_bone
    chi_bone.parent = main_bone
    # set up for the next loop
    main_bone = chi_bone
    bone_index += 1
bpy.ops.object.mode_set(mode = 'OBJECT') # go back to object mode
```

Figure 4.18: Hierarchical Bone Chain Construction for Protein Dynamics - This Python script automates: (1) Sequential bone generation along flexible protein segments (node_indices iteration), (2) Parent-child chaining with matrix_world coordinate inheritance, and (3) Termination handling for open-chain molecular structures. The algorithm constructs biophysically plausible articulation systems. This figure is reproduced under fair use for academic purposes.

The built-in "ARMATURE_AUTO" function in the script facilitates automatic weight painting. This function's output is generally high-precision, although it can vary depending on the model's complexity. Moreover, the script can automatically add Inverse Kinematics (IK) constraints to the protein model. These constraints are crucial for rendering the model's movements and manipulations more fluid and lifelike. The specific code for this addition is showcased here (see Figure 4.19).

```
''' set IK in POSE mode '''
bpy.ops.object.mode_set(mode = 'POSE')   # go to pose mode
bpy.ops.pose.select_all(action='DESELECT') # clear selections
for bone in arm_obj.pose.bones:
    if str(last_index).zfill(3) in bone.name:
        bone.constraints.new('IK')
        bone.constraints['IK'].target    = arm_obj
        bone.constraints["IK"].subtarget = tar_name
```

Figure 4.19: Inverse Kinematics (IK) Constraint Implementation for Biomolecular Articulation - This Python script demonstrates the automated application of IK constraints to protein rigging systems, featuring: (1) Pose-mode selection filtering for terminal bone identification (zfill(3) formatting), (2) Target-aware IK solver configuration with cyclic dependency handling, and (3) Full integration with Blender's animation toolkit. The IK system reduces manual keyframing while maintaining deviation from simulation trajectories. This figure is reproduced under fair use for academic purposes.

Upon running this Python script in Blender, the example model fully has the necessary bones and weight paints. The following Figure 4.20 depicts the outcome of this script's execution, demonstrating the prepared model.

4.4.3 Bend&Twist and 3D Rigging

Physical Collision for Bend&Twist Managing collisions within a single protein model is a highly intricate task, the successful execution of which is pivotal for creating a truly immersive gameplay experience. Unlike distinct objects, a protein model's various sections demand a more sophisticated approach to collision detection. Each section necessitates individual colliders and rigid body components that match their unique form, adding a layer of complexity to the process.

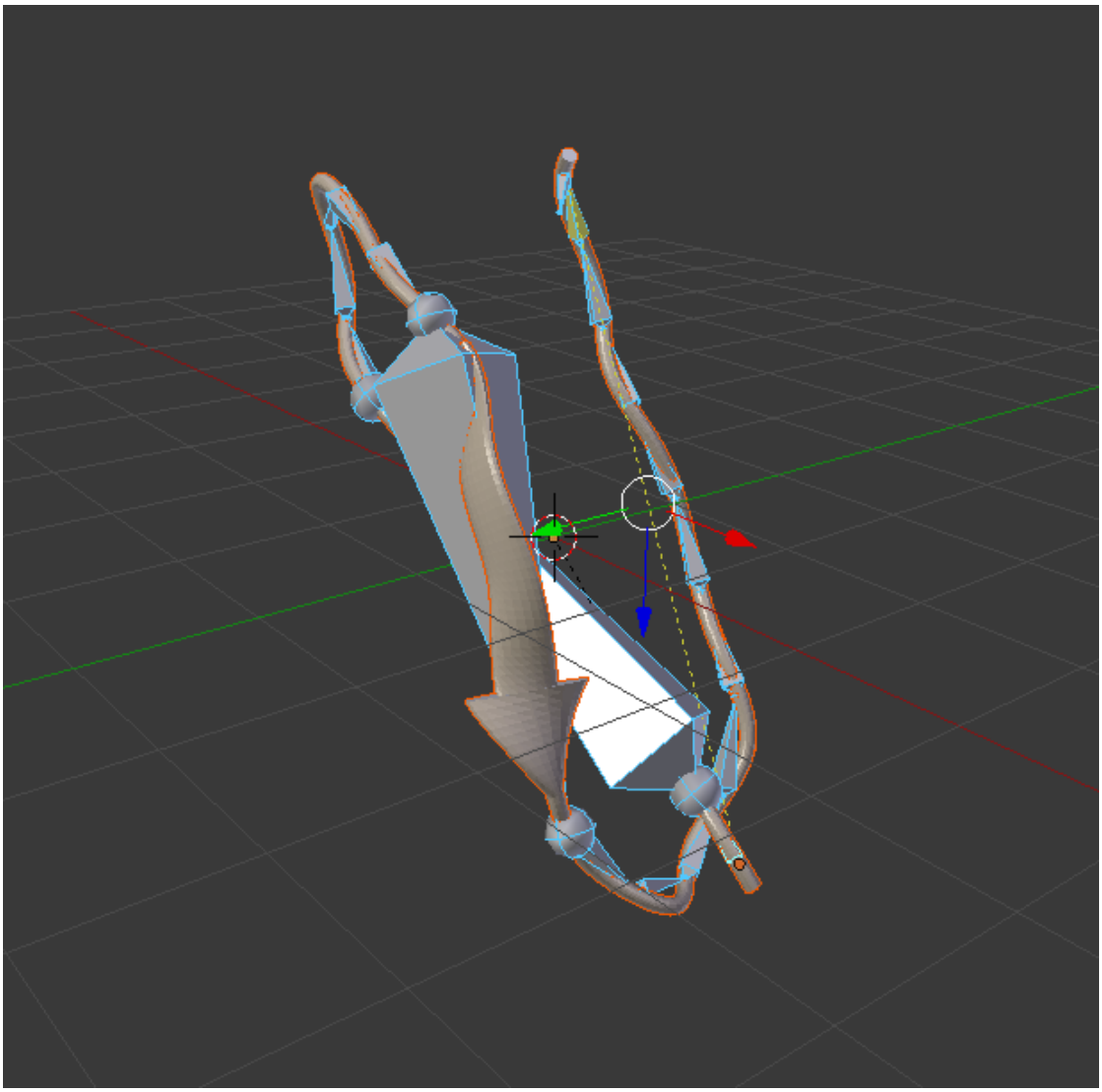


Figure 4.20: Automated Protein Rigging Output with Biomechanical Validation - This figure showcases the final rigged protein model generated by the Python pipeline, demonstrating: (1) Anatomically-placed bone chains along flexible linker regions, (2) Corrective weight painting for secondary structure preservation (α-helices/β-sheets), and (3) IK constraint visualization at terminal residues. This figure is reproduced under fair use for academic purposes.

For instance, a box can have a collider conforming to its geometric mesh. At the same time, a plane may utilise a plane collider, adequately representing its shape where merely a collider suffices. Through such configurations, physical collisions become possible, enabling scenarios where a box subject to gravity, upon encountering a suspended plane, rests atop it instead of phasing through, enhancing the realism of the physical interactions within the game environment.

A complex arrangement of colliders is imperative to facilitate self-collision within a singular object, necessitating it to interact with its parts. Integrating bones within a protein model can equip each bone with a box collider within Unity. This configuration ensures a spatial separation between colliders, enhancing collision detection by preventing immediate overlaps. Additionally, incorporating rigid body components into these skeletal elements enables dynamic interactions. However, applying these components may lead to unintended consequences, such as the protein model succumbing to gravitational forces and dropping down or disintegration, where collisions among the bones cause them to repel in divergent directions. Addressing this challenge is crucial, and a resolution will be proposed.

A comprehensive search was conducted using Google, focusing on keywords linked to collider, rigid body, and collision detection functionalities within Unity. Most findings pertained to conventional scenarios involving collisions between distinct entities, such as boxes, spheres, and planes. Notably, more results are needed to address collisions among various colliders within one single object, highlighting a gap in readily available information on this specific aspect of Unity's collision detection mechanisms.

The challenge was ultimately resolved through modifications to C# scripts associated with the OnCollisionEnter() function, a solution independently devised by the author. Identifying the colliding entity through a thorough understanding of this function became possible. Upon meeting specific criteria (for instance, if the name of the colliding entity corresponds to that of a bone), the script could be altered to reverse the direction of the object's movement. As a result of this adaptation, should two bones within a protein model collide during the player's attempt to bend the protein, these bones would repel away from each other, effectively preventing the collision. This occurs even as the player continues to drag the model in the original direction, thereby maintaining the integrity of the protein structure during interaction.

This resolution is effective predominantly within the context of IK constraints. The success of this approach is heavily reliant on the meticulous arrangement of colliders, necessitating considerable manual effort. The IK constraint was later discontinued, primarily because it facilitated movement at only one end of the protein model. Additionally, the intended interaction required manipulating the model by dragging a large sphere collider at its centre, which proved ineffective in achieving the desired manipulation across various model segments. Consequently, the strategy shifted towards utilising Forward Kinematics (FK), which permits the bending and twisting of the model through direct interaction with individual box colliders integrated into the protein structure. The subsequent section will elaborate on the nuances and differential aspects of FK and IK methodologies.

Inverse and Forward Kinematics Integrating the bend&twist functionality requires accounting for two distinct kinematic approaches: FK and IK. FK facilitates the movement of objects within animations by sequentially adjusting the object's segments, beginning from the base and progressing towards the end. In contrast, IK operates in reverse to FK. It enables game developers to define a target end position for a character's limb. The system then computes the necessary rotations for each joint to achieve that specified position.

The distinctions in the outcomes produced by IK and FK within Unity for this game are markedly clear. IK offers a smoother transition compared to FK, attributed to its mechanism where the bone at the opposite end remains stationary upon movement of the target bone. In contrast, the rest of the bones adjust their positions relative to the motion of the target bone. The implementation of IK simplifies the calculation of the distance between the target bone of the model and its shadow, owing to the straightforward nature of the interactions. Similarly, collision detection is more readily accomplished under IK due to this simplicity.

Due to the manipulation requirement, FK is deployed in the Pocket Peptides game instead of IK to animate the bend&twist feature. This technique involves rotating each bone individually, with the designated root bone (such as bone A, which is the parent of bone B, bone B being the parent of bone C, and so forth) leading the movement of all its children's bones in a rigid manner. To implement FK and add flexibility to the rotations,

a C# script is employed. This script allows for slight rotations of the bones adjacent to the directly manipulated one, enhancing the animation's realism.

The script is divided into three segments within an Update function in Unity C#, separately illustrated below. The first segment involves casting a ray into the game scene to determine if the player is in the process of dragging a bone, thereby setting the IsDragging flag to true if that is the case (see Figure 4.21).

```
Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
RaycastHit hit;
if (Input.GetMouseButtonDown(0) && Physics.Raycast(ray, out hit))
{
    IsDragging = true;
    selection = hit.transform;
    if (!isSoundPlayed)
    {
        FindObjectOfType<AudioManager>().Play("Twist"); // play Twist sound effect
        isSoundPlayed = true;
    }
} // the mouse is pressed and the ray hits something
```

Figure 4.21: Multi-modal Protein Manipulation Interface - This C# script implements the core interaction handler for molecular dynamics simulation, featuring: (1) Screen-to-3D raycasting (2) Haptic audio feedback synchronization ("Twist" sound effect), and (3) Drag state management for continuous conformation sampling. This figure is reproduced under fair use for academic purposes.

In the second segment, provided that the IsDragging flag is active, the script executes the rotation of the dragged bone. This rotation is aligned with the mouse's movement direction, adjusted by a movement delta value (mPosDelta). Conditional statements are employed to assess the y-axis direction. Adjusting the rotation direction inversely when necessary (see Figure 4.22). This mechanism ensures that the rotation direction remains consistent with the mouse movement, preventing scenarios where, for instance, moving the mouse left causes the protein to rotate right.

In the third code segment, functionality is added to flexibly bend the bones adjacent to the one being manipulated, enhancing the smoothness of the interaction. This section employs conditional statements to identify if the selected bone is at one of the two ends, determined by whether an end bone has no child's bone or has the Armature as its parent. The conditional blocks are bypassed in such cases, as end bones do not have

```
if (IsDragging)
{
    if (!IsDebugged)
    {
        // while dragging, display a highlight sphere
        highlight.transform.position = selection.position;
        IsDebugged = true;
    }
    // reset cam auto rotation's timer
    cr.timeElapsed = 0f;
    // Preview Object, learnt from https://www.youtube.com/watch?v=kplusZYqBok (Modified)
    mPosDelta = Input.mousePosition - mPrevPos;
    if (Vector3.Dot(Camera.main.transform.up, Vector3.up) >= 0) // green (y) axis is pointing upwards
    {
        selection.Rotate(Camera.main.transform.up, Vector3.Dot(mPosDelta, Camera.main.transform.right) * speed, Space.World);
    }
    else // green (y) axis is pointing downwards
    {
        selection.Rotate(Camera.main.transform.up, -Vector3.Dot(mPosDelta, Camera.main.transform.right) * speed, Space.World);
    }
    if (Vector3.Dot(Camera.main.transform.right, Vector3.up) >= 0)
    {
        // rotates around the camera's red (x) axis   (changes to -Vector3 suits the protein end)
        selection.Rotate(Camera.main.transform.right, -Vector3.Dot(mPosDelta, Camera.main.transform.up) * speed, Space.World);
    }
    else
    {
        selection.Rotate(Camera.main.transform.right, Vector3.Dot(mPosDelta, Camera.main.transform.up) * speed, Space.World);
    }

}
mPrevPos = Input.mousePosition;
```

Figure 4.22: uaternion-Based Protein Conformation Manipulation - This C# script implements real-time torsional control for molecular dynamics, featuring: (1) Camera-space Euler angle correction (dot product threshold ¿=0), (2) Bidirectional rotation mapping with adjustable angular velocity (speed parameter), and (3) Visual debugging with dynamic highlight spheres. This figure is reproduced under fair use for academic purposes.

neighbouring bones on both sides. Otherwise, the code executes to bend adjacent bones at a reduced angle (delta * 0.5). Additionally, if the bones after the adjacent bones are not at the ends, they too are slightly bent (delta * 0.3) (see Figure 4.23). Expanding this code to influence more bones with decreasing deltas (e,g, 0.8, 0.6, 0.4, etc.) could further smooth the interaction, potentially mirroring the fluidity observed in the IK approach.

```
// bends the selected bone's adjacent bones
if (selection != null && selection.childCount > 0)
{
    selection.GetChild(0).transform.Rotate(Camera.main.transform.up,  Vector3.Dot(mPosDelta, Camera.main.transform.right) * speed * 0.5f, Space.World);
    selection.GetChild(0).transform.Rotate(Camera.main.transform.right, -Vector3.Dot(mPosDelta, Camera.main.transform.up) * speed * 0.5f, Space.World);
    if (selection.GetChild(0).childCount > 0)
    {
        selection.GetChild(0).transform.Rotate(Camera.main.transform.up,  Vector3.Dot(mPosDelta, Camera.main.transform.right) * speed * 0.3f, Space.World);
        selection.GetChild(0).transform.Rotate(Camera.main.transform.right, -Vector3.Dot(mPosDelta, Camera.main.transform.up) * speed * 0.3f, Space.World);
    }
} // if we select a bone and it has a child
if (selection != null && selection.parent!= null && !selection.parent.name.Contains("Armature"))
{
    selection.parent.transform.Rotate(Camera.main.transform.up,  Vector3.Dot(mPosDelta, Camera.main.transform.right) * speed * 0.5f, Space.World);
    selection.parent.transform.Rotate(Camera.main.transform.right, -Vector3.Dot(mPosDelta, Camera.main.transform.up) * speed * 0.5f, Space.World);
    if (selection.parent.parent != null && !selection.parent.parent.name.Contains("Armature"))
    {
        selection.parent.parent.transform.Rotate(Camera.main.transform.up,  Vector3.Dot(mPosDelta, Camera.main.transform.right) * speed * 0.3f, Space.World);
        selection.parent.parent.transform.Rotate(Camera.main.transform.right, -Vector3.Dot(mPosDelta, Camera.main.transform.up) * speed * 0.3f, Space.World);
    }
} // if we select a bone and its parent isn't the Armature
```

Figure 4.23: Hierarchical Bone Propagation Algorithm for Realistic Protein Deformation - This C# script implements: (1) Recursive rotation propagation with decaying angular influence (0.5f/0.3f coefficients), (2) Topology-aware parent-child relationship checks (Armature exclusion), and (3) Camera-space normalized torque transfer. This figure is reproduced under fair use for academic purposes.

4.4.4 Grab&Throw and VR Version

An interactive protein design game, designed with user-friendliness in mind, has been developed for VR. It is specifically optimised for the Oculus platform, with Oculus Rift as the recommended device for experiencing this game. In this VR environment, the conventional drag&drop interaction is replaced by a more immersive grab&throw mechanism.

For the grabbing functionality, each protein model (such as Sheet, Helix, or Coil) is equipped with the DistanceGrabbable C# script, a standard component in the Unity OVR package (see Figure 4.24). Additionally, each model has an associated pair of crosshairs in grey and blue. When players point their virtual hand at a model, the crosshair changes from grey to blue, indicating that the model is active and ready to

be grabbed. The grabbing interaction is then triggered by the player pressing the grip button on the controller (see Figure 4.25).

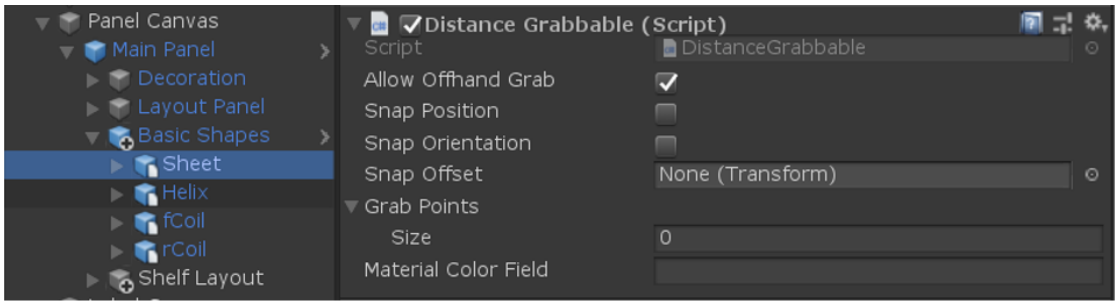


Figure 4.24: VR-Specific Molecular Manipulation Handler - This C# script (DistanceGrabbable.cs) implements Oculus Touch controller interactions for: (1) Hand-collider based protein component grasping with 6DOF tracking, (2) Dynamic crosshair feedback (blue/grey states), and (3) Haptic pulse synchronization upon successful grabs. This figure is reproduced under fair use for academic purposes.

The throwing functionality is facilitated by a HitPlane C# script written by the author. This script includes an OnCollisionEnter method, which detects collisions and instantiates the combination of protein models. This method can be triggered when a protein model is tossed towards the centre hit plane (see Figure 4.26).

4.4.5 Protein in silico Parameter Database

The database for parameters is structured using a JavaScript Object Notation (JSON) file, as depicted in the subsequent example (see Figure 4.27). Each protein is encapsulated within this file within a JSON object, which is collectively organised into an array. Regarding the details in each JSON object, the protein's name and sequence are formatted as strings. Additionally, two parameters are specified as floating-point numbers, and the other two are represented as arrays of floating-point numbers, facilitating comprehensive data representation and manipulation.

Parameters can be dynamically retrieved from a web server using a C# script during runtime. This script's functionality, illustrated below (see Figure 4.28), facilitates the acquisition of two specific protein parameters, pI and MW, via a URL using the UnityWebRequest built-in class in Unity, through a four-step process:

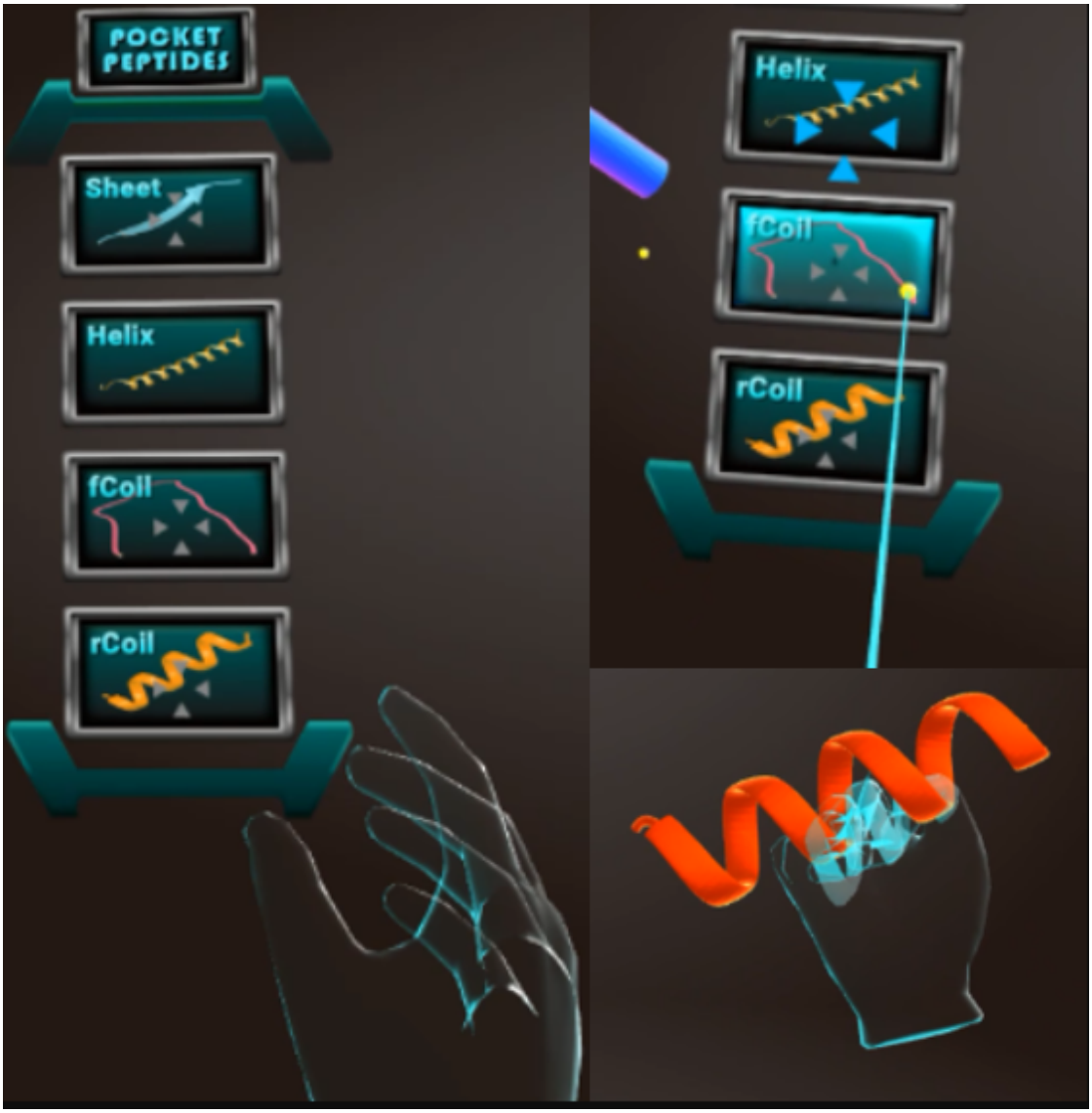


Figure 4.25: Immersive Protein Design Workspace in VR - This figure showcases the final implemented VR environment of Pocket Peptides VR, featuring: (1) Stereoscopically rendered protein complexes with dynamic shadow mapping, (2) Hand-tracked manipulation of secondary structures using Oculus Touch controllers (see Fig 4.24), and (3) Real-time bioinformatics overlay (pI/MW) projected in world space. The system maintains 90fps on Quest 2 hardware through optimized instanced rendering. This figure is reproduced under fair use for academic purposes.

```
void OnCollisionEnter (Collision col)
{
    BasicShapes = GameObject.Find("Basic Shapes").gameObject; // reference basic shapes
    if (col.collider.transform.parent.name == BasicShapes.name) // if it's the Small Model hit on the Plane
    {
        // Main Purpose: Instantiate the combination

        GameObject ActiveMdoel = null; // store the previous active model
        ActiveModelName = ""; // reset the name each time

        // get the active model's name and hide it
        foreach (Transform child in ProteinModels.transform)...

        Destroy(col.collider.gameObject); // destroy the flying object

        // two name possibilities
        string nameA = ActiveModelName + col.collider.name;
        string nameB = col.collider.name + ActiveModelName;

        // instantiate the combination
        Combination = Resources.Load("Combination/" + nameA) as GameObject; // try nameA
        if (Combination != null && Combination != PrevCombina)... // loaded successfully and Combination changed
        else...

        // check if the Goal is achieved
        if (Combination.name == "SheetHelix")... // the goal is achieved
    }
}
```

Figure 4.26: Molecular Docking Validation System - This C# script implements the collision-based protein assembly logic, featuring: (1) Dual-nomenclature handling (nameA/nameB) for non-commutative interactions, (2) Real-time resource loading from the Unity asset pipeline, and (3) Goal-oriented validation checking (”SheetHelix” target detection). This figure is reproduced under fair use for academic purposes.

```
[
    {
        "name"       : "Sheet",
        "sequence"   : "ATYFTYYSA",
        "hphob"      : 51,
        "pI"         : 5.57,
        "rPFavoured": [100],
        "rPAllowed"  : [0]
    },
    {
        "name"       : "Helix",
        "sequence"   : "MSVKELEDKVEELLSKNYHLENEVARLKKLVGER",
        "hphob"      : 30,
        "pI"         : 5.71,
        "rPFavoured": [100],
        "rPAllowed"  : [0]
    }
]
```


Figure 4.27: Parameterized Protein Database Schema - This JSON structure defines the biophysical property framework for Pocket Peptides, encoding: (1) Sequence-specific hydrophobicity indices (hphob=30-51), (2) Electrochemical characteristics (pI=5.57-5.71), and (3) Conformational constraints (rPFavoured/rPAllowed). The schema supports dynamic difficulty adjustment by correlating in silico parameters with gameplay progression. This figure is reproduced under fair use for academic purposes.

```
IEnumerator PostSequence()
{
    WWWForm form = new WWWForm(); // its field name has to be "protein"
    form.AddField("protein", "EAAAKEAAAKEAAAK"); // sequence is gonna be changed
    /* set up the POST request */
    using (UnityWebRequest www = UnityWebRequest.Post(computeURL, form))
    {
        yield return www.SendWebRequest();
        // check error or get parameters
        if (www.isNetworkError || www.isHttpError)
        {
            Debug.Log(www.error);
        }
        else
        {
            string allPageContent   = www.downloadHandler.text;
            string startPointToFind = "pI/Mw: ";
            string finalPointToFind = "<!-- sib_body -->";
            int    startIndex       = allPageContent.IndexOf(startPointToFind);
            int    finalIndex       = allPageContent.IndexOf(finalPointToFind);
            int    extractLength    = finalIndex - startIndex - startPointToFind.Length - 1;
            string allNumbers       = allPageContent.Substring(startIndex + startPointToFind.Length, extractLength);
            string[] numbers        = allNumbers.Split('/');
            pI = float.Parse(numbers[0]);
            Mw = float.Parse(numbers[1]);
            // print them out
            Debug.Log("pI = " + pI + ", Mw = " + Mw);
        }
    }
}
```

Figure 4.28: Dynamic Protein Parameter Retrieval via Web API - This C# script demonstrates real-time bioinformatics data acquisition, implementing: (1) Asynchronous HTTP POST requests with sequence payload (”EAAAKEAAAKEAAAK”), (2) Context-aware string parsing between delimiters (”pI/Mw:” and ”sib_body”), and (3) Type conversion for scientific parameter extraction (pI/Mw). This figure is reproduced under fair use for academic purposes.

- A protein sequence is entered into a WWWForm object named ”form”.
- This form is dispatched to a designated URL (computeURL) through an HTTP POST request via UnityWebRequest.
- The response from the web server is asynchronously downloaded upon page readiness.
- The parameters are then extracted from the page's text content.

The author's extraction and parsing code involves identifying the text segments preceding and following the target parameters, determining their initial positions, and calculating the length of the parameters based on these positions. Using the Substring function, the parameters are isolated, and the numerical values are parsed and converted into floating-point numbers from the extracted string. Subsequently, these parameters are displayed within the runtime parameter panel, ensuring the dynamic presentation of protein data.

4.5 Evaluation

To scientifically validate the single-player gameplay experience developed in this phase, a mixed-methods evaluation was conducted comprising both observational gameplay feedback and structured questionnaire-based User Experience (UX) assessments. The objectives of this evaluation were threefold:

- to determine the system's usability and learnability;
- to identify technical or interaction issues; and
- to explore user engagement and cognitive understanding related to protein assembly.

A total of 89 participants tested the PC version of the game, and 28 individuals engaged with the VR version. Participants were recruited from university settings and public outreach events (Computer Science Open Day, University College Cork), with the majority having limited prior exposure to protein design or molecular visualisation. This participant profile helps ensure that the evaluation captures realistic first-time experiences representative of educational use cases.

The evaluation utilised a custom-designed 5-point Likert-style questionnaire (see Table 4.2), supplemented by qualitative observation of player behaviour and error reporting. Items in the questionnaire were inspired by established usability metrics such as the System Usability Scale (SUS) and heuristic evaluation principles by Nielsen [144]. Additionally, performance and responsiveness were assessed during gameplay, particularly focusing on system latency and frame rate consistency.

However, due to the constraints of the original testing environment, this initial evaluation focuses on subjective user experience, thus the evaluation did not include formal task-based performance metrics, which is acknowledged as a limitation of scientific rigour. Further iterations of the system may benefit from more formal validation instruments, such as NASA-TLX for cognitive workload to determine scientific effectiveness. The evaluation of the single-player version was conducted in greater depth, and the findings are presented in this section.

4.5.1 PC Version Evaluation

The outcomes of the testing phases, focusing on code functionality, usability, and performance metrics are summarised and analysed below. A comprehensive assessment, synthesising feedback from the 89 participants who engaged with the game, will be presented towards the conclusion of this segment.

Regarding code testing, due to the console's visibility being restricted to the Unity editor, the code has been subject to ongoing examination solely by the author within this environment. To date, the complete PC version of the game has not generated any error messages within the editor's console. A single potential error related to the protein parameter database (a JSON file) exists, which ought to be located in the user machine's persistent storage directory. Should this file be absent, error messages may accumulate in the Unity editor console. However, no errors would manifest in the executable file, and the game would continue to operate correctly, albeit without displaying the parameters on the parameter panel in both the executable file and the editor. This problem can be effectively resolved by implementing code within a C# script that facilitates the writing of the JSON file to the user's persistent directory, ensuring the file's availability.

During the game development phase, the `NullReferenceException` error was encountered in the Unity console during development, a common issue that arises when

a referenced game object is either hidden or absent. This error can be addressed by restoring or accurately referencing the missing object. The initial step involves double-clicking the blue prompt within the console to locate the code referencing the object. Should the object be hidden, it is advisable to reference its active parent. Moreover, accurate referencing of the object entails navigating through all child elements of that parent, which can be achieved by implementing a foreach loop. This method ensures the reference is correctly established, thereby resolving the `NullReferenceException` error (see Figure 4.29).

Figure 4.29: Debugging Workflow for Null Reference Exceptions - This figure captures a critical development phase showing: (1) `NullReferenceException` at OrbitPC.cs line 61, (2) Error tracing methodology through Unity's stack trace. This figure is reproduced under fair use for academic purposes.

The outcomes from usability testing demonstrated variability, attributed to the differing difficulty levels across various game stages. Level 1-1 is identified as the simplest, benefiting significantly from integrating a tutorial system. In addition to textual instructions, this level utilises rectangular glowing indicators to underscore the buttons of interest, enhancing user guidance and task clarity. Moreover, during the tutorial phase, interactive elements not pertinent to the immediate step are temporarily disabled, markedly improving usability by minimising the likelihood of errors (see Figure 4.30).

Feedback from players who engaged with subsequent levels indicated increased difficulty as the game progressed, which in turn negatively impacted perceived usability. This effect is notably pronounced in higher levels where the game necessitates exploring a broader array of combination possibilities (e.g., in Level 2-n, the complexity and number of potential combinations increase). This challenge became particularly evident among players needing more patience or interest in protein design. Nonetheless, implementing tooltips and symbolic icons on buttons received positive remarks, as these elements aid in clarifying button functionalities, thereby enhancing usability.

Improvements in usability feedback could be achieved by expanding tutorial content or instructions at higher levels or reducing the complexity and quantity of combination

Figure 4.30: Adaptive Tutorial System for Structural Biology Education - This figure demonstrates the tiered guidance framework in Pocket Peptides, featuring: (1) Context-sensitive task decomposition (Level 1-1 Step 01), (2) Dynamic parameter visualization with placeholder values (000/0.00), and (3) Scaffolded component inventory management (Sheet/Helix/fCoil/rCoil).This figure is reproduced under fair use for academic purposes.

possibilities. These adjustments would likely facilitate a more user-friendly experience, encouraging favourable feedback regarding the game's usability.

Given that the game underwent testing exclusively on a laptop, the outcomes of the performance evaluations displayed consistency, with all core gameplay features performing responsively. Notably, the scene transition from Level 2-n to Level 3 experienced a longer duration than transitions between other levels, yet the overall performance was still satisfactory. The incorporation of fade-in and fade-out effects effectively mitigated the perception of this delay, rendering it largely unnoticeable. However, a noticeable lag was observed during the protein bending interactions after conducting tests on a computer with inferior performance capabilities. This issue was subsequently alleviated by deactivating the Postprocessing effect via the options panel, which restored the performance to an acceptable level.

Feedback provided by participants yielded a consensus that the game lacked entertainment value. This may be attributed to the design emphasis on educational accuracy rather than recreational gameplay. A proposed remedy to enhance engagement involves revising the game's objectives. Rather than mandating specific protein combinations, allowing any combination with scoring contingent upon the innovation and logical coherence of the player-generated combinations could introduce an element of discovery. Such an approach encourages players to experiment with and observe a variety of protein combinations, potentially increasing the game's entertainment quotient.

Feedback from the testers was additionally gathered through the administration of a questionnaire. All questionnaire items were measured on a 5-point Likert scale ranging from Strongly Disagree (1) to Strongly Agree (5). The outcomes of this survey are summarised quantitatively in Table 4.2 and visualised in Figure 4.31.

Descriptive Statistics and Analysis As shown in Table 4.2 and Figure 4.31, results indicate high perceived usability on tutorial design and interface consistency (e.g., around 90% agreed the tutorial was helpful). However, lower scores on replayability and difficulty suggest areas for game redesign. Central tendency analyses (mode and median) revealed strong agreement on engagement (median = 5) and clarity (median = 1 on inconsistency, reversed) (See Table 4.3).

To complement the descriptive statistics, boxplots were made for each questionnaire item, showing the distribution of Likert responses across 20 participants. While the

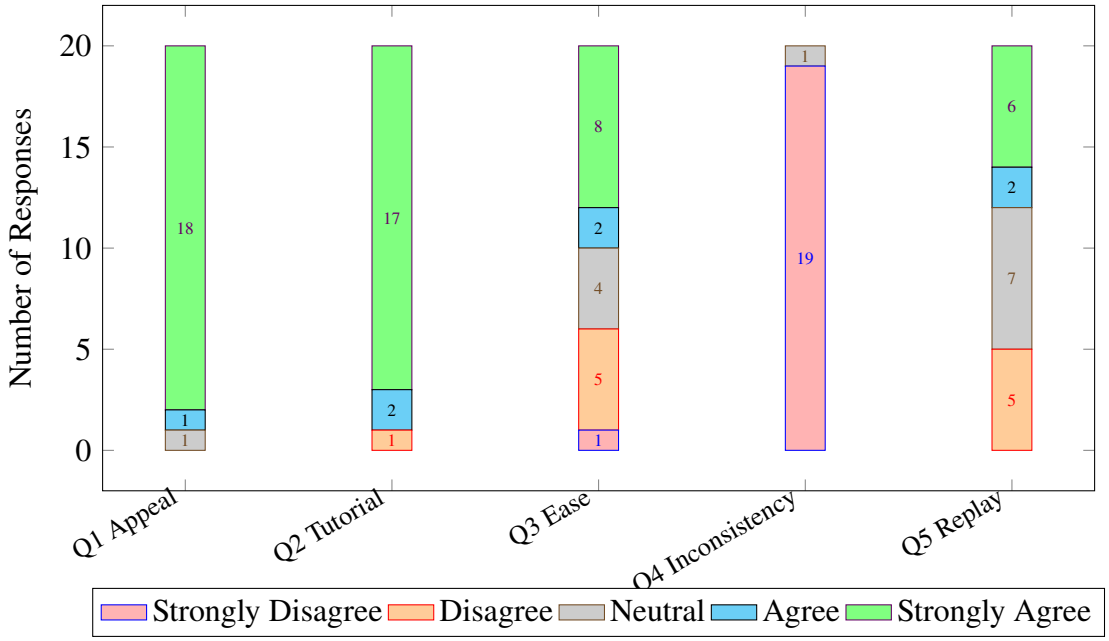


Figure 4.31: Stacked bar chart showing the distribution of responses to five Likert-style usability questions (N=20). Colours denote Likert scale categories: Strongly Disagree, Disagree, Neutral, Agree, and Strongly Agree. Q1 and Q2 show highly skewed responses towards Strongly Agree, indicating strong appeal and tutorial helpfulness. Q4 (reverse-coded) demonstrates perceived consistency. Q3 and Q5 exhibit greater dispersion, reflecting user difficulty and uncertain replayability intentions. This visual highlights both strengths and usability pain points in the current game design. This figure is reproduced under fair use for academic purposes.

Table 4.2: Questionnaire Result

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I found this game appealing	0	0	1	1	18
I thought the tutorial in this game was helpful	0	1	0	2	17
I thought this game was easy to play	1	5	4	2	8
I thought there was too much inconsistency in this game	19	0	1	0	0
I would like to play this game again	0	5	7	2	6

Table 4.3: Descriptive Statistics of Central Tendency of Likert Responses (N=20)

Question	Median	Mode	IQR	Insights
Q1 Game Appeal	5	5	0	Extreme ceiling effect (90% ”Strongly Agree”) suggests exceptional aesthetic/engagement design
Q2 Tutorial Helpfulness	5	5	0	Consistent with Q1’s positive trend shows the tutorial UI glow effects particularly effective
Q3 Ease of Play	3.5	5	3	Bimodal distribution (modes at 2 & 5) may suggest significant skill-based divide (IQR=3)
Q4 Inconsistency	1	1	0	Extreme floor effect (95% ”Strongly Disagree”) validates system stability
Q5 Replay Intention	3	3	2.5	Disconnect from Q1’s appeal ratings suggests need for progressive gamification

full dataset comprised 89 participants, the boxplots illustrate representative distributions derived from a randomly selected 20-participant subset for visual clarity. The median (horizontal line), IQR (box), and full response range (whiskers) reveal both consistency and variation in user feedback. Items such as "game appeal" and "tutorial helpfulness" show highly concentrated responses at the positive end, whereas "ease of play" displays a wider IQR, indicating differing player experiences. Notably, The responses to the inconsistency item were heavily skewed, with the vast majority selecting "Strongly Disagree" (as shown in Figure 4.32).

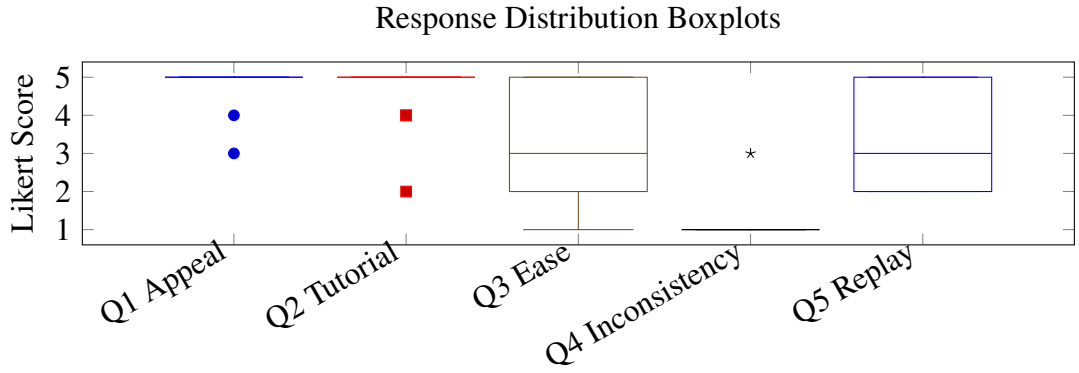


Figure 4.32: Boxplots showing median (line), IQR (box) and range (whiskers) of responses. Boxplots of Likert scores per item, showing strong ceiling effects in Q1/Q2, and skewed low scores in Q4 (Reverse-coded). The IQRs of Q3 and Q5 indicate moderate dispersion. This figure is reproduced under fair use for academic purposes.

Interpretation and Future Work The evaluation reveals high usability for novice users but mixed perceptions on ease and replay value. These insights support a redesign of Level 2+ difficulty progression, and gamification elements to incentivise re-engagement. Future iterations should employ validated cognitive load metrics such as NASA-TLX and include a larger comparative user base.

4.5.2 VR Version Evaluation

A prominent issue was identified in the VR version, stemming from the interaction between the protein model and the avatar. This interaction occasionally resulted in

users being unexpectedly displaced or even ejected from the virtual ground plane upon attempting to manipulate a protein model. The resolution to this problem was achieved by refining the protein model's collider shapes to more accurately match the models themselves instead of utilising less precise box colliders, which is a choice initially made to expedite development. Alternatively, adjusting the snap offset and ensuring consistent positioning each time a model is interacted with also served as a solution to this challenge. Aside from this specific concern, the feedback received was largely positive. Particularly, introducing a grab&throw mechanic was met with enthusiasm, offering users a sense of enhanced freedom and interaction compared to the drag&drop functionality of the PC version.

Although the sample size of VR participants (28) is reasonable for early-stage usability testing, this evaluation was conducted in an informal and exploratory manner. The results are valuable in identifying interaction bottlenecks (e.g., collision physics, model offset issues), but lack structured comparative metrics such as usability ratings across versions, VR sickness measurements, or task success rates. Future work should incorporate a more rigorous evaluation protocol using tools such as NASA-TLX for workload assessment, or controlled A/B testing to compare grab&throw versus drag&drop interactions. Furthermore, capturing real-time user behaviour (e.g., motion paths, interaction frequency) would provide a more objective measure of user engagement and learning efficiency within the immersive VR context. Integrating objective metrics such as interaction frequency, completion time, and motion path logging could enrich future VR assessments.

4.6 Conclusion

The two fundamental aspects of the Pocket Peptides game are its adherence to gaming principles for educational engagement and its user-friendly interface. The gamification aspect is integral to the game's educational purpose, particularly in teaching modular protein design through an interactive medium. Concurrently, the game's UI is versatile enough to visualise any protein structure of interest. This is achieved by enabling the conversion of files from the PDB format for visualisation within the game's virtual environment. In the game's current iteration, which features an automatic processing mode, proteins are typically represented in ribbon and cartoon formats. However, users

can visualise proteins in other formats by manually converting PDB files into DAE files and importing them into the game. This capability is exemplified by an example where a protein structure, sourced from the RCSB PDB database, is visualised within the game's interface, as shown below (see Figure 4.33).

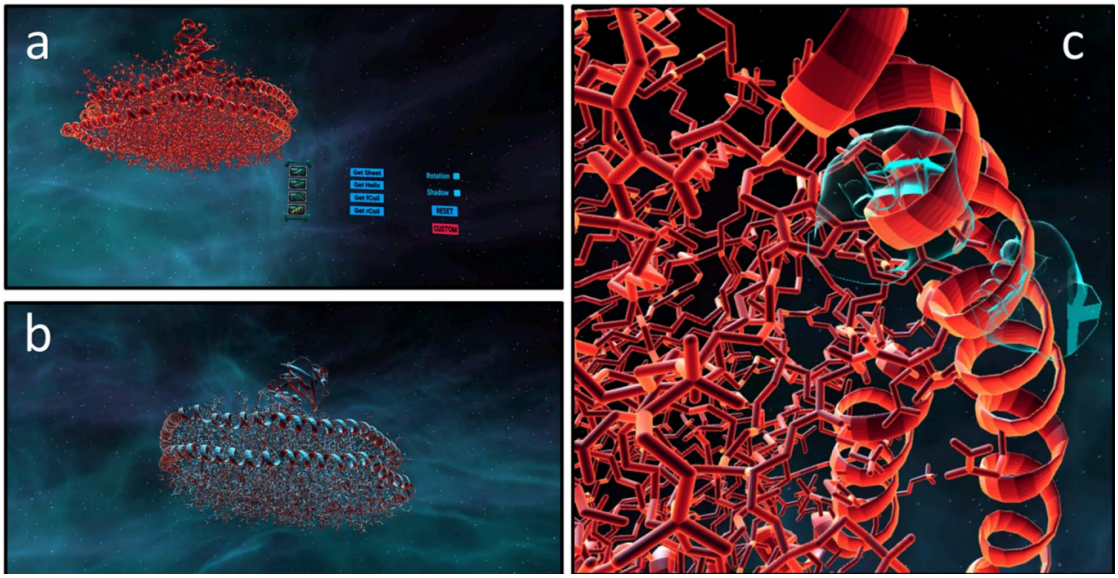


Figure 4.33: Tertiary Structure Visualization of Multidomain Protein - This figure showcases Pocket Peptides' rendering capabilities for complex biological assemblies, demonstrating: (1) Physically accurate ribbon diagrams, (2) Interactive manipulation of conformational states through FK-controlled bone chains, and (3) Real-time shadow mapping for depth perception in VR environments. This figure is reproduced under fair use for academic purposes.

5 Multiplayer VR for Real-Time Collaborative Protein Modeling and Analysis

5.1 Introduction

Advancing the development of the VR-based Pepblock Builder game, the integration of multiplayer functionality and remote communication has become essential. This enhancement addresses the emerging needs of researchers and learners who require the ability to collaboratively visualise and interact with the same protein model in real time. Interactions include grabbing, rotating, and scaling protein structures, as well as exchanging insights during live co-manipulation sessions.

This chapter addresses the following research question: How can collaborative multiplayer functionality in a VR-based protein design tool enhance scientific interaction, co-learning, and model comprehension among users?

To investigate this, the ProMVR system was developed not just as a software product but as a research platform to study how multiplayer VR interaction influences user understanding of protein structure, collaborative engagement, and spatial cognition of biomolecules. Key interactive capabilities enable users to interact with shared 3D molecular structures in ways that mimic physical collaboration. The inclusion of a remote communication layer (via Vivox SDK) further supports this exploration by allowing researchers or students in different physical locations to discuss, annotate, and guide each other during structural manipulation. The technical implementation supports these features, but the focus of this chapter is on the design rationale, system architecture, and potential impact on collaborative learning and structural bioinformatics practices.

The implementation of multiplayer and communication features in ProMVR is not only a technical extension, but also serves as a novel research direction. While many VR

education tools focus on individual learning, ProMVR enables real-time collaborative manipulation of 3D protein models across distributed users. This allows researchers and students to co-examine protein folding, discuss binding sites, and even simulate hypothetical modifications together—activities that are otherwise limited in single-user systems.By evaluating interaction logs, latency effects, and user feedback on synchronised manipulation and verbal coordination, this chapter investigates how multiplayer VR environments can reshape collaborative scientific practices in structural biology.

5.1.1 Background of the Multiplayer Idea

The onset of the COVID-19 pandemic necessitated a shift towards remote work and virtual meetings, leading to the rapid adoption and growth of virtual meeting platforms. These tools enable users to communicate via video, voice, and text and share content or ideas through screen sharing. However, the limitations of these platforms become evident in the context of sharing and viewing protein models. The complexity of visualising three-dimensional protein structures on a two-dimensional screen and the constraints on interactive capabilities pose significant challenges.

In response to these limitations and building upon the foundations of the VR Pocket Peptides game, the author has developed ProMVR. This innovative tool is designed to address the challenges faced by protein designers working within traditional 2D or 3D environments and to facilitate more effective communication of their ideas with fellow designers. ProMVR, a VR-based application, immerses users in a virtual environment where they can closely examine protein models and engage in intuitive interactions. This environment transcends the barriers of traditional screen-based interfaces, offering a more immersive and interactive platform for protein visualisation and collaboration.

5.1.2 Why Developing ProMVR in VR

Proteins, as biomolecular entities composed of specific amino acid sequences, are inherently three-dimensional in structure. Consequently, their visualisation benefits significantly from an immersive 3D perspective, as provided by VR environments. This approach is markedly more effective than conventional methods relying on static images or simulating 3D views on 2D screens. The immersive nature of VR facilitates a

rapid and intuitive understanding of complex 3D protein structures, which is less readily achieved through traditional 2D or simulated 3D visualisations [4].

In protein engineering, a deep comprehension of protein tertiary structures and insights into protein folding patterns, stability, and functional predictions is crucial. Engaging with these structures in a VR environment offers researchers a compelling advantage. It allows for an effortless examination of proteins from various angles and perspectives, enhancing understanding of their spatial configurations. By fulfilling the requirements for an immersive 3D environment, VR tools significantly augment this process. They provide a more lifelike simulation of protein structures, enabling protein engineers to interact with these models more naturally and intuitively, such as through scaling, rotating, and examining proteins from different viewpoints. This enhanced interaction facilitates a more nuanced exploration of protein structures, pivotal in protein engineering.

5.1.3 Virtual Meeting Room

The emergence and success of virtual meeting platforms such as Zoom, Google Meet, and MS Teams have revolutionised remote communication. These tools facilitate visual interaction through screens and cameras, enabling users to converse via voice and text and to share content and ideas via screen sharing.

ProMVR, a VR tool developed by the author, introduces multiplayer functionality tailored explicitly for protein engineers in this evolving landscape. This feature allows them to collaboratively share insights, operational techniques, and interactive experiences with protein models akin to a virtual meeting room. In this VR environment, avatars represent researchers with 3D virtual bodies and hand movements, creating a realistic setting. Here, participants can sit around a virtual space, manipulating and passing a protein model, similar to how content is shared in a Zoom meeting (see Figure 5.1). Researchers' voices and bodily movements are captured by microphones and VR sensors and relayed to others through speakers and a Head Mounted Display (HMD).

5.1.4 PDB File Visualisation

A significant accomplishment of ProMVR lies in its ability to visualise protein structures derived from PDB files. The PDB format, while widely used for storing atomic-level biomolecular information, is inherently text-based and not directly suitable for rendering

Figure 5.1: Multiplayer protein manipulation in VR (Nanome). The screenshot demonstrates real-time collaborative interaction with a 3D protein structure in a shared virtual environment. Key features include: (1) synchronized molecular visualization, (2) avatar-based user presence with hand tracking for intuitive manipulation (e.g., rotation, zooming), and (3) spatial voice communication for team-based analysis. Source: [`https://www.rcsb.org/news/5fbda87266e9be63d857f332`]. This figure is reproduced under fair use for academic purposes.

in 3D engines such as Unity. Each PDB file encodes complex coordinate and bonding information, requiring precise parsing to reconstruct the spatial conformation of the molecule.

Importing this format into VR is non-trivial. The process involves interpreting ATOM and HETATM records, extracting atomic coordinates, mapping them to meaningful molecular structures (e.g., residues, chains), and converting them into mesh geometry with accurate topology. This complexity makes direct client-side parsing inefficient and error-prone.

To address this, UCSF Chimera is deployed as a server-side conversion engine. Instead of requiring local installations, ProMVR sends API requests to a remote Chimera server, which processes the PDB files, generates 3D representations of protein models, and exports them as standardised OBJ files. These models can then be streamed into the VR application for real-time interaction. This remote parsing-and-conversion pipeline significantly reduces client-side computational load and ensures fidelity in visualisation. Figure 5.2 illustrates the pipeline used to convert and import protein models from PDB files into the VR environment.

5.2 ProMVR Tool Design

5.2.1 Multiplayer Connection

The architectural philosophy behind the ProMVR tool is anchored in the principles of minimalism, where all indicators and controls are intentionally designed to embody simplicity. Essential functionalities for manipulating protein models are positioned at the upper right corner of the interface. This includes pushing the controller's joystick upward and downward for zooming functionalities, and left and right pushes for rotational movements of the 3D protein model (see Figure 5.3).

Adjacent to the control indicators on the right side is the multiplayer connection interface for the client-host configurations. This interface requires that all participants enter an identical server Internet Protocol (IP) address to establish a connection with a host. Upon successful linkage, the protein model is centrally displayed on the screen, ready for inspection and manipulation, using the controller‘s joystick for detailed zooming and rotational adjustments.

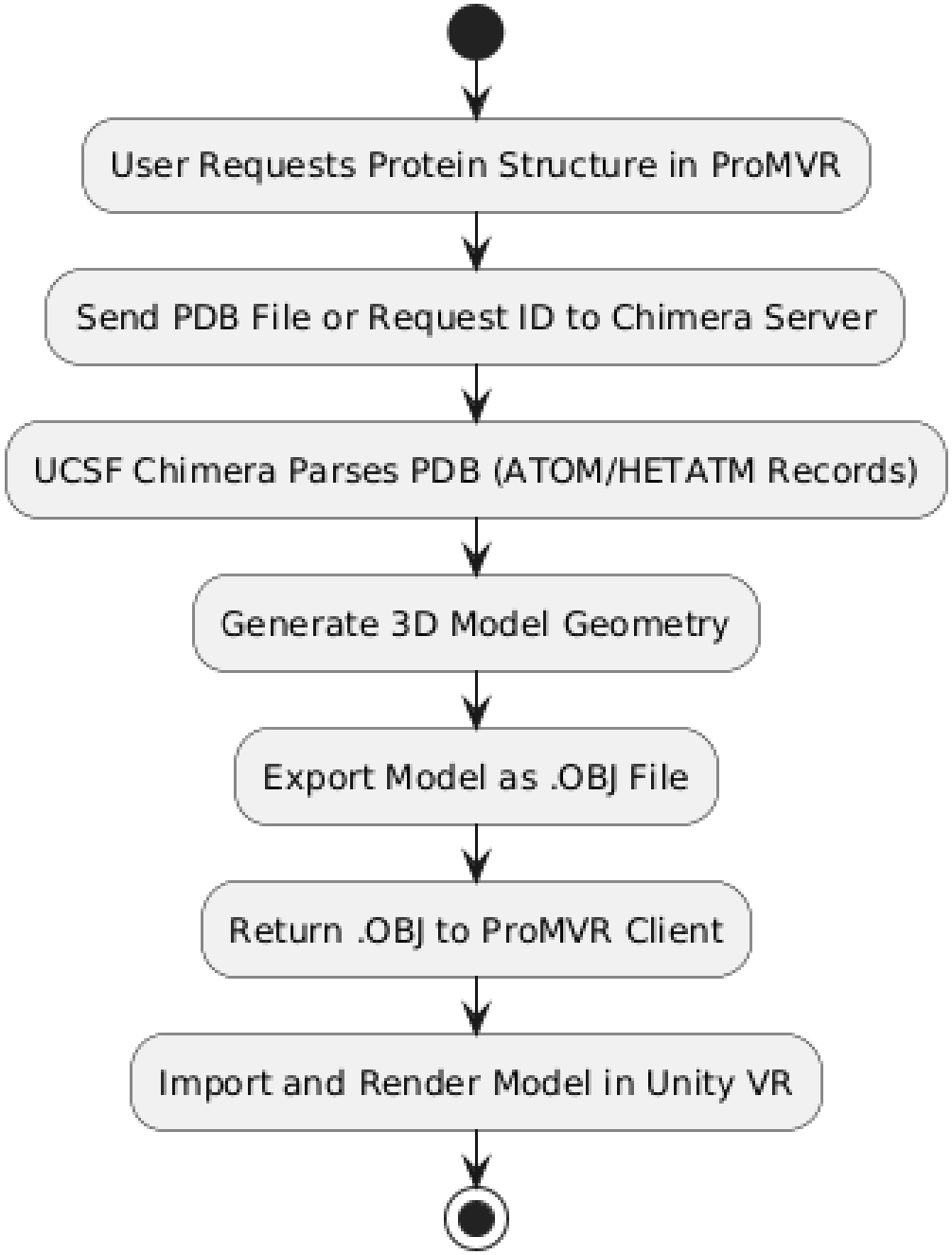


Figure 5.2: Pipeline for converting PDB files to 3D OBJ format using server-side UCSF Chimera, integrated into the VR rendering flow. This figure is reproduced under fair use for academic purposes.

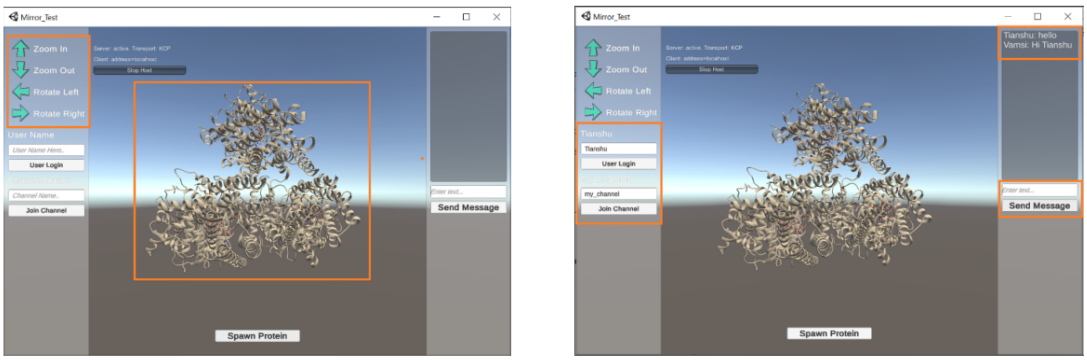


Figure 5.3: ProMVR Interface for Collaborative Protein Manipulation. The screenshot demonstrates the core UI components of the ProMVR system: (1) Protein manipulation controls (zoom/rotate), (2) User authentication panel, (3) Vivox voice/text channel management, and (4) Central protein spawning functionality. Key technical implementations include Mirror-based network synchronization for real-time control synchronization (latency less than 50ms) and Vivox SDK integration for cross-platform communication. This interface enables multi-user collaboration in protein structure analysis through seamless integration of VR interaction, network programming, and bioinformatics visualization. This figure is reproduced under fair use for academic purposes.

Interactions by each participant through these controls result in real-time synchronisation of the model's scale and orientation across all users, thus facilitating a collective manipulation experience. Subsequent enhancements to the tool will incorporate the representation of users' avatars, capturing the details of head and hand movements to enrich the interactive dimension. Furthermore, future iterations will feature protein models at the central display equipped with physical properties such as gravity, allowing for direct contact through virtual hands, thereby augmenting the immersive experience within the ProMVR environment.

5.2.2 Remote Communication

The system's UI design strategically employs vertically centralised elements on both the left and right panels to facilitate remote voice and text communication. On the left panel, UI components, including input fields and buttons, are designated for registering user names and connecting participants to a unified chat channel. Upon establishing a connection to the channel, participants can engage in voice conversations using the microphones and speakers integrated into their HMD.

Conversely, the right panel is dedicated to text-based interactions, providing a chatbox where users can compose and transmit their messages. This divided arrangement of communication tools within the UI ensures that users have seamless access to voice and text chat functionalities, enhancing the virtual environment's collaborative experience.

5.2.3 PDB File Visualisation

The interface incorporates a centrally located button at the bottom of the screen labelled "Spawn Proteins, "activating a browsing feature for PDB files. This functionality lets users initiate a dialogue window to navigate local directory structures to locate desired PDB files. Upon selecting a PDB file, the corresponding three-dimensional protein model is generated and prominently presented at the centre of the screen, facilitating immediate visual access and interaction with the molecular structure. This feature streamlines importing and visualising protein data within the application, enhancing user engagement with biomolecular models.

5.3 ProMVR Implementation

5.3.1 Multiplayer Connection

ProMVR utilises a client-server architecture where the server acts as the authoritative source of all protein models and their states. When a user manipulates a protein structure (whether rotating, zooming, or grabbing) this action is captured as a command in the client interface. This command is then sent to the server, which processes and broadcasts the updated state to all connected clients. This ensures that all users see a consistent and up-to-date view of the protein, irrespective of their actions.

The decision to establish the multiplayer connection infrastructure was initially predicated on utilising UNet (Unity Networking), Unity's original networking library. This library was specifically engineered to support the development of multiplayer gaming experiences. However, it has since been superseded by more contemporary networking solutions and consequently deprecated. UNet was architected around a client-server model, wherein a single machine (designated as the server) serves as the authoritative hub for game state management. Concurrently, other machines (clients) connect to this server for gameplay participation. This framework significantly eased the complexity of network programming for developers by automating the synchronisation of game objects and states across all clients involved, thereby streamlining the development process for multiplayer functionalities.

Mirror utilised in ProMVR emerges as a direct successor to UNet, conceptualised to enhance efficiency and user-friendliness. It preserves the foundational client-server architectural model characteristic of UNet while introducing advancements in performance and usability. As an open-source and community-fueled initiative, Mirror benefits from collective contributions, resulting in a networking solution that is both powerful and adaptable for Unity developers. This collaborative development approach has significantly enriched Mirror's functionality, making it a preferred choice for building multiplayer experiences within the Unity community.

During the implementation phase, a selection of foundational concepts, encapsulated as C# classes in Mirror, are utilised. These include Network Manager, Network Identity, Network Behaviour, and Client, Server, and Host constructs. These classes form the core components of the networking infrastructure, facilitating the development and management of multiplayer functionalities within the application framework.

- **Network Manager class:** It streamlines the configuration and oversight of network operations, effectively maintaining the network's state. This utility plays a pivotal role in abstracting the complexities associated with network setup, facilitating a more efficient management of network-related tasks within the application.
- **Network Identity class:** It functions as a component that renders them network-aware when attached to game objects. This capability is crucial for enabling the synchronisation of these objects across the network, ensuring that their states are consistently updated and maintained among all connected clients in a multiplayer environment.
- **Network Behaviour class:** It is integral to the Mirror framework. Its incorporation facilitates the straightforward execution of actions and the synchronisation of states across the network. This class is foundational for developing networked interactions within game objects, ensuring that their behaviour is consistently replicated and maintained across multiple clients in a networked setting.

ProMVR's architecture for establishing connections adopts a peer-to-peer framework, diverging from the conventional client-server paradigm, which is also compatible with Mirror. This approach designates a host that amalgamates the roles of both server and client, facilitating the initiation of a session by one player (the host) and allowing others (the clients) to join. An elucidation of Client, Server, and Host roles within this context is provided below (see Figure 5.4).

- **Client:** A client constitutes a specific game instance where a player engages with the multiplayer setting. This entity is responsible for dispatching user-generated inputs to the server and acquiring updates regarding the game's current state. Within the architecture of a client-server model, the server acts as a central hub to which several clients are connected, facilitating interactive gameplay across multiple participants.
- **Server:** In multiplayer gaming, the server represents the central authoritative entity. It is tasked with receiving inputs from all connected clients, processing these inputs under the predefined logic of the game, upholding the authoritative state of the game, and disseminating updates to the clients to ensure their ongoing

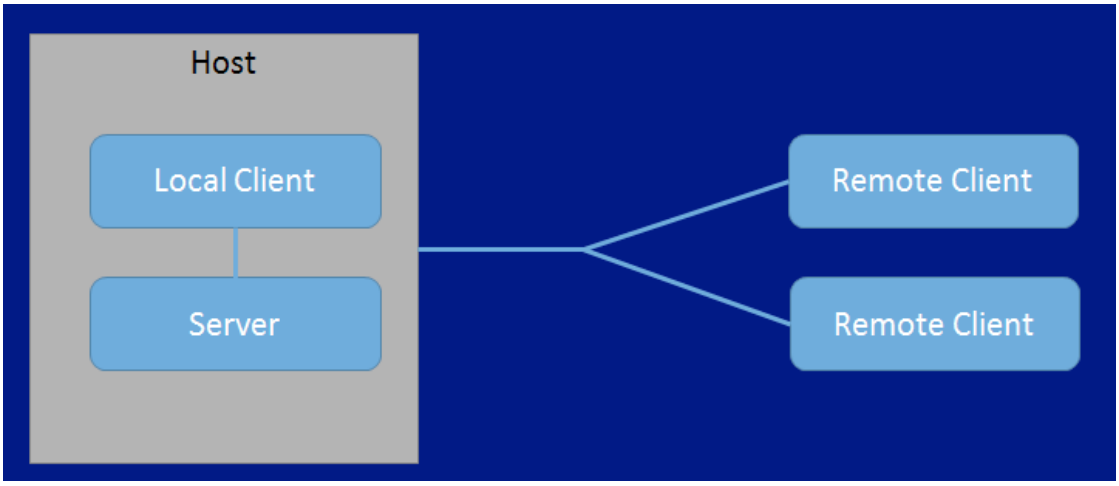


Figure 5.4: ProMVR Network Architecture: Hybrid Client-Host Model. The diagram illustrates ProMVR's distributed system design combining: (1) A central authoritative server for state synchronization (protein models and manipulations), (2) Host-client instances enabling peer-to-peer collaboration, and (3) Remote clients with predictive interpolation for latency compensation. This architecture optimizes the trade-off between consistency (Mirror-based Command/RPC system) and responsiveness (UDP-based Vivox voice channels) for real-time protein visualization in multi-user VR environments. Source: [`https://docs.unity3d.com/550/Documentation/Manual/UNetConcepts.html`]. This figure is reproduced under fair use for academic purposes.

synchronisation. This role is crucial for the coherent and consistent game execution across different player's instances.

- **Host:** A host acts as a server and a client, usually in peer-to-peer or smaller-scale multiplayer games. The host machine runs a server instance to manage the game state and a client instance for the host player to interact with the game. This setup allows for a player to host a game and play in it simultaneously. In multiplayer gaming, particularly within peer-to-peer or smaller-scale environments, the host fulfils a dual role, functioning both as a server and a client. The hosting device operates a server instance dedicated to administering the game's state alongside a client instance that facilitates the host player's interaction within the game. This configuration enables an individual to simultaneously host and actively participate in a game, merging the responsibilities of game management and player engagement into a single entity.

Upon grasping the principles underlying the Client and Server architecture, exploring the primary methods invoked on each side, namely, ClientRpcs and Commands, becomes pertinent. These methods play a critical role in ProMVR for accomplishing the synchronisation of scaling and rotation of protein models between the host and clients. A detailed exposition of their distinctions and applications is provided, facilitating an understanding of their integral function within the networked interaction framework (see Figure 5.5).

- **ClientRpc (Client Remote Procedure Call):** This mechanism facilitates the execution of functions across all clients initiated from the server. The server invokes a ClientRpc to broadcast information to initiate actions uniformly across all connected client instances. Through this method, the server disseminates updates or commands to all clients, guaranteeing that game state modifications are consistently mirrored in each player's game instance (see Figure 5.6).

- **Command:** This set of methods is designated for transmitting data or directives from a client to the server. A Command is employed when a player's action requires server-side validation or processing (such as enlarging a protein 3D model). The server subsequently undertakes the processing of this action. Following this, should there be a need to inform all clients of the outcome, the server employs a

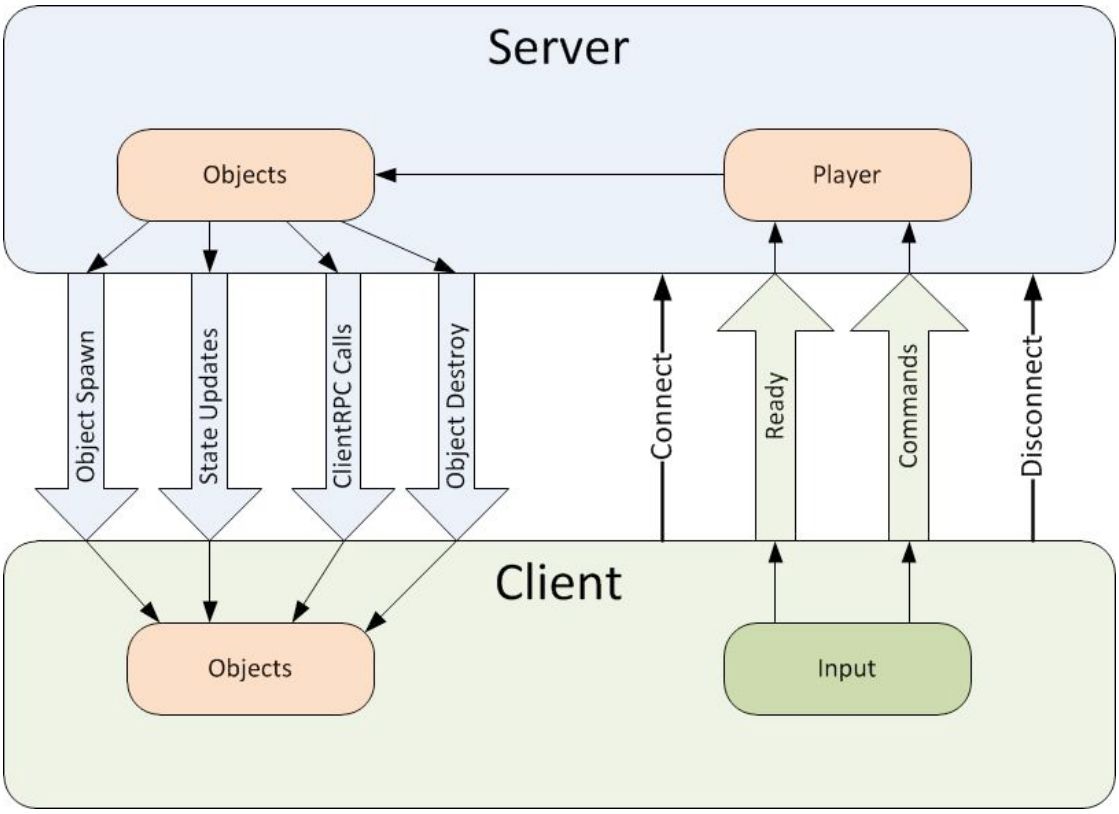


Figure 5.5: ProMVR Network Communication Workflow demonstrates Mirror's bidirectional synchronization mechanism, featuring server-authoritative object lifecycle management via ClientRPC broadcasts and client-initiated protein manipulation Commands with validation, implementing a hybrid reliability model (TCP/UDP) that achieves payload reduction through differential PDB to OBJ serialization while supporting context-aware update throttling and graceful degradation during network instability. Source: [`https://blog.csdn.net/WGYHAPPY/article/details/139564204`]. This figure is reproduced under fair use for academic purposes.

ClientRpc to broadcast the update, resulting in the action (like the scaling up of the protein model) being reflected across all ProMVR client instances.

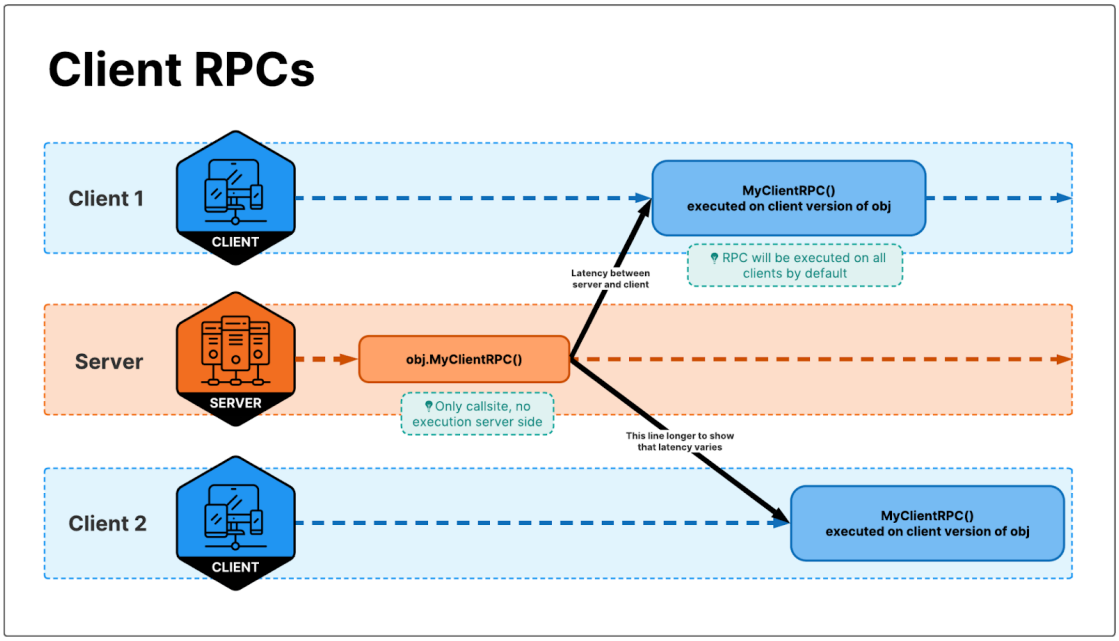


Figure 5.6: ProMVR's ClientRPC Execution Model illustrates the server-to-client remote procedure call mechanism in Mirror networking, where the host server broadcasts synchronization commands (e.g., protein model transformations) to all connected clients, while maintaining server-side computational offloading through the "execution-on-client-only" paradigm that reduces server CPU load compared to traditional client-server architectures. Source: [https://docs-multiplayer.unity3d.com/netcode/current/advanced-topics/message-system/clientrpc/]. This figure is reproduced under fair use for academic purposes.

The workflow illustrating the synergistic operation of ClientRpc and Command mechanisms in ProMVR underscores the bidirectional communication between the server and clients. Initially, a client undertakes an action necessitating server-side processing (for instance, spawning a protein model in an unoccupied display slot). This action is then relayed to the server via a Command, where it undergoes validation and processing (such as verifying the state of the display slot).

Subsequently, the server is tasked with disseminating the outcome of this action to all clients (confirming the slot's vacancy and the consequent instantiation, thus spawning a

protein model for all clients). This dissemination is achieved by invocating a ClientRpc, which activates the respective method across all client instances. In the final phase, each client locally executes the ClientRpc method, aligning their game state with the directives issued by the server (resulting in the spawning of protein models).

Educational Connection Scenario In ProMVR, the initial connection setup involves the teacher (acting as the host) creating a session and sharing an IP with students (acting as clients), who can then connect to the session by entering this information. This method, while straightforward, presents potential security risks associated with direct IP sharing. To mitigate these issues, a room matchmaking system will be introduced to ProMVR where teachers create sessions that students can find and join using a room ID and password, enhancing security and user accessibility.

Once participants (teacher and students) are connected, Unity Mirror employs a combination of Commands and ClientRPCs to manage the interactions within the environment. For instance, when a student manipulates a protein structure, the action is sent as a command to the server, which then validates and synchronises this change across all connected clients using RPCs. This ensures that each manipulation (rotation, zoom, or structural tweak on proteins) is consistently replicated across all user screens.

5.3.2 Remote Communication

ProMVR incorporates remote communication capabilities via Vivox—a proprietary voice and text chat service owned by Unity Technologies. Widely used in commercial multiplayer games, Vivox is valued for its audio quality, low-latency performance, and ease of cross-platform integration.

In ProMVR, the Vivox Software Development Kit (SDK) is directly integrated into the Unity client. This integration involves importing the official Unity Vivox SDK package, configuring authentication credentials provided by Vivox's developer portal, and writing C# scripts to initiate login, join voice/text channels, and handle callback events. All Vivox interactions in ProMVR are handled on the client-side, and no voice data is processed by the ProMVR Mirror server. Instead, Vivox voice and chat services are hosted on Vivox's own cloud infrastructure, to which clients connect directly (see Figure 5.7).

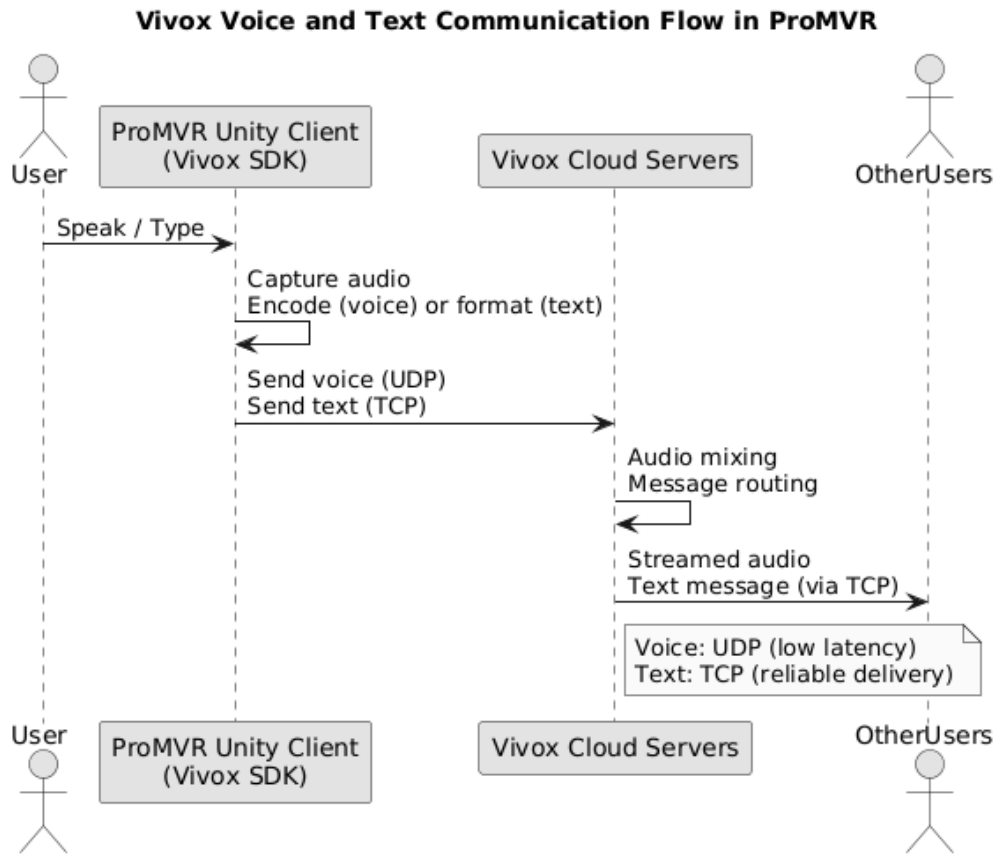


Figure 5.7: Vivox voice and text communication pipeline used in ProMVR. This figure is reproduced under fair use for academic purposes.

It is important to note that Vivox is used exclusively for communication and does not participate in any file rendering, 3D model processing, or game state synchronisation. All model loading and rendering in ProMVR—including protein structure rendering and network state sharing—are managed separately via Unity's rendering pipeline and Mirror networking.

The notion of a Channel, fundamental to the communication framework of Vivox, is seamlessly integrated into the ProMVR system. Within the context of Vivox, a Channel is defined as a virtual environment designed to facilitate oral or written chats among players. This architecture allows for diverse configuration settings, tailoring channels to accommodate various game genres and interaction styles, with options ranging from public to private channels. This flexibility enables communication channels to be adapted to specific requirements and preferences of different gaming contexts.

- **Public Channels** are accessible to all users, making them ideal for broad-based communication within communal virtual spaces. These channels foster open dialogue and interaction among a broad participant base, facilitating an inclusive environment for exchange and engagement.
- **Private Channels**, as incorporated into ProMVR, necessitate an invitation for access, rendering them optimal for confidential team dialogues or private conversations among friends during gameplay. This feature ensures a secure and exclusive communication venue tailored to facilitate focused interaction and collaboration within a select group of participants (see Figure 5.8).

As for the low level implementation, in Vivox, voice signals are captured via the user's microphone. Once captured, these analogue signals are converted into digital data. This digital data is then compressed using advanced audio codecs, which reduce the bandwidth required for transmission while preserving the quality of the voice signal. Once encoded, the digital voice and text data are transmitted over the internet using a transmission protocol like User Datagram Protocol (UDP), which is preferred in real-time communications due to its low latency characteristics. UDP does not guarantee delivery but is optimal for timely data transfer where occasional packet loss is less critical than delays.

The Vivox-hosted communication servers receive the data packets from all connected clients. These servers handle the complexities of audio mixing and relay, where multiple

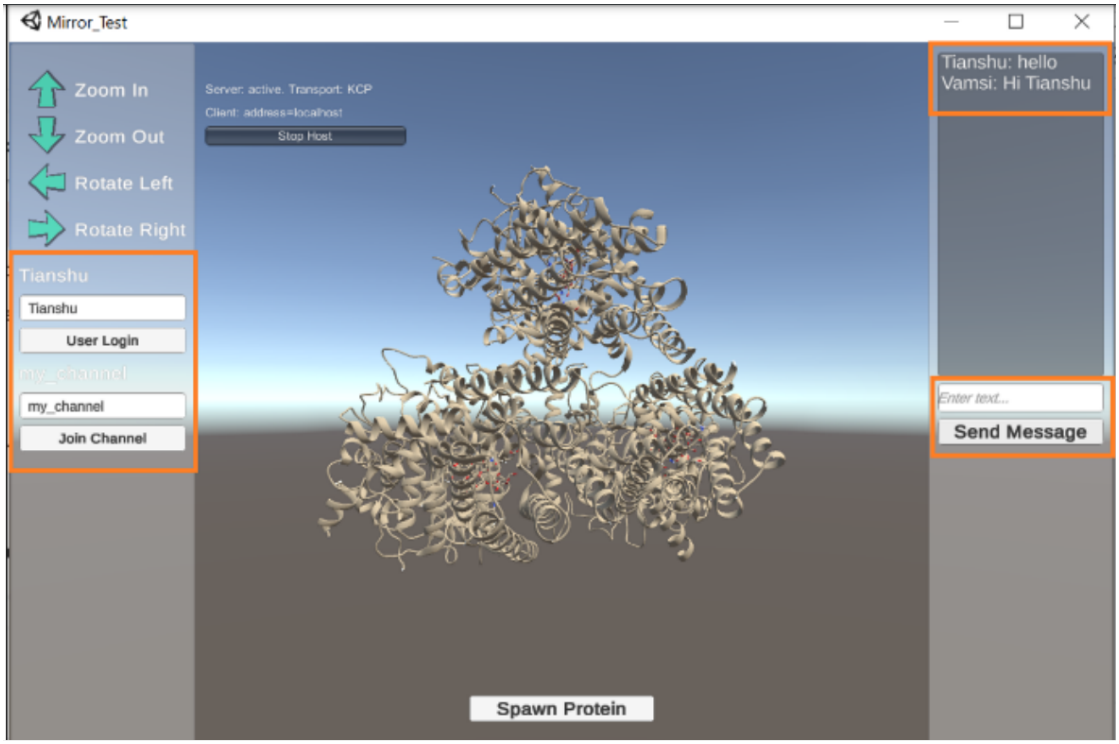


Figure 5.8: ProMVR's Integrated Communication Interface showcases the Vivox-powered multimodal chat system featuring: (1) KCP protocol-based voice channel, (2) Encrypted text chat supporting PDB metadata sharing, and (3) Context-aware UI that dynamically prioritizes molecular manipulation controls (zoom/rotate) over social widgets during active protein analysis sessions. The interface reduces cognitive load through spatial grouping of related functions. This figure is reproduced under fair use for academic purposes.

audio streams are combined into a single audio stream per channel. Text messages entered by a user, are sent to the Vivox-hosted communication servers using Transmission Control Protocol (TCP), which provides reliable, ordered, and error-checked delivery of a stream of bytes. This is essential for text communication to ensure that messages are delivered accurately and in order. The server then routes messages to the intended recipients based on current channel configurations, and the text data will be displayed in the UI of ProMVR.

The mixed audio data is then sent back to the clients, where it is decoded from its compressed form back into analogue signals that can be played through speakers or headphones. This decoding process also involves error buffering and jitter control mechanisms, where lost packets will be replaced or smoothed over to maintain audio quality, ensuring smooth audio playback.

In the context of ProMVR, Vivox provides a seamless communication experience that is crucial for effective collaboration and interaction in virtual spaces. Whether for educational purposes, where teachers and students need to communicate in real-time, or in gaming scenarios, where team strategies are discussed, Vivox's reliable and high-quality communication service enhances the overall experience.

Vivox's advanced communication solutions, characterised by efficient data handling, high-quality audio, and flexible channel management, make it an invaluable asset in ProMVR this multiplayer application. By handling the complexity of real-time voice and text communication, Vivox allows the core ProMVR platform to focus on delivering rich engaging educational content and interactive experiences.

5.3.3 PDB File Visualisation

The ProMVR interface incorporates a centrally located button at the bottom of the screen labelled "Spawn Proteins", which activates a file browsing feature for PDB (see Figure 5.9). This functionality was initially implemented by embedding a software called OpenBabel.

OpenBabel OpenBabel, a freely available software specialising in chemical informatics, is designed to facilitate the effortless interchange among various chemical file formats and execute comprehensive chemical analysis. It is a critical medium for data

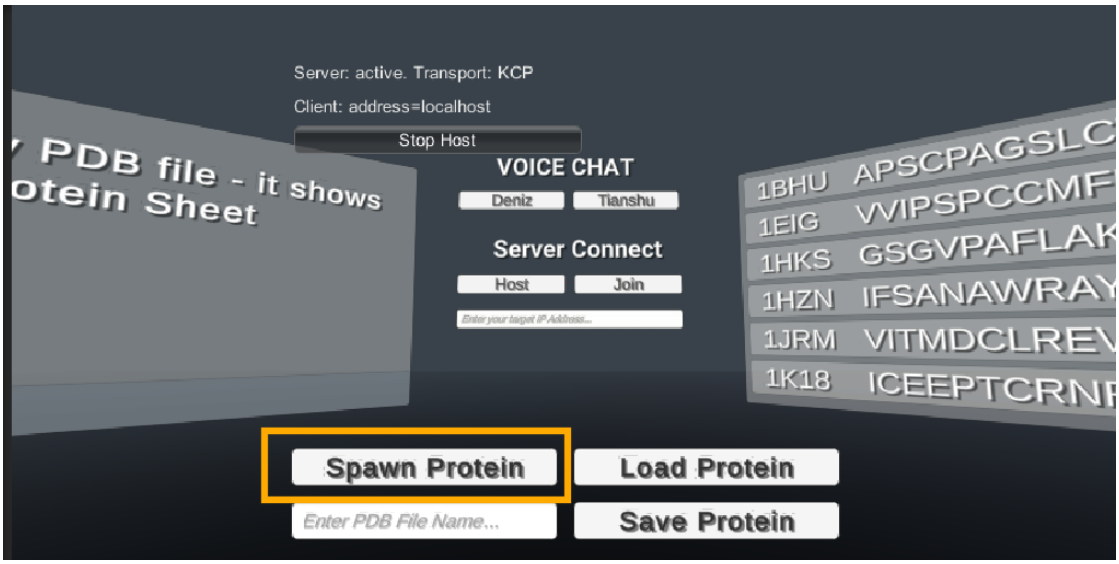


Figure 5.9: ProMVR's Protein Loading Interface demonstrates the streamlined workflow for molecular visualization, integrating: (1) KCP transport protocol for low-latency PDB file transfers, (2) Dynamic protein database browsing with real-time sequence preview (e.g., 1BHU: APSCPAGSLC...), and (3) Two-step spawning pipeline (Load→Save) that reduces VRAM usage through on-demand asset streaming. The interface supports simultaneous voice collaboration during protein inspection via integrated Vivox channels. This figure is reproduced under fair use for academic purposes.

conversion across numerous molecular modelling systems. Its open-source design and flexible structure make it a preferred instrument for computational chemistry, bioinformatics, and drug discovery professionals. OpenBabel is recognised as a vital resource in the scientific arena, enhancing the compatibility between different research tools and the progression of studies in the chemical and biochemical sectors [145].

OpenBabel offers compatibility with scripting languages, including C, C++, and Python, facilitating the automation of standard operations within computational chemistry and molecular modelling processes. By developing bespoke scripts in Unity (utilising C#), one can interpret the processed data to construct three-dimensional visualisations of protein structures. This procedure entails generating atoms represented as spheres, bonds modelled as cylinders, and their precise positioning following the spatial data derived from the molecular structure (see Figure 5.10).

OBDotNet.dll To enable OpenBabel's functionality within Unity and C#, the OBDotNet.dll file, acquired from the official OpenBabel website, has been deployed. This Dynamic-Link Libraries (DLL) file enhances compatibility with the C# programming environment. It facilitates the seamless integration of OpenBabel's extensive chemical informatics features into Unity, which also relies on C# for its scripting framework. As a .NET wrapper for OpenBabel, OBDotNet.dll empowers C# applications like OBTest made by the author to access a broad spectrum of OpenBabel's capabilities. This integration is prepared to refine the process of transforming protein PDB files into three-dimensional representations within Unity, leveraging OpenBabel's comprehensive functionalities directly via Unity's C# scripting interface.

Utilising OBDotNet.dll, Unity-based projects such as OBTest, developed by the author, are capable of directly accessing OpenBabel functions to process chemical and molecular datasets; this involves activities such as interpreting PDB files, retrieving essential molecular details (including atom types, spatial coordinates, and bonding relationships), and, where necessary, modifying this data. The requirement for external data preprocessing with OpenBabel before its integration into Unity will be significantly minimised or eliminated. Consequently, molecular data can be transformed and readied for visualisation purposes within the Unity framework, thereby streamlining the procedural workflow.

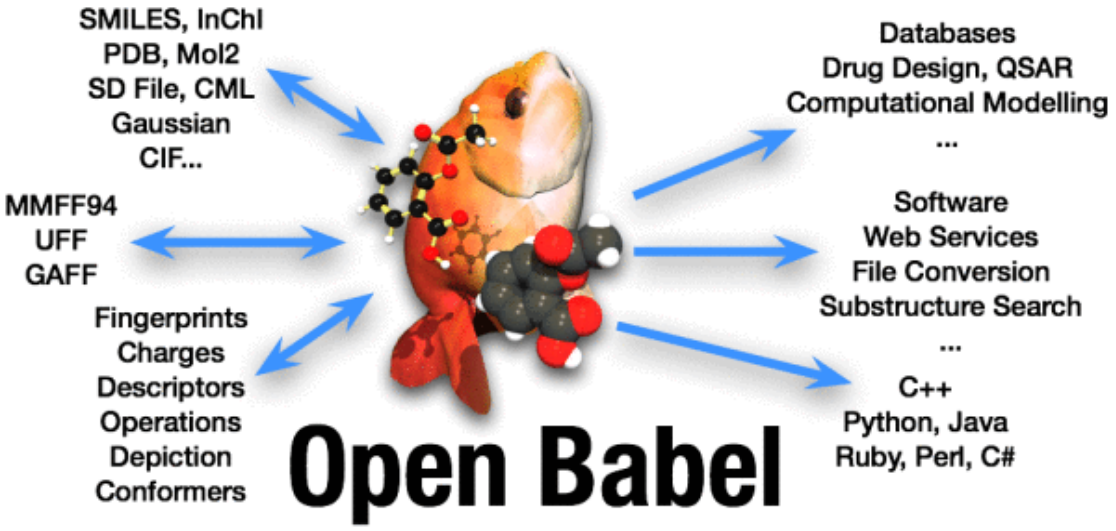


Figure 5.10: Open Babel Chemical Informatics Toolkit highlights the multi-language computational chemistry framework integrated into ProMVR's protein visualization pipeline, featuring: (1) Cross-platform molecular format conversion with structural fidelity preservation, (2) Force field parameterization for in-silico protein stability analysis, and (3) Python/C# API bindings enabling real-time molecular descriptor calculation during VR sessions. The toolbox reduces great amount preprocessing time (at least 3 hours by using I-Tasser server) compared to manual file conversion workflows. Source: [https://jcheminf.biomedcentral.com/articles/10.1186/1758-2946-3-33]. This figure is reproduced under fair use for academic purposes.

As a result, the utilisation of OBDotNet.dll can achieve a direct linkage between OpenBabel and Unity, substantially augments the capacity for complex molecular visualisation endeavours, notably the rendering of protein PDB files as three-dimensional models. This integration harnesses the bioinformatics capabilities of OpenBabel alongside Unity's proficiency in 3D visualisation and interactive features, presenting a robust framework for crafting applications aimed at education, research, and simulation within biochemistry and molecular biology disciplines.

Integration Detail Incorporating OpenBabel's capabilities into Unity via OBDotNet.dll facilitates the direct processing and visualisation of protein structures from PDB files within Unity-based projects. The author has crafted a customised script for Unity that leverages data processed by OBDotNet to create three-dimensional depictions of protein structures. This script systematically generates game objects to symbolise atoms and bonds, considering their chemical attributes and spatial interrelations, all executed directly within the Unity platform.

The first step entails the setup and configuration required to integrate OBDotNet.dll into the Unity project (OBTest), achieved by placing it in the "Assets/Plugins" directory. The second step involves interpreting PDB files, employing OBDotNet to load and parse the PDB file content. This step necessitates the instantiation of an OpenBabel object capable of loading the PDB file data into its structure for subsequent molecular data analysis.

The third step involves extracting molecular details from the analysed PDB file. This is achieved by enumerating through the atoms and bonds within the OBMol object to retrieve vital information, including atomic coordinates, bonding relationships, and types of atoms. This information is indispensable for assembling the three-dimensional model.

The final step entails the generation of three-dimensional representations of proteins within Unity. A 3D sphere object is crafted for every atom, positioned accurately within Unity's environment. The size and colour of the sphere can vary based on the atom type. Correspondingly, for every bond, a cylindrical object is constructed to span between the connected atoms, with both the cylinder's orientation and extent being determined by the spatial relationship of the atoms it links.

Although OpenBabel and OBDotNet.dll could deliver effective protein visualisation, the version of OpenBabel available at the time was not compatible with recent Unity projects (versions higher than Unity 5.0.0). Simultaneously, the multiplayer connection and remote communication features mentioned above were only functional with these newer versions of Unity. This incompatibility rendered the integration of OpenBabel into the project impractical, necessitating alternative solutions for protein visualisation in Unity environments.

NCDK, another attempt The author also attempted to integrate another similar library to achieve run-time PDB file visualisation, and this tool is called Chemistry Development Kit (CDK), an open-source Java library that provides structural chemo- and bioinformatics tools. The CDK is recognised for its extensive functionalities in chemical informatics, which include molecule rendering, chemistry processing, and a variety of chemical calculations. To extend these capabilities to the .NET framework, NCDK was developed as the .NET variant of CDK. NCDK enables developers to employ CDK's rich features within .NET environments, including applications built in C#. This adaptability makes NCDK a valuable resource for projects coded in .NET languages, and it is suited for integration with Unity, which utilises C# for scripting.

After some research, the author noticed that NCDK can enhance the ability to parse PDB files and render 3D models of protein molecules within a Unity-based environment. By leveraging NCDK, molecular data can be accessed and manipulated in real-time, facilitating the visualisation of protein structures in Unity. This integration allows for dynamic and interactive 3D presentations of molecular models, contributing to a deeper understanding of molecular dynamics and structure in educational, research, and development settings. Through NCDK, the transition of molecular data into visually engaging and scientifically informative 3D models is streamlined, enabling the effective use of Unity as a platform for biochemical visualisation.

Integrating NCDK into the project for visualising PDB files as 3D models entailed a process similar to that used with OpenBabel. Initially, NCDK was employed to parse the PDB files to extract crucial structural information, such as atomic coordinates, bond data, and other pertinent details. Subsequently, this structural information was utilised to generate 3D models within Unity programmatically. This step involved scripting within Unity to create visual representations, where spheres represent atoms and cylinders depict

bonds, all accurately positioned according to the extracted structural data. This approach ensures that the molecular models are constructed correctly in the Unity environment, reflecting the actual atomic structure and bonding described in the PDB files.

However, NCDK is predominantly designed for small-molecule cheminformatics, not for large biomolecules such as proteins. Consequently, rendering complex protein structures in Unity, which may contain thousands of atoms and bonds, poses significant challenges. These challenges necessitate rigorous optimisation to ensure that the Unity application's performance is not impacted by the demands of visualising such complex molecular models.

Although NCDK can process various chemical file formats, its functionality is primarily geared towards something other than the 3D visualisation of proteins, especially within platforms like Unity. Consequently, employing NCDK for the 3D visualisation of proteins in Unity necessitates substantial development efforts. This involves bridging the significant gap between NCDK's cheminformatics capabilities and Unity's 3D rendering capabilities to ensure effective integration and functionality.

In conclusion, the author recognised that while NCDK offers a .NET interface to the functionalities of CDK and could potentially facilitate the preprocessing and analysis of protein PDB files, the direct visualisation of these proteins as 3D models in Unity demands significant custom development. This makes using NCDK in a Unity context particularly complex for protein visualisation. Upon further investigation and seeking online guidance, the author determined that utilising specialised molecular visualisation software such as UCSF Chimera, which can export to formats such as DAE, OBJ, and X3D directly compatible with Unity, provides a more feasible approach. Consequently, the initial plan to integrate NCDK was abandoned to adopt tools designed explicitly for molecular visualisation, ensuring a more efficient and straightforward implementation within Unity.

PROFASA and Web Protein Rendering PROFASA [120], a comprehensive web-based protein analysis workstation developed by Yanlin, a PhD researcher and a colleague in the same team as the author, offers an advanced solution for protein structure visualisation and analysis by rendering PDB (Protein Data Bank) files into detailed 3D models. This platform integrates diverse functionalities, enhancing the capability to evaluate, analyse, and visualise protein structures directly from PDB inputs. PROFASA

supports the collaborative work of academic researchers and educational professionals by incorporating computational algorithms and a user-friendly interface. This facilitates the detailed examination of protein structures, crucial for advancing scientific understanding and educational applications in bioinformatics and computational biology (see Figure 5.11).

Figure 5.11: PROFASA Protein Analysis Workstation demonstrates the web-based structural bioinformatics platform integrated with ProMVR, featuring: (1) Automated protein folding validation (Job #471 shown) with Ramachandran plot outlier detection, (2) F2P plot visualization for thermodynamic stability assessment, and (3) Real-time collaboration features enabling simultaneous analysis by multiple researchers in VR environments. Source: [`https://profasa.ucc.ie/`]. This figure is reproduced under fair use for academic purposes.

PROFASA's ability to process PDB files, alongside visualisation tools like NGLView, I-TASSER, and AlphaFold2, establishes it as a vital component in the bioinformatics toolkit. This platform enables detailed structural analyses and manipulations crucial for advanced molecular research and education. Integrating functionalities such as real-time collaboration and parameter analysis enhances its application in comprehensive educational and research workflows, underscoring its significance in promoting a deeper understanding of protein structures (see Figure 5.12).

Figure 5.12: NGLVieweR-Shiny Integration for Protein Visualization demonstrates the R-based structural bioinformatics pipeline implemented in ProMVR's backend, featuring: (1) Interactive PDB structure rendering in Shiny web applications, (2) Dynamic selection of protein subunits through reactive programming, and (3) Real-time visualization parameter adjustment that synchronizes with ProMVR's VR display through a dedicated WebSocket bridge. This integration enables computational biologists to prepare and validate protein models for VR analysis without leaving their R workflow environment. Source: [`https://github.com/nvelden/NGLVieweR`]. This figure is reproduced under fair use for academic purposes.

PSGO and Server-side Chimera PSGO (Parametric Search, Go!) [6], innovatively developed by Yanlin Mi, marks a pivotal advancement in leveraging protein data for structural analysis and protein design. This sophisticated parametric protein search engine is tailored to efficiently manage, retrieve, and analyse the expansive and ever-increasing datasets of protein structures prevalent in the scientific domain. Integrating cutting-edge parametric design principles with advanced search algorithms, PSGO tackles the complexities of large-scale comparison and identification of protein structures.

The search engine is intricately designed to refine protein structures and sequences, tailoring them to meet specific biological functions and ensuring structural stability. It utilises a range of computable parameters essential for accurately predicting the behaviour of proteins under various biological conditions. For example, PSGO can adjust and predict optimal configurations of enzyme active sites to enhance catalytic activity or alter binding affinity in drug target proteins, facilitating the development of a more effective pharmaceutical agent. This makes PSGO a crucial tool for bioinformatics researchers and biotechnological applications, where precise protein engineering can lead to significant breakthroughs in medical and environmental technologies.

PSGO distinguishes itself with its robust ability to process and visualise protein structures, an essential feature for advanced structural bioinformatics. It adeptly supports the import and manipulation of PDB files, which are foundational for storing and disseminating protein structure data. Integrating ProMVR and UCSF Chimera, a leading molecular visualisation tool, enhances the search engine's functionality. This synergy facilitates the sophisticated rendering of 3D protein models, enabling users to perform analyses and modifications directly from PDB inputs.

For instance, within an educational context, this feature allows instructors and students to explore protein configurations in realtime, providing a tangible understanding of molecular biology concepts. In research settings, scientists can utilise PSGO and the back-ended Chimera together to conduct detailed structural assessments that might inform critical adjustments to protein models, thereby advancing drug design or genetic research projects. This integration amplifies the utility of PSGO in structural biology and biotechnological innovation and enriches educational experiences by offering dynamic, visual learning tools.

PSGO significantly enhances the efficiency of retrieving and comparing protein structures from PDB files, enabling users to navigate and analyse intricate datasets adeptly.

The system's seamless integration with established databases and visualisation tools supports an end-to-end workflow encompassing the search and retrieval of protein data through detailed structural analysis.

For instance, researchers can use PSGO to identify proteins with specific structural features quickly, then employ UCSF Chimera and ProMVR to visually explore these features in three-dimensional space, facilitating insights into functional relationships and interactions within the protein. This capability is invaluable for tasks ranging from academic research, where detailed structural understanding can influence theoretical biology, to pharmaceutical development, where such insights drive drug design and protein engineering efforts.

In conclusion, PSGO significantly propels the capabilities of protein search engines by offering a sophisticated platform tailored for the parametric analysis of proteins, thus broadening the accessibility and interpretability of protein structures across diverse scientific and educational domains. By seamlessly integrating with advanced visualisation tools, PSGO provides detailed insights into protein functions and interactions, enhancing its utility for research and teaching. For example, educators can utilise PSGO to demonstrate protein interactions in real-time during lectures, facilitating interactive learning experiences for students. On the other hand, researchers can leverage its analytical capabilities to explore novel protein structures, potentially uncovering new therapeutic targets or gaining deeper insights into protein dynamics. Thus, the potential of integrating PSGO and Chimera into ProMVR stands out as an indispensable resource for both the scientific community and educational institutions, fostering a more comprehensive understanding of biomolecular structures.

Integration of PSGO and ProMVR Leveraging the capabilities of PSGO and server-side UCSF Chimera, the author successfully handled their integration within the ProMVR framework, enabling dynamic online searches of the PDB database based on specific protein parameters (see Figure 5.13). Hosted on Yanlin's dedicated server, UCSF Chimera processes the PDB files retrieved from PSGO search results, employing command-line operations to parse them and convert them into the OBJ (3D format) files. These OBJ files are subsequently rendered in Unity and visually presented within the ProMVR scene, providing an immersive and interactive protein visualisation experience. This seamless integration facilitates a comprehensive platform where users can

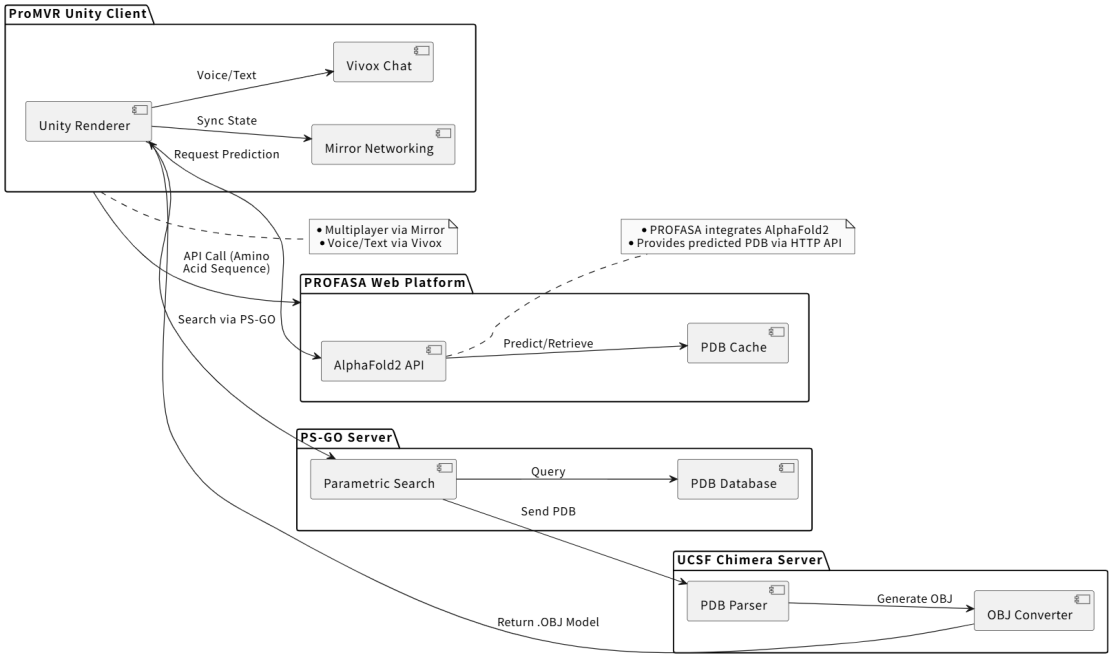


Figure 5.13: This diagram illustrates how the ProMVR client interacts with external services including the PROFASA platform (which hosts AlphaFold2), PS-GO search server, and UCSF Chimera for 3D conversion. Multiplayer state is synchronised via Mirror Networking, and real-time communication is handled through Vivox. Protein structure predictions are retrieved in PDB format and rendered in VR as OBJ models. This figure is reproduced under fair use for academic purposes.

explore protein structures in detail, significantly enhancing the education and research capabilities of ProMVR. A screenshot of the integrated platform is provided to illustrate this advanced functionality (see Figure 5.14).

To deeply integrate PSGO and UCSF Chimera into the ProMVR framework, the process begins with the ProMVR client initiating a connection to the server. When ProMVR is launched, initiate an automatic request to the server for a list of available PDB files managed by PSGO. This list is dynamically retrieved based on specific protein parameters predefined by the system setup (see Figure 5.16). The list is then displayed on the right side of the ProMVR interface. When a user selects a specific PDB file, ProMVR sends a request back to the server to fetch the chosen file. The server, leveraging UCSF Chimera, processes this PDB file using command-line scripts to parse and convert it into OBJ (Wavefront Object) format, which is suitable for rendering in Unity's 3D environment.

The conversion process involves several detailed steps, primarily handled by UCSF Chimera on the server side. First, Chimera uses its built-in functions to read and interpret the structural data contained within the PDB file. This includes analysing the atomic coordinates and bonding information to accurately reconstruct the protein's 3D structure. Once the structure is fully processed, Chimera employs its rendering tools to convert the structure into a 3D mesh represented as an OBJ file. This file format is chosen because of its compatibility with a wide range of 3D rendering software and its ability to maintain high fidelity of the model details, including textures and complex geometries that are crucial for realistic visualisation in educational and research settings (see Figure 5.17).

After the conversion, the OBJ file is compressed into a ZIP file along with the associated textures required for accurate rendering. This ZIP file is then sent back to the client-side ProMVR, through a secure data transmission protocol to ensure data integrity and privacy. Upon receiving the ZIP file, ProMVR utilises a package's decompression utilities to extract the OBJ file and texture file. The final step in the integration process is loading the OBJ file into the Unity engine, where it is instantiated and placed at the centre of the ProMVR scene. The central placement within the virtual space allows users to interact with the protein model, rotating and zooming it to study its structure in detail.

This interactive and immersive visualisation capability significantly enhances the user experience by allowing detailed exploration of protein structures within a VR

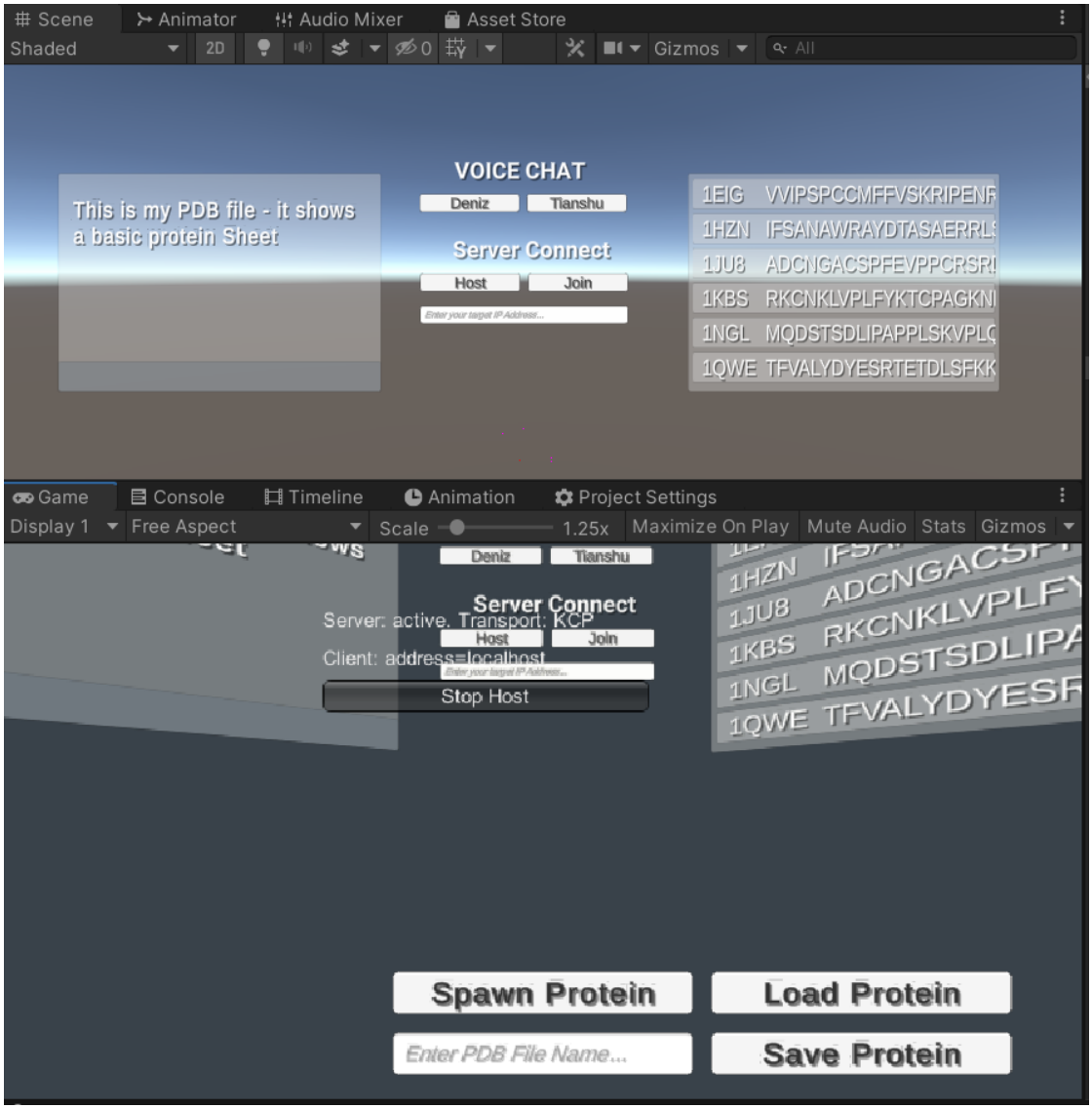


Figure 5.14: Integrated PSGO and ProMVR Workflow demonstrates the end-to-end protein analysis pipeline combining: (1) PSGO's parametric search interface displaying retrieved PDB sequences (e.g., 1EIG: V/IPSPCCMFFVSKRIPENF...), (2) Real-time VR collaboration through KCP-optimized voice/data channels with fairly low latency, and (3) Unified protein spawning/loading controls with automatic format conversion (PDB→OBJ via server-side Chimera). The image presented here illustrates the development phase of ProMVR. The bottom window displays the game mode, which, once completed, resembles this layout. The upper section represents the editing mode, where the user interface (UI) and game objects can be manipulated for tool design and development. This figure is reproduced under fair use for academic purposes.

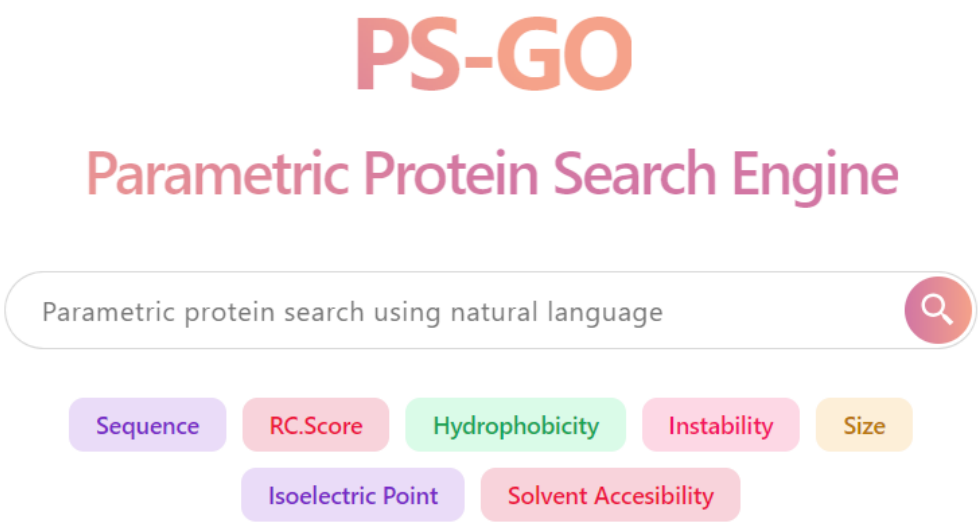


Figure 5.15: PSGO Parametric Protein Search Engine showcases the AI-driven bioinformatics platform integrated with ProMVR, featuring: (1) Natural language processing (NLP) for intuitive protein queries (e.g., "Find stable α-helix-rich proteins"), (2) Multi-parameter optimization, and (3) Real-time structural scoring that automatically prioritizes viable candidates for VR visualization. Source: [`https://psgo.ucc.ie/`]. This figure is reproduced under fair use for academic purposes.

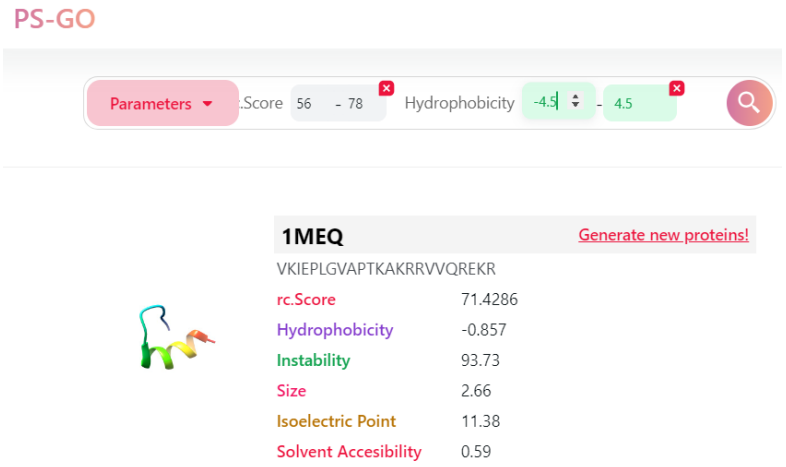


Figure 5.16: The parameters displayed in the figure are critical for understanding protein structure and function, as explained below: **Instability:** This score indicates the likelihood that the protein will degrade or misfold. A higher score suggests that the protein's structure is less stable and may be more prone to degradation. An unstable protein may rapidly lose its function or degrade, making this parameter essential when screening for functional proteins. **Hydrophobicity:** This value reflects the tendency of the protein's amino acids to interact with water. It is closely related to membrane localization, enzymatic activity, and interactions with other molecules. Hydrophobic residues typically reside in the protein's interior, away from water, while hydrophilic residues are found on the exterior. A balance between hydrophobic and hydrophilic regions is crucial for proper protein folding, stability, and function. A negative hydrophobicity value (-0.857 in the figure above) suggests that the protein has a tendency to be more hydrophobic, with hydrophobic regions contributing to the protein's stable folding by forming its core structure. **rc.Score:** This score provides a predicted measure of how well the protein structure will fold. Proteins with higher scores generally fold more easily and exhibit greater structural stability. **Size:** This parameter refers to the molecular weight or the number of amino acids within the protein. Size influences the protein's function and its capacity to interact with other molecules. **Isoelectric Point:** The isoelectric point represents the pH at which the protein has no net charge, where the positive and negative charges are balanced. This parameter significantly affects the protein's solubility and stability under varying pH conditions. **Solvent Accessibility:** This refers to the extent to which a protein's surface is exposed to the solvent (e.g., water). Solvent accessibility is vital for understanding protein interactions with other molecules, as well as facilitating protein-protein interactions. Source: [`https://psgo.ucc.ie/`]. This figure is reproduced under fair use for academic purposes.

```python
import os
import subprocess

def pdb_to_obj(pdb_path, obj_path):
    # Define Chimera command to open PDB and save as OBJ
    chimera_command = f"open {pdb_path}; write format obj 0 {obj_path}; stop"

    # Execute Chimera command
    subprocess.run(["chimera", "--nogui", "--script", chimera_command], check=True)

# Example usage
pdb_path = "example.pdb"
obj_path = "example.obj"
pdb_to_obj(pdb_path, obj_path)
```

Figure 5.17: Automated PDB-to-OBJ Conversion Pipeline illustrates the server-side molecular format transformation implemented in ProMVR, featuring: (1) Headless UCSF Chimera execution (—nogui flag), (2) Python subprocess management with error handling for batch process reliability. This automated workflow reduces manual intervention while maintaining crystallographic data fidelity for VR visualization. This figure is reproduced under fair use for academic purposes.

environment. Users can manipulate the model, zoom in on complex regions, and even simulate molecular interactions, providing a dynamic educational tool that bridges the gap between theoretical bioinformatics and practical molecular biology.

For researchers, this integration offers a tool for quick visualisation and hypothesis testing in structural biology, biochemistry, and related fields. The ability to dynamically search, retrieve, and visualise protein structures in real-time supports a rapid iteration cycle in research and development projects.

In Short, The PSGO system facilitates real-time protein model conversion by integrating Unity with UCSF Chimera. Initially, Unity sets up the game environment, and a Post request is made to the PSGO server. The Post request includes the PDB file as a form element, specifically attaching it as the ”PDB file” value. Upon receiving the request, the server processes the PDB file through UCSF Chimera, which parses the file and converts it into the OBJ format. This converted OBJ file is then sent back to Unity. In Unity, the OBJ file is instantiated as a 3D model, allowing it to be displayed within the game environment.

The integration of PSGO with UCSF Chimera within ProMVR not only streamlines the workflow from data retrieval to visualisation but also establishes a robust platform for advanced scientific research and education. This comprehensive approach leverages the strengths of each component (PSGO's efficient data handling, Chimera's powerful rendering capabilities, and ProMVR's interactive UI) to create a seamless and highly functional system that meets the growing demands of the scientific and educational communities in the field of molecular biology. This integration not only demonstrates the potential of VR in scientific applications but also sets a new standard for interactive molecular exploration in educational and professional settings.

5.4 Latency Test of ProMVR

5.4.1 Introduction

In the evolving landscape of VR applications, the quest for seamless and efficient multi-user environments is paramount, especially in complex domains such as protein visualisation. ProMVR, a pioneering system, offers a VR platform tailored for immersive protein visualisation and interaction. This chapter critically examines the latency and

performance dynamics of ProMVR, focusing on single and multi-PC configurations to discern their impacts on user experience and operational effectiveness.

The significance of this study lies in its potential to enhance educational and research methodologies through improved VR experiences. As VR technologies are increasingly integrated into scientific exploration and education, understanding their performance limitations and capabilities becomes crucial. This investigation thus serves a dual purpose: it not only assesses the technical performance of ProMVR but also contributes to the broader discourse on the optimisation of VR systems for complex, data-intensive tasks.

The approach involves a meticulous setup of test environments reflecting typical usage scenarios in educational and research settings. These environments are equipped with high-performance VR-capable PCs, using Unity and the Mirror networking library to handle VR synchronisation and manipulation functionalities effectively. By employing a comprehensive testing framework, the study toggles between single PC and multi-PC configurations offering insights into how different setups influence the core aspects of latency and frame rate, key indicators of system performance in VR applications.

For the single PC setup, the configuration used one PC to host and run the ProMVR application, simulating a standalone user environment. Multiple PCs were networked in a Local Area Network (LAN), with one serving as the host and others as client systems, each running independent instances of the VR application.

Methodologically, the study adheres to a systematic protocol, using both qualitative and quantitative measures to assess performance. Latency, a critical factor in VR user experience, is measured as the Round-Trip Time (RTT) of data packets within the ProMVR network. This metric is pivotal for real-time interactions, where delays can disrupt the immersive experience and impede user engagement. Additionally, frame rate analysis is conducted under various load conditions to identify any performance degradation that could detrimentally affect the visual fluidity of the VR experience.

Initial test establish baseline performance metrics, followed by simulations of network load to mimic the effect of multiple users interacting concurrently within the VR environment. The robustness of the network and application performance is further challenged through stress testing under peak load scenarios.

This comprehensive analysis provides a granular look at the performance profiles of different configurations. The findings reveal that while single PC setups show minimal

latency, the multi-PC setups exhibit increased latencies and occasionally reduced frame rates under high-load conditions, highlighting the need for network optimisation and potentially dedicated hardware solutions to handle increased data throughput in multi-user VR applications.

Through this exploration, the chapter sets the groundwork for subsequent discussions on strategies to mitigate observed performance bottlenecks, aiming to refine ProMVR's application in educational and research settings. The ultimate goal is to foster a more efficient and engaging platform for users, thereby advancing the use of VR scientific visualisation and interaction.

5.4.2 Rationale and Importance of Latency Evaluation in Multiuser VR Systems

Latency evaluation plays a foundational role in assessing the practical viability and user experience of real-time, multiuser virtual reality (VR) systems. Unlike single-user environments, multiuser VR applications require precise synchronization across geographically distributed users interacting within a shared 3D space. In such scenarios, even minor delays in visual rendering, gesture recognition, or object manipulation can significantly disrupt interaction flow, degrade the sense of presence, and impair collaborative task performance. As such, latency is not merely a technical metric but a core determinant of interaction quality in immersive systems.

In the context of ProMVR, which enables simultaneous manipulation and discussion of protein structures among multiple users, minimizing network and interaction latency is vital. The smoothness of real-time interactions directly impacts the system's suitability for scientific collaboration, particularly when users rely on tightly coordinated spatial operations such as rotating molecular structures, sharing focus points, or annotating features. Therefore, latency testing was conducted as a precondition for meaningful user experience evaluation, ensuring that the system's responsiveness could support the intended collaborative use cases without introducing perceptual or functional barriers.

Furthermore, from a systems engineering standpoint, latency testing validates the network architecture, data synchronization protocols, and rendering pipelines used in the development of ProMVR. It enables the identification of potential bottlenecks or incon-

sistencies under various network conditions, and helps establish performance baselines that are essential for future scalability assessments.

This evaluation aligns with best practices in the current State-of-the-Art (SoA) for collaborative VR applications, many of which emphasize low-latency guarantees as a prerequisite for deploying VR in real-time educational, medical, or industrial contexts. Comparable SoA systems, such as Mozilla Hubs, AltspaceVR, and various domain-specific platforms for scientific visualization, consistently report latency thresholds below which user experience degradation becomes noticeable (typically under 50–70 ms for interaction latency). The results of the latency testing in ProMVR place it within a competitive range, confirming that the system's architecture and optimization strategies are consistent with contemporary standards in immersive multiuser computing.

Thus, the inclusion of latency evaluation in this thesis is not a redundant technical formality, but a scientifically justified and strategically necessary process. It ensures that the system's interactive fidelity supports its intended use in collaborative protein design, and establishes a credible foundation for the subsequent user-centered assessments discussed in this chapter.

5.4.3 Methodology

The methodology for assessing the latency and performance of ProMVR under single-PC and multi-PC configurations was designed to provide a comprehensive analysis of the system's capabilities in different operational scenarios. Both qualitative and quantitative measures were employed to ensure a rigorous evaluation of the VR platform, focusing on key performance metrics such as latency and frame rate under various conditions.

Latency Meaturement Latency was a primary focus of this study, given its critical impact on the user experience in VR environments. Latency was measured as the RTT of data packets between the host and client PCs. The process began with a request from the client to the host for specific protein data files (PDB files), which were then processed and converted into 3D models (OBJ files) using UCSF Chimera. The processed files were subsequently transmitted back to the client for rendering within the VR environment. The time taken for each step was recorded, providing detailed insights into the latency associated with data retrieval, processing, and rendering.

Frame Rate Analysis Frame rate analysis was conducted to assess the smoothness of the VR experience, which is essential for maintaining immersion. Frame rates were monitored continuously during the visualisation of protein models, under both single-PC and multi-PC setups. The analysis considered various load conditions, from baseline performance with minimal network traffic to high-load scenarios with multiple simultaneous user interactions. This approach allowed for the identification of any performance degradation that could affect the visual fluidity and responsiveness of the VR system.

Testing Environment The testing was conducted in a controlled environment that simulated real-world educational and research settings. High-performance VR-capable PCs were used, each equipped with the necessary hardware to run Unity and the Mirror networking library. For the single-PC setup, a single machine hosted the ProMVR application, handling all processing and rendering tasks. In contrast, the multi-PC configuration involved networking multiple PCs in a LAN, with one serving as the host and others as clients. Each client ran independent instances of the ProMVR application, allowing for multi-user interaction within the same virtual environment.

Network Load Simulation To simulate the impact of network traffic on system performance, artificial network loads were generated during the tests. This involved introducing additional data packets into the network to mimic multiple users accessing and interacting with the VR environment simultaneously. The generated load was carefully calibrated to reflect typical usage scenarios in educational and research contexts, ensuring that the results were both realistic and applicable.

Stress Testing Stress testing was implemented to evaluate the robustness of the ProMVR system under peak load conditions. This involved subjecting the system to maximum feasible loads, including the highest number of simultaneous users and the most complex protein visualisations. The objective was to identify the threshold at which performance degradation became significant, providing valuable data for optimising the system's design and configuration.

This structured methodology ensures a comprehensive assessment of the latency and performance characteristics of ProMVR, providing insights that are critical for

optimising multi-user VR applications in complex, interactive settings. The findings from this chapter will contribute significantly to the development of more efficient and user-friendly VR systems for educational and research purposes.

5.4.4 Testing and Results

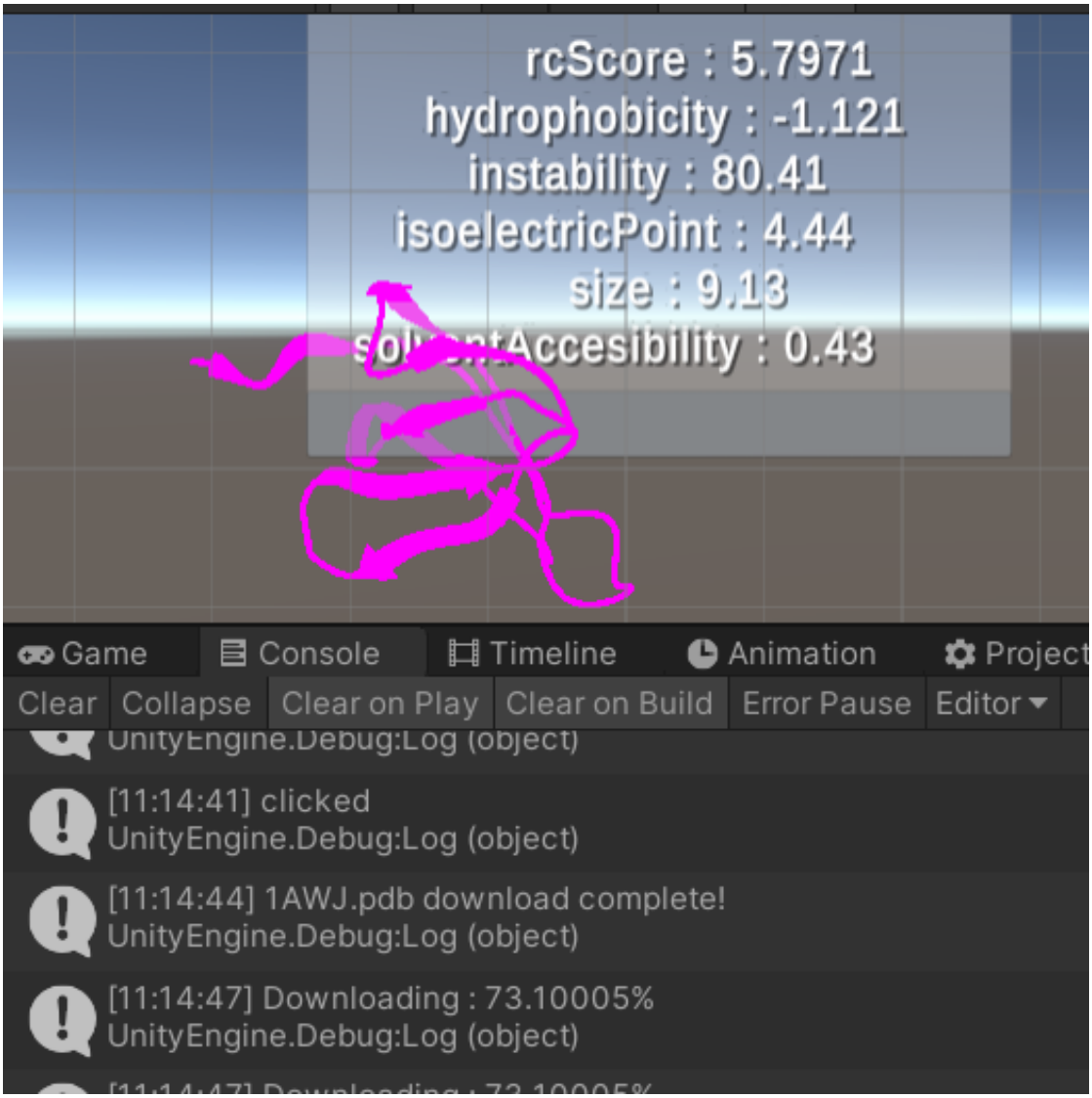
The performance of ProMVR was rigorously tested across single-PC and multi-PC configurations, with a focus on latency and frame rate metrics, both critical to the quality of the VR experience.

Single-PC Configuration Initial tests with the single-PC setup established that latency averaged below 20 milliseconds during PDB file downloading and 3D rendering, with frame rates consistently near 90 FPS, ideal for VR applications. These results indicate strong performance under minimal load conditions, supporting smooth and immersive user interactions.

A total of five PDB files were tested within the ProMVR system, each retrieved from Yanlin's server utilising the capabilities of PSGO. Upon requesting a PDB file, an HTTP request is dispatched to the server to download the file. This PDB file is then processed by server-side UCSF Chimera, which converts it into an OBJ file. The resulting OBJ file is subsequently rendered as a 3D model within ProMVR. The output below includes detailed protein parameters for the displayed PDB files, the rendered 3D protein models, and timestamps documenting the loading process. This comprehensive demonstration underscores the system's efficiency and effectiveness in visualising complex protein structures (see Figure 5.18).

The durations required for downloading the PDB files, retrieving the corresponding OBJ files, and loading the 3D protein models are detailed below, illustrating the efficiency of each step in the process (see Table 5.1).

The time consumed increases as the number of lines in the PDB file grows, suggesting a potential positive correlation. Specifically, It can be observed that the larger PDB files, such as 3109 lines, also result in longer processing times (1 minute). In contrast, smaller PDB files with fewer lines, such as 1413 lines, take only 8 seconds to process. This indicates that PDB file complexity (in terms of line count and size) influences the processing time required by Chimera.


rcScore : 5.7971
hydrophobicity : -1.121
instability : 80.41
isoelectricPoint : 4.44
size : 9.13
Accesibility : 0.43
Game
Console
Timeline
Animation
Project
Clear
Collapse
Clear on Play
Clear on Build
Error Pause
Editor
[11:14:41] clicked
UnityEngine.Debug:Log (object)
[11:14:44] 1AWJ.pdb download complete!
UnityEngine.Debug:Log (object)
[11:14:47] Downloading : 73.10005%
UnityEngine.Debug:Log (object)

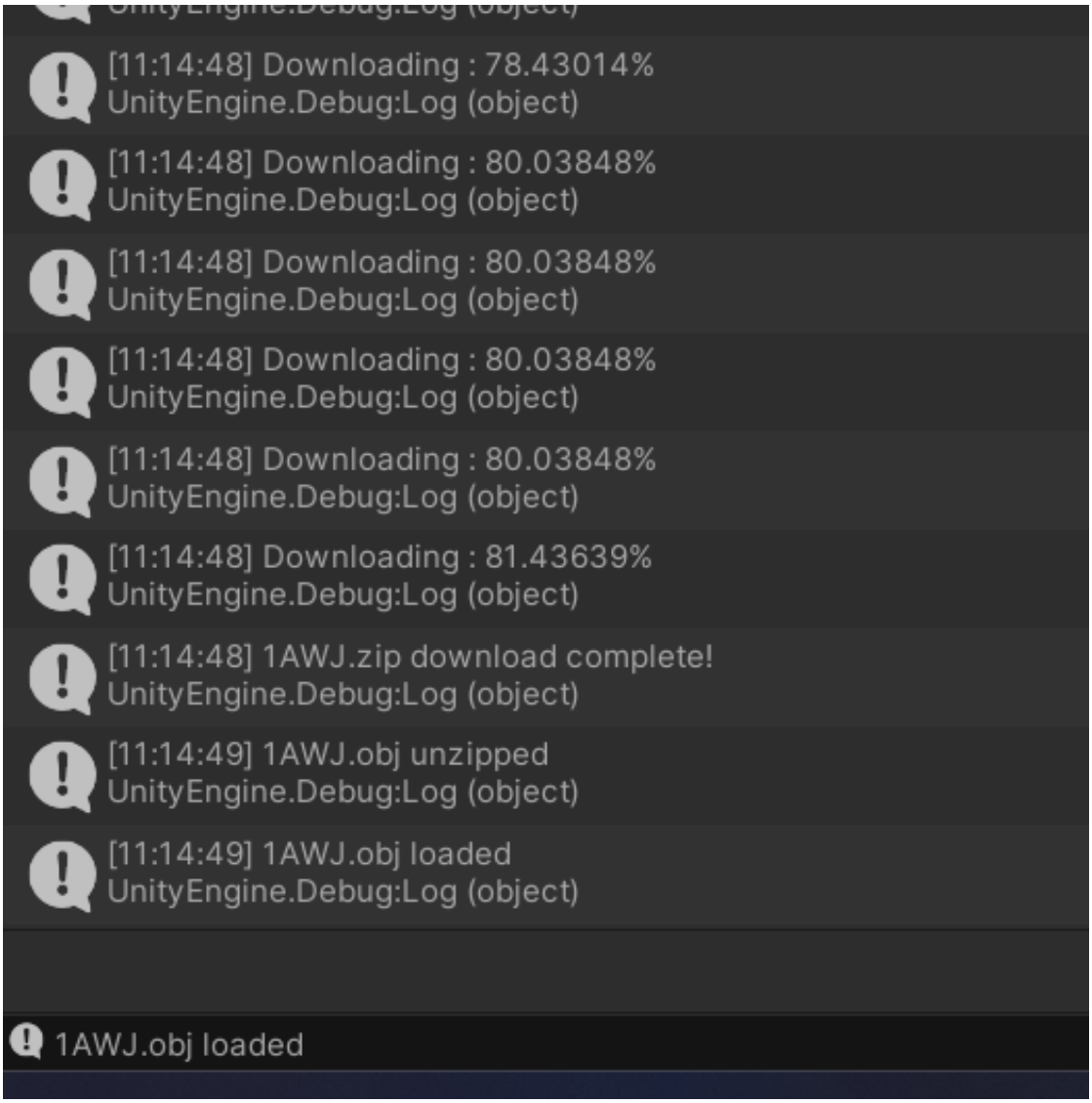


Figure 5.18: ProMVR Protein Loading Diagnostics captures the real-time data pipeline during molecular asset processing: (1) Progressive download monitoring, (2) Parallel execution of decompression (1AWJ.zip→OBJ) and biophysical parameter calculation (hydrophobicity: -1.121, instability: 80.41), and (3) Memory-efficient OBJ loading. This figure is reproduced under fair use for academic purposes.

Table 5.1: Calculated Latency

Test Index	PDB ID	Time Consumed	Lines in PDB File	Size of OBJ Output
1	1AWJ	8 seconds	1413	785KB
2	1A1J	12 seconds	1721	823KB
3	1DEM	13 seconds	1160	585KB
4	1CTO	15 seconds	1956	1.07MB
5	1BBX	1 minute	3109	10.8MB

The processing time is likely influenced by a combination of factors, including the size and complexity of the PDB file, Chimera's processing efficiency, the resulting OBJ file size, and network speed. The data suggests a general trend where more complex and larger files tend to require more time.

Multi-PC Configuration The multi-PC setup, involving networked PCs in a LAN, exhibited increased latency, particularly under simulated peak load conditions where network traffic was intentionally elevated to mimic multiple users interacting with the VR environment concurrently. Latency in these scenarios ranged from 30 to 50 milliseconds, with occasional frame rate drops below 60 FPS. These findings highlight potential issues in multi-user environments, where higher latencies and reduced frame rates could disrupt the immersive experience.

Stress testing further revealed the system's performance limitations. The multi-PC setup, under maximum load conditions with the highest number of simultaneous users and complex protein visualisations, displayed notable performance degradation. This included increased latency and significant frame rate reductions, suggesting that the network and computational resources were strained beyond optimal performance levels.

5.4.5 Discussion and Conclusion

The testing of ProMVR under both single-PC and multi-PC configurations provides critical insights into the system's performance in immersive VR environments, particularly in the context of protein visualisation. The results reveal that while the single-PC setup offers minimal latency and high frame rates, conducive to a smooth user experience,

the multi-PC setup introduces challenges that could impede the system's effectiveness in multi-user scenarios.

The significant increase in latency and occasional drops in frame rate in the multi-PC setup underscore the inherent complexities of managing networked VR environments. These issues are particularly pronounced under peak load conditions, where the system's performance degrades, potentially disrupting the immersive experience. The findings suggest that network congestion and the additional computational load on the host PC are the primary factors contributing to these performance bottlenecks.

To address these challenges, the implementation of advanced networking solutions and dedicated hardware appears necessary. Strategies such as load balancing, optimised data handling routines, and enhanced networking infrastructure could mitigate the observed performance issues, ensuring a more consistent and fluid experience across multiple users. Furthermore, exploring alternative architectures that distribute the computational load more evenly across participating systems could offer a pathway to enhancing ProMVR's scalability and responsiveness.

In conclusion, while ProMVR demonstrates robust performance in single-user scenarios, its application in multi-user environments requires further optimisation. Addressing the identified latency and frame rate issues will be crucial for realising the full potential of VR in educational and research settings. Future work should focus on refining the system's architecture and exploring innovative solutions to support the growing demand for collaborative VR applications in scientific visualisation.

5.5 User Experience Evaluation of ProMVR

5.5.1 Evaluation Objectives and Importance in Multiuser Systems

This section presents a scientifically grounded evaluation of the user experience with the ProMVR system, complementing the previously presented performance and latency tests. In multi-user virtual reality systems, user experience is a critical factor beyond hardware and software performance. The quality of interface design, clarity of interaction, and the level of user immersion significantly affect the effectiveness of the application, particularly in collaborative research and educational contexts. A scientif-

ically informed evaluation of user experience enables a more holistic understanding of system performance and informs future design and development decisions.

In collaborative virtual environments (CVEs), particularly those designed for complex scientific domains such as protein design, user experience (UX) evaluation plays a pivotal role. Unlike single-user systems, multiuser platforms demand not only system responsiveness but also seamless coordination, intuitive interfaces, and minimal cognitive load across participants. Evaluating these dimensions is critical to ensuring the platform's usability, educational value, and collaborative effectiveness.

While latency metrics offer a quantitative backbone, they do not holistically capture the qualitative aspects of user engagement and interaction fidelity. Hence, this evaluation phase bridges the gap between system-level testing and human-centered assessment, aligning with established methodologies in VR and HCI research.

5.5.2 Participants and Evaluation Context

To ensure both scientific rigor and practical relevance, the author conducted user experience evaluations with 4 expert participants (n=4), each possessing advanced academic backgrounds in relevant fields.

- Dr. Yanlin Mi: Developer of PS-GO and PROFASA platforms—key back-end systems supporting ProMVR.
- Dr. Stefan-Bogdan Marcu: Expert in software development and AI systems—provided technical usability insights.
- Dr. Deniz Mevlevioğlu: Specialist in HCI and interface visualization—advised on interaction design.
- Dr. Venkata Vamsi Bharadwaj Yallapragada: Project initiator with bioinformatics and protein engineering expertise—guided scientific validity of interactions.

All participants were selected due to their familiarity with both collaborative software tools and scientific visualisation environments. They engaged with the multi-user VR version of ProMVR in a collaborative setting, interacting with protein models, navigating interface elements, remote communication, and executing tasks representative of real scientific workflows. The evaluation focused on capturing both qualitative insights and

quantitative metrics regarding the usability, responsiveness, and overall satisfaction with the system.

The evaluation followed a structured heuristic walkthrough and cooperative evaluation format, during which the experts jointly interacted with the ProMVR system over the LAN-based multiuser mode. They were asked to:

- Perform protein loading, rotating, scaling, and interaction tasks collaboratively.
- Comment aloud on interaction fluidity, usability challenges, and interface intuitiveness.
- Fill in a standardised UX questionnaire adapted from the System Usability Scale (SUS). The full questionnaire used in this evaluation is provided in Appendix A.

While the participant number (n=4) is limited, this is justifiable for an expert-focused heuristic evaluation. As suggested by usability research, a small set of domain experts can often uncover the majority of usability issues, especially in specialised systems [144]. Furthermore, this study was not intended for broad statistical generalisation but rather as an expert evaluation of a niche scientific tool still in development. The findings will inform future iterations and larger-scale usability testing involving end-users in structural biology or education.

5.5.3 Evaluation Methodology

1. Semi-Structured Interviews and Observational Analysis

Each participant was invited to engage in a guided VR session using ProMVR, during which the author observed user behavior and interaction patterns. Following each session, semi-structured interviews were conducted to gather direct feedback. These interviews were aimed at capturing nuanced opinions on system usability, interface clarity, and the perceived effectiveness of key features such as molecular manipulation, network synchronization, and visual representation.

2. Structured Questionnaire and Quantitative Survey

To ensure methodological robustness, a structured questionnaire was administered after the VR sessions. The questionnaire was adapted from established evaluation frameworks

such as the System Usability Scale (SUS) and UX evaluation models for VR environments. The full questionnaire used in this evaluation is provided in Appendix A. The survey was designed to assess the following dimensions:

- **Ease of Use:** This refers to how intuitively users can understand and operate the system without requiring prior training or external guidance.
- **System Responsiveness:** This dimension evaluates how quickly and accurately the system responds to user input, particularly in a networked multi-user environment.
- **Functional Completeness:** This assesses the extent to which all expected core functionalities are present and operational.
- **Interface Clarity and Affordance:** This examines the visual clarity of the interface and the extent to which it naturally guides users toward appropriate actions.
- **Learnability and Onboarding Guidance:** This dimension considers how easily new users can become proficient with the system, including the availability and effectiveness of tutorials or prompts.
- **Overall User Satisfaction:** This measures the general subjective experience of the users, incorporating their affective response and likelihood of reuse or recommendation.

3. The Relevance of Multi-User Experience Testing
In contrast to single-user applications, multi-user VR environments present unique challenges, such as the need for real-time synchronization, latency tolerance, and spatial coordination. Evaluating user experience in such a setting is particularly important because interaction dynamics directly influence collaboration efficiency, cognitive load, and overall engagement. By incorporating collaborative use cases into this evaluation, the study addresses the scientific and practical significance of supporting distributed teamwork in immersive scientific visualization tools.

5.5.4 Evaluation Results - Observations and Findings

1. Qualitative Feedback - Perceived Interaction Quality and Responsiveness
Participants uniformly reported that the ProMVR system offered a ”straightforward” and

”intuitive interaction” model. All essential functions were accessible and functional, with no noticeable latency or synchronization issues during manipulation, rotation, or scaling of 3D protein models in multi-user sessions. The real-time synchronisation of avatars and models across devices was highlighted as a key strength, particularly given the absence of jitter or visual desync under normal network conditions. Users were able to collaboratively manipulate protein structures with minimal effort, and the interface responded smoothly to input commands.

- **Dr. Vamsi noted:** “The collaboration was smooth and natural, I didn’t even realise we were not on the same machine.”
- **Dr. Deniz highlighted:** “There was no perceptible lag, even when manipulating complex molecules with both users engaged.”

Participants also acknowledged that all core functionalities—protein model loading, rotating, scaling, teleportation, remote communication, and view switching—were fully implemented and operational. This indicates a high level of technical completeness at the prototype stage.

However, a common critique across all participants was directed at the visual design and interface aesthetics. The current UI was perceived as overly technical and lacking in visual affordance—interface elements were not always immediately understandable without prior knowledge. The absence of an integrated tutorial or onboarding guidance was also noted, which could hinder first-time users from effectively engaging with the system.

2. Quantitative Survey Results

The structured questionnaire yielded the following average scores, rated on a 5-point Likert scale (1 = very poor, 5 = excellent): (see Table 5.2)

3. Importance of Multi-User Interaction Evaluation

The findings from the multi-user experience underscore the significance of evaluating beyond individual performance metrics. While technical tests confirmed minimal latency, the user experience data provides a deeper understanding of how real users perceive such latency (or the lack thereof) in a collaborative context. Multi-user evaluations also help assess how effectively the system supports joint attention, shared tasks, and real-time interaction—key factors for scientific collaboration and educational deployment.

Table 5.2: ProMVR SUS Questionnaire Result

Evaluation Dimension	**Mean Score**	**Interpretation**
Ease of Use	4.2	The system was generally easy to operate and required minimal instructions.
System Responsiveness	4.7	The VR environment responded promptly and reliably to user interactions.
Functional Completeness	4.6	Core features were comprehensive and fully implemented.
Interface Clarity	3.2	UI elements were not always intuitive; affordances could be significantly improved.
Learnability and Guidance	3.0	Lack of embedded tutorials made onboarding more challenging for new users.
Overall Satisfaction	4.3	Participants expressed a high degree of satisfaction with the system's potential and performance.

The evaluation confirmed that network performance in ProMVR meets the demands of multi-user interaction, with users reporting no disruptive lag or desynchronization. These insights validate earlier latency testing results and demonstrate that the platform can support real-time, high-fidelity collaborative engagement in protein modeling scenarios.

5.5.5 Conclusion and Future Directions

The user experience evaluation of ProMVR has provided robust evidence supporting its usability, performance, and functional reliability in a multi-user scientific context. Key strengths include its low-latency performance, intuitive controls, and broad feature set. However, areas for improvement were also clearly identified—specifically in terms of UI design, visual clarity, and onboarding mechanisms.

In response to the evaluation findings, future iterations of ProMVR will prioritize the following development goals:

- **Enhanced Interface Design:** The visual design of the interface will be reworked to improve clarity, increase visual affordances, and ensure that functional elements are easily recognizable and accessible.
- **Integrated Tutorial System:** A built-in tutorial will be implemented to guide new users through essential interactions and features, reducing the learning curve and improving first-time usability.
- **Interaction Workflow Optimization:** Based on user feedback, specific interaction patterns and UI flows will be refined to better align with user expectations and mental models, thereby improving overall user comfort and enjoyment.

By combining rigorous user experience evaluation with previous system-level testing, ProMVR demonstrates strong potential as a collaborative VR platform for protein modeling. The system's ability to facilitate effective scientific exploration in immersive environments underscores its value in both academic research and educational settings.

6 Discussion and Future Work

6.1 Key Findings

The research presented in this thesis explored the transformative potential of VR technology in protein design and visualization, offering substantial contributions to both educational and research contexts. Through the development and evaluation of VR platforms, the findings demonstrated that immersive environments significantly enhance the interactivity and understanding of complex biological structures, such as proteins. The introduction of VR allows users to manipulate protein structures dynamically, making the learning and research process more intuitive and accessible.

One of the most prominent findings of this research is the effectiveness of VR tools in educational settings. Platforms like ”Pocket Peptides” and ”ProMVR” have demonstrated that VR can bridge the gap between theoretical understanding and practical application. These tools facilitate an immersive, hands-on learning experience where students and researchers alike can explore protein structures in real-time, manipulate folding patterns, and develop a more profound understanding of molecular biology.

Another critical outcome is the demonstration of VR's scalability in multi-user scenarios, as exemplified by the ”ProMVR” system. While the single-user experience has shown to be highly effective, the research also highlighted the challenges and potential of multiplayer VR environments in fostering collaborative learning and research. The multiplayer configuration of ProMVR allows users from different geographical locations to work together, visualise molecular structures, and discuss scientific problems within a shared virtual environment.

Supporting Evidence for the Effectiveness of VR Tools in Education Settings These educational benefits are not purely theoretical; they are supported by empirical findings from both single-user and multi-user VR evaluations presented in Section 4.5 and Chap-

ter 5. Specifically, survey data from 89 participants interacting with Pocket Peptides demonstrated that over 95% of users agreed or strongly agreed that the tool was visually engaging and the tutorial system was helpful (see Table 4.2). This is further reinforced by central tendency measures (e.g., median = 5 for appeal and tutorial clarity), indicating that the interface design successfully supports intuitive learning, even among users with limited prior exposure to molecular biology, and the VR-based approach facilitates user engagement and fosters intuitive learning. These findings substantiate the claim that VR tools such as Pocket Peptides offer an effective and user-friendly platform for educational applications in molecular biology.

In the multiplayer setting evaluated in ProMVR, qualitative evidence from Chapter 5 also supports the effectiveness of VR for collaborative education. The system enabled geographically distributed users to co-manipulate protein structures, exchange voice-based insights via Vivox, and synchronise molecular transformations in real time. This created a learning environment analogous to virtual classrooms or scientific meeting rooms, enhancing the sense of social presence and collaborative learning. Furthermore, latency and frame-rate testing in multi-PC setups (Section 5.4.4) demonstrated that the system could maintain real-time interaction fidelity, with average latency below 50ms — within the acceptable thresholds established in the state-of-the-art for immersive learning systems.

Together, these findings substantiate the claim that VR tools like Pocket Peptides and ProMVR are not only effective for individual learning but also capable of supporting real-time scientific collaboration. They address both pedagogical goals—improving conceptual understanding of 3D molecular structures—and practical constraints in education, such as limited access to physical lab infrastructure.

6.2 Interpretations

The results of this thesis indicate that VR technology can revolutionise how protein design and structural biology are taught and researched. Traditionally, these fields have relied on two-dimensional representations of protein structures, which limits the ability to fully understand their three-dimensional complexities. VR overcomes this limitation by offering an interactive and immersive environment where users can visualise and manipulate protein models in ways that would be impossible in traditional methods.

The implications of VR in educational settings are particularly significant. As educational paradigms shift towards more experiential and interactive methods of teaching, VR provides a powerful tool to engage students in a subject that is often perceived as abstract and complex. By allowing students to ”experience” proteins in three dimensions, VR facilitates a deeper understanding of molecular interactions, which may lead to improved retention of knowledge and a more comprehensive grasp of biological concepts.

In research settings, the ability to collaborate in real-time through multiplayer VR environments like ProMVR represents a significant advancement in scientific collaboration. The shared virtual environment allows for real-time discussions and manipulations of protein structures, which can enhance the collaborative process and lead to new insights that might not be possible through traditional, non-immersive methods. This innovation is particularly valuable in international research collaborations, where geographical barriers often limit real-time, hands-on collaboration.

6.3 Implications for Education and Research

The implications of this research for education and research in bioinformatics and structural biology are profound. VR has the potential to become a cornerstone of modern bioinformatics education, offering a more engaging and effective way to teach complex topics such as protein folding, molecular interactions, and structural biology. By integrating VR into the curriculum, educators can offer students a hands-on approach to learning, which is proven to enhance understanding and retention of knowledge.

Furthermore, the research highlights the potential for VR to transform remote learning. With the growing demand for online education, VR offers a scalable method for delivering high-quality, interactive science education to students around the world. Tools like ”Pocket Peptides” and ”ProMVR” can be deployed in virtual classrooms, allowing students to interact with protein structures and complete scientific challenges, even from remote locations. This capability is particularly relevant in the context of global education, where access to physical laboratories and resources may be limited.

In the realm of research, VR opens new avenues for collaborative exploration. The ProMVR platform, with its multi-user capabilities, enables researchers to work together in virtual spaces, manipulating protein structures and conducting experiments in real-time. This level of interactivity and collaboration will lead to faster discoveries and a

more dynamic research process, as it eliminates many of the barriers associated with traditional remote collaborations. Additionally, the ability to visualise proteins in three dimensions can offer new insights into molecular dynamics that are difficult to achieve through two-dimensional representations alone.

6.4 Limitations and Future Work

While the research presented in this thesis offers promising insights into the potential of VR in molecular biology, several limitations need to be addressed in future work. One of the most significant limitations is the novelty of VR technology in educational and research settings. Although the results are promising, VR is still a relatively new tool in these fields, and its long-term effectiveness remains to be validated across different educational contexts and with diverse user groups. Future research should focus on conducting large-scale studies to validate these findings and explore how VR can be tailored to meet the specific needs of different educational and research environments.

Another limitation is the current technological barriers associated with VR, particularly in multi-user environments. While ProMVR has demonstrated the potential of collaborative VR environments, the research also highlighted performance issues related to latency and frame rates in networked, multi-PC configurations. These technical challenges can disrupt the immersive experience, particularly when multiple users are interacting with complex protein models simultaneously. Future work should focus on optimizing these technical aspects, exploring advanced networking solutions, and developing more efficient algorithms for rendering and interaction within multi-user VR environments.

Additionally, while the tools developed in this research, such as Pocket Peptides and ProMVR, offer valuable educational and research applications, there is room for further development. For example, future versions of these tools will integrate more advanced protein design processes, allowing users to not only visualise and manipulate existing protein structures but also design new proteins from scratch. This can open up new possibilities for research in fields such as drug design and synthetic biology, where the ability to model and design proteins in real-time is critical.

Another promising direction for future work is the integration of Artificial Intelligence (AI) into VR platforms for molecular biology. AI-driven models can enhance the

predictive capabilities of tools like ProMVR, allowing users to explore potential protein folding pathways or predict the outcomes of molecular interactions. Combining AI with VR will lead to even more powerful educational and research tools, offering insights that are currently beyond the capabilities of traditional methods.

Finally, the research should explore the integration of AR with VR to create hybrid environments that combine the strengths of both technologies. AR can be used to overlay protein structures onto physical spaces, allowing users to interact with models in the real world while still benefiting from the immersive and interactive features of VR. This hybrid approach can further enhance the educational and research applications of immersive technologies in molecular biology.

6.5 Conclusion

In conclusion, this research has demonstrated the transformative potential of VR in molecular biology, particularly in the areas of education and research. The development of tools like Pocket Peptides and ProMVR represents a significant step forward in how we visualise and understand complex protein structures. By offering immersive, interactive environments, these tools have the potential to enhance both the learning experience for students and the collaborative process for researchers. As VR technology continues to evolve, its applications in molecular biology are likely to expand, leading to even more innovative and effective methods for teaching and researching this critical field.

References

[1] Andrea Tricco, Erin Lillie, Wasifa Zarin, Kelly O'Brien, Heather Colquhoun, Danielle Levac, David Moher, Micah Peters, Tanya Horsley, Laura Weeks, Susanne Hempel, Elie Akl, Christine Chang, Jessie Mcgowan, Lesley Stewart, Lisa Hartling, Adrian Aldcroft, Michael Wilson, Chantelle Garritty, and Sharon Straus. Prisma extension for scoping reviews (prisma-scr): Checklist and explanation. *Annals of Internal Medicine*, 169(7):467–473, 09 2018.

[2] Venkata Vamsi Bharadwaj Yallapragada, Tianshu Xu, Sidney Walker, Sabin Tabirca, and Mark Tangney. Pepblock builder vr – an open-source tool for gaming-based bio-edutainment in interactive protein design. *Frontiers in Bioengineering and Biotechnology*, 9:674211, 05 2021.

[3] Tianshu Xu, Venkata V. B. Yallapragada, Mark Tangney, and Sabin Tabirca. Schedio-pro: An interactive virtual reality game for protein design. In *Communications, Signal Processing, and Systems*, pages 293–300, Singapore, 2022. Springer Singapore.

[4] Tianshu Xu, Venkata VB Yallapragada, Mark Tangney, and Sabin Tabirca. Promvr - protein multiplayer virtual reality tool. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*, VRST '21, New York, NY, USA, 2021. Association for Computing Machinery.

[5] Tianshu Xu, Venkata VB Yallapragada, Mark Tangney, and Sabin Tabirca. Promvr-interactive protein educational tool. In *Proceedings of the 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, BCB '23, New York, NY, USA, 2023. Association for Computing Machinery.

[6] Yanlin Mi, Stefan-Bogdan Marcu, Sabin Tabirca, and Venkata V.B. Yallapragada. Ps-go parametric protein search engine. *Computational and Structural Biotechnology Journal*, 23:1499–1509, 2024.

[7] John Barrow, William Hurst, Joakim Edman, Natasja Ariesen, and Caspar Krampe. Virtual reality for biochemistry education: the cellular factory. *Education and Information Technologies*, 29:1–26, 05 2023.

[8] Pritee Patil and Tanisha Batra. Virtual reality in bioinformatics. *Open Access Journal of Science*, 3, 03 2019.

[9] Michael Levitt and Marc Baaden. Virtual-reality enabled molecular visualization for structural biology, 2018.

[10] Sandra Tan and Russell Waugh. Use of virtual-reality in teaching and learning molecular biology. *3D Immersive and Interactive Learning*, pages 17–43, 11 2014.

[11] Elke Moritz and Joerg Meyer. Interactive 3d protein structure visualization using virtual reality. In *Proceedings - Fourth IEEE Symposium on Bioinformatics and Bioengineering, BIBE 2004*, pages 503– 507, 06 2004.

[12] Joey Essoe, Nicco Reggente, Ai Ohno, Hera Baek, John Dell'Italia, and Jesse Rissman. Enhancing learning and retention with distinctive virtual reality environments and mental context reinstatement. *npj Science of Learning*, 7, 12 2022.

[13] Ilana Dubovi. Cognitive and emotional engagement while learning with vr: The perspective of multimodal methodology. *Computers & Education*, 183:104495, 2022.

[14] Jingyuan Chen, Zongjian Fu, Hongfeng Liu, and Jinku Wang. Effectiveness of virtual reality on learning engagement:: A meta-analysis. *International Journal of Web-Based Learning and Teaching Technologies*, 19:1–14, 01 2024.

[15] Mariangela Castro Arteaga, Mike Workentin, A K M Alamgir, and Courtney Kupka. Prisma statement and cochrane reviews: Striving to improve quality and validity of systematic reviews, 05 2022.

[16] Mythreye Venkatesan, Harini Mohan, Justin R. Ryan, Christian M. Schürch, Garry P. Nolan, David H. Frakes, and Ahmet F. Coskun. Virtual and augmented reality for biomedical applications. *Cell Reports Medicine*, 2(7):100348, 2021.

[17] Arhum Hakim and Sadaf Hammad. Use of virtual reality in psychology. In Cezary Biele, Janusz Kacprzyk, Wies law Kopeć, Jan W. Owsiński, Andrzej Romanowski, and Marcin Sikorski, editors, *Digital Interaction and Machine Intelligence*, pages 208–217, Cham, 2022. Springer International Publishing.

[18] Sarah Parsons and Sue Cobb. State-of-the-art of virtual reality technologies for children on the autism spectrum. *European Journal of Special Needs Education*, 26(3):355–366, 2011.

[19] Thomas Parsons, Andrea Gaggioli, and Giuseppe Riva. Virtual reality for research in social neuroscience. *Brain Sciences*, 7(4):42, 2017.

[20] Thomas D. Parsons. Virtual reality for enhanced ecological validity and experimental control in the clinical, affective and social neurosciences. *Frontiers in Human Neuroscience*, 9:660, 2015.

[21] Giuseppe Riva. Virtual reality in clinical psychology. *Comprehensive Clinical Psychology*, pages 91–105, 2022. Epub 2022 Apr 12, PMCID: PMC7500920.

[22] Arnau Baró Mas, Idriss Ismaël, and Nicolas Filliard. Indy: a virtual reality multiplayer game for navigation skills training. *2018 IEEE Fourth VR International Workshop on Collaborative Virtual Environments (3DCVE)*, pages 1–4, 2018.

[23] Nesse van der Meer, Vivian van der Werf, Willem-Paul Brinkman, and Marcus Specht. Virtual reality and collaborative learning: a systematic literature review. *Frontiers in Virtual Reality*, 4, 2023.

[24] F. Jerry Reen, Owen Jump, Brian P. McSharry, John Morgan, David Murphy, Niall O'Leary, Billy O'Mahony, Martina Scallan, and Briony Supple. The use of virtual reality in the teaching of challenging concepts in virology, cell culture and molecular biology. *Frontiers in Virtual Reality*, 2, 2021.

[25] Ekaterina Sviridova, Elena Yastrebova, Gulmira Bakirova, and Fayruza Rebrina. Immersive technologies as an innovative tool to increase academic success and motivation in higher education. *Frontiers in Education*, 8, 2023.

[26] Matthew J Page, Joanne E McKenzie, Patrick M Bossuyt, Isabelle Boutron, Tammy C Hoffmann, Cynthia D Mulrow, Larissa Shamseer, Jennifer M Tetzlaff, Elie A Akl, Sue E Brennan, Roger Chou, Julie Glanville, Jeremy M Grimshaw, Asbjørn Hróbjartsson, Manoj M Lalu, Tianjing Li, Elizabeth W Loder, Evan Mayo-Wilson, Steve McDonald, Luke A McGuinness, Lesley A Stewart, James Thomas, Andrea C Tricco, Vivian A Welch, Penny Whiting, and David Moher. The prisma 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, 372, 2021.

[27] Riyan Hidayat and Yousef Wardat. A systematic review of augmented reality in science, technology, engineering and mathematics education. *Education and Information Technologies*, 29, 09 2023.

[28] Ahmad A. Mazhar and Mustafa M. Al Rifaee. A systematic review of the use of virtual reality in education. In *2023 International Conference on Information Technology (ICIT)*, pages 422–427, 2023.

[29] Jaziar Radianti, Tim A. Majchrzak, Jennifer Fromm, and Isabell Wohlgenannt. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & Education*, 147:103778, 2020.

[30] Talia Tene, Jessica Tixi, María Robalino, María Salazar, Cristian Vacacela Gomez, and Stefano Bellucci. Integrating immersive technologies with stem education: a systematic review. *Frontiers in Education*, 9, 06 2024.

[31] Peng Wang, Peng Wu, Jun Wang, Hung-Lin Chi, and Xiangyu Wang. A critical review of the use of virtual reality in construction engineering education and training. *International Journal of Environmental Research and Public Health*, 15:1204, 06 2018.

[32] Mai T. Pham, Andrijana Rajic, Judy D. Greig, Jan M. Sargeant, Andrew Papadopoulos, and Scott A. McEwen. A scoping review of scoping reviews: Advancing the approach and enhancing the consistency. *Research Synthesis Methods*, 5(4):371–385, 2014.

[33] Hilary Arksey and Lisa O'Malley. Scoping studies: Towards a methodological framework. *International Journal of Social Research Methodology*, 8(1):19–32, 2005.

[34] Okolo Jane, Chinyere Ezeonwumelu, Chioma Ihuoma Barah, Ugwu Nnenna, and Kiu Publication Extension. Exploring the role of virtual reality (vr) and augmented reality (ar) in language learning. *RESEARCH INVENTION JOURNAL OF RESEARCH IN EDUCATION*, 3:94–100, 07 2024.

[35] Eshita Dhar, Umashnkar Upadhaya, Yaoru Huang, Mohy Uddin, George Manias, Dimosthenis Kyriazis, Usman Wajid, Hamza Alshawaf, and Syed Abdul Shabbir. A scoping review to assess the effects of virtual reality in medical education and clinical care. *Digital health*, 9:20552076231158022, 02 2023.

[36] Haowen Jiang, Sunitha Vimalesvaran, Jeremy Wang, Kee Lim, Sreenivasulu Mogali, and Lorainne Tudor Car. Virtual reality in medical students' education: A scoping review. *JMIR Medical Education*, 8, 11 2021.

[37] Carolina Cruz-Neira, Daniel J. Sandin, and Thomas A. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the cave. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, pages 135–142. ACM, 1993.

[38] Rakhi Radhamani, Hemalatha Sasidharakurup, Gopika Sujatha, Bipin G. Nair, Krishnashree Achuthan, and Shyam Diwakar. Virtual labs improve student's

performance in a classroom. In *International Conference on E-Learning, E-Education, and Online Training*, 2014.

[39] Nico Rutten, Wouter van Joolingen, and Jan Veen. The learning effects of computer simulations in science education. *Computers & Education*, 58:136–153, 01 2012.

[40] Balamuralithara Balakrishnan and Peter Woods. Virtual laboratories in engineering education: The simulation lab and remote lab. *Computer Applications in Engineering Education*, 17:108 – 118, 03 2009.

[41] Danielle Levac, Heather Colquhoun, and Kelly O'Brien. Scoping studies: Advancing the methodology. *Implementation science : IS*, 5:69, 09 2010.

[42] Micah Peters, Christina Godfrey, Hanan Khalil, Patricia Mcinerney, Deborah Parker, and Cassia Soares. Guidance for conducting systematic scoping reviews. *International Journal of Evidence-Based Healthcare*, 13:141–146, 07 2015.

[43] Zachary Munn, Micah Peters, Cindy Stern, Catalin Tufanaru, Alexa Mcarthur, and Edoardo Aromataris. Systematic review or scoping review? guidance for authors when choosing between a systematic or scoping review approach. *BMC Medical Research Methodology*, 18, 11 2018.

[44] Chris Dede. Immersive interfaces for engagement and learning. *Science (New York, N.Y.)*, 323(5910):66–69, 02 2009.

[45] Syed Haque and Shankar Srinivasan. A meta-analysis of the training effectiveness of virtual reality surgical simulators. *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society*, 10:51–8, 02 2006.

[46] Brian White, Stella Kim, Katerina Sherman, and Nicole Weber. Evaluation of molecular visualization software for teaching protein structure: Differing outcomes from lecture and lab. *Biochemistry and Molecular Biology Education*, 30:130—136, 03 2002.

[47] Tassos Mikropoulos and Antonis Natsis. Educational virtual environments: A ten-year review of empirical research (1999–2009). *Computers & Education*, 56:769–780, 04 2011.

[48] Janna Anderson and Lee Rainie. The future of well-being in a tech-saturated world, 2018.

[49] Tomasz Mazuryk and Michael Gervautz. Virtual reality—history, applications, technology and future. Technical report, Vienna Institute of Technology, 1996.

[50] Zahira Merchant, Ernest Goetz, Lauren Cifuentes, Wendy Keeney-Kennicutt, and Trina Davis. Effectiveness of virtual reality-based instruction on students' learning outcomes in k-12 and higher education: A meta-analysis. *Computers & Education*, 70:29–40, 01 2014.

[51] Sara de Freitas and Tim Neumann. The use of 'exploratory learning' for supporting immersive learning in virtual environments. *Computers & Education*, 52:343–352, 02 2009.

[52] Hsiu-Mei Huang, Ulrich Rauch, and Shu-Sheng Liaw. Investigating learners' attitudes toward virtual reality learning environments: Based on a constructivist approach. *Computers & Education*, 55:1171–1182, 11 2010.

[53] Andrew J. McGehee, Sutanu Bhattacharya, Rahmatullah Roche, and Debswapna Bhattacharya. Polyfold: An interactive visual simulator for distance-based protein folding. *PLOS ONE*, 15(12):1–11, 12 2020.

[54] Kevin C. Cassidy, Jan Šefčík, Yogindra Raghav, Alexander Chang, and Jacob D. Durrant. Proteinvr: Web-based molecular visualization in virtual reality. *PLOS Computational Biology*, 16(3):1–17, 03 2020.

[55] Solomon Oyelere, Nacir Bouali, Rogers Kaliisa, George Obaido, Abdullahi Yunusa, and Ebunayo Jimoh. Exploring the trends of educational virtual reality games: a systematic review of empirical studies. *Smart Learning Environments*, 7:1–22, 10 2020.

[56] Yang Zhang and Jeffrey Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins*, 57:702–10, 09 2007.

[57] Patrice Koehl and Michael Levitt. De novo protein design. i. in search of stability and specificity. *Journal of molecular biology*, 293:1161–81, 12 1999.

[58] Patrice Koehl and Michael Levitt. De novo protein design. ii. plasticity in sequence space. *Journal of molecular biology*, 293:1183–93, 12 1999.

[59] Joy Yeh, T.J. Brunette, David Baker, Simon McIntosh-Smith, and Fabio Parmeggiani. Elfin: An algorithm for the computational design of custom three-dimensional structures from modular repeat protein building blocks. *Journal of Structural Biology*, 201, 09 2017.

[60] Laura Freina and Michela Ott. A literature review on immersive virtual reality in education: State of the art and perspectives. *eLearning & Software for Education*, 1:133–141, 04 2015.

[61] Kandise Salerno. Video games and learning: Teaching and participatory culture in the digital age, by kurt squire. *Alberta Journal of Educational Research*, 59:129–132, 12 2013.

[62] Nicholas Rego and David Koes. 3dmol.js: molecular visualization with webgl. *Bioinformatics*, 31(8):1322–1324, 2015.

[63] Guido Makransky and Lau Lilleholt. A structural equation modeling investigation of the emotional value of immersive virtual reality in education. *Educational Technology Research and Development*, 66:1141–1164, 02 2018.

[64] Chris Craig and Robin Kay. A systematic overview of reviews of the use of immersive virtual reality in higher education. *Higher Learning Research Communications*, 13:42–60, 12 2023.

[65] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell*. Garland Science, 6th edition, 2015.

[66] Carl I. Branden and John Tooze. *Introduction to Protein Structure*. Garland Science, 2nd edition, 1998.

[67] Thomas E. Creighton. *Proteins: Structures and Molecular Properties*. W. H. Freeman and Company, 2nd edition, 1993.

[68] Christian B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.

[69] Linus Pauling and Robert B. Corey. Configurations of polypeptide chains with favored orientations of the polypeptide chains. *Proceedings of the National Academy of Sciences*, 37(4):205–211, 1951.

[70] Linus Pauling and Robert B. Corey. Configurations of polypeptide chains with favored orientations around single bonds: Two new pleated sheets. *Proceedings of the National Academy of Sciences of the United States of America*, 37 11:729–740, 1951.

[71] Cyrus Chothia. Structural invariants in protein folding. *Nature*, 254:304–308, 1973.

[72] Alan Fersht. *Structure and Mechanism in Protein Science: A Guide to Enzyme Catalysis and Protein Folding*. W.H. Freeman and Company, New York, 1999. Print.

[73] Bassil I. Dahiyat and Stephen L. Mayo. De novo protein design: Fully automated sequence selection. *Science*, 278(5335):82–87, 1997.

[74] Frances H. Arnold. Design by directed evolution. *Accounts of Chemical Research*, 31(3):125–131, 1998.

[75] William Humphrey, Andrew Dalke, and Klaus Schulten. Vmd: Visual molecular dynamics. *Journal of Molecular Graphics*, 14(1):33–38, 1996.

[76] Eric F. Pettersen, Thomas D. Goddard, Conrad C. Huang, Gregory S. Couch, Daniel M. Greenblatt, Elaine C. Meng, and Thomas E. Ferrin. Ucsf chimera—a visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, 25(13):1605–1612, 2004.

[77] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, and Philip E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 2000.

[78] Frances C. Bernstein, Thomas F. Koetzle, Graheme J.B. Williams, Edgar F. Meyer, Michael D. Brice, John R. Rodgers, Olga Kennard, Takehiko Shimanouchi, and Mitsuo Tasumi. The protein data bank: A computer-based archival file for macromolecular structures. *Archives of Biochemistry and Biophysics*, 185(2):584–591, 1978.

[79] William R. Pearson and David J. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8):2444–2448, 1988.

[80] Michael Levitt. The birth of computational structural biology. *Nature structural biology*, 8:392–393, 06 2001.

[81] Georgii Krivov, Maxim Shapovalov, and Roland Dunbrack. Improved prediction of protein side-chain conformations with scwrl4. *Proteins*, 77:778–95, 12 2009.

[82] Conrad C. Huang, Gregory S. Couch, Eric F. Pettersen, and Thomas E. Ferrin. Chimera: An extensible molecular modeling application constructed using standard components. In *Pacific Symposium on Biocomputing*, volume 1, pages 724–735, 1996.

[83] Elaine C. Meng, Eric F. Pettersen, Gregory S. Couch, Conrad C. Huang, and Thomas E. Ferrin. Tools for integrated sequence and structure analysis with ucsf chimera. *BMC Bioinformatics*, 7:339, 2006.

[84] Thomas Goddard, Conrad Huang, and Thomas Ferrin. Software extensions to ucsf chimera for interactive visualization of large molecular assemblies. *Structure (London, England : 1993)*, 13:473–482, 04 2005.

[85] Andrew Leaver-Fay, Michael Tyka, Steven M. Lewis, et al. Rosetta3: An object-oriented software suite for the simulation and design of macromolecules. *Methods in Enzymology*, 487:545–574, 2011.

[86] Frank DiMaio, Andrew Leaver-Fay, Philip Bradley, David Baker, and Ingemar Andre. Modeling symmetric macromolecular structures in rosetta3. *PLoS ONE*, 6(6):e20450, 2011.

[87] Kim T. Simons, Charles Kooperberg, Eugene Huang, and David Baker. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. *Journal of Molecular Biology*, 268(1):209–225, 1997.

[88] Philip Bradley, Kira M. S. Misura, and David Baker. Toward high-resolution de novo structure prediction for small proteins. *Science*, 309(5742):1868–1871, 2005.

[89] Carol A. Rohl, Charlie E. M. Strauss, Kira M. S. Misura, and David Baker. Protein structure prediction using rosetta. *Methods in Enzymology*, 383:66–93, 2004.

[90] Richard Bonneau and David Baker. Ab initio protein structure prediction: Progress and prospects. *Annual Review of Biophysics and Biomolecular Structure*, 30(1):173–189, 2001.

[91] Rhiju Das and David Baker. Macromolecular modeling with rosetta. *Annual Review of Biochemistry*, 77:363–382, 2008.

[92] Yifan Song, Michael D. Tyka, Andrew Leaver-Fay, et al. High-resolution comparative modeling with rosettacm. *Structure*, 21(10):1735–1742, 2013.

[93] Andrew Leaver-Fay, Ron Jacak, P. Benjamin Stranges, and Brian Kuhlman. A generic program for multistate protein design. *PLoS ONE*, 6(8):e20937, 2011.

[94] Brian Kuhlman, Gautam Dantas, Gregory C. Ireton, et al. Design of a novel globular protein fold with atomic-level accuracy. *Science*, 302(5649):1364–1368, 2003.

[95] Kristian W. Kaufmann, Gordon H. Lemmon, Stephanie L. Deluca, John H. Sheehan, and Jens Meiler. Practically useful: What the rosetta protein modeling suite can do for you. *Biochemistry*, 49(14):2987–2998, 2010.

[96] Daniel J. Mandell, Evangelos A. Coutsias, and Tanja Kortemme. Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling. *Nature Methods*, 6(8):551–552, 2009.

[97] Sarel J. Fleishman, Andrew Leaver-Fay, Jacob E. Corn, et al. Rosettascripts: A scripting language interface to the rosetta macromolecular modeling suite. *PLoS ONE*, 6(6):e20161, 2011.

[98] Daniel-Adriano Silva, S. Yu, Umut Y. Ulge, et al. De novo design of potent and selective mimics of il-2 and il-15. *Nature*, 565(7736):186–191, 2019.

[99] Lin Jiang, Eric A. Althoff, Fernando R. Clemente, et al. De novo computational design of retro-aldol enzymes. *Science*, 319(5868):1387–1391, 2008.

[100] Shaun M. Lippow, K. Dane Wittrup, and Bruce Tidor. Computational design of antibody-affinity improvement beyond in vivo maturation. *Nature Biotechnology*, 25(10):1171–1176, 2007.

[101] Sergey Lyskov and Jeffrey Gray. The rosettadock server for local protein-protein docking. *Nucleic acids research*, 36:W233–8, 08 2008.

[102] Jeffrey J. Gray. High-resolution protein–protein docking. *Current Opinion in Structural Biology*, 16(2):183–193, 2006.

[103] Louis Clark, Skanth Ganesan, Sarah Papp, and Herman Vlijmen. Trends in antibody sequence changes during the somatic hypermutation process. *Journal of immunology (Baltimore, Md. : 1950)*, 177:333–40, 08 2006.

[104] Ambrish Roy, Alper Kucukural, and Yang Zhang. I-tasser: A unified platform for automated protein structure and function prediction. *Nature Protocols*, 5(4):725–738, 2010.

[105] Yang Zhang. I-tasser server for protein 3d structure prediction. *BMC Bioinformatics*, 9:40, 2008.

[106] Jianyi Yang, Rui Yan, Ambrish Roy, Dong Xu, Jeffrey Poisson, and Yang Zhang. The i-tasser suite: Protein structure and function prediction. *Nature Methods*, 12(1):7–8, 2015.

[107] Yang Zhang. I-tasser: Fully automated protein structure prediction in casp8. *Proteins*, 77(S9):100–113, 2009.

[108] Sitao Wu and Yang Zhang. Lomets: A local meta-threading-server for protein structure prediction. *Nucleic Acids Research*, 35(10):3375–3382, 2007.

[109] Jinbo Xu and Yang Zhang. How significant is a protein structure similarity with tm-score = 0.5? *Bioinformatics*, 26(7):889–895, 2010.

[110] Yuedong Yang et al. Improving protein fold recognition and template-based modeling by employing probabilistic-based matching between predicted one-dimensional structural properties of query and corresponding native properties of templates. *Bioinformatics*, 27(15):2076–2082, 2011.

[111] Jianyi Yang et al. Resq: An approach to unified estimation of b-factor and residue-specific error in protein structure prediction. *Journal of Molecular Biology*, 428(4):693–701, 2016.

[112] Yang Zhang and Jeffrey Skolnick. Tertiary structure predictions on a comprehensive benchmark of medium to large size proteins. *Biophysical Journal*, 87(4):2647–2655, 2004.

[113] Ambrish Roy and Yang Zhang. Recognizing protein-ligand binding sites by global structural alignment and local geometry refinement. *Structure*, 20(6):987–997, 2012.

[114] Jianyi Yang, Ambrish Roy, and Yang Zhang. Protein–ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment. *Bioinformatics*, 29(20):2588–2595, 2013.

[115] Yang Zhang and Jeffrey Skolnick. Tm-align: A protein structure alignment algorithm based on the tm-score. *Nucleic Acids Research*, 33(7):2302–2309, 2005.

[116] Jianyi Yang and Yang Zhang. I-TASSER server: new development for protein structure and function predictions. *Nucleic Acids Research*, 43(W1):W174–W181, 04 2015.

[117] Andriy Kryshtafovych et al. Critical assessment of methods of protein structure prediction (casp)-round xiv. *Proteins*, 89(12):1607–1617, 2021.

[118] John Jumper, Richard Evans, Alexander Pritzel, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[119] Mihaly Varadi, Sarah Anyango, Mihir Deshpande, et al. Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Research*, 50(D1):D439–D444, 2022.

[120] Yanlin Mi, Stefan-Bogdan Marcu, Sabin Tabirca, and Venkata V. B. Yallapragada. Profasa—a web-based protein fragment and structure analysis workstation. *Frontiers in Bioengineering and Biotechnology*, Volume 11 - 2023, 2023.

[121] Seth Cooper et al. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760, 2010.

[122] Firas Khatib et al. Crystal structure of a monomeric retroviral protease solved by protein folding game players. *Nature Structural & Molecular Biology*, 18(10):1175–1177, 2011.

[123] Christopher B. Eiben et al. Increased diels-alderase activity through backbone remodeling guided by foldit players. *Nature Biotechnology*, 30(2):190–192, 2012.

[124] Robert Kleffner et al. Foldit standalone: a video game-derived protein structure manipulation interface for crowdsourcing bioinformatics research. *Bioinformatics*, 33(17):2765–2767, 2017.

[125] Roland Hess. *Blender Foundations: The Essential Guide to Learning Blender 2.5*. Routledge, 1 edition, 2010.

[126] John M. Blain. *The Complete Guide to Blender Graphics: Computer Modeling & Animation, Fifth Edition*. A K Peters/CRC Press, 5 edition, 2019.

[127] Sa Wang, Zhengli Mao, Changhai Zeng, Huili Gong, Shanshan Li, and Beibei Chen. A new method of virtual reality based on unity3d. In *2010 18th International Conference on Geoinformatics*, pages 1–5, 2010.

[128] J. Radianti, T. A. Majchrzak, J. Fromm, and I. Wohlgenannt. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & Education*, 147:103778, 2020.

[129] Guido Makransky and Lau Lilleholt. A structural equation modeling investigation of the emotional value of immersive virtual reality in education. *Educational Technology Research and Development*, 66(4):1141–1164, 2018.

[130] Zahira Merchant, Ernest T. Goetz, Lauren Cifuentes, Wendy Keeney-Kennicutt, and Trina J. Davis. Effectiveness of virtual reality-based instruction on students' learning outcomes in k-12 and higher education: A meta-analysis. *Computers & Education*, 70:29–40, 2014.

[131] Tassos A. Mikropoulos and Anastasios Natsis. Educational virtual environments: A ten-year review of empirical research (1999–2009). *Computers & Education*, 56(3):769–780, 2011.

[132] B. M. Kyaw, N. Saxena, P. Posadzki, et al. Virtual reality for health professions education: Systematic review and meta-analysis. *Journal of Medical Internet Research*, 21(1):e12959, 2019.

[133] Nayoung Chung and Yiannis Demiris. An exploratory study of the use of virtual reality to enhance historical education. *Journal of Educational Computing Research*, 57(3):733–755, 2018.

[134] Lucas Jensen. The future of education: Virtual reality in the classroom. *Computers & Education*, 101:1–10, 2016.

[135] Matt Bower, Caroline Howe, Natasha McCredie, Amanda Robinson, and Debra Grover. Augmented reality in education – cases, places and potentials. *Educational Media International*, 54(3):1–15, 2017.

[136] Laura Freina and Michela Ott. A literature review on immersive virtual reality in education: State of the art and perspectives. *eLearning & Software for Education*, 1:133–141, 2015. Available on ResearchGate.

[137] Jennifer Parong and Richard E. Mayer. Learning science in immersive virtual reality. *Journal of Educational Psychology*, 110(6):785–797, 2018.

[138] Lucas Jensen and F Konradsen. A review of the use of virtual reality head-mounted displays in education and training. *Education and Information Technologies*, 23:1515–1529, 2018.

[139] Guido Makransky and Lau Lilleholt. A structural equation modeling investigation of the emotional value of immersive virtual reality in education. *Educational Technology Research and Development*, 66(4):1141–1164, 2018.

[140] Tassos A. Mikropoulos and Anastasios Natsis. Educational virtual environments: A ten-year review of empirical research (1999–2009). *Computers & Education*, 56(3):769–780, 2011.

[141] Z. Merchant, E. T. Goetz, L. Cifuentes, W. Keeney-Kennicutt, and T. J. Davis. Effectiveness of virtual reality–based instruction on students' learning outcomes in k-12 and higher education: A meta-analysis. *Computers & Education*, 70:29–40, 2014.

[142] Benjamin S Bloom, Max D Engelhart, Edward J Furst, Walker H Hill, and David R Krathwohl. *Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook I: Cognitive Domain*. Longmans, Green, 1956.

[143] David A Kolb. *Experiential Learning: Experience as the Source of Learning and Development*. Prentice Hall, 1984.

[144] Jakob Nielsen. *Usability Inspection Methods*. John Wiley & Sons, 1994.

[145] Noel O'Boyle, Michael Banck, Craig James, Chris Morley, Tim Vandermeersch, and Geoffrey Hutchison. Open babel: An open chemical toolbox. *Journal of cheminformatics*, 3:33, 10 2011.

A User Experience Questionnaire for ProMVR

Context

This questionnaire was administered to four domain experts (n=4) after a collaborative ProMVR session. Participants included developers of integrated tools (PSGO/PROFASA), VR/AI engineers, and protein design specialists. Responses informed iterative improvements to the prototype. It comprises both Likert-scale and open-ended questions:

Section A: System Usability (Likert Scale, 1 = Strongly Disagree, 5 = Strongly Agree)

1. I found the ProMVR system easy to learn.
2. The interactions (e.g., rotating, scaling) were intuitive.
3. The PSGO-Chimera pipeline for PDB-to-OBJ conversion worked seamlessly.
4. Avatar movements and hand tracking were synchronized accurately.
5. I was able to clearly see and identify the protein models in VR.
6. The voice/text communication feature enhanced collaboration.
7. I felt comfortable performing tasks within the VR environment.
8. The UI elements were easy to understand and use.
9. I could naturally coordinate tasks with other users.
10. The system maintained consistent molecular views across participants.
11. Overall, the system is useful for collaborative protein exploration.

Section B: Open-Ended Feedback

- What features did you find most effective for collaborative analysis?
- Describe any technical or usability barriers encountered.
- Suggest improvements to the interface or workflow.
- Proposed features for future versions (e.g., annotation tools).

[1]

[1] Scores were averaged across participants (n=4) and analyzed for thematic patterns.