

Protein Folding Demo Application for the C-Station

Graduation Internship Report



Amiril Rahmi Faradini

11 June 2013

Amsterdam

GRADUATION REPORT
FONTYS UNIVERSITY OF APPLIED SCIENCES
HBO-ICT: English Stream

Data student:	
Family name , initials:	Faradini, A. R.
Student number:	2147921
project period: (from – till)	28 January 2013 – 10 June 2013
Data company:	
Name company/institution:	PS-Tech
Department:	Technology development
Address:	Falckstraat 53hs 1017VV Amsterdam, The Netherlands
Company tutor:	
Family name, initials:	Koomen, Wim
Position:	Technology development specialist
University tutor:	
Family name , initials:	Aarts, Henk
Final report:	
Title:	Protein Folding demo application for the C-Station
Date:	10 June 2013

Approved and signed by the company tutor:

Date: 7 June 2013

Signature:

Preface

This report describes the processes and the result of my graduation project at Personal Space Technology (PS-Tech) in Amsterdam. This report is written in an informative manner in order to receive good marks from the jury during the final assessment. Hence, I can obtain my bachelor degree and start my professional career.

I would like to express my deep appreciation to my company tutor, Mr. Wim Koomen for giving me motivations, expert guidance and support throughout the graduation project. I would like to thank my colleagues in PS-Tech for giving me general guidance and their sense of humor always brought laughter on daily basis.

I would also like to thank my school tutor, Mr. Henk Aarts, who is keeping me in track and answering my questions by mail, phone or meeting appointment.

Last but not least, I would like to thank my parents and family in Indonesia for their constant support.

Amiril Rahmi Faradini

7 June 2013

Table of Contents

Summary	3
Glossary	5
1. Introduction	7
2. Company and products	8
2.1. Company	8
2.2. Company products	8
2.2.1. PST	8
2.2.2. C-Station	9
3. Assignment overview	11
3.1. Current situation	11
3.2. Issues	11
3.3. Assignment definition and initial requirements	11
3.4. Project Plan and project phases	12
4. Research	14
4.1. Research questions	14
4.2. Resources	15
5. Analysis phase	16
5.1. PST communication option	16
5.1.1. PST Software Development Kit	16
5.1.2. Virtual Reality Peripheral Network	17
5.2. Unity3D	17
5.2.1. Integration solutions	17
5.2.2. Solution: Unity Indie VRPN Adapter	18
5.2.3. Solution: MiddleVR	18
5.3. Autodesk Showcase and integration option	19
5.3.1. Other Autodesk products	19
5.4. Demo application concepts	19
5.4.1. Protein folding	20

5.4.2. Showcase	21
6. PST - Unity3D integration phase: Design, implement and test	22
6.3. Design phase: Architectural design	22
6.4. Implementation and test phase	23
6.4.1. Integration by using UIVA	23
6.4.2. Integration by using MiddleVR	25
7. PST – Autodesk Showcase integration phase: Design, implement and test	26
7.3. Design phase: Architectural design	26
7.4. Implementation and test phase	26
8. Demo application development phase: Design, implement and test	27
8.1. Final demo application concept	27
8.2. Functionality list: Must Have	27
8.3. Backbone packing level	28
8.4. 3D objects and user interface design	28
8.5. Implementation and test for the Must Have functionalities	29
8.5.1. Issue detectors in the protein folding demo application	29
8.5.2. Interaction with the protein object	30
8.5.3. Stereoscopic 3D	32
9. Protein folding demo application	34
9.1. Flow diagram	34
9.2. 3D interaction and mouse interaction mode	35
9.3. Protein folding Unity3D custom package	36
10. Conclusions and recommendations	37
Personal Evaluation	38
References	39
Appendices	40
Appendix I: Internship Survey	40
Appendix II: Project Plan	42
Appendix III: Project schedule	50

List of Figures and Tables

Figure 1 PS-Tech organogram	8
Figure 2 PST with 3D interaction devices	9
Figure 3 PST software is tracking 3D interaction devices	9
Figure 4 C-Station and colocated interaction	10
Figure 5 Vesalius3D with the 3D MRI brain data	10
Figure 6 Methodology	13
Figure 7 Interfacing option in the PST software	16
Figure 8 Unity3D editor window with an example game scene	17
Figure 9 MiddleVR configuration tool	18
Figure 10 Autodesk Showcase user interface	19
Figure 11 Fold.it game	20
Figure 12 PST - Unity3D communication flow with UIVA and with MiddleVR	22
Figure 13 3D interaction devices used in this project	23
Figure 14 Microsoft Kinect and the PST live motion tracking	24
Figure 15 Side-by-side configuration error in UIVA	24
Figure 16 PST - Autodesk Showcase architectural design	26
Figure 17 Protein model	28
Figure 18 3D interaction devices representatives	29
Figure 19 User interface in the protein folding demo application	29
Figure 20 Backbone packing in fold.it	30
Figure 21 Backbone packing in the protein folding demo application	30
Figure 22 Protein movement according to the 3D interaction device	31
Figure 23 World and local space in the protein folding demo application	31
Figure 24 Side-chains movement according to the 3D interaction device	32
Figure 25 Stereoscopic 3D image in the C-Station that displays left and right image at the same time	32
Figure 26 Stereoscopic camera that represents both eyes and the screen in the demo application	33
Figure 27 Protein folding demo application run in the C-Station and being played by a user	34
Figure 28 Protein folding demo application flow diagram	34
Figure 29 Joystick interaction in the 3D interaction mode	35
Figure 30 Mouse interaction in the mouse interaction mode	36
Table 1 Section and research question relation table	15
Table 2 The Must Have functionality list for the protein folding demo application	27
Table 3 Section and the Must Have functionality relation table	28
Table 4 Protein 3D model description	28

Summary

Both the keyboard and the 2D mouse are commonly used input devices for computers. Unfortunately, these input methods come across as being unintuitive when the user has to interact with 3D data. The 2D mouse only has 2 degrees of freedom, whereas 3D interaction requires 6 degrees of freedom. To overcome this restriction, 3D motion tracking is used and allows the user to virtually grab and interact with the 3D data. With 3D motion tracking, the user will be able to interact with the 3D data in an intuitive way than the traditional interaction methods. An example of a system that uses 3D motion tracking is the C-Station, which is a desktop VR station developed by PS-Tech in Amsterdam.

The current demo applications that PS-Tech has for the C-Station are still lacking in some areas. Their graphics quality is low and looks unprofessional. They are not representing a real-life scenario and also the activities that the user can do in the old demo applications are too simple. These issues cause the old demo applications to be unappealing and do not fully encourage people to try the C-Station. Therefore, the technology development department in PS-Tech initiated a project to develop a new demo application for the C-Station.

The approach for this project consists of three main phases. In the analysis phase, the required information to utilize the technologies is analyzed and the possible demo application concepts are collected and discussed with the company. In the integration phase, the environments to develop the new demo application are integrated with PS-Tech tracking system so the outgoing tracker data can be received. The final phase is the demo application development whereby the final concept is selected and developed in an iterative cycle of design, implement, test, report, collect feedback and update activities. Research activities are also performed throughout the project to obtain the required information. The result of these activities is a protein folding demo application for the C-Station.

The protein folding demo application solves the issues in the old demo applications. High quality graphics was one of the main focuses in the project, so the protein folding demo application has better graphics and looks more professional. Protein folding is a real-life scenario and one of the important activities in biology. The activities that the user can do in the demo application are based on fold.it which is a PC game to learn protein folding, and the selected activities from the game are not too simple to do. Moreover, the user can obtain basic knowledge in protein folding with the new demo application. The protein folding demo application is, hopefully, more appealing and fully encourage people to try the C-Station.

In conclusion, the goal defined in the beginning of this project were achieved. The required deliverables were fulfilled, submitted and approved by the company. Moreover, valuable experience and professional knowledge obtained makes this a valuable project.

Glossary

Terms	Definition
2D	2 dimensional space, commonly called length and width.
3D	3 dimensional space, the physical universe.
4D	3 dimensional space added with time element as the fourth dimension.
3D interaction	A human-machine interaction in real physical space and/or virtual space representation simulated in the computer. Human can perform actions to the machine by using any input devices that detects the 3D position and 3D orientation of the action. These actions will be projected onto the virtual world and will perform similar actions.
3D interaction devices	Rigid objects detected by the PST to interact with virtual 3D objects in client application.
3D motion tracking	A technology to recognize an object and follow its orientation and position in a defined space.
3D visualization	A technique for creating 3D images or animations to communicate a message.
API	Application Program Interfaces used in order to enable communication between software components.
Colocation or colocated interaction	With the combination of the PST in the C-Station, the movement of the on-screen 3D objects will match the movement of 3D interaction devices that the user has on their hand.
C-Station	A complete desktop VR system developed by PS-Tech for an intuitive analysis and presentation of 3D and 4D data.
CWI	Centrum voor Wiskunde en Informatica (Center for Mathematics and Computer Science).
Development environment	An application that is used to develop the new demo application in this project.
Library	A collection of implemented behavior and can be used by multiple programs that have no connection with each other.
Middleware	A computer software that connects software components or people and their applications.
MoSCoW	A priority checklist with Must, Should, Could and Would Not have category.
Optical tracking systems	3D motion tracking technology that makes use of optical sensors (i.e. cameras) to estimate object poses.
PC	Personal Computer
Plugin	A software component that adds specific feature to any software applications.

PST	Personal Space Tracker, a 3D optical tracking system developed by PS-Tech that allows the user to track the position and orientation of 3D interaction devices in 3D and in real-time.
SDK	Software Development Kit, a set of software development tools that allows for the creation or customization of applications.
Stereoscopy	A technique for creating and enhancing the illusion of depth in an image or 3D imaging.
UIVA	Unity3D Indie VRPN adapter, a socket based middleware that bridges any VR devices with Unity3D through VRPN.
Unity3D	A multi-platform game engine to develop games for web player, PC, consoles and mobile devices. Any Unity3D project can import this package into their folder and utilize the assets.
Unity3D package	An Unity3D feature to transfer game assets such as library files, scripts, game objects, etc.
VR	Virtual Reality, a computer simulated environment that simulates real world into an imaginary world.
VRPN	Virtual Reality Peripheral Network, a system that provides an unified interface to input devices, such as mouse, joystick and VR devices.
Wrapper	A thin layer of code which translates an existing library into compatible formats.

1. Introduction

From the beginning of their existence, computers have mainly relied on the keyboard and the 2D mouse as input devices. The functionalities provided by computers are accessible to the user by mapping them to button presses and 2D movement of the mouse. However, the movement of the 2D mouse is restricted to the 2D desktop, which makes this input device is not always the most intuitive method for creating and manipulating 3D data.

The technology that can be used to overcome the restrictions is the 3D motion tracking. With this technology, the user will be able to virtually grab and interact with the 3D data in an intuitive way than the traditional interaction methods. An example of a system that uses 3D motion tracking is the C-Station, which is a desktop VR station developed by PS-Tech in Amsterdam.

The current demo applications that PS-Tech has for the C-Station are lacking in some areas. These issues causes the demo applications to be unappealing and do not fully encourage people to try the C-Station. Therefore, the technology development department in PS-Tech initiated a project to develop a new demo application for the C-Station.

This report is organized as follows: Chapter 2 introduces the company and products used in this project. Chapter 3 describes the assignment and introduces the project phases. Chapter 4 outlines the research questions. Chapter 5 until Chapter 8 covers the activities in the project phases. These chapters also covers the encountered issues and the selected solution. Chapter 9 describes the final demo application. Chapter 10 covers the conclusions and recommendations. The final chapters in this report consists of personal evaluation, references and appendices for related documents, such as the internship survey, the Project Plan, the project schedule and the demo application functionality list.

2. Company and products

This chapter introduces the company and the related company products for this project.

2.1. Company

Amsterdam-based PS-Tech was founded in 2005 as a spin off from the CWI (Dutch national research organization for mathematics and computer science). The company employs nine employees and is situated in the center of Amsterdam, the Netherlands.

Figure 1 below describes the departments organized in PS-Tech. The technology development department is the department that initiated this project. This department is responsible for developing, improving and altering current technologies used in the company. They are also responsible for identifying and evaluating new technologies in the market.

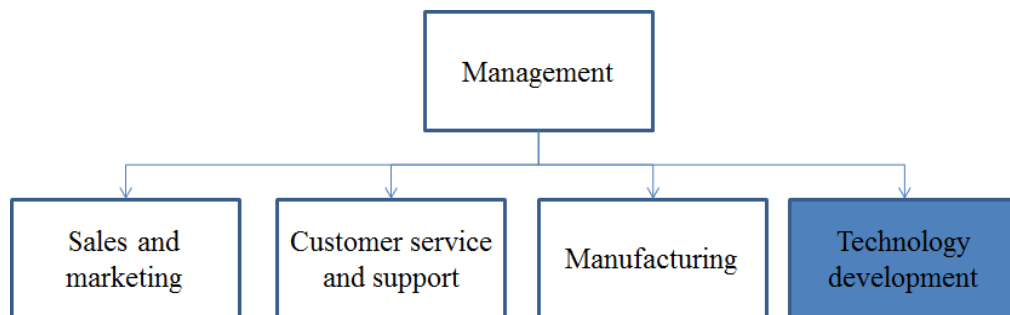


Figure 1 PS-Tech organogram

PS-Tech has two main areas of expertise:

1. Professional 3D visualization and 3D interaction
2. Optical tracking and measurement systems for medical and industrial applications.

Their products can be found globally in research departments, museums, hospitals, industrial design, training and simulations industries.

2.2. Company products

This section introduces the PST and the C-Station that are important in this project.

2.2.1. PST

The PST is a complete tracking system that can track objects in 3D using two cameras and infrared lighting. Rigid objects attached with retro-reflective markers can be tracked by the PST (see Figure 2) and these objects are generally referred to as 3D interaction devices in this report. The PST can be connected to a PC or a laptop directly, or through a network switch or hub. Once they are connected, the user can track the 3D interaction devices via the PST client software (see Figure 3). This tracking system can be used as a standalone machine or combined with the C-Station which will be described in the next section.



Figure 2 PST with 3D interaction devices

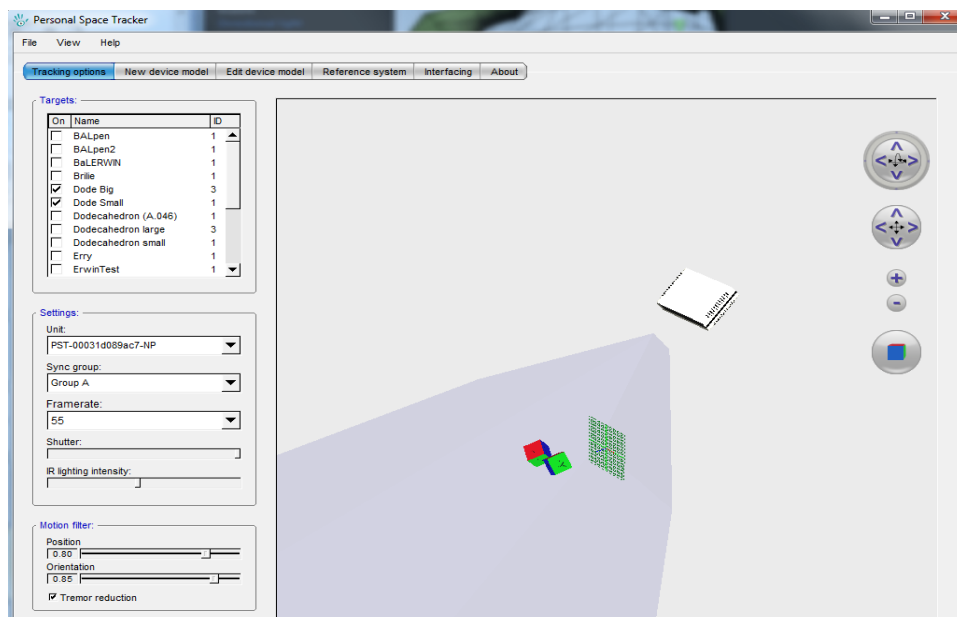


Figure 3 PST software is tracking 3D interaction devices

2.2.2. C-Station

One of the company products that was briefly mentioned in the Introduction is the C-Station (see Figure 4). This machine is used for 3D or 4D data visualization and analysis. The C-Station consists of: a PST, an adjustable mirror, a computer and a stereoscopic LCD screen.

The C-Station is working on the following principles: the mirror is placed in front of the user at an angle with the LCD screen above the mirror. The user looks at the screen via the mirror and this effectively projects the image underneath the mirror – this reflection is called the virtual focus plane (VFP). The PST is aimed at this space so the user can manipulate the VR environment using 3D interaction objects which are held in this space. This result is called colocated interaction which gives the user an idea of directly manipulating the VR 3D objects.

With the colocated interaction, the movement of the on-screen 3D objects will match the movement of the 3D interaction devices detected by the PST. The stereoscopic LCD screen is capable of showing stereoscopy view to the user. Whenever the user brings their hands into the same environment as the VR 3D objects, the visual image won't be interrupted. The user will see the whole object and they will be able to virtually grab and interact with the object by using both hands.

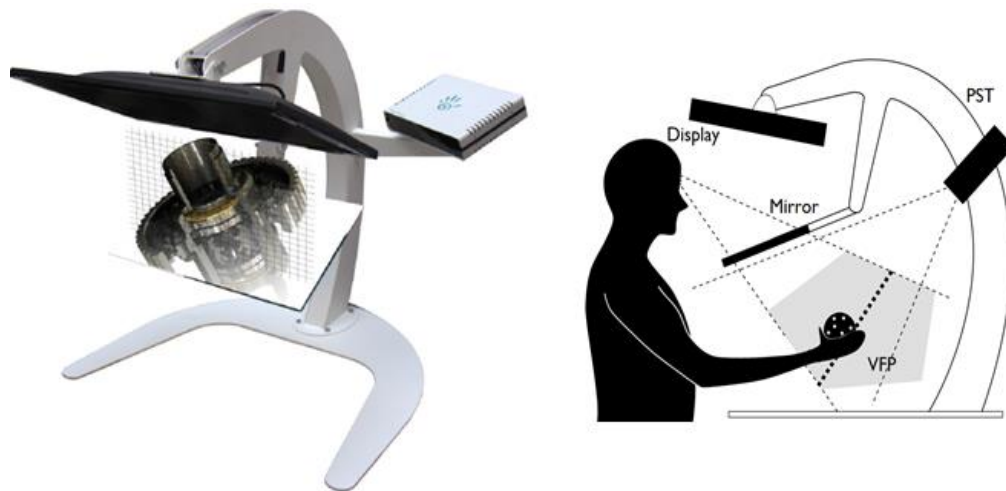


Figure 4 C-Station and colocated interaction

One of the applications for the C-Station is Vesalius3D. With the combination of colocated interaction in the C-Station, this application supports an intuitive visualization and analysis of the 3D and 4D medical imaging data (see Figure 5) such as MRI and CT data. Vesalius3D allows the user to have a direct 3D interaction with 3D and 4D medical images, for example the MRI data can be moved and orientated using one hand while the other hand measure the data.

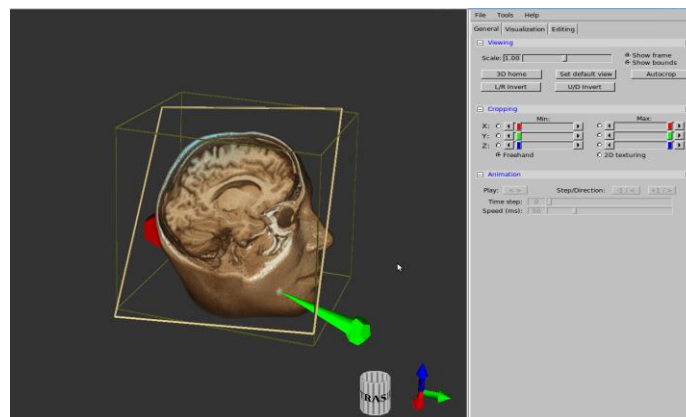


Figure 5 Vesalius3D with the 3D MRI brain data

3. Assignment overview

This chapter describes the current situation in the company, the issues that triggered the assignment, the assignment definition and the initial requirements. The Project Plan and project phases are also introduced in this chapter.

3.1. Current situation

Using a 2D screen, a mouse and a keyboard to do any complex 3D interaction can be challenging. It often requires the user to use unintuitive interaction methods to overcome the lack of degrees of freedom provided by the 2D mouse. The user cannot easily explore the 3D data by using traditional interaction devices.

The way the C-station approaches 3D visualization and 3D interaction is different from the traditional way of computer interaction. With the combination of the two handed interaction, the data is visualized at the location where the user's hands are and the C-Station allows the user to experience an intuitive interaction. Furthermore, the user still has an access to traditional interaction devices for additional functions such as click on-screen button. A number of demo applications have been developed to allow people to experience these advantages.

3.2. Issues

Unfortunately, PS-Tech is not satisfied with the old demo applications for the C-Station that they have. The old demo applications are lacking in some areas, such as:

1. Their graphics quality is low and looks unprofessional.
2. They do not represent a real-life scenario. Moreover, the activities that the user can do in the old demo applications are too simple.

These conditions cause the old demo applications to be less appealing and do not fully encourage people to try the C-Station.

3.3. Assignment definition and initial requirements

Due to the current issues described in the previous section, the technology development department at PS-Tech initiated an assignment to develop a new demo application for the C-Station that can solve the issues. The demo application could be anything, ranging from games to medical applications. Most importantly, the demo application should highlight the advantage of an intuitive 3D interaction and 3D visualization using the C-Station over traditional interaction tools.

The initial requirements of this assignment are listed as follows:

1. The new demo application must produce better quality in graphics and looks more professional than the old demo application. Therefore, the graphic quality must become one of the main focuses in the new demo application.
2. Unity3D or Autodesk Showcase can be used as the development environment and the selected environment will determine the programming language used to develop the new demo application.

3. The new demo application must implement a real-life scenario and also the activities that can be performed by the user are not too simple. By utilizing the 3D visualization and 3D interaction feature of the C-Station, the demo application should allow the user to easily perform activities that are difficult in reality. Moreover, the user will be able to obtain useful information from the new demo application.
4. Two handed interaction must be supported in the new demo application. The user must be able to interact with the 3D objects using both hands and perform different activities.
5. The colocation parameters of the C-Station must be used in the new demo application. Therefore, while the new demo application is running in the C-Station, the on-screen 3D objects will match with the 3D interaction devices held by the user.
6. The new demo application must be easy to maintain and extend over time.

3.4. Project Plan and project phases

The Project Plan is one of the required documents in the graduation project and was constructed in the first two weeks. The Project Plan consists of: project constraints, project deliverables, expected risks and people involved in this graduation project such as the project leader and the company tutor. Please refer to Appendix II for the approved Project Plan. With the approval from the company tutor for the Project Plan in week 2, the research questions are constructed to obtain the required information throughout the graduation project. These research questions will be described in Chapter 4.

The Project Plan also describes the initial project phases for this graduation project. These phases were made to anticipate the possible risks described in Section 2.8 of the Project Plan. However, this initial plan was re-arranged in week 4 because some activities took longer to finish (see Appendix III for the duration of the new development phases). The new project phases are listed as follows:

- Introduction phase that consists of several activities such as gather the initial requirements, set up the required technologies and write the Project Plan.
- Analysis phase that consists of several activities such as analyze the required technologies and search for the possible demo application concepts.
- The third phase is a combination of design, implementation and test activities for the PST and Unity3D integration.
- PST and Autodesk Showcase integration phase that has similar activities as the previous integration phase.
- Demo application development phase that consists of several activities such as select the final demo application concept, specify the functionalities, design, implement and test the functionalities. An iterative cycle of test, deliver, collect feedback, and update are also scheduled for this phase.
- Round-up phase is the final phase to create the required documents.

Details for the activities during the analysis phase until the demo application development phase will be explained in Chapter 5 onwards. As described in the list above, the demo application development phase accompanied with an iterative cycle of test, deliver, collect feedback, and update activities. This iterative cycle is based on the agile methodology (see Figure 6). With this methodology, unpredictable situations can be responded quickly through regular work rhythm and

a potentially shippable product can be presented on each deliverance. This methodology will ensure to develop the desired end product.

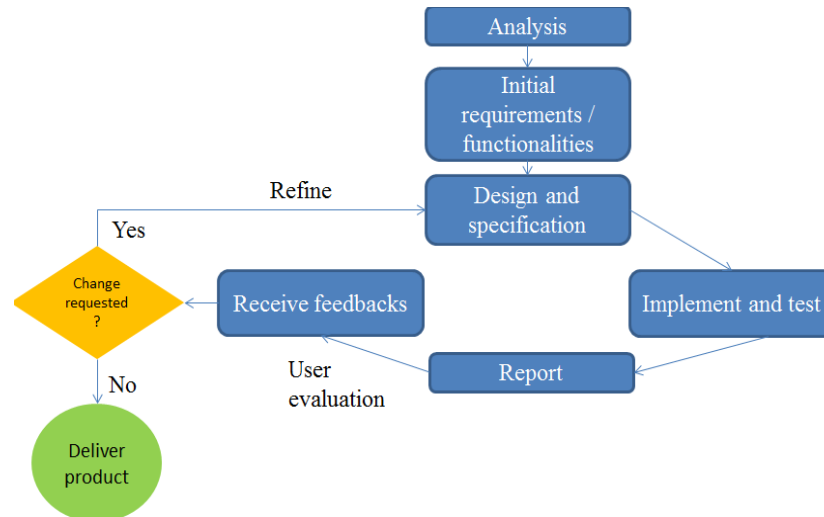


Figure 6 Methodology

The agile methodology is combined with the MoSCoW technique to prioritize the functionalities for the new demo application. Every additional requirements from the company are important, but these will be prioritized in order to deliver the best and the most immediate results.

4. Research

Due to the unfamiliarity with some technologies in the beginning of this graduation project, a research is needed in order to obtain the correct information for utilizing the technologies. This activity was performed not only in the beginning but also throughout this graduation project.

This chapter consists of the research questions and the resources. Chapter 5 onwards will describe the answers for these research questions and the reason behind the selected answer.

4.1. Research questions

These research questions were constructed based on the initial requirements in Section 3.3.

1. In order to develop an application for the C-Station, the development environments have to be integrated with the PST to receive the outgoing tracker data. Therefore, an information regarding the options on communicating the PST with different application is needed.
2. Section 3.3 mentioned Unity3D and Autodesk Showcase as the possible environment to develop the new demo application. Due to the unfamiliarity with both environments, a research is needed to find out what they are and how to operate their features.
 - a. The standard solution to integrate Unity3D and Autodesk Showcase with different applications is needed. Does it match with the PST communication option from the first research?
 - b. Whenever there are multiple integration solutions available, which one is the simplest solution and why?
3. Finding demo application concepts that meets the initial requirements in Section 3.3 is not simple. Therefore, an analysis on the possible demo application concept is needed.
4. An information on how to use the colocation parameters from the C-Station in the new demo application is needed. This information will be implemented and tested so the on-screen 3D object movement in the new demo application will match the movement from the user.

Table 1 below relates the section in the report that answers the research question. Each section will describe the answer and the reason behind the selected answer.

Section R.Q.	5.1 p.16	5.2 p.17	5.2.1 p.17	5.3 p.19	5.4 p.19	6. p.22	8.5.3 p.32
1.	x						
2.		x		x			
2.a.			x	x			
2.b.						x	
3.					x		
4.							x

Table 1 Section and research question relation table

4.2. Resources

There are a number of useful resources used during the research activities. The technical information regarding the PST and the C-Station were obtained from the company tutor and colleagues. Information from the internet regarding the development environments were also used and listed in References.

5. Analysis phase

This chapter covers the activities during the analysis phase. The next phase which is the PST and Unity3D integration phase will be explained in Chapter 6.

The outline of this chapter is as follows: Section 5.1 provides the options for the PST to communicate with different application. Section 5.2 covers Unity3D and its integration option. Section 5.3 covers Autodesk Showcase, the integration option and other Autodesk products that are possible as the development environment. Lastly, Section 5.4 covers the possible concepts for the new demo application.

5.1. PST communication option

PST uses a proprietary interface in their software to communicate their tracking data to the computer over the network. Any application that needs the data can communicate with the PST software by using different options, such as Software Development Kit (SDK), Virtual Reality Peripheral Network (VRPN) and DTrack emulation (see Figure 7).

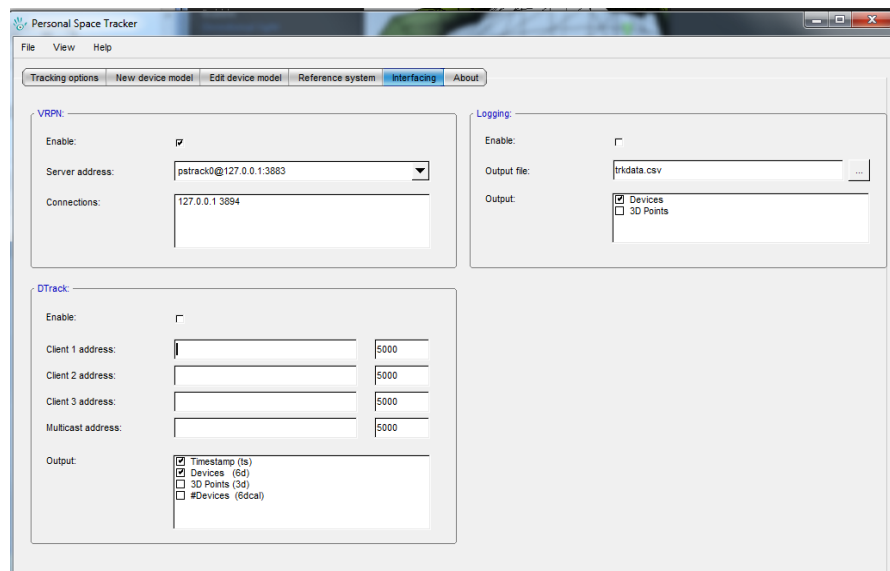


Figure 7 Interfacing option in the PST software

With an information from the company tutor, both SDK and VRPN are the commonly used communication options for the PST. This information answers the research question number 1 in Section 4.1. Description for the SDK and VRPN can be found in Section 5.1.1 and 5.1.2. With these standard options, Unity3D and/or Autodesk Showcase will be integrated with the PST by using one of these in later phases.

5.1.1. PST Software Development Kit

The PST SDK provides an interface between the PST with the client software application. The SDK includes C/C++ library, API and a documentation. By using this SDK in our software application, our application will recognize the PST and will be able to receive the outgoing tracking data. The outgoing tracking data consists of 3D interaction device name and ID, 4x4 transformation (position and orientation) matrices and the timestamp of the moment the data was recorded.

5.1.2. Virtual Reality Peripheral Network

VRPN provides the connection between applications and input devices. VRPN is a set of classes within a library and a set of servers that are designed to implement a network-transparent interface between application programs and the set of physical devices. The idea is to have a PC or other host at each VR station that controls the peripherals such as tracker, buttons and haptic device. VRPN is stable, multi-platform and supports many VR devices and applications [1].

PST already has its own VRPN server which has to be enabled in order to communicate with different applications. Therefore, the new demo application will only need to have a VRPN client to receive the outgoing tracking data from the server. The tracking data consists of 3D interaction device ID and its position and orientation in 3D space.

5.2. Unity3D

Unity3D is a multi-platform game engine developed by Unity Technology. This game engine is developed in C/C++ and supporting code written in C# or JavaScript [2]. C# was selected as the programming language to develop the new demo application in Unity3D due to personal experience. This game engine has Pro version that can set the graphics quality of the developed application to be high [3], this native feature will help to produce a high quality graphics for the new demo application. The editor window is the main window where the user create scenes, add game objects, drag and drop the scripts onto the objects to create actions and more (see Figure 8).

This information answers the research question number 2 in Section 4.1.

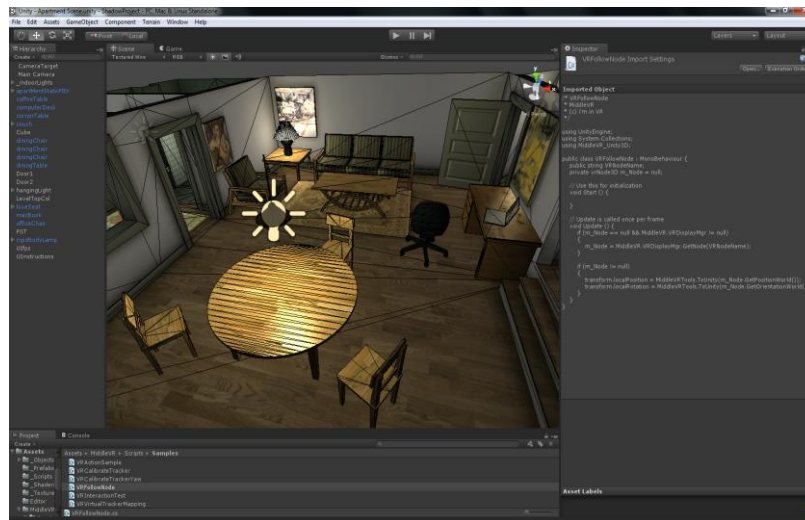


Figure 8 Unity3D editor window with an example game scene

5.2.1. Integration solutions

VRPN is the standard and the best option to integrate Unity3D with different applications. This information matches the PST communication option in Section 5.1. Therefore, Unity3D can be used to develop the new demo application for the C-Station. This information also answers the research question number 2.a in Section 4.1.

On the internet, there are a number of Unity3D – VRPN solutions developed by researchers. These solutions are available because Unity3D has no native support to create the VRPN client. From the available solutions, Unity Indie VRPN Adapter (UIVA) and MiddleVR are selected. The description for both solutions and the reason behind the selection will be described in Section 5.2.2 and Section 5.2.3.

5.2.2. Solution: Unity Indie VRPN Adapter

UIVA is a socket based middleware that bridges any VR devices with Unity3D through VRPN. UIVA is developed by Jia Wang from Human Interaction in Virtual Environments lab in Worcester Polytechnic Institute and distributed for research and non-commercial purposes [4].

This integration solution is free and using C# as the programming language. The default folder from the UIVA website consists of a documentation, a Unity3D game example and implemented VRPN client for a number of input devices such as 2D mouse and Microsoft Kinect. By following the instructions in the documentation, adding support for new devices is simple.

5.2.3. Solution: MiddleVR

MiddleVR is a VR middleware that simplifies the creation and the deployment of VR applications into Unity3D through VRPN connection. MiddleVR support the configuration for most VR devices such as CAVE, 3D TV and monitor via MiddleVR configuration tool [5] (see Figure 9).

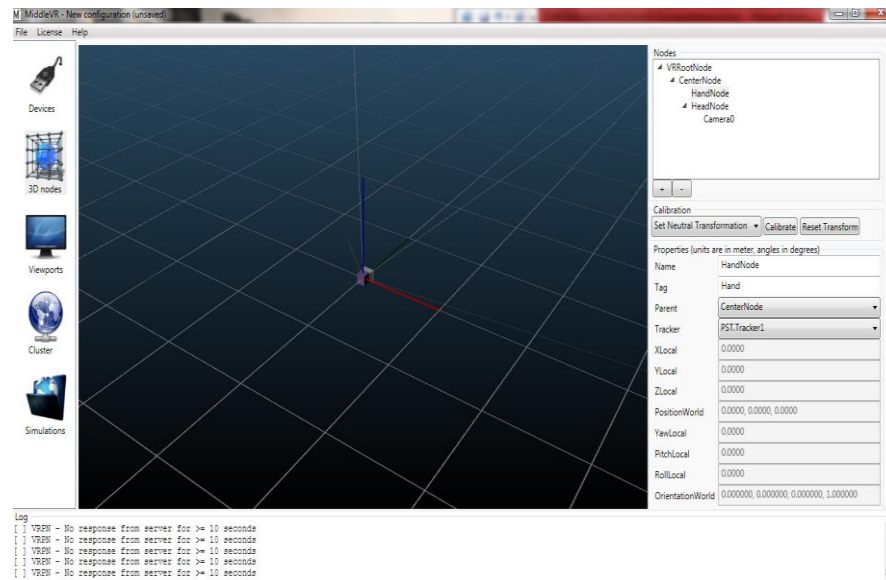


Figure 9 MiddleVR configuration tool

MiddleVR installation folder consists of an Unity3D package that contains scripts to call any configuration file into the Unity3D application. The script enables the Unity3D application to receive the tracker data from the configured VR devices.

PS-Tech was first contacted by the company who develop MiddleVR and was asked to test the compatibility of the MiddleVR with PS-Tech tracking devices. The company tutor conducted a small test with the PST and received a positive result. However, because the PST and Unity3D have been successfully integrated using UIVA at the time, MiddleVR became the alternative integration solution in the beginning of this graduation project.

5.3. Autodesk Showcase and integration option

This section answers the research question number 2 and 2.a in Section 4.1.

Showcase is a product from Autodesk that supports high quality 3D presentation and 3D visualization. This native feature will also help to produce a high quality graphics for the new demo application. Autodesk Showcase supports an interactive walk-throughs and presentations with a keyboard or a mouse which allows the user to evaluate and design the 3D models in real-time [6].

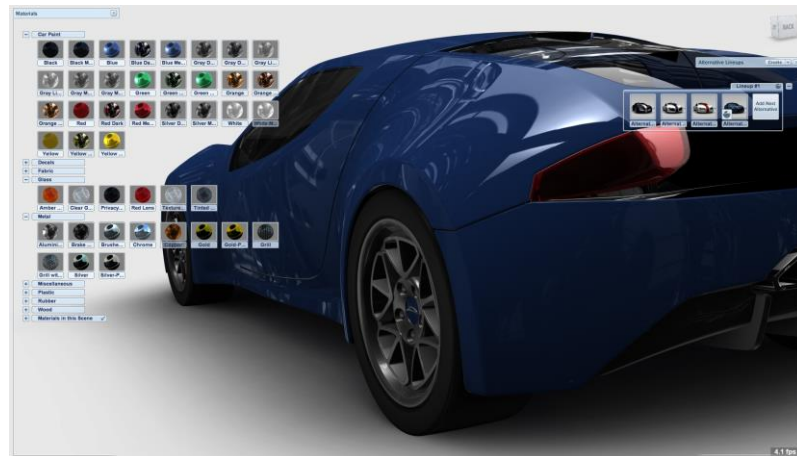


Figure 10 Autodesk Showcase user interface

Autodesk Showcase has online and offline SDK documentation for developers. The documentation covers the tutorial to customize the appearance and behavior of Showcase using HTML, JavaScript and Python. Python was selected as the programming language in developing the demo application due to personal experience. From the documentation, it is very clear that SDK is the standard way to communicate with different applications. This integration information matches the PST communication option in Section 5.1. Therefore, Autodesk Showcase can be used to develop the new demo application for the C-Station.

5.3.1. Other Autodesk products

Due to the strong personal preference in C# than Python, other Autodesk products that uses C# SDK were proposed. The company tutor approved this idea in order to see various development environment that can be used to develop applications for the C-Station. The proposed products are: Autodesk 3DSMax which is a 3D modeling application and Autodesk Design Review which is similar to Autodesk Showcase. Unfortunately, both products are lacking in a complete and clear documentation. Therefore, these proposal were terminated in the middle of the graduation project.

5.4. Demo application concepts

The information in this section answers the research question number 3 in Section 4.1.

There are two types of concepts for the new demo application; the Unity3D concept and the Autodesk Showcase concept. Section 5.2 and Section 5.3 mentioned that Unity3D and Autodesk Showcase have different intended purposes. Unity3D is intended for creating game and has more

advanced features than Autodesk Showcase. Therefore, not every demo application concept will be suitable to be developed in both environments.

The concepts gathered for Unity3D are listed as follows: organizing books in the book shelves or a drawer, driving simulation, building Lego blocks, tying the scouting knots and protein folding. The concept gathered for Autodesk Showcase was an interactive and intuitive showcase demo application concept. These concepts were proposed and discussed with the company tutor.

Unity3D gives freedom to think of unique concepts, but too much freedom is not always better. Most of the proposed concepts for Unity3D are too simple and do not give an useful information to the user. In the final discussion with the company tutor, the protein folding concept and intuitive showcase concept were selected. Both concepts will be explained in Section 5.4.1 and 5.4.2. These proposed concepts were documented into Demo application concept proposal and submitted to the company tutor.

5.4.1. Protein folding

Protein folding is a process where the protein structure assumes its functional shape naturally or using assistive devices. Proteins in our body need to have the correct folding in order to function correctly. In some cases, wrong folded proteins can create toxins which endanger our health [7]. This idea was proposed because protein folding is a real-life scenario and an important activity in biomedical. Furthermore, PS-Tech is an expert in medical solutions as mentioned in Section 2.1 so this concept fits with PS-Tech area of expertise.

An online-offline PC game called fold.it is commonly used to learn the correct protein folding [8] (see Figure 11). People without a life science, biotechnology or bioinformatics background can play the game. In fact, most of the users are not life scientist [9]. Fold.it became the main reference of the new demo application with Unity3D.

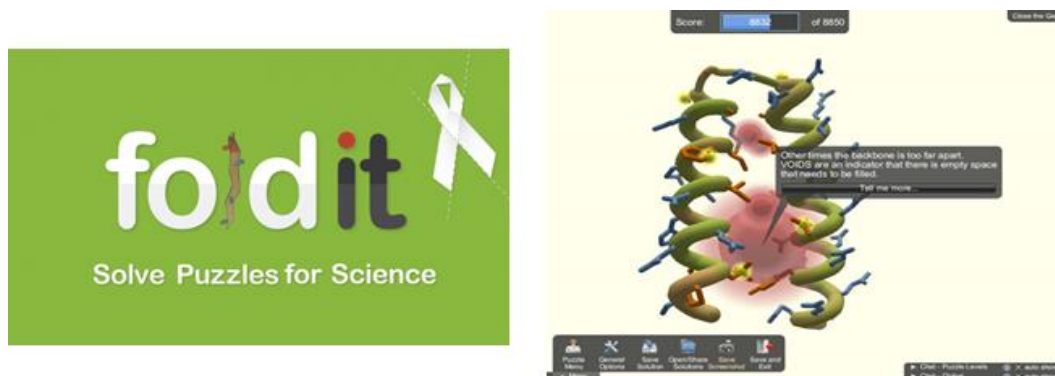


Figure 11 Fold.it game

The goal of each level in the game is to fold the protein correctly. 3D protein models are used in the game and the user have to use a 2D mouse to interact with it. However, using a 2D screen and 2D mouse makes the interaction unintuitive. If this 2D mouse interaction are transformed into the 3D interaction in the C-Station, the user will be able to fold the protein easily. The user will be able to use both hands to interact with the 3D protein model and perform multiple activities at the same time. For example, one hand virtually grab and rotate the 3D object around, whereas the other hand can fold part of the protein. Moreover, the user will also be able to obtain basic knowledge about protein folding.

5.4.2. Showcase

Section 5.3 mentioned that Autodesk Showcase is supporting 3D objects presentation and showcase by using a mouse and a keyboard input. Instead of using these traditional interaction devices, the new demo application will use 3D interaction devices to interact with the 3D object. The user will be able to 'grab', translate, orientate and perform different activities to the 3D object with both hand at the same times. For example, a complete house model with doors and windows is used as the main 3D object. The user can grab it with one hand and move them around, while the other hand can open-close the door and window or move the furniture.

This idea was proposed because showcasing and presenting 3D designs in Autodesk Showcase are commonly used by designers, engineers, etc. Exploring the 3D objects will be a lot simpler with 3D interaction devices than the keyboard and the 2D mouse interaction. However, the new demo application will be targeted to the general audience to allow the m to experience the intuitive 3D objects showcase and presentation by using the C-Station.

6. PST - Unity3D integration phase: Design, implement and test

Section 5.2.1 described UIVA and MiddleVR as the selected integration solutions. However, only one solution will be used in the new demo application. This chapter describes the simplest solution out of the two and answers the research question number 2.b in Section 4.1. The next phase which is the PST and Autodesk Showcase integration phase will be explained in Chapter 7.

The outlines of this chapter is as follows: Section 6.1 covers the high level architecture design for UIVA and MiddleVR. This design informs on how the PST and Unity3D communicate with each other using both solutions, which part is already there and which part needs to be implemented. To get the correct high level view, the design was first made with UIVA and MiddleVR documentation as the reference and then discussed with the company tutor. Lastly, Section 6.2 covers the implementation and test activities for the PST – Unity3D integration using UIVA and MiddleVR.

The technical procedures and evaluations of the integration solutions in this phase are combined into the PST integration document and submitted to the company tutor.

6.3. Design phase: Architectural design

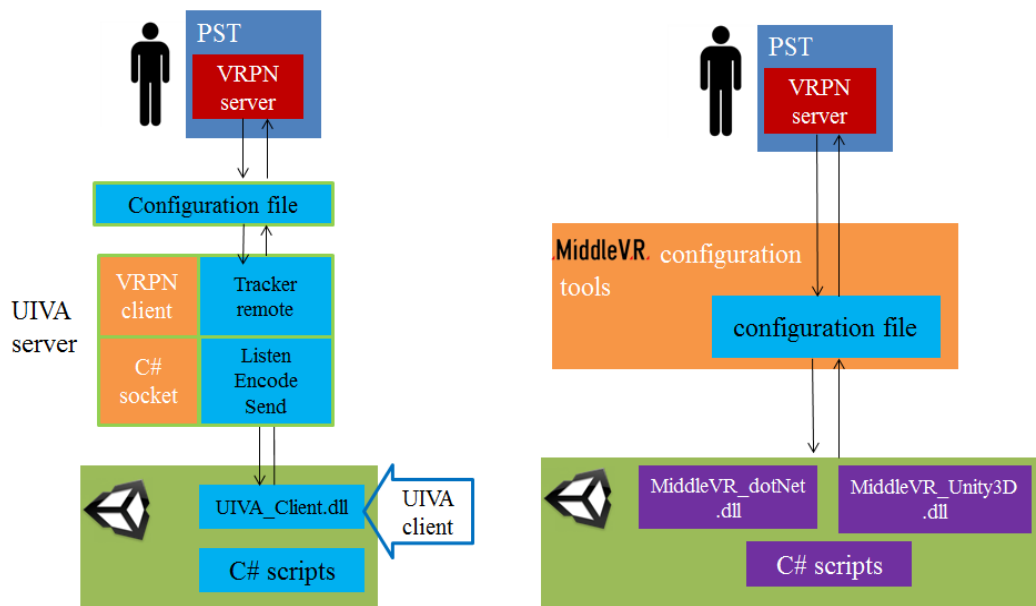


Figure 12 PST - Unity3D communication flow with UIVA and with MiddleVR

Color	Description
	Part(s) to be implemented.

Figure 12 above shows that MiddleVR is the simplest solution out of the two. In order to integrate Unity3D with the PST via UIVA, four parts need to be implemented. These parts are: the configuration file, the UIVA server, the UIVA client and C# scripts. This is a contrast from MiddleVR that only needs to create a description file for the PST via MiddleVR configuration tool. Next is the description of the communication flow for each solution.

In UIVA, the server side has one or more VRPN objects for each device supported. These objects requests tracking data from the VRPN server on demand of the client side. Whereas the client side is a DLL file that resides in the Unity3D project and exposes tracking data polling functions to the C# scripts from devices that employ it. Therefore, UIVA server is a middle layer between the PST VRPN and Unity3D, which acts as a server for Unity3D and also as a client for the VRPN server. Whenever the connection is erected, Unity3D will poll the newest tracking data from UIVA server, which will in turn poll newest tracking data from PST VRPN server.

In MiddleVR, the description file in MiddleVR configuration tool will become the VRPN client and will receive the tracking data from the PST VRPN server. Required libraries and C# scripts to test the connection are already available from the MiddleVR installation folder.

6.4. Implementation and test phase

The following activities were performed based on UIVA and MiddleVR documentations and also based on the architectural design in the previous section. Even though the documentation from UIVA provides a step-by-step tutorials, there was still an issue where UIVA failed to start. By tracking the errors using a tracing tool, the issue was solved and the UIVA could run properly. The implementation and test activities for UIVA, as well as the encountered issue and the solution will be explained in Section 6.2.1. Section 6.2.2 describes the integration activities by using MiddleVR.

Both solutions were able to integrate the PST with Unity3D. Out of the two solutions, MiddleVR is proven to be a lot simpler and easier than UIVA. Therefore, MiddleVR is used as the main integration solution for the PST with Unity3D and UIVA became the alternative solution.

To test both integration solutions, two 3D interaction devices were used (see Figure 13). These devices are also used while developing the new demo application in later phase. These 3D interaction devices are the common 3D interaction devices used in PS-Tech and will act as the user's hands in the virtual world.



Figure 13 3D interaction devices used in this project

6.4.1. Integration by using UIVA

There are three steps to add the PST as the supported devices in UIVA. The first step is to edit the UIVA server script with a description for the PST. The description is similar to the Kinect description in the script because both applications (Kinect and the PST) has similar way of work, which is a 3D motion tracking (see Figure 14). The next step is to edit the UIVA client and then built it into a library file. The last step is to add the configuration file with the PST VRPN server identification.



Figure 14 Microsoft Kinect and the PST live motion tracking

In order to test the validity of the PST tracker data received in Unity3D, two C# scripts have been created. The first script is instantiating the VRPN client and the second script is transforming the tracker data into the position and the orientation of the attached object. In the Unity3D editor, two cubes and a sphere were added. The first script is attached to the sphere, each cube received the second script and then ran the editor. When the editor is running, the cubes move around the screen following the 3D interaction devices movement. This result shows that this integration was a success.

One encountered issue was a side-by-side configuration error that causes the UIVA client failed to start. To determine what cause the error, sxstrace tool that generates a detailed log file was used [10]. The log file below shows that the MSVCR90.dll version 9.0.21022.8 is missing (see Figure 15). The dll is only available in Visual Studio 2008, whereas Visual Studio 2010 was used to build both UIVA server and client. Unfortunately, this software requirement is not mentioned in the documentation. To easily solve this error, Visual Studio 2008 is installed on the PC and then finally the UIVA client receives the outgoing tracker data from the UIVA server.

```
ERROR: Cannot resolve reference
Microsoft.VC90.CRT,processorArchitecture="x86",
publickeyToken="1fc8b3b9a1e18e3b",type="win32",
version="9.0.21022.8".
```

Figure 15 side-by-side configuration error in UIVA

6.4.2. Integration by using MiddleVR

By following the MiddleVR tutorial, the PST can be easily configured as one of the supported device. Section 5.2.3 described that the MiddleVR installation folder provides a script that can call any custom configuration file into Unity3D application. This script has been edited to call the configuration file for the PST for test purposes. MiddleVR installation folder also provides a script that can be attached to an object in Unity3D editor, and the object will follow the movement from the configured devices. No new scripts are necessary to use the MiddleVR in our application.

The test activities follows the same scenario as the UIVA: Two cubes and a sphere were added into the Unity3D editor. The first script is attached to the sphere, each cube received the second script and then ran the editor. When the editor is running, the cubes move around the screen following the 3D interaction devices movement that are detected by the PST. This result shows that this integration was a success.

7. PST – Autodesk Showcase integration phase: Design, implement and test

This chapter covers the design, implementation and test activities for integrating Autodesk Showcase with the PST. This phase also answers the research question number 2.b constructed in Section 4.1. The next phase which is the demo application development phase will be explained in Chapter 8.

The outlines of this chapter is as follows: Section 7.1 covers the high level architecture design for the Autodesk Showcase. This design informs on how the PST and Autodesk Showcase communicate with each other, which part is already there and which part needs to be implemented. To get the correct high level view, the design was first made with Autodesk Showcase documentation as the reference and then discussed it with the company tutor. Lastly, Section 7.2 covers the implementation and test activities for the PST – Autodesk Showcase integration solution.

7.3. Design phase: Architectural design

Section 5.1.1 mentioned that the PST SDK uses C/C++ library. Python, the selected programming language for Autodesk Showcase, cannot call functions in C/C++ libraries directly. Therefore, the existing PST SDK library needs to be wrapped for usage in Python (see Figure 16). There are number of options to create Python wrapper and with feedback from the company tutor, the ctypes feature is selected because ctypes is the simplest solution for Python to use any shared library [11].

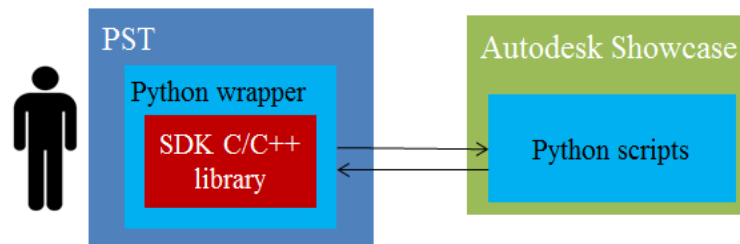


Figure 16 PST - Autodesk Showcase architectural design

Color	Description
	Part(s) to be implemented

7.4. Implementation and test phase

By following the Autodesk Showcase documentation and added with an assistance from the company tutor, a wrapper script for PST SDK library has been implemented. This script will receive the outgoing tracker data from the PST and send it to any Python scripts that needs it. Autodesk Showcase provides an example script that prints out the 2D mouse position. This script has been edited to print the received tracking data from the PST. With the edited script, Autodesk Showcase prints out the 3D interaction device ID and its position in 3D coordinates. The printed data has been compared with the result from the VRPN standalone test program that prints the similar data. This comparison showed similar results which means this integration was a success.

8. Demo application development phase: Design, implement and test

This chapter covers the activities during the new demo application development phase. The outlines of this chapter is as follows: Section 8.1 covers the selected concept for the new demo application. Section 8.2 introduces the Must Have functionality list. Section 8.3 covers the 3D objects and the user interface designs. Finally, Section 8.4 covers the implementation and test activities for the Must Have functionalities. The final demo application will be explained in Chapter 9.

8.1. Final demo application concept

From the previous two integration activities, the PST is compatible with both Unity3D and Autodesk Showcase. These environments can be used to develop the new demo application for the C-Station. Section 5.4 described two possible demo application concepts for Unity3D and Autodesk Showcase. However, in order to deliver the desired demo application within the remaining days, only one development environment will be used.

After a discussion with the company tutor, Unity3D became the main development environment. With Unity3D, numerous applications for the C-Station can be developed. Whereas Autodesk Showcase is limited to presenting 3D objects. The new demo application will be a good starting point for the company to use Unity3D for future applications. Therefore, the protein folding concept with Unity3D is the main demo application in this graduation project. This decision meets the initial requirements number 2 and 3 in Section 3.3.

8.2. Must Have functionality list

Table 2 below outlines the Must Have functionality list for the protein folding demo application. This list was constructed based on the initial requirements in Section 3.3.

	Must have
1.	One level of fold.it game to be solved by the user
2.	Use similar protein model used in the fold.it game
3.	The user can rotate different part of the model
4.	The movement of the on-screen 3D objects match the movement from the 3D interaction devices held by the user
5.	Stereoscopic 3D

Table 2 The Must Have functionality list for the protein folding demo application

As mentioned in Section 3.4, every additional requirements from the company are important, however these will be prioritized to deliver the best and the most immediate results (please refer to Appendix IV for the complete functionality list). The result of the implemented Should Have and Could Have will be described in Chapter 9.

The activities to implement the Must have functionalities, encountered issue and the solution for the issues will be explained in the next few sections. Table 3 below outlines the section in the document that covers the functionalities.

Section	8.3.	8.4.	8.5.1.	8.5.2.	8.5.3.
Func.	p.28	p.28	p.29	p.30	p.32
1	x		x		
2		x			
3		x			
4				x	x
5					x

Table 3 Section and the Must Have functionality relation table

8.3. Backbone packing level

Section 5.4 introduced the fold.it game as the main reference for the new demo application. There are a number of levels in the game and only one level will be selected to meet the Must Have number 1. The Backbone Packing level is selected because the activities that needs to be performed are not too simple or too difficult. This decision also meets the initial requirements number 3 in Section 3.3.

8.4. 3D objects and user interface design

The design activities in this section are performed in an iterative cycle of design, report, collect feedback and update with the company to ensure the quality of the delivered graphics. This cycle is helped to make the graphics more appealing and more professional than the old demo applications.

To meet the Must Have number 2, 3DS Max modeling software is used to create the similar protein model as in fold.it. Unity3D has no native support to create custom 3D objects. The final design consists of a backbone (main protein) as the parent and few side-chains as the children. To meet the Must Have number 3, each part of the protein is a separate object. This design will allow an independent movement for each part. Figure 17 below shows the final model used in the folding protein demo application.

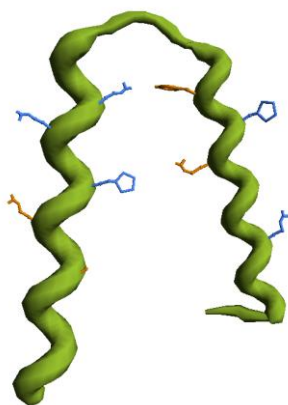


Figure 17 Protein model

Table 4 Protein 3D model description

Color	Description
Green	Backbone
Dark blue and light orange	Side-chain

The protein folding demo application uses two sphere to represent the 3D interaction devices detected by the PST (see Figure 18). The big sphere (identified as the protein detector) will be used to select, translate and orientate the whole protein. While the small sphere (identified as the side-chain detector) will be used to select and orientate the side-chains. These objects meets the initial requirements number 4 in Section 3.3.

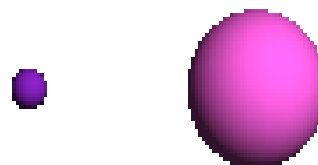


Figure 18 3D interaction devices representatives

The protein folding demo application uses the similar user interface as fold.it, so it was not necessary to design a custom user interface. The user interface consists of plain white background, a number of on-screen instructions and score result (see Figure 19).

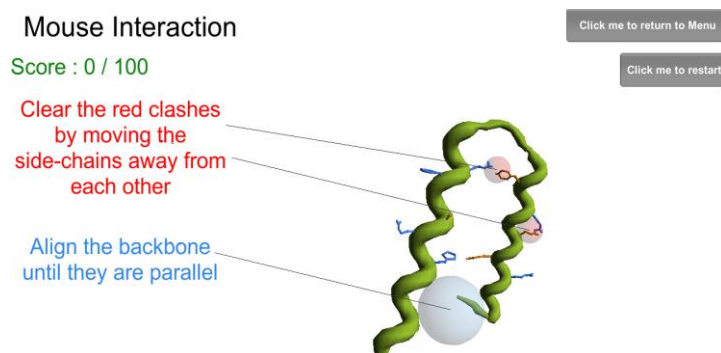


Figure 19 User interface in the protein folding demo application

8.5. Implementation and test for the Must Have functionalities

This section covers the implementation and test activities for the Must Have functionalities. The decisions, the encountered issues and the solutions will also be explained. The activities in this section were performed in an iterative cycle of implement, test, report, collect feedback and update with the company to ensure the desired functionalities are met.

8.5.1. Issue detectors in the protein folding demo application

Section 8.3 mentioned the Backbone Packing level from fold.it. There are two issues to be solved by the user in this level: The gap between the backbones are too wide and the gap between the side-chains are too small. In fold.it, these issues are indicated with pulsating translucent red spheres (see Figure 20).

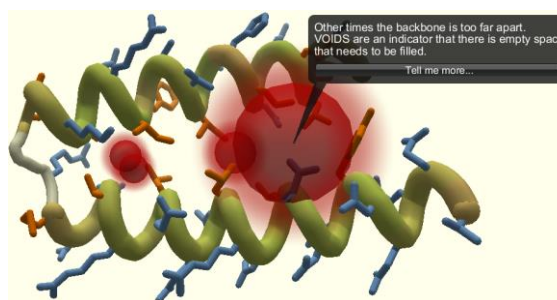


Figure 20 Backbone packing in fold.it

These issue detectors have to be implemented into the protein folding demo application to meet the Must Have number 1. The simplest solution is to use the trigger components which is a commonly used component for the collision detection between two or more objects in Unity3D [12]. The triggers were added to the side-chains and a red sphere will be visible whenever they are colliding with each other. This red spheres informs the user to drag the side-chains away from each other. Trigger was also added to the backbone and a blue sphere will be visible whenever the backbone triggers are not colliding with each other. This blue sphere informs the user to drag the backbone closer to each other. The result of this implementation is shown in Figure 21 below.

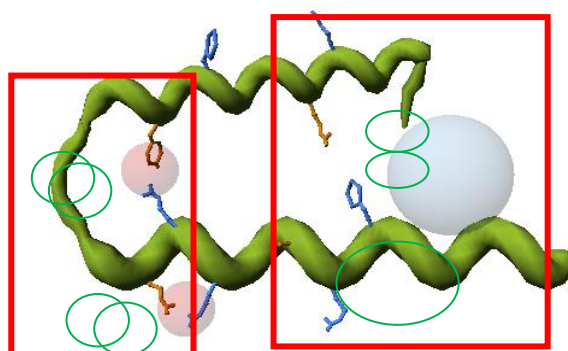


Figure 21 Backbone packing in the protein folding demo application

8.5.2. Interaction with the protein object

One of the most important features of the C-Station is the direct interaction with the 3D data. The user holds 3D interaction devices on their hand to interact with the on-screen data. This interaction is supported in the protein folding demo application. The information in this section meets the initial requirement number 5 in Section 3.3 and the Must Have number 3 and 4.

Section 8.4 mentioned that the protein detector and the side-chain detector are representing the user's hands in the virtual world to interact with the protein 3D object. The movement of these detectors corresponds with the 3D interaction devices in the user's hands. The integration scripts from Chapter 6 are used again in this activity to match the movement of the detectors with the 3D interaction devices.

In order to have an interaction between the protein detector with the backbone, a trigger component is added to both objects. Whenever the trigger collides, the protein detector can translate and orientate the backbone (see Figure 22).

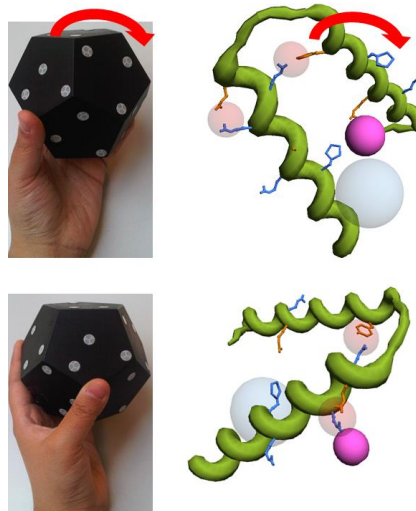


Figure 22 Protein movement according to the 3D interaction device

The side-chain detector in the protein folding demo application is used to orientate the side-chains, however this is not as simple as the protein detector. There are two types of space in Unity3D; world and local space. Every object in the protein folding demo application has their own local space and will not always match the world space (see Figure 23). One of the biggest challenges in this graduation project is to orientate the side-chains realistically in their local space, whereas the side-chain detector also has its own local space. If the side-chain is rotated based on the local space of the side-chain detector, it will be jumpy and unrealistic.

To achieve the desired side-chains orientation, the side-chain detector position and orientation in the world space are converted into the local space of the selected side-chain. Therefore, any movement by the side-chain detector will affect the side-chain orientation directly and correctly (see Figure 24).

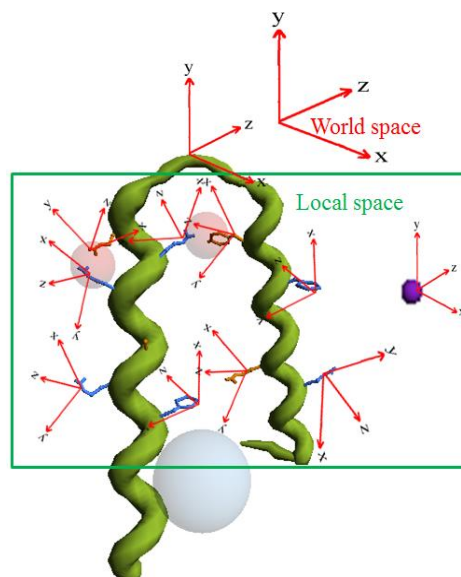


Figure 23 World and local space in the protein folding demo application

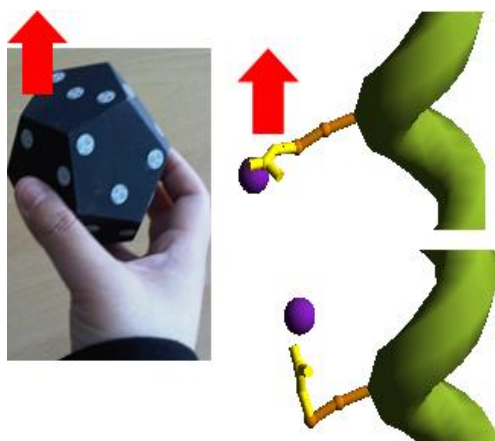


Figure 24 Side-chains movement according to the 3D interaction device

8.5.3. Stereoscopic 3D

There are two main types of stereoscopy; active and passive stereoscopy. Active stereoscopy displays the left-right images alternatively on the screen and the user needs to wear an active shutter glasses to experience the 3D effect. Whereas the passive stereoscopy is achieved by displaying the left-right images at the same time and the user needs to wear a 3D glasses with polarized lenses [13]. Passive stereoscopy is used in the C-Station (see Figure 25).

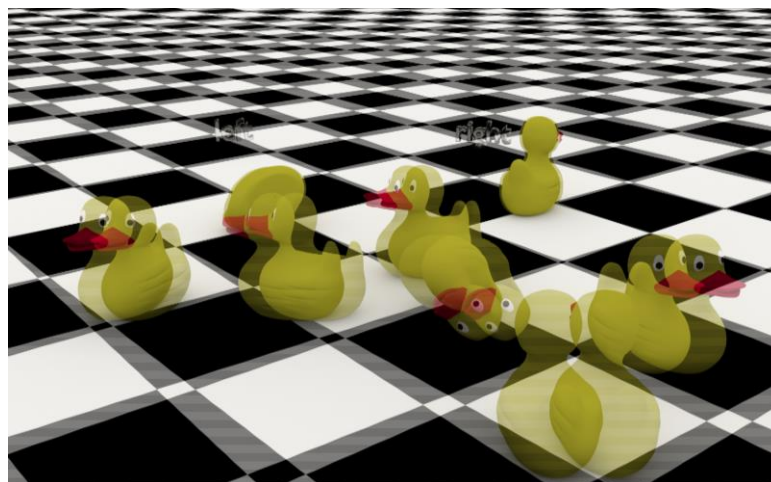


Figure 25 Stereoscopic 3D image in the C-Station that displays left and right image at the same time

Unfortunately, Unity3D does not have native support for stereoscopic view. Therefore, the protein folding demo application needs different approach to meet the Must Have number 5. There were a number of approaches considered such as use the Stereoscopic plugin, configure the graphical output directly from the NVIDIA driver or use the MiddleVR tool.

There are a number of active stereoscopic plugin on the Unity3D website, but none of them are free and none have good ratings or good reviews from users. By configuring the NVIDIA driver on the PC, the developer will lose their control over the 3D visualization and some graphic effects will not work properly [14]. With these results, both approaches were discontinued and then MiddleVR is used in the protein folding demo application.

MiddleVR supports both active and passive stereoscopy. One of the requirements to use the passive stereoscopy is having professional graphics card such as NVidia Quadro cards or ATI FireGL pro cards [15]. Fortunately, the C-Station uses one of the required graphic cards on the PC. Another requirement is to have a stereo camera and a screen created in the MiddleVR configuration tools (see Figure 25). These objects have to be positioned correctly to get the best stereoscopic effect.

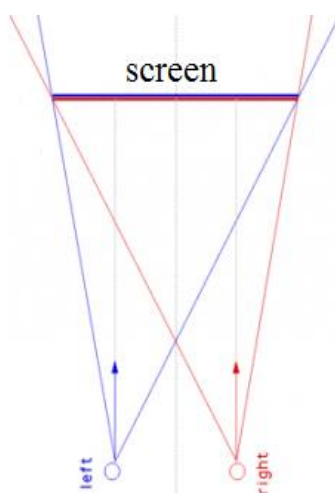


Figure 26 Stereoscopic camera that represents both eyes and the screen in the demo application

With feedback from the company tutor, the best solution to position these objects is to use the colocation parameters from the C-Station. These parameters are created in the C-Station utility software and consists of the position and the orientation for the virtual screen and the head (camera) in 3D space. This implementation was a challenge because both MiddleVR and the C-Station configuration files are needed in the protein folding demo application. In the end, the C-Station utility software has been adjusted to automatically create and position the stereo camera, the virtual screen and then saved them as the MiddleVR configuration file format.

With this adjusted utility software, the protein folding demo application features a stereoscopic effect while running in the C-Station. The result of the adjusted utility software also support the interaction described in Section 8.5.2. The movements of the on-screen 3D objects will match the movement of the 3D interaction devices held by the user while running in the C-Station. This implementation also meets the initial requirement number 5 in Section 3.3.

9. Protein folding demo application

The result of the demo application development phase explained in the previous chapter is a new protein folding demo application for the C-Station (see Figure 27). In this chapter, the flow diagram and other features from the final protein folding demo application will be covered. These features are the successfully implemented and tested Should Have and Could Have functionalities in appendix IV. The final version of the protein folding demo application has been reported, evaluated and approved by the company tutor.



Figure 27 Protein folding demo application run in the C-Station and being played by a user

9.1. Flow diagram

Figure 28 below shows the flow diagram of the protein folding demo application in the C-Station.

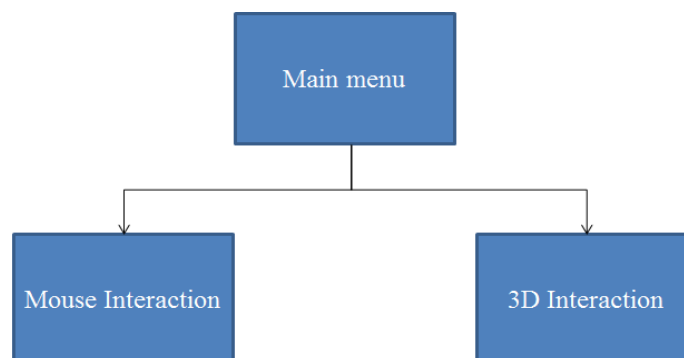


Figure 28 Protein folding demo application flow diagram

The final demo application consists of three main scenes. The main menu is the default scene and the user can select the interaction mode they wanted to try:

1. Mouse interaction
2. 3D interaction

Both the mouse interaction and the 3D interaction mode use the same protein model and have the same issues to solve. By experiencing these interaction modes, hopefully the user will experience more intuitive interaction by using 3D visualization and 3D interaction of the C-Station than the 2D mouse. Due to the fact that the user can perform different activities with both hands at once in 3D interaction mode, whereas they can only perform one activity at a time with the 2D mouse.

This demo application flow was first implemented for the Laval Virtual Conference at 20 – 22 March 2013, an annual conference on VR and converging technologies in Laval, France [16]. Due to the small amount of time to fully implement all functionalities for this conference, the protein folding demo application was a lot simpler with only side-chains issues to solve. This simple version could only be played in a controlled demo environment. The current final version of the protein folding demo application can be played by anyone and not in a controlled environment.

9.2. 3D interaction and mouse interaction mode

In the 3D interaction mode, the user can interact with the protein and solve the issues by using 3D interaction devices. The user can also use joystick interaction (see Figure 29) that accompanies the implementation in Section 8.5.2. This interaction is similar to the joystick interaction which is currently used in other PS-Tech applications.

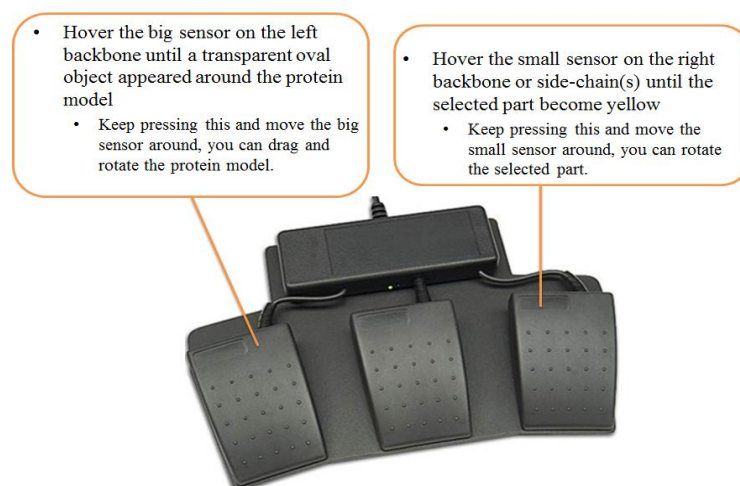


Figure 29 Joystick interaction in the 3D interaction mode

Figure 30 below shows the result for the mouse interaction mode. The user can use left button and middle scroll of the 2D mouse to interact with the protein and solve the issues. This interaction is similar to the 2D mouse interaction in the fold.it.

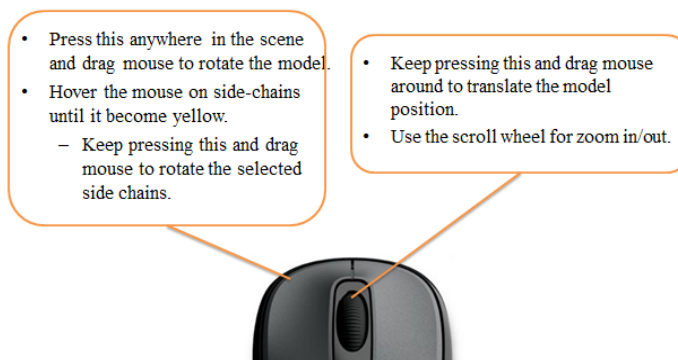


Figure 30 2D Mouse interaction in the mouse interaction mode

9.3. Protein folding Unity3D custom package

Throughout the demo application development phase, the codes behind the protein folding demo application are separated into reusable scripts. These reusable scripts allows the protein folding demo application to be easy to maintain and extensible in the future. This feature meets the initial requirement number 6 in Section 3.3.

The reusable scripts and the integration solutions implemented in Chapter 6 are also exported into a Unity3D custom package. Any Unity3D application that wants to receive the outgoing tracker data from the PST added with the joystick interaction, can import this custom package into their scene and start to use those features.

10. Conclusions and recommendations

After a thorough and careful research, analysis, implementation and test activities throughout this graduation project, a new demo application for the C-Station has been successfully developed. An iterative cycle of design, test, deliver, collect feedback, and update activities were also performed during the demo development phase to ensure the delivered quality.

The protein folding demo application can solve the current issues that appears in old demo applications. First, the graphics quality is better and looks more professional than the old demo applications. Secondly, protein folding is a real-life scenario and one of the important activities in biology. The activities that the user can do in the protein folding demo application are based on fold.it, which is a PC game to learn protein folding, and the selected activities from the game are not too simple to do. Moreover, the user can obtain basic knowledge in protein folding with the new demo application. The protein folding demo application is, hopefully, more appealing and fully encourage people to try the C-Station.

The protein folding demo application highlights the advantages of an intuitive 3D interaction and 3D visualization of the C-Station over traditional interaction tools. The 2D mouse interaction and the 3D interaction are supported in the protein folding demo application. Therefore, the user will be able to experience more intuitive approach of the C-Station compare to the 2D mouse. The 2D mouse only allow one interaction at a time, whereas the 3D interaction mode is supporting the two handed interaction that allow the user to perform different activities at once.

The protein folding demo application uses the colocation parameters from the C-Station. Therefore, the on-screen 3D objects movements are matched with the movement from the 3D interaction devices detected by the PST. This parameters also enables the protein folding demo application to have the correct stereoscopy view. Moreover, the code behind are separated into reusable codes which makes the protein folding demo application to be maintainable and extensible over time.

In conclusion, the goal and initial requirements defined in the beginning of this report were achieved. The required deliverables were fulfilled, submitted and approved by the company tutor.

Throughout this graduation project, there are a number of aspects that can be improved in the future. The following recommendations are offered to PS-Tech:

1. Unity3D is a good development environment to work with and also compatible with the C-Station. It is recommended for PS-Tech to use this environment to develop more applications in the future.
2. The demo application uses MiddleVR to support the colocated interaction and the 3D stereoscopic feature in the C-Station. MiddleVR is a third party application and if the application is down, the maintenance can cost time and money. Therefore, a different solution is recommended. For example, PS-Tech can create their custom Unity3D plugin. This solution can be controlled and configured exactly to work with the C-Station.
3. Section 6.2 described the dodecahedron 3D interaction devices used for the demo application. As the 3D objects is getting smaller and more complex, it will be easier and more convenience to interact by using smaller 3D interaction devices.
4. Currently, only one level is implemented in the demo application. It would be better to implement more levels for variations.
5. Chapter 7 described that PST can be integrated with Autodesk Showcase and it would better if this integration can be extended with a demo application.

Personal Evaluation

This graduation project is one of the most valuable professional experiences I have ever obtained during my academic years in the Netherlands. Throughout the graduation project, there are a lot of valuable lessons that I will always remember and will apply whenever it's possible.

First of all, this graduation project adds the VR specialist as one of the potential career path.

Secondly, my time management has improved. I learned more on prioritizing work and to separate work time with my personal time. I also learned to be more aware of changes, whether it comes from me or the environment. Therefore, constant progress checking is really important. Fortunately, staff members were available most of the times and ready to guide me.

However, the most important lesson I learned throughout this graduation project is bravery. An intern cannot be afraid to ask questions, we have to accept the fact that we may not know everything and know nothing at all. Therefore, I figured out the best way to accumulate questions at once then ask colleagues as to not bother them whenever something was confusing to me.

If I was given an opportunity to restart this graduation project, I would do my research and analysis activities more careful and more thorough than before. I will spend more time to find various answers and compare them. Therefore, I would obtain more reliable and better results for my next activities.

References

- [1] VRPN website that describes their history and tutorials, accessed 31 January 2013. <http://www.cs.unc.edu/Research/vrpn/index.html>.
- [2] Unity3D game engine Wikipedia, accessed 29 January 2013. [http://en.wikipedia.org/wiki/Unity3D_\(game_engine\)](http://en.wikipedia.org/wiki/Unity3D_(game_engine)).
- [3] Unity3D graphical quality setting, accessed 29 January 2013. <http://docs.unity3d.com/Documentation/Components/class-QualitySettings.html>
- [4] UIVA website that contains details and place to download UIVA, accessed 31 January 2013. <http://web.cs.wpi.edu/~gogo/hive/UIVA/>.
- [5] MiddleVR configuration tool for Unity3D, accessed 15 April 2013. <http://www.imin-vr.com/middlevr-for-Unity3D/>.
- [6] Autodesk Showcase, accessed 29 January 2013. <http://www.autodesk.com/products/showcase/overview>.
- [7] Science reference: protein folding definition, accessed 15 February 2013. http://www.sciencedaily.com/articles/p/protein_folding.htm.
- [8] Fold.it wiki, accessed 15 February 2013. http://foldit.wikia.com/wiki/Foldit_Wiki.
- [9] Fold.it website and their game can be downloaded here, accessed 15 February 2013. <http://fold.it/portal/>.
- [10] Side-by-side configuration error, accessed 3 February 2013. <http://www.codeproject.com/Articles/43681/Side-by-Side-Configuration-Incorrect>
- [11] C/C++ wrapping for Python, accessed 14 February 2013. <http://intermediate-and-advanced-software-carpentry.readthedocs.org/en/latest/c++-wrapping.html>.
- [12] Trigger components in Unity3D game, accessed 27 February 2013. <http://technology.blurst.com/Unity3D-physics-trigger-collider-examples/>.
- [13] Active and passive 3D differences, accessed 4 March 2013. <http://gcn.com/Articles/2013/03/06/Differences-active-passive-3D.aspx?Page=2>.
- [14] Stereoscopic3D in Unity 3D, accessed 4 March 2013. http://wiki.etc.cmu.edu/unity3d/index.php/Stereoscopic_3D
- [15] MiddleVR stereoscopy online documentation, accessed 15 April 2013. <http://www.imin-vr.com/doc/1.2/ch04s05.html>.
- [16] Laval Virtual annual conference, accessed 19 March 2013. <http://www.laval-virtual.org/>.

Appendices

Appendix I: Internship Survey

1. Describe the problem analysis:

The C-Station is a desktop Virtual Reality station, which is used for 3D/4D data visualization and analysis. Using stereoscopy, the data is presented to a user directly in 3D and by using a 3D optical motion tracker the user is also able to interact with the data in three dimensions.

The way the C-station approaches 3D visualization and interaction significantly differs from the traditional way of computer interaction using a mouse and keyboard. Doing complex 3D interaction using traditional tools can be challenging and often requires the user to use unintuitive interaction methods to overcome the lack of degrees of freedom provided by 2D input devices.

PS-Tech is looking for demos that make use of the unique 3D interaction and visualization capabilities of the C-station. These demos can be showed at conferences and trade shows to highlight the added value of intuitive 3D interaction and visualization with the C-station. PS-Tech is looking for these demos to be developed in a high-level 3D development platform, such as Vizard, Virtools or 3DVIA Studio.

2. Describe the graduation assignment.

The objective of the graduation assignment is the development of a 3D visualization and interaction demo that shows the power of the C-station over traditional interaction tools. This demo has to be created in a high-level 3D development environment such as 3DVIA Studio.

The student is free, in collaboration with PS-Tech, to come up with a concept for a demo. This can be anything, ranging from games to medical applications. The only strict requirement for this assignment is the presence of a 3D interaction component that shows the power of the C-station, as compared to traditional interaction.

Preferably, the student also performs a small study, using the developed demo, where the performance of the C-station is shown in comparison to traditional input methods.

3. What is the research component of this assignment?



In analysis phase, I will mostly do the interviews and requirements gathering. Then move on to specifications design (such as architectural design and GUI). In last phase, I will implement the analysed and designed system, and test them according to test plan.

4. What are the methods and tools?

The intern will have access to a C-station, containing a 3D workstation computer, PST-55 optical tracking system and stereoscopic monitor. For development of the software a development environment will be available (e.g. 3DVIA studio).

5. How and by whom will you be guided by the company?

During the course of the internship, one employee of PS-Tech will be assigned as the day-to-day internship coordinator and will generally be the person that answers questions and provides suggestions. Because PS-Tech has many years of experience with the development of virtual and augmented reality applications there is a great deal of expertise available on which the intern can rely.

6. What fields of Study play an important factor in realizing the graduation assignment?

Software engineering (requirements, design, and implementation), Virtual Reality, Computer Graphics, Human Computer Interaction, Mathematics (linear algebra, geometry, etc)

Appendix II: Project Plan

Approved and delivered: February 5, 2013.

Table of Contents

1. Introduction	43
2. Project Statement	43
2.1. Formal client	43
2.2. Project leader	43
2.3. Brief current situation	43
2.4. Project justification	43
2.5. Project product	43
2.6. Project deliverables and non-deliverables	43
2.7. Project constraints	44
2.8. Project risks	44
3. Project Phasing	45
4. Management plan	46
4.1. Money	46
4.2. Skills	46
4.3. Quality	47
4.4. Information	47
4.5. Time	48
4.6. Organisation	49

1. Introduction

In order to show the advantage of the C-Station, a desktop Virtual Reality station developed by PS-Tech Amsterdam, over any traditional interaction tools, a 3D visualization and interaction demo application is needed. This document consists of project statement in chapter 2, project phasing in chapter 3 and management plan in chapter 4.

2. Project Statement

2.1. Formal client

The formal client in this project is Mr. Wim Koomen, as the technology development specialist in PS-Tech Amsterdam.

2.2. Project leader

The project leader is Amiril Rahmi Faradini, student of Fontys ICT Hogeschool Eindhoven.

2.3. Brief current situation

The way the C-Station approaches 3D visualization and interaction significantly differs from the traditional way of computer interaction using a mouse and keyboard. Doing complex 3D interaction using traditional tools can be challenging and often requires the user to use unintuitive interaction methods to overcome the lack of degrees of freedom provided by 2D input devices.

2.4. Project justification

PS-Tech is looking for demo application that make use of the unique 3D interaction and visualization capabilities of the C-Station. These demos can be showed at conferences and trade shows to highlight the added value of intuitive 3D interaction and visualization with the C-Station.

2.5. Project product

A demo application that shows the advantage of the C-Station over traditional interaction tools, such as 2D mouse.

2.6. Project deliverables and non-deliverables

Deliverables	Non-deliverables
Application proposal and list of requirements	Maintenance
Description of PST integration report ¹	Training
Evaluation of used software(s)	Application updates
Working demo application	

¹ Please refer to Project Phasing for detail.

2.7. Project constraints

Type	Description
Environment	PC, C-Station containing a 3D workstation computer, PST-55 optical tracking system and stereoscopic monitor.
	Use C# .NET and/or Python.
	Use Unity3D and/or Showcase for the environment development.
Time	18 – 19 weeks.

2.8. Project risks

Risk name	Alternative plan	Uncovered exception
Requirements misunderstanding. Requirements were only partially known at the start.	Make additional agreement with the client on the understanding the requirements. Obtain missed requirements and make possible improvement by testing and seek for approval before different phases.	Client cannot to provide completed requirements or make significant modification of requirements without increase necessary working time.
Insufficient time due to the requirements (and additional) requested.	Assign levels of important to the use cases by MoSCoW principle, and negotiate possible solutions with client.	Client denies reasonable project time extensions.
The application delivered still have unexpected bugs	Make proper modification based on feedbacks.	Fail to obtain necessary feedbacks.
Project leader may not cooperate in common standards which company uses.	Discussions to make an agreement.	-
Lack of technical and general support due to the staffs' schedule.	Make an early appointment and use the time wisely.	-
Clients don't give enough feedbacks.	Schedule more appointments to have a proper discussion.	-

3. Project Phasing

Activities	Description
Introduction phase	
Set up the environment	Install all the needed applications on the PC.
Communication plan	How and when the project leader and school tutor communicate during the project.
Project plan	
Analysis phase	
Brainstorm ideas and proposal	Idea(s) for the end application; will be reported, receive feedbacks and approved by the company tutor.
PST and C#	PST has C# example application in its SDK. Source: the manual and API pdf.
Unity3D Game Engine	Both are advised application by the company to be integrated with the PST.
Autodesk Showcase	<p>Unity3D is a cross platform 3D game engine, whereas Showcase is 3D presentation software. Here I will find how to work with the programs by following the online tutorials.</p> <p>I will be focusing more on Unity3D at the beginning then move onto Autodesk Showcase.</p>
Find information on VRPN	One of the possibilities for communicating the PST with other virtual environment is by using its VRPN (Virtual Reality Peripheral Network); PST has its own VRPN server and here I need to find the solution to pair it with other VRPN client.
Test UIVA with other device(s)	UIVA is one of the solutions for the VRPN connection, it is a middleware between VRPN server of any interaction device with Unity3D. Will test the software by using mouse and/or Kinect.
PST VRPN	What outputs that is given out by the VRPN server. The client will be <code>vrpn_print_devices.exe</code> which is a template from VRPN itself.
Quaternions	This is one of the outputs given by the VRPN server, I find out how a orientation is represented in quaternions.
List of requirements and functionalities of the demo application concept	This will be started after the idea has been approved.

Design, implement and test phase	
Integrate Unity3D with PST	This activity will refer to the result from the analysis phase.
Integrate Autodesk Showcase with PST	
3D objects and environment	Design and implement in the final development environment.
Demo application	Design, implement and test the demo application by using PST and the C-Station. The final deliverable will be done by the end of internship period.
Round-Up phase	
For Final report and Presentation activities will be consists of report, receive feedbacks, update and final deliverable.	
PST integration report	Description on work needed in order for the PST to integrate with other software.
Software(s) evaluation report	Personal evaluation regarding Unity3D and Showcase.

4. Management plan

4.1. Money

Due to the non-budget project, the tangible cost cannot be calculated.

Intangible Costs

Table below provides a summary of possible intangible costs associated with the development and operation of an information system.

Operational inefficiency	No face-to-face interaction
	Meeting delay
	Illness
	Long vacations
Morale	Late for work
	Ask for early leave or absent too often
	Careless about the work quality
Loss of client goodwill	Serious mistakes occurring

	Bad communications
	Fail fulfilling promises

4.2. Skills

In order to complete the project, steps for development which planned in advance are needed. Here are the skills which are important for each phase in the Project Phasing:

- Analysis

Social, forward-thinking and good communication skills are needed for interacting with other parties. Be decisive in choosing which technique to use for integrating the applications.

- Design: Good in designing a 3D environment, objects and application design.
- Implementation: Experience in programming with C#.NET or Python language, virtual reality, computer graphics, human computer interaction and mathematics.
- Test: Precise in testing the demo application, decision maker and decision taker.
- Round-up: Good report writing and presentation skills.

4.3. Quality

The quality of the deliverables can be measured by:

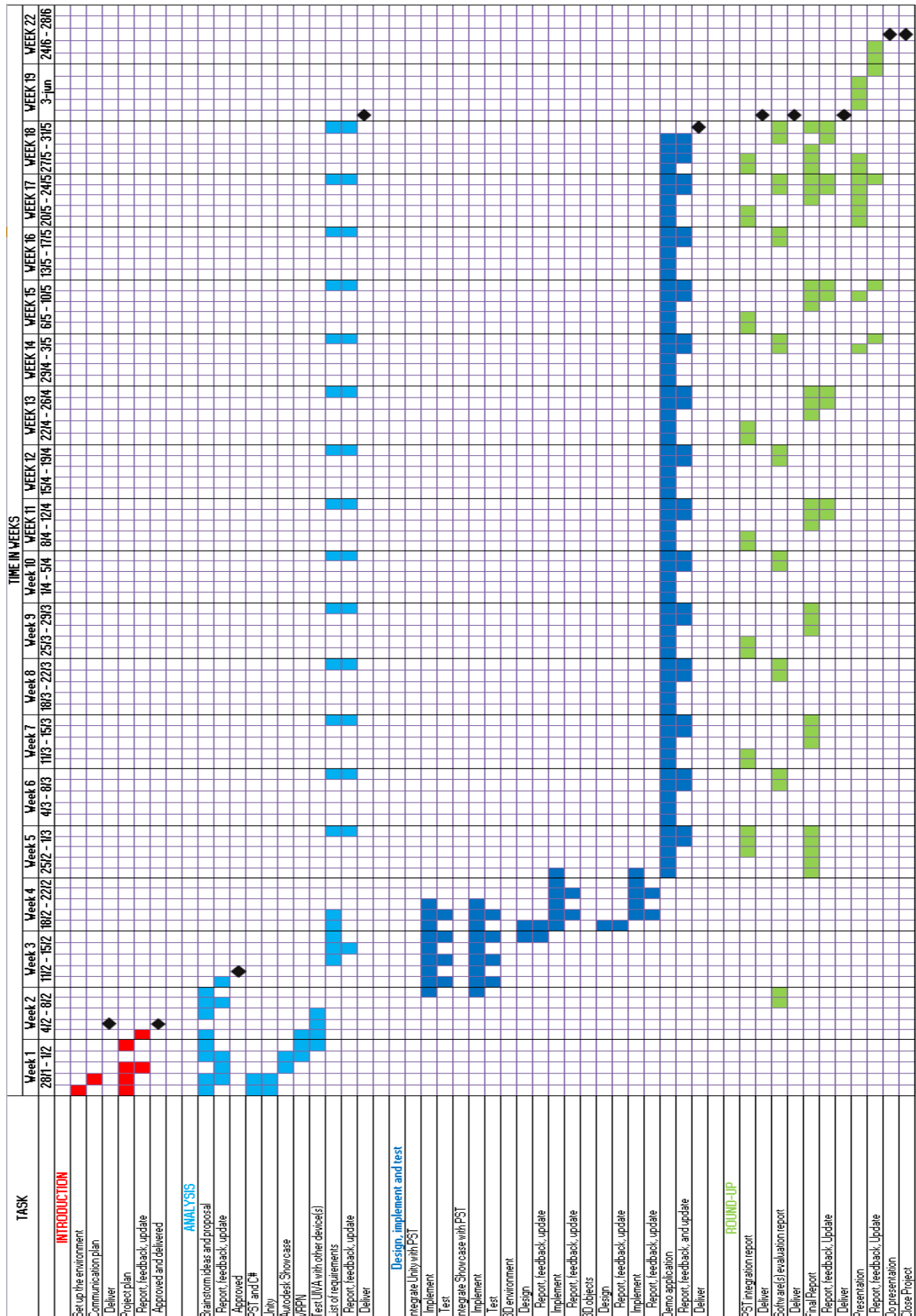
- Deliverables meets the client requirements and feedbacks during the project:
- A working, stable and maintainable demo application that justifies the advantage of the C-Station over traditional interaction tools.
- Clear and structured written documents.

4.4. Information

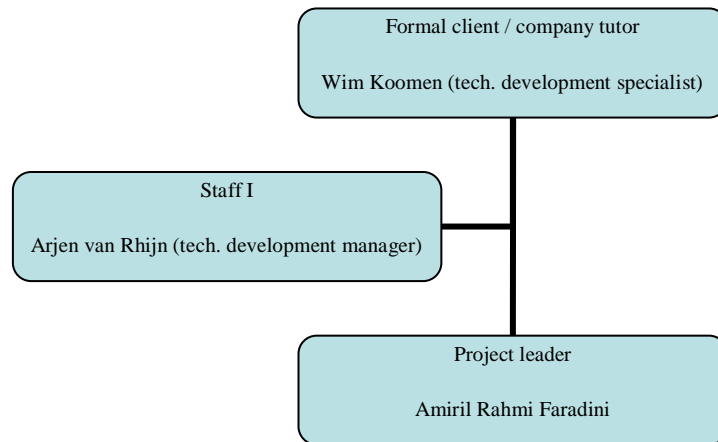
	Project Plan	Application proposal , list of requirements, PST integration report, software(s) evaluation report	Final report	Application
formal client	R, Di, A	R, Di, A	R, Di, A	R, A, Ar
project leader	Dr, S, Di, Ar	Dr, S, Di, Ar	Dr, S, Di, Ar	Dr, S
school tutor	R, Di		R, Di, A	

Legends: Dr Draw up ; Di Discuss; A Approve
S Send ; R Receive; Ar Archive

4.5. Time

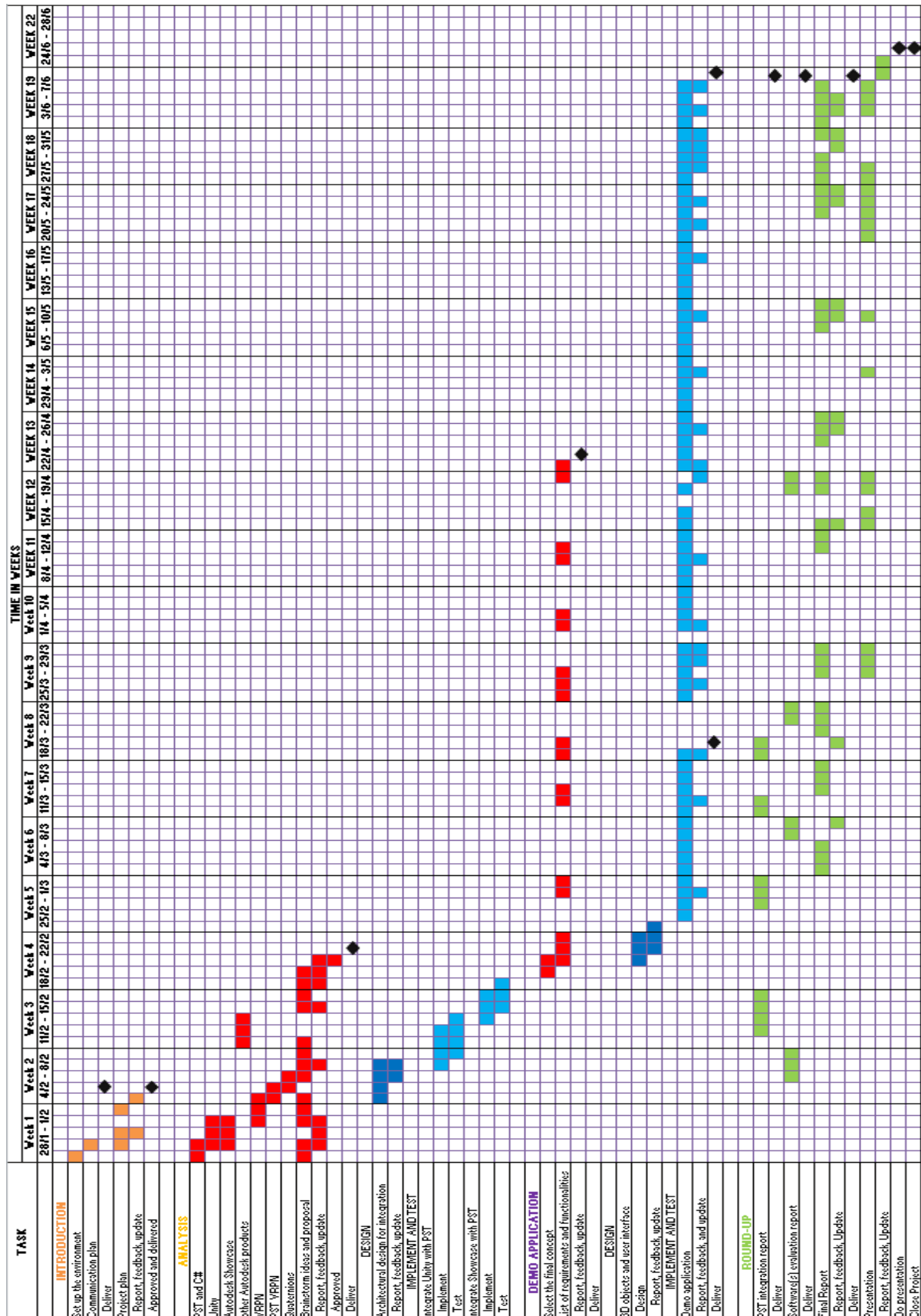


4.6. Organisation



Project leader is the decision maker, the main contact with the formal client and the schools tutor regarding the project, make an appointment with the school tutor and document handler. Company tutor and staff 1 give the general and technical guidance during the project.

Appendix III: Project schedule



Appendix IV: Protein folding demo application functionality list

	Must have		Should have
1.	One level of fold.it game to be solved by the user	6.	Use joystick for additional interaction with the protein model, e.g.: <ul style="list-style-type: none"> - Left pedal to grab - Right pedal to manipulate
2.	Use similar protein model used in the game		
3.	User can rotate different part of the model		
4.	Translation and orientation of the objects in the demo are based on the 3D interaction devices detected by the PST	7.	Configuration file
5.	Stereoscopic 3D	8.	Simple user interface
	Could have		Would not have
9.	On-screen instructions	15.	Sound effects
10.	Score	16.	Pause and save current state features
11.	Mouse mode to compare with 3D interaction mode	17.	Actual folding algorithm
12.	Restart level		
13.	Different levels implemented		
14.	Timer		