VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wānanga o te Ūpoko o te Ika a Māui

School of Engineering and Computer Science
Te Kura Mātai Pūkaha, Pūrorohiko

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Internet: office@ecs.vuw.ac.nz

Interactive 3D Visualisation of Exoplanets

Owen Bannister, 300172912

Supervisor: Stuart Marshall

Submitted in partial fulfilment of the requirements for
Bachelor of Software Engineering with Honors.

Abstract

Large amounts of information has been discovered about planets outside our solar system, and this can be accessed by anyone from the Kepler Exoplanets dataset [28] [34]. However this information is often complex and therefore not easily understood by laypeople who lack in depth knowledge of stellar objects. This is problematic as it means that the data gathered about these planets is not being used effectively to communicate to the masses. To resolve this a visualisation has been created that can convey this information in a way that interested laypeople can understand. This visualisation can be used as an information source for those wanting to increase their knowledge about the planets residing outside of our solar system. This report outlines the project carried out and includes the planning of the deliverable visualisation, its implementation, and its evaluation to discover its effectiveness at fulfilling the requirements driving its creation.

Acknowledgments

I would like to thank Dr Stuart Marshall the supervisor of this project. He provided me with endless advice and mentorship over the course of this project making it a very positive experience. Thanks also goes to the Victoria University of Wellington Human Computer Interaction Group who were always available to help and to provide me with assistance.

Contents

Figures

Chapter 1

Introduction

The Kepler Exoplanet dataset [28] [34] contains information about planets from outside of our solar system. This information was discovered by the Kepler space observatory, shown in Figure 1.1. The Kepler space observatory was launched in 2009 by NASA to discover Earth like planets orbiting other stars [14].

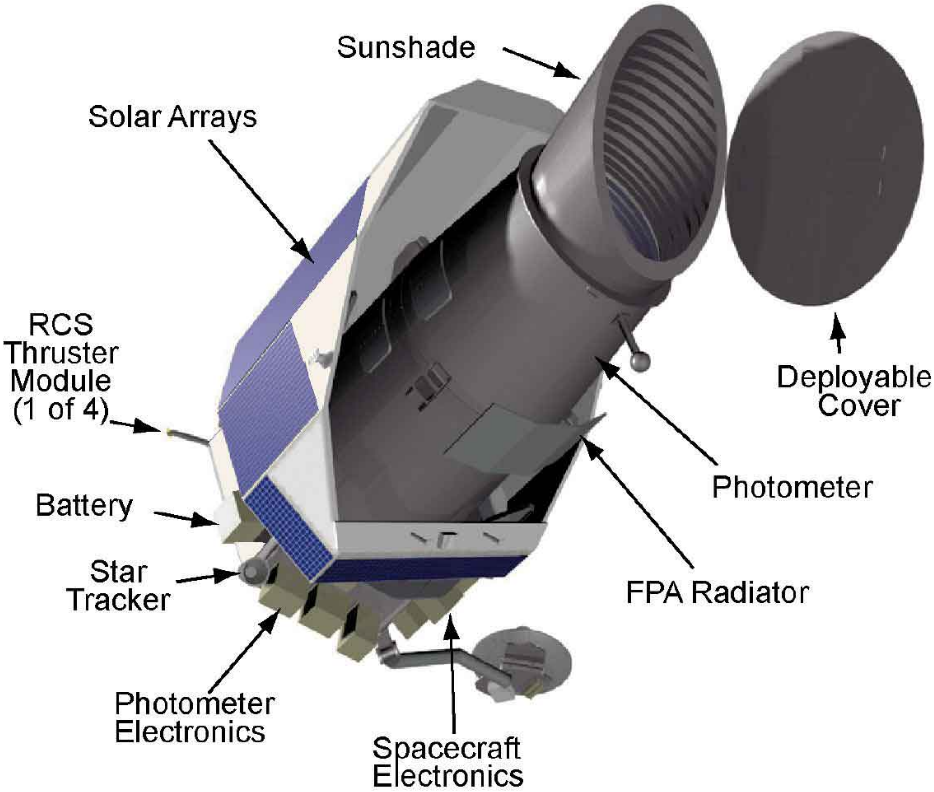

Figure 1.1: A rendering of the Kepler space observatory

This project seeks to design, implement, and evaluate an interactive 3D visualisation software system for displaying the data contained in the Kepler Exoplanet dataset using both gestures and traditional interactive methods. This deliverable is intended as a standalone 3D visualisation with two modes of interaction: keyboard & mouse or Microsoft Xbox Kinect sensor [6]. The resulting visualisation will present the information from the dataset in a way that the target users, laypeople who have an interest in astronomy, can understand and interact with.

1.1 Problem Statement

Many planets have been identified outside of our own solar system, these are called exoplanets, and are referred to interchangeably as planets and exoplanets for the remainder of this report. Information about these exoplanets is available in the Kepler Exoplanet Dataset . This project seeks to develop and evaluate an interactive 3D visualisation software system for the Kepler Exoplanets Dataset. Visualisations are valuable as they foster immersion and enjoyment, both of which result in improved learning and recall. This means that creating an effective and engaging visualisation will help convey the information in the dataset effectively to the users.

The complex nature of the information involved in this project causes a range of problems revolving around understandability and access to the data in the dataset. This project attempts to address these issues in a way that results in increased accessibility of the information contained in the dataset for laypeople. The following subsections outline these in detail.

1.1.1 Understanding the Content in the Dataset

Understanding and analysing large datasets whose size defies simplistic or trivial analysis by humans is a known issue, and one that many areas of research are attempting to address. These areas of research include data mining, data analytics, and visualisations in order to discover and highlight important features in the data so that people can more easily understand and use it [16].

Humans often rely on internal visualisation to solve problems. We create an image in our mind of a situation in order to make sense of it which allows for a faster and more comprehensive understanding [29]. The content in the dataset used for this project is made up of records of every one of the 2234 exoplanets discovered by the Kepler Mission. Each of these records contains 46 fields, shown in Figure 1.2, which makes it next to impossible for someone to internally visualise as there is so much information, especially as most of it consists of floating point numbers. This means that an external way of visualising the data is needed.

KOI	Dur	Depth	SNR	t0	t0_unc	Period	P_unc	a/R*	a/R*_unc
r/R*	r/R*_unc	b	b_unc	Rp	a	Teq	EB prob	V	FOP
N	P. Zone Class	P. Mass Class	P. Composition Class	P. Atmosphere Class	P. Habitable Class	P. Gravity (EU)	P. Esc Vel (EU)	P. Period (days)	S. Hab Zone Min (AU)
S. Hab Zone Max	P. HZD	P. HZC	P. HZA	P. HZI	P. Int ESI	P. Surf ESI	P. ESI	S. HabCat	P. Habitable
P. Hab Moon	P. Confirmed	P. Disc. Method	P. Disc. Year	S. Name	S. Constellation				

Figure 1.2: The fields of the Kepler Exoplanet dataset, the amount and variety of which makes them difficult to comprehend

This project aims to find a way to display this information visually so that users can better understand it.

1.1.2 Comprehension of Planetary Information

Much of the information regarding planets is cryptic and unintuitive, this makes it difficult to understand. Data visualisation can therefore be effective because it shifts the balance between perception and cognition to take advantage of the brain's abilities. Seeing (i.e. visual perception) which is handled by the visual cortex located in the rear of the brain, is extremely fast and efficient. We see immediately, with little effort. Thinking (i.e. cognition), which is handled primarily by the cerebral cortex in the front of the brain, is much slower and less efficient [19]. Therefore this project will find ways of visually simplifying this data so that it conveys more meaning to users.

1.1.3 Existing Solutions Lack Functionality

Existing data visualisations using this exoplanet dataset lack the ability to display sufficient detail for each exoplanet and do not fully utilise the data available. Existing solutions display only the size, temperature, and orbital information about the exoplanets. While this is useful information that informs users of important facts about the planets, it leaves a lot of potentially useful information unseen and overlooked. Examples of this include, information about types of planet, solar system information, planet similarities, and habitability. This project will therefore focus on researching, implementing, and evaluating a new interactive visualisation system that displays information contained in the dataset but not included in existing visualisation systems.

1.1.4 Effective User Interaction with Visualisation

A visualisation that solely displays information without effective methods of interaction limits the immersive qualities that keeps users engaged.

To address this, interactive visualisations emerged, generally these visualisations allow users to modify the representation of information rather than the information itself. This means allowing users to control properties of how the data is represented, be it something as simple as the layout of elements or more complex. Many mediums of interaction are possible from the mundane keyboards, mice, or touchpads to the more esoteric wired gloves, motion sensors, and gesture recognition or even a combination of devices. Existing visualisations surrounding planetary data only allow access to a small subset of the information contained within the Kepler Exoplanet Database. In addition to this most solutions do not allow interaction with any advanced means with the exception of Exo [3] which uses wired gloves to detect the gestures of of users. To address this this project incorporates a range of interactive components that allow the user to access large amounts of data. It also introduces a new novel method of interacting with the visualisation via gestures with a Microsoft Kinect sensor.

1.2 Key Issues Project Addresses

To summarise the above sections, this project addresses the following key issues:

Issue 1. Content in the Kepler Exoplanet Database is difficult to view and understand due to its amount and labeling.

Issue 2. Planetary information is complex and difficult to comprehend without a visual reference due to its scale.

Issue 3. Existing visualisations for exploring planetary data have minimal functionality for exploring the information in effective ways.

Issue 4. Visualisations need to allow user interaction to make the most of the data they display.

These issues are explored in Chapter 3 to create the project requirements.

1.3 Contributions of this Project

This project provides an interactive 3D visualisation that conveys more information and contains better interactivity than other visualisations in the same field. This extension is evaluated both qualitatively and quantitatively by a within subjects user experiment to ensure that it successfully conveys the information in the Kepler Exoplanet Database. A key part of this project is the development of interactive techniques to provide the best access to the data. These techniques are a mix of keyboard and mouse as well as a novel gesture based approach using a Microsoft Kinect sensor. The work and research completed for this project will provide the opportunity for further improvement of the produced visualisation in the future.

Chapter 2

Project Methodology

2.1 Project Management Approach

Project management is the discipline of planning, executing, monitoring, and evaluating a project. It is a vital role as it is the glue between all of the components that go into a project. It is beneficial as it encourages critical thinking about requirements, design, and testing before coding is commenced. For this project it helped to avoid the problem of following a code-and-fix approach which is 1) write some code. 2) Fix the problems in the code [13]. The code-and-fix approach may be suitable in some small scale applications, but as soon as a system becomes complex it increases the chance of a system turning into a nightmare of high coupling, low cohesion, no modularity, minimal structure, and little consistency [21]. A widely used method of managing IT projects is to use a Software Development Life Cycle (SDLC), a form of project methodology that details the stages that a project must pass through whilst being completed.

2.1.1 Software Development Life Cycle (SDLC)

The SDLC chosen for this project was a customized Spiral Model [13] made up of requirements analysis, design, implementation, and evaluation stages as shown in Figure 2.1. A software project repeatedly iterates through these stages for each functionality or component produced. The first stage of the spiral involves analysing the requirements of the functionality being created. Following this designs are created and then implemented to produce a deliverable artifact. Finally in the fourth stage the artifact is user tested by one or more people. This evaluation is in addition to the final user evaluation at the completion of the project.

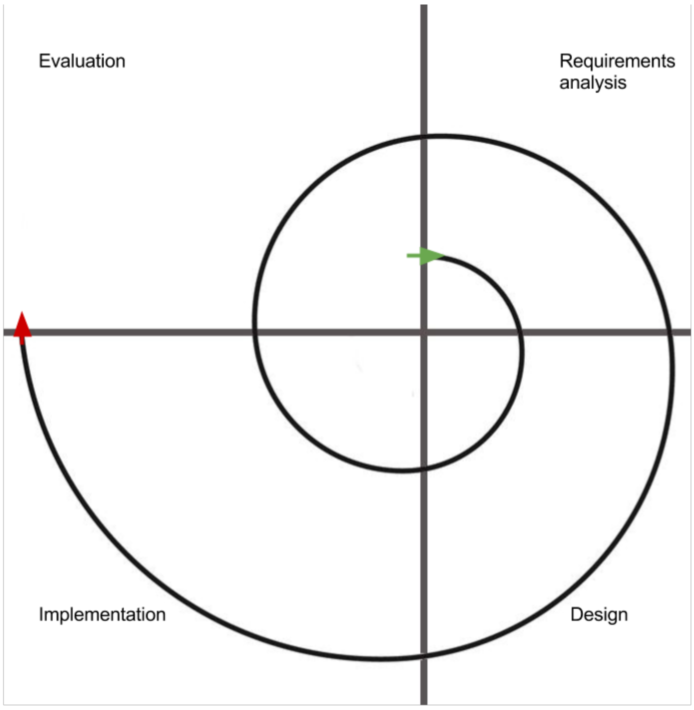

Figure 2.1: The Spiral Model was the SDLC chosen for this project, the above image shows the stages in the spiral

This SDLC supported the creation of a visualisation as it allowed the flexibility to add and remove components from the visualisation as they were discovered to be beneficial or not. The choice of this project management approach meant that whilst I had the freedom to explore visualisation options I also had a structured software development life cycle to guide me and the project through the necessary steps to end in the successful completion of each component. Using a spiral model also allowed me to produce a deliverable feature at the end of each iteration of the model which occurred each week to coincide with a meeting with the supervisor of the project, thus ensuring that I did not become delayed or stuck in my development with nothing produced. Using this methodology also helped to prioritise the features that were the most important to the visualisation, reducing the risk of missing or incomplete artifacts at the end of the project. Following this SDLC allowed me to achieve all of the goals that I set, although some time frames and deadlines had to be revised as elements were completed before or behind schedule. nightmare

2.2 Alternative Methodologies Explored

Two other project methodologies were explored before the Spiral Model was chosen. The alternatives were a loose agile method SCRUM [31], and a strict Structured Method, the Waterfall Model [30]. The Spiral model provided me with the benefits of a structured work flow that is a feature of the Waterfall Model as well as a flexible iterative process that is a feature of Agile methodologies without needing to adhere only to the strict guidelines of either. Following a pure waterfall methodology would not have allowed me to iteratively design, develop, and evaluate each feature and would have forced more upfront design limiting the flexibility and support for changing requirements needed for this project. A pure agile approach was not suitable as Agile methodologies are more beneficial to team based projects. As it was, I was agile in my approach to the project and embraced changing requirements, collaborated with my supervisor (the customer) regularly, and emphasized working software over large amounts of documentation [10].

2.3 Supporting Tools for Project

Supporting this project methodology with other project management tools such as Gantt charts (Appendix B) encouraged efficient documentation of planning and work completed in the project as well as displaying the upcoming stages required to complete the project.

Version control was important for this project as it mitigated against the risks of file system crashes and corruption. It also maintained effective revision history that could be used to backtrack to or view changes that were made earlier in the project. Version control was also valuable as it allowed me to maintain multiple branches (versions) of my project that could be swapped between whilst developing multiple features. This was important for the creation of the keyboard & mouse system and the Microsoft Kinect system as they were on separate branches in the repository. The version control tool used for this project was Git which allowed the repository to be hosted on the repository hosting web service Github. A limitation of this choice was that the repository was hosted on a free Github license which meant that it would be open to the public to view.

Weekly meetings with the supervisor of the project, Dr Stuart Marshall, were used to provide guidance and ideas for innovation of the visualisation throughout the project. These meetings ensured that vital components and deliverables were implemented in the required timeframe and also provided a sounding board for ideas. Another important aspect of having an involved supervisor was that he provided the guidance of an experienced academic which was indispensable when navigating the administrative side of organizing delicate matters such as ethics approval for the user evaluation of the visualisation.

Chapter 3

Requirements Analysis

To guide the creation of the visualisation Cooper et al.'s User Oriented design approach was used [17], in particular making use of user personas (also known as archetypal users). These personas are created to give a sense of empathy and understanding for the foreseen users of the visualisation in order to better understand the requirements and design decisions to be made. These personas personify the needs of a larger group of related users. They act as stand-ins for real users, describing them in terms of their goals and personal characteristics, and although they are fictitious, they are based on knowledge of the traits of the intended users. An additional tool used was User Scenarios which describe the foreseeable interactions of the user personas within the visualisation. A Scenario is made up of a functional goal for the visualisation and describes how it is carried out by a persona.

Both of these tools force you to think about the tasks needed for the visualisation and their context in the system as a whole. Once the personas and scenarios have been completed you can then start to design specific elements of the user interface and visualisation based on the requirements and interactions described in the scenarios. The User Models and User Scenarios for this project are described in the following sections.

3.1 User Models Used

Below are the two personas that were used in this project. They depict users that would use the visualisation in the context of a terminal or display in an observatory environment. These personas can be validated during the user evaluation of the visualisation by finding real users that match the core values of these personas. Although this visualisation would also be suited towards teaching children about stellar information this was not focused on during planning. This was because of the increased ethical complexity of carrying out an evaluation with children. This could be done in the future once the visualisation is deemed successful for users of other age ranges. For each of the personas below there are a range of User Scenarios that outline the functionality needed for the deliverable of this project.

3.1.1 John Truman (Primary Persona - The interested layperson)

24 year old John is interested in planets and space and has a basic knowledge about both. He frequently visits attractions catering to this interest at locations such as planetariums and observatories. Some of his favourite things to do when visiting these attractions is to go to interactive computer terminals that allow users to choose what information they wish to access.

John is used to playing computer games and using visualisations and is not overwhelmed understanding and using new systems. He finds that he learns better when provided with visual examples than when reading or listening to large amounts of information. John is most comfortable using keyboard and mouse when interacting with a computer.

Scenario 1: View planets ordered by their similarity to Earth

When John first sees the system the first thing he notices is that there are many planets orbiting what looks like a star. He doesn't have any point of reference for these planets so their sizes, colours, and movement speeds are meaningless. By providing a way of comparing the planets to Earth it gives a point of reference which is well documented and known by most.

Procedure:

1. John selects that he wants to view the exoplanets compared by their similarity to earth.

2. The planets on screen move so that they are placed in a way that John can compare them to Earth.

3. John can now select any of the planets for further analysis.

Scenario 2: Select ranges for attributes of each planet displayed

John has become comfortable with selecting planets and has some idea of their scale and basic attributes. Now he wants to select more planets to find out more information. However due to the large number of planets he finds it difficult to make an accurate selection due to overlapping and fast moving small planets.

Procedure:

1. John uses a range of filters to remove planets from his view that don't match the criteria he chooses.

2. As planets disappear the graph of planets expands into the space that frees up, this causes more space to appear between planets making them more selectable.

Scenario 3: Select planets to display more information

John wants to see more information about the planets he can see orbiting in the visualisation. To do this he wants to be able to select the planets and have textual information appear on screen.

Procedure:

1. John has the option to pause the rotation of planets in order to make more accurate selections.

2. John selects a planet.

3. The planet selected becomes more visible.

4. A text window has information about the planet selected added to it.

Scenario 4: View planets in the same solar system

John is curious about which of the planets he can see in the visualisation are in the same Solar System. He wants that when a planet is selected all other planets in the same Solar System become highlighted.

Procedure:

1. When John selects a planet, all planets in the same Solar System become more visible.
2. A label appears on these planets indicating that they are related planets.

Scenario 5: View the Goldilocks zones of each exoplanets star

Looking at the planets orbiting the sun in the visualisation John wonders whether any of them could support life. To see this John wants to see which planets are in the habitable zones of their stars.

Procedure:

1. John selects that he wants to view the exoplanets compared to their stars habitable zones.
2. The habitable zone of the selected planets star become visible.
3. When a planet from a different star system is selected then the visible habitable zones will change to the new selected planets stars habitable zone.

Scenario 6: Select two planets to compare against one another

When John is selecting planets to view more information he often finds that he wants to compare his selections against another planet. To do this John wants to be able to make multiple selections to compare two planets against one another.

Procedure:

1. John selects a planet and chooses to compare it against another planet.
2. Information about this second planet appears so that John can make comparisons.

3.1.2 Cara Thompson (Secondary Persona - Likes gesture based systems)

Cara is a 23 year old woman interested in visiting interesting attractions. She especially enjoys using interactive visualisations with novel means of interaction when visiting attractions. This is because she finds that they are more entertaining and provide a more immersive experience than a visualisation with a keyboard and mouse.

Scenario 3: Select planets to display more information

Cara wants to see more information about the planets she can see orbiting in the visualisation. To do this she wants to be able to perform a gesture to select a planet to access more information.

Procedure:

1. Cara hovers her hand over a planet to make a selection
2. The planet selected becomes more visible.
3. A text window has information about the planet selected added to it.

Scenario 4: View planets in the same solar system

Cara is curious about which of the planets she can see in the visualisation are in the same Solar System. To discover this she wants all other planets in the same Solar System to become highlighted when an exoplanet is selected.

Procedure:

1. When Cara selects a planet, all planets in the same Solar System become more visible.
2. A label appears on these planets indicating that they are related planets.

Scenario 7: Navigate the visualisation with gestures

Cara doesn't find using a keyboard and mouse interesting for interacting with a visualisation. She would rather navigate around a visualisation by using hand gestures as it's more immersive.

Procedure:

1. By moving her hand the visualisation pans in the corresponding direction, i.e. if the hand goes to the top of the screen the visualisation pans up.
2. By moving her hand backwards and forwards the visualisation zooms in and out.

Both of these users are similar in their need for information from the visualisation but differ in the methods that they wish to access the information and interact with the visualisation. John wants to interact with keyboard and mouse as it is more straight forward and accurate. Cara wants to interact with gestures as she finds it more of a novelty and more immersive.

3.2 Requirements Summary

From these User Scenarios we can see that behind each of them there is a requirement that a visualisation can address. The requirements that were extracted from the User Scenarios are introduced in the following subsections.

3.2.1 Functional Requirements

Functional requirements define the behaviour of a system and are derived from the scenarios described above. These functions are described as a set of inputs, the behavior, and outputs from the system. The functional requirements for this visualisation are as follows:

R1. The visualisation needs to display planetary information to convey knowledge to users.

R2. The visualisation needs to allow exoplanets to be compared against one another.

R3. The planets need to be able to be ordered by their similarity to earth (ESI) and by their Kepler Object of Interest number (KOI).

R4. The visualisation needs to allow users to define ranges of planetary attributes to filter which planets are displayed.

R5. Users need to be able to view the habitable zones (Goldilocks zones) of stars in relation to the planets orbiting them.

3.2.2 Nonfunctional Requirements

Functional requirements are supported by nonfunctional requirements. Nonfunctional requirements impose constraints on the design or implementation (such as performance, security, or usability) of a system. The nonfunctional requirements for this visualisation are as follows:

R6. All interaction methods must be visible and intuitive.

R7. The visualisation must remain uncluttered to reduce information overload.

R8. There needs to be two modes of interaction in the system, keyboard & mouse vs gesture based.

Key:

Functional Requirements
Nonfunctional Requirements

	Issue 1	**Issue 2**	**Issue 3**	**Issue 4**
Req 1	✓	✓	✓	
Req 2				✓
Req 3	✓	✓	✓	✓
Req 4			✓	✓
Req 5	✓	✓	✓	✓
Req 6		✓		✓
Req 7		✓		
Req 8				✓

Figure 3.1: A matrix of the project requirements mapped to the issues they attempt to address.

Chapter 4

Related Work

This section discusses four existing visualisations themed around space or planets. Following a short description, each visualisation is analysed to discover which of the User Scenarios detailed in the previous chapter exist within the system.

4.1 Worlds: The Kepler Planet Candidates - Non Interactive

Worlds [27] displays planet candidates found by NASA's Kepler mission. These candidates are animated in orbit around a single star. They are drawn to scale with accurate radii, orbital periods, and orbital distances. They range in size from 1/3 to 84 times the radius of Earth. Colors represent an estimate of temperature with red indicating warmest, and blue indicating coldest candidates.

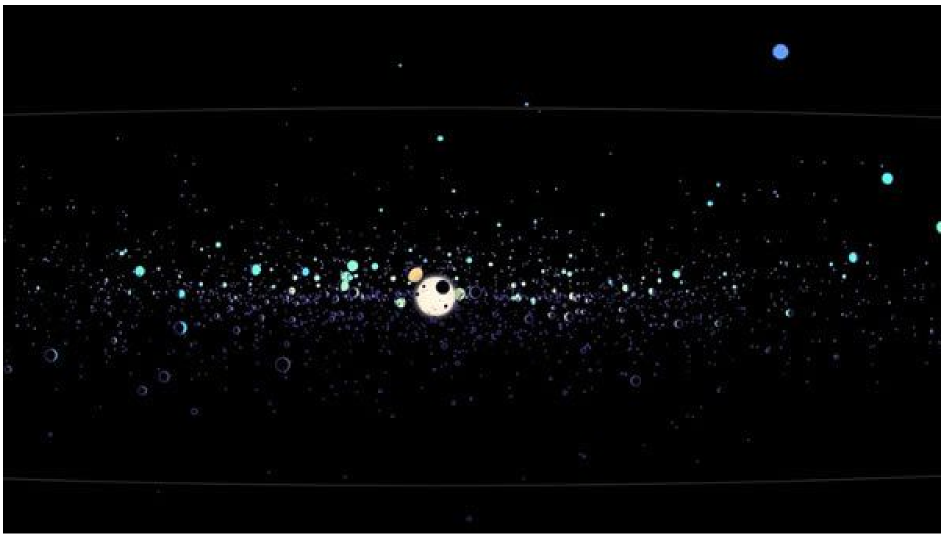

Figure 4.1: The Worlds visualisation, in it can be seen exoplanets orbiting around a single star. Each exoplanet is colour coded and sized to represent an attribute

Worlds has a visually appealing layout that effectively displays the basic attributes of each planet. By examining this visualisation we can see how displaying the planets orbiting a single star allows users to visually make a basic comparison between each planet. The below paragraphs discuss the User Scenarios that Worlds fulfills.

Scenario 1. View planets ordered by their similarity to Earth:

Worlds has comprehensive functionality for comparing the different exoplanets to one another, however it does not offer any functionality regarding comparisons to earth

Scenario 2. Select ranges for attributes of each planet displayed:
Worlds doesn't offer any functionality for any filtering of exoplanets, this means that users can only see all planets at once which can be overwhelming and causes exoplanets to be excluded from view due to overlapping and clustering. The reason that this is done is to convey how many exoplanets there are and how their scale differs among on another.

Scenario 3. Select planets to display more information:
Worlds is non interactive meaning that users are not able to request further information about the visualisation elements that they are seeing. This ability to interact with the visualisation to find out more information is a key part of the interactive visualization needed for this project.

4.2 The Kepler Orrery and The Kepler Orrery 2 - Non interactive

The Kepler Orrery [18] illustrates exoplanets in their own solar systems. The orbit radii are to scale with respect to each other and planet sizes are to scale with respect to each other, but orbits and planet sizes are different scales. The colors are in order of semi-major axis: two-planet systems (242 in all) have a yellow outer planet; 3-planet (85) green, 4-planet (25) light blue, 5-planet (8) dark blue, 6-planet (1, Kepler-11) purple.

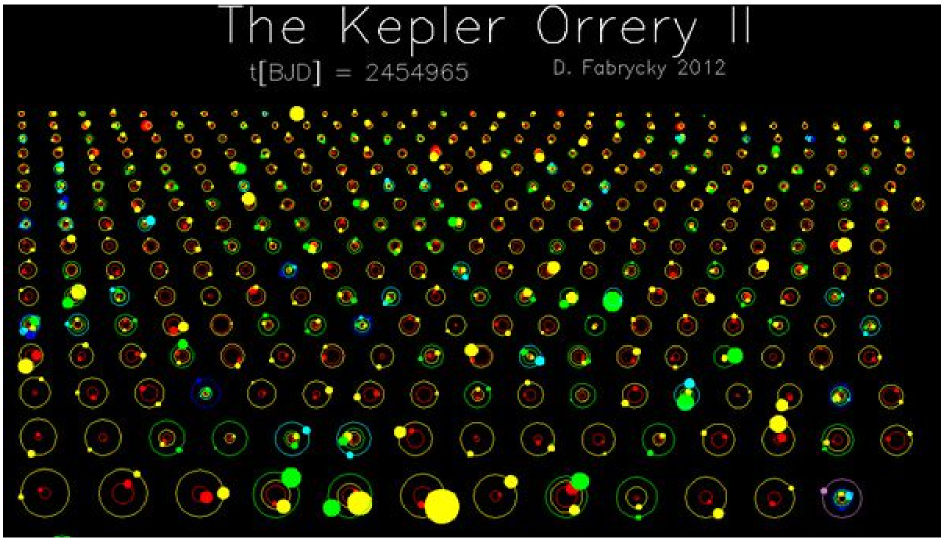

Figure 4.2: The Kepler Orrery visualisation, in it can be seen exoplanets grouped and displayed in the solar systems that they reside.

This system exhibits small multiples, a grid of small similar graphics or charts, allowing them to be easily compared. This provided insights into how to use small multiples to display information about groups of planets. This was important for displaying which planets share a solar system. The below paragraphs discuss the User Scenarios that The Kepler Orrery fulfills.

Scenario 1. View planets ordered by their similarity to Earth:
Like Worlds, The Kepler Orrery shows the similarities between each of the exoplanets but

does not have the functionality to allow users to make a comparison to earth and our own solar system.

Scenario 4. View planets in the same solar system:
The layout of the visualisation uses small multiples to group each solar system of exoplanets and stars together which removes the issue of overcrowding and overlapping elements. Displaying each exoplanet orbiting its own star removes the risk of confusion about what planets are actually orbiting which could be the case with Worlds.

4.3 Celestia - Interactive

Celestia [1] is a 3D space simulation written in C++. Celestia does not include any stars that are more than a few thousand light-years from the Sun because the distant stars are too small difficult to measure, meaning that it doesn't contain the distant exoplanets discovered by the Kepler mission.

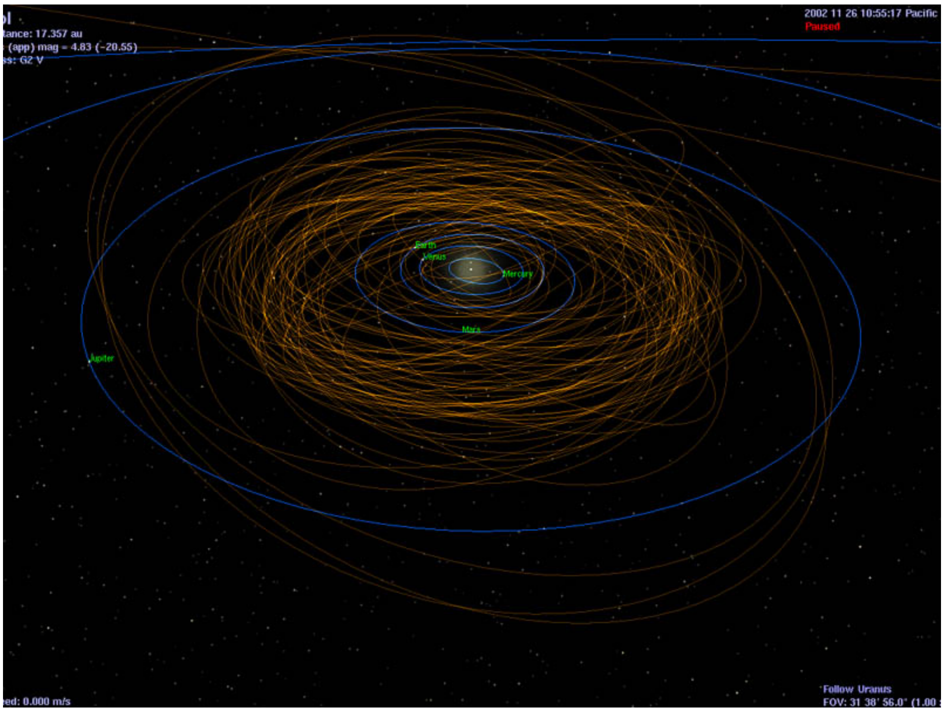

Figure 4.3: The Celestia visualisation, in it can be seen the orbital view of a solar system showing the orbit paths of each planet.

This visualisation is a much larger and more encompassing system than is needed for this project, as it is a full simulation. However, it does offer insights into how to effectively portray planets and their orbits as in Figure 4.3. It also provides textures that can be used to depict what planets look like to increase user immersion. The below paragraphs discuss the User Scenarios that Celestia fulfills.

Scenario 3. Select planets to display more information:
Celestia allows users to view a large set of information about each of the visible planets. The information is available in toolbars that can be accessed. Examples of this information are radius, phase angle, rotation speed, and temperature of planets.

Scenario 4. View planets in the same solar system:
Celestia allows users to explore a range of solar systems and the planets in them. This is a key feature of the experience that Celestia tries to give users, i.e. letting them explore the vastness of space in a 3D simulation.

4.4 Kepler Visualisation Tool

The Kepler Visualisation tool [12, 33] is a 3D visualisation built with Processing (A Java library and development environment). It is a simple visualisation focusing on displaying the estimated size, orbital speed, and orbital separation of each exoplanet. All exoplanets are color-coded to visually represent their estimated temperatures.

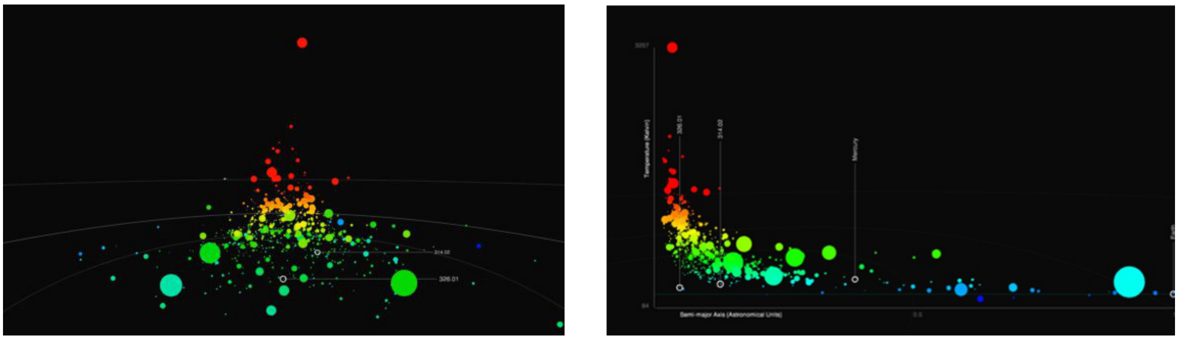

Figure 4.4: The Kepler Visualisation Tool, in it we can see a view of exoplanets orbiting around a star ordered by their temperature shown by their colour. We can also see a graph view with the same ordering

The existing work in this system would serve as foundation for this project as much of the visual aspects, and initial data manipulation of this visualisation is already complete. It means that implementing the features needed for this project could be focused on more heavily, and larger improvements could be undertaken. The below paragraphs discuss the User Scenarios that Kepler Visualisation Tool fulfills.

Scenario 1. View planets ordered by their similarity to Earth:
Like Worlds and the Kepler Orrery, The Kepler Visualisation Tool has functionality to display the similarity of each exoplanet to each other. However, it also has some limited functionality of comparing these to earth which the others lack. It does this by displaying Earth, Mars, and Jupiter by the same method as the exoplanets. This gives users a point of common reference with which to make comparisons.

Scenario 2. Select ranges for attributes of each planet displayed:
The Kepler Visualisation Tool allows users to sort the exoplanets on the Y axis by their size and temperature, but does not allow users to specifiy ranges of these to filter them.

4.5 Summary of Existing Applications

Existing System	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6	Scenario 7
Worlds	No	Partial	No	No	No	No	No
The Kepler Orrery	No	Partial	No	Yes	No	No	No
Celestia	No	No	Yes	Yes	No	No	No
Kepler Visualisation Tool	No	Partial	No	No	No	No	No

Figure 4.5: Matrix of existing solutions mapped to user scenarios. We can see that none of the existing solutions provide a solution to each of the user scenarios although some fulfill some completely or partially.

Chapter 5

Technology

Many technologies were examined and experimented with as the candidates for the basis of the visualisation tool. Then the positives and negatives of each was weighed up before a decision about the suitable choice was made. It came down to three potential technologies, the next three subsections outline these in detail.

5.1 Processing - Chosen Technology

Processing [8] is an open source programming language and development environment that was created to teach the fundamentals of computer programming with a visual context. Using processing would mean that the visualization could be built with Java while still using an effective visualisation framework that supports 3D elements. The most complete existing visualization using the same exoplanet dataset (The Kepler Visualization Tool) is built using Processing. Using this solution would involve learning the semantics of Processing but as it is a library built in Java so the syntax is only slightly different. This means the learning curve should be shallow.

5.2 D3 (Data Driven Documents) - Alternate Technology

D3 [15] is a JavaScript library that allows the data to be displayed in dynamic graphics. Embedded within an HTML web page, the JavaScript D3.js library uses pre-built functions to select elements, create Scalable Vector Graphics (SVG)objects, style them, and add transitions, dynamic effects, and tooltips. Large datasets can be easily bound to SVG objects using simple D3 functions to generate rich charts and diagrams. D3 was created because of the need for a balance of expressiveness, efficiency, and accessibility that previous visualisation toolkits did not allow [4].

D3 allows the binding of input data to arbitrary input elements. This means that the exoplanet dataset can easily be bound to SVG elements to create a visualisation. D3 adopts the W3C Selectors API to identify document elements queried. This results in a rich but concise selection method of elements in a visualisation. It allows debugging due to Google Chrome and other modern browser development tools. A downside to D3 is that it does not allow 3D diagrams, although it does allow pseudo 3D by using the painters algorithm and 3D textures.

5.3 Prefuse - Alternate Technology

Prefuse is a set of software tools for creating rich interactive data visualizations [23]. The Prefuse toolkit provides a visualisation framework for Java. It supports a set of features for visualising and interacting with data. It can be used to build standalone applications, visual components embedded in larger applications, and web applets. Prefuse greatly simplifies the process of representing and efficiently handling data, mapping data to visual representations (e.g. through spatial position, size, shape, color, etc), and interacting with the data. To use Prefuse a basic familiarity with the Java is required, including setting up and building Java projects. A knowledge of Swing or another similar user interface toolkit is also useful for understanding some of the concepts behind Prefuse and for integrating Prefuse visualisations into larger applications. Prefuse is a very powerful tool that has a very high learning curve due to the amount of development power that it has. This means that learning it and using it pushed it out of scope for this project.

5.4 Decision of Technology

The technology chosen needed to offer a combination of low learning curve, strong visualisation ability, and 3D support. The final decision was to use Processing, this is because it had all of the desirable qualities that this project required as Figure 5.1 shows.

	D3 (Appendix A.1.2)	Processing (Appendix A.1.1)	Prefuse (Appendix A.1.3)
Potential for 3D	***No***	***Yes***	***No***
Has low learning curve	***Yes***	***Yes***	***No***
Prior evidence of successful visualisations	***Yes***	***Yes***	***Yes***
Interactive	***Yes***	***Yes***	***Yes***
Dynamic transitions	***Yes***	***Yes***	***Yes***
Has existing solution related to planets	***No***	***Yes***	***No***

Figure 5.1: Matrix of technology options to desirable qualities. We can see that Processing is the most suitable as it satisfies each of the desirable qualities followed by D3 and then Prefuse

Choosing Processing as the development tool allowed me to extend an existing visualisation using the same data set, The Kepler Visualisation Tool in Section 4.4. Building upon an existing solution allowed the project to progress faster towards fulfilling the project requirements as less of the groundwork needed to be carried out. This was a large advantage as doing this groundwork would limit the amount of time spent on new features.

Chapter 6

Solution Design: Improved Kepler Visualisation Tool (IKVT)

This section discusses the design of the deliverable visualisation, the Improved Kepler Visualisation Tool (IKVT). It details the key design decisions revolving around structure, aesthetics, and functionality that were made about the visualisation. This project aimed to improve an existing visualisation, The Kepler Visualisation Tool which was discussed in the previous chapter. This visualisation displays exoplanets and some of their features but lacks effective interactivity for users to access the detailed information contained in the Kepler Exoplanet Database. The IKVT expands on this pre-existing visualisation by adding key elements of interactivity missing in the existing visualisation as well as further enhancing the range and amount of data that is available to users. The IKVT also incorporates a novel gesture based interactive mechanism for controlling the visualisation using a Microsoft Kinect Sensor.

6.1 System Design and Structure

As this project builds upon the Kepler Visualisation Tool, complete comprehension of how it is designed and how it functions is important. Going ahead in extension without this knowledge would create opportunities for mistakes and incorrect assumptions. To solve this issue diagrams using the Unified Modeling Language(UML) were used. In UML there are two basic categories of diagrams: structure diagrams and behavior diagrams. Every UML diagram belongs to one these two diagram categories. The purpose of structure diagrams is to show the static structure of the system being modeled i.e. class diagrams, and object diagrams. Behavioral diagrams show the dynamic behavior between the objects in the system, including aspects like their methods, collaborations, and activities i.e. use case and sequence diagrams [11]. For this project sequence diagrams and class diagrams were used to understand the existing system and plan out the extensions, these are discussed next.

1. **UML Sequence Diagram** A sequence diagram primarily shows the interactions between objects in the sequential order that those interactions occur. For this project it was used to understand how each of the objects in the system worked together to create the visualisation. Without understanding which objects were responsible for each part of the render cycle it would have been difficult to extend the visualisation. Through the sequence diagram we can see when each of the objects in the system are updated, and when they are rendered.

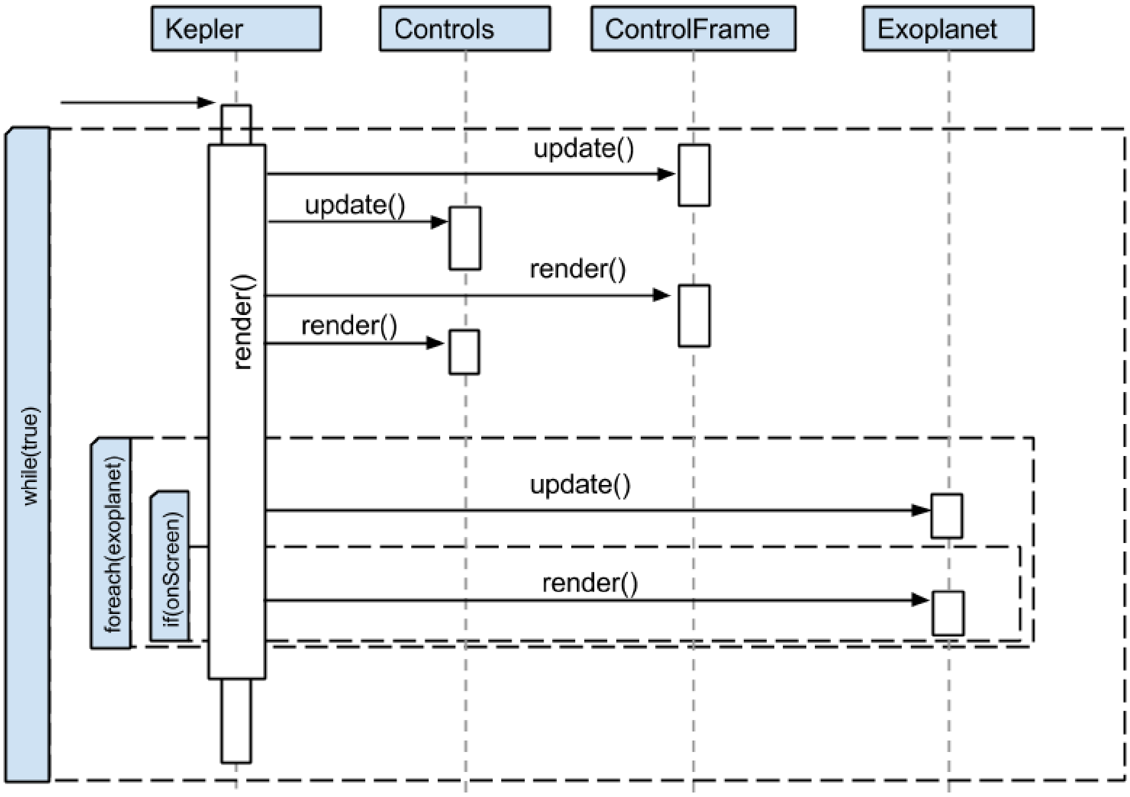

Figure 6.1: Sequence Diagram of IKVT render cycle, it shows when each component of the visualisation is updated and rendered.

2. **UML Class Diagram** Class diagrams describe the structure of a system by showing the systems classes, their attributes, methods, and the relationships among objects. Developers can use class diagrams to design and document the systems coded (or soon to be coded) classes. For this project class diagrams were used to understand the makeup of the existing Kepler Visualisation Tool and then to plan the extensions. The diagram in Figure 6.2 shows the structure and attributes of the Kepler Visualisation Tool. This class diagram was updated with the improvements introduced in the visualisation to provide an overview that could be used for further planning.

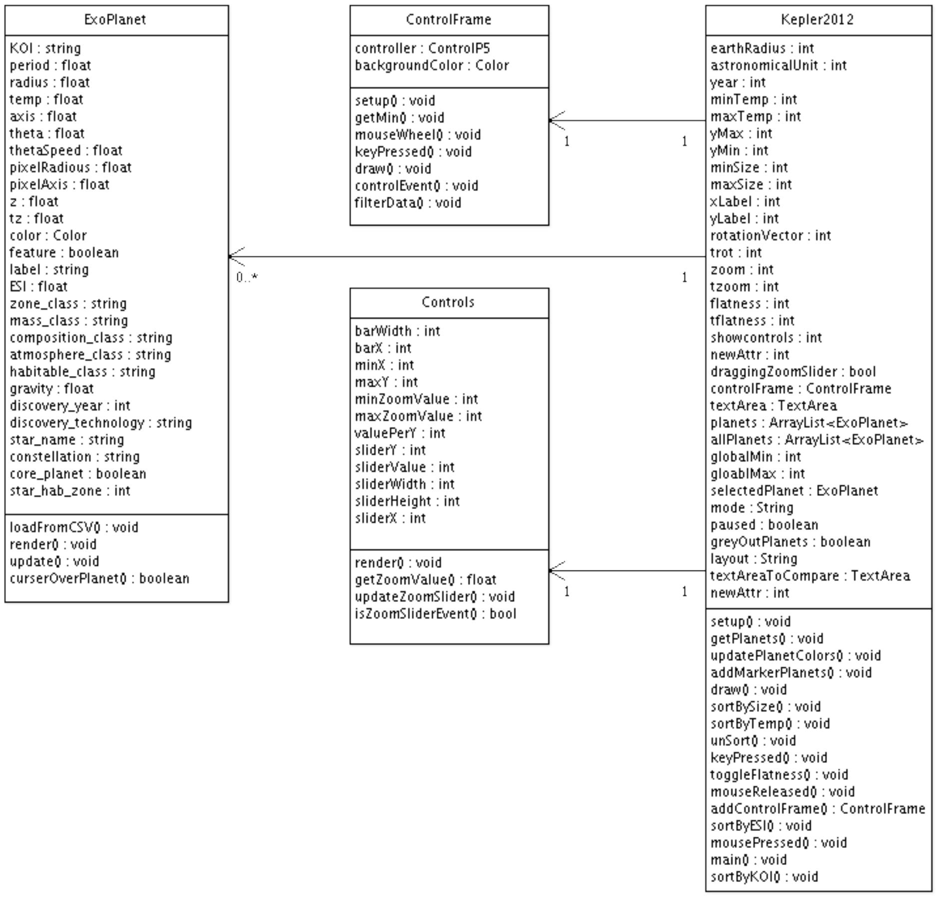

Figure 6.2: The class diagram used to understand the structure of the existing Kepler Visualisation Tool and plan the extensions.

By using the sequence diagram in Figure 6.1 coupled with the class diagram in Figure 6.2 we can see where each method and field is located and modified. This is a powerful way to understanding an existing system and then planning extensions.

6.2 Visualisation Design

The requirements produced in Chapter 3: Requirements Analysis provide a description of the functionality needed for this visualisation. By adding additional details to these requirements we can specify how the visualisation should look, behave, and function. This is done in the following subsections.

6.2.1 Functional Requirements

R1. The visualisation needs to display planetary information to convey knowledge to users.

This requirement needs a textual display in order to convey information about exoplanets to the user. This can be done with a Java text area object to display each of the key attributes of exoplanets. The following figure is an example of the text area showing the information about each planet that would be displayed and the method calls that would be used.

```
KOI: planet.getKOI()
Temperature: planet.getTemp()
Gravity: planet.getGravity()
Radius: planet.getSize()
Zone Class: planet.getZoneClass()
Mass Class: planet.getMassClass()
Composition: planet.getComposition()
Habitability: planet.getHabitability()
Atmosphere: planet.getAtmosphere()
Method of discovery: planet.getDiscoveryMethod()
Year of discovery: planet.getDiscoveryYear()
Earth Similarity: planet.getESI()
```

Figure 6.3: Example of the text area that displays information about selected exoplanets. Also included are the planned method calls for populating the fields

R2. The visualisation needs to allow exoplanets to be compared against one another.

There are two steps to fulfilling this requirement
1. Allow selection of exoplanets
2. Display additional information about exoplanets when they are selected.

The selection of exoplanets involves detecting when and where a user clicks, and then finding whether any planets are located in that space. This is more complex in this system as it requires detecting if the 3D space of each planet coincides with the 2D location of the mouse click. When a planet is successfully selected it needs to provide feedback to the user to inform them that it has been selected and also to provide information about the selected exoplanet as in Figure 6.3.

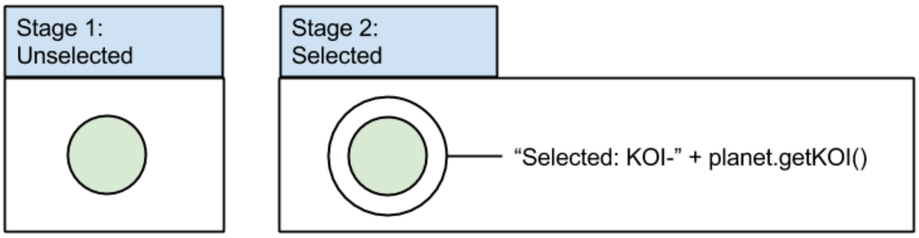

Figure 6.4: Example of exoplanet selection process. It displays the result of an exoplanet being selected, first the planet expands, then a white circle expands out to outline it.

In addition to this, when a planet has been selected all other planets in the same solar system (sister planets) need to become highlighted. This can be done by treating them as if they were selected and providing an additional indication to the user why they were highlighted as in Figure 6.5.

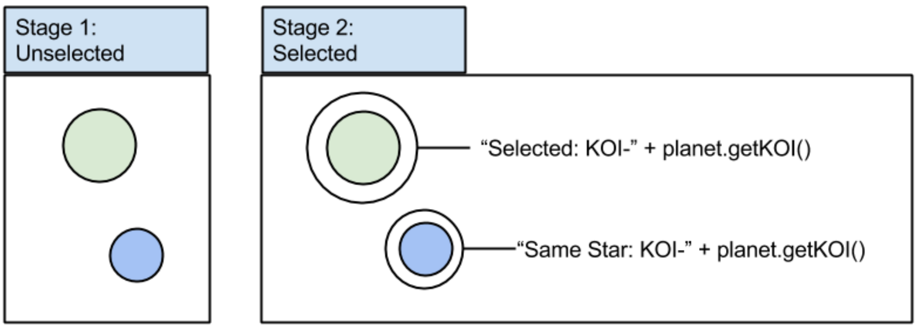

Figure 6.5: Example of exoplanets in the same solar system. It displays how the planets in the same solar system as the selected planet become highlighted.

There needs to be an efficient way for users to make comparisons between the detailed textual information of exoplanets. By providing an additional text box and allowing users to select multiple planets this can be accomplished as shown in Figure 6.6.

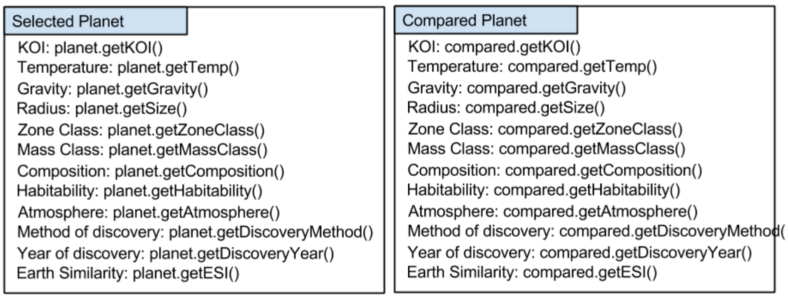

Figure 6.6: Example of exoplanet comparison text areas, included are the planned method calls for populating the fields of the second text area

R3. The planets need to be able to be ordered by their similarity to earth (ESI) and by their Kepler Object of Interest number (KOI).

The visualisation needs to allow users to view the exoplanets in a way that uses the Earth as a point of reference and their Earth Similarity Index (ESI) to order them and control their position. Figure 6.7 displays an example of how this would be done.

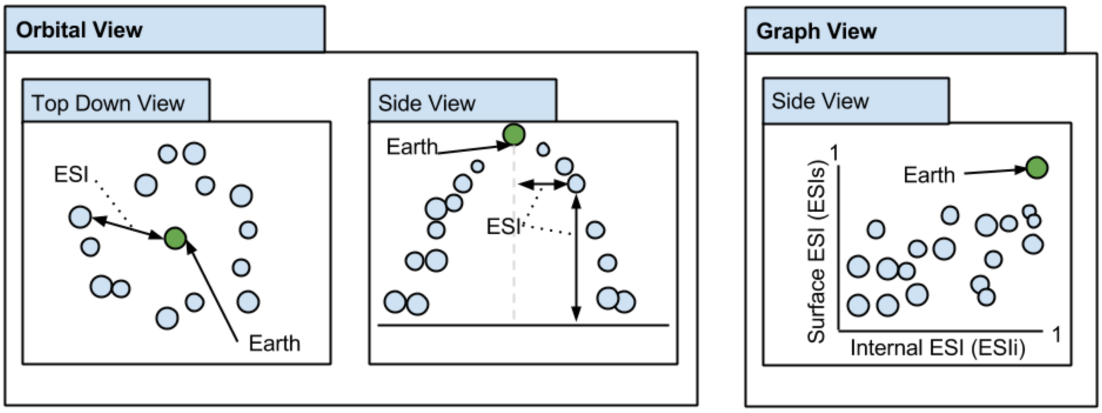

Figure 6.7: Example of ESI views, it shows how the ESI is used to order the exoplanets in the orbital and graph views.

R4. The visualisation needs to allow users to define ranges of planetary attributes to filter which planets are displayed.

A method of filtering the exoplanets displayed to the user is needed. A common method of achieving this is to use a set of sliders that allow a user to filter something by a set of values. In this system sliders would be used to control the planets displayed by the exoplanet attributes: size, temperature, Kepler Object of Interest number (KOI), and Earth Similarity Index (ESI) as in Figure 6.8.

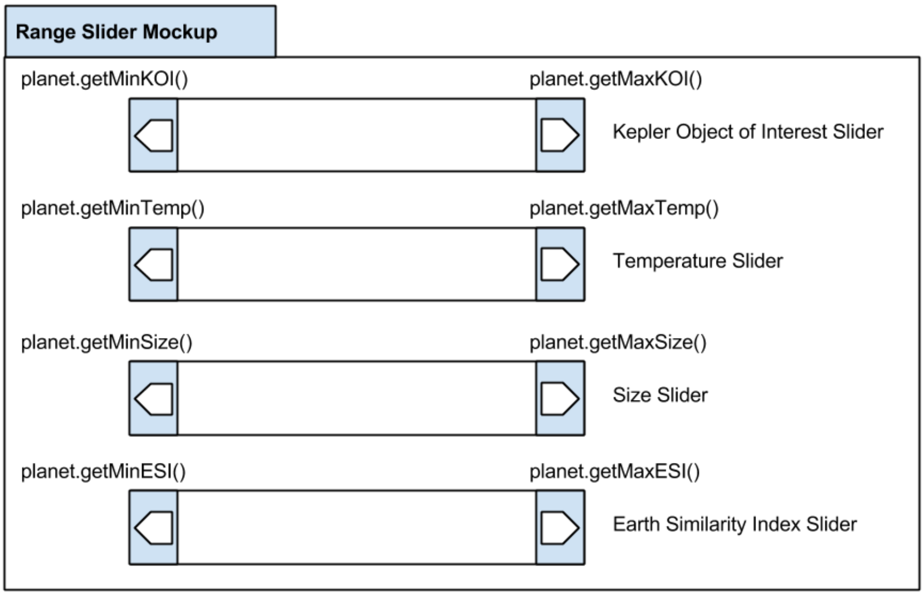

Figure 6.8: Example of range sliders, it displays the method calls used for the ranges

In addition to this users should be able to sort the exoplanets to make the most of the filtering. By allowing users to display the exoplanets sorted vertically by the same attribute as the filters, they can see how changing the filters affects the exoplanets displayed as exoplanets on either side of the specified range are omitted. This would be achieved by a set of interactive buttons as in Figure 6.9.

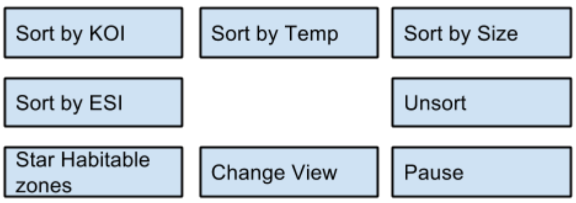

Figure 6.9: Example of interactive buttons displaying each of the key interactive functions

R5. Users need to be able to view the habitable zones (Goldilocks zones) of stars in relation to the planets orbiting them.

To view the Goldilocks zone each exoplanets star requires a way to display the hot (to close to the star), cold (to far from the star), and the habitable zone (in between hot and cold zones). This can be done by showing the selected exoplanet in relation to its star by means of coloured circles depicting where the exoplanet sits inside the zones, as displayed in Figure 6.10.

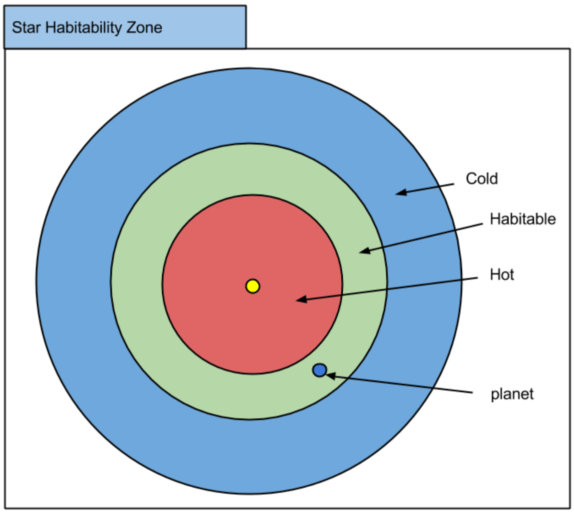

Figure 6.10: Example Goldilocks zones showing how the coloured rings are used and how the planets relation to them infers their habitability.

6.2.2 Nonfunctional Requirement

R6. All interaction methods must be visible and intuitive.

To make an interactive visualisation usable the controls and interactive methods need to be clear to users. To achieve this for IKVT an interactive control panel is introduced. This panel contains all of the interactive elements of the visualisation in a single central place that is spatially separated from the main visualisation. This spacial separation is important as it reduces cognitive load on users [20]. It does this by allowing them to focus on the visualisation itself until they need to use the control panel in which case they can switch focus to that [26]. It also means that users only need to look in one place for all of the interactive elements which reduces the risk of confusion. The control panel would contain all of the interactive elements as discussed in the previous requirements as displayed below in Figure 6.11.

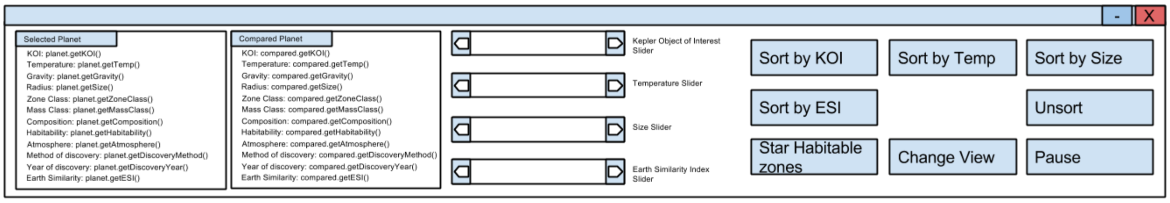

Figure 6.11: Example of the control panel showing the placement of each of the components talked about previously.

In the gesture based system intuitivity can be achieved by reusing effective and widely recognised gestures from other visualisations. In this case these would be push, pull, sweep left/right, raise, lower, and hover.

R7. The visualisation must remain uncluttered.

The visualisation must not show so much information that it causes information overload for users. The ability to filter and sort the exoplanets gives a user the tools needed to reduce the quantity of planets displayed in the visualisation and thus the information load imposed on users [25]. In addition to this, having the control panel separate from the main visualisation reduces the cognitive load on users as they don't have to use this component until they want to. Having the interaction broken up into three sections also assists with this as it utilises Gestalt's law of proximity [22] which states that when an individual perceives an assortment of objects close together they are seen a group. This layout is shown in Figure 6.12.

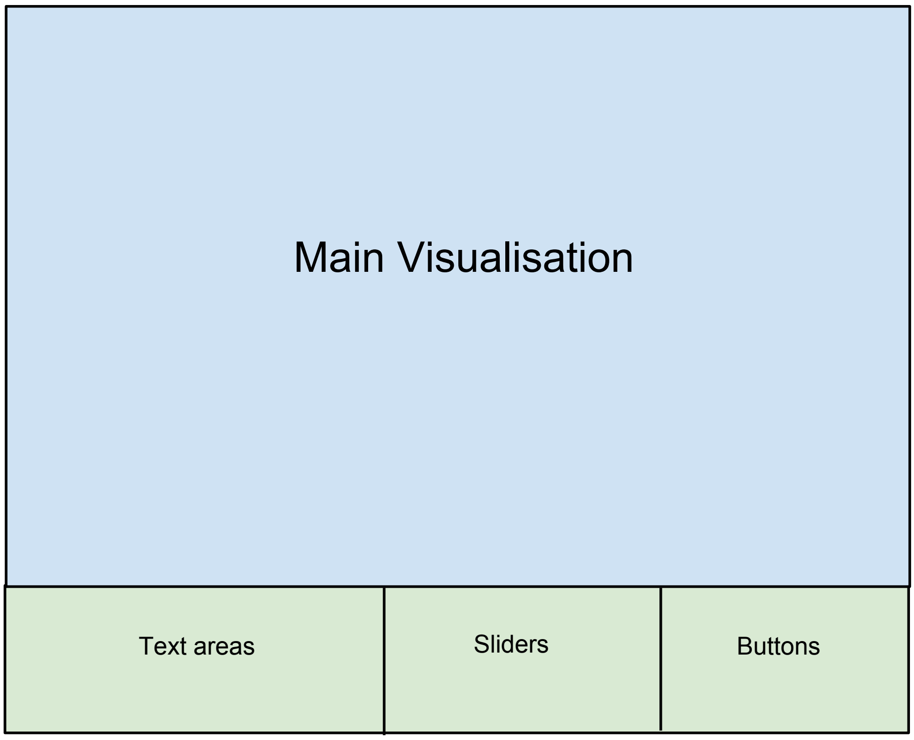

Figure 6.12: Layout of visualisation showing how each element is arranged to utilise grouping of similar components

R8. There needs to be two modes of interaction in the system, keyboard & mouse vs gesture based.

Incorporating a novel interactive method using a Microsoft Kinect sensor gives users an alternative method of controlling the visualisation. This requires incorporating a means of detecting the gestures of users and linking them to an action in the visualisation. A simple way to do this is to provide an area for the user to gesture to on the screen that controls the movement of the camera in the visualisation as shown in Figure 6.13.

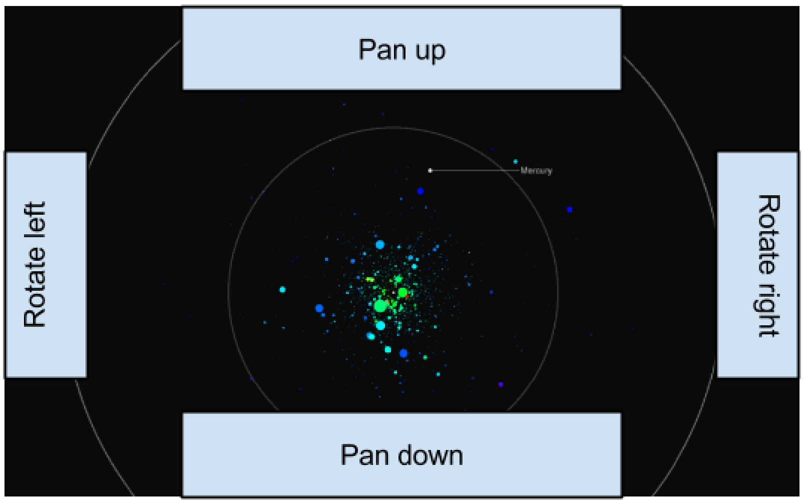

Figure 6.13: Example of the Kinect system showing where the user gestures to for moving the camera in the visualisation

Users also need to be able to zoom in and out of the visualisation as well as select exoplanets for further examination. To do this users should be able to push and pull their hand from the screen to zoom in and out, as well as hover over a planet to make a selection. Providing feedback to users is important in keeping them informed about the state of the visualisation, this means incorporating new cursors to indicate the gestures that have been detected. There are seven states that the cursor needs to be able to display to inform the user of what action they are performing. These states are

1. **Default cursor, hand is at rest or hovering over a planet**
2. **Panning up, hand is raised**
3. **Panning down, hand is lowered**
4. **Rotating left, hand is to the left**
5. **Rotating right, hand is to the right**
6. **Zooming in, hand is pressed forward**
7. **Zooming out, hand is pulled backwards**

Having a range of icons that clearly display these states is vital for keeping the user informed of what they are doing. The proposed cursor icons for this are displayed in Figure 6.14

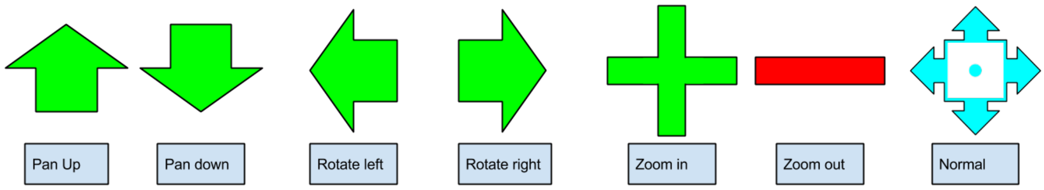

Figure 6.14: Example of cursor images for Kinect system to inform user of the gesture they are performing.

No gesture design has been included for interacting with the control panel. This is because the Microsoft Kinect sensor implementation in this visualisation is intended as a proof of concept that interaction via gestures allows users to access the information within the system. If the user study discussed in Chapter 8: Visualisation Evaluation shows that gesture interaction is successful in this regard then incorporating it further could be undertaken. The solution designs discussed above were used for the implementation of IKVT and are discussed in the next chapter.

Chapter 7

Visualisation implementation of the Improved Kepler Visualisation Tool (IKVT)

This chapter discusses the implementation of the visulisation using the designs from Chapter 6 to fulfill the requirements of this project. It details the tools used, the deliverable features produced, and the problems encountered.

7.1 Tools and Artifacts Used

7.1.1 Keyboard & Mouse System

In addition to Processing there was an additional open-source library required for effective user interface components called ControlP5 [2]. This library provides customisable and intuitive user interface components. It allows for the creation of visually appealing and precisely aligned interactive GUI components.

7.1.2 Microsoft Kinect System

For the Microsoft Kinect version of IKVT two additional libraries were required to integrate the Kinect sensor with Processing, these were:

1. NITE [7].
2. SimpleOpenNi [9].

These libraries provided drivers to run the Kinect sensor in Processing (SimpleOpenNi) as well as basic gesture recognition and body tracking (NITE). However these libraries were opensource options due to the official Microsoft Kinect SDK not being compatible with Processing. This meant the gesture recognition was not as user friendly or effective as the official libraries. The effect of this was that the gesture tracking used in the system had to be created sup-optimally from the open-source libraries which took more time and resources.

7.2 Implementation of IKVT

IKVT displays all 2234 exoplanets in the Kepler exoplanets dataset [34]. Each of these exoplanets are represented as coloured ellipses, of which the colour and size are representative of the exoplanets temperature and size respectively. IKVT displays all of these exoplanets as if they are orbiting a single star. In reality this would result in planetary collisions, but in this visualisation provides users with a way to effectively make observations and comparisons about each of the exoplanets in a single view.

There are two panels that make up the visualisation: the visualisation panel, and the control panel. The visualisation panel is where the exoplanets are displayed along with a text box describing the state of the visualisation to keep the user informed. The control panel contains all of the interactive components that the user can use to change the state of the visualisation. The components it contains are; two text areas that can be used interchangeably to display information about selected planets (Requirment 1), four range sliders (Requirment 4) shown in Figure 7.1 that are used to filter the exoplanets as discussed previously, and eight buttons to toggle the state of the visualisations. These buttons are “Sort by KOI”, “Sort by Temp”, “Sort by Size”, “Sort by ESI”, “Change View”, “Suns Habitable Zone”, “Pause”, and “Unsort” as shown in Figure: 7.2.

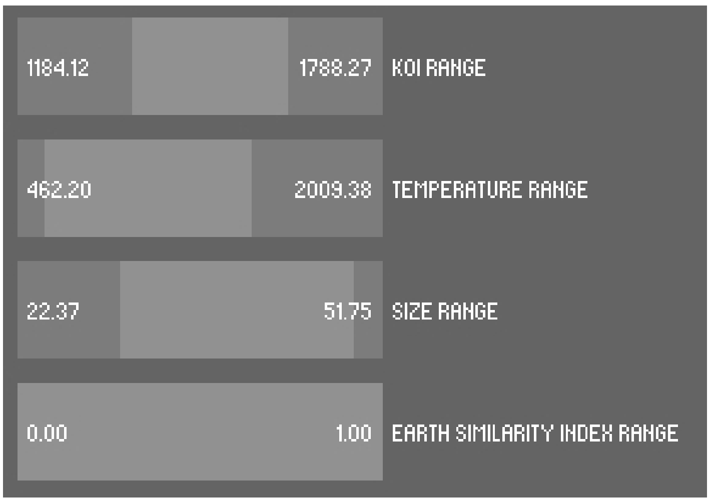

Figure 7.1: Implementation of interactive range sliders a various stages of filtering. The light grey bar in the center of the sliders is the range of the attributes selected to filter by. These can be modified by clicking and dragging on either end of the filter to change the ranges, or selecting the middle to move the selected range up and down the continuum

Figure 7.2: Implementation of interactive buttons, each one of which corresponds to a key functionality in the visualisation

Detailed information can be accessed about each exoplanet by clicking on them in the main visualisation window to make a selection. To do this a user can click on any of the orbiting exoplanets. The effect of this selection is that a text box will have further textual information about the selected exoplanet appended to it. In addition to this users have the option of clicking a button labeled “Compare”. This allows the user the option to select another exoplanet so that its information is appended to a second text area (Requirment 2). This can be seen in Figure 7.3.

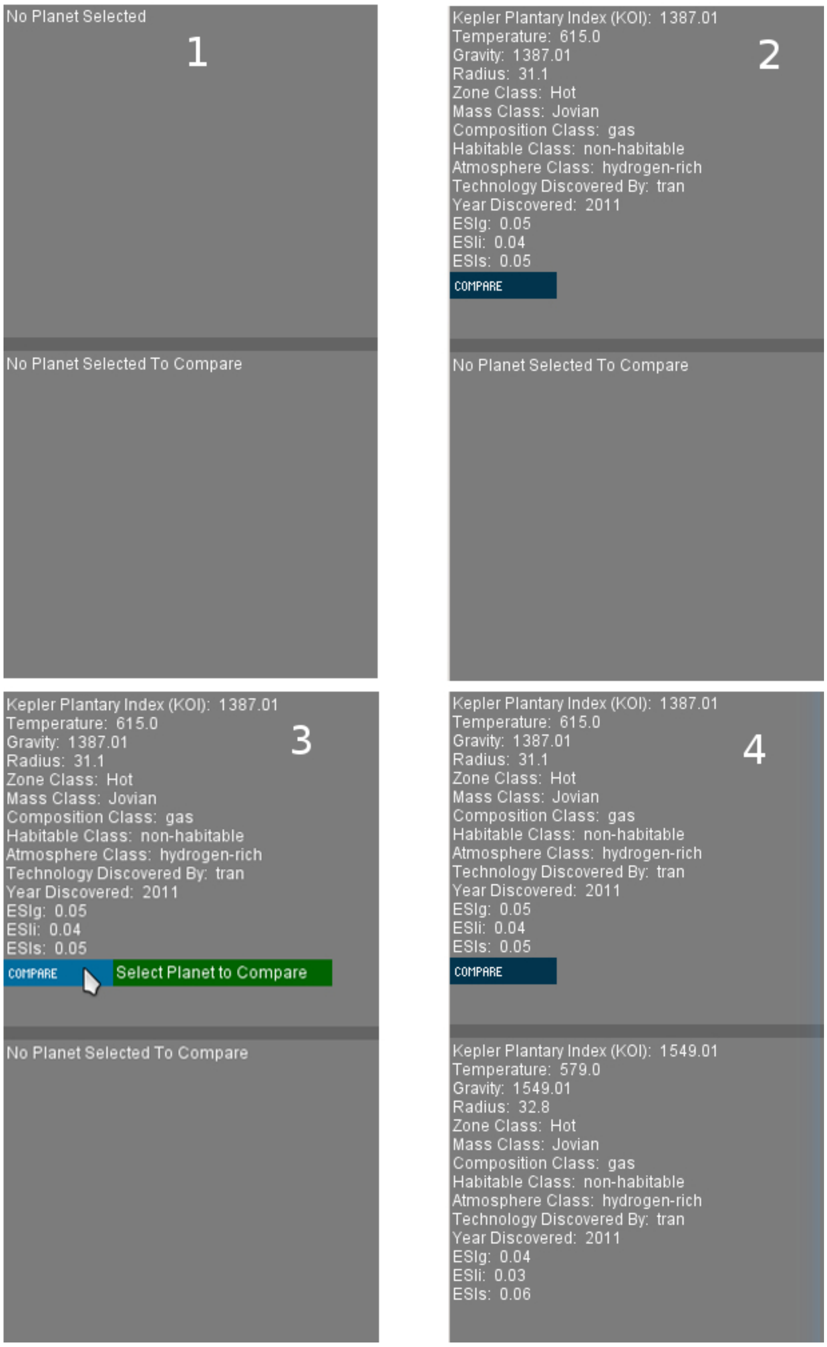

Figure 7.3: Implemented text areas in each possible state, Stage 1: No selection has been made so boxes are empty, Stage 2: A single exoplanet has been selected and its information is displayed as well as the option to compare, Stage 3: The user has chosen to compare the selected exoplanet to another, Stage 4: A second exoplanet has been seleced and its information added to the second text area.

When a user is unable to accurately select an exoplanet due to the clustering or overlapping of exoplanets they can move the camera around in space to gain a better viewing position. If this is not enough, the user can use a set of range selectors to filter the exoplanets displayed, as shown in Figure 7.4. The exoplanets can be filtered by their KOI, temperature, size, and ESI. These filters allow for users to fine tune the exoplanets they wish to view, thus allowing them to work with small multiples rather than the entire dataset. Using these filters also causes a zooming effect to occur as exoplanets are filtered out. This zooming occurs each time the filters are changed as the exoplanets spread out vertically and allows more space between them to make selections.

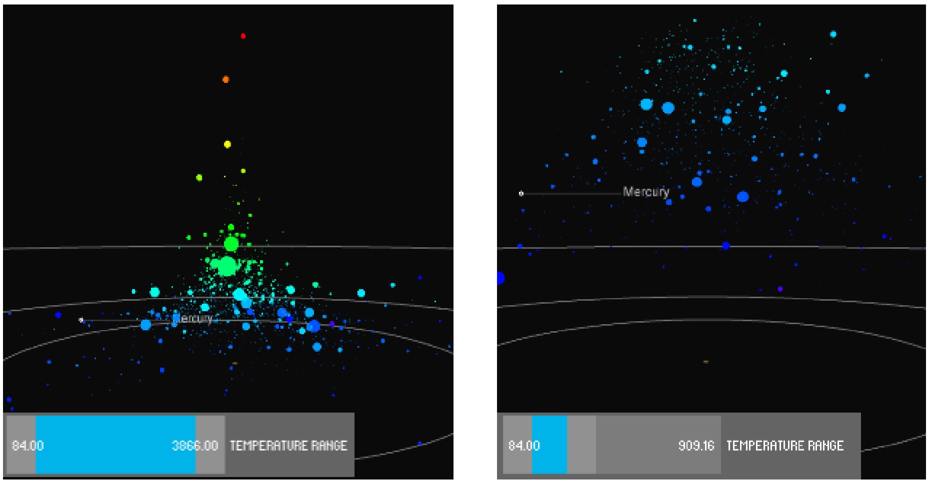

Figure 7.4: The above images display the zooming effect provided by filtering. In the left image we can see that the filters have not been used yet and all exoplanets are present which causes clustering. In the right image the range has been shrunk to filter out hot exoplanets. This causes the colder planets to take up the space that was originally occupied by all exoplanets which allows them to spread out.

The existing Kepler Visualisation Tool allowed users to sort the exoplanets on the Y axis by their temperature and size. In IKVT this has been extended to allow sorting by KOI and ESI (Requirment 3)as shown in Figure 7.5.

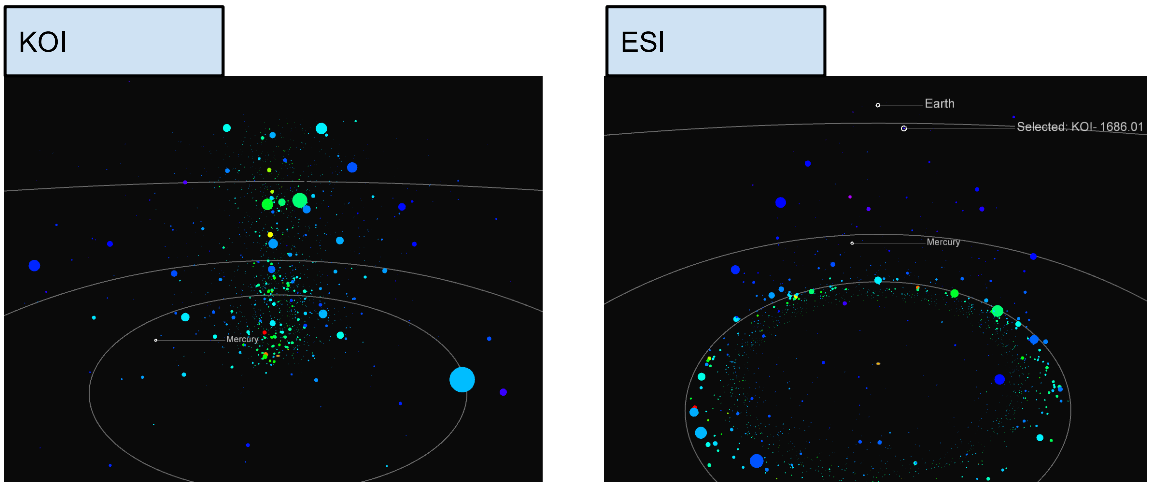

Figure 7.5: The above images display the implementation of sort by KOI and ESI. The left image shows the exoplanets sorted on the Y axis by their KOI, this allows users to get an idea of which exoplanets share a solar system as they will have similar KOI values. In the right image exoplanets are sorted by their ESI on both the Y and X axis so that planets closer to the center and the top are most similar to Earth

When a exoplanet is selected it is important that the user gets feedback about what they have done. In IKVT this happens in the form of the selected exoplanet becoming larger and a white outlining ring expanding out to encircle it. In addition to this a label appears to the right of the exoplanet stating the KOI.

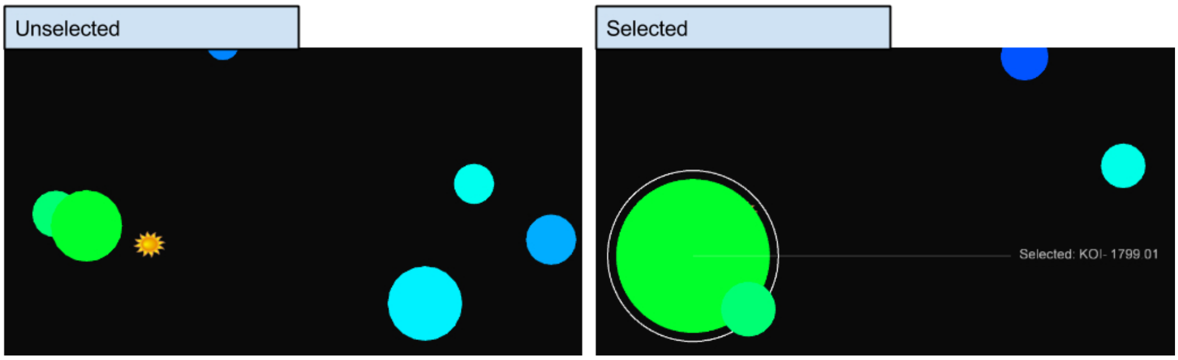

Figure 7.6: The above images display the implentation of the exoplanet selection process. We can see the result of an exoplanet being selected, first the planet expands, then a white circle expands out to outline it and a label appears to inform the user about the exoplanet they selected.

Another effect of a user making a selection is that all exoplanets that share the same star as the selected exoplanet also become highlighted in the same way as outlined above. The only difference is that in the label to the right of each exoplanet the message now informs the user that it has the “Same Star” as displayed in Figure 7.7.

Figure 7.7: This image displays how the planets in the same solar system as a selected planet become highlighted in the same way but with a different label.

To display the Goldilocks zone of selected exoplanets coloured rings appear to display the habitable and inhabitable zones of the star (Requirment 5). Each time a new planet is selected the coloured rings change to represent the newly selected exoplanet’s star, this can be seen in Figure 7.8. In addition to the coloured rings the selected planet also becomes highlighted and all of the other exoplanets become transparent making it stand out to help users understand whether it is in a habitable zone or not.

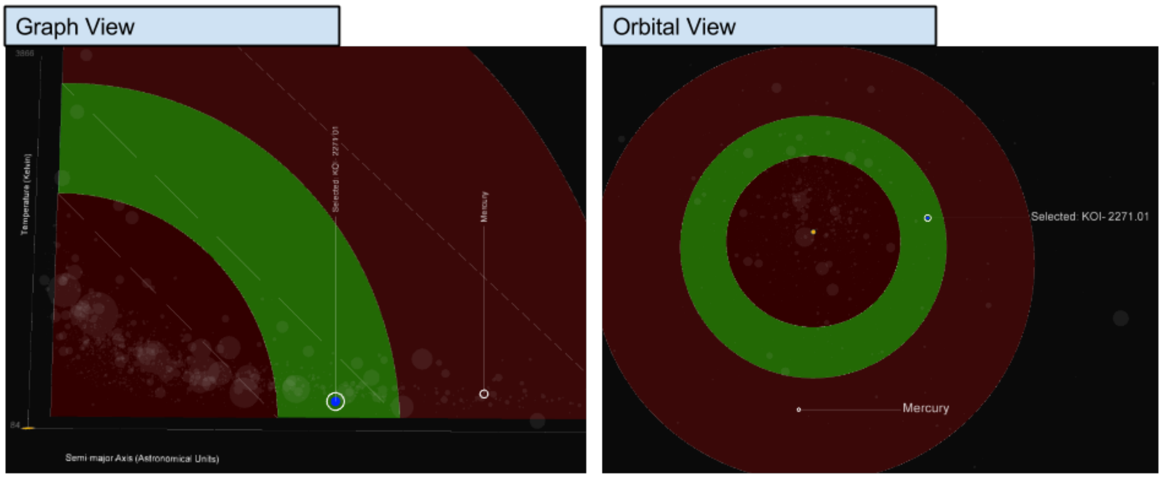

Figure 7.8: This image displays the implementation the view Goldilocks zones feature which show how the coloured rings are used, and how the planets relation to them infers their habitability.

Each of these implemented elements need to be visually apparent and intuitive to use to ensure that the system can be easily used without prior experience. This is done by providing clearly labeled interactive elements and tooltips explaining what they do as in Figure 7.9. These tooltips are widely used as a method of informing a user about the purpose of an item by hovering over it removing the need to click on a button to discover its effect.

Figure 7.9: This image displays the implemented tooltips which allow users to hover over interactive elements to discover what they do which removes the need to click on a button to discover its effect

Due to the need for effective user interaction with the visualisation, a window is required to house all of the components discussed above. Having this window spatially separated from the main visualisation window means that users will not be drawn away from the visualisation by overlapping and intrusive controls. Each of the controls discussed above are included in this panel as shown below in Figure 7.10.

Figure 7.10: Shows the implemented Control Panel that contains all of the interactive components discussed previously.

At first the visualisation was laid out so that the control panel took up a 300 pixel strip along the bottom of the screen with the main visualisation window taking up the rest of the space as in Figure 7.11. This was found to be unsuitable as the majority of the interaction and animation of visualisation elements occurs in the center of the window which this layout did not effectively utilise as it caused a low aspect ratio that made the top and bottom of the visualisation to be cut off. It was more effective to use two vertical columns as in Figure 7.12 to view and control the visualisation. This is because the higher aspect ratio allowed more of the content to be seen on the screen at once as the majority of computer screens have a wide aspect ratio. As you can see by comparing the two figures, Figure 7.12 cuts off less of the visualisation and so is a more suitable choice.

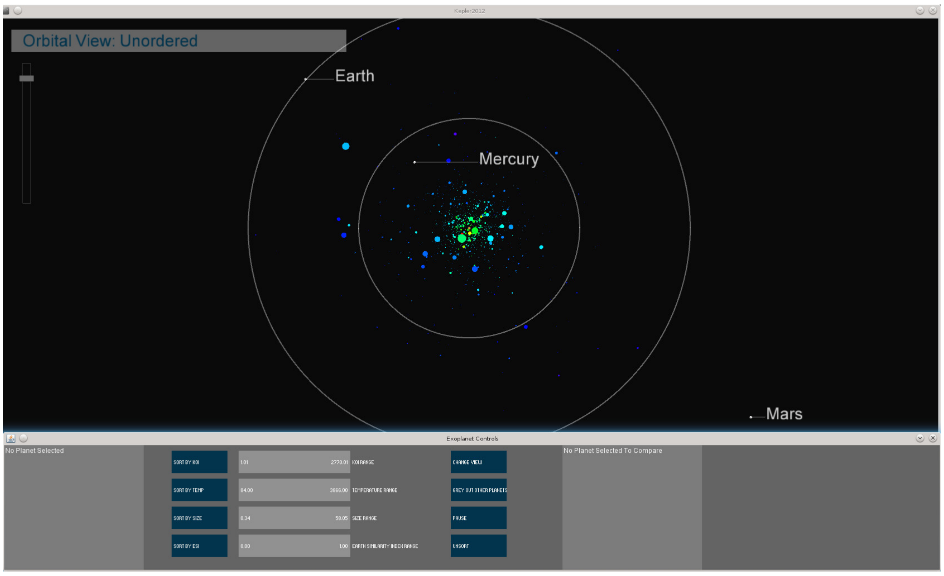

Figure 7.11: Original Horizontal Layout that had an unsuitable aspect ratio that occluded part of the visualisation.

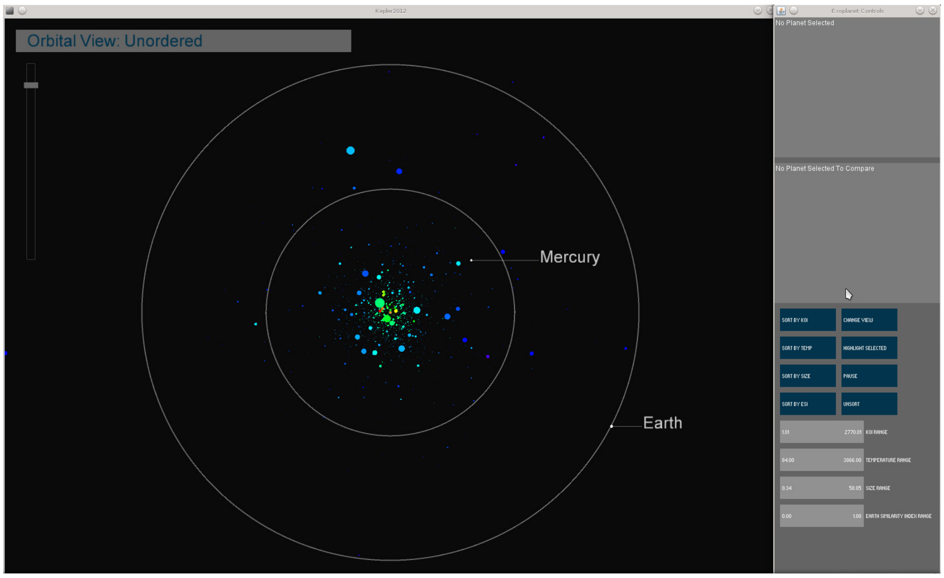

Figure 7.12: Improved Vertical Layout that had a higher aspect ratio that did not hide any of the visualisation.

The Kinect Sensor system uses all of the features discussed above and extends it by incorporating in gesture based control by utilising a Microsoft Kinect sensor. Incorporate these gesture based controls was completed by specifying gestures are detected by the visualisation which in turn modifies its state.

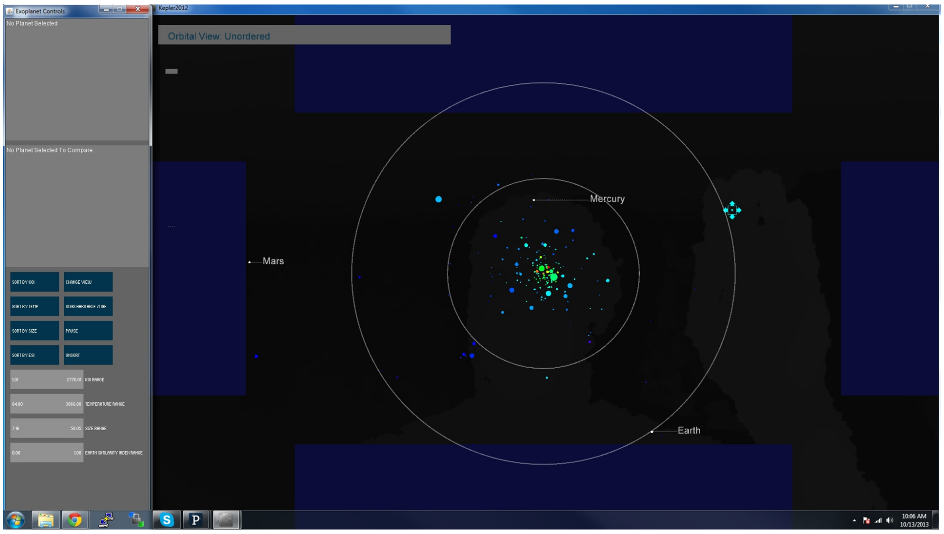

Figure 7.13: Kinect system visualisation window with the basic cursor visible on the outline of the users hand. Also visible are the blue areas on the outsides of the visualisation indicating how to perform gestures

As shown in the solution design stage there were eight gestures that were needed for the visualisation, each of which was implemented as described below

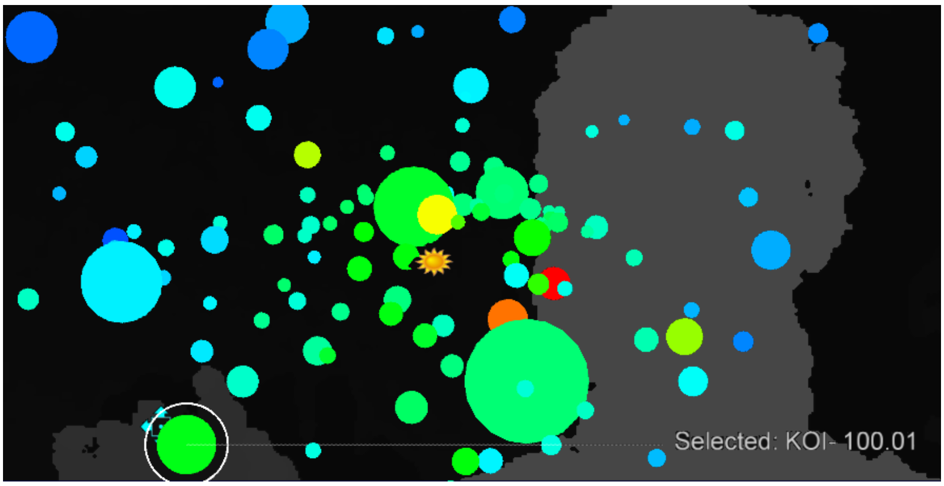

Figure 7.14: Selecting a planet by gesture by hovering a hand over it. The basic cursor appears behind the selected planet.

1. Default cursor, hand is at rest or hovering over a planet.

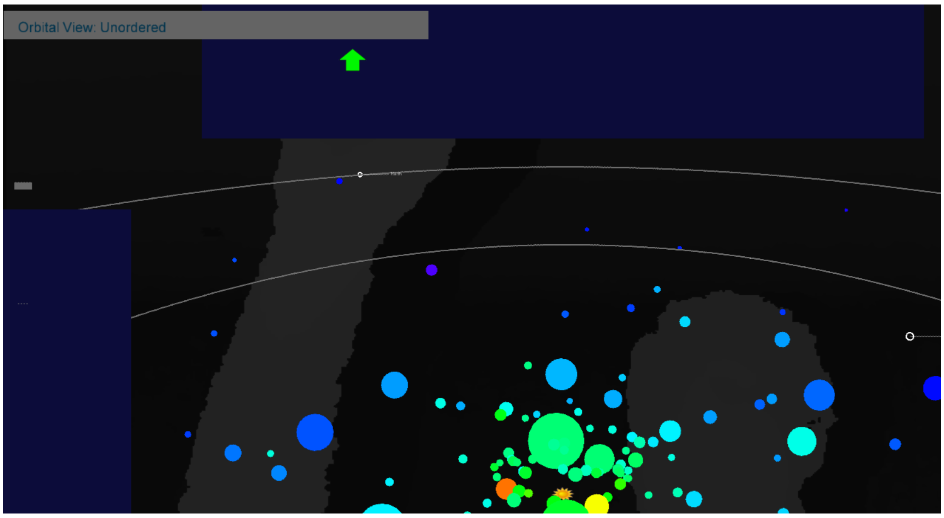

Figure 7.15: Panning the visualisation by raising hand to the top blue area. The cursor indicates the visualisation is panning up.

2 & 3. Panning up, hand is raised or panning down, hand is lowered.

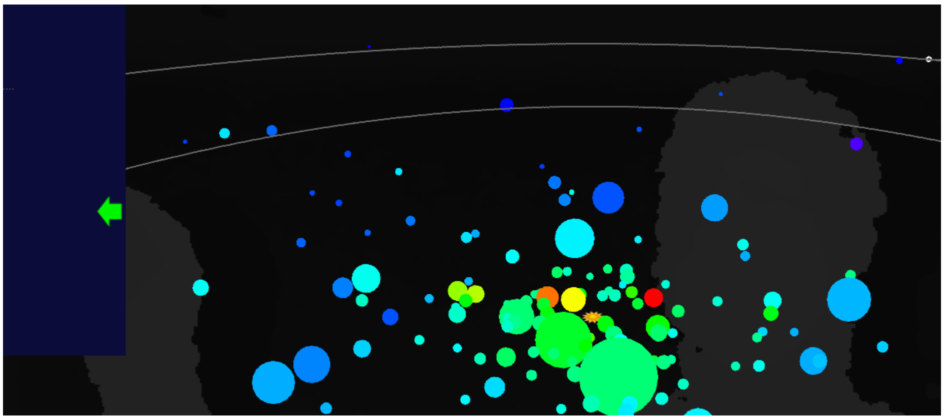

Figure 7.16: Rotating visualisation be swiping hand to the left of the screen. The cursor indicates the visualisation is rotating left.

4 & 5. Rotating left, hand is to the left or rotating right, hand is to the right.

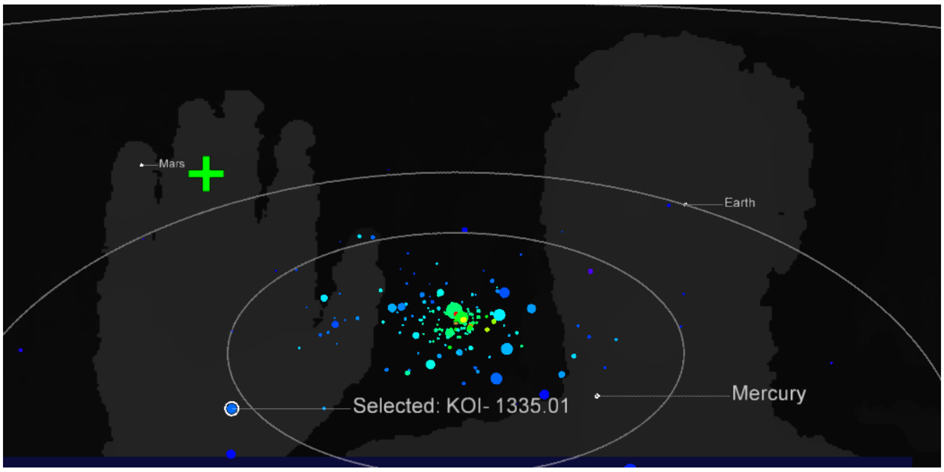

Figure 7.17: Zooming in by pushing hand towards the screen. The cursor indicates the visualisation is zooming in.

6. Zooming in, hand is pressed forward.

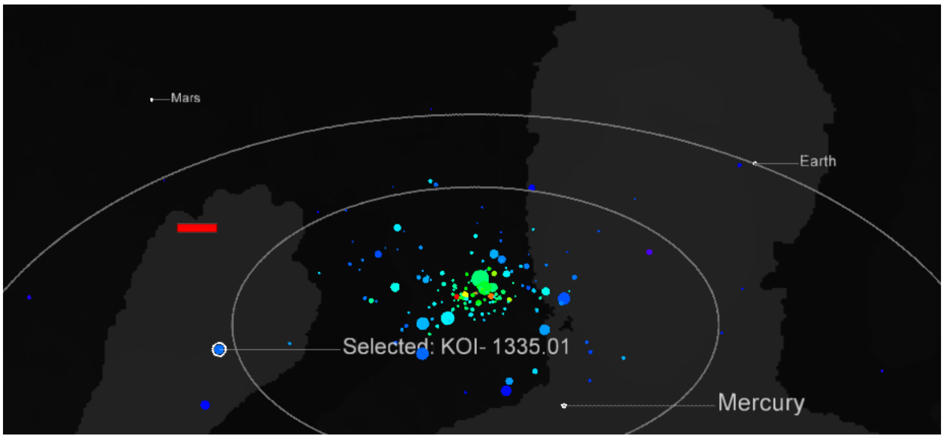

Figure 7.18: Zooming out by pulling hand from the screen. The cursor indicates the visualisation is zooming out.

7. Zooming out, hand is pulled backwards.

In addition to this, the screen displays the user in relation to the the visualisation, this is done by displaying a transparent outline of the user in the background of the visualisation as can be seen in the above figures. This is important as it gives the user a way to infer where they are gesturing and as a way to show them that they are being detected by the system.

Each of the features discussed were implemented using the designs created in Chapter 6 to fulfill the functional requirements. The visualisation IKVT that was created can successfully display planetary information to convey knowledge to users (Requirement 1), allow exoplanets to be compared against one another (Requirement 2), order exoplanets by their ESI and by their KOI (Requirement 3), allows users to define ranges of planetary attributes to filter which planets are displayed (Requirement 4), and view the habitable zones of stars in relation to the planets orbiting them (Requirement 5). These functionalities are evaluated in the next chapter to ensure that they successfully fulfill the functional requirements as previously stated, as well as the non functional requirements that emphasises usability for users.

7.3 Problems Encountered

As this project builds upon a previous system, much of the existing code and execution flow required modification. This required understanding of how the system was originally built and designed. Because this system did not have any unit or integration tests, going ahead without a comprehensive knowledge of the core functionality would be have led to ineffective planning and errors being introduced into the system. To address this, extra time and resources were allocated to the requirements analysis and solution design stages of the project.

Due to the number of elements that needed to be displayed on screen at any one time (i.e. 2234 ellipses to represent the exoplanets), the load placed on the system was very high. This uncovered a bug in the Processing library in which the memory use of the visualisation would periodically increase until it crashed due to an Out of Memory Exception. After much experimentation with how to overcome this issue, I discovered that rather than trying to render a native ellipse shape in processing, if I instead rendered a Scalable Vector Graphic this bug would not manifest.

Using the Processing framework meant using a non commercial IDE that is not completely robust, for example when undoing multiple times in a row the file being modified would periodically become corrupted by lines of code being taken away or inserted into the wrong locations. The solution to this issue was to ensure that any changes were committed regularly to my repository on Github [4]. Doing this meant that if at any time a file became corrupted the incorrect changes could be easily compared against the previous commit and manually fixed. Any bugs found were reported in the hopes that others using Processing in the future wouldn't need to address them.

The libraries used for gesture detection with the Kinect sensor were open-source as this was the only way to get it to work with Processing. These open source libraries did not have the functionality, documentation, and support that the official Software Development Kit (SDK) that is available from Microsoft has had. Whilst this was not a roadblock in terms of development it did mean that more time was required for research and experimentation during implementation.

Chapter 8

Visualisation Evaluation

Following the completion of the implementation of IKVT a final user evaluation was carried out. This evaluation was designed to discover whether the implemented visualisation successfully fulfilled the requirements of the project. The evaluation was a within subjects experiment conducted with 10 participants. This means that all participants were exposed to every condition in the experiment. Each participant took turns using the IKVT with both keyboard & mouse and Kinect sensor whilst answering a set of questions. The evaluation was designed to qualitatively and quantitatively assess users reactions and experiences with the IKVT. By mapping the experiences of users to the project requirements it was possible to evaluate the success of IKVT. The choice not to perform comparative evaluation against other existing visualisations was made for three main reasons; (1.) new features were being implemented that no previous visualisations had, (2.) the inclusion of interaction via gesture was intended as a proof of concept and so users experiences were important, and (3.) The goal of the evaluation was to evaluate IKVT against the project requirements.

8.1 Purpose of Evaluation

User studies offer a scientifically sound method to measure the performance of a visualisation. Studies can be used to evaluate the strengths and weaknesses of different visualisation techniques or show that a visualisation technique is useful in a practical sense, according to some objective criteria, for some specific task [24]. The fundamental goal of conducting user studies is to seek insights into why particular visualisation techniques are effective [24].

8.2 Experimental Design

8.2.1 Expectations of Evaluation

An expectation in this evaluation was that users would take approximately three minutes to feel comfortable with using the visualisation for the basic tasks of moving the camera around, sorting the exoplanets, and using the range sliders to filter the exoplanets.

Following this acclimitisation time it was expected that users could accurately complete the set of questions in a worksheet (APPENDIX QUESTIONS) whilst using the visualisation. During this stage users were expected to use both interaction methods (Keyboard & Mouse, and Kinect sensor). It was predicted that during the keyboard & mouse portion of the experiment the users should make more accurate selections and exhibit more effective data seeking behavior, whereas during the Kinect portion they would be more interested in experimenting with gestures rather than attempting to gain information about exoplanets.

When users have finished using the visualisation and fill in the questionnaire asking them about their experience. The expectation was that they would detail the areas of the visualisation that they had trouble with, felt needed improvement, and that they found useful and fun. These two forms of results gathering provided both qualitative and quantitative results describing the users experiences. The expectation was that these qualitative and quantitative user results would provide a means of viewing how successful the visualisation was in terms of fulfilling the system requirements.

8.2.2 Participants

The user study was undertaken by 9 participants referred to as P2 through P10 as well as a pilot study by a single participant referred to as P1. All were either students or young professionals from a mix of specialties, aged between 21 and 26 and a mix of genders with 5 females and 4 males. 6 of these participants had extensive prior experience with Kinect sensors, and all participants had experienced a 3D visualisation before. 6 out of the 10 participants had knowledge of the exoplanets discovered by the Kepler telescope. These are shown in the following table in Figure 8.1

Users	Gender	Previous Experience with Kinect	Previous Experience 3D Visualisations	Previous knowledge of exoplanets
P1	M	Yes	Yes	Yes
P2	F	No	Yes	No
P3	M	Yes	Yes	Yes
P4	F	Yes	Yes	No
P5	M	Yes	Yes	Yes
P6	F	No	Yes	Yes
P7	F	No	Yes	No
P8	M	Yes	Yes	Yes
P9	F	Yes	Yes	Yes
P19	M	Yes	Yes	Yes

Figure 8.1: This table displays the attributes of each of the participants in the evaluation.

8.2.3 Evaluation Environment

The environment of the evaluation was set in a closed computer laboratory of twelve computer stations. One duel monitor PC was set asside for the evaluation. This PC had an i7 processor, 8gb ram, and was running KDE [5] as the opperating system.

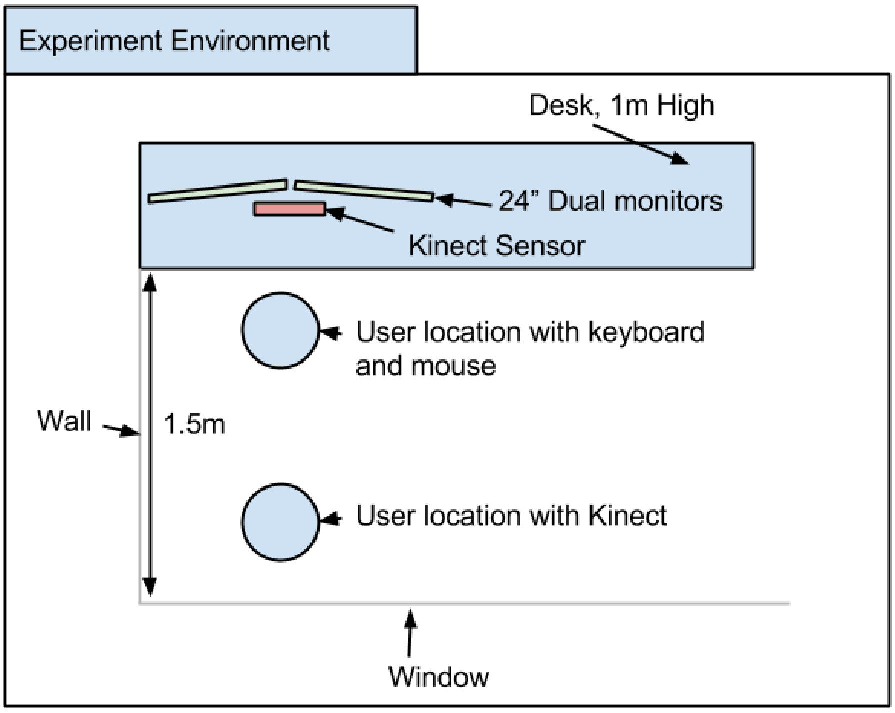

Figure 8.2: Image of evaluation environment showing the seating positions and hardware locations.

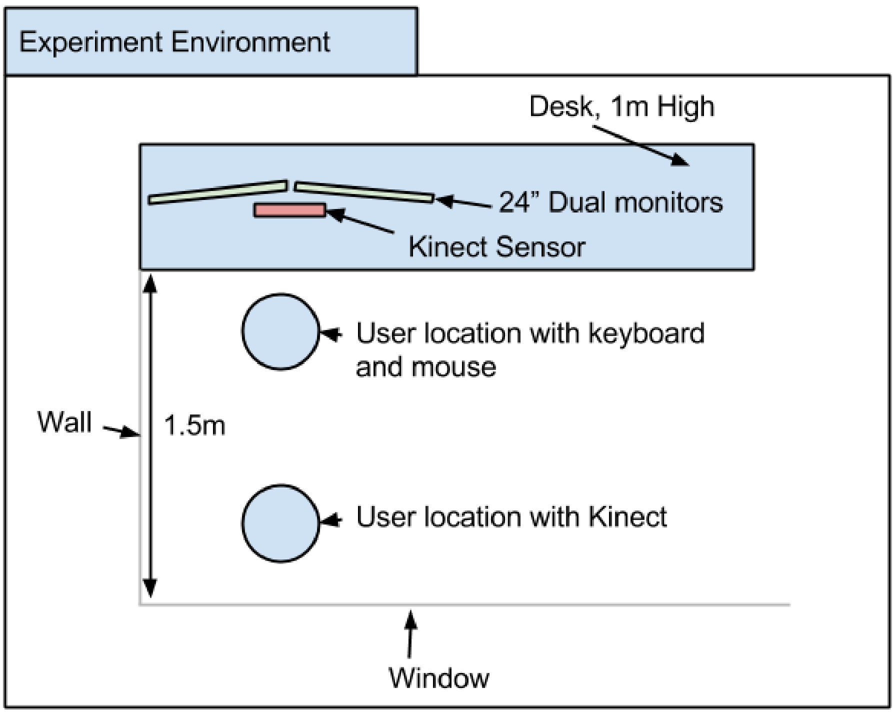

Figure 8.3: Evaluation environment plan showing layout of all hardware components.

8.2.4 Evaluation Method

A poorly designed experiment will only yield results of limited value, because of this it was important to ensure that each stage of the experiment was focused on evaluating a specific area of the visualisation in relation to the project requirements [32]. There were three key methods of gathering results during this evaluation; a worksheet to fill out whilst using the visualisation made up of two sets of questions (one for the keyboard & mouse system and one for the Kinect system)(APPENDIX), a questionnaire to fill in afterwards about the experience (APPENDIX), and the examiners observations about how the users interacted with the system.

The following are the steps that were carried out during each user evaluation to ensure that the variables were the same each time

1. The participant enters the room and sits down at the computer.
2. They are handed the consent form and information sheet.
3. After these are completed they are handed the user questionnaire and the set of questions to answer while using the system. On this questionnaire there are two sets of questions, the first is for the keyboard & mouse system, and the second if for the Microsoft Kinect system.
4. They are then given a brief introduction into each of the visualisation components and what the visualisation as a whole represents
5. Following this they are advised that they have 5 minutes to get familiarised with the system but they do not need to use all of this time (the amount of time taken will be recorded for analysis of how user friendly and intuitive the system is).
6. Following this the participant is asked to complete the worksheet by first using the mouse and keyboard system. When they feel they have answered the first set of questions they will notify the examiner who will move the user to the Kinect system to continue with the second set.
7. Once the participant has completed both sets of questions they are asked to fill in the qualitative user questionnaire about their experiences using the visualisation.
8. Following this if the examiner has no follow up questions the user is free to leave.

8.2.5 Pilot Study

Due to the significant financial and resource costs associated with running an experiment, it is valuable to conduct a pilot study with one or two participants. This allows testing and refining of the experimental design before starting a full-fledged study with numerous participants [24].

The reason for conducting a pilot study for this project was to ensure that the experiment was producing the data needed to evaluate IKVT as well as taking the correct amount of time to complete. It was also used to discover whether any aspects of the study would interfere with the results. One participant was asked to take part in a pilot study, this participant is referred to as P1. P1 was asked to complete all of the activities that make up the main experiment. This pilot study took approximately 15 minutes as intended including the time needed for the explanation and completion of paperwork, as well as the experiment itself.

P1 successfully completed each of the questions in the worksheet whilst using the visualisation with only limited assistance from the examiner. This assistance was required due to the wording of some of the tasks being ambiguous causing unnecessary confusion which could have interfered with the results. These ambiguous questions and tasks were removed prior to the main user study. During the main study only P4 and P10 asked for clarification on the questions or tasks.

P1 had some initial difficulty using the range sliders but after a small period of experimentation began using them for the majority of tasks which turned out to be effective. P1 also found that whist the range sliders were useful, they did not allow fine enough control for making small changes to the filters. The component of the visualisation that P1 had the most difficulty with was using the Goldilocks view. This difficulty seemed to stem from the lack of a common point of reference for each planet due to each planet having a different star with its own Goldilocks zone.

During the Kinect portion of the experiment P1 found that being in a sitting position whist interacting with the visualisation did not feel natural due to "being required to reach out and exert effort to hold an upright position of the arms for an extended period of time". The following figure displays P1s quantitative results from the questionnaire.

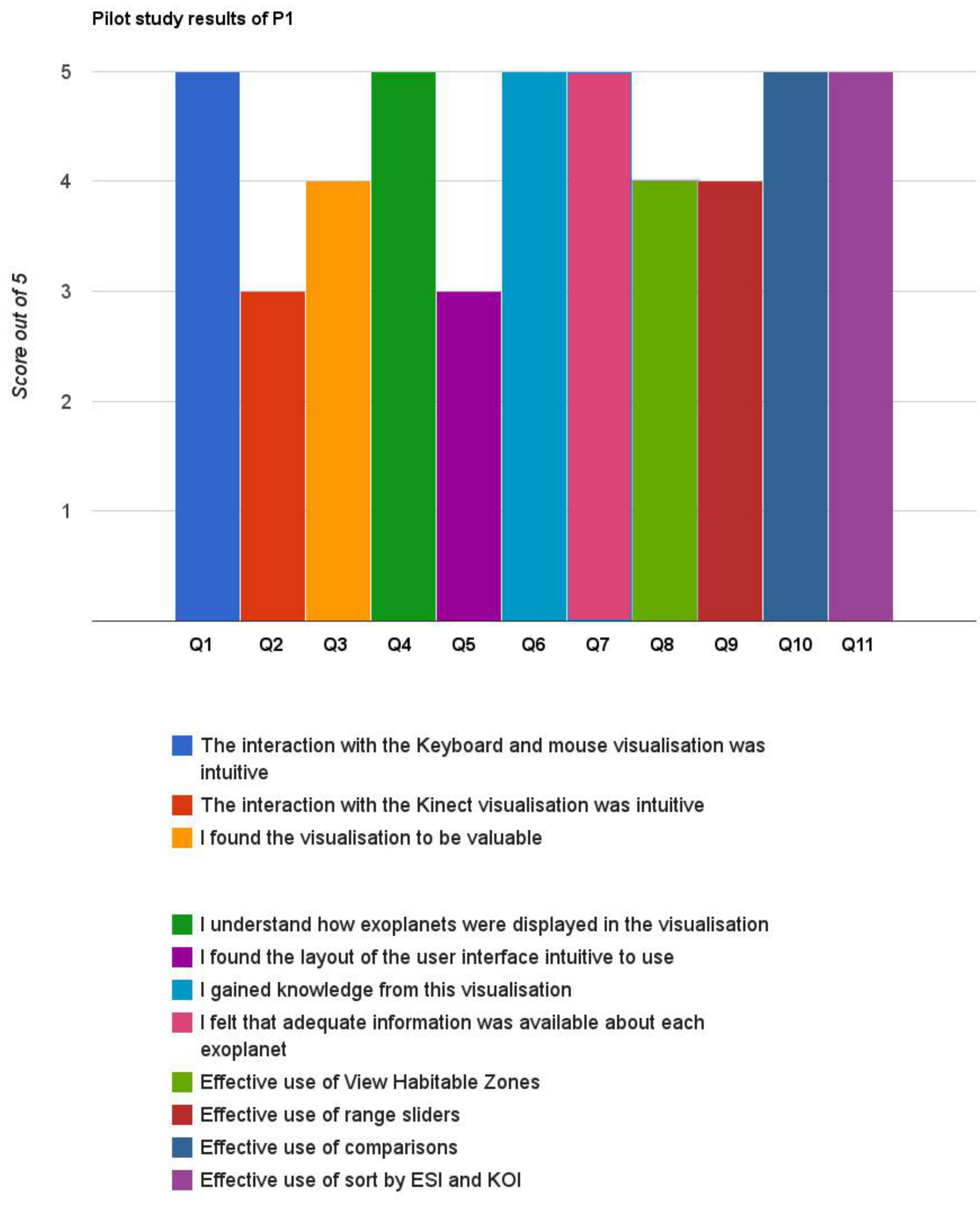

Figure 8.4: Pilot study quantitative results of P1. Points of interest in this graph are that Q2: The Kinect visualisation was intuitive, and Q5: I found the layout of the user interface intuitive to use, both scored lowly but The interaction with keyboard and mouse was intuitive and P1 was still effectively used the visualisation.

8.3 Results of Evaluation

The keyboard & mouse portion of the experiment was primarily intended to evaluate how the interaction techniques introduced into the visualisation aided users in their information seeking behavior. This was evaluated through the first part of the worksheet filled in by users whilst using the visualisation. These questions were designed in a way that encouraged users to make use of each of the interactive features that was implemented as part of this project.

The Microsoft Kinect portion of the experiment was intended to evaluate how users would react to interacting with the visualisation by gesture. The worksheet for this portion of the evaluation were designed to find out whether users could successfully navigate the visualisation without keyboard & mouse. The following subsections discuss the qualitative and quantitative results of this evaluation.

8.3.1 Qualitative Results

This subsection discusses the qualitative and quantitative results returned by the user evaluation.

Evaluation of Functional Requirements

R1. The visualisation needs to display planetary information to convey knowledge to users.

The experiment demonstrated that the increased amount of information available in IKVT about each exoplanet allowed participants to access more information contained in the Kepler Exoplanet database than in the previous system. All participants were able to successfully complete the worksheet questions that involved accessing information about exoplanets stored in the text areas in the control panel. P8 found that a weakness of the system was the amount of concepts that needed to be understood about exoplanets and so including a glossary would be an improvement. A glossary would allow users to discover the meaning to any terms in the visualisation mitigating the risk of user confusion.

All participants reported learning something from the visualisation that they did not previously know, and those who had no knowledge of exoplanets prior to the experiment felt that they had learned something valuable. For example P4 felt that using IKVT broadened her perception of how much information we know about planets so far away.

R2. The visualisation needs to allow exoplanets to be compared against one another.

Whilst all participants used the comparison functionality during the familiarisation stage, only three of the participants (P3, P8, P10) used it while completing the worksheet. This could be because it was not made obvious enough, or because the questions asked of users did not force users to use this feature to get an answer. When the participants did use the comparison tool they were able to use it successfully and most tried it multiple times with multiple exoplanets. P3 found that being able to compare the exoplanets against one another was useful as he could leave an interesting exoplanet selected whilst comparing it against multiple other exoplanets. He found that it was easy to compare the information for each exoplanet and to get a sense of the attributes of exoplanets.

R3. The planets need to be able to be ordered by their similarity to earth (ESI) and by their Kepler Object of Interest number (KOI).

All participants used the sort by KOI and ESI functionality effectively without any confusion. P8 liked being able to toggle between the different attributes sortings as he found that it gave a clearer view of the attributes in a visual manner rather than needing to look at the text areas. P7 found the exoplanets sorted by ESI to be the most interesting view. She spent an extended period of time experimenting with this during question 8 of the worksheet before moving on the later questions. P7 stated that she could find the potentially habitable planets easily as she could see that they were similar to Earth which was displayed clearly.

R4. The visualisation needs to allow users to define ranges of planetary attributes to filter which planets are displayed.

All participants successfully used the range sliders whilst completing the worksheet questions. The range sliders were used frequently by all participants to discover the exoplanets that were the outliers in regard to their attributes (i.e. high temperatures). 3 out of 10 participants had trouble when moving between questions on the worksheet due to forgetting to reset the range sliders after using them and thus only a subset of the exoplanets were displayed. P5 found the zooming effect that occurred when planets were filtered useful for making more accurate selections and spent time experimenting with it. P6 found that being able to sort the planets according to attributes and then filtering them to remove the planets she did not want was a powerful tool.

R5. Users need to be able to view the habitable zones (Goldilocks zones) of stars in relation to the planets orbiting them.

The evaluation of this requirement showed that whilst the functionality was successfully implemented it lacked usability. All but one user in the study found that this was because it was unintuitive. What caused this was that the Goldilocks zone of each star is different. This meant that each time a user selected a new exoplanet the location of each of the zones changed as was intended. This confused all but one user (P10) as they were expecting the zones to stay the in the same locations.

Evaluation of Nonfunctional Requirements

R6. All interaction methods must be visible and intuitive.

The experiment found that all interactive methods were able to be seen and used by users. However due to the design of the components being low key as to not draw attention away from the main visualisation, the participants often forgot about them, especially as the control panel was at the side of the screen. For example, P5 found that because the information panel was not central in the visualisation it required effort to break away from the main visualisation window and thus broke the immersion. Another issue found was that the text of the components in the interaction was too small for some users to see clearly and quickly. A common consensus was that the layout of the interactive components were fine but could be improved by making them "pop out more" (P10) especially as many participants didn't initially spot the text areas changing as planets were selected until reading the worksheet questions.

R7. The visualisation must remain uncluttered to reduce information overload.

All users felt that the amount of information displayed in the visualisation was appropriate apart from P7 who felt it displayed to much information at once which caused confusion. P10 felt that the control panel contained the right amount of information, but felt that it could be improved by changing its design to make it stand out more and emphasis each component contained in it. P4 felt that the names of some of the attributes were not made clear enough (E.g. ESI, KOI). P5 felt that limiting the amount that the camera could move would be an improvement, especially in the graph view where their is a fixed plane where content is displayed.

R8. There needs to be two modes of interaction in the system, keyboard & mouse vs gesture based.

Figure 8.5: Participant using Kinect visualisation performing selection of an exoplanet

The experiment found that both the interactive mediums had different strengths and weaknesses. The keyboard & mouse system was the most effective for information seeking as it provided more accuracy and was easier to use the control panel elements. The Microsoft Kinect system was worse for discovering information, but it was the most fun out of the two options due to its novelty. All of the users felt that had they had more time with the Kinect system, they would have been more effective at accessing the information available. This was because they were preoccupied performing basic actions like moving the camera and selecting planets due to the short time available. For example P6 found that she used more time playing around with the Kinect system trying to select planets rather than trying to get information. All participants but P4 felt that the visualisation responded with the appropriate actions and magnitudes for each gesture. P4 expected the magnitude of the visualisation changes to be more than they were, ie it should have rotated faster. P6 felt that there needed to be more physical movement space in which to use the Kinect as some movements felt cramped. Another weakness discovered about the Kinect sensor was that it detected the whole hand rather than a more controllable area like a finger. P10 found that overlapping planets made it hard to make selections with the Kinect and that he needed a way to cancel a selection without having to select another planet.

8.3.2 Quantitative Results

Evaluation of Functional Requirements

A key part of this visualisation is that it should be intuitive enough that a user can walk up to it and feel comfortable using it to explore data within a short period of time. The results showed that some users felt that they could effectively use the visualisation very quickly (within 1 to 4 minutes) whilst others took longer (up to 5 minutes) as Figure 8.6 shows.

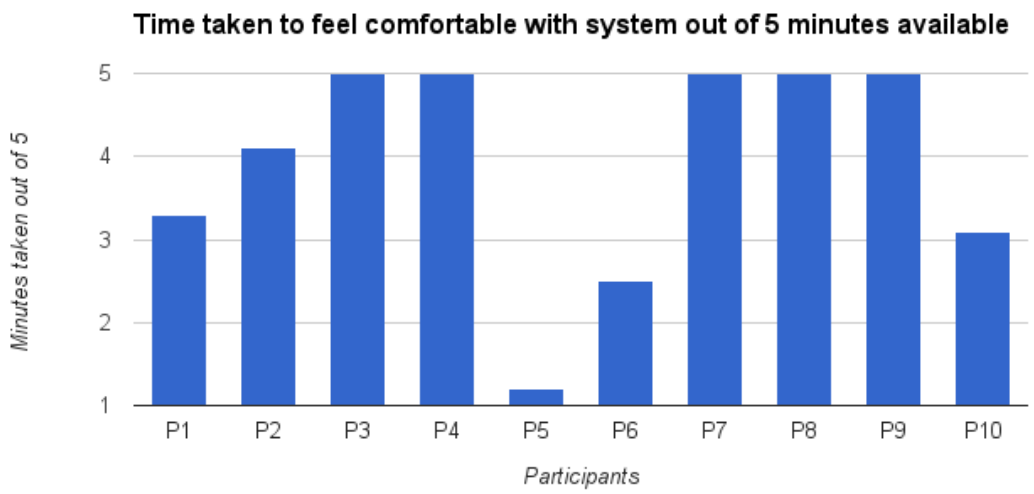

Figure 8.6: Amount of time taken out of a possible 5 minutes for users to feel comfortable with visualisation before moving onto the worksheet. A notable reult from this is that even though P5 took the shortest amount of time, he was also very successfull while using the visualisation as shown in Figure 8.8. This could mean that some users will naturally find the visualisation easier to use than others like P4 who took the full amount of time and still had trouble IKVT

As we can see below in Figure 8.7, all users performed well in all aspects apart from with “Req: 5 Effectively used habitable zones” which the qualitative analysis also found. Another notable result was that the keyboard & mouse was rated as more intuitive than the Kinect sensor, again this is supported by the qualitative results. Apart from these key areas of interest the results are as expected with each area being scored highly.

	1(worst)	2	3	4	5(best)
Req 1: I gained knowledge from this visualisation				3	6
Req 1: I found the visualisation to be valuable			2	3	4
Req 2: Effectively used exoplanet comparisons			1	3	5
Req 3: Effectively used sort by ESI and KOI				3	6
Req 4: Effectively used range sliders			1	4	4
Req 5: Effectively used view habitable zones	7		1		1
Req 6: Comprehended how planets were represented			2	4	3
Req 6: GUI layout was intuitive		2	1	6	1
Req 7: Correct amount of information was displayed			1	4	4
Req 8: Keyboard and mouse was intuitive				4	5
Req 8: Kinect sensor was intuitive		1	3	5	

Figure 8.7: Summary of the quantitative results gathered from participants. Notable points are the negative responce surrounding Requirement 5 which matches the qualitative results. Apart from this all other areas scored highly.

Seeing how different participants scores attribute to the table in Figure 8.7 helps to understand the distribution of scores. As Figure 8.8 shows, some users consistently gave higher or lower scores which could have caused a bias in the results. However it does show that some users enjoyed using the visualisation whilst others did not.

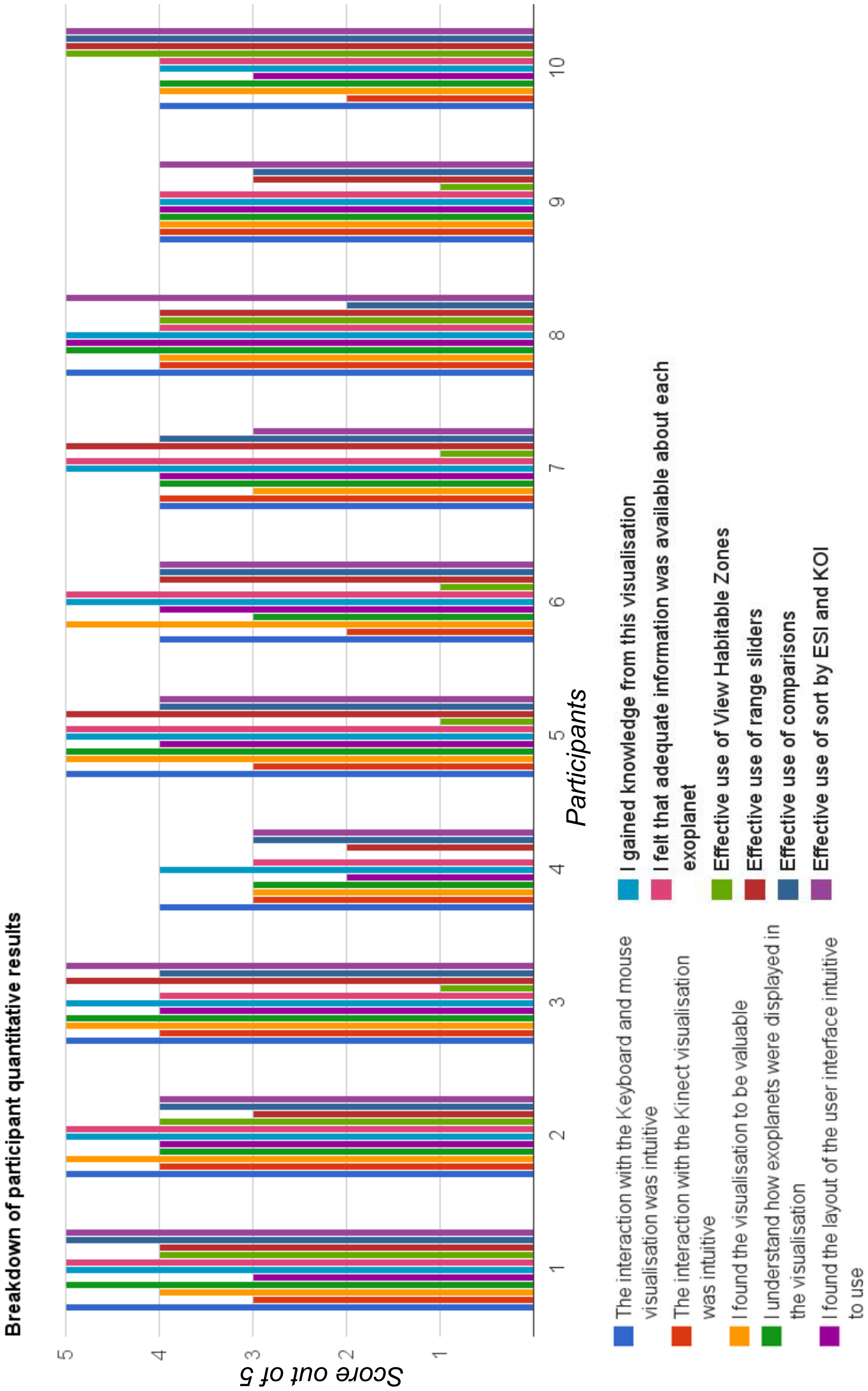

Figure 8.8: Summary of quantitative results between users. A notable finding is that some users consistently gave higher or lower scores which could have caused a bias in the results. It also showed that some users enjoyed the visualisation more than others.

8.4 Discussion

8.4.1 Threats to Validity

A key factor in this user evaluation was the small number of users chosen due to the time limit of 300 hours which did not allow for a larger and more in depth study. Because of this, the results gathered may not be representative of a larger population. It was also limited to a narrow demographic of young professionals and students which does not cover all of the intended users of IKVT. To amend this, the population of this study would need to be expanded to contain age ranges above and below those in this study.

A further threat to the validity of this evaluation was the limit placed on participants for the familiarisation stage. Some participants may have required even more time than the 5 minutes alloted for. By cutting the familiarisation stage off at this point it may have caused some participants to be unprepared to complete the worksheet and thus skew the results.

8.4.2 Potential Improvements

Due to the negative results surrounding the viewing of Goldilocks zones, this area of the evaluation should be analysed further to determine whether the tasks in the evaluation relating to it were flawed, or whether the functionality and its design were flawed. To discover this it would be important to evaluate the questions used in the study to determine whether or not they skewed the result or if the the functionality needs to be redesigned.

8.5 Summary of Evaluation

This evaluation of IKVT provided several key results and insights into the success of IKVT. The general consensus by the participants of the study was that the system had a low learning curve, was enjoyable to use, and allowed access to interesting information. Participants also agreed that the Kinect system was more fun to use than the keyboard & mouse because of its novelty, but lacked the control that the keyboard & mouse allowed which made it less effective at accessing the information in the visualisation. The keyboard & mouse was found to the be the most effective method for navigating the visualisation because of the amount of control that it offered users as well as being the interactive medium that participants had the most experience with.

The evaluation revealed one problem area in the visualisation that was related to Requirement 5 (Users need to be able to view the habitable zones of stars in relation to the planets orbiting them), and Requirement 6 (All interaction methods must be visible and intuitive). This problem occurred because users found that viewing where each exoplanet was located in relation to their stars habitable zones was unintuitive and difficult to use. This result could have occurred for a range of reasons, the most likely being that the evaluation questions were counterproductive to using this functionality, or that the functionality itself is flawed.

This user study served to evaluate how effectively IKVT could convey the information in the Kepler Exoplanet dataset. The findings pointed to it being successful in this regard. However, there are areas that this evaluation could be improved to strengthen this result. The main improvement for this evaluation is to include a larger number of participants more representative of the wider population. These results were first gathered in the qualitative study, and were then confirmed by the quantitative results that found the same.

Chapter 9

Summary

This project contributes the design, implementation, and evaluation of an interactive 3D visualisation called the Improved Kepler Visualisation Tool (IKVT) to the field of Human Computer Interaction. IKVT has two modes of interaction; the traditional keyboard & mouse and the more novel Microsoft Kinect sensor.

9.1 System Design

The requirements for IKVT were designed using Cooper et al.'s user oriented design approach which emphasised using user models and user scenarios [17]. Two user models and seven user scenarios were used in the creation of eight project requirements that were used extensively in the design, implementation and evaluation of IKVT to ensure that it solved the four key issues that this project hoped to address.

Issue 1. Content in the Kepler Exoplanet Database is difficult to view and understand due to its amount and labeling.

Issue 2. Planetary information is complex and difficult to comprehend without a visual reference due to its scale.

Issue 3. Existing visualisations for exploring planetary data have minimal functionality for exploring the information in effective ways.

Issue 4. Visualisations need to allow user interaction to make the most of the data they display.

9.2 System Implementation

IKVT was implemented in Processing, an open-source Java programming language and integrated development environment (IDE). It was built upon an existing visualisation, the Kepler Visualisation Tool [12, 33] to create more effective interaction techniques and allow users to access more information in the Kepler Exoplanet dataset. During the implementation stage of the project each of the designs created in the system design stage were implemented in fulfillment of the project requirements.

9.3 System Evaluation

IKVT was evaluated in a user study that gathered both qualitative and quantitative results. The evaluation was designed to gather users experiences with the IKVT to examine whether or not it fulfilled the system requirements designed to solve the issues mentioned above.

The evaluation found that all of these requirements were fulfilled by the visualisation. In addition to this the participants in general found that the IKVT had a low learning curve, was enjoyable to use, and allowed access to interesting information. There was also a common consensus among the participants that the Kinect system was more fun to use than the keyboard & mouse because of its novelty, but lacked the control that the keyboard & mouse allowed which made it less effective at accessing the information in the visualisation. The evaluation also revealed that one area, viewing the habitable zones of stars, lacked usability because it was unintuitive for the users participating in the evaluation.

9.4 Future Work

IKVT can be taken further in many ways depending on how it is intended to be used. There is the option of using the system as a terminal that users would use at an observatory or attraction where prior knowledge of the system is limited and amount of time users would spend on the system would be small. In this case further expanding the user experience and improved Kinect interaction would be beneficial as immersion would be the decider on its success. Another option would be for a standalone desktop system that users would use multiple times and so prior knowledge of how to use the system could be expected. This would mean that more complex functionality could be introduced with the expectation that it could be learned and used by users. The systems current state could me modified to fit into either of these two options.

A weakness in the system that needs to be addressed is the Goldilocks zone view which the evaluation discovered was not intuitive enough to make it effective. To address this the functionality needs to be evaluated to discover which aspects of it cause it to be unintuitive. The first improvement is to provide a common point of reference for the habitable zones. This is because the area that confused users the most was that when they selected a new planet Goldilocks zones changed.

The user evaluation found that gesture based control of IKVT was the most enjoyable way for users to interact with IKVT even though it was less effective at accessing information. This opens up the opportunity for further work to improve the level of gesture based control and accuracy so that it rivals the keyboard & mouse. This could be done in a range of ways from simply enhancing the intuitivity, to more advanced methods of gesture detection such as joining gestures together.

These three future improvements are ordered by importance below.

1. Address Goldilocks zone view
2. Improve gesture based control
3. Focus IKVT towards a set environment to focus futher development

Bibliography

[1] Celestia. http://www.shatters.net/celestia/. Accessed: 2013-05-10.

[2] Controlp5. www.sojamo.de/libraries/controlP5. Accessed on: 2013-04-20.

[3] Exo: A visualization of kepler's exoplanet candidates. http://vimeo.com/41655330. Accessed on: 2013-08-20.

[4] Github. https://github.com/. Accessed on: 2013-04-01.

[5] Kde - experience freedom. http://www.kde.org/. Accessed on: 2013-08-20.

[6] Kinect. http://www.xbox.com/en-GB/kinect. Accessed on: 2013-05-20.

[7] Nite. www.primesense.com. Accessed on: 2013-05-20.

[8] Processing. www.processing.org. Accessed: 2013-05-10.

[9] Simpleopenni. code.google.com/p/simple-openni. Accessed on: 2013-05-20.

[10] BECK, K., BEEDLE, M., VAN BENNEKUM, A., COCKBURN, A., CUNNINGHAM, W., FOWLER, M., GRENNING, J., HIGHSMITH, J., HUNT, A., JEFFRIES, R., KERN, J., MARICK, B., MARTIN, R. C., MELLOR, S., SCHWABER, K., SUTHERLAND, J., AND THOMAS, D. Manifesto for agile software development, 2001.

[11] BELL, D. Uml basics: The class diagram.

[12] BLPRNT. Github:kepler visualisation tool. https://github.com/blprnt/Kepler-Visualization. Accessed: 2013-05-10.

[13] BOEHM, B. W. A spiral model of software development and enhancement. *Computer 21*, 5 (1988), 61–72.

[14] BORUCKI, W. J., KOCH, D., BASRI, G., BATALHA, N., BROWN, T., CALDWELL, D., CALDWELL, J., CHRISTENSEN-DALSGAARD, J., COCHRAN, W. D., DEVORE, E., ET AL. Kepler planet-detection mission: introduction and first results. *Science 327*, 5968 (2010), 977–980.

[15] BOSTOCK, M., OGIEVETSKY, V., AND HEER, J. D3: Data-driven documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)* (2011).

[16] CHAN, W. W.-Y. A survey on multivariate data visualization. *Department of Computer Science and Engineering. Hong Kong University of Science and Technology 8*, 6 (2006), 1–29.

[17] COOPER, A., REIMANN, R., AND CRONIN, D. *About Face 3 The Essentials of Interaction Design*. Whiley Publishing Inc. Indianapolis Indiana, 2007.

[18] FABRYCKY, D. The kepler orrery, 2012. kepler.nasa.gov. Accessed: 2013-05-11.

[19] FEW, S. *Data Visualization for Human Perception*. The Interaction Design Foundation, Aarhus, Denmark, 2013.

[20] FITTS, P. M. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology 47*, 6 (1954), 381–391.

[21] FOOTE, B., AND YODER, J. Big ball of mud. *Pattern languages of program design 4* (1997), 654–692.

[22] HALL, R. Within subjects designs, 1998. web.mst.edu. Accessed: 2013-05-14.

[23] HEER, J., CARD, S. K., AND LANDAY, J. A. Prefuse: a toolkit for interactive information visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2005), ACM, pp. 421–430.

[24] KOSARA, R., HEALEY, C. G., INTERRANTE, V., LAIDLAW, D. H., AND WARE, C. Thoughts on user studies: Why, how, and when. *IEEE Computer Graphics and Applications 23*, 4 (2003), 20–25.

[25] LEWIS, D., HAROZ, S., AND LIU MA, K. Layout of multiple views for volume visualization: A user study.

[26] MENDEL, J., AND PAK, R. The effect of interface consistency and cognitive load on user performance in an information search task. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting 53*, 22 (2009), 1684–1688.

[27] PARKER, A. Worlds: The kepler planet candidates, 2012. kepler.nasa.gov. Accessed: 2013-05-10.

[28] PHL. Phls exoplanets catalog. phl.upr.edu/projects/habitable-exoplanets-catalog/. Accessed: 2013-04-15.

[29] PIGGOTT, J., AND WOODHAM, L. Thinking through, and by, visualising, 2008.

[30] ROYCE, W. W. Managing the development of large software systems. In *proceedings of IEEE WESCON* (1970), vol. 26, Los Angeles.

[31] SCHWABER, K. *Agile project management with Scrum*. O'Reilly Media, Inc., 2004.

[32] STASKO, J. Position statement evaluating information visualizations: Issues and opportunities.

[33] THORP, J. A new view of the galaxy: Exclusive kepler data visualization, 2011. http://boingboing.net/2011/02/08/a-new-view-of-the-ga.html.

[34] WRIGHT, J. T., FAKHOURI, O., MARCY, G. W., HAN, E., FENG, Y., JOHNSON, J. A., HOWARD, A. W., VALENTI, J. A., ANDERSON, J., AND PISKUNOV, N. The Exoplanet Orbit Database. http://arxiv.org/abs/1012.5676. Accessed: 2013-04-20.

Appendix A

Project Gantt Chart

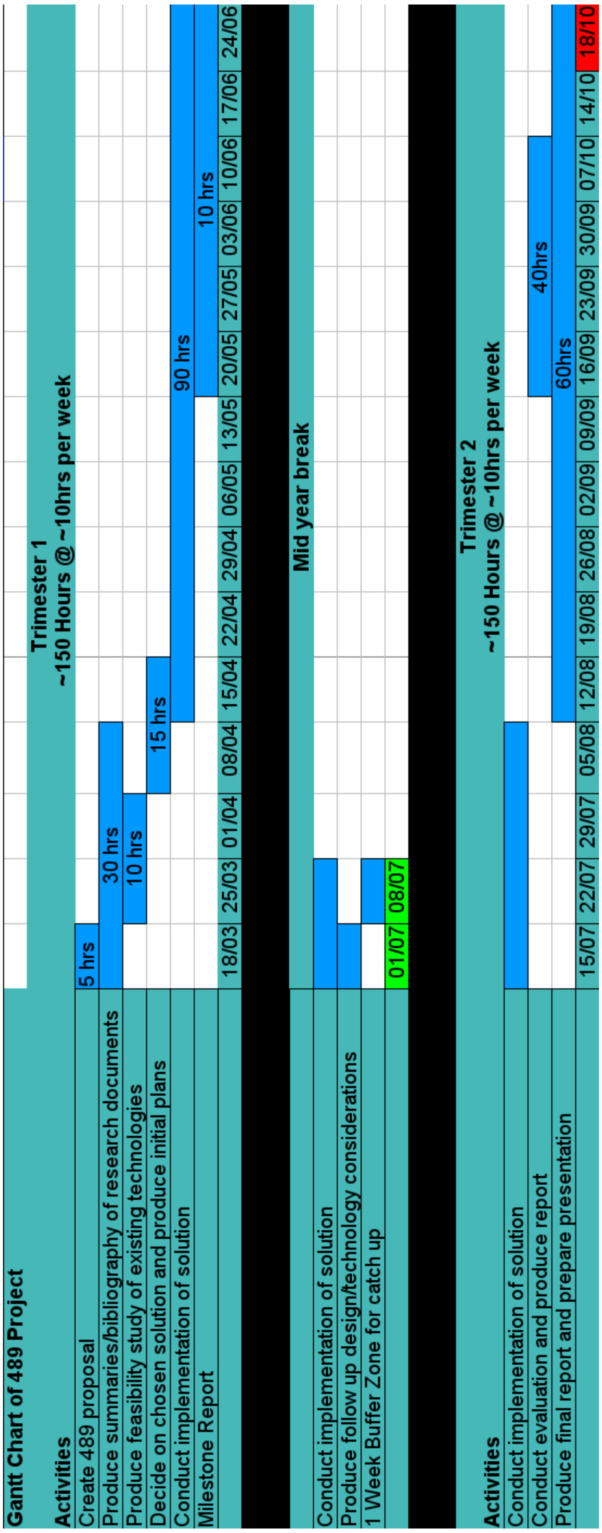

Figure A.1: Gantt Chart used for project

Appendix B

Ethical Approval for Experiment

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI

VICTORIA UNIVERSITY OF WELLINGTON

MEMORANDUM

Phone 0-4-463 5676
Fax 0-4-463 5209
Email Allison.kirkman@vuw.ac.nz

TO	Owen Bannister
COPY TO	Stuart Marshall
FROM	Dr Allison Kirkman, Convener, Human Ethics Committee

DATE	2 October 2013
PAGES	1

SUBJECT	**Ethics Approval: 19472** Graph layout for human interaction/ Interactive 3D Visualisation of Exoplanets

Thank you for your request to amend ethics approval for RM# 19472 (original applicant Roman Klapaukh). This has now been considered and the request granted.

The application has approval until 28 February 2015. If your data collection is not completed by this date you should apply to the Human Ethics Committee for an extension to this approval.

Best wishes with your research.

Allison Kirkman
Human Ethics Committee