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

We have large amounts of information about planets outside of our solar system, and this can be accessed by anyone from the Kepler Exoplanets dataset [18]. However this information is often complex 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 including the planning of the deliverable visualisation, its implementation, and its evaluation to discover how 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

This project seeks to design, implement, and evaluate an interactive 3D visualisation software system for displaying the content in the Kepler Exoplanet dataset using both gesures and traditional interactive methods [18].

The Kepler Exoplanet dataset contains information

This deliverable is intended as a standalone 3D visualisation with two modes of interaction, keyboard and mouse or Microsoft Xbox Kinect sensor [4]. 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

There have been many planets located 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 stored in the Kepler Exoplanet Dataset [18]. 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 data involved in this project caused a range of problems revolving around understandability. This project attempts to address these issues in a way that results in increased accessability 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 from data mining, data analytics, and visualisations in order to discover and highlight important features in the data so that people can more efficiently understand and use it.

Humans often rely on internal visualisation when we solve problems. We create an image in our mind of a situation in order to make sense of it [15] which allows for faster and more comprehensive understanding. 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 1.1 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 1.1: Fields of the dataset to be visualised

Therefore 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 [11]. 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 lots of potential information unseen and overlooked. For example, 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 something 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 exeption of Exo [**?**] 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:

I1. Content in the Kepler Exoplanet Database is difficult to view and understand due to its amount and labeling.

I2. Planetary information is complex and difficult to comprehend without a visual reference due to its scale.

I3. Existing visualisations for exploring planetary data have minimal functionality for exploring the information in effective ways.

I4. Visualisations need to allow user interaction to make the most of the data they display.

These issues are used in Chapter 3: Requirements Analysis to create the project requirements.

1.3   Contributions of this project

This project provides a visualisation that conveys more information and contains better interactivity than other visualisations in the same area. This extension is evaluated qualitatively by a user experiment to ensure that it is successful in conveying the information contained in the dataset.

The interactivity that is provided by this project is a mix of keyboard and mouse as well as a more novel approach using a Microsoft Kinect sensor. This sensors provides a further contribution of gesture based interactivity of a 3D visualisation.

The work and research completed for this project will provide the opportunity for further improvement of the created visualisation by other developers and researchers. This will create further exposure of the Kepler dataset which will encourage learning about exoplanets and the Kepler mission.

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 different components that go into a successful project. It is beneficial as it encourages 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 described as 1) write some code. 2) Fix the problems in the code [9]. The code-and-fix approach may be suitable in some very 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 [12]. A widely used method of managing IT projects is to use a Software Development LifeCycle (SDLC) which details the stages that a project must pass through when being created.

2.1.1 Software Development Life Cycle (SDLC)

The project methodology/SDLC chosen for this project was a customized Spiral Model [9] made up of requirements analysis, design, implementation, and evaluation phases as shown in Figure 2.1.

A software project repeatedly passes through these phases in iterations (called Spirals in this model). For each iteration of the spiral a piece of functionality is completed. The first stage of the spiral involves analysing the requirements of the functionality being created. Second the designs are created, including the element structure and visual elements. Third the functionality is implemented. Finally in the fourth stage the functionality is user tested by one or more people. This evaluation is in addition to the final user evaluation with multiple users.
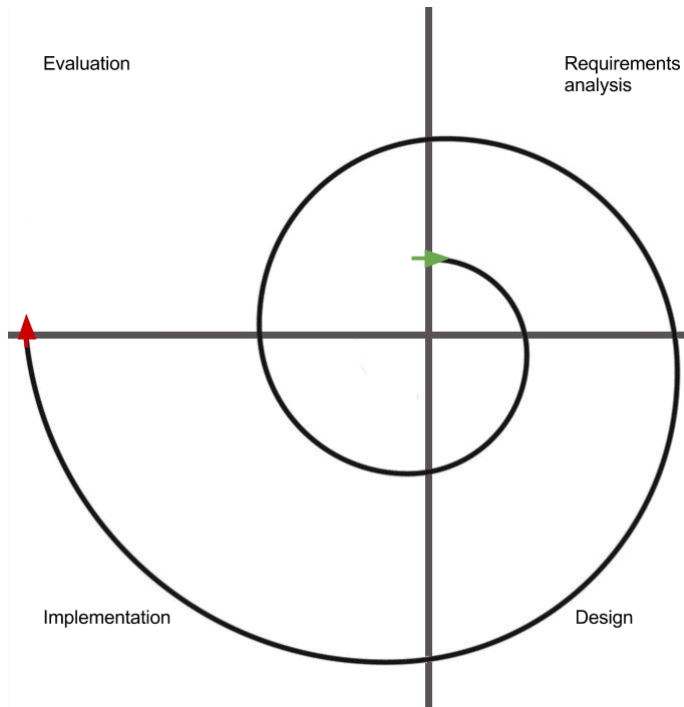
5

Figure 2.1: Spiral process model followed

This SDLC technique supported the creation of a visualisation as it allowed the flexibility to add and remove components into the visualisation as they were discovered to be beneficial or not. It also supported the expansion of the project brief to include using a Microsoft Kinect sensor in order to interact with the visualisation. 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 my supervisor, thus ensured that I did not become delayed or stuck in my development with nothing to show. By using this methodology it also allowed me to prioritize the features that were the most important to the visualisation which reduced the risk that there would be missing or incomplete components at the end of the project.

Following this SDLC allowed me to achieve all of the goals that I set, although it did not always happen within the time frames that I had planned and some deadlines had to be revised. The delays were caused by increasing the scope of the project to include the Microsoft Kinect sensor which required further planning, implementation and testing. Although there were these delays, due to the planning and project management approach that was used, all project elements were completed.

2.2   Other project methodologies explored

Two other project methodologies were explored before the choice to use the Spiral Model was chosen. These alternatives were a loose agile method SCRUM, and a strict Structured Method, the Waterfall Model. The advantages of the Spiral model over these alternatives was that it 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 which limits flexibility and support for changing requirements as was needed for this project. Following a pure agile approach would not have been optimal either as most Agile methodologies(ie SCRUM [16]) are more beneficial to projects with more than a single person working on them. As it was I was agile in my approach to the project as embraced changing requirements, collaborated with my supervisor (the customer) regularly, emphasized working software over large amounts of documentation [7].

Chapter 3

Requirements Analysis

To guide the creation of the visualisation a user oriented design approach was used, in particular making use of user models (personas). These personas were 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.

The design of the visualisation was based heavily on User Centered Design as it provided a method of user interface design as well as visualisation design. User Centered Design is a process in which the needs, wants, and limitations of the end users of a system are given extensive attention. To achieve this, personas were created (also known as archetypal users), which are a personification the needs of a larger group of related users. These personas 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 real users. This design methodology supported my understanding of how users were likely to use the visualisation.

An additional tool used during requirements analysis was User Scenarios which describe the foreseeable interactions of the user personas with 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

Below are the two personas that were used in the design of the visualisation for 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 evaluation of the visualization by finding real users that match the core values of the personas.Although this visualisation would be suited towards teaching childeren about stelar information they were not focussed on during its planning. This was because due to the increased ethical complexity with carrying out an evaluation with childeren. This could be done in the future once the visualisation is deemed successfull.

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 the computer terminals that allow users to choose what information they see.

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 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. From here John can select any of the planets for further analysis.

Scenario 2: Select ranges for attributes of each planet displayed

John has become comfortable with selecting the planets and has some idea of the scale and basic attributes of the planets. 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 accurately select them 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 (temp ,size ,KOI , ESI).

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 each of 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 clicks on a planet orbiting a planet.

3. The planet selected becomes larger and its outline grows, making it more visible.

4. The text window has all of the 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. To discover this he wants that when a planet is selected all other planets in the same Solar System as the selected planet to 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 planet

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 habitble zones of the selected planets star become visible.

3. When a planet from a different star system is clicked then the habitable zones will change to match the new selected planets stars habitable zones.

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)

23 year old Cara likes using interactive visualisations when visiting attractions, she finds that they are more entertaining and provide a better level of interaction and more of a novelty experience with a visualisation than simply a keyboard and mouse.

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.

Scenario 3: Select planets to display more information

Cara wants to see more information about each of the planets she can see orbiting in the visualisation. To do this she wants to be able to hover her hand over a planet to get the information to display on screen.

Procedure:

1. Cara hovers her hand over a planet to make a selection

2. The planet selected becomes larger and its outline grows, making it more visible.

3. The text window has all of the 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 that when a planet is selected all other planets in the same Solar System as the selected planet to 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 7: Navigate the visualisation with gestures

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

1. By moving her hand the visualisation pans in the corresponding direction, ie if the hand goes to the top of the screen the visualisation pans up.

2. By moving her hand backwards and forwards the visualisation will zoom in and out.

3.2   Requirements summary

3.2.1   Functional Requirements

Functional requirements define the functions 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 of stars in relation to the planets orbiting them.

3.2.2 Nonfunctional Requirements

Functional requirements are supported by non functional requirements. Non functional requirements impose constraints on the design or implementation (such as performance, security, or usability) of a system.

The non functional 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 with the system, keyboard and 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: Matrix of project requirements to issues project is attempting to address

Chapter 4

Related Work

This section discusses 4 existing visualisations that have a theme depicting space or planets. Following the description, each visualisation is analysed to discover which of the User Scenarios detailed previously existed within the system.

4.1 Worlds: The Kepler Planet Candidates - Non Interactive

This animation [14] shows 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: Image of Worlds Visualisation

The layout of this animation is very similarly to the Kepler Visualisation Tool that I am extending. This means that it provides insights into how my visualisation can be improved as Worlds is a much more visually appealing system. By researching how it displays its Exoplanets I can further improve my own visualisation.

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 does not offer any functionality for any filtering of Exoplanets, this means that a user can only see all planets at once which can be overwhelming and causes many of the exoplanets to be excluded from the user 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 each Exoplanet

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

Scenario 4. View planets in the same solar system:
Worlds shows all Exoplanets as if they are orbiting a single star. This allows for each of them to be compared against one another easily. However this does remove the ability for a user to see which planets are together in the same solar system.

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

The Kepler Orrery [10] illustrates the exoplanet candidates 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.
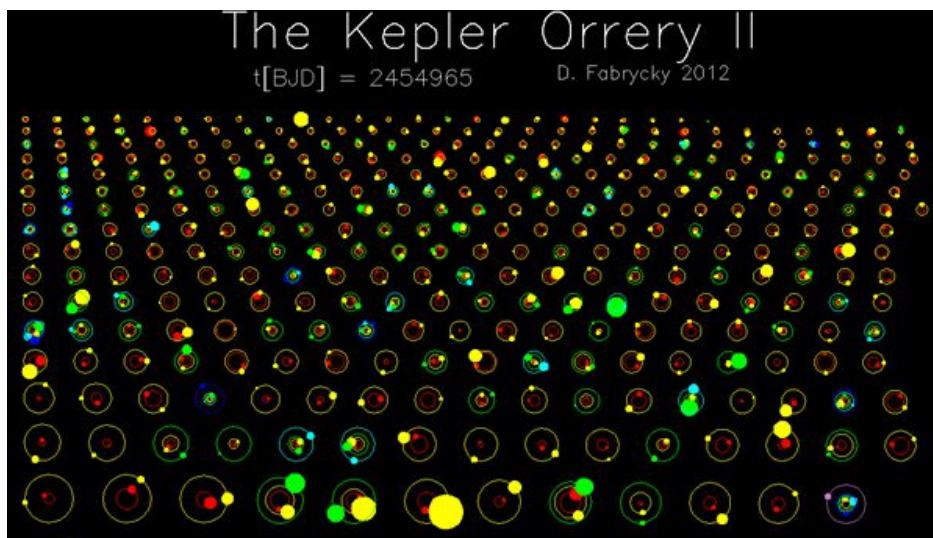
Figure 4.2: Image of The Kepler Orrery Visualisation

This system exhibits small multiples, a grid of small similar graphics or charts, allowing them to be easily compared. This provides insights into how I can use small multiples to display information about groups of planets. This will be important for displaying which planets share a solar system.

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 2. Select ranges for attributes of each planet displayed:
As The Kepler Orrery is non interactive it is not possible to change the ranges of the Exoplanets being displayed. However due to the layout of the visualisation, which uses small multiples by grouping each solar system into its own visualisation element it removes a lot of the issue of overcrowding and overlapping.

Scenario 4. View planets in the same solar system:
This visualisation uses small multiples to great effect at displaying the groupings of each panel into each solar system. By displaying each planet orbiting its own star it removes the risk of confusion about what the planet is actually orbiting which could be the case with Worlds.

4.3   Celestia - Interactive

Celestia [1] is a free real-time space simulation that lets you visually experience the universe in three dimensions. It is an open source system written in C++.
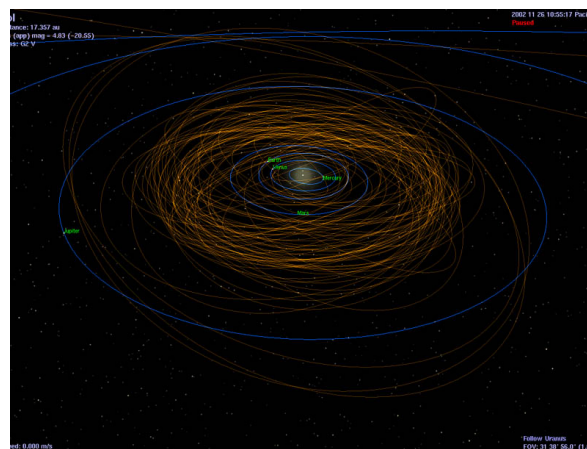
Figure 4.3: Image of Celestia Visualisation

This visualisation is much larger and more encompassing system than is needed for this project, as it is a full 3D space simulation. However is does offer insights into how to effectively portray planets and their orbits (See Figure 2.3). It also provides textures that can be used in my visualisation to depict what planets actually look like to increase user immersion.

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

Scenario 2. Select ranges for attributes of each planet displayed:
Maybe

Scenario 3. Select planets to display more information:
Maybe

Scenario 4. View planets in the same solar system:
Maybe

Scenario 5. View the Goldilocks zones of each planet:
Maybe

Scenario 6. Select two planets to compare against one another:
Maybe

Scenario 7. Navigate the visualisation with gestures:
Maybe

4.4   Kepler Visualisation Tool

An existing system built with Processing is the Kepler Visualisation Tool[8, 17]. It is a simple visualisation focusing on displaying the candidate Exoplanets temperatures and their locations in relation to their distance from their nearest star, so that a sense of scale can be perceived. Each candidates estimated size, orbital speed, and orbital separation is accurately depicted, and each planet is color-coded according to its estimated effective temperature.
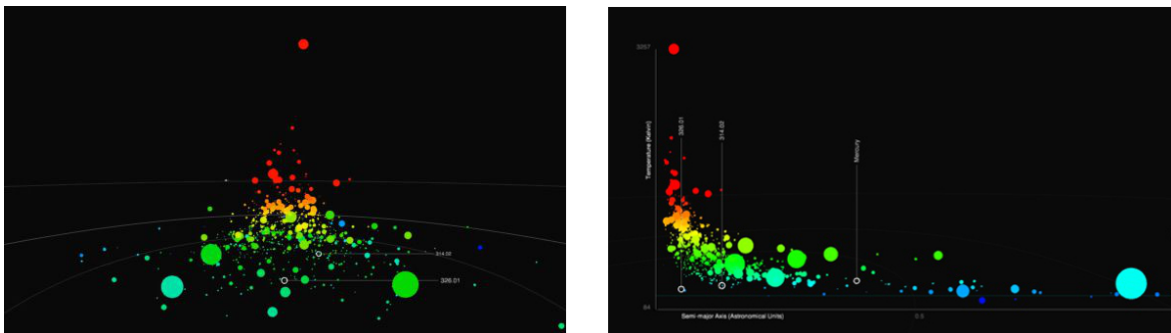
Figure 4.4: Kepler Visualisation Tool Orbital View

The existing work in this system would serve as foundation for this project. Because much of the visual aspects, and initial data manipulation of the existing system are already complete. It means that implementing the features needed for this projects completion could be focused on more heavily and larger improvements to the existing system can be undertaken, such as better labeling and information displays and user interaction methods.

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 with the same colours, size, and orbit speed as the Exoplanets. This gives a user a chance to comprehend the scale and difference of some of the Exoplanets.

Scenario 2. Select ranges for attributes of each planet displayed:
The Kepler Visualisation Tool allows users to change which attributes the planets are ordered by on the vertical plane (Y axis) ie, by size or temperature, but does not allow users to select ranges of these attributes to filter the Exoplanets

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 scenarios

Chapter 5

Technology

Many technologies were looked into and experimented with. Then the positives and negatives of each option was weighed up before a decision was made about which would be the suitable choice for the visualisation. It came down to 3 potential technologies that would be suitable for the project, the next 3 subsections outline these in detail.

5.0.1 Processing

Processing is an open source programming language and development environment that was initially created to serve as a software sketchbook and 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 a successful visualisation framework. The most complete existing visualization using the same exoplanet dataset (Kepler Visualization Tool) is built using Processing. Using this solution would involve learning the Processing language, however Processing is a library built in Java so the syntax is the same. This means the learning curve should be shallow. Using processing means that 3D elements could be included, this wouldn't be possible with D3.

5.0.2 D3 (Data Driven Documents)

D3 is a JavaScript library that allows the displaying of data in dynamic graphics. Embedded within an HTML web page, the JavaScript D3.js library uses pre-built JavaScript functions to select elements, create Scalable Vector Graphic (SVG)[17] 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 visualization 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 for creating visualizations. 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 thanks to Google chrome and other modern browsers 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.

5.0.3 Prefuse

Prefuse is a set of software tools for creating rich interactive data visualizations [13]. The Prefuse toolkit provides a visualization framework for Java. It supports a set of features for visualizing and interacting with data. It provides optimized data structures for tables, graphs, and trees. It can be used to build standalone applications, visual components embedded in larger applications, and web applets. Prefuse to 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 visualizations into larger applications. Experience with database systems is also helpful. However the complexity of Prefuse means that the learning curve will be out of scope for this project.

5.0.4 Decision of technology

The final decision of technology was to use Processing, this is because it had many positive aspects that the others did not and minimal negatives as the below table illustrates.

	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: Table of technology choices

As this project was created using Processing, it allowed me to extend the previous visualisation using the same data set, The Kepler Visualisation Tool [8, 17]. As the time was short for this project building upon a previous solution increased the amount of progress that could be made in the time afforded.

Taking this approach meant that the languages being used would be Java using processing libraries.

As this is such a large project involving many different iterations, version control was important for maintaining records and backups of important changes which was stored on remote servers to ensure against file loss in system failures.

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 aims to improve an existing visualisation, The Kepler Visualisation Tool which was discussed in the previous chapter. Whist this existing visualisation displays exoplanets and some of their features, it lacks interactivity for users trying to use it to gain information contained in the Kepler Exoplanet Database effectively. 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 about exoplanets. The IKVT also incorporates a novel gesture based interactive mechanism to the visualisation using a Microsoft Kinect Sensor.

6.1 System design and structure

Because this project builds upon an existing system, complete comprehension of how it is designed and how it functions is important. Going ahead in the creation of the visualisation without this knowledge would create opportunities for mistakes and incorrect assumptions about how the visualisation needs to be created. To solve this issue diagrams 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 ie class diagrams, and or object diagrams. Behavioral diagrams, on the other hand, show the dynamic behavior between the objects in the system, including things like their methods, collaborations, and activities ie, use case, and sequence diagrams http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/. For this project sequence diagrams and and class diagrams were used to help understand the exisiting system and plan out the extensions, these are discussed next.

1. **UML Sequence Diagram** A sequence diagram is a diagram that 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 would have made it 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.
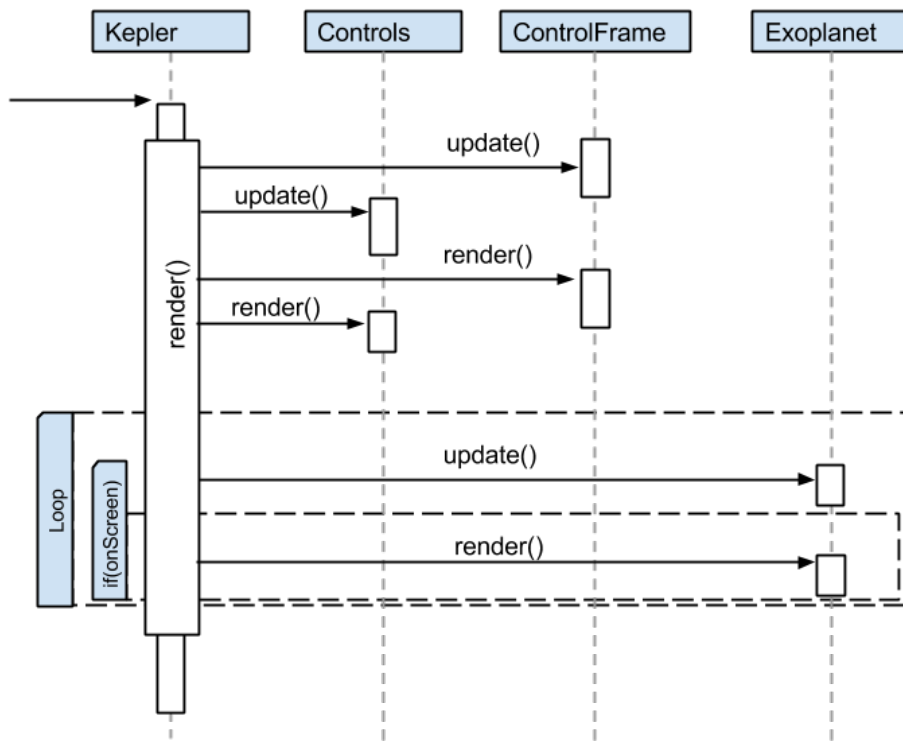
Figure 6.1: Sequence Diagram of IKVT render cycle

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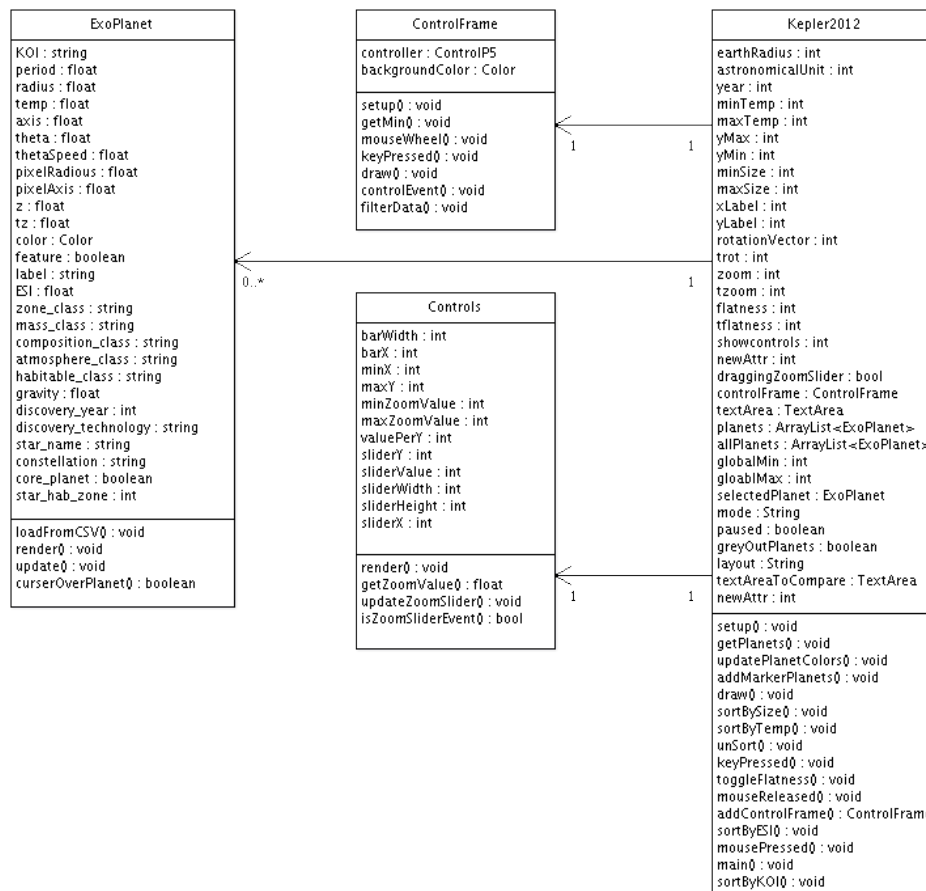
Figure 6.2: Class Diagram of IKVT

By using the sequence diagram in Figure 6.1 coupled with the class diagram in Figure6.2 you can see where each method and field is located and modified. This is a powerfull way of begining to understand an exisitng system and then planning extensions.

6.2 Visualisation Design
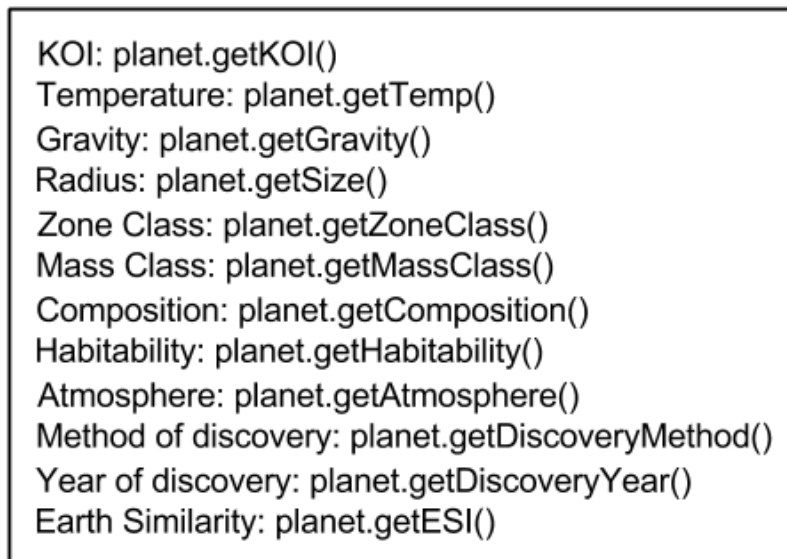
The requirements produced in Chapter 3: Requriements Analysis provide a description of the functionality that needs to be designed for this visualisation. By adding additional details to these requirements we can specify how the visualisation should look, behave, and function.

likely

6.2.1 Functional Requirements

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

This requirement needs some form of textual display in order to convey enough of the information about exoplanets to the user. This can be done with a Java TextArea object to display each of the key attributes of each Exoplanet. The following figure is a mockup 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: Mockup of the TextArea

To populate this text area it needs to be possible to select exoplanets, which is discussed in the next requirement.

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

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

The selection of exoplanets involves detecting when 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 and information to the user to inform them that it has been selected and also to provide relevant information.
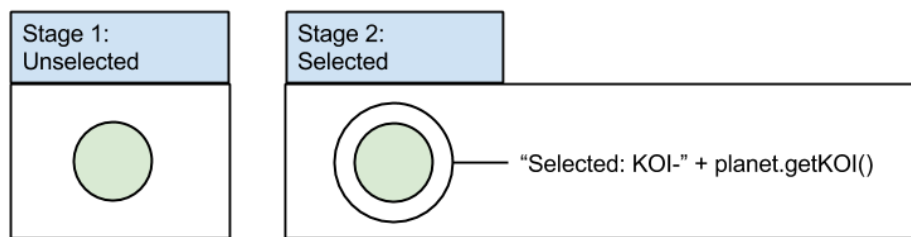
Figure 6.4: Mockup selection process

In addition to this, when a planet has been successfully selected all of the 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.
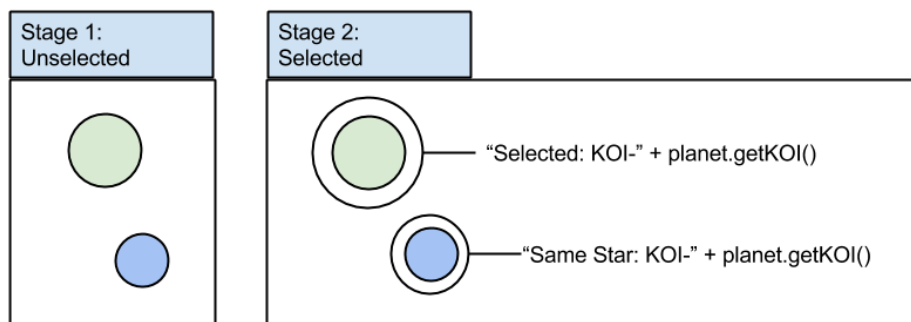
Figure 6.5: Mockup highlight sister planets process

There needs to be an efficient way that users can make comparisons between the detailed textual information about exoplanets. By providing an additional text box and allowing users to select multiple planets this can be accomplished.

```
┌─────────────────────────────────────┐   ┌─────────────────────────────────────────┐
│ Selected Planet                     │   │ Compared Planet                         │
├─────────────────────────────────────┤   ├─────────────────────────────────────────┤
│ KOI: planet.getKOI()                │   │ KOI: compared.getKOI()                  │
│ Temperature: planet.getTemp()       │   │ Temperature: compared.getTemp()         │
│ Gravity: planet.getGravity()        │   │ Gravity: compared.getGravity()          │
│ Radius: planet.getSize()            │   │ Radius: compared.getSize()              │
│ Zone Class: planet.getZoneClass()   │   │ Zone Class: compared.getZoneClass()     │
│ Mass Class: planet.getMassClass()   │   │ Mass Class: compared.getMassClass()     │
│ Composition: planet.getComposition()│   │ Composition: compared.getComposition()  │
│ Habitability: planet.getHabitability() │ │ Habitability: compared.getHabitability()│
│ Atmosphere: planet.getAtmosphere()  │   │ Atmosphere: compared.getAtmosphere()    │
│ Method of discovery: planet.getDiscoveryMethod() │ │ Method of discovery: compared.getDiscoveryMethod( │
│ Year of discovery: planet.getDiscoveryYear() │ │ Year of discovery: compared.getDiscoveryYear() │
│ Earth Similarity: planet.getESI()   │   │ Earth Similarity: compared.getESI()     │
└─────────────────────────────────────┘   └─────────────────────────────────────────┘
```

Figure 6.6: Mockup compare planets text areas

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).

To fulfill this requirement 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. The following figures display an abstract mockup of how this would be done.
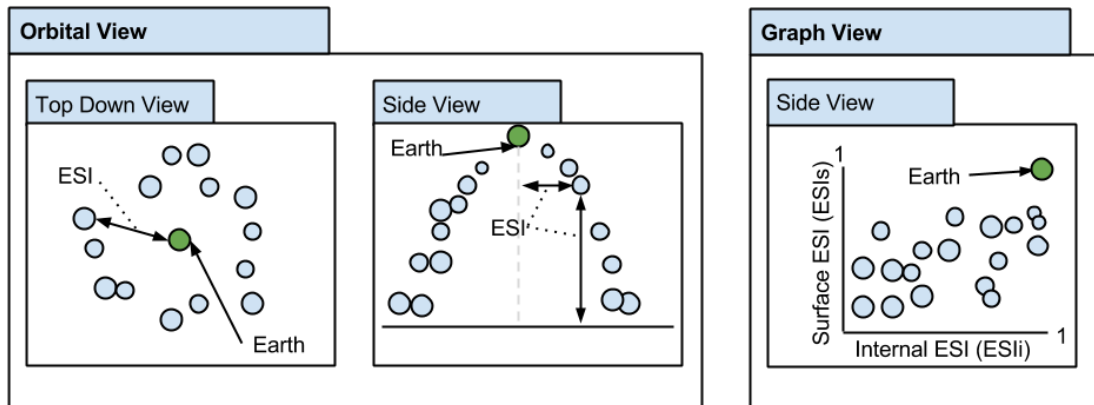
Figure 6.7: Mockup of ESI views

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

To fulfill this requirement a method of filtering exoplanets by their attributes in order to control the number and types of planets displayed to the user. 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. For example in this system a slider could be used to control the size of planets that are displayed.
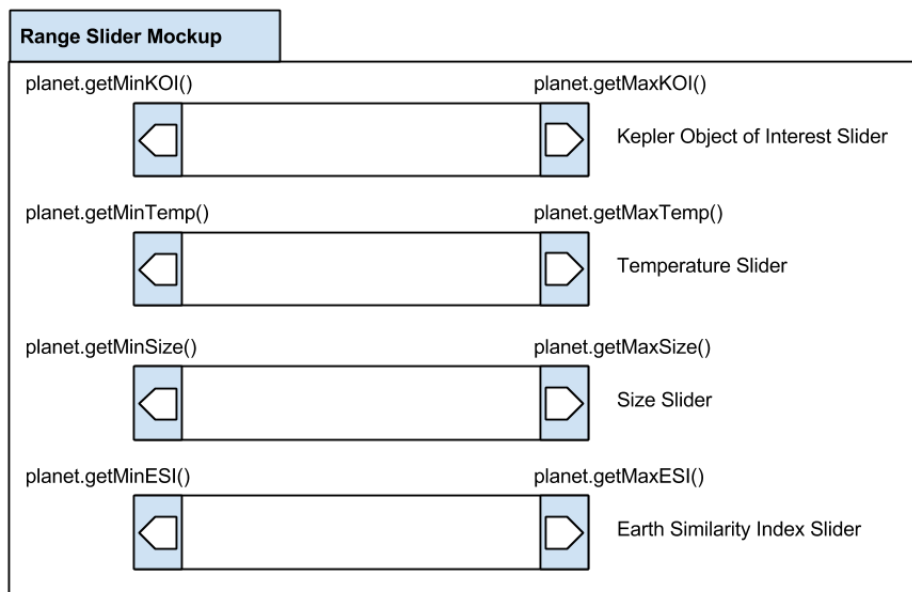
Figure 6.8: Mockup of range sliders

In addition to this users should be able to easily 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 easily see how changing the filters affects the exoplanets displayed
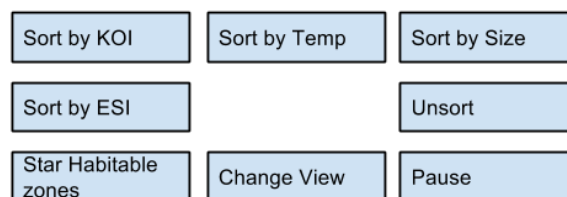
Figure 6.9: Mockup of interactive buttons

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

To display the habitable zones of the stars of each exoplanet 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) is required. 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.
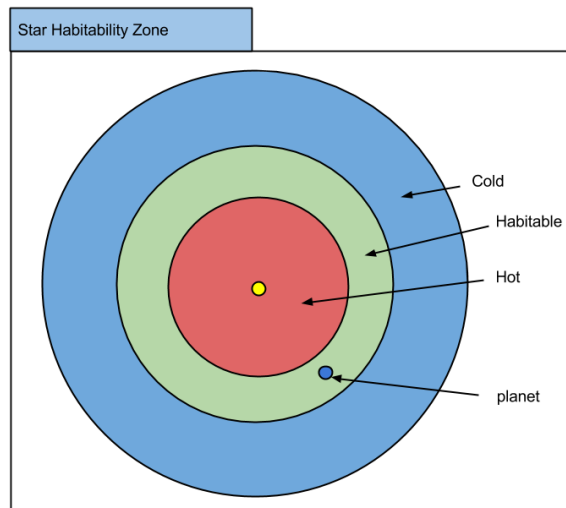
Figure 6.10: Mockup star habitability zone

6.2.2 Nonfunctional Requirement

R6. All interaction methods must be visible and intuitive.

To make an interactive visualisation useable the controlls and interactive methods need to be clear to users. To achieve this for IKVT an interactive panel should be introduced. This panel should contain all of the interactive elements of the visualisation in a single central place that is spacially seperated from the main visualisation. This spacial seperation is imporatant as it reduces cognitive load on users by allowing them to focus on the visualisation iteslef until they need to use the panel in which case they can focus on that. It also means that users only need to look in one place for all of the interactive elements which reduces the risk of confusion. In this visualisation this interactive panel would contain all of the interactive elements as discussed in the previous requirements as displayed below in Figure 6.11.
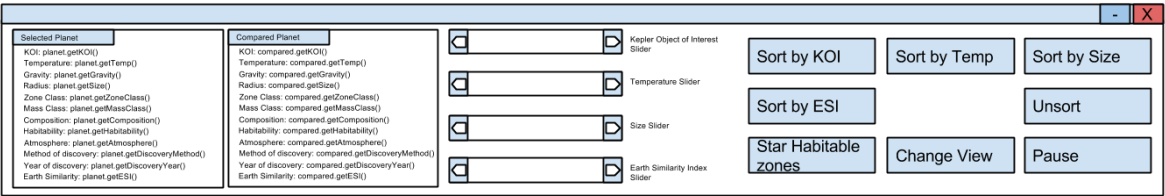
Figure 6.11: Mockup of the interaction panel

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. In addition to this, by having the text area that displays the information about selected planets separate from the main visualisation it reduces the cognitive load on users as they don't have to use this component until they want to. ...

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

The above requirements mostly relate to the keyboard an mouse system although some of the requirements can carry over to a gesture based system. By incorporating a novel interactive method using a Microsoft Kinect sensor it gives users an alternative method of controlling the visualisation. For this to be successfull it means 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 6.12.
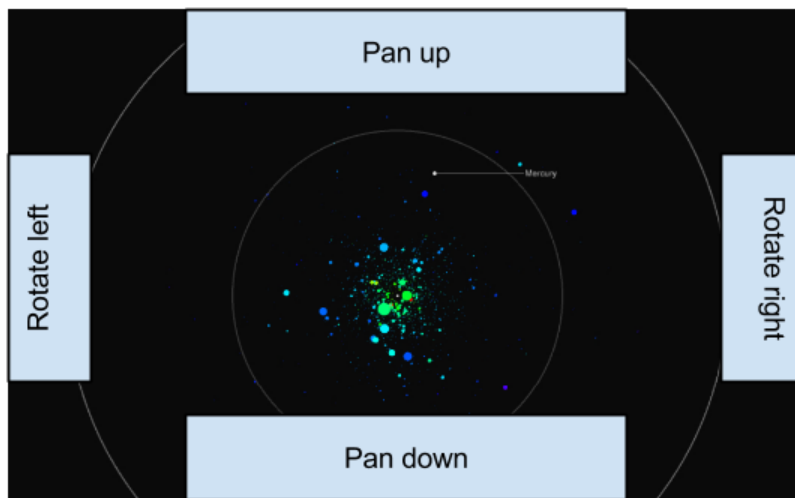
Figure 6.12: Mockup of the Kinect system

In addition to controling the movement of the camera users need to be able to zoom in and out of the visualisation as well as select exoplanets for further examination as needed for the requirements. To do this a user should be able to push and pull his/her hand from the screen to zoom in and out, as well as gesturing to a planet to make a selection.

As the Kinect sensor no longer requires the use of a mouse the visualisation design needs to be modified to accommodate the use of gestures. This meant incorporating new cursors to indicate the state of the visualisation. There are 7 states that the cursor needs to be able to be in to inform the user of what action they are performing. These states are

 (a) default cursor, hand is at rest

 (b) panning up, hand is raised

 (c) panning down, hand is lowered

 (d) rotating left, hand is to the left

 (e) rotating right, hand is to the right

 (f) zooming in, hand is pressed forward

 (g) 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 icons for this are displayed below in Figure 6.13
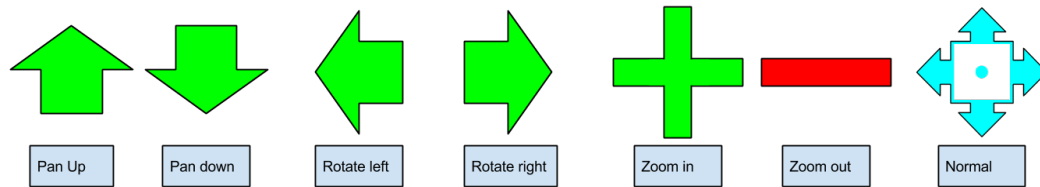
Figure 6.13: Mockup of cursor images for Kinect system

No gesture design has been included for interacting wiht the interaction panel. This is because the Microsoft Kinect sensor implementation in this visualisation is intended as a proof of concept that interaction via gestures can allow users to access the information within the system. If the user study discussed in Chapter 8: Visualisation Evaluation shows that gesture interaction is successfull in this regard then incorporating it further can be undertaken.

These solution designs were used for the implementation of IKVT which is discussed in the next chapter, Chapter 7.

Chapter 7

Visualisation implementation of the Improved Kepler Visualisation Tool (IKVT)

This chapter discusses the implementation of the visulisation using the designs discussed in the previous chapter to fulfill the requirements for this project. It details the tools used, the deliverable features produced, and the problems encountered.

7.1 Tools and artifacts used

7.1.1 Dataset used

The dataset for this project is a comma separated values file (CSV) that contains all of the data pertaining to the Exoplanets. At runtime this dataset is read into the system and an Exoplanet object is created for each element and contains all of the information from the dataset record.

7.1.2 Integrated Development Environment (IDE)

The IDE used for this project is one that is provided with Processing. This IDE provided all of the tools and functionality that were required to effectively implement the solution for this project. Another option that could have been taken was to use Eclipse with the processing package and external libraries imported. However I found that the processing IDE was entirely suitable for the majority of my needs creating the visualisation. The key things that I did not have access to with this choice was the Eclipse Debugger and JUnit tests.

7.1.3 Keyboard and Mouse System

In addition to Processing in the main system there was an additional opensource library required for effective user interface components, this library was called ControlP5 [2]. This library is a customisable and intuitive interactive user interface. It allows for easy creation of visually appealing and precisely layed out interactive GUI components.

7.1.4 Microsoft Kinect sensor system

For the version of the IKVT system that uses the Kinect sensor for user interaction two additional libraries were required to integrate the hardware with Processing, these were:

1. NITE [5]

2. SimpleOpenNi [6]

These libraries provided drivers to run the Kinect sensor in Processing as well as basic gesture recognition and body tracking. However as the libraries were opensource due to the official Microsoft Kinect SDK not being compatible with Processing, 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 opensource libraries.

7.2 Implementation of IKVT

IKVT displays all 2234 exoplanets in the Kepler exoplanets dataset [18]. 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 which in reality would result in planetary collisions but in the 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 all of the exoplanets are displayed as well as text boxes 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 cane be used interchangeably to display information about selected planets, four range sliders (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.
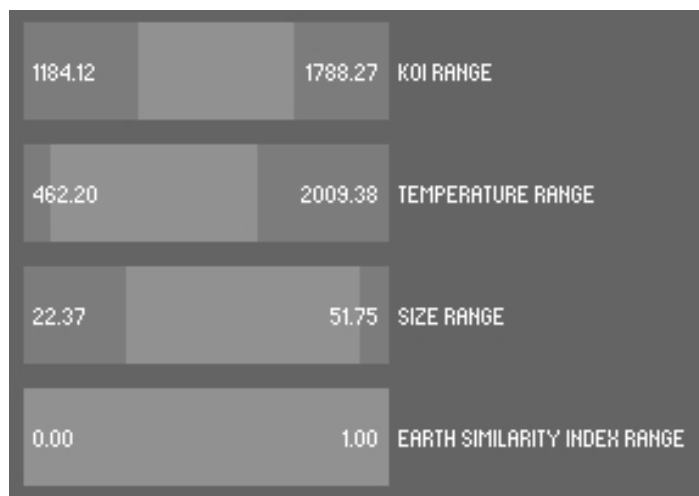
Figure 7.1: Panel of interactive range sliders
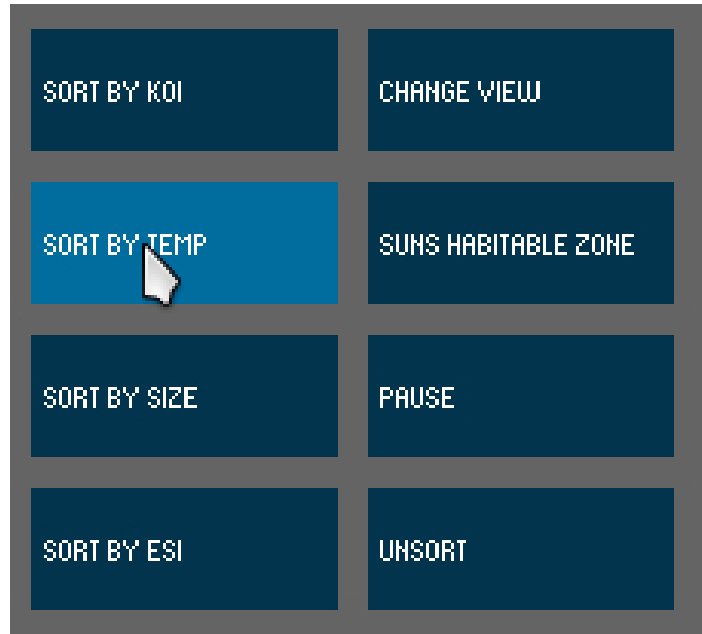
Figure 7.2: Panel of interactive buttons

Detailed information can be accessed about each exoplanet by clicking on them 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 to provide the user with more detailed information 7.3. When a user is unable to accurately select an exoplanet due to clustering or overlapping of exoplanets they can move the camera around in space to gain a better viewing position with which to make their selection. If this is not enough, the user can use a set of range filters to filter the exoplanets displayed as in Figure: 7.4. The attributes that can be filtered by are are Kepler Object of Interest number (KOI), temperature, size, and Earth Similarity Index (ESI). These filters allow for users to fine tune the exoplanets they wish, thus allowing them to work with small multiples rather than the entire dataset. A further effect of using these filters is that a zooming effect occurs as less exoplanets are displayed. This zooming occurs each time the filters are changed as the exoplanets spread out vertically so that it allows more space between them to make selections.

41

No Planet Selected

1

No Planet Selected To Compare

Kepler Plantary Index (KOI): 1387.01
Temperature: 615.0
Gravity: 1387.01
Radius: 31.1
Zone Class: Hot
Mass Class: Jovian
Composition Class: gas
Habitable Class: non-habitable
Atmosphere Class: hydrogen-rich
Technology Discovered By: tran
Year Discovered: 2011
ESlg: 0.05
ESli: 0.04
ESls: 0.05
COMPARE

2

No Planet Selected To Compare

Kepler Plantary Index (KOI): 1387.01
Temperature: 615.0
Gravity: 1387.01
Radius: 31.1
Zone Class: Hot
Mass Class: Jovian
Composition Class: gas
Habitable Class: non-habitable
Atmosphere Class: hydrogen-rich
Technology Discovered By: tran
Year Discovered: 2011
ESlg: 0.05
ESli: 0.04
ESls: 0.05
COMPARE   Select Planet to Compare

3

No Planet Selected To Compare

Kepler Plantary Index (KOI): 1387.01
Temperature: 615.0
Gravity: 1387.01
Radius: 31.1
Zone Class: Hot
Mass Class: Jovian
Composition Class: gas
Habitable Class: non-habitable
Atmosphere Class: hydrogen-rich
Technology Discovered By: tran
Year Discovered: 2011
ESlg: 0.05
ESli: 0.04
ESls: 0.05
COMPARE

4

Kepler Plantary Index (KOI): 1549.01
Temperature: 579.0
Gravity: 1549.01
Radius: 32.8
Zone Class: Hot
Mass Class: Jovian
Composition Class: gas
Habitable Class: non-habitable
Atmosphere Class: hydrogen-rich
Technology Discovered By: tran
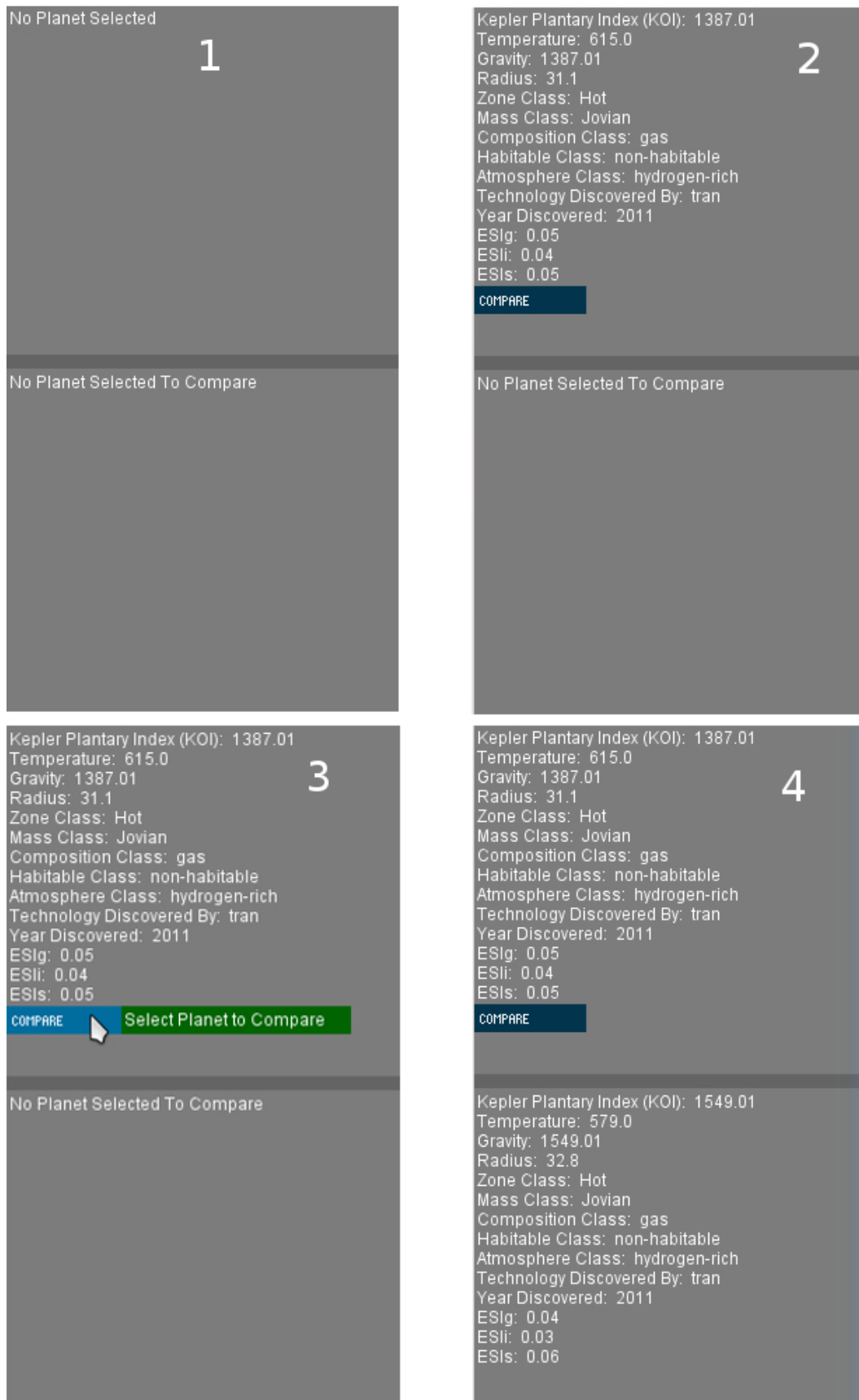Year Discovered: 2011
ESlg: 0.04
ESli: 0.03
ESls: 0.06

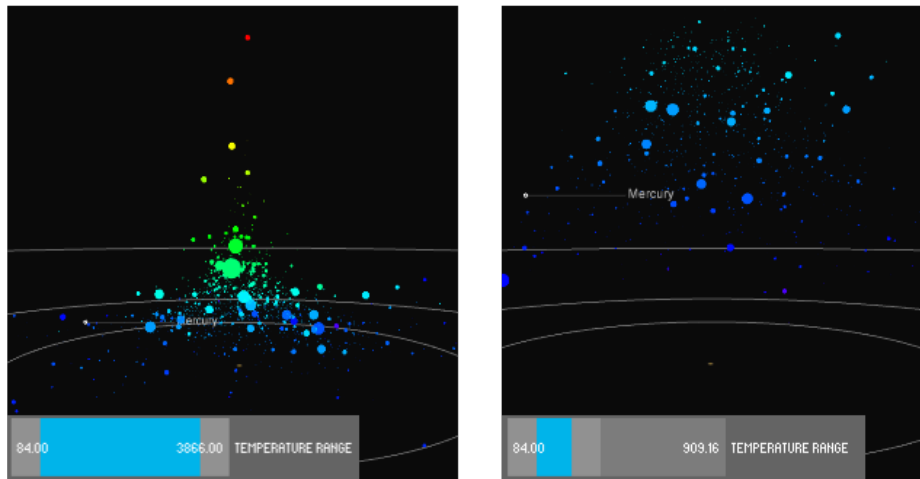Figure 7.3: Text boxes in each possible state
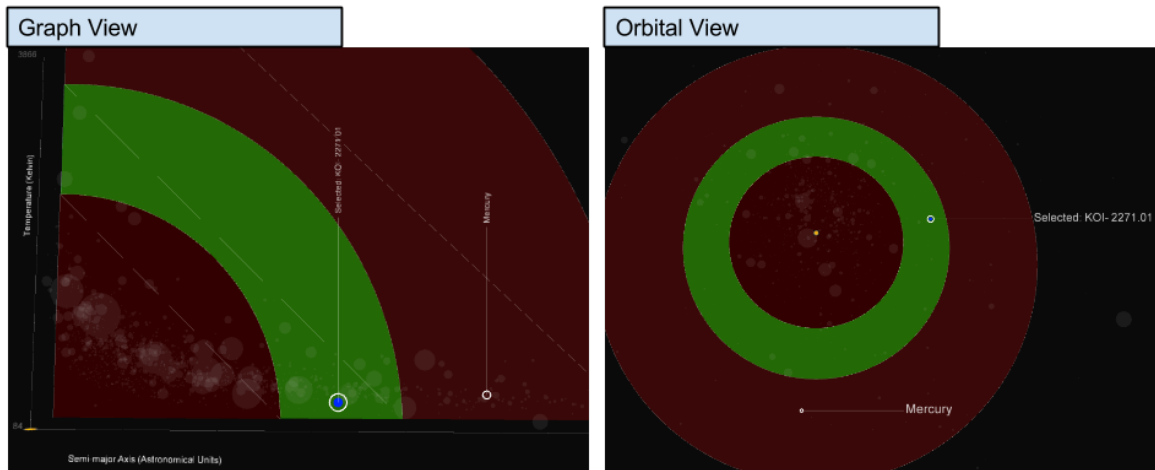
Figure 7.4: TODO

Figure 7.5: TODO

In addition to the orbital view already discussed, there is a graph view that displays the exoplanets on a graph with the exoplanet attributes mapping to the x,y coordinates. This allows the user to modify what they want on each axis of the graph. Having these two views allows the user the option of how they wish to visualise the exoplanets, the less exact and more visually appealing orbital view, or the more exact and less exiting graph view.

The Kepler Orrery visualisation as discussed in the Existing Systems section of Chapter 3 details a contrasting system that displays each exoplanet and its sister planets orbiting its own star. By incorporating this idea into IKVT, when a planet is selected it should be highlighted and all of its sister planets should also become highlighted. This will provide the same effect as in the Kepler Orrery.

Interaction Panel
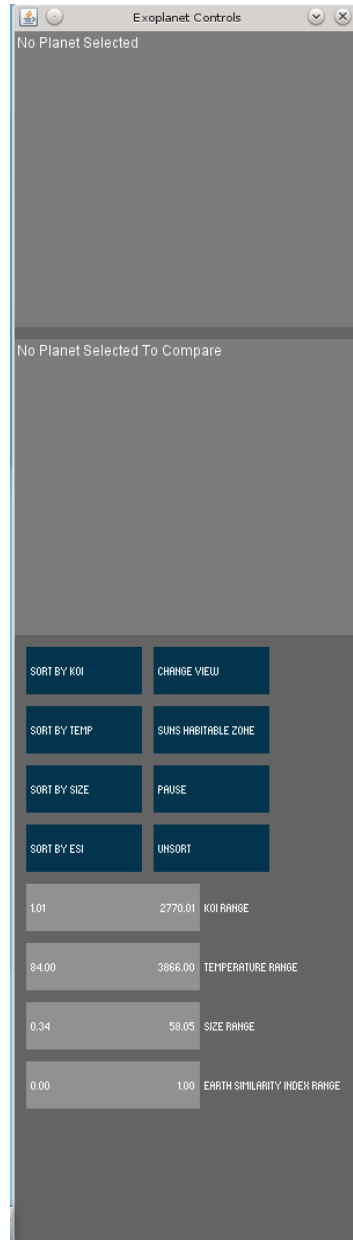
Description of navigation window

Figure 7.6: Navigation Panel Overview

Due to the need for increased user interaction with the visualisation a window is required to house the buttons, range selectors, and text areas. These elements are needed as the different methods that users can use to interact with the visualisation need to be visually apparent to ensure that the system can be easily used without prior experience. A way to do this is to provide clearly labeled interactive elements and tooltips explaining what they do. These tooltips are widely used as a method of informing a user about the purpose of an item by hovering over it. This removes the need to click on a button to discover its effect.

In addition to this, the screen needs to display the user in relation to the screen, an effective way to do this is to display a washed out representation of themselves in the background of the visualisation.

Spacial arrangement of components

As the majority of the interaction and movement of visualisation elements occurs in the center of the window it caused a aspect ratio that was not suitable . It was BETTER to use 2 vertical columns to view and control the visualisation as it had a higher aspect ratio which allowed more of the content to be seen on the screen at once thanks to the fact that the majority of computer screens have a wide ratio.
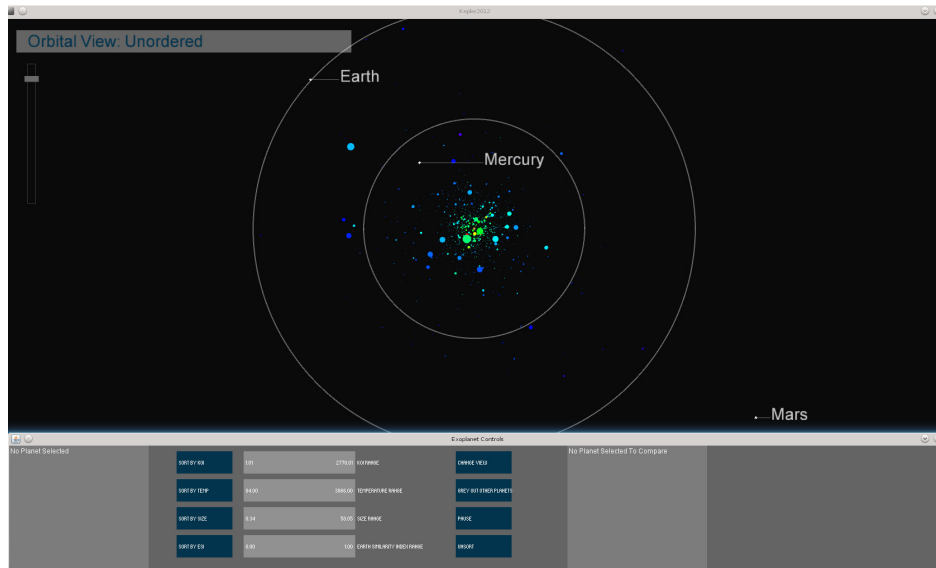

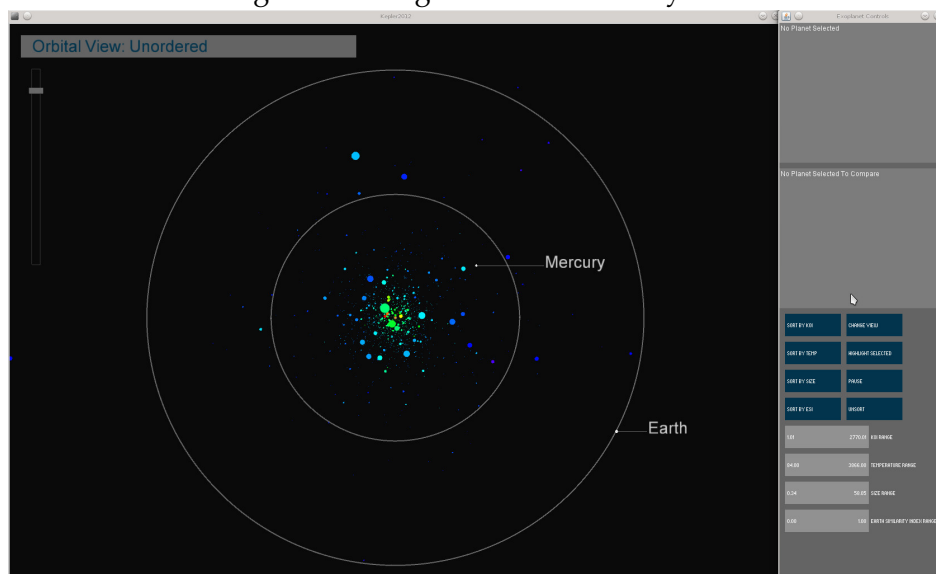Figure 7.7: Original Horizontal Layout

Figure 7.8: Improved Vertical Layout

7.3 Problems encountered

Due to the number of elements that needed to be displayed on screen at any one time (ie 2234 exoplanets), the load placed on a system is very high due to the need to render 2234 eclipses to represent the planets. 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 of 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.

Libraries used for gesture detection in kinect are opensource in order to work with processing did not have decent detection

Using the Processing framework meant using a non industrial??? IDE that had many bugs, 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 I regularly committed any changes to my version controlled system on Github [3]. Doing this meant that if at any time a file became corrupted I could easily see the changes in the file when compared against the precious commit and manually fix the file.

As this project builds upon a previous system much of the existing code and execution flow needs to be modified. This requires understanding of how the system was originally built and designed. Because this system does 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.

Chapter 8

Visualisation Evaluation

Following the completion of the implementation stage of this project a final user evaluation was carried out on the visualisation. This evaluation was primarily designed to discover whether or not the visualisation created was successful in fulfilling the requirements of the project.

8.1 What did we do

8.2 Why did we do it

Visualisation methods are often designed and evaluated by presenting results informally to potential users. No matter how efficient a visualisation technique may be, or well planed it is, if it does not convey information effectively, it is of little use. User studies offer a scientifically sound method to measure a visualisations performance. There are many reasons to pursue user studies. Studies can be used to evaluate the strengths and weaknesses of different visualization techniques or show that a visualization technique is useful in a practical sense, according to some objective criteria, for some specific task [13]. The fundamental goal of conducting user studies is to seek insights into why particular visualisation techniques are effective. [13].

The evaluation of this project was designed to qualitatively assess users reactions and experiences with the IKVT. By mapping users experiences to the project requirements I was able qualitatively evaluate how successful the visualisation was a fulfilling these requirements. Qualitative evaluation was chosen over quantitative because of 3 main reasons ....

8.3 Experimental Design

8.3.1 Expectations of evaluation

From this evaluation the expectations were that users would take approximately 3 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 accostomisation 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 are expected to use both interaction methods (Keyboard & Mouse, and Kinect sensor). During the keyboard & mouse portion of the experiment the users should make more accurate selections and exhibit more effective data seeking behav-

ior, whereas during the Kinect portion they would be more interesting 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 experiences with the system the expectation was that they would detail the areas of the visualisation that ...

8.3.2 Participants

The user study was undertaken by 9 participants P2 to P10 as well as a 1 user pilot study by P1. All were either students or young professionals from a mix of specialties aged between 21 to 26 with a mix of genders with 5 females and 4 males. 5 of these participants had extensive prior experience with Kinect sensors, and all participants had experienced a 3D visualisation before.
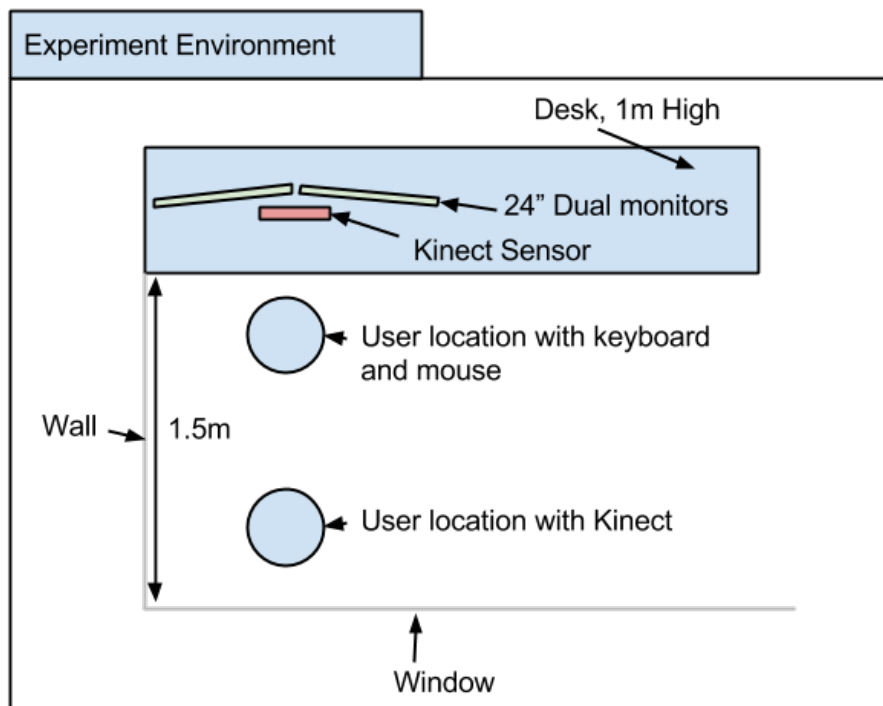
8.3.3 Evaluation Environment

Figure 8.1: Evaluation environment

8.3.4 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. There were 3 key methods of gathering results during this evaluation; a worksheet to fill out whilst using the visualisation made up of 2 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

- The user enters the room and sits down at the computer.

- They are handed the consent form and information sheet.

- 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 and mouse system, and the second if for the Microsoft Kinect system.

- They are then given a brief introduction into each of the visualisation components and what the visualisation as a whole represents

- 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).

- Following this the user 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.

- Once the user has completed both sets of questions they are asked to fill in the qualitative user questionnaire about their experiences using the visualisation.

- Following this if the examiner has no follow up questions the user is free to leave.

8.3.5 Pilot Study

Due to the significant costs associated with running an experiment, it is valuable to conduct a pilot study with one or two participants. This allows testing and refining the experimental design before starting a full-fledged study with numerous participants [13].

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

P1 successfully each of the questions whilst using the visualisation with only limited assistance from the examiner. This assistance was required due to the wording of some of the tasks users were asked to complete being ambiguous which caused 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 very efficient. P1 also found that whist the range sliders were effective, they did not allow fine enough control for making small changes. The component of the visualisation that P1 had the most

difficulty was using the view depicting the location of planets in relation to their stars habitable zones. 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 habitable 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 reactions to questions about the experience of using the IKVT.
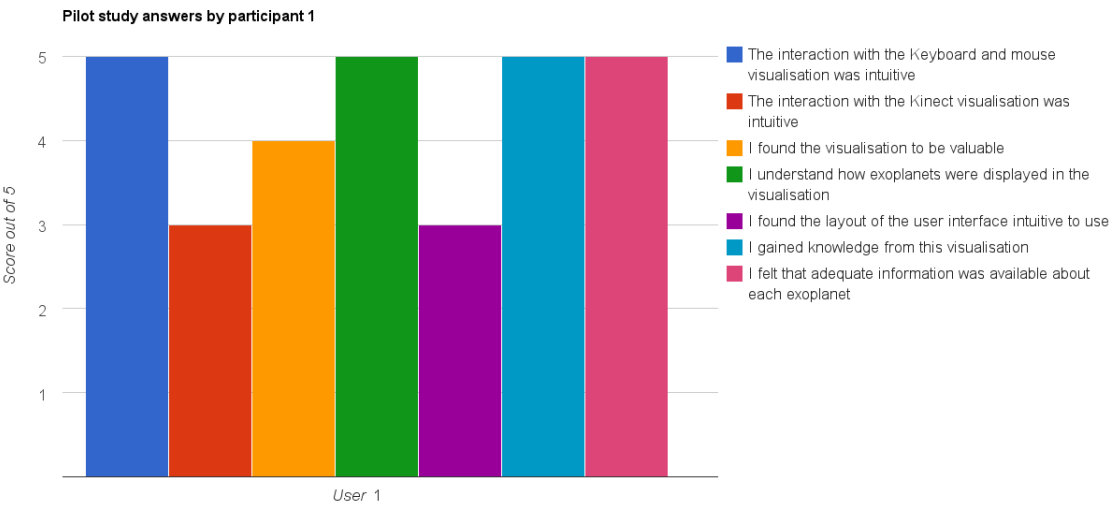

Figure 8.2: Pilot study results of P1

8.4   Results

The keyboard & mouse portion of the experiment was primarily intended to evaluate how the interaction techniques that were introduced into the visualisation can aid users in their information seeking behaviors. This was evaluated through the first part of the question set that users filled in whilst using the visualisation. These questions were designed in a way that encouraged users to make use of each of the interactive features that were introduced.

The Microsoft Kinect portion of the experiment was intended to evaluate how users would react to interacting with the visualisation by gesture. The questions for this portion of the evaluation were designed to find out users experiences with interacting via Kinect.
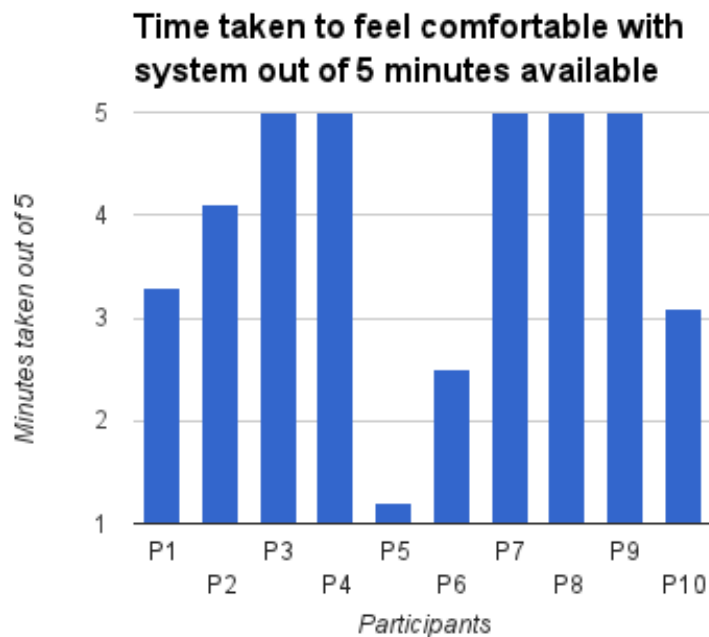
Figure 8.3: Amount of time taken for users to feel comfortable with visualisation

P2. P2 found that the information displayed in the IKVT was easy to follow even without knowing exactly what each piece of information actually meant.

P2 also found that without more of an in depth introduction into each of the tools

P2 Did not realise that it was possible to move the camera around until it was performed accidentally.

P2 had trouble when moving between questions on the worksheet due to forgetting to reset the range sliders after using them.

P2 sometimes got confused after changing the view multiple times which caused a loss of perspective.

"Do exoplanets actually exist"

P2 had trouble with Habitability zone

Spent lots of time playing with the Kinect sensor due to the novelty of it.

Had trouble when zoomed in to far with the Kinect sensor due to getting flustered

P2 predicted that if there was more time to use the visualisation they would have felt more comfortable and learned more rather than focusing on trying to use it.

Found the layout easy to use and simple to understand however found the range sliders slightly confusing.

Found the outline of the user in the background of the visualisation useful for maintaining perspective of where they were interacting with the system

Found that all of the information was layed out clearly in the text area. This information was interesting

The text size of the interaction panel could have been bigger to make it more easy to glance at.

P3.

P4. P4 found that the key strengths of the system was the ability to sort exoplanets, zooming.

P4 found that the range sliders were not intuitive. P4 also found that the zooming out speed was not fast enough in the kinect system. Also found that the rotation speed was to fast.

P4 suggested improvements of making a more consistent speed of zooming and making the range sliders more intuitive.

P4 found that the mouse system was more useful for gaining information due to increased accuracy and because the Kinect was more of a novelty and so more time was spent preoccupied by playing with it rather than looking at the text area.

Keyboard and mouse were easier due to being used to using them.

The body movements for the kinect were what P4 expected but the speeds at which the visualisation responded were not as expected

P4 did not have any experience looking at 3D visualisations and had only used 2D graphs.

P4 understood the attributes of the exoplanets bu found it hard to read them and interpret the visualisation at the same time.

Due to the size of the text in the information panel being size 12font it was difficult to read quickly. The range tool was also hard to use due to its size.

P4 also wanted more control over where the camera was centered.

P4 felt that more work needed to be put into making the information stand out at the user rather than needing to be "searched" for.

P4 felt that using IKVT broadened her perception of how information can be graphically represented.

P4 also felt that the names of some of the attributes were not made clear enough (Eg, ESI, KOI).

Notes
P4 seemed to have more difficulty using the visualisation than the other users.

P5. P5 found the kinect was fun and very engaging. He found that the use of size and colours of the visualisation were useful for making the information easy to understand

P5 also found the zooming effect that occurred when planets were filtered with the range sliders useful for making more accurate selections.

P5 found that some of the strengths of the visualisations were that is gave a sense of the space between exoplanets, the size and the vast numbers of them.

P5 found that because the information panel was not central in the visualisation that it required effort to break away from the main visualisation window which broke the immersion.

P5 felt that limiting the amount that the user could move around in the visualisation(stop being able to go to unnecessary places) would be an improvement, especially in the graph view.

P5 found that the mouse & keyboard system was the most effective for gaining information due to the accuracy that it afforded the user and the fact that it required less time to learn the gesture based controls and more time exploring the data.

Kinect required more of an introduction than was given

P5 found that having the visuals interactive and organised made it better to understand the complex information.

P5 felt that the layout of the interactive components were fine but could be improved by making them "pop out more"

P6. P6 did not initially spot the text areas changing as planets were selected until reading the worksheet

P6 had difficulty with habitability question.

P6 found that using the Kinect sensor was the most fun and enjoyed the method of hand recognition.

P6 also liked being able to change between views in the visualisation

P6 found that the interaction panel was "simple - not filled with to much information" and so was easy to use.

She found that the use of 3D components and being able to change views, move the camera around for better viewing angles, and zooming in and out were some of the key strengths of the system.

P6 also 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.

P6 felt that a weakness of the Kinect sensor was that it detected the whole hand rather than a more controllable interface like a finger.

However P6 did find that the 3D camera was sometimes difficult to control

A suggestion for the system was that smoother movements of the camera and planets could be incorporated and refine it to make it more user friendly so that it can be used without any explanation.

A further improvement would be to make the menu more aesthetically pleasing.

P6 found the mouse & keyboard system the easiest to use because it was the interactive method that she had the most experience with.

P6 found that the interactive method that allowed the most learning was the mouse & keyboard. This was because more time was spent with it and because she was closer to the screen and so could read the textual information better. P6 found that she used more time playing around with the Kinect system trying to select planets rather than trying to get information.

P6 felt that the seating location was to far away to accurately see what she was selecting with the Kinect sensor.

P6 noted that a system like this would be perfect for use in schools and education centers and would interest many people.

P6 felt that a key improvement would be to have used more 2D views over 3D views and to make the interaction panel and visualisation panel more aesthetically similar to "connect them together".

P7. P7 found that the visualisation was very easy to use and the way that she was able to control the exoplanets was fun.

P7 found that a weakness of the system was that it displayed to much information at once which was confusing.

P7 also found that the keyboard and mouse system was the most useful as it felt more natural and allowed access to more information as well as making it easier to access the information she wanted.

P8. P8 liked being able to toggle between the different planetary sorting and filtering. He also found that the Kinect system was more fun to use but lacked the control that the keyboard & mouse allowed.

P8 also found that the visualisation was informative and a "neat" way of visualising what would normally be stacks of information that he would struggle to navigate.

P8 found that a weakness of the system was that there was a lot of concepts that needed to be understood about the planets and so a glossary would have been an improvement.

He also found that the lack of movement space while using the Kinect made it harder to use it freely.

He also found that overlapping planets made it hard to make selections with the Kinect. He also would have liked a way to cancel a selection without having to select another planet.

P8 found that the keyboard & mouse system gave more control and access to the filters and sorting.

P8 found that he learned many things whilst using the visualisation and gained an interest in learning more about Exoplanets which he previously had no real interest in.

P9. P9 found it very easy to make comparisons between exoplanets due to the sorting ability.

P9 enjoyed the Kinect side of the visualisation due to it allowing her to get a sense of the scale of exoplanets found.

P9 found that the strengths of the visualisation with regard to the Kinect system were that it allowed a fun way of interacting with a serious visualisation. The ability to rotate the system with its realistic 3D as well as select specific planets and zooming in an out made it very effective.

The keyboard & mouse side of the visualisation provided an effective way of accessing the information thanks to the sorting and filtering ability.

Initially P9 was confused as to where the information was displayed about each selected exoplanet but once she realised it was very easy to use.

P9 found the keyboard & mouse system more effective but enjoyed the interaction with the Kinect system more.

P10. P10 found the animations in the visualisation were enjoyable whilst exploring

He found that it was easy to compare the information for each exoplanet and to get a sense of the quality of exoplanets.

P10 found that the interaction panel was good but its location was awkward with the sliders down the left hand side of the screen.

He found that the keyboard was the most effective interaction method for exploring the data but the novelty of the Kinect sensor was more fun due to its novelty

P10 felt that the interaction panel felt like it was hidden and that there was to much information displayed in it.

Observations. All users had trouble with the understanding the habitability zones of stars due to there being no standard habitability to compare.

All users that had extensive experience with 3D software adopted the system much faster than those that hadn't and were in general more comfortable.

All users enjoyed the Kinect version more than keyboard & mouse even though it was harder to gain information from the visualisation

Once users worked out how to use the range sliders they used them extensively

8.5 Performance evaluation

Entire render cycle takes less than

8.5.1 Evaluation of Requirements

Evaluation of Functional Requirements

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

Successful

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

Successful

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).

Successful

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

Successful

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

Successful but due to unintuitiveness the usability is reduced

Evaluation of Nonfunctional Requirements

R6. All interaction methods must be visible and intuitive.

Successful but needs to be more prominent

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

Successful

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

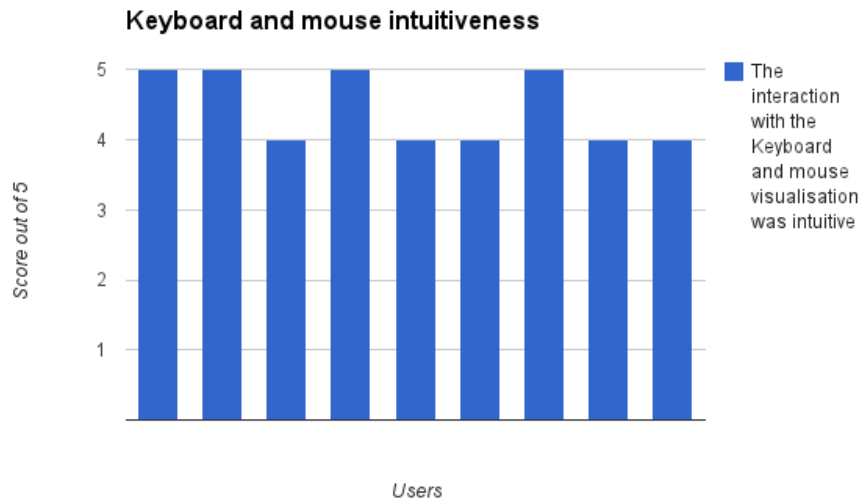Successful but Kinect was worse for information seeking.

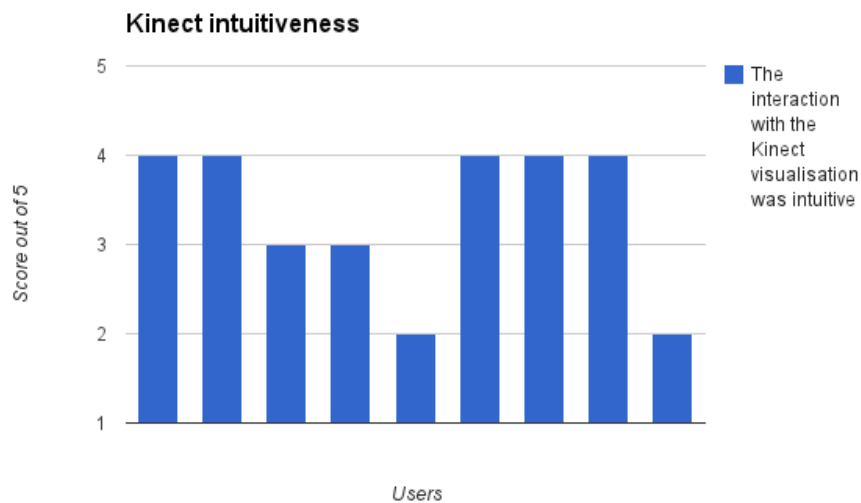Figure 8.4: Intuitivity of keyboard and mouse

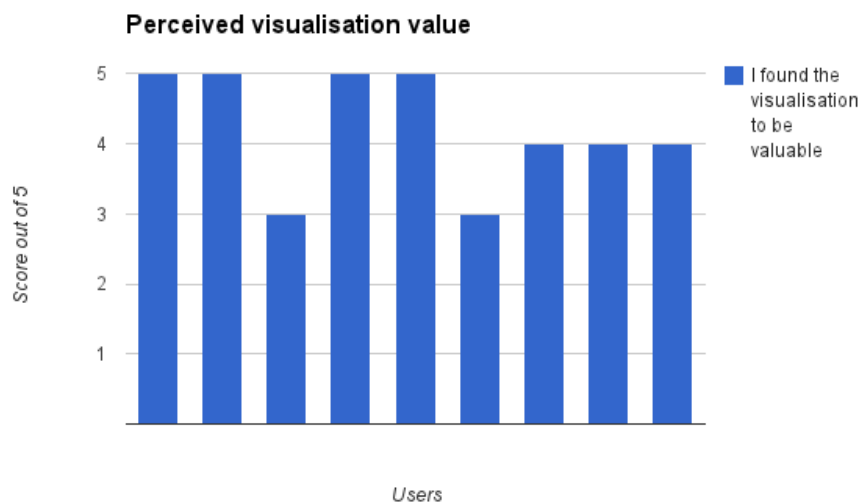

Figure 8.5: Intuitivity Microsoft Kinect sensor

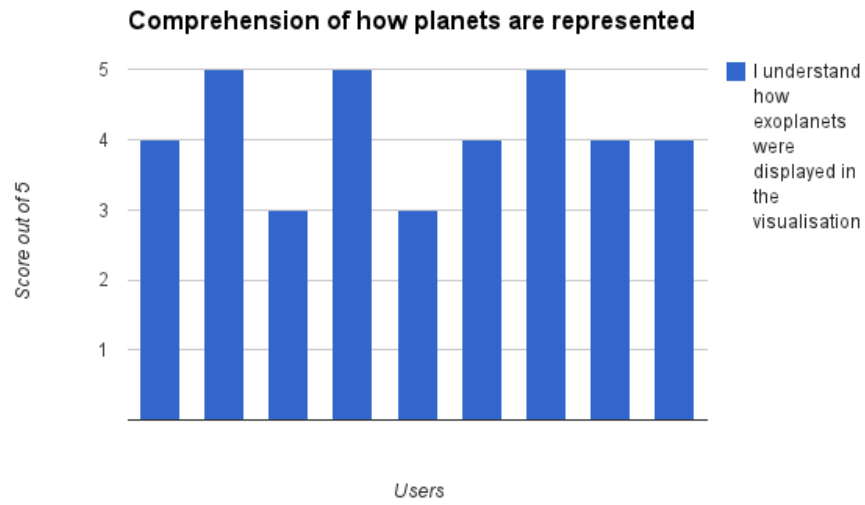Figure 8.6: Perceived value of visualisation
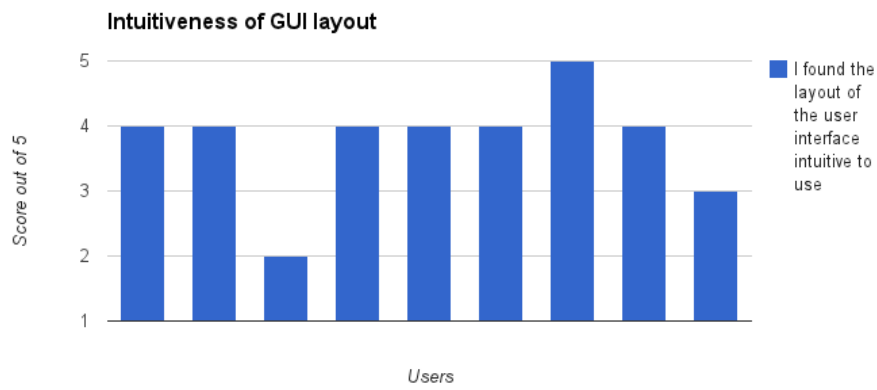
Figure 8.7: User comprehension of visualisation
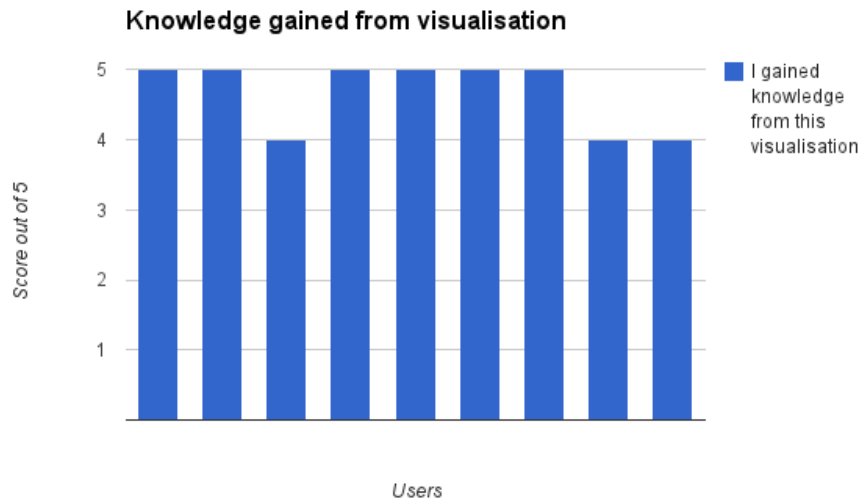

Figure 8.8: GUI layout intuitivity

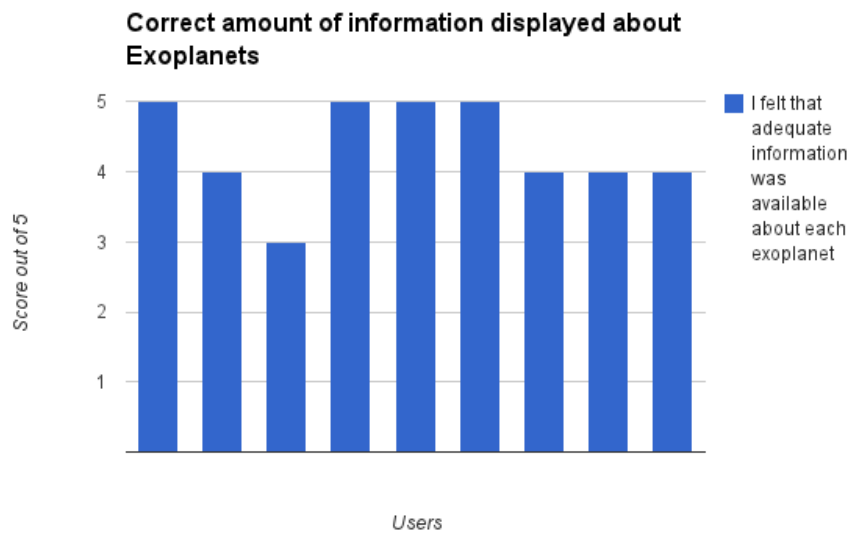Figure 8.9: Knowledge gained from the visualisation

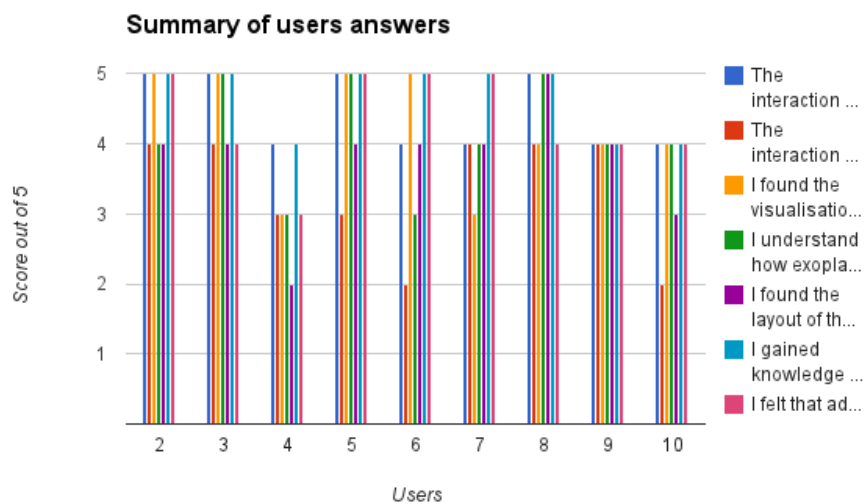Figure 8.10: Correct amount of information displayed

Figure 8.11: Summary of results

8.6 Discussion

Lack of control system. Not evaluated in the evironment it is designed for ie observatory. Not enough instruction Not all participants were comfortable moving around in 3d space Some users had more trouble learning how to use the system than others Users taking less time to become comfortable may not have experimented enough with the visualisation

8.7 Supporting Tools for project

By supporting this project methodology with other project management tools such as Gantt charts [APPENDIX] and work breakdown structures(WBS) [APPENDIX], it 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 I could swap between. This was important for the creation of the Keyboard and mouse system and the Microsoft Kinect system as they were on separate branches in a repository.

The version control tool that was used for this project was Git which allowed for the repository to be hosted on the repository hosting web service Github. This version control option was chosen due to the ease of use of as well as my prior experience with Git which has all been positive. There was a minor limitation of this choice as I used a free Github license which meant that the repository would be open to the public.

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 meeting ensured that vital components and deliverables were implemented in the required timeframe and also provided a sounding board for ideas for elements to be included in the visualization. Another important aspect of having an involved supervisor was that he provided me the guidance of an experienced academic which was indispensable when navigating the administrative side of organizing delicate matters such as ethics approval for human evaluation of the visualisation.

Chapter 9

Conclusions

9.1 Future Work

The work from this project can be taken further in many different 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 for the system 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 he expectation that it could be used by users. The systems current state could me modified to fit into either of these two options.

Occulus rift, look at paper boy

Bibliography

[1] Celestia. http://www.shatters.net/celestia/.

[2] Controlp5. www.sojamo.de/libraries/controlP5.

[3] Github. https://github.com/.

[4] Kinect. http://www.xbox.com/en-GB/kinect.

[5] Nite. www.primesense.com.

[6] Simpleopenni. code.google.com/p/simple-openni.

[7] 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.

[8] BLPRNT. Github:kepler visualisation tool. https://github.com/blprnt/Kepler-Visualization.

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

[10] FABRYCKY, D. The kepler orrery, 2012. http://kepler.nasa.gov/multimedia/animations/?ImageID=219.

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

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

[13] 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.

[14] PARKER, A. Worlds: The kepler planet candidates, 2012. http://kepler.nasa.gov/multimedia/animations/scienceconcepts/?ImageID=229.

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

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

[17] 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.

[18] 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.