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. This can be accessed from a database by anyone. However this information is complex and cannot be easily understood by laypeople. This is a problem as it means that the information gained about these planets is not being used effectively to convey the intended information to the masses. To resolve this a visualisation has been created that can convey this information in a way that interested lay people 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, the planning and decisions made, the visualisation created, and its evaluation to discover its effectiveness at fullfilling the goals driving its creation.

Acknowledgements

Id 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 free to help and 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 Exoplanets dataset ((REF)). This deliverable is intended as a standalone 3D visualisation with two modes of interaction, keyboard and mouse or Microsoft Xbox Kinect sensor ((REF)). The resulting visualisation will convey the information in the dataset in a way that the target users, laypeople who have an interest in astronomy, can understand and interact with.

1.1 Motivation

There are many planets that have been located outside of our own solar system, these are called exoplanets, these are referred to interchangeably as planets and exoplanets for the remainder of this report . Information about these exoplanets are stored in the Kepler Exoplanet dataset ((REF)). This project seeks to develop and evaluate an interactive 3D visualisation software system for the Kepler exoplanets dataset [28] . We can leverage how humans respond to visualisations by experiencing more enjoyment and more immersion, both of which result in improved learning and recall. This means that creating a effective and engaging visualisation will help convey the information in the dataset effectively to the users.

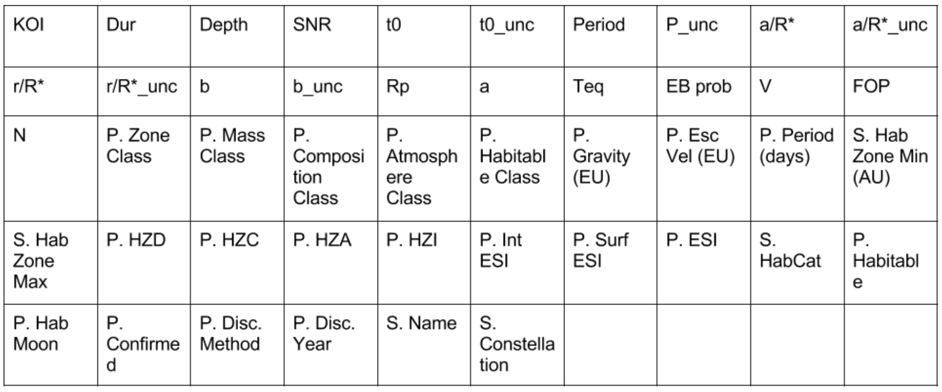
1.2 Problem statement

The complex nature of the data involved in this project causes a range of problems revolving around understandability to arise, which this project attempts to address. The following subsections outline these in detail.

1.2.1 Understanding the content in the dataset

Understanding and analysing large datasets whose size defies simplistic or trivial analysis by humans is a known issue that many areas of research are attempting to address. These areas of research range from data mining to visualisations in order to discover or highlight important features in the data so that people can more efficiently use or understand 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 http://nrich.maths.org/6447. This allows for a much more comprehensive understanding of the content being visualised. 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 which contains 46 fields. It is next to impossible for someone to internally visualise so much information, especially as most of it is floating point numbers. This means that an external way of visualising it is needed, which is one of the problems that this project attempts to address.



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

1.2.2 Comprehension of planetary information

Much of the information regarding planets is cryptic and unintuitive, this make its understandability difficult. Visualisations in general attempt to address this by displaying data in simplistic ways that allows improved user comprehension.

1.2.3 Existing solutions lack functionality

Existing data visualisation techniques using this exoplanet dataset lack the ability to display sufficient detail for each exoplanet and do not fully utilise the data in the dataset. 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 does leave a lot of potential information unseen and overlooked. For example, information about the type of planet, planets with similar traits, solar system information, similarity to earth and habitability. This project will therefore be focused on researching, implementing, and evaluating a new interactive visualisation system that will display additional information to contained in the dataset but not included in existing visualisation systems.

1.2.4 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 omnidirectional treadmills or even a combination of devices.

With interactive visualisations, response time of the system to user actions is important and so changes made by the user must be incorporated into the visualization in a timely manner. Experiments have shown that a delay of more than 20 ms between when input is provided and a visualisation is updated is noticeable by most people ((REFERENCE)). Thus it is important for an interactive visualization to provide a rendering based on human input within this time frame or else risk breaking user immersion.

1.3 Key issues project addresses

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

- I1. Content in database form is difficult to view and understand.
- I2. Planetary information is complex and difficult to comprehend without a visual reference.
- I3. Existing visualisations for this dataset have minimal functionality and have lacking usability.
- I4. User interaction is needed in a visualisation to make the most of data displayed.

1.4 Contributions of this project

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

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 Methodologies

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 successfull project. It is benificial 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 [**?**]. 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 consistancy. (REF BIG BALL MUD)

The project management methodology/software development lifecycle chosen for this project was a customized Spiral Model (REF BARRY BOEHM) 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. Finaly 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.

This project management 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 provide SOMETHING HERE the project through the necessary steps to end in 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 occured 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.

The advantages of this methodology over other choices such as the waterfall model or an agile approach such as Scrum 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. Following a pure waterfall methodology would not have

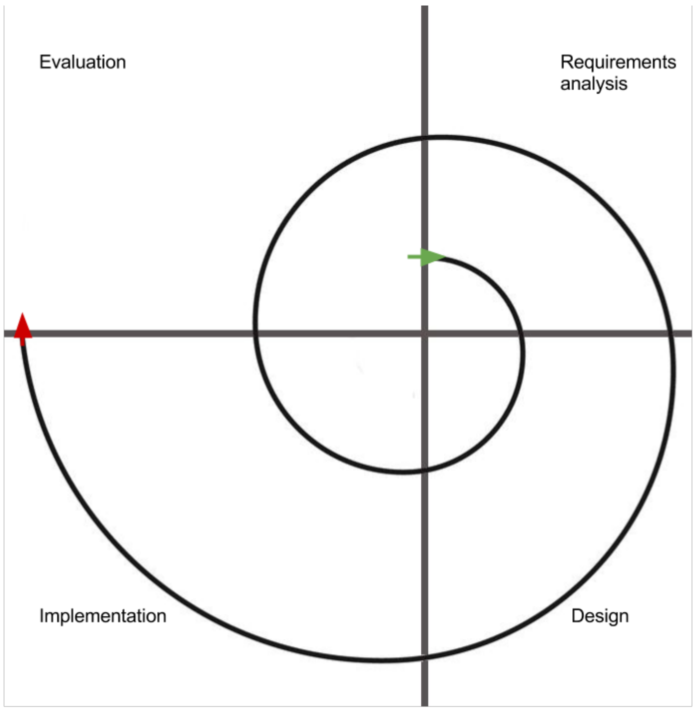


Figure 2.1: Spiral process model followed

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 [REFERENCE]) 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) regularily, emphasized working software over large amounts of documentation (REFERENCE AGILE MANIFESTO).

Following this project managment approach allowed me to achieve all of the goals that I set, although it did not always happen within the timeframes 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 Kinext sensor which required further planning, implementation and testing. Although there were these delays, due to the planning and project managment approach that was used, all project elements were completed.

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.

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.

2.2 Difficulties in project

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. This project had a time constraint of approximately 300 hours over the course of a year, this meant that I needed effective time management techniques to ensure that I spent the right amount of time on each project element, that I prioritised important tasks, set appropriate deadlines, and planned each stage of the project effectively.

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) which 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.

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.

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.

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.

3.2 Scenarios

A good use scenario does a number of things:

- Describes the user's goals and motivations.
- Describes a specific task or tasks that need to be accomplished.
- Describes some of the interaction, with enough detail to make it compelling, but not so much detail as to be overwhelming.
- Provides a shared understanding for everyone on your team about what a user might want to do and how they might do it.
- Helps you construct the sequence of events that are necessary to address in your user interface.
- Can be sketchy, as long as it provokes ideas and discussion.

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

Primary Persona:
When John first sees the system the first thing he notices is that their 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 clicks a prominent button stating view planet similarity to earth.
2. The planets on screen move so that the earth is located at the center and top of the screen with all others orbiting it. The planets with a high internal similarity to earth are higher on the Y axis whilst the planets with a high surface similarity to earth are closer to the center of the orbiting planets on the X axis.
3. From here John can select any of the planets for further analysis.

3.2.2 Scenario 2: Select ranges for attributes of each planet displayed

Primary Persona:
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.

3.2.3 Scenario 3: Select planets to display more information

Primary Persona:
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.

Secondary Persona:
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.

3.2.4 Scenario 4: View planets in the same solar system

Primary and Secondary Personas:
When a planet it selected John and Cara want all ofter planets that are in the same Solar System as the selected planet to become highlighted.

Procedure:

1. When a planet is selected, all planets in the same Solar System become larger and its outline grows, making it more visible.
2. A label appears on these planets indicating that they are related planets.

3.2.5 Scenario 5: View the Goldilocks zones of each planet

Primary Persona:
John wants to see which planets are in the habitable zones of their stars.
Procedure:

1. John clicks a button saying ”Show habitable zones”
2. Coloured rings appear showing the cold (blue), habitable (green), and hot (red) zones of the selected planets star.
3. When a planet from a different star system is clicked the coloured rings will change to that stars zones.

3.2.6 Scenario 6: Select two planets to compare against one another

Primary Persona:
John wants to be able to compare two planets against one another. **Procedure:**

1. When John selects a planet a button becomes ungreyed called ”Compare” with a note next to it saying ”Please select another planet to compare to”.
2. When the second planet is selected a second text box fills up with the information about the second planet. This information can be compared with that in the first text box.

3.2.7 Scenario 7: Navigate the visualisation with gestures

Secondary Persona:
Cara wants to be able to navigate around the visualisation by using hand gestures **Procedure:**

1. By moving her hand to the edges of the screen the visualisation with pan in the correstponding 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.

3.3 Requirements summary

3.3.1 Functional Requirements

Functional requirements define the functions of a system. 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 have two text areas to allow the more detailed textual information about planets to be displayed.

R2. The planets need to be able to be ordered by their similarity to earth (ESI) and by their Kepler Object of Interest number (KOI).

R3. The visualisation needs to have a set of sliders that can be used to change which planets are visible by setting ranges of their core attributes.

R4. All planets need to be selectable and react appropriately when clicked.

R5. When a planet is selected all other planets in the same solar system need to become highlighted.

R6. Visualisation needs to have the option to view the habitable zones of stars and show where the planets orbiting them are in relation.

R7. There needs to be the option to select two planets at a time to compare against one another.

3.3.2 Nonfunctional Requirements

Functional requirements are supported by non functional requirements. Non functional requriements 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:

R8. The visualisation needs to have a range of interactive buttons for each element of interactivity in the system to make it more visible to users how to use the system.

R9. All interaction methods must be visible and intuitive.

R10. The visualisation needs to display attributes of each of the Exoplanets.

R11. The visualisation must remain uncluttered.

R12. The visualisation must not show so much information that it causes information overload for users.

R13. There needs to be two modes of interaction with the system, keyboard and mouse vs gesture based.

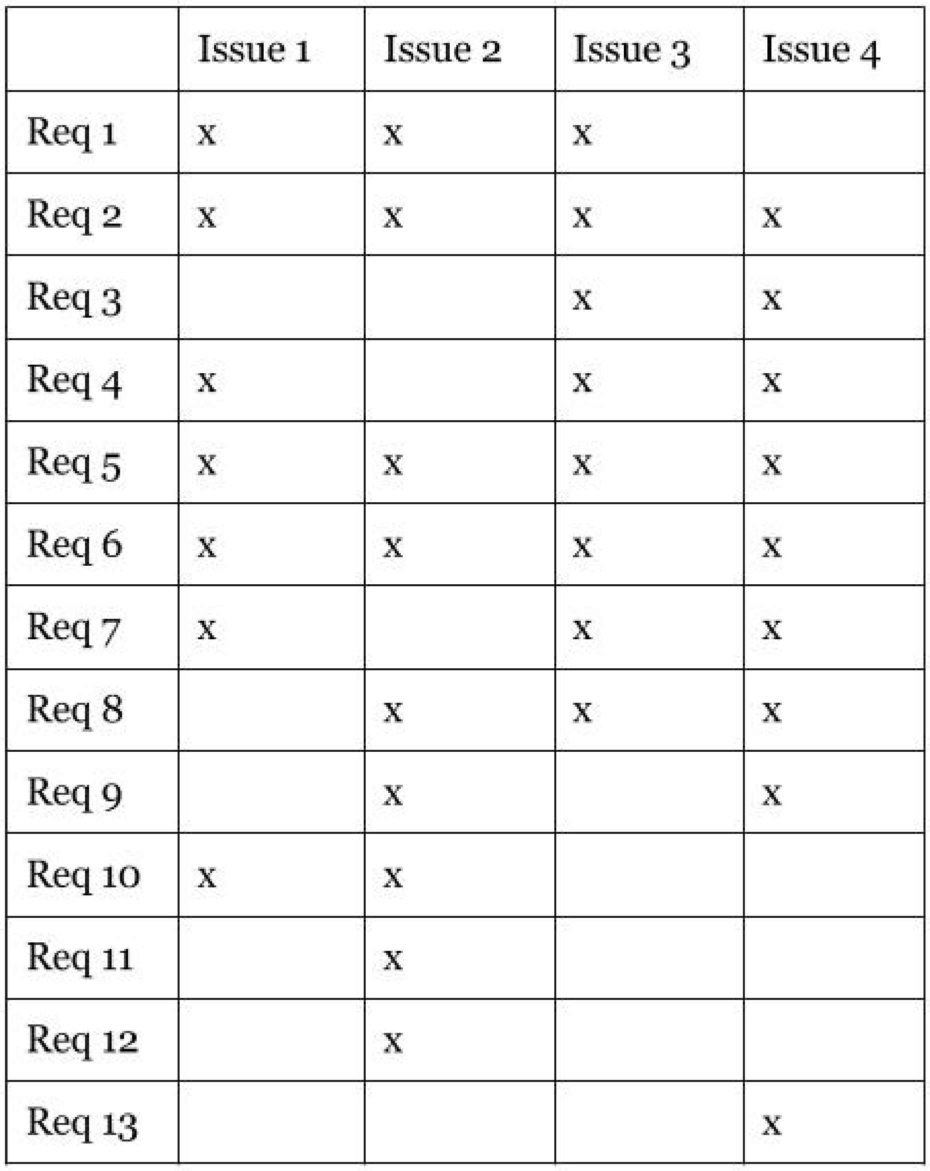
3.4 Existing systems

3.4.1 Worlds: The Kepler Planet Candidates - Non Interactive

This animation [?] 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. 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



	Issue 1	Issue 2	Issue 3	Issue 4
Req 1	x	x	x	
Req 2	x	x	x	x
Req 3			x	x
Req 4	x		x	x
Req 5	x	x	x	x
Req 6	x	x	x	x
Req 7	x		x	x
Req 8		x	x	x
Req 9		x		x
Req 10	x	x		
Req 11		x		
Req 12		x		
Req 13				x

Figure 3.1: Matrix of project requirements to issues project is attempting to solve

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.

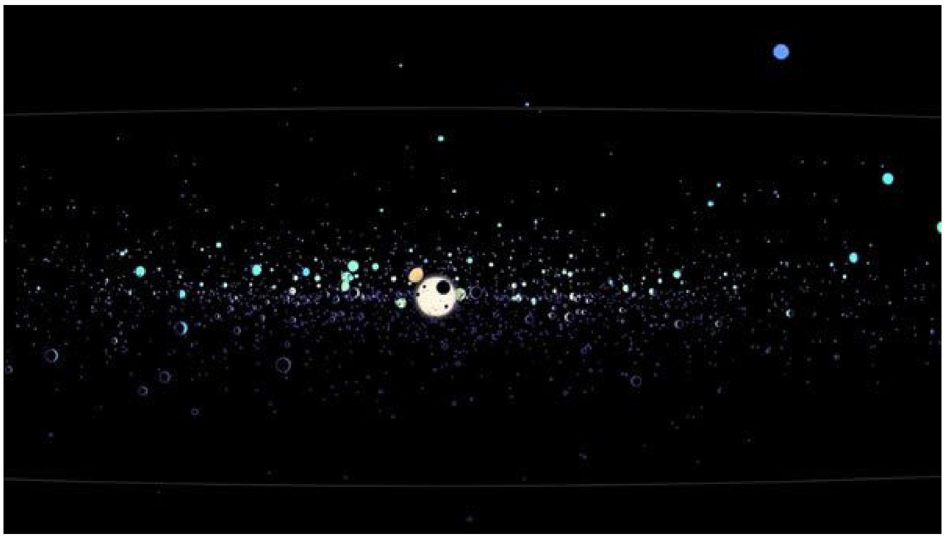


Figure 3.2: Image of Worlds Visualisation
git a

Scenario 5. View the Goldilocks zones of each planet:
Worlds does not offer this functionality.

Scenario 6. Select two planets to compare against one another:
Not applicable as worlds does not allow selections.

Scenario 7. Navigate the visualisation with gestures:
Not applicable

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

The Kepler Orrery [?] 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. 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 Exo-planets 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.

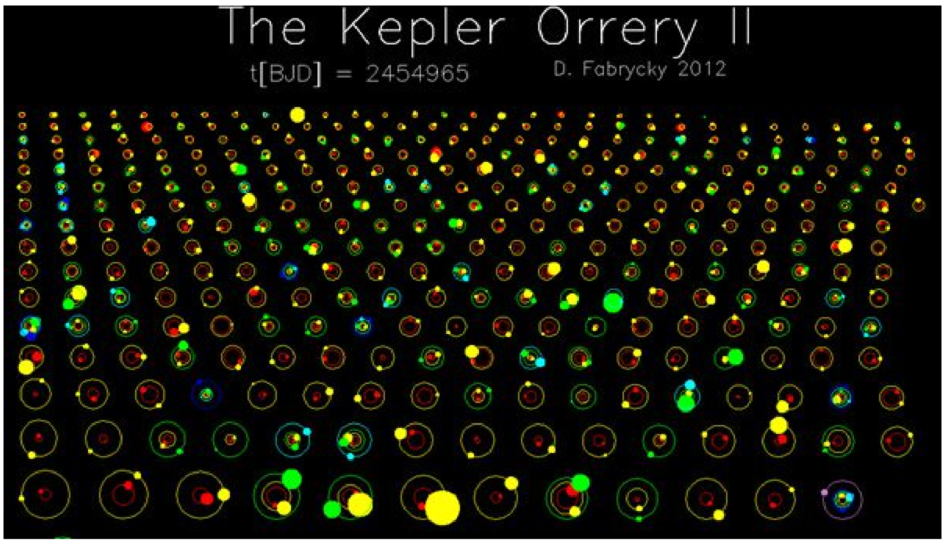


Figure 3.3: Image of The Kepler Orrery Visualisation

Scenario 3. Select planets to display more information:
Not possible

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 tbe the case with Worlds.

Scenario 5. View the Goldilocks zones of each planet:
Not a feature

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

Scenario 7. Navigate the visualisation with gestures:
Negative

3.4.3 Celestia - Interactive

Celestia [?] 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++. 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

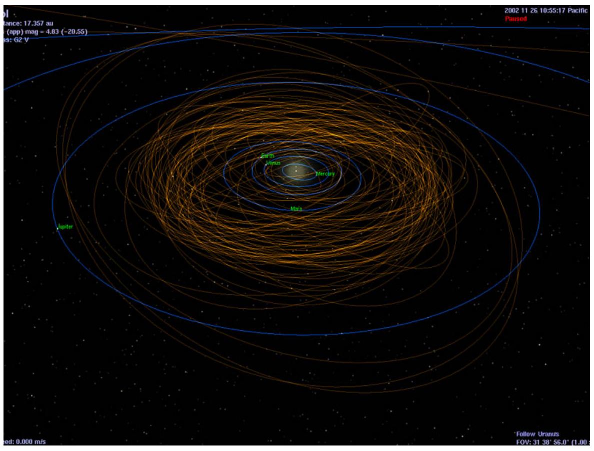


Figure 3.4: Image of Celestia Visualisation

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

3.4.4 Kepler Visualisation Tool

An existing system built with Processing is the Kepler Visualisation Tool[**?**, **?**]. 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. 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:

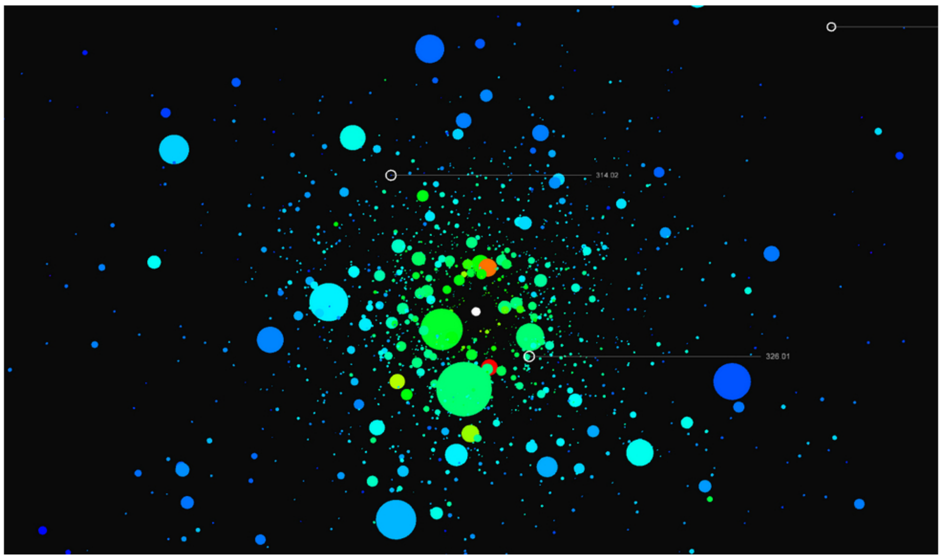


Figure 3.5: Kepler Visualisation Tool Orbital View

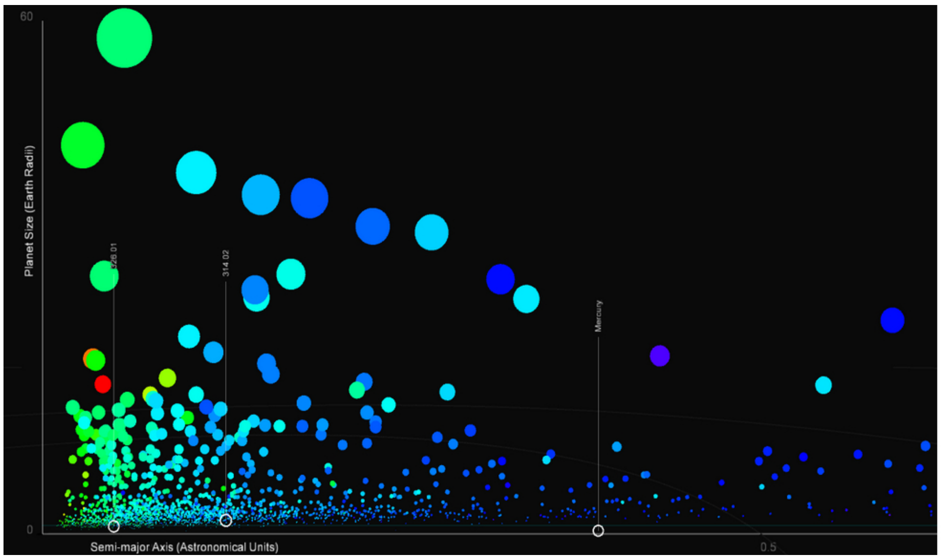


Figure 3.6: Kepler Visualisation Tool Graph View

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 limmited functionality of comparing these to earth which the others lack. It does this by displaying Earthm, 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

Scenario 3. Select planets to display more information:
There is no selection functionality in the Kepler Visualisation Tool which means that a lot of

the potential information about each Exoplanet goes undisplayed.

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

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

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

Scenario 7. Navigate the visualisation with gestures:
Nope

3.4.5 Summary of Existing Applications

Matrix compared to requirements

3.5 Analysis of technology options

Many technologies were looked into,experimented with, and the positives and negatives of each weighed up before a decision was made about which would be the 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.

3.5.1 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.

D3 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.

3.5.2 Prefuse

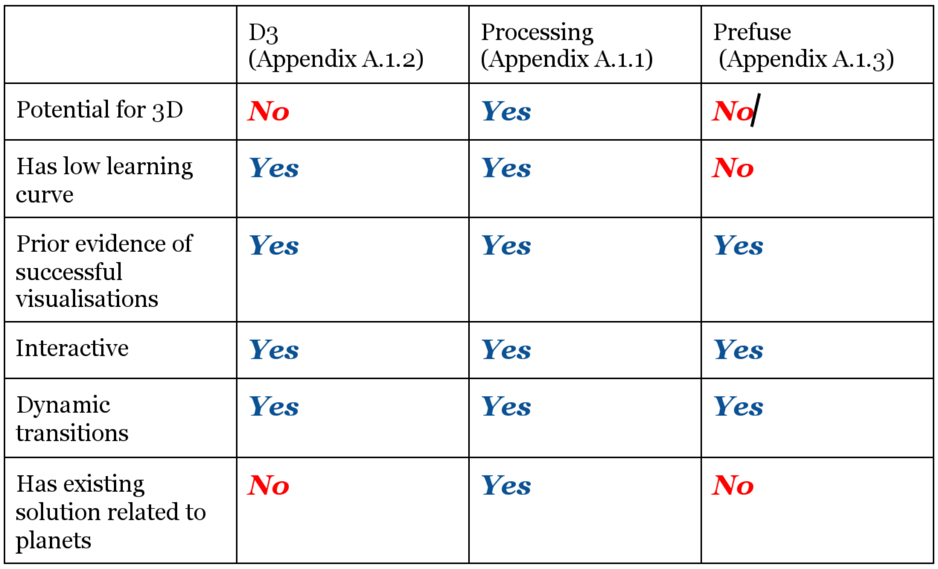
Prefuse is a set of software tools for creating rich interactive data visualizations [13]. The Prefuse toolkit provides a visualization framework for Java. It supports a set of features for visualizing and interacting with data. It provides optimized data structures for tables, graphs, and trees. It 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.

3.5.3 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 wouldnt be possible with D3.

3.5.4 Decision of technology

The final decsion of techology was to use Processing, this is because it had many positive aspects that the others did not and minimal nigatives as the below table illistrates.



	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 3.7: 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 [**?**, **?**]. As the time was short for this project builing 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 4

Solution Design: Improved Kepler Visualisation Tool (IKVT)

This section discusses the design of the deliverable visualisation, The Improved Kepler Visualisation Tool(IKVT) that was created for this project. It details the key design decisions revolving around structure, asthetics, and functionality that were made about the visualisation. To help users to understand more about the planets outside of our solarsystem, this project aims to improve an existing visualisation, The Kepler Visualisation Tool that was discussed in the previous chapter, which whist displaying exoplanets and some of thier features, it lacks interactivity for users trying to use it to help comprehend the ... . The IKVT expands on the preexisting Kepler Visualisation Tool by adding key elements of interativity that are missing with the existing visualisation as well as further enchancing the range of data that is available to users about each exoplanet.

To aid in the comprehesion of they exoplanets IKVT displays all .... exoplanets in the .... datase. 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 comparisions about each of the exoplanets in a single view.

To make this selection, a user can click on any of the orbiting exoplanets. A further 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.

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. These filters 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 to see which allows them to work with small multiples rather than the entire dataset.

In addition to the orbital view already discussed, there is a graph view taht 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.

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 interchangably to display information about selected planets, four range sliders that are used to filter the exoplanets as discused 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″.

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 hightlighted. This will provide the same effect as in the Kepler Orrery.

4.1 System design and structure

BASIC Class structure
Data structure
UML CLass Diagram
Sequence Diagram

4.2 Design Features

4.2.1 Main interface components

The visualisation was designed to emphasis small multiples and filtering of the Exoplanets to display the information more clearly to users.

Instead of the visualisation only answering the 5 key questions as proposed in the proposal, the aim will now be displaying as much of the information in the dataset as possible without detracting from the effectiveness of the visualisation whilst still answering the questions.

This is because the 5 questions did not fully utilize the information in the dataset.

However I will need to ensure the effectiveness of the visualisation does not become diminished by trying to convey to much information which would lead to cluttering and overlapping in the visualisation, as well as information overload for users. There will also be larger emphasis placed on making the existing system more usable by improving the interaction methods for users. The following list outlines the new requirements for the visualisation being developed. This will be done by providing GUI elements for each form of interaction with the system, as well as ensuring all interaction methods are intuitive for users.

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.

Navigation Window(BETTER TITLE)

Description of navigation window

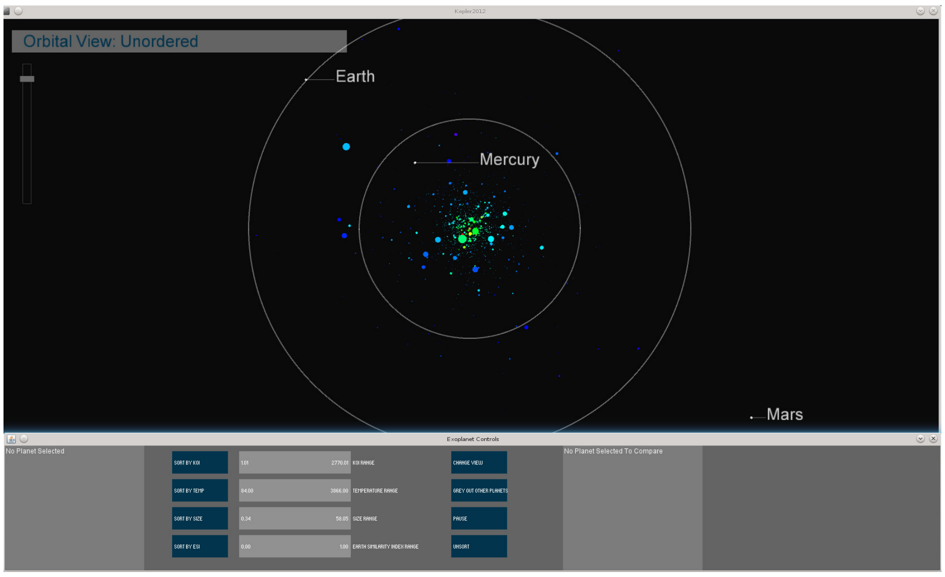


Figure 4.1: Original Horizontal Layout

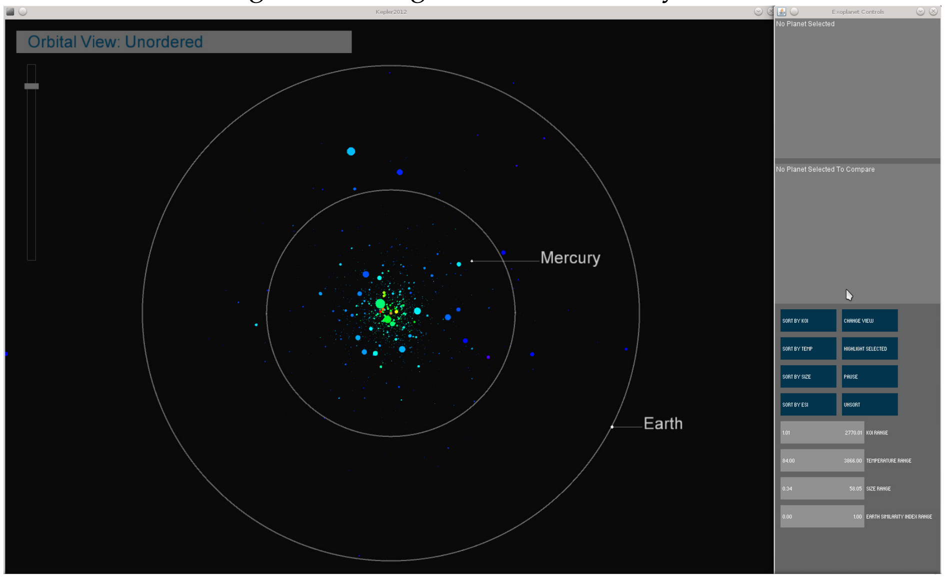


Figure 4.2: Improved Vertical Layout

Description of each button

Description of each slider

Description of text boxes

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 visualistaion 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 labelled interactive elements and tooltips explaining what they do.((REFERENCE)) . 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.

Visualisation Layout for Kinect sensor

As the Kinect sensor no longer requires the use of a mouse the visualisation design needs to be modified to accomodate the use of gestures. For this project this meant incorporating new cursers to indicate the state of the visualisation. There are 7 states that the curser needs to be able to be in to inform the user of what action they are performing. These states are

1. default curser, hand is at rest
2. panning up, hand is raised
3. panning down, hand is lowered
4. panning left, hand is to the left
5. panning right, hand is to the right
6. zooming in, hand is pressed forward
7. zooming out, hand is pulled backwards

Having a range of icons that clearly display these states is vital for keeping the user informed of what they are doing. The icons designed for this purpose are in the following figure

Figure 4.3: Cursers for Kinect sensor

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.

Chapter 5

Visualisation implementation

This chapter discusses the implementation of the visulisation using the designs discussed previously. It details the tools used, the deliverable features produced, and the problems encountered

5.1 Additional Tools and artifacts used

5.1.1 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 REF . 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.

5.1.2 Microft 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 ref
2. SimpleOpenNi ref

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 friendy or effective as the official libraries. The effect of this was that the gesture tracking used in the system had to be created supoptimally from the opensource libraries.

5.2 Implementation of planned features

All of the features that were planned were completed for the visualisation, this is including the additional work that was planned around the Kinect sensor.

Below is a description of each of the main visualisation features that were completed. Image of navigation window

Image of each slider Image of text boxes Image of using kinect system

Image of using non kinect system

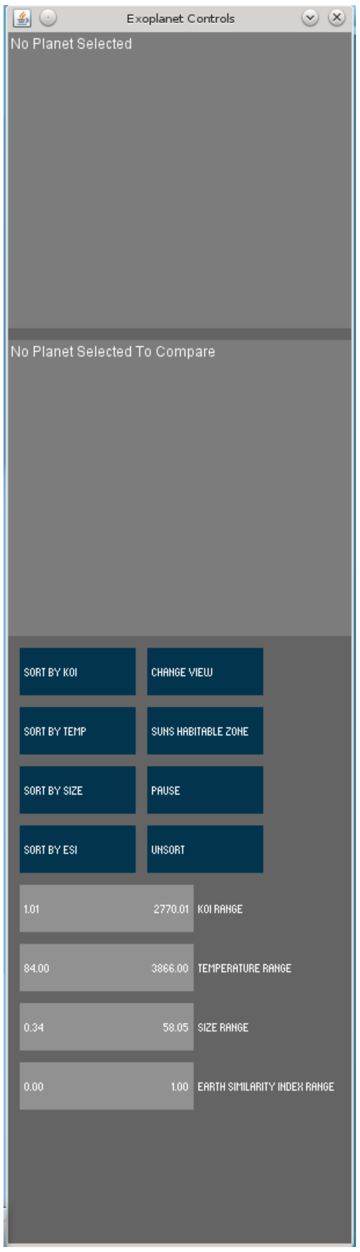


Figure 5.1: Navigation Panel Overview

5.3 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 elipse 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

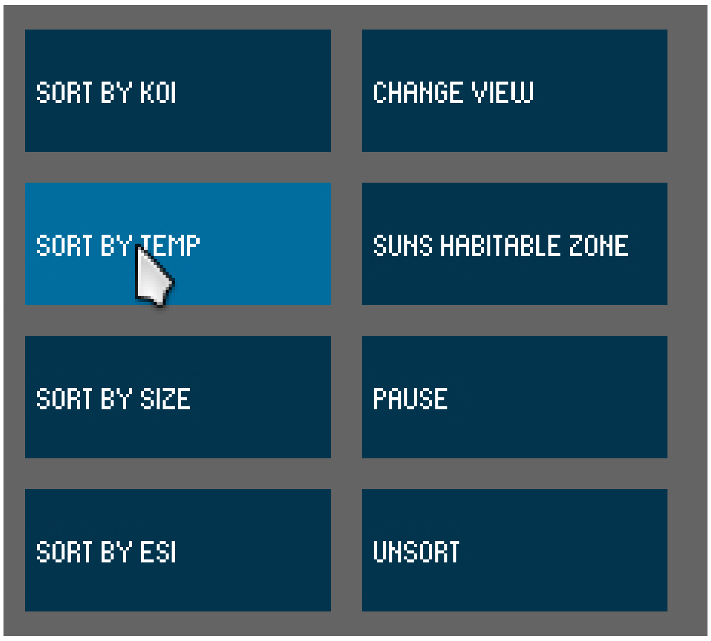


Figure 5.2: Panel of interactive buttons

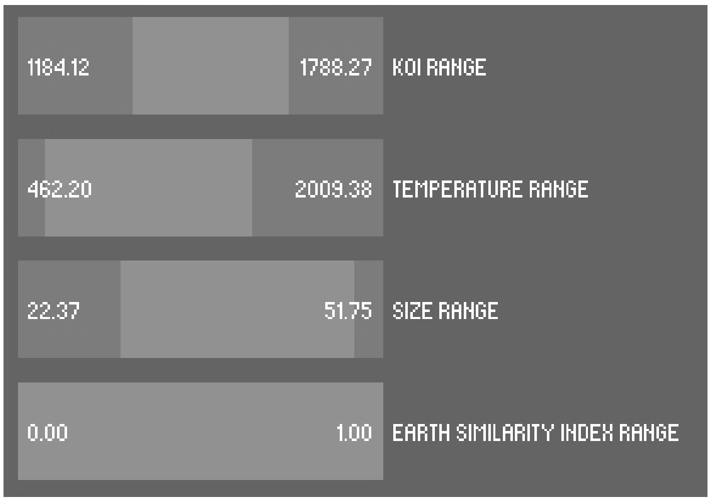
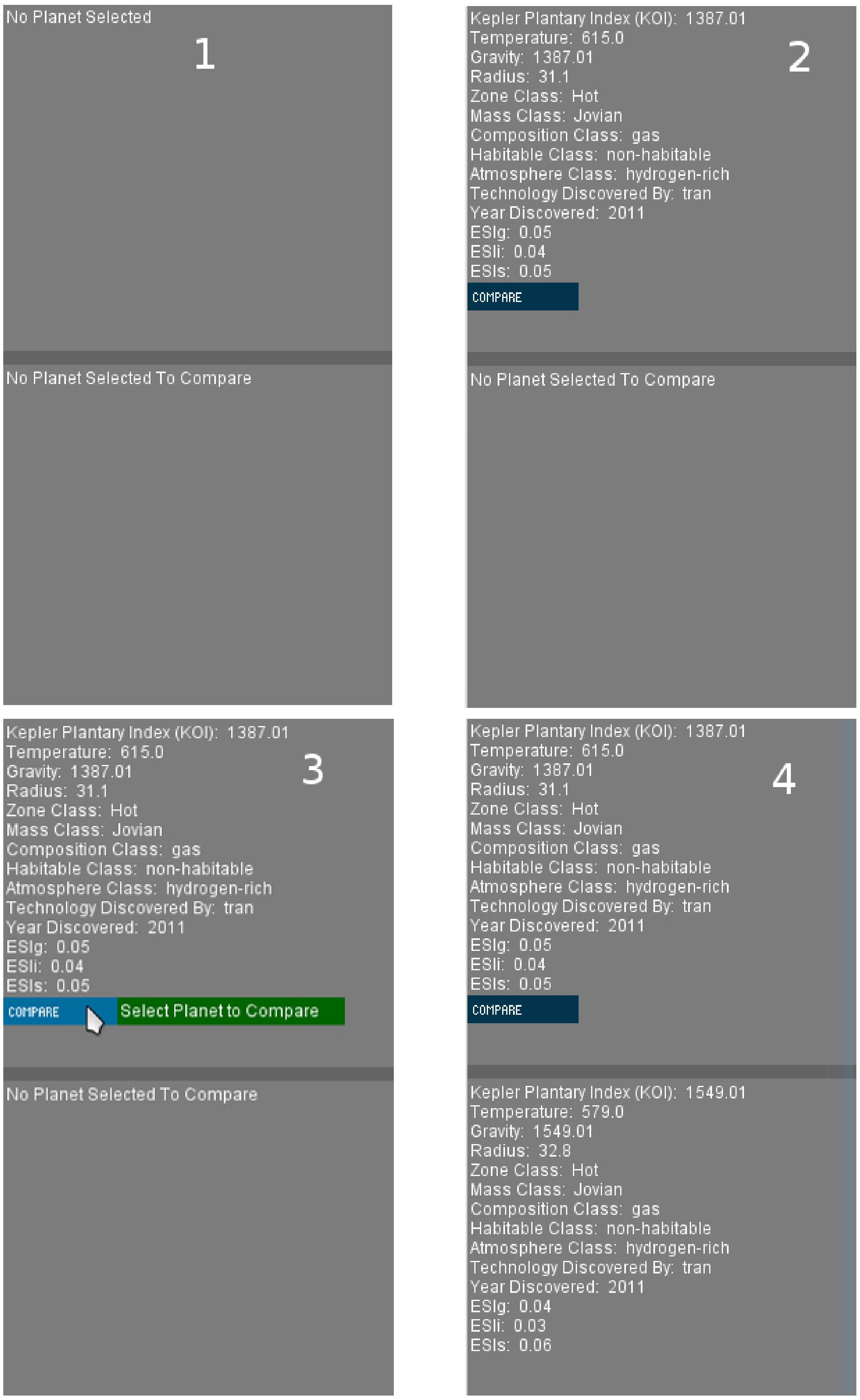


Figure 5.3: Panel of interactive range sliders

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 regularily commited any changes to my version controlled system on Github ((REFERBTECE)). 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.

Performance limitations SCREENSHOT



No Planet Selected

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
ESIg: 0.05
ESIi: 0.04
ESIs: 0.05

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
ESIg: 0.05
ESIi: 0.04
ESIs: 0.05

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
ESIg: 0.05
ESIi: 0.04
ESIs: 0.05

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
ESIg: 0.04
ESIi: 0.03
ESIs: 0.06

Figure 5.4: Text boxes in each possible state

Chapter 6

Visualisation Evaluation

Following the completion of the implementation stage of this project a final user evaluation was carried out on the visualisation to discover whether the visualisation designed and implemented

Chapter 7

Conclusions

7.1 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 limmited 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 benificial 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 witht 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