Space Hop

3rd Year Project

Functional Specification

- Darragh Connaughton
    13469978

- Thomas Feeney
    13567763

1. Introduction

1.1 Overview

Space hop is an interactive space explorer game where the user will be able to traverse the solar system learning about our neighbouring planets, moons and other such man driven space missions and advancements. It will provide a 3D environment in which the user can freely explore space and visually interact with the to-scale environment. Users will be able to pause the game at any time and retract to a dynamic 2D view of the solar system, where they will be able to observe the orbit of each planet around the sun and choose their next destination to explore.

The audience of our project will be aimed at primary school students, and will be mainly educational. Our vision is to create an interactive and immersive environment, providing school children with a fun, visual way of learning about our solar system which isn't confined to static images from textbooks or boring rhymes from which to remember the ordering of the nine planets!

1.2 Business Context

Our aim is to integrate Space Hop into the primary school education curriculum. Space Hop has the potential to transform how we educate our children about the solar system they live in. Computers are becoming the new platform of learning in our schools. Textbooks are being replaced by software and this is the niche we wish to manipulate. Take them out of the textbooks and place them at the centre of our solar system. Give them the power to explore space at their own leisure.

1.3 Glossary of terms

UNITY 3D:
Unity is a development platform for creating 3D and 2D games.

AutoDesk:
AutoDesk is a 3D design and engineering software.

JavaScript:
JavaScript is a high-level interpreted programming language.

2. General Description

Below is a preliminary list of functions, open to any additional features we feel worth adding at a later date.

2.1 User Characteristics and Objectives

The target audience will be young children, in particular primary school children. Although the game is open to all age brackets, we feel primary school children would have the most to gain from Space Hop. Therefore, our user interface will be easy to implement as we are anticipating little to no computer knowledge from our users.

The main objective of Space Hop will first and foremost be as an educational platform. Practical work, in particular for children, has been shown by studies to be a more effective way of learning and retaining information. Being able to traverse through space and get a clearer image of what our solar system looks like will make it easier for the student to develop an interest in learning about space. In that sense, in order to make Space Hop as fun and interactive as possible for the student, we will focus on making the controls as easy as possible to use so that anybody can easily use and interact with the system.

Our secondary objective will be to simulate, as accurately as possible, real space physics. Due to complexity, our world to an extent will be static. However, where applicable we will incorporate real world space physics that affect controls, vision and sound produced by the game when the user interacts with our space environment.

2.2 Product/System Functions

Because we have chosen school children as our demographic, we will avoid the use of a username/password. Simply, the user will start up the program and be presented with a menu that will incorporate simple graphics; displaying a Start Game button and also a Controls button that will inform the user of the in-game controls. After starting the game, our user will be presented with an interactive, 2D representation of our solar system where he/she will be able to pick a coordinate to traverse through space. Our user will have the option of 3 spaceships.

Upon entering our space explorer, the user may approach planets and/or representations of space missions/satellites and interact with the on-screen information presented by these objects. At any given point our user will be able to return to the 2D window where he/she can choose another point in space to teleport to if the on-screen journey is too far to appease the patience of our user. From this view the user will also be able to control the flow of time, granting them control the rotational speed of the planets around the sun. This view will offer the user a mental mapping of how each of that planets interact with each other and each of the planet's place in our solar system.

The 3D view will be of a static solar system. Users will be able to fly to whatever planet they are interested in. An information panel will appear, giving the information about the planet. This could be the atmospheric temperature, gravitational pull and size comparable to Earth just to name a few. Going to close to the planet will cause the spaceship to experience turbulence. Our user will be warned about the impending doom that lurks around the corner if they don't pull back, and more importantly, why they are in trouble. This doom is simply a bluff whose purpose is to teach the user more about gravity than drag them to their untimely demise.

We feel that it is better that we don't supply the user with a username. The state of the game under any circumstance will not be saved. Since the user isn't making any trackable progress and due to the easily navigational nature of the game, we feel it unnecessary to introduce a database to hold their information.

2.3 Operational Scenarios

(1) Kevin boots up the game, excited by the prospect of exploring mars. He has only dreamed about this after playing the latest Doom game and being promptly terrified by it. However, he is in a classroom surrounded by his peers so he feels uncannily brave and reassured by the prospect of rectifying his past apprehension. His mother had informed him that there were no aliens on mars, but he doesn't believe her.

Reaching the startup menu, he fingers at the keyboard, taking in the on screen visuals and the silhouette of the words Space Hop against the backdrop of a myriad of stars and distant planets. He eventually notices the Start Game button on the screen and quadruple clicks it. A black screen appears informing him that he can press the Esc key at any time to provisionally return to the start menu.

Three spaceships appear in front of him, each as visually appetizing as the next. He eventually settles on the blue ship; his favourite colour.

A 2D image of planets revolving around a static sun is represented to him. A little confused, he clicks on the biggest planet, but not before realizing he had missed the rotating orb captioned 'Mars'. He hastily waits for the game to boot up, anxious but also a little relieved that he now may never be able to visit mars.

Wonder and enthusiasm quickly overshadow angst as the screen flashes and he is visually propelled amongst the stars. He moves the mouse around, taking in the looming image of a large planet with a strange looking disc of dust dancing around its belly. He pokes at the keyboard, suddenly realising he has no idea what is going on.

John pokes him in the back, a little too hard, and cries into his ear that there was a controls navigator on the start menu. Pressing the Esc key, he returns to the menu and the Controls tab is there as promised. At this point his teacher, Mr. Daly approaches and helps him deduce the keys he needs to press in order to interact with the space environment. He hovers his fingers over the arrow keys and returns to the 2D menu, this time valiantly clicking the mars icon.

Almost instantly, he is catapulted a seemingly safe distance from the looming red planet. It is tinted a warm, red-ish hou, and he is instantly captivated by the graphics. There is a faint shadow hiding from the glow of the distant sun on the periphery of its surface , and two small orbs locate themselves in close proximity to the planet’s surface. Hovering the mouse over the small orbs, he is informed by the screen that these are infact moons named Phobos and Deimos.

Mr. Daly magnanimously butts in to pronounce these peculiar names, but not before the screen informs him that Phobos means ‘fear’ and Deimos ‘panic’, sending a resounding shudder throughout the class.

Fiddling with the controls, he begins to traverse his way towards the red planet. The imaginary engine bravely shudders and the screen gently shakes. He is intrigued by the distant allure of the stars, his trance almost convincing him that after mars he will be able to explore distant galaxies and hopefully encounter extraterrestrials. Seemingly in chorus with his thoughts, a classmate beckons the question to which Mr. Daly gently laughs it off and informs everyone that only the Milky Way is explorable. A collective sigh sweeps the room.

Gentle typings of text begin to float among the stars on the bottom of the screen as he approaches the planet. The information informs him that mars is the fourth planet from the sun, is named after the Roman God of War, and that its atmosphere is mainly formed of carbon dioxide.

The beating in Kevin’s chest intensifies as he nears the planet. It’s surface covers most of the screen now. He fears that the planet will burst out of the screen and engulf him, but he bravely continues on in light of his audience; the angst of social alienation glueing his finger to the UP arrow key. The screen starts to strangely tint redder, and the screen shakes as his spacecraft speeds up and begins to be sucked into the planet. He turns the ship away, clinging to the distant stars for help, but they only gleam and shake back at him as his blue ships pulls itself away from the planet. A window appears at the bottom of the screen. It’s his ship’s commander, informing him that he CANNOT under any circumstance retain close proximity to a planet for fear of being sucked in by it’s gravity.

Panting, sweaty hands, perspiration attaching polyester to skin like man clings to his spacesuit for fear of suffocation, he peels himself from the hot-seat and lets his impatient classmate take over control of the craft. His space exploration for the day had thankfully come to an end, yet the strange allure of the unknown pervaded his mind for the rest of the week as he quietly anticipated his next space adventure.

(2) The sound of heels rises like a crescendo. The door springs open. The kids, second class now, race back to their seats as if it was Friday and they are playing musical chairs during PE. Today, announces Ms. Daly, they are going to explore Saturn!

One computer each, they follow Ms. Daly's instructions and log on to Space Hop. The first challenge is to locate Saturn on the 2D space environment provided on screen. If they had been paying attention in the previous lesson, they would know that Saturn is the 6th planet from the sun and the second largest in the solar system. After a bit of difficulty and some help from peers, everyone eventually locates the ringed planet.

The screen flashes and all of a sudden the class are propelled amongst the stars. The first challenge, provided as Question 1 on their lab sheet, is to count the moons of saturn and write down the names of each one. Once complete, second on the agenda is to write down as many interesting facts as possible about Saturn that they encounter whilst on their space travels. After this, they are free to explore the solar system and get ahead on the next lesson!

After class, Peter announces to his peers that he will one day be an astronaut, as does Conor. They fight about who will be the better spaceman. Space Hop had begun to revolutionise school children's interest in space, and in 2030 Peter becomes the first man on Saturn, accrediting his early interest in space travel to the much coveted space game. He eventually donates a million dollars to Thomas Feeney and Darragh Connaughton, who dutifully accept.

2.4 Constraints

Below is a list of possible constraints that we may encounter while working on our project

Time Constraints

Generally speaking, our project is quite large. It is somewhat fortuitous that our main audience are school children as this provides us with scope to make the functionality of the 'gaming' aspect as minimally viable as is possible. This helps our time constraints. Less fortunate, however, is the complexity of having to accommodate for real space physics, and supplying the user with educational information, however minimal, may push what is possible for us to implement as we arrange the construction of the project around our march deadline.

Informational constraints

As we are labelling this as an educational space exploration game, a huge constraint is on how much educational information we can provide to the end user. The most important aspect of the project is creating and implementing a viable space explorer, and thus most information we can provide from there simply becomes a bonus if not an eventual pipe dream. Therefore, we plan to make the information minimal but still interesting. As we are dealing with school children, we want to provide 'fun' facts instead of boring information that the end user won't find interesting.

Space physics

We plan to make as accurate space physics engine as possible, however this may prove more difficult than first anticipated. The scale alone of our solar system calls for us to alter reality. Travelling between planets may take a tediously long time. Even if we grant our user the ability to travel at the speed of light, it would still take 4 hours for our user to reach Pluto, never mind a return journey. We plan to loosely mimic the gravity assist of planets.This is another fanciful idea, as truly implementing gravity assist would more often than not cause our young astronauts to plummet to their demise. The sun, however distant from the user whilst in-game, will be nondescript and bright; an enlightening source of energy but a treacherous journey to embark on for the user as the screen will fill with heat and uninviting noises if he/she attempts to approach the glowing conductor of our solar system; the planets orbiting like vibrating instruments in symphony to its paralysing glow. Scaling this hot ball of plasma is another impossibility in our game. Our aim is to model our universe as accurately as possible, within our capabilities.

Artistic Constraints

Of course, representing our solar system will visually not be an easy feat. It requires precision, patience, close attention to detail and a lot of coffee to coordinate this project with our accompanying 6 modules and looming january exams. We will need to push our artistic skills to the precipice in order to coordinate genius (ah hem) with UNITY and AutoDesk. But we do believe in ourselves.

3. Functional Requirements

3.1 Space Travel

Description

First and foremost, our user must be able to traverse through space. The system to an extent will be free roam, with the option of teleporting to a different location in space through use of the 2D interface supplied. Space Hop will be operated in 3rd person mode, with the mouse used to change direction and the arrow keys or WASD keys used to move forward and sideways. W/UP will be used to move forward, A/LEFT to move sideways, D/RIGHT to move right, and S/DOWN to brake/slow down.

Criticality

This functional requirement is critical to the success of Space Hop as it encompasses the whole idea of the basic functionality of the game. Alternatively, our user could have selected somewhere in space and let the screen show them around. Using a free roam explorer, however, makes it a lot more interesting for the person using the game, and also more challenging for us to implement.

Technical Issues

The main technical issue here is how our spaceship is going to interact with the environment. As we are implementing space physics, we are going to have to simulate the space ship and screen shaking, brightness increasing/decreasing etc. Put more simply, our space travel functionality is going to have to interact with the environment. Therefore, our main technical issue will be getting the spaceship to adequately interact with the environment in order to fulfill our space physics needs.

Scaling will be a serious challenge. We want to encapsulate the gargantuan size of our neighbouring planets without having to spend hours travelling around them. We need to be able to balance usability with reality to find a medium; a game that gives you our solar system in all its enchantment while remaining a fluid and exciting experience for our user.

Another technical issue is making the space ship as easy as possible to control for the user. The problem here will be coordinating the mouse with the arrow keys/WASD. The easier and funner it is for our space explorer to traverse space, the more our user will want to play Space Hop. Other small technicalities like implementing boost power and how our space ship will interact with gravity will also be other related issues.

Dependencies with other requirements

As aforementioned, our space traveller will be interacting with the environment so therefore our free-roaming space travel requirements will be dependent on our user interacting with his/her environment. Things like crashing into objects, temperature, gravity etc will need to interact with the coordinates of our spaceship in our solar system map in order to fulfill both requirements for our space physics and also for our space travel.

We will need to scale down the gravity of some of our celestial objects. Controlling your spaceship will become futile if your find yourself too close to Jupiter. We want our features to engage in a collegial, harmonious relationship rather than a destructive one.

3.2 Space Physics

Description

As our space explorer traverses through space he/she will be able to interact with the environment. This will make it a lot more interesting for our user, as traversing through a static 3D implementation of space would be a little tedious and blatantly devoid of creativity. Gravity issues, temperatures close/far from the sun, the rocking of the ship as it travels through dust clouds and asteroid belts, the shaking of the screen and the burning up of the ship if it leaves our solar system are all potentially interesting features that we plan to add. And if, by chance, our young protagonist Kevin happens to get sucked into Mars on his future space missions, we could add in additional features like the screen shaking intensely and then turning to a static red when the spaceship gets sucked into the planet’s gravity field.

We wish to implement a free roam, zero gravity environment. The only gravity in our game will come from our planets. The gravity will be acting towards the centre of the planet, and may even cause our spaceship to orbit if our adventurer approaches it correctly. We want the game to be as hazardless as possible for our user. Toning down some of the real world dangers of space. Instead we want to use these dangers to educate our user about physics. We wish to implement the conservation of momentum also. We could implement this feature by adding rockets to the front of our ship. Whenever the user needs to slow down, he will have to balance his forward moving momentum with a backwards force. We could display the net force on screen, negative indicating backwards and positive indicating forwards.

Criticality

Although not as functionally important as our Space Travel functionality, incorporating space physics will be the more difficult aspect of our project. The Space physics aspect of our project will be the most important in terms of criticality as it will potentially make or break our project. If we can implement well designed, interactive space physics then playing Space Hop will be a highly enjoyable experience for our user.

Technical Issues

Creating a gravitational pull that is sufficient for gravity assist, while at the same time not creating a hazardous for our user may prove to be difficult. Our gravity might not even have a noticeable effect on our ship, only time will tell if this feat is possible. Creating a forward facing rocket engine might prove a lot more difficult than expected. If we position the engine in the wrong place, it may cause our user to spin out of control.

Dependencies with other requirements

The physics of the game is essential for creating the full educational, space exploration experience. If our physics is off then our Space Travel will suffer greatly. Our physics needs to be tame enough to allow Space Travel to be viable, but sufficiently assertive so it is noticeable. Our physics brings our game to life, it paints the finer details on our solar system and will truly make our game stand out.

3.3 Start/Pause menu

Description

We will implement a basic start and pause menu. When the user boots up the system, he/she will be presented with a window displaying the words Space Hop along with a Start Game option and also a Control option to view the game controls. The backdrop will be an image of space displaying stars and distant planets. Before entering the game the user will be presented with a 2D dynamic image of the planets rotating around the sun in our solar system. This window will also duplicate as our pause menu. After choosing a coordinate to begin exploring, our user can pause the game at any time and return to the 2D image of the solar system, where he/she will be able to view their current position on the map and also be able to transport to a different location at will. The Controls option located in the Start Menu will also be displayed in the pause menu. We plan to make the implementation of these windows as simple as possible given our demographic.

Criticality

Having a start/pause menu is critical to our system as the 2D image of our solar system is an important functionality of our space explorer. Without it, the user would have to traverse across space without the option of where to begin travelling. Although our version will be scaled down and the user will be travelling at unrealistic speeds whilst exploring our 3D world, it would take too long for our traveller to traverse across space without the implementation of our 2D pause menu.

For us, a start menu is not a critical feature. However, we have decided it will be a nice feature to have to display our Controls menu and also gives us the chance to display the name of our program, Space Hop, across the screen.

Technical Issues

One issue could be having an exit program button in our Start/Pause menu that will safely shut down the program.
Another issue is implementing a 2D solar system in motion that is also fully interactive. The user will have the option of quick travelling by simply clicking on a planet. In order for this to be possible, each planet will have to be able to respond from a mouse click and also be mapped to a location within our 3D model.

Dependencies with other requirements

As mentioned before, our pause menu is an essential feature of our game. It will help greatly in Space Travel, reducing the amount of time it takes to get from location A to location B. It will also act as a visual guide to aid our user if they find themselves lost along their travels.

4. System Architecture

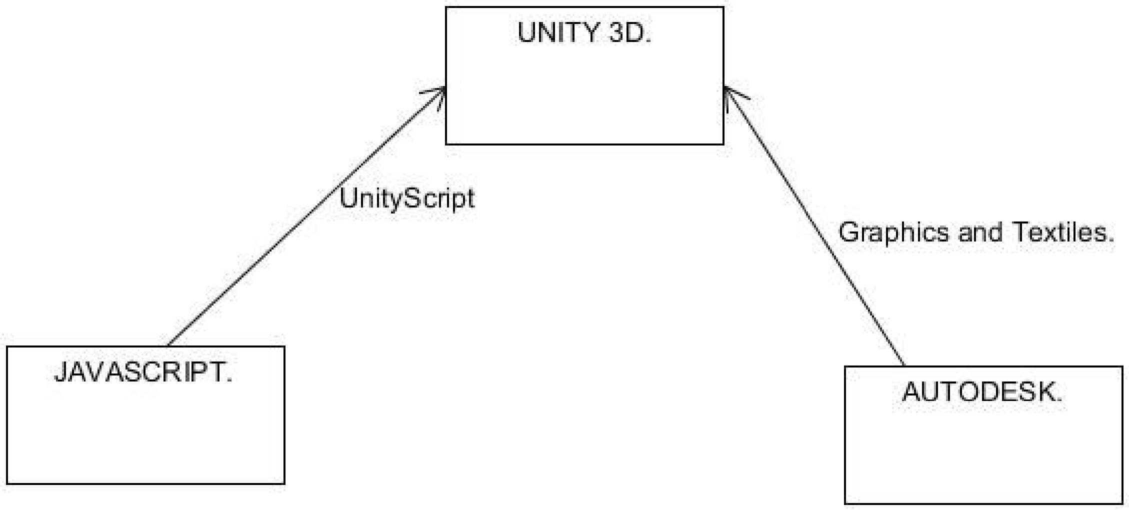

Fig 4.1. **Shows the architecture of our system. Unity will be the higher level. We have three distinct areas.**

Unity

Unity is our game development platform. From here we will be able to modify our game physics, build interactive and 3D objects. It also acts as a platform for our users to run the game off. Unity is the highest level of our project and will inherit its features and usability from our other two areas.

JavaScript

Also known as UnityScript, is the language that Unity runs off. We will be using JavaScript to define the attributes of our objects, the relationships between our objects and their environment and really any other functionality we wish to install into the game.

AutoDesk

AutoDesk is a 3D design software. We will use AutoDesk to bring our celestial planets to life and in full detail. AutoDesk allows us to transform these mundane objects into lifelike replicas of their heavenly siblings in full colour. It allows us to customise our spacecraft and our astronauts to stunning detail.

5. Preliminary Schedule

We aim to have a successful 2D model of our solar system, with all planets orbiting the sun based on their relative orbit path. This screen will double as our start and pause menu. After this has been complete we will move onto the graphical design of each planet. We wish to have this much complete before our January exams.

This leaves us with the plotting of our 3D version of the game and the game mechanics to be completed between February and March. This is just a preliminary schedule,but we may finish sooner, or later.

6. High-Level Design

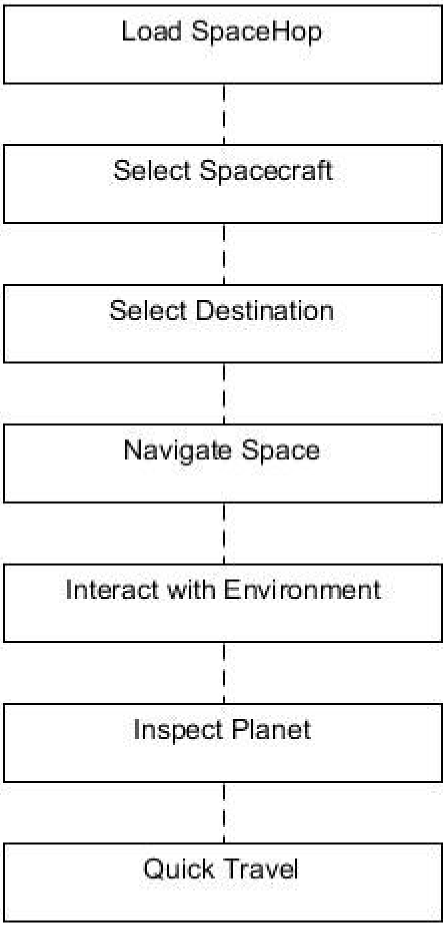

Fig 6.1.

High-Level Design Description. Fig 6.1.

- **Load SpaceHop:** User loads game from icon on Desktop.
- **Select Spacecraft:** User will have a choice of spacecraft they wish to use to explore our game environment.
- **Select Destination:** Our user will be able to select their starting destination in the game. They may choose any of planet as their primary location, once chosen they will be catapulted beside the celestial being.
- **Navigate Space:** Once entering the Space Hop environment, our user will be able to freely navigate his/her chosen area.
- **Interact with Environment:** User may then be able to interact with any objects they see in space.
- **Inspect Planet:** Planets come with the option to toggle for information, showing the stats of each of the planets in our solar system.
- **Quick Travel:** At any moment our user may travel to whichever location they wise at the click of a button, this will allow our game to become more fluid and keep the user engaged.

7. Appendices

Research Tools:

- http://unity3d.com/5?gclid=CLrStoXtkMkCFWGc2wodp5gClw
- http://www.autodesk.com/