**Project requirements – Slyggdrasil**

**Contents**

* Analysing the influencing factors on the project
* Requirements analysis
* Chosen approach to development
* Development resources
* References

**Analysis influencing factors on the project**

When approaching a large project like Slyggdrasil, there are bound to be a fair share of risks, constraints and problems with the development of said project. In this section, I will cover the more practical and industry risks that will influence the project as well as the more personal constraints and how I will attempt to tackle them.

To be completely truthful about the matter, Slyggdrasil is a huge undertaking, not only do I need to design all 9 levels, I also need to make sure the game itself runs smoothly; this is our first risk: Memory. 9 is quite a big number when it comes to levels in a game that is meant to be completed in a short time period and so; computer memory could be at risk due to not being able to properly optimise the space we have in the amount of time given. Possible solutions to this problem could obviously be to make less levels and concentrate on optimising a smaller amount of them. Another thing we could do is keep the large level count and just make the levels small. Not only do we need to optimise these levels, we also must design the art and music for them. If I had the budget and opportunity to hire a team of artists, I would, but seeing that I will need to do it solo, I am choosing to keep the art simple as it would help to move the progress of the project along faster.

Our second problem comes in how the upgrades will be programmed to edit the player and alter gameplay. This can actually be easily tackled as the upgrades can be functions of the player character and can be called by the level depending on what upgrade is chosen.

Other risks on this technical side will be mentioned in the matrix below;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Given risk name | Brief risk description | Risk likelihood (Ranked out of 5, 1 being not likely and 5 being incredibly likely) | Risk severity (Ranked out of 5, 1 being little effect and 5 being game-breaking) | Possible solution(s) |
| Memory issues | The amount of planned content may be too big to store and/or build. | 4 | 2 | The amount of levels and/or the content with in each level can be reduced for the end product |
| Optimisation issues | If the memory issues don’t cause severe problems, optimisation and how smoothly the game runs may suffer. | 4 | 4 | The same solution as for the memory issues can be applied |
| Upgrade implementation | Upgrades could possibly take longer to implement than expected | 1 | 4 | The number of planned upgrades could be reduced |
| Background interaction animation | When making the game’s background interact with the level, animations could not sync up with the actions in the level | 2 | 3 | Animations can be redone at later dates or in future updates. |
| Background interaction effects | The background interactions proposed may not actually be feasible implementation in the given timeframe | 2 | 2 | The amount of planned interactions could be reduced |
| Core principle reception | Due to how the game has been inspired from Norse mythology, the target audience may not understand where it came from, hence they may not find the heap of levels enjoyable without context for why there are that many. | 1 | 4 | In the beginning of the game, context for the number of levels can be given, filling the user in on important context. |
| Dual interaction reception | The interactions between players may become tedious when trying to progress | 1 | 3 | Updates that reduce the players’ effects on one another can be dimmed down. |
| Data loss | During development, the overall project file and/or minor class files could be lost or corrupted | 2 | 5 | Saving the project in a multitude of places and keeping project backups should lessen the possibility of this happening, but it is not guaranteed. |
| Personal illness | I may become physically or mentally ill, halting or slowing production | 2 | 5 | Making sure I keep a standard of cleanliness as well as being mindful not to overwork myself, this should not happen. |

Moving onto my personal problems and restraints when tackling the production of this game, there are some, but not too many risks.

Considering my autism and ADD (Attention deficit Disorder), I cannot work efficiently or really at all when at home, meaning I may need to adapt to working at the college on days when classes are not planned as to get ahead and catch up on dire parts of the project. On the brighter side, a working room is being built at my home, so this condition of not being able to work at home may change. When at college I mostly have the scheduled graded unit classes to work on this project as well as any free time after classes have ended; this totals up to around 5 hours per week spread over 3 days, if I were to come in on Monday and Friday, this would add up to around a much more feasible 21 hours over 5 days per week to work.

One other thing to consider is my other classes. As a college student, I have an additional 5 classes to study for and hand in work for. This will bite down on my time and may end up hindering the end result.

The last thing to understand is my current lack of extended experience. This is my third Unity game I’ve sat down and designed, so I don’t have too much to go off of other than two other games of completely different genres. To help me along with this, I will make sure to look at tutorials from YouTube programmers such as Code Monkey and Brackeys.

**Requirements analysis**

Looking at requirements, they can be split into two types; Functional and Non-Functional. Functional requirements are requirements relating to the different functions of the game, things like the requirement of having movement controls to move the player character. Non-Functional requirements are system requirements such as the platform being played on.

Below is a table showing the functional requirements;

|  |  |
| --- | --- |
| Requirement Name | Requirement bio |
| Player 1 movement | WASD to move player character |
| Player 1 menu navigation | IJKL to navigate UI menus |
| Player 2 movement | Arrow keys to move player character |
| Player 2 menu navigation | 8426 on the num pad to navigate UI menus |
| Menu start button | Press confirm on the start button to start the game |
| Menu quit button | Press confirm on the quit button to stop the game |
| Upgrade selection function | Functioning script for upgrade selection |
| Upgrade bank for each level | Some kind of storage for each level’s given upgrades |
| Upgrade application for each upgrade | Each upgrade should have a script that runs and applies the upgrade |
| Player collision | Collision between player objects where they can knock each other around |
| Background sent objects | Objects that supposedly come from the background and affect the players. |

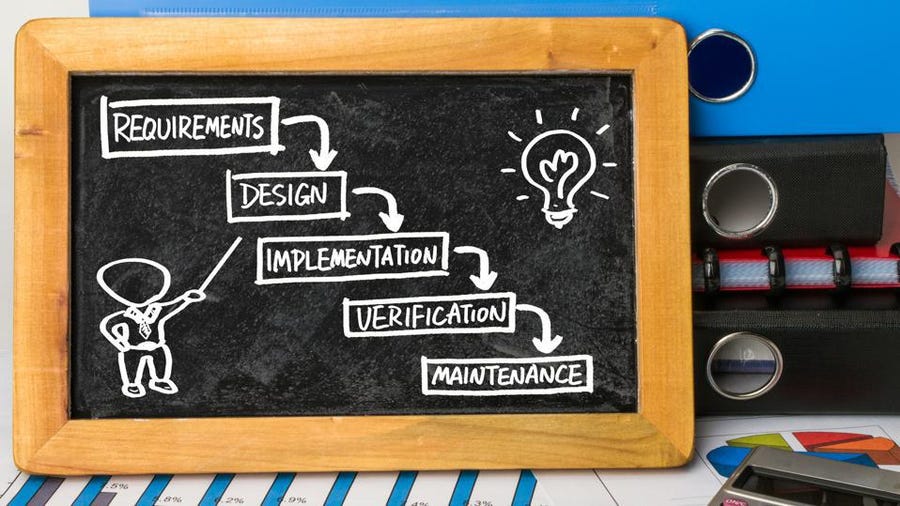
Below is a table showing the non-functional requirements;

|  |  |
| --- | --- |
| Requirement Name | Requirement bio |
| Platform | Windows PC |
| Control scheme | Using keyboard mapped to joysticks and buttons only (WASD/Arrow keys for joysticks and IJKL/Num pad arrow keys for buttons) |
| Chosen game engine | Unity (version 2021.3.8f1) |
| Audio hardware | Default PC speakers |
| Tutorials/Guides | Brackeys: <https://www.youtube.com/@Brackeys> Code Monkey: <https://www.youtube.com/@CodeMonkeyUnity> |
| Target audience | Must be suitable for 12 to 18 year olds |
| Setting | Must be appropriate for an arcade (Have score, short play sessions, action gameplay) |
| Gameplay flow | Must continuously loop (Return to title screen after win/lose) |
| 3D asset must | Must have 1 or more 3D aspects (model/introduction) |
| Audio must | Must include an audio element |
| Copyright | All non-original work must comply with copyright laws |

**Chosen approach to development**

Video game development can be done with different approaches just like other development areas of work. Two examples of these approaches are the waterfall methodology and the agile methodology. Both of these approaches go through similar steps to achieve the same goal but have very different ways of navigating these steps and are better suited to different types of projects.

Exploring the waterfall methodology; It is a fixed timeline with a fixed budget and has specific goals/milestones along their journey to the end goal. Usually, the waterfall method has little interaction with the client across development and often starts with an in-depth interview of sorts to collect the most information possible. This method is best used for a project with a rigid schedule and a set due date. (Hoory, 2022).

 (Hoory, 2022)

On the other hand, the agile methodology is an iterative methodology working in sprints for teams to work on at the same time while adjusting their strategies for more sprints. It is a method that continuously improves and builds on the project even past what was planned. It is best for dynamic projects that have lots of room for change. (Laoyan, 2022).

 (Laoyan, 2022)

Considering both of these styles with the timeline and restrictions I have, I will choose the waterfall method. The reason I’m choosing this method is because of my already established timeline and the in-depth knowledge I will have along the development of the project, making it easy to not have to consult a client about the project.

**Development resources**

|  |  |
| --- | --- |
| Resource name | Resource type |
| Microsoft Windows 10 computer | Hardware OS |
| Unity Hub 3.2.0 | Software |
| Unity 2021.3.8f1 | Software |
| Audacity 3.1.3 | Software |
| Microsoft word version 1808 | Software |
| Microsoft Project version 1808 | Software |
| Github Desktop 3.2.1 | Software |
| Google Chrome | Software |
| Studio quality microphone | Hardware |
| Audio output Speakers | Hardware |
| Me | Personnel |
| Youtube Tutorials (Brackeys) | Informational resource |
| YouTube Tutorials (Code Monkey) | Informational resource |
| Fife college stenton campus room S3.2 | Other |

**Citations**

Hoory, L. (2022) *What is waterfall methodology? here's how it can help your project management strategy*, *Forbes*. Forbes Magazine. Available at: https://www.forbes.com/advisor/business/what-is-waterfall-methodology/ (Accessed: March 8, 2023).

Laoyan, S. (2022) *What is agile methodology? (a beginner's guide) [2023] • asana*, *Asana*. Asana. Available at: https://asana.com/resources/agile-methodology (Accessed: March 8, 2023).