Slyggdrasil – From the roots to the leaves

**Contents**

* Introduction
* Assignment outline and meeting original requirements
* Strengths and weaknesses of the final product
* Project modifications through development
* Development hindsight and newly found improvement potential
* Recommendations for future development
* Self-evaluation and performance

**Introduction**

Just like a tree, Slyggdrasil has been a growing and changing project, first digging into the notion of 3 main selling points; In-depth story exploration, subversion of story tropes and low pay to play. Other important ideas and concepts like the Norse origins, separation between realms, art style and dual player interactivity that the written GGD mentions would end up as the main motivation for this world’s tree to grow into a final project that is yet to release it’s first breath.

In this evaluation, I will be discussing the origins, process, end product and future goals and how this all compares as a full self-evaluation of my performance on the project and how the future of this game looks. To help me with this, I will be looking over the original brief, GDD, project plan, project requirements, testing plan, test cases and user testing sheet in order to understand what exactly was planned for the project, how strong or weak the end project is, modifications that the project went over throughout development, how I could have done better with the skills I have now and how well I performed.

**Assignment outline and meeting original requirements**

The assignment was clearly outlined in the Project brief and I took the key details into my project requirements document. The details that I took were as follows; Limited control scheme for an arcade cabinet, arcade setting, target audience of 12-18 years, continuous gameplay flow, 3D asset inclusion, audio asset inclusion, object-oriented approach to programming and adherence to copyright law.

Going over how I presented this outline in my project requirements document, I used the given control mapping as my control scheme, the arcade setting as my pixel-art retro style, the target audience as it is, a continuous gameplay flow of looping through the end screens to the main menu, 3D asset inclusion as a simple passable game object, audio asset inclusion for my music, object oriented programming as my chosen OOP-focused game engine of Unity and the compliance with copyright as my permission to create my own assets without sourcing them.

Along the project, I’d like to go over how things were drawn out. First of all is the initial ideas that I presented; Battle of Realities – a fighter game like smash bros that follows the “Fantasy or Reality” formula that I’ve used throughout my time in the course, Arcade Game Game Game – a puzzle game with a tricky mechanic where the screen would keep getting smaller with each level and The Sky ISN’T The Limit – a doodle jump game where chaos would increase with each level. These ideas are ambitious as I always am but ended up being too much for me, so I took a fresh perspective with the third and safest idea. This ended up taking my idea of using a doodle jump formula and applying it to an old game that I made in high school – a game where you race up a tall level to try and catch up to your buddies. The hurdle of the GDD was my largest and most passionate part of the process. I took all elements into consideration – the art, the mechanics, the narrative, the characters, the colour palette and even the separate objects and how they function. The GDD was an entire book of 6000+ words while still remaining quite specific on the expectations for what the end product should be, little did I know that the project would stray from the original vision. The next step in the project was the part that I feel I did poorly on; the project plan. Due to how little experience I had in Microsoft project along with the standard I had to meet with the GDD, the result of the project plan ended up being messy when it comes to dates and even had tasks that should have been split into smaller tasks. Despite this, I moved onto implementing my plans into Unity, scrapping many planned features but ending up with something that most people that played it really enjoyed. The testing was simple enough in the next step, I spent some time marking off test cases that I would carry out before completing the bland process of poking at the code and making sure everything worked. The second part of testing was End-User testing; for this, I put out 10 user questionnaires – 8 of which were filled – and scanned them in digitally to store along with my technical testing. All that’s left now is to evaluate and apply my conclusions to the future development of Slyggdrasil.

In the end product, I did indeed adhere to the given requirements but my planned solutions were not exactly identical to the final piece, these changes will be discussed later.

**Strengths and weaknesses of the final product**

Jumping right to the final product, I will be going over what went well and what went poorly when comparing the product to the user testing that were filled in at the video games showcase event.

From the collected user test data, I can easily see that there are large strengths spotted through a player’s eyes. One of these strengths are a deliberately engineered cuteness of my characters as well as the overall simplicity of the art style. When asked about the characters of the game, comments like “Adorable”, “Loved them!” and “Superb” appear on the surveys. When asked about graphics, people wrote things like “Wholesome”, “Simple” and “Clear to see where to jump”. I am overjoyed to see that my art and graphics got across a cute and simple message, it is also good that the simplicity of the style assisted the players in guiding their way.

When considering strengths outside of user testing, I personally understand the strength of dual player interactivity. With the players being able to interact and push one another, it makes for some interesting battles against one another even if the game itself isn’t competitive.

Where there are strengths, there are also weaknesses. These weaknesses actually very much effect the expectations set up for story. The fact that I didn’t have enough time to implement a story into the game shows up a lot more than other features since a question relating to story was put onto the questionnaires before implementation began. When asked about story, comments like “?”, “Didn’t really feel like there was a story” and “Didn’t see story” would pop up. Another weakness I could find in user testing is a lack of Norse theming in the final product, the only real Norse theme in the game is the inclusion of the world tree that the game is named after. The question about Norse themes left lots of players guessing what the actual themes were other than Yggdrasil, this is another weakness that goes against the original expectations of heavy Norse theming.

Aside from user testing, I myself noticed two key weaknesses relating to the requirements.

The requirement for a 3D aspect in the game came down to having one very unnoticeable 3D platform that appeared as a 2D aspect anyway, taking away from the idea that I’m any good at 3D modelling.

The other weakness comes from the bare use of audio in the game. The game features a single looping music track that plays at all times, there was nothing else. Personally, I loved creating the music for the game, so it disappoints me slightly that I had no time to include any other sound elements.

**Project modifications through development**

With a huge project like Slyggdrasil, there are bound to be some reworks and changes that come along with it and this one had some huge modifications. From changing the features that the game focuses on to scrapping whole levels, my end game is an entire evolution away from the expected outcome.

The first modification I’d like to talk about is the change from a group of story-based selling points to a group of gameplay-based selling points. When introduced in the GDD, Slyggdrasil’s unique selling points came to In-depth story exploration, subversion of story tropes and low pay to play. At the end, the low pay to play ended up being irrelevant due to the fact that the game will not actually be played in an arcade. The story aspects being changed was not a conscious choice made by me but a developing change that happened as a result of time restrictions. Now, Slyggdrasil’s unique selling points come down to the entertaining player-on-player interactions, the simplistic art style and the replay ability.

The second modification is the removal of levels. In the planning stage, there were 9 proposed levels, 1 for each realm that Slyggdrasil connects to. Again, due to time restrictions, only 5 of these levels ended up in the final project.

The third modification is the removal of game mechanics. Originally there were supposed to be 7 unique mechanics. This was shortened to the 2 most vital ones; Player interaction and Platform diversity. As with all of the modifications thus far, this was due to time restriction.

Minor changes like obstacles being removed along with their designated levels, different kinds of music for each level, animations and custom fonts were also removed because of time.

**Development hindsight and newly found improvement potential**

Throughout my journey of developing Slyggdrasil, I have learned many things about sources for code, prioritizing steps in development and even more.

First of all, a large problem I had throughout the project was getting the character controller to work in the right situations. I now know that I should have just taken a simpler controller and made it largely barebones until real and vital decisions have been made. This may have allowed for extra features that were scrapped to be actually implemented on time.

Another mistake I fell for throughout development was creating art assets before making sure I had a functional project. This mistake did not harm the project but I had gotten lucky with my lecturer reminding me that a pretty game is only good if it actually works.

The last improvement is less to do with development and more to do with my priorities. I focused on this project far more than I should have and I could have definitely spent less time fiddling with code and more time on completing other tasks that were vital at the time.

**Recommendations for future development**

I plan to develop Slyggdrasil further and have recommendations on how this should be done. First of all, I should check back more regularly with planning documents and schedules – they are there to help and guide me throughout development. I also should make sure my work environment is separate from my home environment – this is something I haven’t done enough of and need to commit myself more on. The last recommendation is to keep a hold of my user testing data – the data shows what players want and that is the goal I should strive for.

**Self-evaluation and performance**

After all of this reflection and analysis of the project, I have good reason to say that this is a huge leap from where I started. To think that I could go from making a barely passable game last year to creating such a largely enjoyed game among my lecturers and test users this year makes me incredibly proud of myself and has me looking forward to the future. My discipline of myself during planning, design and development is miles above what it was last year and my consideration of the due dates for submission is impressive for my standards. However, my prioritisation of this project above all others has me scrambling to complete things in the final week of college and could end with me failing the course. This part I am very saddened by, it has me stressing my mind out and almost keeping me up at night as I try to submit the bare minimum for my missed submissions. Overall, I believe I have made huge improvements in my game design, game planning and game development skills but hope I manage to improve on my task priorities.