Project Proposal

Team Number 26

Corey Anderson, Jake Beesley, Luke Beesley, Samuel Gilchrist, Luke Less'ard-Springett, Rory Reidy

Project Name: Walkthrough Canon

Project Synopsis

This project is a puzzle adventure game with story elements from Robinson Crusoe, Moby Dick, and 20,000 Leagues under the sea.

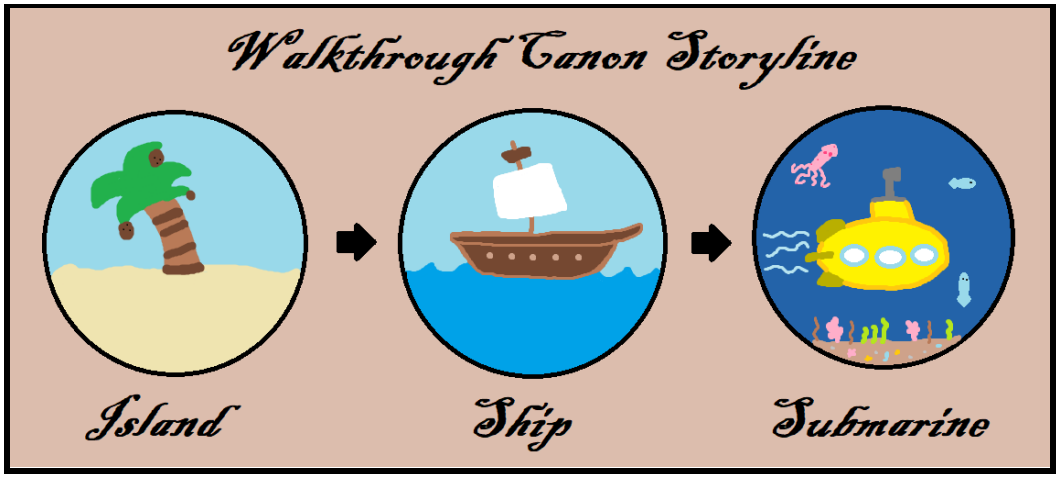


Figure 1. Overview of Storyline

Project Description

This project is being undertaken primarily because of our group's collective interest in mysteries, video games, and literature. Our group intends to address the problem of general knowledge of classic literature becoming increasingly rare in younger people today, as these source materials become older and less relevant to each generation. The goal of this project is to familiarize and teach our player audience about characters, themes, and other references of classic cannon books in a fun, engaging way that makes this knowledge more accessible and relevant to them. Our team also aims to challenge and exercise players' critical-thinking skills in various puzzles throughout a play-through of the project. The end result of the project will be a program containing a single, specially designed educational experience that will allow players to freely maneuver around a 3D world, interact with various characters, and solve puzzles in a story unique to the project.

Project Milestones

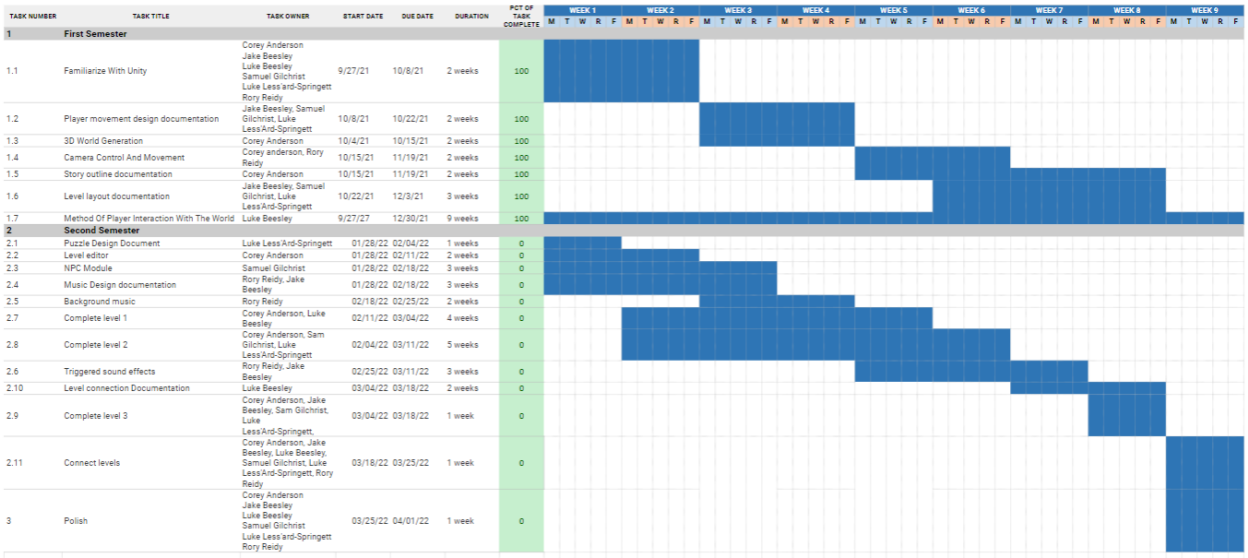
Our first-semester objectives include 3D World Generation, Camera Control and Movement, and Player/World Interaction. We have attached a Gantt Chart with estimated completion dates and delegation of work. We aim to further refine this Gantt Chart as we delve into the process.

First Semester:

1. Familiarize with Unity (10/8/21)
2. Player movement design documentation (10/22/21)
3. 3D World Generation (10/15/21)
4. Camera Control and Movement (11/19/21)
5. Story outline documentation (11/19/21)
6. Level layout documents(12/3/21)
7. Method of Player Interaction with the World (12/3/21)

Second Semester:

8. Puzzle design Document(2/04/22)
9. Level editor(2/11/22)
10. NPC Module(2/18/22)
11. Music design Documentation(2/18/22)
12. Background music(2/25/22)
13. Complete level 1(3/4/22)
14. Complete level 2(3/11/22)
15. Triggered sound effects(3/11/22)
16. level connection documentation (3/18/22)
17. Complete level 3(3/18/22)
18. Connect levels (3/25/22)
19. Polish(4/1/22)



TASK NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	**First Semester**					
1.1	Familiarize With Unity	Corey Anderson Jake Beesley Luke Beesley Samuel Gilchrist Luke Less'ard-Springett Rory Reidy	9/27/21	10/8/21	2 weeks	100
1.2	Player movement design documentation	Jake Beesley, Samuel Gilchrist, Luke Less'Ard-Springett	10/8/21	10/22/21	2 weeks	100
1.3	3D World Generation	Corey Anderson	10/4/21	10/15/21	2 weeks	100
1.4	Camera Control And Movement	Corey anderson, Rory Reidy	10/15/21	11/19/21	2 weeks	100
1.5	Story outline documentation	Corey Anderson	10/15/21	11/19/21	2 weeks	100
1.6	Level layout documentation	Jake Beesley, Samuel Gilchrist, Luke Less'Ard-Springett	10/22/21	12/3/21	3 weeks	100
1.7	Method Of Player Interaction With The World	Luke Beesley	9/27/27	12/30/21	9 weeks	100
2	**Second Semester**					
2.1	Puzzle Design Document	Luke Less'Ard-Springett	01/28/22	02/04/22	1 weeks	0
2.2	Level editor	Corey Anderson	01/28/22	02/11/22	2 weeks	0
2.3	NPC Module	Samuel Gilchrist	01/28/22	02/18/22	3 weeks	0
2.4	Music Design documentation	Rory Reidy, Jake Beesley	01/28/22	02/18/22	3 weeks	0
2.5	Background music	Rory Reidy	02/18/22	02/25/22	2 weeks	0
2.7	Complete level 1	Corey Anderson, Luke Beesley	02/11/22	03/04/22	4 weeks	0
2.8	Complete level 2	Corey Anderson, Sam Gilchrist, Luke Less'Ard-Springett	02/04/22	03/11/22	5 weeks	0
2.6	Triggered sound effects	Rory Reidy, Jake Beesley	02/25/22	03/11/22	3 weeks	0
2.10	Level connection Documentation	Luke Beesley	03/04/22	03/18/22	2 weeks	0
2.9	Complete level 3	Corey Anderson, Jake Beesley, Sam Gilchrist, Luke Less'Ard-Springett,	03/04/22	03/18/22	1 week	0
2.11	Connect levels	Corey Anderson, Jake Beesley, Luke Beesley, Samuel Gilchrist, Luke Less'Ard-Springett, Rory Reidy	03/18/22	03/25/22	1 week	0
3	Polish	Corey Anderson Jake Beesley Luke Beesley Samuel Gilchrist Luke Less'ard-Springett Rory Reidy	03/25/22	04/01/22	1 week	0

Figure 2. Gantt Chart (1/2)

Budget

This project will be built using Unity 3D, which is free for students and individuals.

Preliminary Project Design

How the Software Works

Throughout this project, a few pieces of software will be used in the process of making our game. This project will make use of the Unity engine as the framework and primary method of creating the game, along with Git to allow for the connection of each developer and ease of access to the project. For art assets, features in Unity will be used to acquire preexisting assets along with Blender being used to create custom assets made by the team. The combination of Unity, Git, and Blender allows the team to take systems designed for making games and communicate the work done on the project with one another effectively and easily.

The process of using Unity is designed around making games in more efficient manors and creative ways. The best comparison of Unity to other types of software is that Unity is a toolbox with many different tools for different types of jobs. Continuing to use the toolbox analogy most projects are simply not one big piece but an accumulation of many smaller pieces. It is through using scripts created within Unity that the team will combine much of the work done to create and flesh out the larger project as a whole. Throughout this project, our team will use Unity's prefabricated scripts to allow for the creation of mundane interactions within the game we wish to design. All the while allowing for more complicated tasks to be programmed by the team to work efficiently and effectively. This process will create a seamless flow of player interactions and decisions in the game creating the overall gameplay.

Another key aspect of Unity is the ability to procure and placement of free or pre-made assets within the game itself. This built-in marketplace of free and paid assets will allow the team to spend more time on creating the technical aspects of the game while still allowing for the opportunity to create our own assets. Similar to adding scripts within the engine assets can be placed and molded to the desire of the team and made to work in specific ways.

For this project, the intention is to tell and story, and being able to visually display what the player is able to interact with in a meaningful manner is key to the success of this project. For this project should the need arise to create custom 3D assets, the team will make use of Blender an open-source 3D computer graphics toolset used for creating 3D art assets. Using Blenders built-in tools the team will design and fabricate assets based on what is needed in the project. These assets will then be exported and imported into Unity which then will allow for manipulation within Unity.

Communication in this project is key to its success and therefore the team will make use of Git as the primary source of distributing the project to one another. Git is a relatively easy system to use allowing for the fast transfer of information to a repository, in this case, Github will serve as the storage for that repository. Using Git also comes with the added benefit of version control in the case of needing to revert back to older versions of the project.

These 3 key pieces of software will be the underlying foundation of the project allowing for the team to design, create, and combine the work into a concrete game.

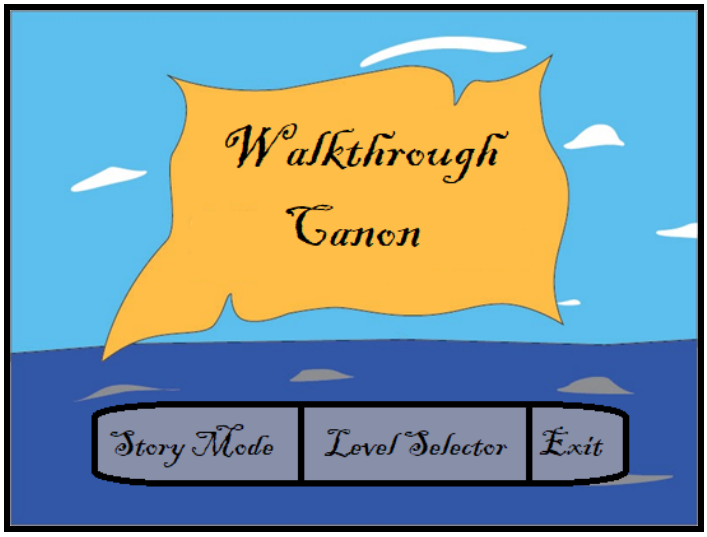


Figure 3. Conceptual Title Screen

Design Constraints

In regard to constraints, there are a few technical constraints relating to the Unity engine. First, the only native language that Unity uses is C#. While all of our team members are familiar with C++, there is a difference between the component-oriented language C# and the object-oriented C++. An effort has to be taken to familiarize ourselves with a language we don't know and be able to build a meaningful project within Unity. This factors into a time constraint, as it limits the amount of work we can accomplish. With various different schedules of our group, and different classes being simultaneously taken, it's important to stay on track of our project milestones and ensure that meaningful progress is being made within a set timeline, while ensuring we are learning the basics of C# and the Unity engine.

The actual design of the program is a major constraint, as it details what our 3D generated world will be made up of. With various open-source libraries within Unity, with many different models and interfaces to choose from, it is important to ensure faithful work within our environment and ensure that everything is implemented from a position of trust. By choosing a specific library to use pre-generated 3D models, we are now constrained by the library's contents, and also by our permission to use such libraries. Specific game functionality might not be found in an open-source library, in which case special effort would need to be taken within Unity to create a new foundation from scratch.

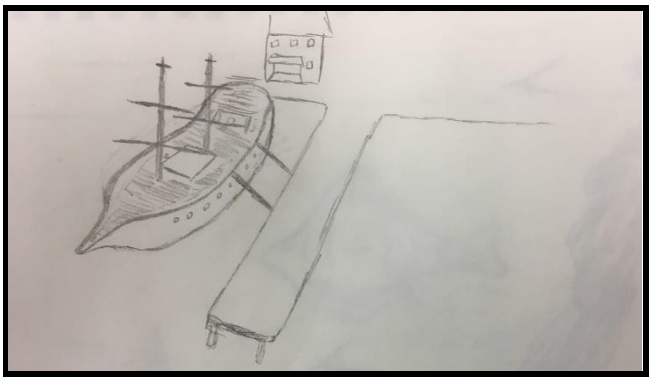


Figure 4. Conceptual Idea of the Starting Level

By wanting to create an original game with a meaningful plot, we decided to borrow some stories that have been in the public domain. This constraint limits what story we can tell (whether or not it is in the public domain), how the story can be represented by our game, and whether or not the story would provide an enjoyable user experience. This limits our selection to adventure stories, which is what our game is primarily based on. Our team's goal is to display a 3D adventure world and tell a story that has educational and historic value. By choosing three stories related to the sea (Robinson Crusoe, Moby Dick, and 20,000 Leagues Under the Sea), we are constrained by the narrative of each story, but are ensuring an enjoyable and interactive user experience using these classic novels.

Operating systems play a critical constraint in regard to this project. First, while Unity is available on Windows and Mac, it is important that whatever OS a team member is working on works appropriately with a different OS. Most of our team works within Windows, so it's important that any changes in our code are compatible with both Windows and Mac. Dependencies depending on the operating system should be addressed, which plays a role in the user's experience with the game. The synergy in team member operating systems is beneficial, as this can ensure a smooth user experience regardless of the user's operating system. By understanding any compatibility issues within our code due to operating system dependencies, any Windows or Mac user should be able to have a seamless experience within our virtual game.

Ethical Issues

Our software aims to provide an educational, virtuous opportunity to introduce new audiences to some of the relevant themes, stories, and lessons from classic literature; however, that does not mean that there are no ethical questions to consider as we develop our project. For instance, it can go without mentioning that the works of literature which we plan to cover in our project are products of older times, where social views and ethical standards for individuals were generally much lower across the board than today, and these changes in our society should be considered along with the themes from the stories.

Additionally, most works considered to be part of the “classical canon,” the body that our project pulls stories from, are Western works of literature disproportionately written from the perspectives of white males, and, therefore, these chosen works may not always reflect universal moral truths, as fair portrayals/viewpoints of minorities, women, and poorer individuals of these times are often not included within these works. To our team, these concerns are relevant to Section 1.4 of the ACM Code of Ethics and Professional Conduct, which guides professionals to be “fair and take action not to discriminate” in their work.

Intellectual Property Issues

The first issue to be addressed is the legality of using classic novels as a basis for our game’s plot. The books we are planning to use are all well-known, “classic” novels. United States Copyright Law gives separate provisions for published and unpublished works. If a work was not first published in and never registered in the US, copyrights extend 70 years past the death of the author. If a work was registered in or first published in the US, the copyright term is dependent upon the publishing date. Moby Dick was published in 1851, meaning that its copyright is expired. Robinson Crusoe was published in 1719, meaning that it is expired. Twenty Thousand Leagues Under the Seas was originally published in France and republished for an American audience in 1873. This book is also free of copyright. [1] As long as we stick to the original, public-domain copies of these novels, we will be free to use them how we wish.

One more potential issue is the use of free assets. Unity provides many assets to their users and maintains a legal distinction between Non-Restricted and Restricted assets. Non-Restricted assets are often free, and users are given a license to “integrate [them] only as incorporated and embedded components of electronic applications and digital media and distribute such electronic application and digital media.” [2]. Essentially, we can use Non-Restricted Unity Assets in our game, but we cannot reproduce and sell them. We are entitled to distribute the Assets as part of our game, but not in their raw form. We are free to alter the assets as such, as long as we don’t sell them.

In the end product, Unity Assets may not be the only ones we use. If we acquire assets from other sources, we must be sure that a license is given alongside the asset. For game art, one of the most common licenses is the Creative Commons copyright system. Licenses under Creative Commons differ; some allow for commercial use, while some allow for sharing [3]. Certain licenses require attribution to be given to the original creator-- in this case we would have to include them in the credits of our game. In any case, we need to pay heed to any licenses we acquire. This will include close documentation and examination of our assets’ licenses.

Change Log

- Updated Project Synopsis (11/1/2021)
    - Expanded details of the core game, and established emphasis of what 3D engine can be used to tell our story.
- Updated Project Milestones (11/1/2021)
    - Established concrete dates in regard to progress deadlines.
- Updated Gantt Chart (11/1/2021)
    - Reflects current progress already made and establishes dates for future milestones.
- Updated Project milestones(1/31/2022)
    - Finalized dates as well as added additional milestones breaking tasks down even further to better reflect progress.
    - Added documentation milestones.
- Updated Gantt chart(1/31/2022)
    - Updated Gantt chart to reflect completed milestones as well as additional milestones.
- Updated Project synopsis(1/31/2022)
    - Adjusted the wording of the project synopsis to be more concise

Sources

[1] Hirtle, Peter B. “Copyright Term and the Public Domain in the United States.” *Copyright Information Center*, Cornell University, 12 Mar. 2021, https://copyright.cornell.edu/publicdomain.

[2] “Asset Store Terms of Service and Eula.” *Asset Store Terms of Service and EULA*, Unity, 31 July 2020, https://unity3d.com/legal/as_terms?_ga=2.248367303.1740138341.1635777801-2107435840.1635777801.

[3] “About the Licenses.” *Creative Commons*, http://creativecommons.org/licenses/.