CSCE 361 Project Proposal

Group 13
Grahm Manley, Eric Tran, Noah Loos, Xiang Ma

Group 13 aims to develop a 2-D Platformer on the Unity Engine featuring stages and a character with abilities to complete stages using basic keyboard controls. The 2-D Platformer is geared towards users that already play video games and have an idea on what a platformer is. Knowing this, Group 13 plan on developing a twist on the platformer genre by changing how one completes stages when compared to well known platformers such as "Super Mario Bros" and "Shovel Knight."
- Provide some ideas on what functionality it will have during phases 1-4 and what functionality could be added during the agile increment in phase 5

The base functionality would include a character or a team of atleast 6 characters that have different abilities to use such as blinking forward, bouncing off things, or other properties associated with them. A key feature of this game will be giving the character that ability to "die" or "freeze" which effectively kills the character but leave their body there, whether it be on a surface or in the air. This would allow the next character coming into use their body as another platform or for example use their old character's body to do things such as bounce off of, launch, or teleport with.
On top of this, about 10 or more stages will be developed to provide a field to play in while show casing the abilities. These stages will have a beginning and end, including a limited selection of characters for each stage. Being 2D, most stages will either progress from left to right, or down and up. If possible, adding sound effects and music to each stage would be another feature during this time. In these stages, obstacles will be presented that would slow down or stop a character from progressing effectively.
Additional functionality that could be done during an agile would be a score system of some sort based on an element such as object or time. Another idea would be to implement additional maps or abilities into the game that would allow for more content. A feasible idea during an agile would be including a mulligan system that allows a player to reverse one step without having the restart the entire stage.
Features that could be added given more time would be stage maker for the player to create their own stages in the game and allowing them to choose characters for that stage. This would allow the player to create stages and play it themselves or share with other players. Another feature that would require significant time, would be a time attack game mode against another

player online where both players would race each other in an attempt to get to the end of a stage first.

- Brief Discussion of the technologies required to create the system. Ex: Language, IDE, Platform

Unity will be used as our game engine due to the fact that it is a well established engine that some of the group members have experience with. The engine has a wide feature set which will allow expansion in any direction the group may choose. Unity uses C++, but has interfacing to allow C# and JavaScript code which the group has limited experience with. The Unity provides a multitude of resources for learning and developing games, specially 2D platformers. On top of that, Unity features a space that allows Unity users to share their Assets as open source to be used for game development. One particularly key feature of the Unity Engine is that it is free to developers unless the developer makes over $100k in which 5% royalties must be paid afterwards. Knowing this as much of our team is college students, the Unity Engine has been installed by most of the group and should be free for most of the project. Lastly, Unity Engine has functionality for interfacing on GitHub for collaboration.

- Feasibility with given deadlines. What qualifications as a team do we have to build the system, the availability of the necessary technological support, and any foreseeable risks that might be an issue in the future

With given deadlines, our team has the ability and time to develop said software. Although most of us are not too familiar with JavaScript, C#, and C++;
- 2 people on the team has some experience with Unity Engine
- 2 people know how to program in C++
- 1 person has experience with JavaScript
- 1 person in game developers club

On top of this as mentioned earlier, some time can be used to start training with the Unity Engine through the various tutorials can require no previous programming experience. This time can come from the ease in defining systems for this platformer. Otherwise, issues can occur as the complexity of the game increases as testing the game may take additional time and may require an outside group to test said platformer for additional input into how the game should be shaped. Despite this we are confident that this platformer is feasible during phases 1-4 and phase 5.