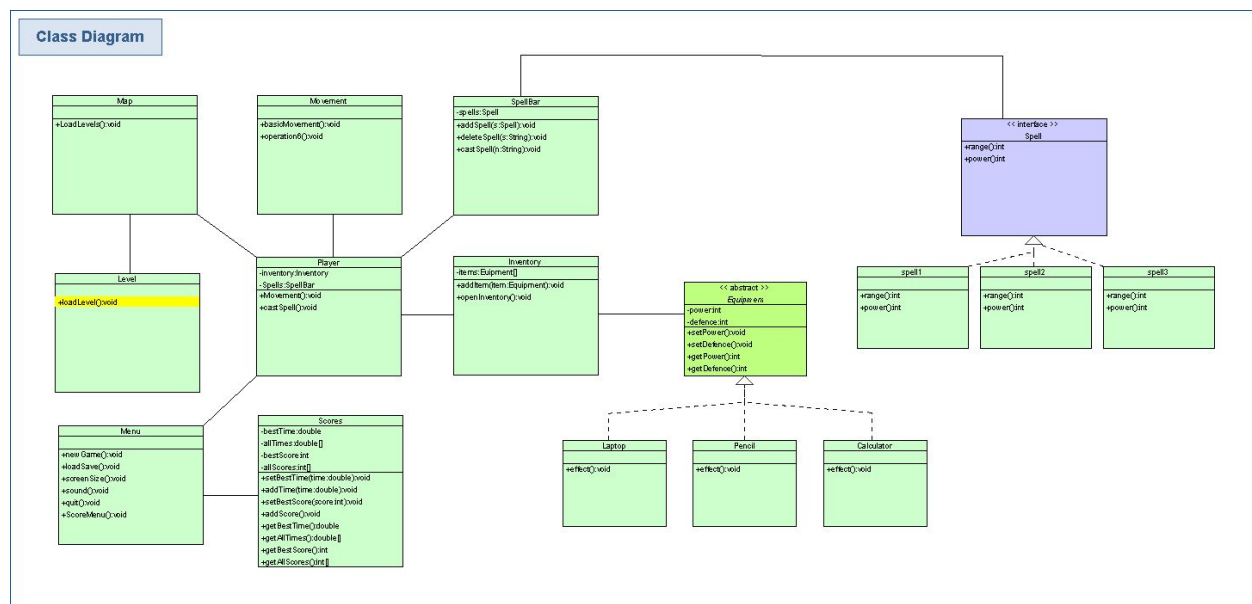
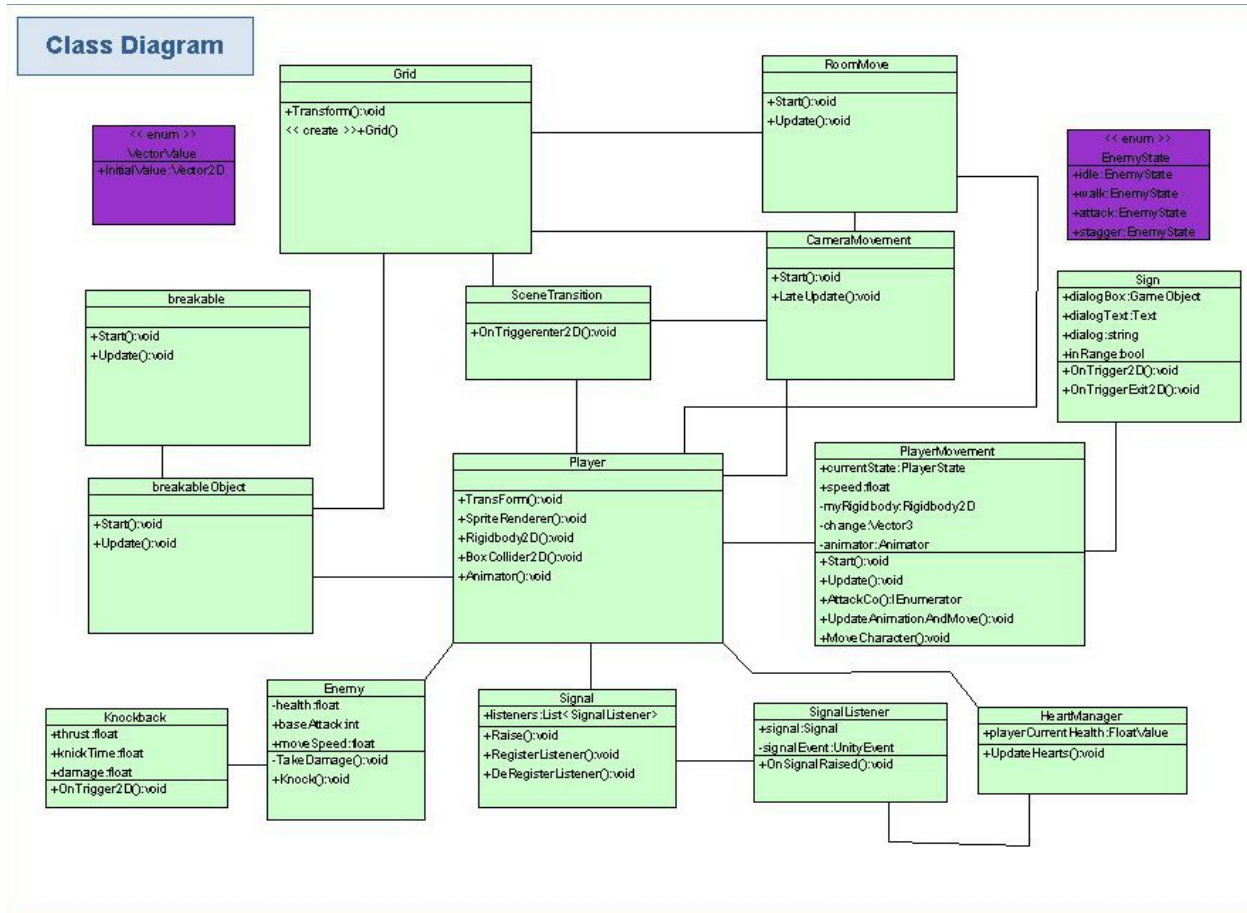


- Final State of System
 - The game starts at the home arena, in which the player starts a new day, just got out of his apartment and ready to attend classes, only to be surprised by the zombie-like students attacking everyone under Grash's influence, in hope for a curve in their grade
 - After the player sees a sign left by an unknown person telling him the current state of the university, and what he can do to save everyone. The player then heads to the Engineering center to start his mission, all while defeating the desperate students who are fighting as Grash's army, the corrupted instructor, for a curve in their grade
 - Implemented features
 - The player is able to do a basic navigation in which he can walk diagonally and straightforward. Also, the player has a health bar that tells him how much health he has left
 - Also, the player can attack, although it is pretty basic. The player swings a small swords, which he uses to defend himself from the mad students, and to defeat Grash and free everyone from Grash's control
 - Maps
 - Home
 - The apartment the player lives in
 - Campus
 - The parking lot of the engineering center
 - Engineering center
 - Inside Engineering center, there's a lounge, and lastly, there's a classroom in which the player will find Grash plans to take over the university through ancient computing techniques sealed deep within the engineering center
 - Enemies
 - The mad student
 - Low attack damage, very fast but dies quickly
 - Grash, the corrupted instructor
 - Very fast, high attack damage, and high HP
 - Features which were not implemented
 - We wanted to implement a dialogue system in which the player can talk to other students and Grash
 - Also, we wanted to implement a follower NPC that follows the player and gives him hints and tips
 - We wanted to create a save system, in which the player could save his progress, and come back to it later
 - We wanted to implement a more diversified set of enemies with different attacks
 - We wanted to put the story within the game
 - Due to the time constraints, we were not able to implement those in time, but we plan on implementing those in future whenever we have the chance.
- Diagram



- Looking at the diagram, we were unable to implement the Menu system, Magic spell, Inventory system. Due to the time constraints we were not able to implement many of the features we desired. Lastly, we planned to have objects to interact with in the game, such as a Laptop, Pen, and Calculator.
- Third-Party Code
 - Third-Party Code
 - <https://docs.unity3d.com/ScriptReference/>
 - We used Unity for our game development, and used a set of assets (images), and we created the animations from those assets.

- <https://opengameart.org/>
 - <https://www.youtube.com/channel/UCZczqDvepgNqy80gTMGnUXw>
 - We used some of the assets from this channel. Also, we learned the concepts of Unity from his tutorials.
 - Original Code
 - Animation were created by us, but the assets for those animations were from the above sources
 - Scripts Folder
 - Everything here is written by us. Basically a folder which controls many different aspects of the game such as the characters and camera movements, and how the scene is portrayed
 - Scenes Folder
 - In this folder, we created a canvas (map) ourselves, but the sprite and images for those canvases were from the above sources
- Overall statement
 - Design process elements
 - Time Constraint (negative)
 - Originally, we had a lot of goals for this game, but we had to set them aside due to time constraints. We started with the map at first, and spent too much time which led to not having enough time to implement our original goals
 - Unity Framework (positive)
 - This is our first attempt at using the unity framework, and it is extremely easy and intuitive. We both come from heavy programming background, and we're used to difficulty in adjusting to a new programming environment, but we didn't face that with Unity. It was extremely basic, and allowed us to develop what I would say a good game in not too long of a time. Also, the Unity framework allowed us to easily build executables for the game for different platforms easily
 - Great artists distributing their artwork for free (positive)
 - One of the points we feared while conceptualizing our project, was not finding art for our game, but luckily for us, the internet has a lot of open sourced assets which were perfect for our game.

Notes:

Design patterns used:

- Observer pattern (Signal.cs, SignalListener.cs)