****

**CP5310 Project – Monster Rush**

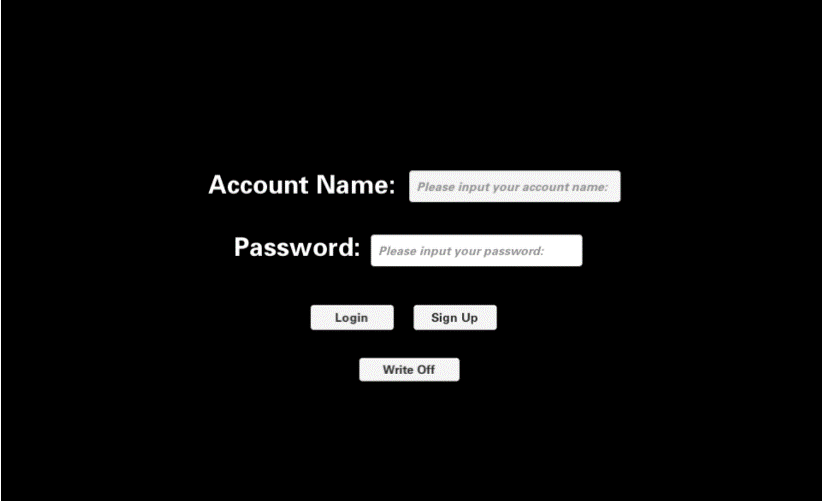
# Project objectives

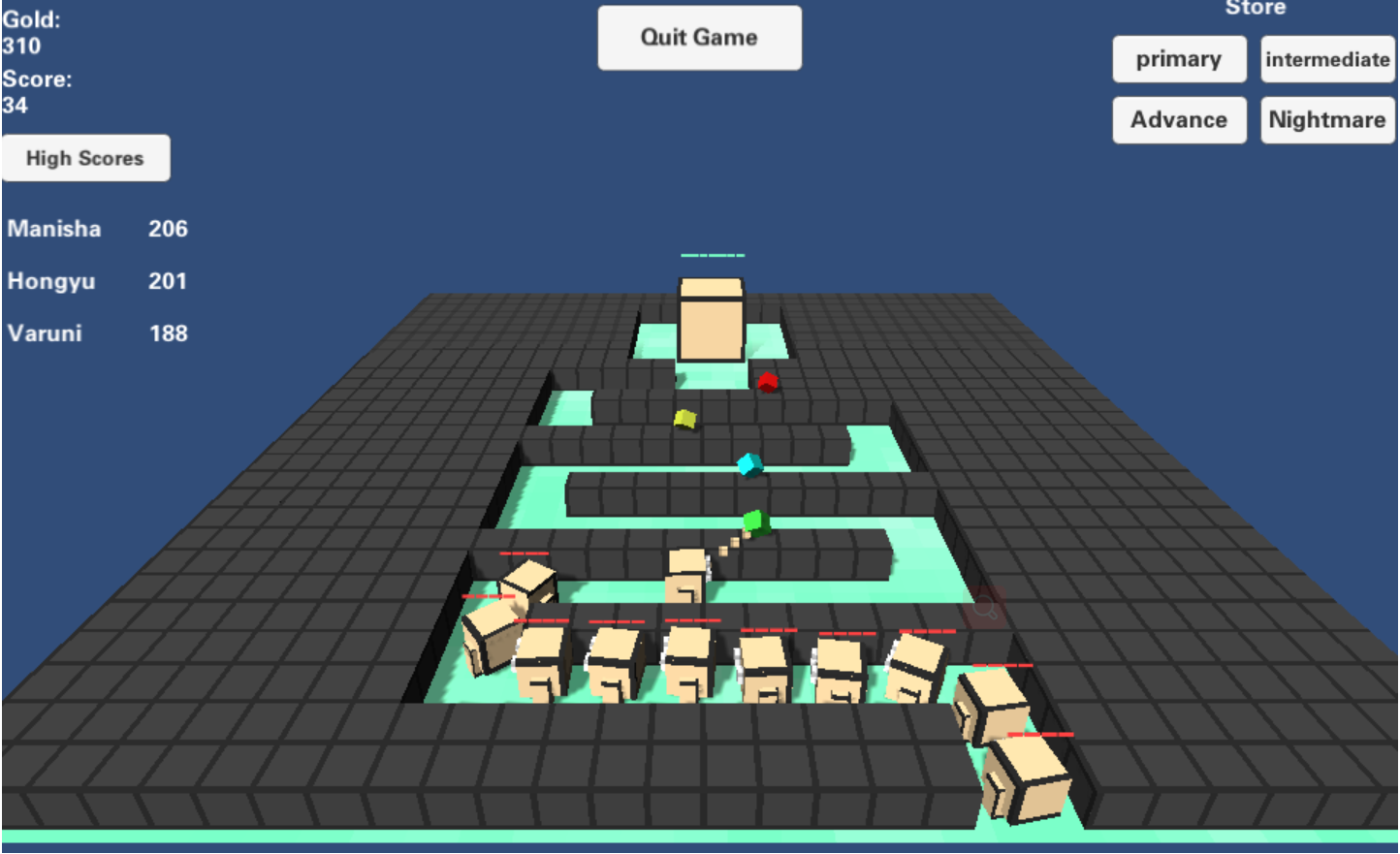
Our Objective was to build a website with a Tower Defense game. Which is fun and entreating for the players. The main objective of the game is to kill the monsters who come to attack the Castle through building new towers which can shoot bullets to attack the monsters coming in their shooting radius. As long as your Castle are safe from the monsters you won’t lose the game. We create simple 3D models (environments and characters of the game) in Unity3D which is a game engine. We also store the data of the web game which can be use in the next time the player logins and displaying to player, for example, the score and the gold amount from the last time the player quitted the game. The web game allows player to sign up and login, so they can keep their score and gold, and continue in the next play. Finally, we use cloud service to publish our website on the Internet.

Tower defense is a subgenre of strategy video game where the goal is to defend a player's territories or possessions by obstructing the enemy attackers, usually achieved by placing defensive structures on or along their path of attack.

# Features implemented

1. 3D environment and characters of the game.
2. Login system for player to sign up and login, to keep their score and gold, and continue in the next play.
3. Scoring system accumulates score each time the player eliminates a monster.
4. High scores ranking list which stores the scores that the player reached in every game and ranks them.
5. Resource system collect gold each time the player eliminates a monster.
6. Store in which player can buy different towers with different attack damages.
7. Automatic navigation system of the monsters which allows the monsters to find the Castle and carry out attacks.





# Features planned but not implemented

1. Different monsters with different powers.
2. Different bullets with different attack damages
3. Different levels of different difficulties in the game.

We did not implement these features because they will take lots of time to do 3D modeling and programming, and we have time constraint for this project.

# Potential improvements

1. Implement the features planned but not implemented above
2. The 3D environment and characters of the game can be more beautiful and complex, detailed.
3. The game can have more complex and scary animations.
4. Improve the algorithms/codes to make the web game more efficient to save the transmitting /computing time and resources

# Framework and technologies adopted

We use C# to program the game in a game engine called Unity3D. 3D modeling is also done in Unity3D, and we use the graphics editor called Adobe Photoshop to make the textures for the 3D models. To build the website for the web game, we need to use an API called WebGL (Web Graphics Library), which can be done in Unity3D. Our version control is archived by Github. We use the website hosting service called SiteGround and Cpanel to publish our website, the FTP (File Transfer Protocol) is also used to transfer the files of for published website to Cpanel.

# Actual implementation schedule

Preparatory Phase

Hongyu did project scheduling in frist week. He made schedule for every aspect of the game. Manisha started doing the documentation, She did all the documentation while others started preparations on their parts. Some work was also done on the game in first week which continued in second week.

Phase 1 & 2 Work Effort

In this week, Varuni started working on the design of the game. she started to create prototype for the game. Manisha started working on 3D models. She was designing all of the model of the game. Hongyu created 3D models used in the game. He created 3D environment and characters of the game. Some work from phase 2 and 3 was also done in this week.

Third week: phase 2 and 3 work effort

In this week, Hongyu started game programming in Unity3D. Varuni was planning for the final steps, she was also tracking the progress as it goes further. Manisha was in charge of the version control of the game. She was taking care of all the version control for the game.

Final testing and delivery phase

In this week, Varuni finished designing and she was tracking progress towards the final steps. Hongyu was working on some back-end works and built the website. He finished his modelling and programming. Manisa started testing and debugging process. She solved any bugs from the game which were creating error when the game is runing. In the end, the game was tested by the team. Varuni did the website publishing and hosting.

# Project setup instructions

Firstly, we need to design the game mechanics:

There is a point where Monsters spawn each few seconds.

The Monsters run through the world towards a Castle that they want to destroy.

The player can build Towers that attack the Monsters.

In addition, we will use Buildplaces in Unity3D. Buildplaces are predefined places on the map where the player can build Towers on. We will use them because it will make the development much easier later on (in compare to being able to build towers everywhere on the map).

Next, we need the deploy and install the development environment:

Game engine and 3D modeling sofeware: Unity3D

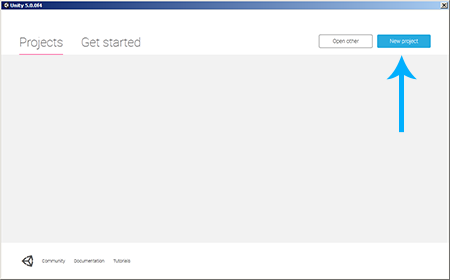
Raster graphics editor for creating textures for 3D modules: Adobe Photoshop

WebGL (Web Graphics Library) module for building the website

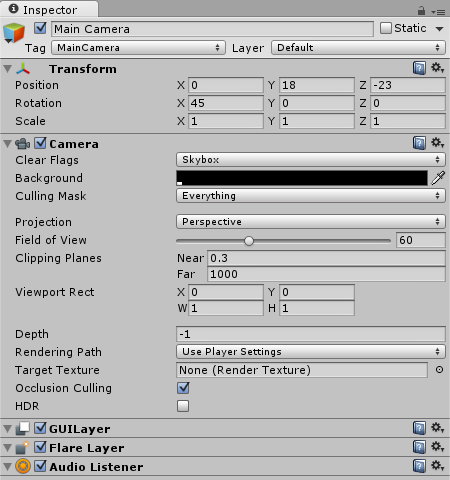
FTP (File Transfer Protocol) software for website hosting: FileZilla

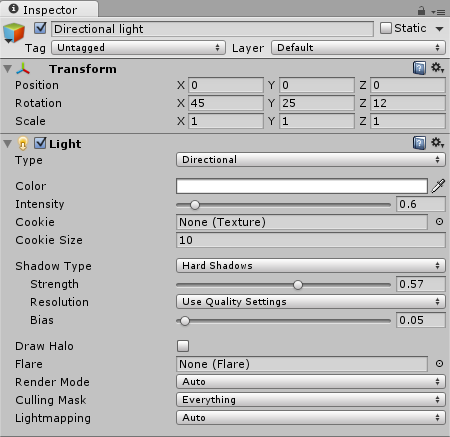
Create SiteGround and Cpanel account for website hosting and publishing

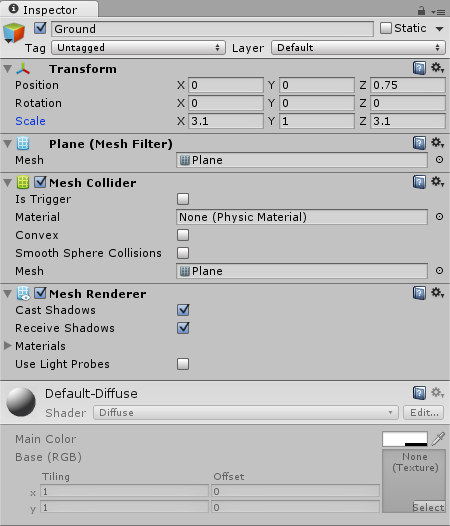
Then, we created a new project in Unity3D:

We start Unity and select **New Project**:  


We name it **Monster Rush**, select any location like **C:\**, select **3D** and click **Create Project**.

We select the **Main Camera** in the **Hierarchy** then we can set the **Background Color** to black, adjust the **Position** and the **Rotation** like shown in the following image, this will make the camera look down onto the game world in a 45° angle later.:  


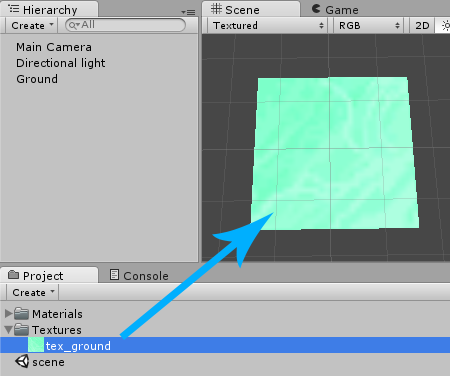
Let's add some light to our game, so that the world won't be too dark later on. We can add a light source by selecting **GameObject**->**Light**->**Directional Light** in the top menu. We will then use the following settings to make sure that the light shines in the perfect angle onto our scene:  


We need some kind of Ground for the Monsters to walk on. Let's add a Plane by selecting **GameObject**->**3D Object**->**Plane** from the top menu. We will name it **Ground** and assign the following **Position** and **Scale** in the Inspector:  


This will be the perfect size to fit all the Buildplaces and the Castle.

Let's open our drawing tool which is photoshop and then create a small 40 x 40 px texture with just some basic green tones. Filling it with some base color and then drawing in a few random lines with a slightly brighter or darker green tone. Here is what we came up with  
Unity Tower Defense Ground Texture

Now we can drag the Texture from the **Project Area**'s Textures folder onto the **Ground** plane:



Above is the setup of the project before we start the development.