CS235 Final Project Report:

3D Enemy Blocks

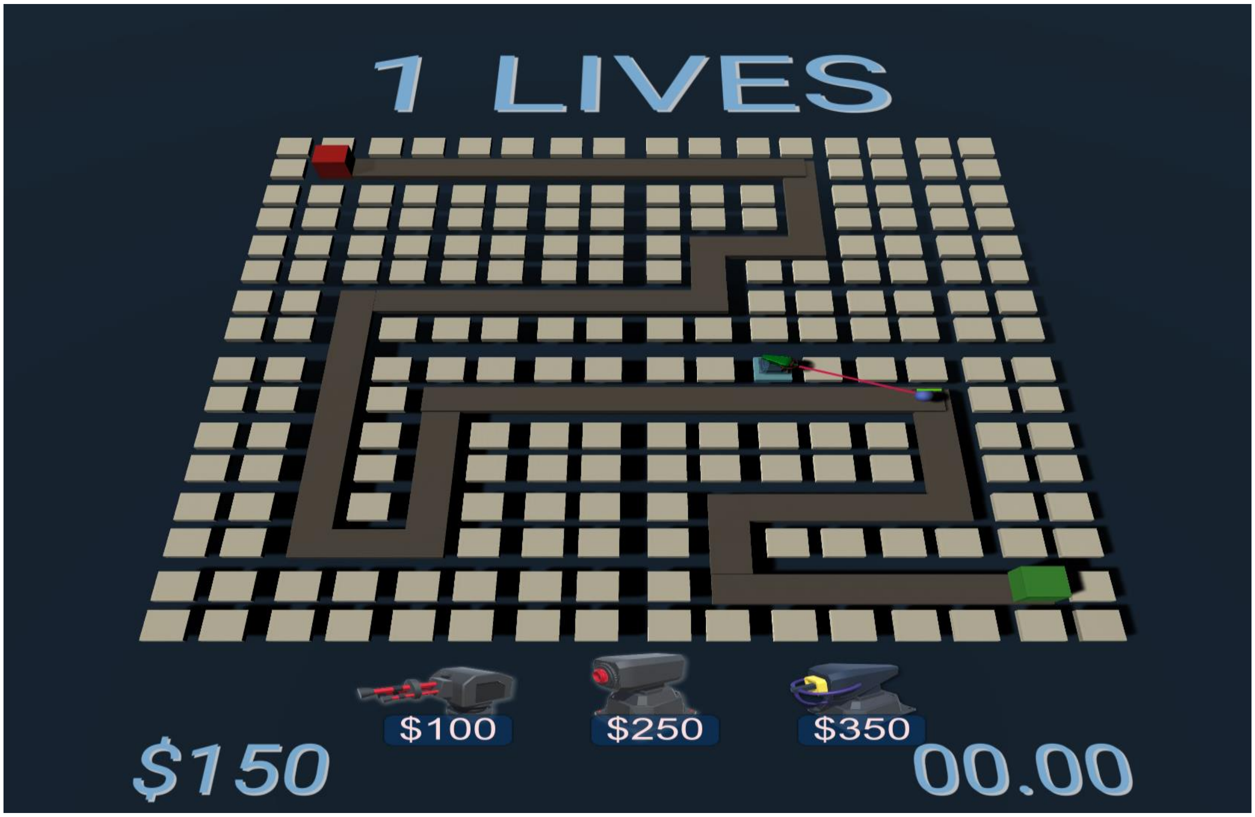
Yogesh K Singh (862187442)
ysing012@ucr.edu

Overview

Game is developed in Unity Engine. The motivation for the game was Tower Défense Game 2D. This is a 3D game and has various other upgrades from the original Tower Défense game. Basic idea of the game is stop enemies from reaching the end point. Every player is given a selected amount of budget and they have plant, upgrade and sell turrets within that budget and stop the enemy blocks from reaching the target. There are three kind of enemies and three kinds of turrets and a total of three levels as of now. Waves of enemies are spawned randomly after destroying the existing wave and each level has fixed number of waves and enemies in a wave. Entire scripting of this game was done in C#.

About the Game

3D enemy block, starts with a home page with two options to play/quit. After hitting play, the user is provided with various options to select the level. The levels are initially locked and unlock as on the user keeps winning levels.



Primary contents of the game are

1. Start and End block
2. Grid with waypoints on ground plane to guide the enemy direction and movement
3. Three turrets: standard turret, missile launcher and laser beamer. Moreover each turret has sell and upgrade option
4. Shop: for purchasing turrets
5. Wavespawner timer and budget information panel
6. Lives left panel
7. Three enemies: Standard, Tough and Fast
8. Main Menu and Pause Menu

On the top of the canvas, you can see the number of lives user has. In the bottom right corner, user has provided with a timer that tells him about the next wave and in the bottom left, it tells about the amount of money user currently has. In bottom centre, shop has been provided. In this shop, user has three options. The standard turret, with average hit and average range. Missile launcher with better hit and range. The laser beamer, with continuous hit, average range and slows down enemy.

The game also has a pause button that provides users with three options. Continue, retry or return. User can also end the game by pressing ‘e’ to end the game. The end game panel tells users, how many rounds it survived. The user can also utilise keys ‘w’, ‘s’, ‘a’ and ‘d’ to move the camera in the scene and use mouse wheel to zoom in on the scene. These options are provided to get the user better view or focus in a single part of scene.

There are three kinds of enemies in total, the first one or the blue enemy has average health and speed, orange enemy has high health, but it is very slow, and the yellow enemy has very less health, but it is very fast. The wavespawner produces a wave, after certain amount of time with a certain number and mix of three kinds of enemies. Next wave is produced after all enemies in the current scene or all the enemies of the current wave are killed. Attached to the enemy is health bar that tells user about the health of enemy.

The user loses money every time it purchases a turret but gains money after killing enemy or selling the turret. If the user doesn’t have enough credit to purchase a turret and yet selects the turret, user will not be able to lay down the turret on the grid and hover color will change from blue to red. Different turrets have different upgrade amount, and upgrading the turret increases its range and damage. Three turrets are designed so as to specifically fit the requirements of three kinds of enemies. Standard turret deals with the average enemy (blue), the missile launcher is for tough enemy as it has good damage and laser beamer is skilled for fast enemy as it slows down its enemies. Correct placement of turret around the scene and constant upgrades in position and the turret and switching the turret in the scene at correct time (between waves) comes in handy for winning the game.

Motivation

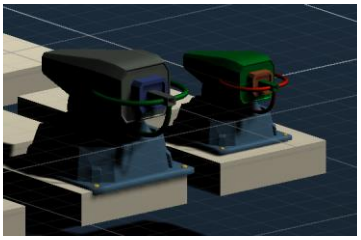
Apart from 2D Tower Défense game, the game develops over the idea of two 2D games namely Spaceship 2D and Pocket tanks. Pixel art Spaceship 2D gives a fair idea of how to upgrade the user-side of the game eventually so that the user always hangs around the game without giving up even if the levels are too high.

On the other hand, pocket tanks is very good bases on variety of powers it provides a single player and giving him the importance of how to judicially use it. One major drawback in these two games is the visuals, which gives user an image of a small 15-minute game. In spaceship 2D, provision of too many lives makes the game less challenging and the main character/object of the games occupies lot less screen than the enemies which give a feeling of repeated nature of this game. In pocket tanks, the setbacks are different but still prominent. Nature of the game gets less challenging and longer and finally the powers provided to the users are static. This means, user never get to earn the upgradation, he just stays in his level and waits for an unlucky time where the game might get challenging.

3D Enemy blocks overcomes all these challenges, where in user machine will have subsequently fair screen size, 3D visuals, he/she will have to earn and buy the upgradations, opponent automatically gets upgraded (faster than user) with each level and finally the game continues to challenge the user even in the initial level, and it is up to the user to play more, earn more credits, upgrade its equipment and the cross the level.

Work

The game currently has 20 prefabs, 5 scenes (Main menu, level select and three levels), 14 materials, 3 animation and 23 scripts. The scripts and the functions are divided so as one can easily update and make changes, add features or remove some without wasting too much time to understand what the script utilises. The entire game was build using Unity Engine. After the setup, first the grid was laid down (as in the figure) with start and end block. Waypoints were added in the game, so as enemy follows a specific track. Each turret with different shooting effects and enemy death effect was also added to make the game look more interesting. The concept of budget makes the game even more challenging and easy to grasp. The turret upgrade gives a heavy look and get the user more capable in winning the level. User also has sell turret option so that the game goes on even if he has lost all money and doesn’t have to depend on killing the money.



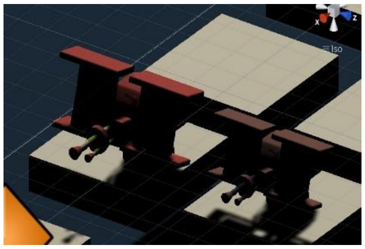


Figure 2: All three turrets and their upgraded versions

Development of the game was a huge learning experience. Made me familiar with variety of options provided in Unity. Different type of UI options, effects option, panel option and 3D object option made the development easier. Also, connection between unity options and providing them in C# script is now a lot easier with so many updates in Unity. Adding animation by selecting frame, and option to change your animation curve using the GUI, makes it a lot easier and fun to improve the look and feel of the game.

Strict adherence to the time schedule and providing regular updates for the development also helped to make the development of the game less taxing. The use of Unity built in PlayerPrefs for storing the progress of user and realising the logic of unlocking level as and when user wins a level is something, I have encountered for the first time. Furthermore, build settings was used to serialize the encounter of all scenes.

Basic and most important learning from the project was: how to build 3D objects, add effects to them on certain points such as hitting target or death of an enemy, associating these effects using the script on correct time, looping over them for certain amount of time. Also laying down canvas, different layout groups to make their alignment automatic was a special learning experience. Development of this game also made me understand the importance of having prefabs, and the importance to keep the document structure organized.

There were many challenges faced during the development, like developing the wavespawner script for implementing the generation on enemy, developing the shop and developing the upgrade and sell button. It was also difficult to design and work on so many prefabs, building so many affects and adding animation to movements. Only the turret assets and icons were downloaded from the internet due to lack of knowledge of photoshop and this saved a lot of time. The final aim was to develop a bug free and interactive yet logical game for this project.

Future Work

The game still has a lot of areas to be toggled with. One can incorporate more levels, more kinds of weapons and different kind of enemies. One of the drawbacks of the game is that enemies are not generated randomly in the game, and that is something one can improve on. Another section that can be worked on is: instead of deducting lives after any of the enemy reaches the final target, the target should also have a health and it should be affected differently on different enemies reaching the target. Game can also have interactive enemies rather than just different color spheres. Another thing is that currently the size and structure of the grid is fixed, one interesting thing to do will be change the layout of grid or make the path longer and with more turns based on different levels.

The grid and shapes used on its development is something one can also work on. The current look and feel of the turrets can also be improved and shooting effect can be made more realistic by adding lighting and more animation. Original setup of the game can be said to be a lot zoomed out and this is something that can also be fixed. Even after this, there are many aspects wherein the game can be realised with better UI and be made more challenging, but these were some of the notable ones.

Conclusion

Given the amount of time, the development of 3D Enemy blocks is complete. The game is interactive, challenging keeps the user surprised with variety in levels, weapons and enemies. The scene and working of the game are bug free. The scripts are neat and well organised, the materials and color combination used in the game well planned and doesn't appear shabby. Finally, the game has been provided with enough effects so as to realise the user what is going on in the scene and pause menu and end game menu makes the game even more realistic and interactive for user. The use of PlayerRefs allows the user to have its progress saved and not lose it every time he ends the game and therefore does not have to restart the same levels again and again. The structure of the game moreover makes it easier for anyone to modify or upgrade the game in future. Currently, the development of the game is complete with a fair amount of options of future work.