SEARCH FOR A STAR X

The Last Fort
SFASX 2020 Entry

Daniel Zammit

MSc Computer Game Technology

Abertay University

Overview of Project

The developed game falls under the tower defence genre. I was able to achieve this gameplay mechanic and therefore I have completed the objective I set out to reach when I started working on the project.

When I first saw the provided code, it was a bit overwhelming since this was the first time I took part in such a challenge. However, after careful consideration, I decided to keep and understand the code provided in the template. I was successful in understanding how the code snippets worked and was able to build upon this functionality.

The result was a fun and challenging tower defence game that can be enjoyed by everyone.

Research & Development

Major research efforts were focused on understanding the template code that was provided, as I could envision the final version of the game making use of the tile pathfinding. I understood how the Solve algorithm worked by getting familiar with lists and neighbour system. Once I was able to understand this, I started to experiment with the Solve algorithm by feeding it player and enemy tiles. The second challenge was to ensure that path chosen by the algorithm from the player tile to the enemy, was the same path from the enemy to the player tile. Given that this was being built using Lists, I managed to solve this hurdle by storing the player path, reversing it and assigning it to the enemy AI.

Although I was able to tweak and modify the existing tile-based pathfinding, I did not delve into the understanding of the full system, such as Parent or Visited. In this case, it was beyond the scope to fully understand how these functioned. Instead, I focused on developing a structured code base that can be understood by other programmers that might take on the project for further development. I decided to use Singletons for specific game controllers. One of the advantages of Singletons is ensuring that only one instance exists in the game, and that instance will be declared the only instance to be used. This offers peace of mind for any developer, especially in large teams as when someone decides to add another Singleton class to the scene, of the same type, this will not be allowed. The following scripts are a Singleton:

- Game Controller
- Audio Controller

I also came across a technique while watching tutorials for establishing a helper script for implementing the same functions across different scripts. By using **interface** keyword, I was able to create an interface between different classes and implement that interface to start off with functions to be defined for that class already in place. This was extremely helpful when working on unit classes.

Gameplay & Features

At the start of the game, the player needs to select a tile where to place their base. In order to create balance, the environment controller script allows the enemy base to spawn anywhere within the second half of the map. Therefore, the player can place the base on the first half of the map. Once the game controller detects that a player base has been set, the economy counter starts going up.

Due to time constraints, I had to come up with a solution for AI unit generation that felt realistic while not overpowering the player. AI units are generated based on a timer and spawn delay was designed in such a way that made the game balanced, engaging but not too easy either. This allows the game to be fun regardless of the player skill level. The player can generate up to three units: soldier, knight and giant. These are the same for the AI, however models are swapped to make a distinction. Once a unit is generated, it will follow the path from the respective base to its destination. The unit will only have a chance to attack an enemy unit once, during movement. This create a gameplay dynamic where perfectly timed units can be matched to achieve victory. The game ends when either of the bases are destroyed.

User Interface

The UI is comprised of several canvases structured under the UI game object. The respective canvas game object is controlled through the game controller script to manage the functionality during different stages of the game. There is a main menu, in-game UI, pause menu and map selection UI.

Closing Remarks

To conclude, creating this game was really challenging however I must admit I did learn quite some useful tricks and techniques. This helped me understand more the concept of smokes and mirrors in the games industry. I am looking forward to taking part in the next game jam as these events really help you develop both one’s social skills but also their technical abilities.

References

PolygonKnights – 3rd Party Low-Poly Asset Pack from Synty Studios

Animations – Sebastian Rausi using Adobe Mixamo

Several Brackleys explanation videos on Youtube - http://brackeys.com/
https://www.youtube.com/channel/UCYbK_tjZ2OrIZFBvU6CCMiA