ROTAT-A-BOT

Task 4: Evaluation

Daniel Cutajar, Dante Camenzuli

7/6/2019

Using the iteration model, similar to the previous assignment work was split into separate tasks that I tried to finish independently. This was extremely helpful due to the fact it wasn't an individual project but group work. Sectioning work helped divide the work load and make our tasks not overlap.

DANIEL

- Progressed with the iteration model completing the base mechanics of character movement and map rotation first.
- The map rotation was by far the hardest section to complete because it tied into so many other components of the game and constantly had to be adjusted and fixed. The approach taken didn't work completely with miniscule offsets but to the naked eye works well.
- Due to time constraints only 3 levels were developed which hindered playability. With this in mind others aspects were heightened such as difficulty and replayability to get the most out of these 3 levels. With more time we could have definitely made a more complete game.
- Level design was my weakness. I have no experience in the field and had to come up with complicated yet functionally progressive levels. The levels have flaws and the batteries are useless some of the time due to the complicated nature of rotation puzzle platformers but I believe I tried my best to create a fun brain teaser.
- Made heavy use of player prefs to deal with all saved data such as scores and levels played. This worked well and had to have many if statements to determine whether to save or not. If a level is opened it should be set to the last level played but only if it is completed should the next level be unlocked on the level select and the stars also made use of player prefs saving the players last score and their high score.

DANTE

- Enemy follows player when within reach and avoids running into spikes. From the code implemented in some game loads it would function perfectly whilst others would have it glitch. With more time this could have been ironed out.
- Enemy turret activates and follows the player by rotating on its origin. Pellets are instantiated every two seconds but do not fire in a straight line. This AI had to be scrapped.
- Score system relies on rotations. Each level starts with ten available rotations thus a score often, and each one reduces the score by one. Batteries add two more rotation and thus two points. Once the score reaches zero the player must attempt the level again.
- The timer ticks down every second until it reaches zero in which the player loses and must attempt the level again. Timer multiplier was scrapped from the score system.
- Level Complete scene displays the win or lose GameObjects as designed. Score from the previous scene is retrieved without issue and saved to PlayerPrefs file.

FUTURE IMPROVEMENTS

- If the game were to be touched upon in the future, more levels would definitely be the focus to create a longer playing experience.
- The AI turret would be worked on in order to create more variety to the game with hopes of adding other unique aspects to each level.
- The workspace needs to be utilized more carefully, gameobjects having null references arised a lot due to the over inclusion of scripts and the folders were not as organized as desired. This is manageable for an individual project but with group work is far more important.
- Lastly communication between teammates would need to be much better. Many times changes were made to the other persons implemented code without confirmation causing back and forth confusion between both parties. A better line of communication would have made development much easier and copable.