**Milestone 8**

**Functionality**

Firstly, I would like to note that I have abandoned creating bullets in the essence of time, as I am getting close to my extension deadline and bullets aren’t that important for the game I am trying to make. Secondly, no changes were needed to detect a game state, as I was able to use the entity system to be able to detect when the player has died. Therefore, all the code changes I made were either small improvements, or to do with the scoring system.

Simulation has some new functions centred around the scoring system. The score will increase each second within the GSupdate() function, and it will call the updateScore() function, which uses the digits created by the createScoreCounter() function to display the score.

**Code Changes**

*Main*

**ADDED** – *if player is dead* – I added a new variable called gameOver and an if to detect if the player character has died. If that is true, it will run the game over state once before turning gameOver to true, which will mean it will freeze in that state.

*Class simulation*

**ADDED** – *int score* – Stores the score of the game.

**ADDED**– *int scoreFrames* – Stores the number of frames since the score was previously increased. This is used to know when to increase the score again.

**ADDED** ­– *std::vector<int> digitIndx* – Stores the objectVector index of each digit created by the createScoreCounter() function.

**CHANGED** – *void GSupdate()* – Now updates the score if keepScore is true. New parameter: bool keepScore.

**ADDED** – *void createScoreCounter()* – Creates 5 objects which are used to display the score. Stores each object’s index in the objectVector in digitIndx.

**ADDED** – *void updateScore()* – Updates the digits visually to match the score stored within the class. Firstly, it will use the new numOfDigits() function to get the number of digits the score has. Then it organises the objects into the right position, and then calculates the individual number for each digit and changes the sprites to match the score.

**ADDED** – *int numOfDigits()* – Takes in a number and outputs the number of digits it contains, up to 6 digits.

**CHANGED** – *void moveObject()* – Some values were tweaked to improve this function. Hopefully now objects are less likely to get stuck in other objects when moving towards them from underneath.

**Tests**

*When the player dies, a game over pops up and a score is shown*

