CS 480

Khoa Le, Thanh Le, Lam Nguyen

Project Progress Report 1

The prototype scenario of our system consists of a 10 by 10 matrix with each element of the matrix represent position of an object in the game. Currently, these are objects that have already implemented in our code-based game and worked fairly well: a player, an enemy, a base (which the player has to protect), walls, water, and bullets.

The goal of our game is to defeat all enemy while protecting the base. The enemy have all the basic moves that the player has, but currently is implemented to move in a pseudo-random way.

The map in our game is currently a 10 by 10 matrix, so in each state, the player and enemy can make their moves (either moving or shooting) which change the state of the game from one to another. Moving will change the position of a player or enemy from one element in the matrix to another. Shooting will create a bullet advancing in player's or enemy's current directions and moving forward one element in the matrix after each state.

As the game progresses, if the player gets hit by a bullet (the positions of the bullet and player are overlapped in the matrix), the game will be over, resulting to a loss for the player. The enemies will be destroyed in a similar way. However, there might be multiple enemies, so the game will not be over until every single enemy is destroyed.

Consider the following simple scenario, a player is going to destroy one enemy. Here is the map (it is the actual graphic prototype that our group wrote)

1 2

3

Starting with state represented in map 1, the green square is the player, the black one is the enemy, arrows represent the advancing directions of player or enemy. In order to destroy enemy, the player will have to "aim" at the at the target and shoot when the enemy is not facing at the player. While bullets are moving, both player and enemy can move in order to dodge, so the above basic process can be repeated until the player wins.

