Overview of game concept and elements

We have designed a turn based role play game that allows users to control the actions of their characters as they encounter obstacles in the quest to reach the castle guarded by their opponent. Two players are able to play the game at one time, passing the controls between each turn. Players will select 3 characters each, based on their skill sets and will begin with their characters in front of their castle (in opposite corners of an 8 by 8 game board). Players can move 2 characters per turn, strategizing to guard their castle and also advance to take the opponent’s castle. Encountering a character controlled by the opponent will result in a duel; attack moves are dictated by the attributes of the character involved. These encounters will result in a loss of health points for one or both of the characters. The object of the game is to move at least one character all the way across the game board and take the opposing player’s castle without being defeated by attacks from the opposing characters. A victory results in more advantageous attributes for the character which will be updated and stored at the end of the game.

A game board is composed of a series of 2 dimensional array LinkedLists made up of unique Nodes that hold can hold either a Character, an Obstacle (such as water, a rock or a hole that the character must navigate around), an Objective (the goal points at opposite ends of the board) or an Empty Space. Each space on the board is coded by a position in the array and a LinkedList (with 8 LinkedLists in total, one for each row of the game board). This structure was used in order to allow the implementation of different object types into the board. The generation of the game board begins by filling the board with empty spaces and then randomly generating obstacles throughout the board. In any given space on the board, the chance of an obstacle being generated is 10%. Objectives for each team are then generated in opposite corners of the board and three characters for each team are generated in the positions surrounding their team’s objective. Characters are able to move one spot in any direction, but are not able to move into spaces that contain an obstacle, and must navigate around the obstacle.

GUIs are used to create an attractive interface where the positions of the character’s relative to one another and to the obstacles are clearly visible. The interface allows the player to click on a character and click on where they would like the character to move to move the character about the game board. GUIs are also used to present the selected character’s stats to the left of the game board. When the characters are selected at the beginning of the game, the stats are uploaded from a file, and at the end of the game the updates stats of that character are then saved to the file once again to be accessed in future games. This allows the player to save characters and gain points for that character over a series of games.

Summary of classes

A Character class has attributes for the type and name of the character as well as all of the stats for the character (including level, hp, strength, dexterity, magic, resistance, defense and XP) which determine the abilities of the character to fight other characters, and a boolean variable that signifies whether or not the character is alive. There are methods to set the type of character and to get the name and alive status. The Character class has defend and attack methods that take into account the type of character and the character that they are opposing.

Mage, Tank, Warrior and Ranger classes inherit from the the Character class. Each one creates an instance of the super class with different values for the stats attributes of the Character. The constructor also sets the character type.