EECS 448 Team 2

Software Architecture Document

For our project three, we believe that we used the Peer-to-Peer architecture strategy. While an N-tier software architecture may be more intuitional using the event-drive design paradigm, because our program is so small it is more of a Peer-to-Peer structure.

When our game is first loaded, the startGame() function is called. This function will call upon the other classes, which will call other classes and functions. Every class and function and piece of code is interconnected together to create the entire game. This is why the game is considered Peer-to-Peer. If our team were to implement even more classes and functions, there is a very real possibility that it could transform into an N-Tier architecture, but with what we have right now, which is the basic functionality for a player to jump over obstacles until they reach the finish line, is definitely peer-to-peer.

To further demonstrate how interconnected our code is, here is an example of when the player loads up a game, misses the first obstacle, and loses the game. When the game is initially loaded, the startGame() function is called. This function will call upon the object of myGameArea which will call the function start. This function will also call on the Character class, the ground variable, and the Obstacle class. The start function will call on event listeners which will call upon a variable “key” which will then determine actions that the character and obstacle do as the player runs forward. This can go on and on. It is clear that every single variable, function, and class are interconnected in a web with no tier system involved, simply due to the simplicity and size of the project.