EECS 448 Team 2

Design Paradigm Documentation

For our EECS 448 Project 3, we have chosen to use the Event Driven Design Paradigm to make our obstacle path game for users to play and interact with. We feel as though it is definitely very obvious to use this design paradigm for our game considering the principles of what the game is.

In our game, the player will have to jump over different obstacles to continue running down the path to the finish line. If they hit an obstacle, then they will lose the game and have to start from the beginning of the course.

An event driven design paradigm is one where a specific action or even given by the provider triggers a change of state within the project that will be transmitted all the way down by message channels to the consumer.

A great example of our program acting as an event driven design paradigm is whenever the user accidentally hits an obstacle. In this scenario, the user is the provider in the fact that they have provided a fact that they hit an obstacle. This will trigger the message channels as functions are called to stop the timer, stop the player from running, and finally end the game because the player has lost. The consumer then will provide a reaction, and in this case the reaction is to display a “Loser” message and give the player the option to restart the game.

It is clear that the event driven design paradigm works for our game because every single function or action that is called all depends on an event given by the player (player runs into an obstacle, crosses the finish line, or successfully jumps over an obstacle). After this even is called, our program will react as necessary to give the player options (“Loser” message, “Winner” message, restart the game, etc.).