Software Design
Brenna Manning & Joe Sutker
March 12, 2015

Mini-Project 4 Interactive Programming

Project Overview:

We wanted to create a fun interactive game in python using the pygame library. In our game, the user could move their character around the game space, avoid enemies, and try to survive for as long as possible.

Results:

We created a game, in which the user moves their character around the screen and tries to avoid the ghosts that are trying to attack it. Over time, the ghosts bounce off of the walls and get faster and faster to make the game more difficult. If your character comes into contact with one of the ghosts, the game is over. The goal of the game is to stay alive as long as possible, and your score is displayed in the top-left corner.

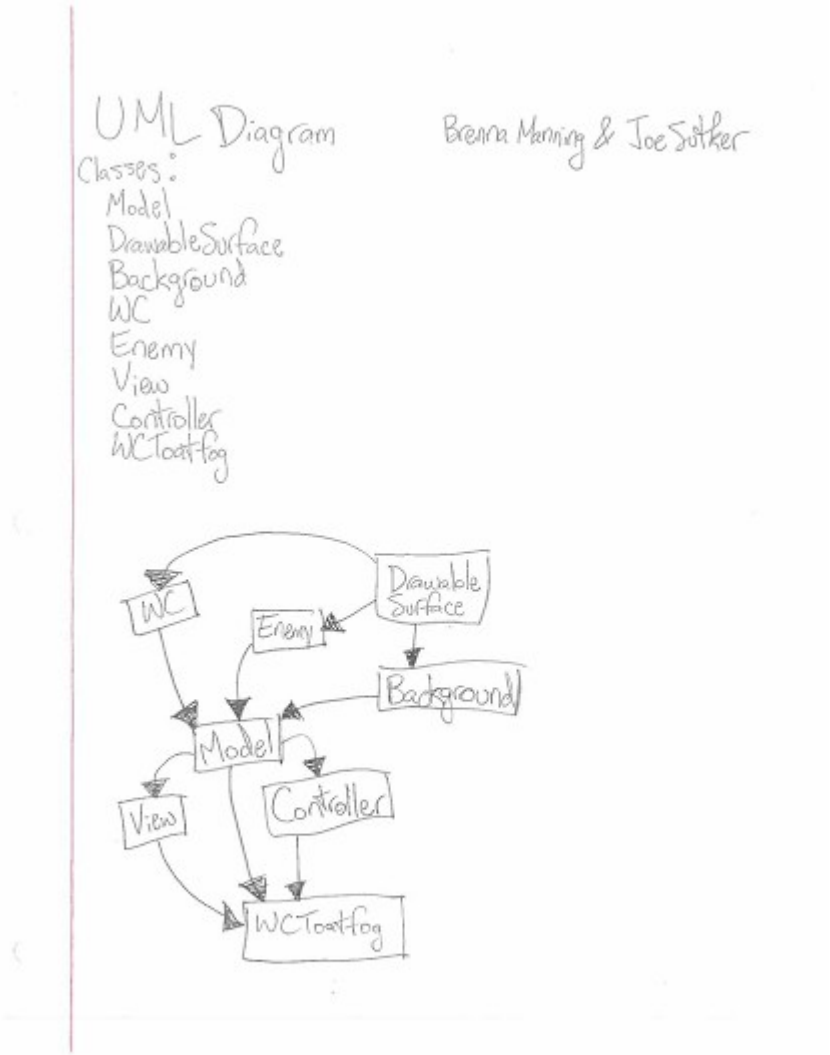
The image on the left shows what our game looks like. The purple hooded figure who we affectionately named W.C. Toatfog is the character that the user controls. The blue ghosts bounce off of the walls and speed up as the game progresses. The longer the player can avoid the ghosts, the higher their score will be.

People who have played the game said that “It was really fun!”

Implementation:

We used the Model/View/Controller method for writing our game. The other classes we used were: DrawableSurface (to make the surface to create the drawable objects on), Background(to draw the background of what the game looks like), WC(the class to create our character), enemy(the class to create our ghosts), and WCToatfog(The class to bring everything else together and run the whole game).

UML Diagram:



Reflection:

Initially, our project was not appropriately scoped. We had wanted our game to include the character moving to different settings, shooting enemies and being shot at while trying to survive. Throughout our process we realized that this was too much for us to do in the allotted time for this project given our current experience/skills. We eventually decided to scale it down to a game that was simpler and easier to implement, but also in our opinions, more fun to play. This is how we eventually settled on the game idea that ended up being our final result. Going forward, we have learned that we should think more about what we think we will be able to accomplish in the given timespan, and set more realistic goals.

For our team process, we did most of our early work in pair programming. We got the general form of the game set up, and we had all of our code written in model view controller format. After this point, we each did some individual work. We didn't specifically divide up by class in the code, but rather took turns implementing or improving aspects of the game until it was complete. If we were to do one thing differently in the future, it would be to plan better and to figure out scheduling earlier. At the beginning of the project, we created a schedule of when things would get done, but we were not able to finish all of these things when we wanted to at the times we had wanted. In the future we would like to plan meeting times for pair programming further in advance so that we could have meeting times that worked better for both of us.