Ashley Weaver, Jason Husted, Tristan Hunt

Status Summary (15 points) - Ashley

Work Done:

Since none of us had worked with pygame before, the majority of our two weeks were spent learning the basics and implementing the base version of our game. We were lucky enough to find some tutorials on the internet (they are cited in our GitHub repository) that helped us set up our initial build. Since then, we've been making/finding various Flappy Bird assets to add to our game and figuring out how to use object-oriented programming design patterns.

Ashley is currently working on setting up a main menu for the start of the game and Jason is working on the Observer pattern for the high score. Tristan is working on formatting the code so it flows better (looks like a proper UML diagram).

Changes or Issues Encountered:

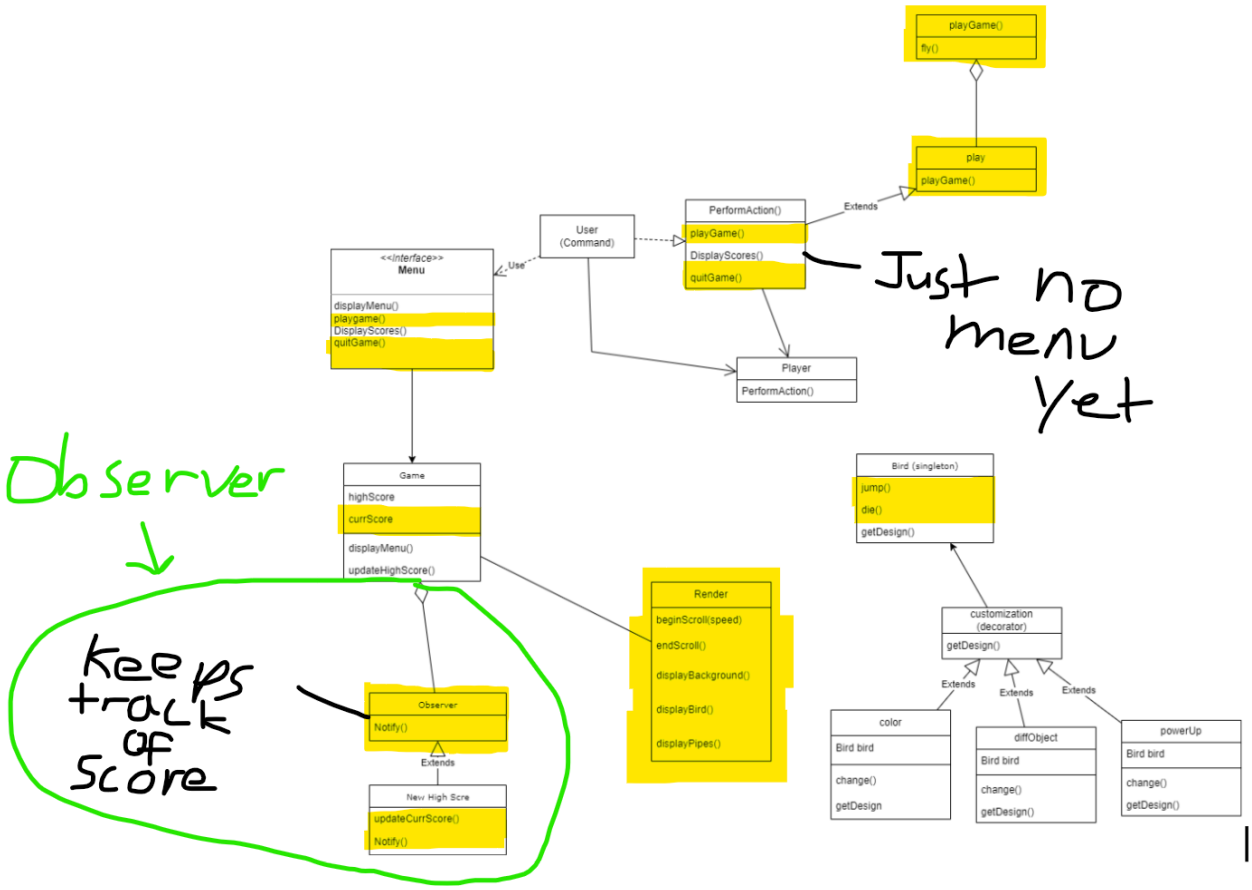
When we first started this project we wanted to program it in Java (specified in the previous milestone), but ultimately had to switch to pygame because it was easier to set up. When we looked up tutorials online about how to get a java environment running to make a game, most of the tutorials were out of date or were no help. After a few hours of troubleshooting, we eventually came to the conclusion that Java wasn't going to work.

Another general challenge we're running into is our knowledge (or lack thereof) in pygame. Since we are all beginners, it has been pretty difficult figuring out how to level our game up so that it's more refined/complicated (not just the base flappy bird game). We have been making progress, though, and it's been fun learning something new!

Design Patterns:

Since we switched to a new language, it's been difficult trying to figure out where we are going to put the design patterns. That being said, the first (and only so far) design pattern that we've made is the observer pattern to keep track of each score a user gets while playing. We will still be referring to our original UML to create our other design patterns.

Class Diagram (10 points) - Jason



Everything in yellow is the things we have implemented, the observer pattern is circled in green.

Plan for Next Iteration (10 points) - Tristan

So far we have a functional FlappyBird game which is able to generate pipes at different heights, display bird motion as well as current score. For our final project 7 delivery, we plan to implement a more detailed user interface which would display high scores (already tracked by Observer) and allow the user to customize features like the map background and bird style. A Command pattern will be developed to handle these user requests. We will also reform the “Bird” class to a Singleton pattern and add a Decorator to allow map/bird style changes. The repository generates an executable to launch the game with a certain command (specified in the ReadMe), but as of now this exe is dependent on classes in its file location, with extra time we may find a way to package the game independently.

GitHub Repo: https://github.com/jaydel14/OOAD_Final_Project.git