1. Name of project and names of all team members

The name of the project is GLADIATOR, The team members include only me, Nick Willson.

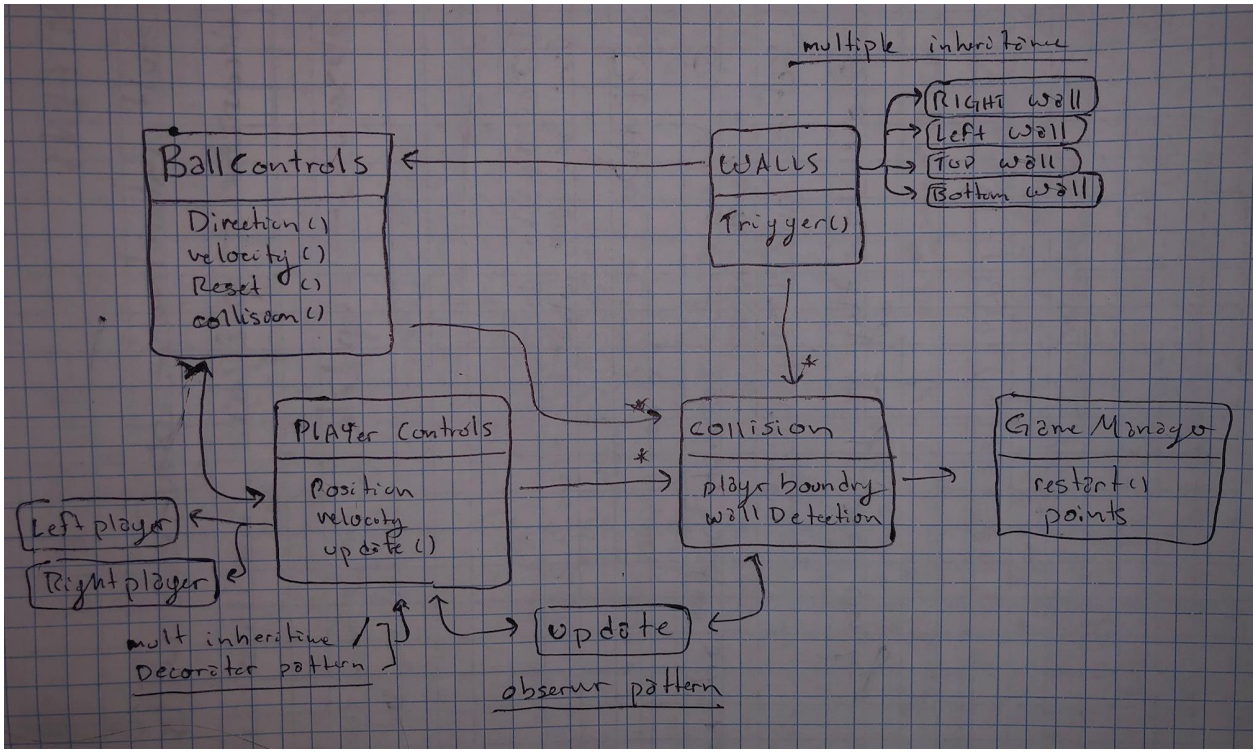
2. Final State of System Statement

• A paragraph on the final state of your system: what features were implemented, what features were not and why, what changed from Project 5 and 6

The final state of my system features more object oriented programming when compared to proj 6 and 5. I did not add too many new features to my game when compared to proj 6, However; I did a lot of changes for when it comes into the code. I added a better sense of inheritance as well as other OO concepts that we reviewed in class. In proj 6 my code was not as OO as I wanted it to be, however the gameplay itself was not drastically changed from proj 6 to proj 7 besides changing some small details such as how the ball and players move.

3. Final Class Diagram and Comparison Statement

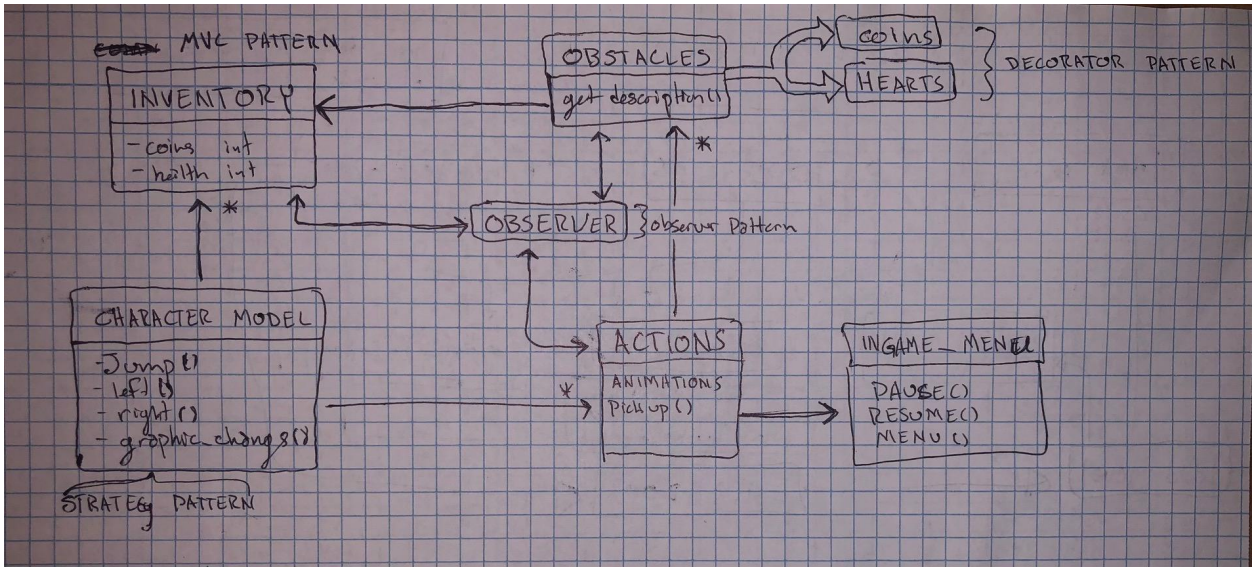
• A thorough UML class diagram representing your final set of classes and key relationships of the system



• Highlight and document in that diagram any patterns that were included (in whole or part) in your design

It is shown in my diagram above, however, I implemented a strategy pattern for my character model, an observer pattern with my update function, and a decorator pattern with my walls of the game. Although not noted on the map I use inheritance to instantiate the walls and players.

• Include the class diagram submitted in Project 5, and use it to show what changed in your system from that point into the final submission



• Support the diagrams with a written paragraph identifying key changes in your system since your design/work was submitted in Projects 5 and 6

Now although the class diagram from proj 5 has a similar structure to my current structure you can see that a alot has changed. First off my final submission UML diagram includes a class for my ball model and took away the previous inventory class. Aling with this the information inside my class diagram has changed to be more accurate to what is actually happening in these classes. Finally the way in which the classes interact with each other has changed in order to better show the correlation between the classes as well as updating the patterns different classes would use.

4. Third-Party code vs. original code Statement

• A clear statement of what code in the project is original vs. what code you used from other sources – whether tools, frameworks, tutorials, or examples – this section must be present even if you used NO third-party code - include the sources (URLs) for your third-party elements

For this project I used a tutorial for creating 2d unity games in order to get an idea of how to start a unity project. Along with this I used a code example from a game that had similar characteristics to my game. I used the example of their code to help me understand how to better use unity to interact with my 2d characters, as well as to get a better understanding of the tools unity has and how to use them as well as how to, make players move, have a GUI, call other classes, moving objects in the game.

https://www.instructables.com/Make-A-2D-Infinite-Runner-with-Unity/ - code example

https://www.youtube.com/watch?v=OR0e-1UBEOU - tutorial

5. Statement on the OOAD process for your overall Semester Project

• List three key design process elements or issues (positive or negative) that your team experienced in

your analysis and design of the OO semester project

- One issue I encountered in my design of this OO project was learning how to apply OO concepts to a unity framework. Unity by design uses OO ideas in order to structure their code with the character models. Because of this I had a difficult time trying to think of unique ways to add more OOP designs into a code base that has OOP built in, kind of.

- Another issue I had experienced was using Unity for the first time. The reason I did this was because I wanted to challenge myself as well as use a coding studio that I have always wanted to use. When it was announced our final project was gonna be a game with OO design concepts and using any code or studio we wanted I knew I had to use unity. I wanted to be a game designer for a long time and knew unity was the best way to get started. So an issue I had was having to learn how to use unity (which isn't too terribly difficult) and on top of that how to go against its coded framework in order to integrate the patterns we talked about in class. An example of how unity is different from most studios when it comes to the OO concept is how a class can call another class. To do this unity has a function called FindGameObjectWithTag('insert objects name'). This was difficult for me to understand and figure out how to use this function with the designs we talked about in class.

- A good and bad issue I had in the design of this OO project was that I only had myself as a team. Although I did not want to have to do this by myself I found it to be the best and worst part of the project. Since I was in a solo team I was able to think of and implement new ideas without having to consult another person. This made the actual coding of the project a bit easier as I didn't have to use github in order to work on code simultaneously. The negative of being alone however was that I could not create a super in depth script for my game as I was by myself. It would have been nice to have created a better game than the one I have now created, but the game I finally came up with seems to work for me

Code Submission – GitHub Repository URL with Complete Semester Project System – 30 Points

done

Demonstration – Recorded demonstration video – 30 Points

done