

Biking Game

Project Part 5: The Final Report

Brandon Barrett

Brandon Jacquez

Isabella Figueroa

Dilara Madinger

I. Implemented Features

Title of Feature	ID
Pause Game	UC-2
Move Player	UC-3
Encounter Obstacles	UC-4
Track Damage	UC-5
Track Score	UC-7
Get Penalty	UC-8

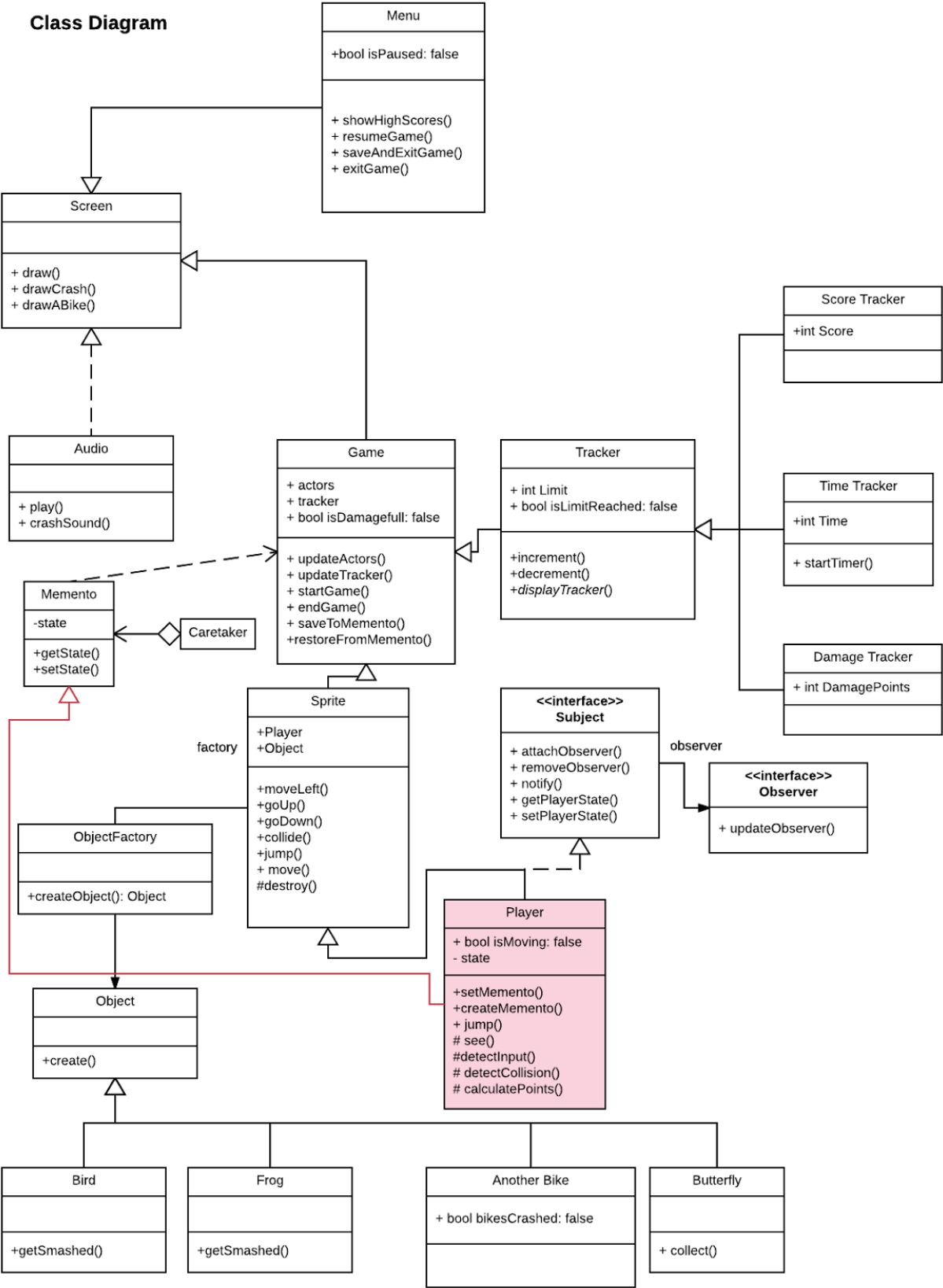
II. Features Not Implemented

Title of Feature	ID
Stop Game	UC-1
Track Time	UC-6
View Scoreboard	UC-9
Have Audio	UC-10
Encounter Another Bike	UC-11
Input Nickname	UC-12

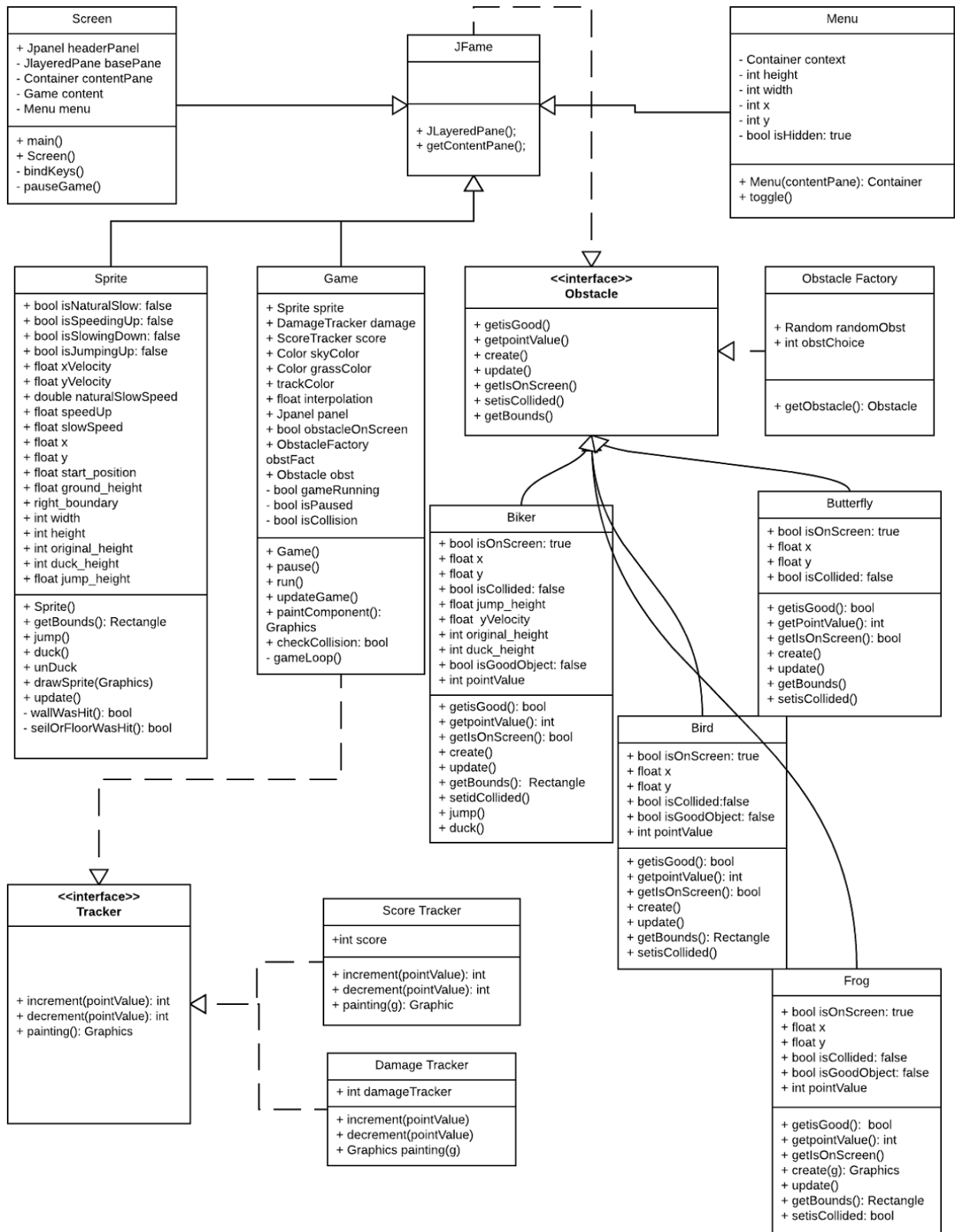
III. Class Diagrams

Original Class Diagram

Class Diagram



Class Diagram after implementation.



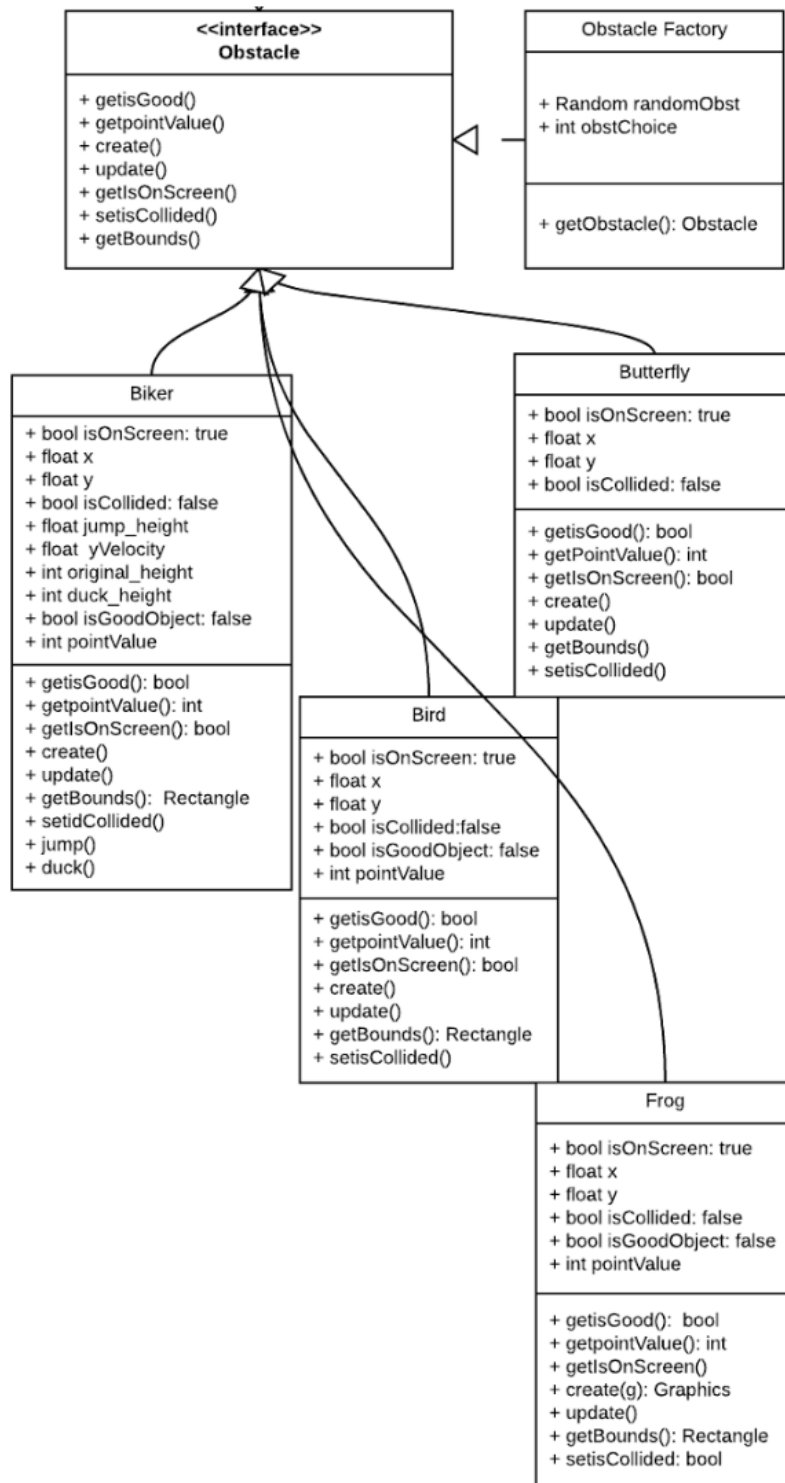
Final class diagram looks a little different, because during the implementation, we realized that the project is more complex and we had to use JFrame class to make all of the windows and simple graphics. Among the items that we didn't need to change was a Factory pattern to generate objects in the game. In general, it was very helpful to have the class diagram, as we followed it in dividing work and making decisions.

IV. Design Patterns

Originally, the team had planned on making use of three design patterns: memento, factory, and observer. Due to the scale of our project we quickly realized that a full blown observer pattern would not be necessary, as we would have had only one subscriber, our sprite. The use of simple key bindings worked very well to for us to move the sprite about the screen. Memento for a short, arcade-style game seemed to be unnecessary for our purposes. The only state we wanted to keep track of was the high score board when we're done and our pause functionality.

In the biking game we mainly made use of the factory design pattern. There were several types of obstacles that our sprite would encounter throughout the game, other bikers, birds, butterflies, and frogs. Using our obstacle factory we were able to randomly generate these various objects each with their own point values, their own update functions, and their own paint functions so they can render themselves.

Class Diagram Depicting Our Factory:



V. Things Learned

For the majority of the group members this was their first time creating a game. Our team took the route of designing the game completely before beginning to code any of the game and as exemplified by our class diagram, this meant a lot of changes throughout the design process. Over the course of the semester our team learned that having a strong design going into a project, and understanding exactly what is required of your final product is of the utmost importance. Even though the final design wasn't exactly the same as our initial design, it provided an excellent point of reference. We could easily see where mistakes were made in our initial design and learn from them quickly as we implemented.

Throughout the development process we also learned that the implementation of design patterns could simplify our code and make our project much easier to interact with. We also found that class diagrams can help understand our project as we're coding it. If we needed to get a variable on the fly, we could look at our diagram and see methods for how to get it or if that was even possible using our design instead of visiting each file and looking at the class definition.