**System Requirements and Design Specification**

**Milestone 2**

Authored by: Gregory Nathan, Spencer Wright, Timothy Anderson, Emily Richardson

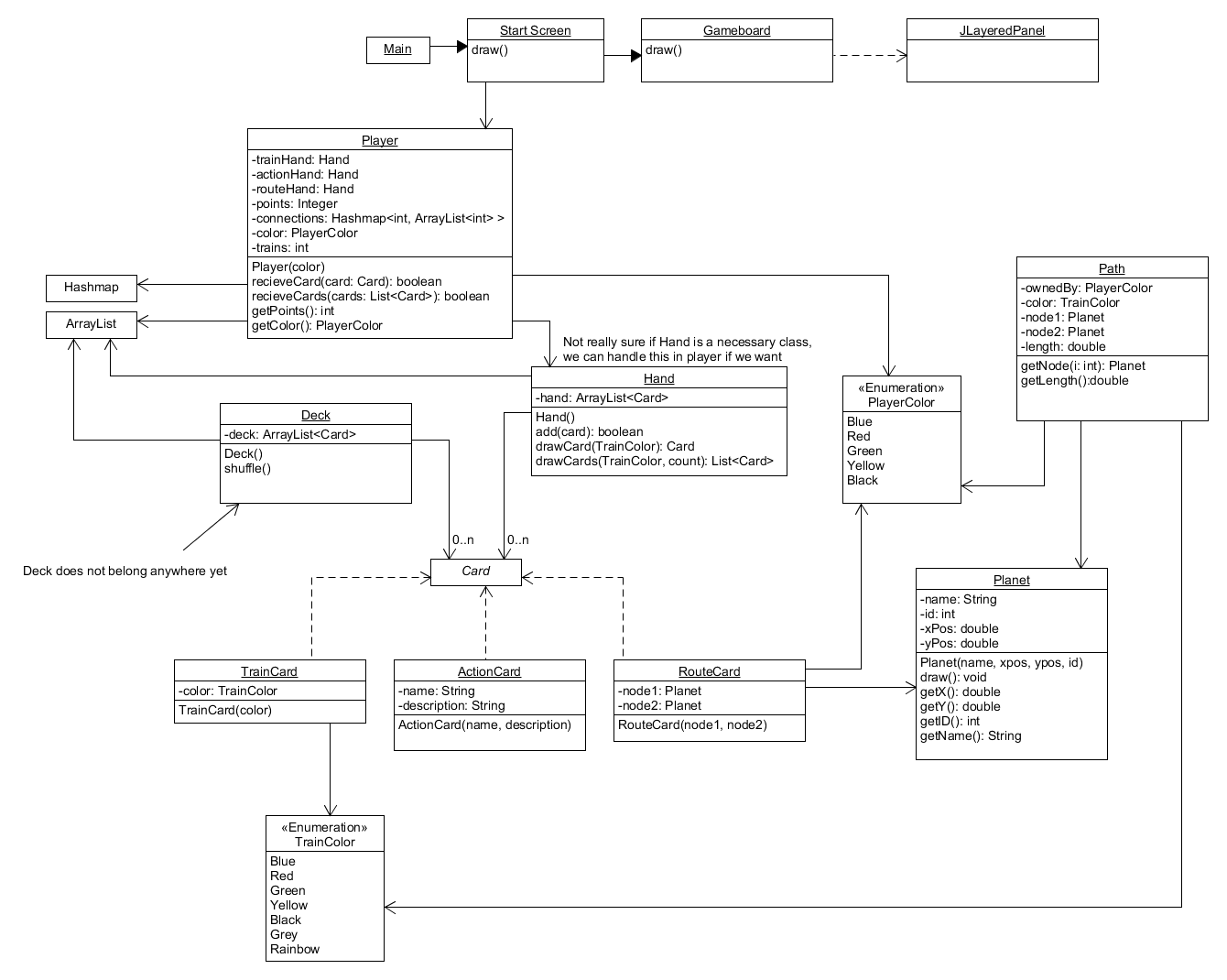
**Requirements and Design Specifications**

Our system requirements are stated here for clarification on what our system should do at the time it is completed. The Doctor Who Ticket To Ride application should provide the end user with a user friendly interface that allows them to play a basic edition the traditional Ticket To Ride game with a Doctor Who twist. Alongside this base game play, the game should also exhibit enhanced graphics and other features that will make gameplay easier and more convenient. Within this category of features, we plan to include graphics that will adjust to any machine’s screen resolution, clarity in drawing cards so that it is more obvious to the user what is happening, and finally, we will provide a method of allowing the user to access game instructions in real time.

Existing features and planned improvements combined, our requirements are simply to make gameplay easier and clearer for the end user. Our end goal is to make a hassle-free game application that users will enjoy playing for times to come. We want the graphics to be better than accepted norm, the game load to take a relatively short amount of time, and the in-game experience to be clear and simple by way of enhanced user interactions via the user interface.

**Software Architecture**

This section of the document describes an architectural overview of our Doctor Who Ticket To Ride application. As we refactor our project throughout the course of the term, our architecture structure will change to conform to our desire to create a clean, straightforward collection of code that is easily modifiable and expandable. As of the end of Milestone 1, our UML looks as follows:



As we continue through our milestone iterations, this UML is expected to become simpler with higher cohesion within classes and lower coupling between classes. A common end goal with this project is that the architecture of of code itself will be much more evident through self-explanatory code, an updated UML diagram, and relevant accompanying comments and documents.