**Software Design Document**

**Rocket League Bot**

1. **Introduction**

This document serves as the Software Design Document for the Rocket League Bot program. The purpose of this document is to have a detailed look at the design of the project as it relates to the stakeholders. Elements of the design to be looked at include design concerns, design views, as well as a design rationale. The bot program will merely tell the “car” what to do through the official Psyonix Rocket League API. It does not edit or change any game files and cannot be used for online play. The program will be limited to running through the official API which will be looked at in this document in terms of how it relates to the design. I will not be changing any program files and will be using the code framework provided by the RLBot organization.

* 1. **Stakeholders**

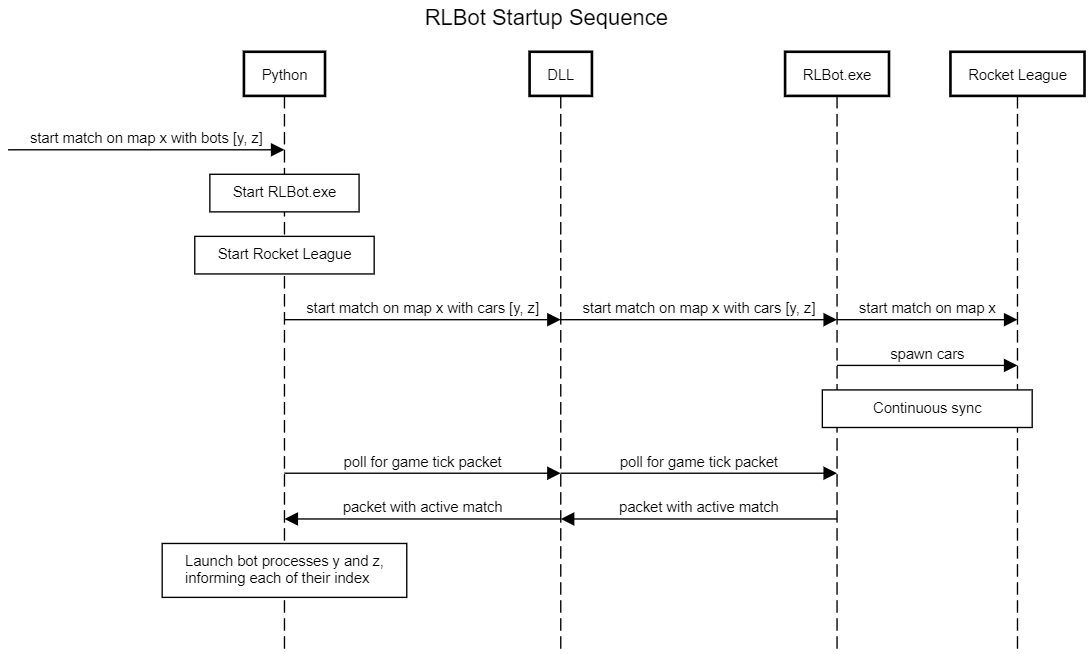
Stakeholders for this project include myself who is creating and programming the bot. Psyonix created the game Rocket League and provides an official API for the purpose of bot creation. RLBot.org is an organization that provides the programming and code framework for the bots so that they can be implemented using the API. Professor Vanselow is the faculty sponsoring this independent study.

* 1. **Design Concerns**

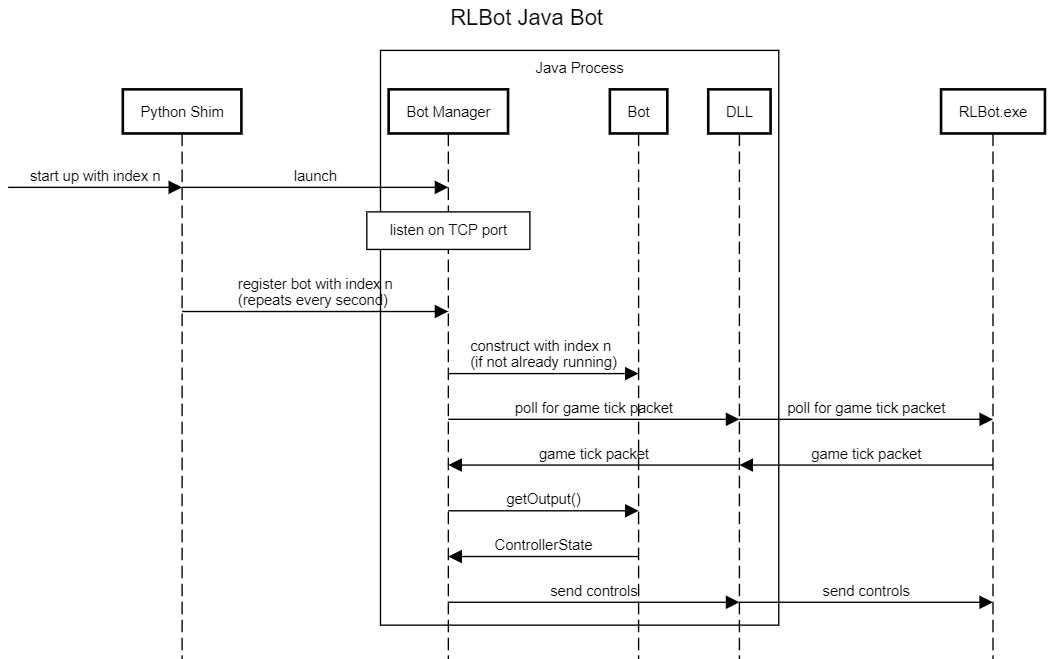
Overall I do not have any particular design concerns. The only thing that is relatively unknown is all the effects the official API may have on the RLBot framework. Previous to a few months ago, RLBot used injection to implement the bots into the game file. While the API is good news for the future of bot development, it is not completely 100 percent backwards compatible with some of the bots made through the original method. While this does not directly affect this project as it is all being done after the API was implemented, it does possibly affect the extent to which the bot can be programmed. Changes that the API made to the RLBot framework include:

* Boost pads and jump state cannot be manipulated.
* Boost respawn timers do not work within the GameTickPacket
* Quick Chat is not supported
* Soccer is the only supported mode
* Game data is capped at 60 hz as opposed to previously matching refresh rate
* The Psyonix API and code interface are closed source

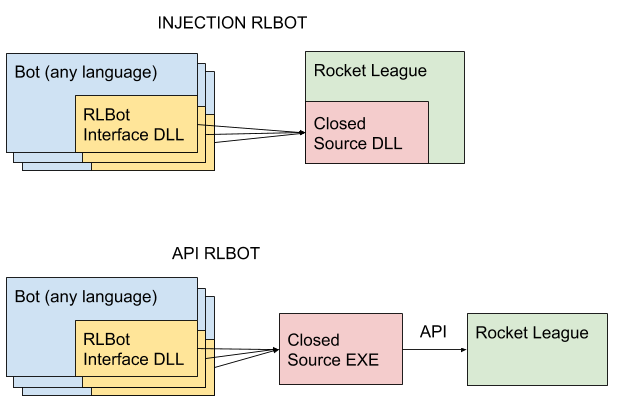
1. **Design Views**
   1. **RLBot Startup Sequence Architecture**



* 1. **RLBot Java Bot Architecture**



* 1. **API Architecture**



1. **Design Rationale**

The rationale for the design is pretty straight forward. The RLBot framework is provided as an open source bot development method. It is fully integrated with the Psyonix API so it was a no brainer to use that instead of attempting the previous method of injection that is no longer supported. The only real choice to be made was in regards to the language used. RLBot provides framework for bots in Scratch, Python, Java, C#, C++, and Rust. I chose to go with Java because it the language I am most familiar with and it provides full design functionality for the project that other languages may not provide.