Design Document

Section 1:

The Robo Sport board game will serve a variety of purposes. First and foremost, Robo Sport is a game, and as such, it should serve to entertain and challenge the human players who use it. The game will consist of clever computer AI's who can beat a human player, but who are not so clever that a skilled human player could not win against them. The game will also provide a platform for multiple human players to face-off against each other. The emphasis on strategy will cater to a user who is interested in both entertainment and mental stimulation. Beyond these general purposes, Robo Sport will solve several problems that a real life board game version would face.

Implementing the Robo Sport board game as a software system will allow for advanced additional functionality that would not be possible on a real life game board. One major advantage the computer game will have is "the fog of war", which hides aspects of the game from the current player's field of view. This creates the additional challenge of locating the other team's robots on the game board. Teams of robots must work together to "see" beyond each of their own visual ranges, allowing them to strategize together against the other team(s). Another major advantage that is offered by a software system is the ability to track and record game statistics. Without effort, the human player is kept informed of everything currently occurring in the game. Once a game is finished, statistics will be uploaded to a server, allowing players to track the progress of different robots.

The use of computer AI's and "the fog of war" are two aspects which will make this game challenging. Users can select from a number of different computer AI's, each of them challenging the user in a different way based on their implementation. This will allow the user to strategize against different kinds of game play, enabling them to improve their level of strategical thinking in many different scenarios. A user can also improve their strategy with or against other human players. Two to three users can play on a team against two to three teams controlled by computer AI's. Two, three, or six users can also join a game and play solely against one another. The variety of ways in which the teams or players can be organized will provide colourful gameplay that does not become boring or repetitive for those who wish to use it.

The game is designed for use by a variety of demographics. Firstly, the game will be limited to people who are somewhat familiar with computers. Secondly, it will target users who wish to improve their skills in strategic thinking. Thirdly, it will aim to engage people who play computer games for the purpose of entertainment. People looking to challenge themselves can play against a variety of computer AI's, or against other people. As a whole, the game is designed for a user who is looking to challenge themselves in a multitude of different ways, as well as to sit down and have fun.

Section 2:

The Architecture for the Robo Sport system is a combination of Model-View-Controller and Component-based Architecture.
In course of creating a complex software system, it is essential to find a common abstraction of the system on which all of its elements can be constructed. The architecture of our system will allow developers to understand the over-arching concepts and major functionality without the complexity of implementation.

Careful consideration of the different high-level components in the system led to the decision to combine two different architectures. The level of interaction between the system and the user make it necessary to have both a View component (for displaying game content to the user) and a Controller component (for gathering and reacting to user input). The Controller component is also responsible, to a degree, for the regulation of the computer AI's. The use of various robots, teams of robots, and the hexagonal game board also calls for a Model component, which will hold the basic structure of each of these objects. Due to the systems ability to access data via a network, the necessity to include additional components to the overall architecture becomes apparent. The system requires a Forth environment for the computer AI's, as well as the ability to access JSON files from a server. Model-View-Controller fails to esthetically and efficiently encapsulate the needs of these unique components. Adding extra components to accommodate network access and the AI functionality will serve to create an architecture which is both practical and system specific. The two architectures in combination will accommodate each of the systems distinct, high-level components, including the Forth environment for the AI's and abstract server communication.
Model-View-Controller plus Component based architecture allows us, the developers, to easily divide the pieces of the system, as well as the work required to construct them.