# TRON Preliminary Specification

# Introduction

The main goal of our project is to utilize Java to build a TRON simulated game upon the computer, allowing for networking as well, and will be done on a grid-based world. Our team will consist of two software development leads, and one project manager. The complete date of this project is set to be June 2nd, 2014.

# Structural Design

## Class Models

We will be having 4 main classes: a Display class, a Main Method class, a Player class, and a GUI class.

#### Display Class

Within this class, we will be outputting values like the score of the player, the time that has passed, the names of the Players, the Colors, and so on to enhance performance. Our main methods involve using a PaintComponent, and the different aspects of the KeyListener Interface.

### *TronMain*

This class will be storing our main methods and run method. We will also be outputting the film

### *TronPlayer*

This class will be using TronPlayer objects to actually play the game. Each player has bounds, and the moment one object hits the bound (either another player’s created wall or the outside boundaries), the game crashes, resets, and the other player gets a point in score. This is all stored within the class itself.

# Object-Oriented Design

For this portion, we will be mainly using TronPlayer objects in our design. Each TronPlayer will be able to access different methods to turn, as determined by the user. They will also have set colors, either Red or Green. The AITronPlayer will be controlled by the CPU, which makes decisions based on the User’s specific turns, and will be substituted by the other player once networking is implemented.

# Detailed Design

For the PaintComponent, we will be using a black background with Red and Green colored TronPlayers as the visuals. This will be done on an applet, using a panel. We will also be containing each direction within the class as well, which will be used for the AITronPlayer to change directions.

Our AITronPlayer has methods such as (getDistanceFromBoundary) and others to detect the specific times when it should turn or stay on the continuous path.

Our TronPlayer has a Boolean method which crashes the game the moment a boundary is hit and a turn method to change the direction. It also has a private color to set, along with initial direction.