

# **Product Requirements**

## **Reinforcement Learning Model for Games**

**Team:** Team WJNKCW

### **Problem Statement**

This project is centered around continuing an on-going project to implement reinforcement learning principles with a few basic games. Currently, two games (tic-tac-toe and connect four) are implemented. The main goals are to polish both of these games to make them more functional, to implement a third game (dots-and-boxes), and to get them running smoothly and properly on mobile.

The goal of this project, conceptually, is to demonstrate the usefulness of reinforcement learning by not only learning to play the games but to engage opponents optimally in a tournament setting. This project is intended to get future students interested in reinforcement learning models and machine learning in general. To that end, the target audience is upcoming students with varying levels of interest and experience. The product is intended to be engaging, fun to play, and clearly display the existence of reinforcement learning and its benefits. It should display elements of league playability.

### **System Requirements**

Required Systems/Tools:

- a. Unity 2019.3.x or later
  - i. Android or iOS Unity Development APK
- b. Python 2.7 or later
- c. IronPython 2.7 or later

Helpful Systems/Tools:

- a. Android Studio to debug Android APKs
- b. MacOSX to debug iOS APKs

The games and systems should run on both desktop and mobile devices.

### **Users Profile**

This system is intended for upcoming students of varying levels that might be interested in computer science, artificial intelligence, machine learning, or STEM fields in general. The students can be of various levels of familiarity and experience, though typically high school students or above, so the product should not be overly-complex, impenetrable, or relying on obscure information and jargon. The product should be easy to play, fun, and engaging in order to get students and players excited about machine learning's potential. Accessibility and ease of use are top priorities.

## List of Features

The product already has existing features from previous groups' works on it. This team's efforts will be focused on completing, polishing, and expanding on the existing code and assets. The features we will be adding and expanding on are as follows.

1. Dots-and-boxes game:
  - a. Implement a command line version of the game using Jupyter Notebook and Python script
  - b. Construct a functional Unity scene for the game using visual assets
  - c. Implement AI Q-tables for all levels of difficulty and verify their functionality
  - d. Implement league play mode
    - i. Betting options
    - ii. Various AI opponents
2. Implement on mobile - Games will currently not run on mobile due to file pathing issues and problems with the Android APK. These issues need to be fixed and the games need to be fully implemented on mobile
3. Polish
  - a. Improved visuals
  - b. Better UI
  - c. Sound effects
  - d. High player engagement

## Functional Requirements

No.	User Story	Description	Priority
R1	Play dots-and-boxes	The user should be able to easily play the dots-and-boxes game with its implementation. It must be fully implemented.	1
R2	Engage with dots-and-boxes AI	The user should be able to play against AI opponents with the reinforcement learning model. The AI should be engaging and challenging.	1
R3	Navigate menus for all games	The existing menus should be expanded to allow the user to move through and play all of the games. (Options include game, betting, opponents, difficulty, etc.)	2
R4	Dots-and-boxes tournaments	The user should be able to select various options and engage in tournaments on dots-and-boxes. This includes league playability.	2
R5	Dots-and-boxes betting	The user should be able to select various betting options to suit league play.	3
R6	Play on mobile	The user should be able to play all three games on mobile with full implementation and functionality.	1
R7	Visual engagement	The user should be engaged by the visuals and they should add to their enjoyment of the games and AI model.	3
R8	Clear expression of reinforcement learning model	The user should not just assume that the reinforcement learning model is present by names and descriptions of the product. There should be elements of the product that make it clear during play that the RL model is at work.	3

## Non-Functional Requirements

### *Usability*

1. High usability - The product should be easily usable by any level of student.
2. High accessibility - The product should not be difficult or a hassle to set up.
3. High engagement - The product should be fun for a student to play.

### *Reliability*

1. Solid reliability - The product should have high reliability to allow users to connect with the AI opponents.

### *Cross-Platform Compatibility*

1. Desktop - The product should be fully implemented and playable on desktop (Windows, Mac, Linux).
2. Mobile - The product should be fully implemented and playable on mobile devices (iOS, Android).

### Sponsor Requirements

I have read and approved of the material in this document.

Mark V. Albert

Print Name

Mark V. Albert

Signature

Feb 5, 2021

Date