Tom Lau, James Seelos, Lane Barnes

Final Scientific/Technical Report

CS 5060 Final Project

Project Manager:	Tom Lau
Project Manager:	James Seelos
Project Manager:	Lane Barnes

Public Executive Summary

The goal of this project is to discover how to have the most amount of fun playing Overwatch 2. We are all busy and don't have as much time as we would like to play the games we love. Our hope is to find the optimal role, and number of games you should play to have the most fun.

We will be utilizing simple algorithms such as Monte Carlo exploration and an optimal stopping algorithm. These will help the user decide what roles to play and when to stop playing the game. All the user has to do is log their happiness after each game which will provide the model with data to use.

Acknowledgements

Accomplishments and Objectives

We successfully created a program that will tell the user how they should play Overwatch. The program successfully tells them what role to play based on how happy they are playing said role, and the program will also tell the user when to stop playing the game at maximum happiness.

A number of tasks and milestones were laid out at the beginning of the project. The actual performance against the stated milestones is summarized here:

Table 1. Key Milestones and Deliverables.

Tasks	Milestones and Deliverables
Task 1: Monte Carlo 1.1 Implement Algorithm 1.2 Validate output	Q1: Milestone as stated in statement of project objectives. **Completed:** (4/21) Have Monte Carlo Simulation working with the correct output. Algorithm looks at happiness across each role played and will suggest the next role to play.
Task 2: Bayes Updating 1.1 Implement Algorithm 1.2 Validate output	Q1: Milestone as stated in statement of project objectives. **Incomplete:** (4/19) Have Bayes updating working with the correct output. Algorithm looks at expected happiness and the prior probability of that happiness and outputs a posterior probability. After trying a few different methods to get an updated value that we thought worked well with our project, we decided to remove the Bayes updating algorithm from the project. We concluded that it made more sense to just use an average of the happiness scores the users provided and that the Bayes updating was more bloat than anything.
Task 3: Early Stopping 2.1 Create data storage 2.2 Implement Algorithm 2.3 Validate output	Q1: Milestone as stated in statement of project objectives. **Completed:** (4/26) Have an early stopping algorithm working. Created a data structure to store user sessions and a way for the user to log their games. The algorithm will look at all previous sessions and find how many games it takes for the user to not be happy anymore. Once it sees where the user starts losing happiness, it will suggest quitting the game

Project Activities

The focus of our project is to analyze sessions of Overwatch and determine what role a user should play based on their happiness while playing using a Monte Carlo simulation. We also implemented an optimal stopping algorithm to determine when a player should stop playing the game. This project will take the stress away from a player and essentially make decisions for them. This should help player performance because this program will tell a player to stop before they go on a losing streak.

For our Monte Carlo section of the project, we utilize the explore/exploit concept to determine what role a user should play. Using an epsilon of .5, the algorithm will tell you what role to play based on the average happiness on that role. The optimal stopping algorithm will look at the user's happiness during previous sessions and will tell the user when to stop playing the game based on this. The optimal stopping algorithm starts working after at least 100 games have been played.

Project Outputs

A. Description of Code

There is a README file in the code that will give instructions how to run the code

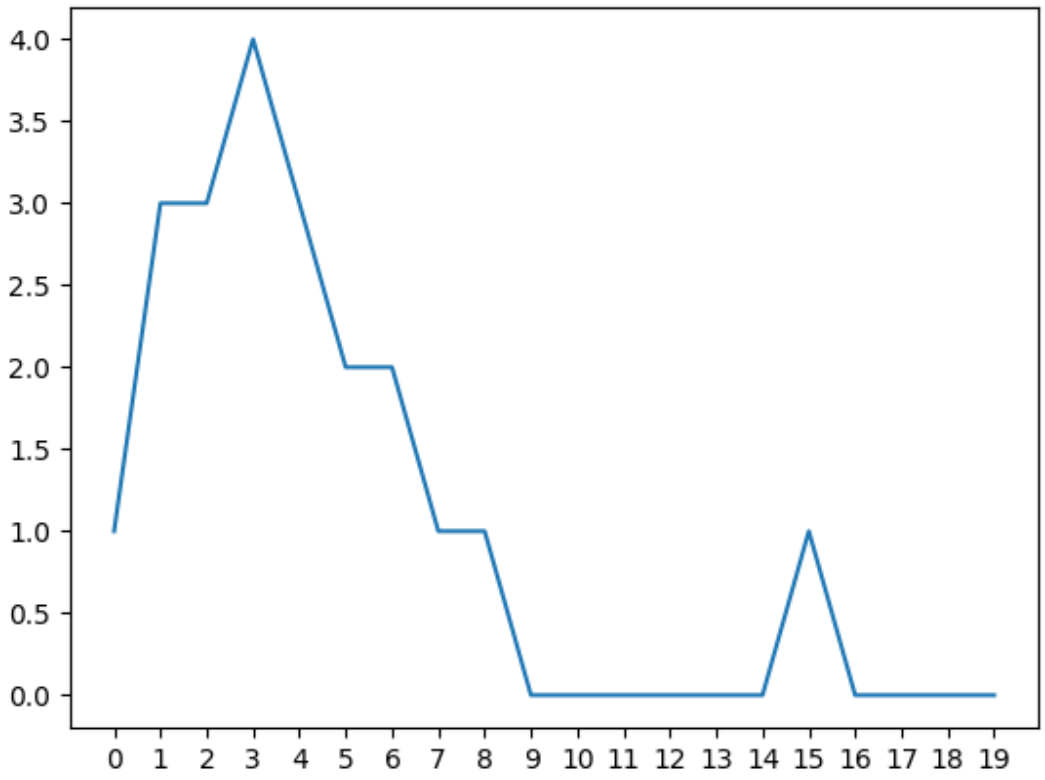
B. Expected Outcomes of Running Code

The expected outcome of the code is what role the user should play based on happiness, as well as the optimal stopping point to a session. The role the algorithm chooses is based on the expected happiness from previous sessions, but there is still a chance for exploration with an epsilon of .5. The optimal stopping algorithm will look at all previous sessions and determine at what point in the sessions the user is happy most often. For example, if the 6th game the user is happiest most often, then the program will tell the user to stop playing after the 6th game to end on a high note. One thing to be aware of is the optimal stopping algorithm will only go into effect after the user has filled the database with 100 played games.

C. Results

The results output the expected outcome that was previously described. We found that this program works best when there is a lot of data available for the algorithms to use. We predict that as the user plays and logs their games, the program will better understand their happiness levels as they play. We also would like to eventually add other game statistics into our algorithm to help determine how well a player is playing. This would add an interesting dynamic to determine when a player should stop playing the game. Additionally, there are better ways to change the epsilon and determine the optimal stopping that we could try to create a more polished experience.

Here is a graph of how the optimal stopping works:



As you can see, the 4th game for this user is most often where they are the happiest. In this case, the program would recommend to the user to stop playing after the 4th game.

D. Other Products (e.g. Databases, Physical Collections, Audio/Video, Software, Models, Educational Aids or Curricula, Equipment or Instruments)

Our program saves a file containing the gameplay history for that user as well as the happiness levels for each role. This is what our program reads from when it runs the algorithms.