Alpha Gomoku

Chengqi Dai (cd3046), Yiqian Wang (yw3225), Wenbo Song (ws2505), Zhongkai Sun (zs2341)

Fall 2018 - COMS W4156 Advance Software Engineering - Prof. Gail Kaiser

Abstract

In this project, we plan to implement a Gomoku game software, which has two modes: game between two people and rivalry between Human and Artifactual Intelligence. The client and user interface will be implemented using Python and the server side will be implemented using Java. To develop further, some reinforcement learning algorithms will be applied in the application to enhance the wisdom of AI chess player.

1. Background

In the March of 2016, AlphaGo beat Lee Sedol in a five-game Go match, which was first time a computer Go program defeated a 9-dan professional player without handicap. Science magazine chose this even as one of the greatest breakthroughs of year 2016. And it also changed people's perception toward AI from distrust to respect.

Inspired by AlphaGo, we decide to develop an Alpha Gomoku program which tries its best to beat human by cutting-edge AI technologies. In addition to man-machine mode, user can choose traditional mode to play with another person.

Making our game more attractive, different chess board designs can provide users with pleasure to have their own game skin. You can own all of these from our item store using game currency. And game coins come from each time of winning or daily check-in.

2. Proposal

The blueprints of our project are based on these user stories:

1. As a user, I want win after repeated failures, my conditions of satisfaction are determined by difficulty of game. If I'm playing with AI, I hope to choose the difficulty of AI. I hope there are three levels of difficulty, Easy, Medium, and Hard, to select from. And if I'm playing with a human player, I hope I can see the player's win rate and total rank among all players, so I can determine whether challenge myself and choose a high rank player or just play for fun and choose a low rank player to play with me.

2. As a user, I want to have a delighted background music while I'm playing, so my conditions of satisfaction are determined by whether I'm into that song by choosing the like and unlike button. I can change the background music or turn off the music and sound effects if I don't like them.

3. As a user, I want to know who is online and I prefer to play with my friends. My conditions of satisfaction are judged by how many people online and whether I can invite my friend to join me.

4. As a user, I want to observe pretty user interface, so there should be a plenty of choices in chessboard appearance.

3. Goal

First iteration:

Implement a basic version of gomoku game. Users can sign up or login by username and password. Users can play local gomoku game with their friends. Game results are uploaded to server and users can browse their game history.

At server side, we will use MongoDB as database to store the users' information and game history.

Second Iteration:

Implement a game lobby for more users to play online gomoku. A player can check in a room for online game. These rooms are listed in the broad of game lobby. Other players can join a room if there is only one person in it. They can also watch the game if the room already has two players inside.

And we will design a rank system using rank score to present the skill of players. The rank score will increase if you win and decrease if lose. The amount of score to change will be determined by the rank of your enemy.

To make our service faster, we will store the room information using Redis.

Final iteration:

Implement an Alpha Gomoku player to offer users offline man-machine game. Optimize the entire system and add some features such as changeable chessboard's skin, store's skin and background music.

We plan to implement the AI player using reinforcement learning and we will train multiple AI players using same model but different training time to achieve different difficulties.

4. Potential Problems

First of our potential problems in building Gomoku game is about the user interface. We will learn how to create a game shop with changeable color, how to design different pieces of chessboard and shows them on other players' computers. Creating a real-time updated game lobby with changing user number is also a problem.

Another potential problem is the connection between clients and server. Because sometimes we may miss data from sockets. We need to verify the data and make sure it is transmitted accurately.

We also have a potential problem about the implement of artificial intelligence algorithms such as Alpha-Beta pruning, Checkmate and Iterative-Deepening Search Algorithm, in order to make Alpha Gomoku player as smart as possible.

5. Technology

Client Side: We use python for user interface design and implement machine learning algorithm for AI player. The trained AI models will not store at the server but client program to reduce the pressure of our server.

Server: Java server will be set up to enable online chess play. The server will open multiple sockets I/O waiting for clients to connect. Use MongoDB to store users' login information and profiles. Use Redis to store information of activate game room.

Deployment: Our server will run locally and any client can join it at any time. Our code will be displayed in Github.

API: We call API in several places. For music play, we use SoundCloud API. For login authentication. We can use Facebook Login for Apps. Also, we want to add a chat board which allows users chat in real-time during the game.