

Answers:

1. Maristella's Gomoku Solver

2. Maristella Jho, 149158

3. I decided to work on this project alone, so I think I deserve all the credits for this assignment. I first came up with the idea of this assignment: Gomoku Solver and decided to make the game playable with mouse using Pygame library.

4. My github repo for this assignment is: <https://github.com/hyunseo6579/Gomoku-Solver> And I referenced my old Pygame project as a starting point to remind myself how to create the window and handle events with Pygame: <https://github.com/hyunseo6579/Graduate-Nya>

5. Gomoku uses the same equipment as Go, but it is a much simpler game where the players take turn to place a stone every turn, and try to make a connected line of 5 with the stones either horizontally, vertically, or diagonally. The original game uses 15 x 15 board, but for this version, it uses 9 x 9 board instead. <http://www.opengames.com.ar/en/rules/Gomoku>

6. My original goal for this assignment was to make a tutorial to help the player learn the game, but I realized that the game's rules are simple enough that it would be more fun to just have a solver to play against. ↯So, my next goal was to make a solver, or really an AI to play Gomoku with, and in the end I was able to somewhat achieve that goal. The most satisfying part of this assignment was actually being able to make a playable Gomoku game as I have imagined in my head. It wasn't a very hard game to make, but I think the part of the reason I love being in Computer Science is because I can't imagine myself to be able to create a program solely on my own, but when I try and put in the work to make them, it gets finished and since I am the only one working on the project, I can make it to work exactly as I imagine it to be in my head. The feeling that I get from achieving that goal makes me really enjoy programming. Thank you for making such a fun experience as part of an assignment! Although, the disappointing part of the project is that my solver is not perfect, and it works with a very simple algorithm hence is easily beatable. Right now, it only starts playing from the top left corner to make a horizontal 5 when it starts, which is not how people usually play Gomoku. Therefore, if I continue to work on this project later, I think I will research for better algorithm, and make it play more like how people play the game. Also, the code has quite a bit of repetitive parts that I want to see if I can shrink into more efficient code if I had the time.

7. After making a friend play over 10 games (which ended with 4:6 win of my friend) with my solver, I came to realize few strength and weaknesses of my solver: a. It always starts the play from top left corner, leaving the player to take up most of the center clump space. b. It can defend perfectly until it hits the point where defense in 1 move is impossible due to the clump made by the player that gives him an advantage of many winning moves. c. It can however throw the player off if he doesn't take advantage of the center space, as it considers every possible winning move which may not be noticeable to the player. (such as 3 stones in a row left unnoticed with open ends) Thus, making the game speed run could put the player at a disadvantage to easily miss an opening.

Diary:

The reason why I picked this project is because I personally love playing Gomoku and is familiar with the rules of Gomoku.

Monday 11/02:

- Decided on a game to do the assignment on and decided to go with either a tutorial or a solver.

Wednesday 11/04:

- Decided to go with a solver and started making plans in my head on how to make the game.

Saturday 11/07:

- Thought that using a pygame will make game play easier as players can click to place a stone rather than enter a point in a large grid.
- Asked on forum if I can use pygame for the assignment.

Total hours spent from Nov 2nd to Nov 8th: 2.5 hours

Tuesday 11/10:

- Went through my previous project to get an idea on how to start off a pygame project
- Built a game window that closes when “x” button is clicked

Wednesday 11/11:

- Drew assets for game title page (looked messy so it got scrapped)

Thursday 11/12:

- Recreated title page assets and created 9x9 board assets and the player’s stone assets
- Took note of measurements of the board’s position in pixels and measured the stones’ size in pixels

Friday 11/13:

- Wrote basic code to handle mouse clicking events
- Created classes to be included in the game with comments on their possible functionality

Sunday 11/15:

- Wrote the code to handle clicking play to take player to game page
- Wrote more code to handle player’s click to place a black stone and wait for AI’s move

Total hours spent (Nov 9th to Nov 15th): 10.5 hours

Monday 11/16:

- Tested/debugged player’s clicks as it would place a stone on a location where play button is clicked on a title page
- Started writing solver code where it first handled to check the progress by iterating through the 2D array of the board’s status and see if anyone has won the game yet
- Then wrote more code that decides the next move if the game isn’t over where it would first defend, then look for a move that can become a line of 5

Tuesday 11/17:

- Tested the functionality of the progress checker and move method

- Move prioritized defending over winning the game, so added another method that makes the AI prioritize winning when it has a line of 4 and can put the 5th stone at the current state.
- Tested gameplay and found out the AI starts from the top left corner when it makes its own move which is not ideal, but decided I have no time to change the algorithm at this point (have 3 more assignments due next week that I haven't started on)
- Wrote the final code and made little assets that handles the event where either the player or the AI has won/ is a draw
- Wrote the report to hand in with the assignment
- Is about to go test the game on Linux and write a readme instruction to launch this game

Total hours spent (11/16th and 11/17th): little over 10 hours