

Mini Project: Guess who?

Remember the children's game *Guess Who?*

You select a character, and your opponent tries to guess who it is using yes or no questions only. They eliminate characters, until only one remains.



Now, think about coding this. What would you need to consider?

MVP

Your task is to build the backend of some very basic *Guess Who?* board game logic. There will be no frontend user interface. We'll build our application through test driven development to gain confidence it works correctly. Do not merge to main, work on a branch and open a PR. Your work will be reviewed on your PR, exactly how it will be when you are on client site.

Start by planning. Think about:

- How will you make characters?
- How would the computer randomly select a character at the start of the game? How will this be kept secret from the player?
- What questions can be asked based off the characters?
- What will happen because of the answers to those questions?
- How will you represent the board?
- Will the board state change throughout the game?
- How will the game end?
- What is the purpose of a ReadMe and a .gitignore file? Should you include either?

Once considered, write some very simple logic. At this point, don't need to think about it being playable, just the logic behind the game and the decision making that could occur.

Try to use as many of the tools we have learnt since starting the Academy. You may need to do a bit of additional research. Don't be afraid to try things out, even if they don't work as you expected, you're still learning!

Remember some of the key lessons we've learnt about developing something:

- Plan the game first, do this in whatever way works best for you.
- Start small and aim for TDD. Write a test and then just enough code to make that test pass. Refactor after.
- Test frequently! Never move too far away from a working solution. Remember to test not just your happy paths, but also that your code can fail at the correct points. Have you considered all edge cases?
- Work on a branch and commit frequently to that branch, don't push directly to main.
- Make sure your commit message is clear. Present tense. If you are needing to use 'and' in the message you probably should have committed sooner!
- Writing code is collaborative, talk to your peers, discuss ideas.
- Have fun with it! Be as silly and creative as you want to be!

Extension 1

Make a new branch here. Branch off MVP (reach out to me if you are unsure about this).

Research how to interact with the command line and turn your logic into a game which can be played as a 1 player game.

Extension 2

Make a new branch here. Branch off Extension1.

Add the functionality that allows the player to access a certain number of hints. Be creative as to what these hints can do.

For example, this hint could remove a person (who isn't the computers randomly selected person) from the remaining characters on the board and display the updated characters left.

Extension 3

Make a new branch here. Branch off Extension2.

In the physical board game, while your opponent tries to guess your character, you are trying to guess theirs. You take it in turns.

Make your game work for two players. Allow players to take turns. Don't worry about making it interact online or hiding the information from the other player. Let's keep this as simple as possible and see what you come up with!!

Even if you don't get round to coding it you can still have a go at the planning/research!