How to start the game

Go into Othello_graphical_interface.py file and run it or type in 'python3' Othello_game/Othello_graphical_interface.py' command into the Linux terminal (you need to be in the Othello folder otherwise some parts will not be able to load correctly) (you also need to have pygame library installed on your computer).

How to play

Start:

First, you have to select one of three options:

> - player vs player
> - player vs computer
> - computer vs computer

Next, you have to select the size of the board. By default, it is set to 8x8, and its size can be from 8 to 30 and it also has to be even. You can change the size of the board on x dimension using left and right arrows on your keyboard and y dimensions using up and down arrows.

After that, you need to press start, then the game will begin. If you don't choose the game type you won't be able to start the game.

The aim of the game is to have more discs of your colour than your opponent. The game starts with two black and two white pieces in the centre. Blacks play as first, you can only play your pieces in less saturated spaces. When you play your pieces every opponent's piece between the one you played and your other one in a diagonal, vertical and horizontal line turns your colour. The game ends when the whole board is filled or three are no spaces where you can play. Wins player with the most pieces.

The last step is to have fun!

Aim of the project

The aim of the project was to recreate the Othello board game using python and add a playing algorithm to it. Adding a graphical interface was optional. The most important thing to create was an algorithm as good as that is possible.

What is Othello game?

The Othello game created by me consists of 8 files placed in the Othello_game subfolder of the Othello folder. Those files are:

- Othello_consts.py
- Othello_space.py
- Othello_board.py
- Othello_bot.py
- Othello_graphical_interface.py
- Othello_io.py
- Othello_interface.py
- Othello_main.py

Created graphical interface makes use of the pygame library, so that library needs to be installed for the game to work.

Othello_consts

This file contains four values, each representing the different value of space. In this file, there is also swap_colour function which changes first_colour to second_colour and the other way around.

Othello_space

This file contains the Space class which represents one space on the Othello playing board. This class allows to change the values of spaces on the board without the need to have the whole board. The most important function of this class is find_space_around function. It allows to minimise calculations needed to find spaces where player can put his piece.

Othello_board

This file contains Board class which represents Othello playing board. Board is initialized with a tuple representing the size of the board. Size ranges from 8 to 30 and can only be even (design arrangements). The most important function in Board class is find_plays function. It changes value of spaces where the player can put his piece to possible_value and returns a set of information for each space informing which pieces need to be turned around if a piece is placed in a given space.

Othello_bot

This file contains class BOT which is the computer algorithm that is responsible for predicting moves and choosing where to play the piece to maximise score. Min-Max algorithm with alpha-beta pruning is used for calculating the best possible score.

Othello_graphical_interface

This whole file is responsible for creating a graphical user interface using the pygame library. We can separate three main functions. One is for starting interface, which gives the user choice of board size and game type, second one is for playing the game, the third is for displaying the score after the finished game. First and third ones run at 20 fps to minimise computers resources usage. Second function runs at 40 fps to minimise delay and the possibility of not detecting clicks. (Lack of game size changing is a design arrangement because of bugs in initial testing.)

Othello_text_ui

This file is for the text user interface. Because of the fact that a graphical interface was created, this file serves as a showcase of progress made during the making of the Othello game. The user interface is created using tabulate library as well as saved to Othello_board.txt file. This file can also be used for testing computing algorithms because it uses less of the computer's resources.

Othello_interface

This file similarly to Othello_main is is for the text user interface and is a showcase of progress made during the making of the Othello game. It has functions responsible for communication with users and displaying the playing board.

Othello_io

This file similarly to Othello_main and Othello_interface is is for text user interface and is a showcase of progress made during the making of the Othello game. This file has functions responsible for writing board and results to file.

Reflections

I completed all the tasks given to me at the start of the project and also added a graphical interface. I think that my algorithm for playing the game is good and it is capable of constantly beating me, but it can definitely be upgraded by changing weights and also giving different weights for different colours (it is possible to do that in my code but it is really hard to find the best combination of weights and results are sometimes confusing). I also did not manage to give the user the ability to change screen size, because during my test I ran into some problems after the end of the game in some cases application just shut down. That is also the reason why after choosing a non-square board window stays square. It is definitely possible but it was not worth the time to do it and I decided to focus on more important things in this project. I also think that the graphical interface can be made prettier by making use of graphical software like Photoshop, but I also think that it was not the aim of the project so I did not focus on it too much. During the making of this project, I learned many things from which most valuable is probably the min-max algorithm because is somewhat of an introduction to Artificial Intelligence. I had some problems with pytest and because of that I had to make a new repository and I couldn't find a way to merge it with the new one (old repository - I put the link to the repository that I was using before issues with pytest to show all my commits). Giving players the possibility to play through the internet is one more thing that I did not do but at the start of the project, it was established that it is too hard of a task for the first semester project. I think that I will come back in the future to this project to implement this function.