

# Reversi Game in Python

---

## Introduction

This project was developed as a part of the coursework for the module "Introduction to Programming I" offered by the University of Mindelo (UM). This is a simple implementation of the reversi game written in python. Developed using all that we studied thus far and researched.

## The Reversi game

This Python implementation of the classic board game Reversi (also known as Othello). The game is played on an 8x8 grid, with two players taking turns placing black and white pieces. The goal of the game is to have more of your color pieces showing when the game ends.

## How to Play

To play the game, you will need to:

1. Download and install Python 3.
2. Install the Pygame library.
3. Clone this repository.
4. Open the `project.py` file in a text editor.
5. Run the `project.py` file.

The game will start with a menu. You can play against another human player. And you can also change the game theme in the options menu.

Once you have chosen your options, the game will start. You will take turns placing black and white pieces on the board. You can place a piece on a square if it is adjacent to an opponent's piece and there is a clear path to the other side of the board.

When you place a piece, any of your opponent's pieces that are flanked by your pieces will be flipped to your color. The game ends when all of the squares on the board are filled. The player with the most pieces showing wins the game.

## Code Overview

The code for this game is organized into several modules:

- `project.py`: This is the main module of the game. It contains the main game loop and calls the other modules to render the game.
- `board.py`: This module contains the code for the game board. It includes methods for creating the board and rendering the board.
- `data.py`: This module contains the code for loading the game pieces.
- `menu.py`: This module contains the code for the game menu.
- `options.py`: This module contains the code for the game options.
- `logic.py`: This module contains all the game logic, and checking for the valid moves.

## Step-by-Step Explanation of the Code

The following is a step-by-step explanation of the code in the `main.py` file:

1. Import the necessary modules.
2. Define the main function.
3. Initialize Pygame and set the title of the screen.
4. Set the window size and the clock (fps) of the game.
5. Load the game pieces.
6. Call the main menu
7. Game loop - while the program is running:
8. Handle events (keyboard presses, mouse clicks).
9. Load the board and the pieces on it, and display them on the screen.
10. Check the moves of the player
11. If won display the message that the user won.

## In Conclusion

This project provides an example of how to create a simple two-player chess game using Pygame, we hope you like the idea and feel welcome to contribute to our game.

## For the future of our game

First of all, we need to make the game to function right, this is just a beginning for us, we do plan to make the game functional. We can add more features such as sound effects when making a move or clicking buttons. We have the idea of adding customizable pieces and more themes for the board.