This manual contains information about how to install the project and its dependencies, how to run the CLI and Web version of the game provided and how to use the code for potential other implementations of the game.

System Requirements:

- Python (version 3.10 or above)
- Flask (external Python library can be installed via the pip command 'pip install flask')
- The project source code

Installation:

1. Download the project

To acquire the project on your device, the project source code and files can be downloaded in various ways:

- Download as a ZIP file from the main repository page (https://github.com/SpoonFish/Reversi-Project)
- Clone using the git command 'git clone https://github.com/SpoonFish/Reversi-Project'

2. Install dependencies

- If you do not already have Flask installed, run 'pip install flask' on the python installation you are using for the project

3. Project folder structure

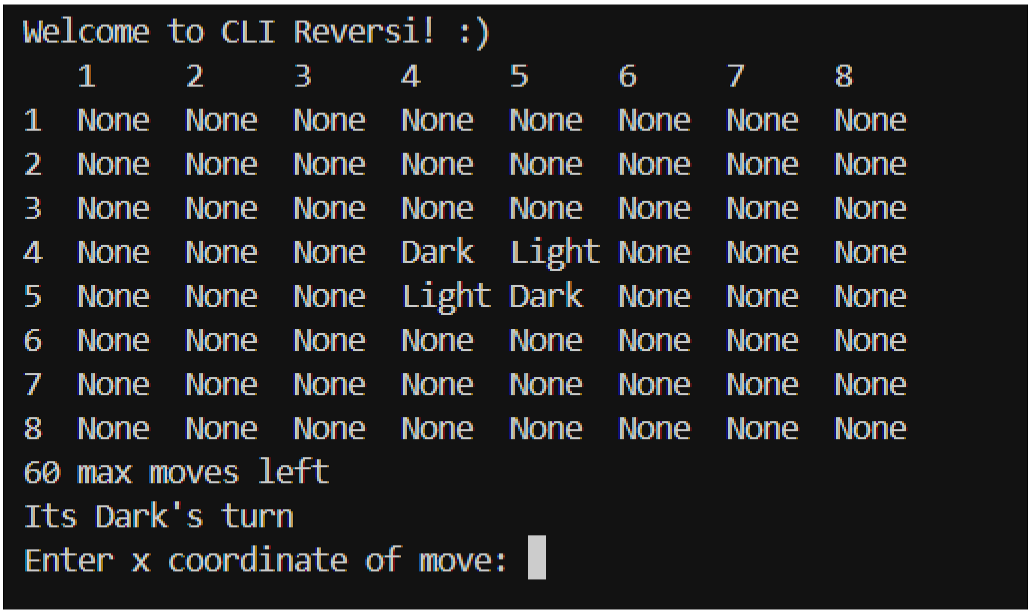
Ensure the project has the same structure as defined below so that the game runs correctly

```
Reveri Project/
  Stage1/
    components.py
    game_engine.py
  Stage2/
    components.py
    flask_game_engine.py
    templates/
      index.html
  Stage3/
    components.py
    flask_game_engine.py
    templates/
      index.html
```

4. Running the applications

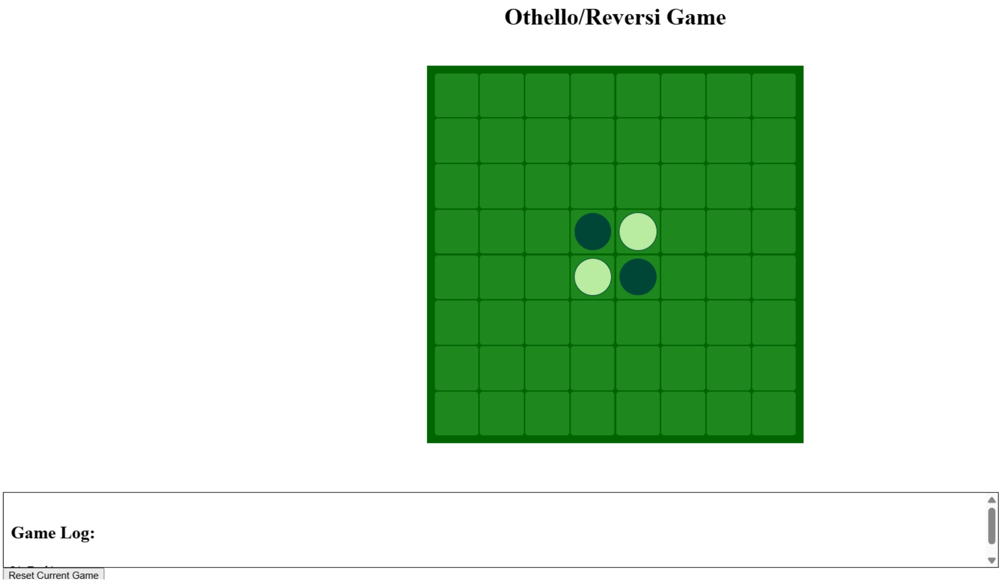
From the project root directory "Reversi Project" run the following commands to play the game.

For the CLI version run 'python Stage1\game_engine.py'. You should see this entered to the console:



```
Welcome to CLI Reversi! :)
    1     2     3     4     5     6     7     8
1  None  None  None  None  None  None  None  None
2  None  None  None  None  None  None  None  None
3  None  None  None  None  None  None  None  None
4  None  None  None  Dark  Light None  None  None
5  None  None  None  Light Dark  None  None  None
6  None  None  None  None  None  None  None  None
7  None  None  None  None  None  None  None  None
8  None  None  None  None  None  None  None  None
60 max moves left
Its Dark's turn
Enter x coordinate of move: 
```

For the completed Web version run 'python Stage3\flask_game_engine.py'. Then open the web page locally hosted at 'http://127.0.0.1:5000/'. It should initially look like this:



Othello/Reversi Game

Game Log:

5. Using the interface

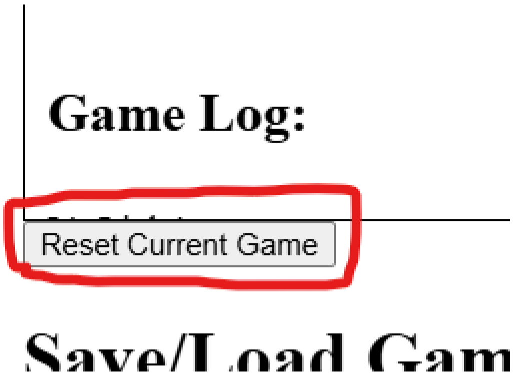
The green board displays where counters are on the board. Dark green counters belong to the player indicated as ‘Dark’ in the game log. This player will go first and is also the human player you control when playing against the AI. Light green counters belong to the player indicated as ‘Light’ in the game log.

The game log will show messages about events happening in the game. It will show whose turn it currently is. It will also show if a move is invalid and what valid moves have been made. When the game is over, the winner will be displayed in the log.

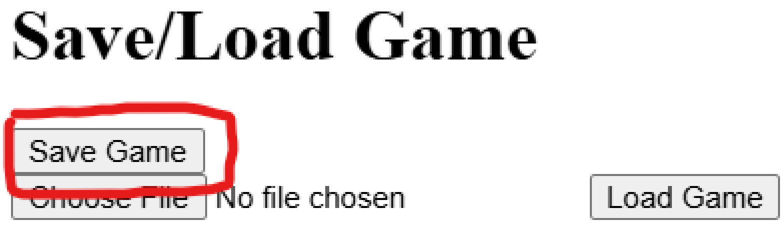
To make a move in the game, click an empty cell on the board where you want to place your counter. The move must be legal (it must outflank at least one of the other player's counters). The result of the move will be instantly displayed on the board. The colour of the counter placed depends on whose turn it is.

Turns are automatically passed when a player makes a legal move or if a player cannot make any legal moves on their turn. The game ends when the board is full and there is no space for more counters or if both players find themself in a position where they cannot make any legal moves.

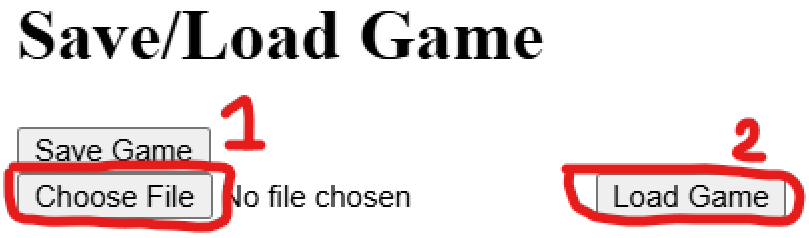
To reset the game, press the 'Reset Current Game' button. This will return the board to its initial state and Dark will begin the game.



To save the current game, press the 'Save Game' button. This will download a file that can be used to load the game at a later date.

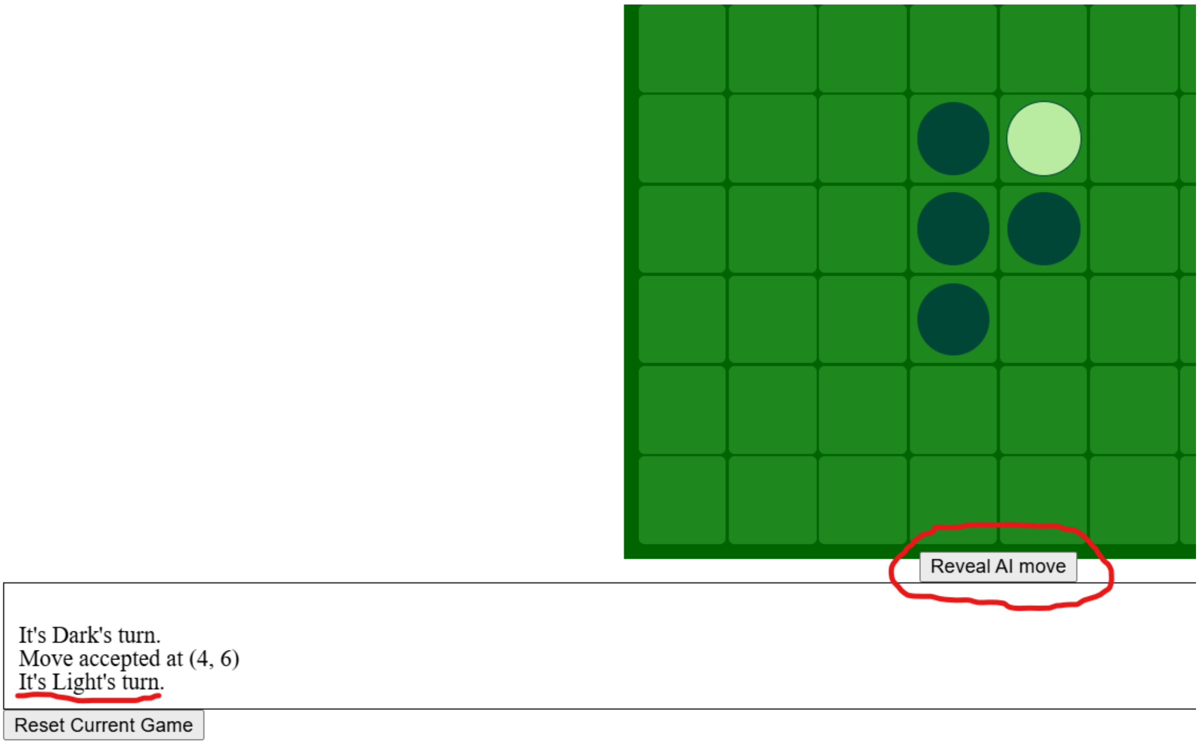


To load a game from a saved file, press 'Choose File' to select which file you want to use as the game save file to load. Then, press 'Load Game'. The board should be in the same state as it was saved and the turn of the player is reserved.



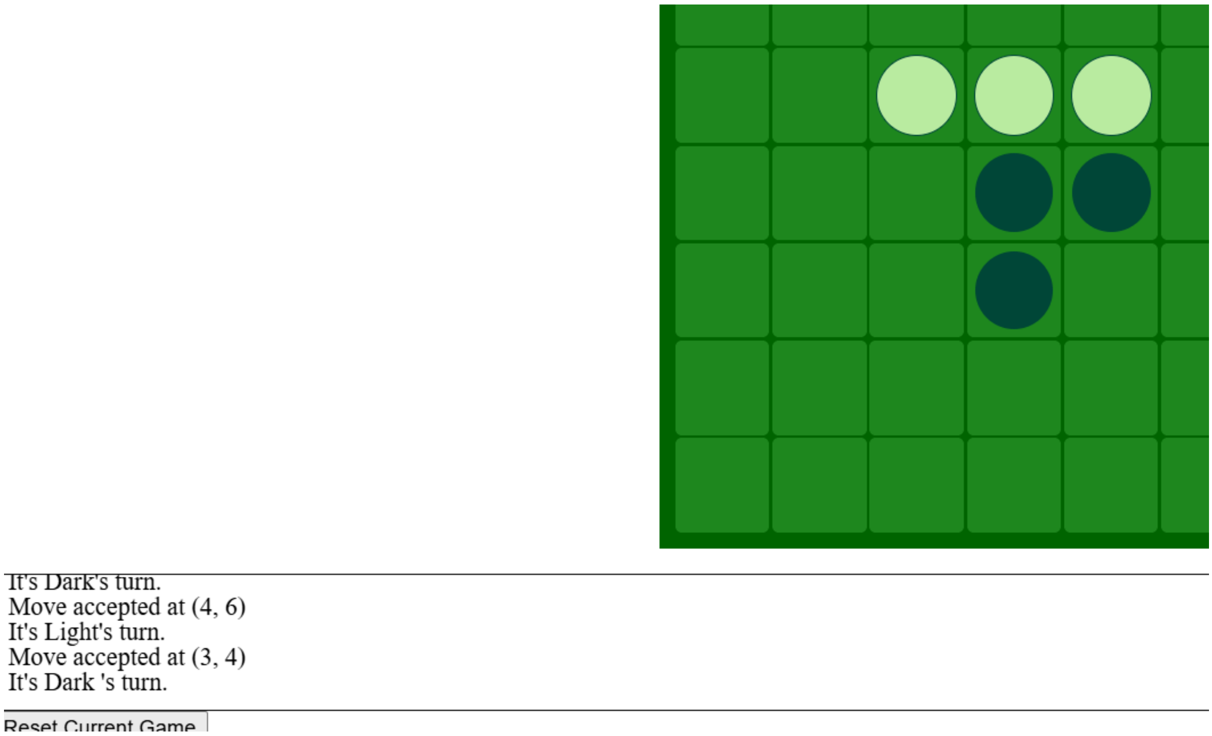
To play against the AI player, make your moves only as the Dark player. When it is Light's turn, a button 'Reveal AI move' will appear. Click this button to make the AI make the move for Light allowing you to play against the AI player.

After your move:



```
It's Dark's turn.
Move accepted at (4, 6)
It's Light's turn.
```

After clicking the AI move button:



```
It's Dark's turn.
Move accepted at (4, 6)
It's Light's turn.
Move accepted at (3, 4)
It's Dark 's turn.
```

6. Using the API

Functions:

The functions in the project can be used in another implementation of Reversi. components.py contains some functions that can be used to debug, setup or check legal moves. flask_game_engine.py contains other functions needed for checking if legal moves are available, executing moves on the board, passing turns, calculating who wins the game and AI move calculation. As long as data inputted into the functions is of the correct format, they can be used for the logic of the game Reversi in any other classic implementation. The ‘board’ must be a list of 8 lists containing 8 values for each cell (“None “, “Dark “ or “Light”). The current player’s turn is represented by either “Dark “ or “Light”. The coordinates must be integers between 1 and 8.

Web routes:
GET /

- Loads the main page for the game

GET /move?x=<num>&y=<num>

- Attempts to make a move at column x and row y on the board. Returns JSON data with the status of the response, the updated state of the board and a message about the move made.
- Example response:

```
{
  "status": "success",
  "player": "Light",
  “Message”: False,
  "board": [[“None “,”None “,”None “,”None “,”None
“,”None “,”None “,”None “],[“None “,”None “,”None “,”None
“,”None “,”None “,”None “,”None “],[“None “,”None “,”None
“,”Dark “,”Light“,”None “,”None “,”None “],[“None “,”None
“,”None “,”Dark “,”Dark “,”None “,”None “,”None “],[“None
“,”None “,”None “,”Dark “,”None “,”None “,”None “,”None
“],[“None “,”None “,”None “,”None “,”None “,”None “,”None
“,”None “],[“None “,”None “,”None “,”None “,”None “,”None
“,”None “,”None “],[“None “,”None “,”None “,”None “,”None
“,”None “,”None “,”None “]
}
```

GET /save

- Downloads a JSON of the board, whose turn it was, and whether the game was won to the user’s device.

GET /ai_move

- Calculates an AI move depending on the current state of the board. Returns a response with a status and the coordinates of the calculated move to be used in /move.
- Example response:

```
{
  "status": "success",
  "x": 5,
  "y": 3,
                    }
```

POST /load

- Extracts game data from the provided save file to set the board to the state in the save file.

POST /reset

- Resets the game to its initial state and reloads the main page.

7. Limitations

For the web version, only an 8x8 board can be used. The CLI version can use any size board of an even integer >2 with a recommended enforced maximum of 16x16.