Modifications:
These are my modifications to the base code (Section 6):

Firstly, I initialised the constants BLUE, RED, LIGHT_RED, YELLOW and BLACK as colours for the different items on the board and the constants HEIGHT and WIDTH to represent the respective dimensions of the game window surface. I also imported the pygame and the math module and initialised pygame. Installing pygame (if not already installed) is described later.

Then I created a small segment of code to allow the user to input an integer between 1 and 10 (inclusive) to create the board size and subsequently initialise the FourInARow class. This board size is then used to determine the appropriate token radius/token size to maintain a constant window size no matter the board size.

Next the pygame 'window' surface is created and a caption "Connect Four" is set for the top of the window before the play() method is called.

I set a clock with a constant fps (60) so that the while loop runs at the same rate no matter the computer it's run on (i.e., loops 60 times a second or if it cannot do 60, as fast as possible)

I also changed the display function. Instead of printing x's and o's and a text-based board, there are red and yellow representing the AI and human player and a blue background to represent the gameboard. I added a call to a display text method in the gameboard class that displays hints for controls and a score counter. This is also the function that updates the surface.

I set a new parameter called 'run' to handle instances where the player closes the window during the game (initialised at True and if the user quits is set to False, ending the while loop and closing the game).

The modified while loop tracks events instead of checking for valid inputs.

- Notices when the player **closes the window** and quits the game.
- Notices **mouse movement** and checks the nearest valid column, putting a shadow/hollow token there to simulate where the disc would go if placed.
- Notices when the **mouse button** is clicked, checks if the column clicked is valid and places a disc in said column if so.
- Notices if the player has pressed the **"r" key** during the game and if so, resets the board (note: this may take a few seconds)

Finally, I added an end game message that shows who won or if it was a draw when all slots have been filled. A wait timer is set to wait 6 seconds after the game finishes so the player can read the end message before closing the window.

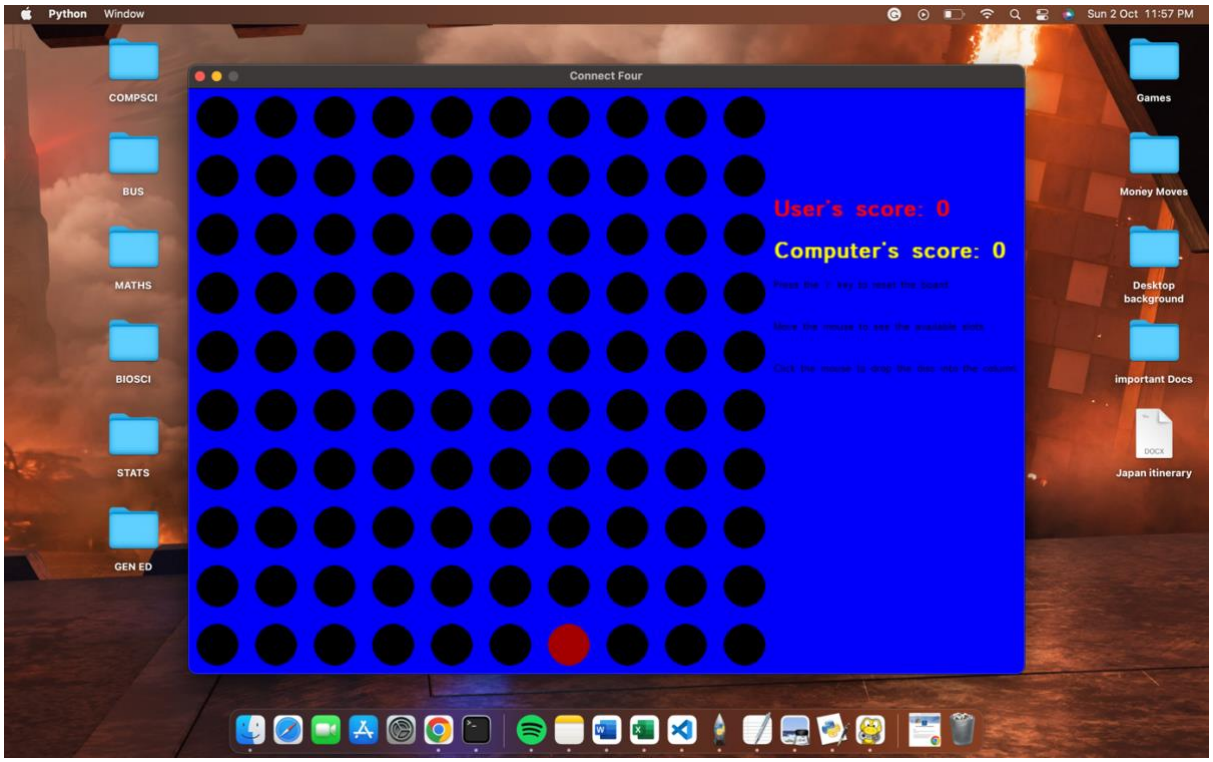
How to install pygame:
In order to install pygame enter one of the following into the command line:

- pip install pygame
- pip3 install pygame
- python -m pip install pygame
- python3 -m pip install pygame

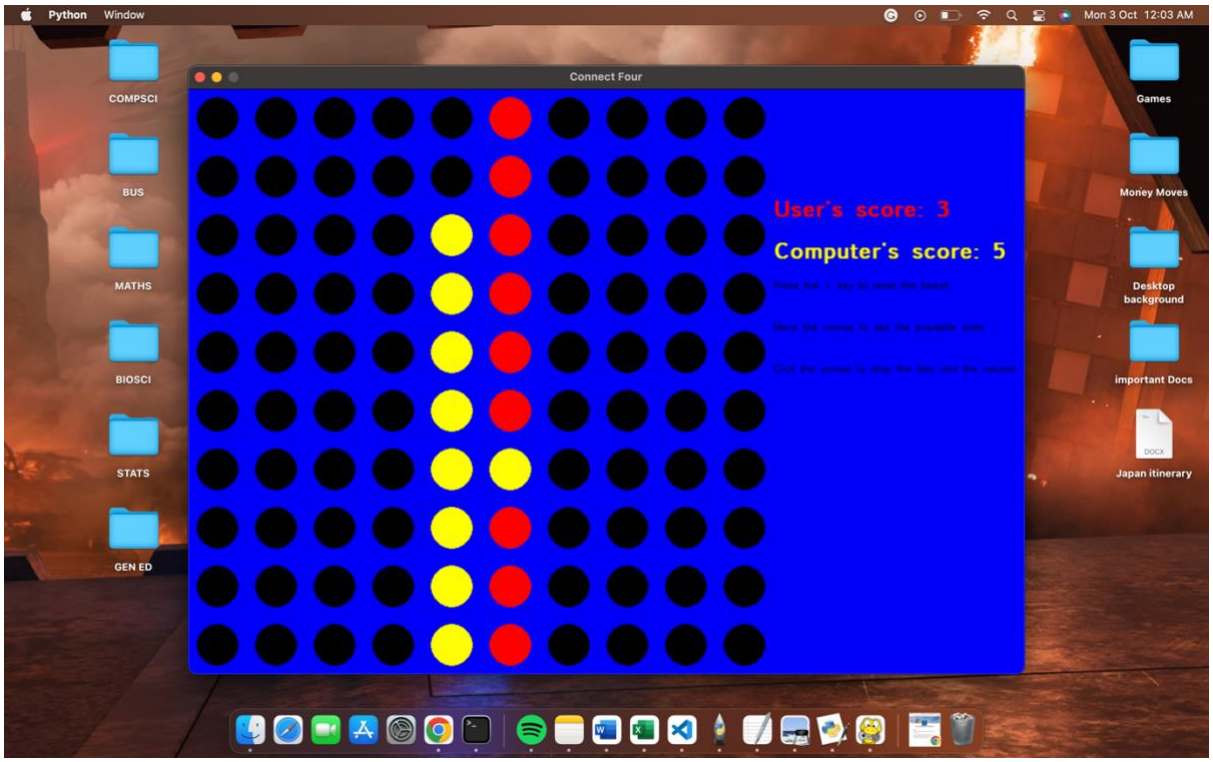
Testcases:

```
experimenting.py - /Users/kunalbhaskar/Desktop/COMPSCI/experimenting.py (3.10.2)
Python 3.10.2 (v3.10.2:a58ebcc701, Jan 13 2022, 14:50:16) [Clang 13.0.0 (clang-1300.0.29.30)] on d
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: /Users/kunalbhaskar/
pygame 2.1.2 (SDL 2.0.18, Python 3.10.2)
Hello from the pygame community. https://www.pygame.org/contribute.html
Please enter an integer board size between 1 and 10 inclusive: apple
Value given is not an integer
Please enter an integer board size between 1 and 10 inclusive: 20
Integer out of range
Please enter an integer board size between 1 and 10 inclusive: -10
Integer out of range
Please enter an integer board size between 1 and 10 inclusive: 0
Integer out of range
Please enter an integer board size between 1 and 10 inclusive: 4.5
Value given is not an integer
Please enter an integer board size between 1 and 10 inclusive: 10
```

As seen, if the input is a string or float (e.g., apple or 4.5) a ValueError is raised and the message “Value given is not an integer” is printed. If the integer given is out of range (i.e., greater than 10 or less than 1) an exception is raised and the message “Integer out of range” is printed. If valid integer is input, then that value is assigned to size and a gameboard with that size is generated.



If the player attempts to click outside the window, there is no effect, similarly clicking on the right side score panel will have no effect (as a column out of range is ignored).



If the player attempts to add a disc into an already full column or hovers the mouse over that column, a shadow disc is not shown and no disc will be placed. Finally, if the user attempts to close the window at any point during the game, it is closed without error.