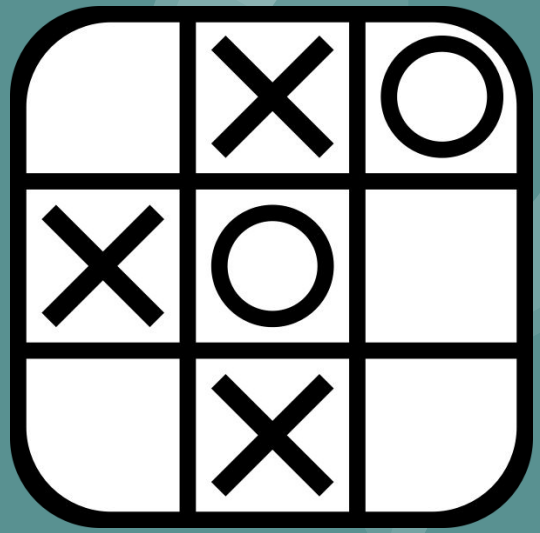
Tic Tac Toe
T1A3

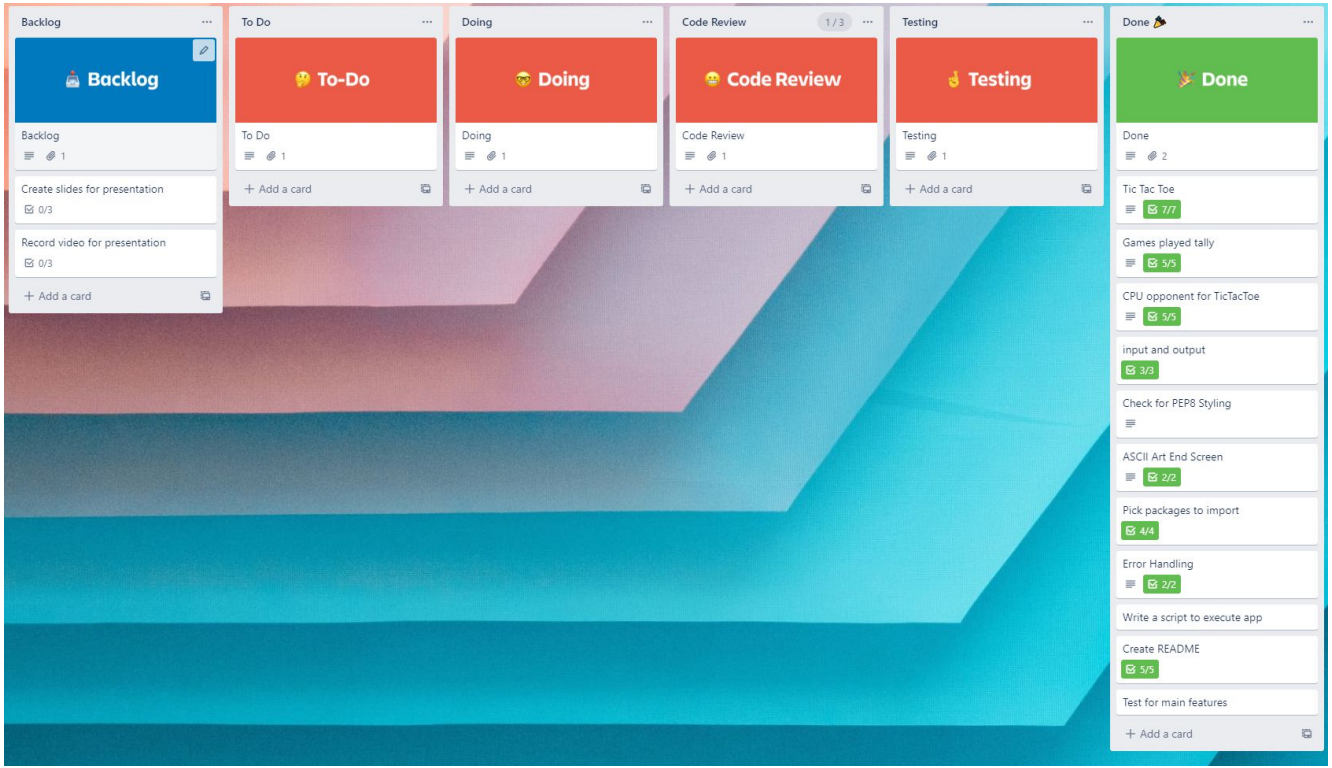


By Harry Rogers Woollett

Initial Pitch

- Play board
- 2 players
- Game tally
- End screen with ASCII art

Trello


Backlog
Backlog
Backlog
1
Create slides for presentation
0/3
Record video for presentation
0/3
Add a card
To Do
To-Do
To Do
1
Add a card
Doing
Doing
Doing
1
Add a card
Code Review
1 / 3
Code Review
Code Review
1
Add a card
Testing
Testing
Testing
1
Add a card
Done
Done
Done
2
Tic Tac Toe
7/7
Games played tally
5/5
CPU opponent for TicTacToe
5/5
input and output
3/3
Check for PEP8 Styling
ASCII Art End Screen
2/2
Pick packages to import
4/4
Error Handling
2/2
Write a script to execute app
Create README
5/5
Test for main features
Add a card

Tic Tac Toe

in list Done

Description

Edit

A playable version of the game

Checklist

Hide checked items | Delete

100%

- [x] ~~create playing board~~
- [x] ~~player input function~~
- [x] ~~check rows function~~
- [x] ~~check columns function~~
- [x] ~~check diagonals function~~
- [x] ~~check win conditions function~~
- [x] ~~switch players function~~

Add an item

CPU opponent for TicTacToe

in list Done

Description

Edit

Opponent playing as "O" will compete against the player

Checklist

Hide checked items | Delete

100%

- [x] ~~import random module~~
- [x] ~~create function for CPU to make random input on board~~
- [x] ~~control swaps back to player after each choice~~
- [x] ~~CPU can't repeat inputs on occupied spaces~~
- [x] ~~game result check after each input~~

Add an item

Tic Tac Toe Game

```
# Globals

board = ["-", "-", "-",
         "-", "-", "-",
         "-", "-", "-"]
```

```
# Display the gameboard
def print_board(board):
    """Prints the gameboard.

    Args:
        board (_type_): _description_
    """
    print(board[0] + " | " + board[1] + " | " + board[2])
    print("----------")
    print(board[3] + " | " + board[4] + " | " + board[5])
    print("----------")
    print(board[6] + " | " + board[7] + " | " + board[8])
```

```python
# Game Running Loop

while game_running:
    print_board(board)
    player_input(board)
    check_win()
    check_draw(board)
    switch_player()
    cpu(board)
    check_win()
    check_draw(board)
```

```python
def check_row(board):
    """Checks rows of gameboard for win condition.

    Args:
        board (_type_): _description_

    Returns:
        _type_: _description_
    """
    global winner
    if board[0] == board[1] == board[2] and board[1] != "-":
        winner = board[0]
        return True
    elif board[3] == board[4] == board[5] and board[3] != "-":
        winner = board[3]
        return True
    elif board[6] == board[7] == board[8] and board[6] != "-":
        winner = board[6]
        return True
```

Computer Opponent

```
Please enter a number between 1 and 9: 1
X | - | -
----------
- | - | -
----------
O | - | -
Please enter a number between 1 and 9: 2
X | X | O
----------
- | - | -
----------
O | - | -
Please enter a number between 1 and 9:
```

```python
# Computer Opponent
def cpu(board):
    """Generates random inputs for computer opponent

    Args:
        board (_type_): _description_
    """
    while current_player == "O":
        position = random.randint(0, 8)
        if board[position] == "-":
            board[position] = "O"
            switch_player()
```

Game Tally

```
# print result to tally.txt file
with open('tally.txt', 'a') as f:
    f.write(f"{current_player}, you win! \n")
```

```
count = sum(1 for line in open('tally.txt'))
print(f'There have been {count} total games played.')
```

```
There have been 2 total games played.
```

src > tally.txt

```
1   X, you win!
2   X, you win!
3
```

ASCII End Screen & Menu