

Animal Tic-Tac-Toe

A TERMINAL APPLICATION

BY DAYLE CLARKE

Welcome to ...

Tic Tac Toe

+++++

How would you like to begin today?

Menu entries can be selected with the arrow or j/k keys.

> Register as a new user
Login as an existing user
Play as a guest (no log-in required)

T1A3

Main Features

1. An **easy mode-** person vs. environment (PvE)
2. An **expert mode** person vs. environment (PvE)
3. Game play order is determined by the outcome of a **Scissor's-Paper-Rock game**
4. **User log-in** with data maintained in a csv file (user_credentials.csv)
5. **Game play data is maintained** in a csv file (player_scores.csv).



1. Easy Mode

Lots of players are around today who would love to play Tic-Tac-Toe with you.
They each have different skill levels and experience.
Here is a table outlining info about each player including their win, tie, and loss history:

```

+-----+ +-----+ +-----+
|O|p|p|o|n|e|n|t| |P|l|a|y|e|r| |I|n|f|o|r|m|a|t|i|o|n|
+-----+ +-----+ +-----+

```

player_name	level	personality	wins	ties	losses	total_games	percentage_loss_ratio
Pete the Panda	beginner	kind-hearted and lovable	7	13	63	83	75.90%
Katie the Koala	beginner	beautiful and creative	8	10	41	59	69.49%
Ollie the Octopus	genius	intelligent and quick-witted	66	79	0	145	0.00%
Danni the Dolphin	genius	innovative and elegant	48	63	0	111	0.00%

To help select the correct player for you, what difficulty level would you like to play on?
Menu entries can be selected with the arrow or j/k keys.

Game difficulty level:

```

> Easy Mode
  Expert Mode

```

1 Easy Mode con.

A player's game history and information about their personality is read from a separate csv file

You have chosen to play on Easy Mode.
There are two players who I recommend you challenge to a game.

Firstly there is the kind-hearted and lovable Pete the Panda.
He has won 7 games out of 83 and has still not discovered a reliable strategy to win.

Secondly there is Katie the Koala.
She is kind-hearted and lovable but is too busy eating eucalyptus leaves to focus long enough to consistently win.
She has won 7 games out of 83.

You can select any player you like but if you wish to play on easy mode those are the two I recommend.

Menu entries can be selected with the arrow or j/k keys.

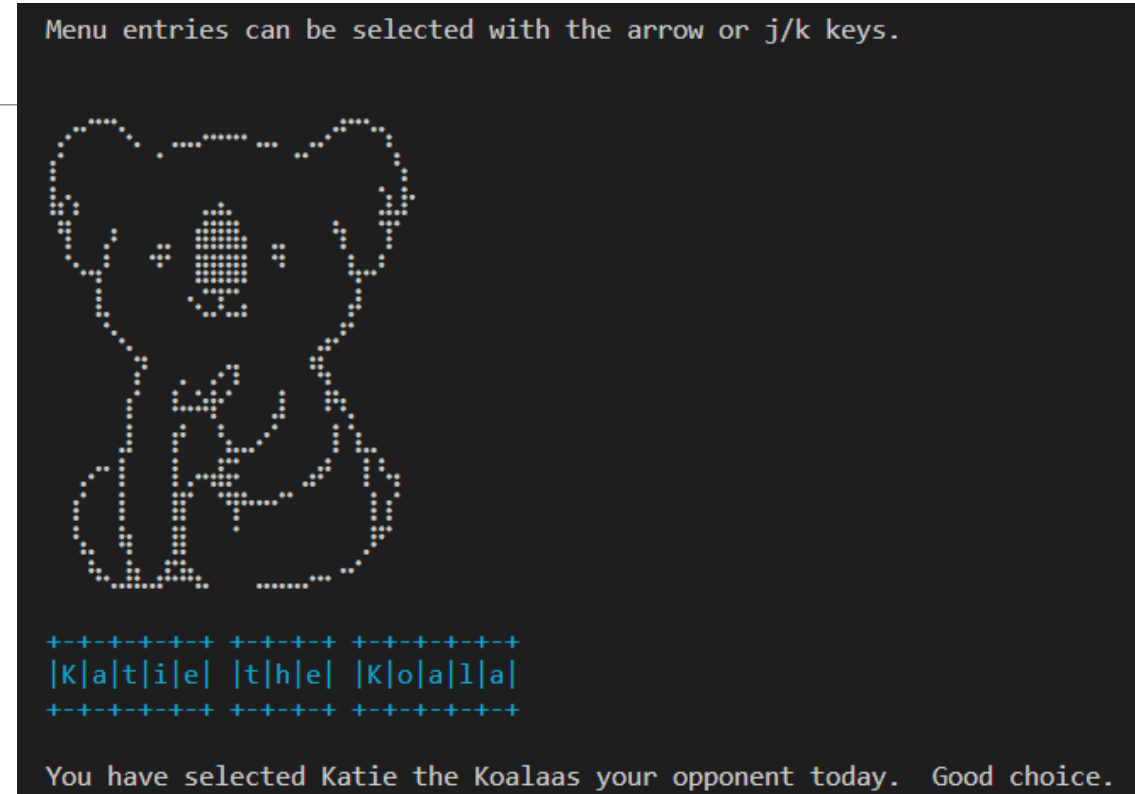
Which character would you like to play with today?:

- > Pete the Panda (easy mode)
- Katie the Koala (easy mode)
- Ollie the Octopus (expert mode)
- Danni the Dolphin (expert mode)

Beginner Characters



Stark, J. G. (2022). *Pandas*. Retrieved September 20, 2022, from Injosoft ASCII Art Archive: <https://www.asciart.eu/animals/pandas>



Stark, J. G. (2022). *Marsupials*. Retrieved September 20, 2022, from Injosoft ASCII Art Archive: <https://www.asciart.eu/animals/marsupials>

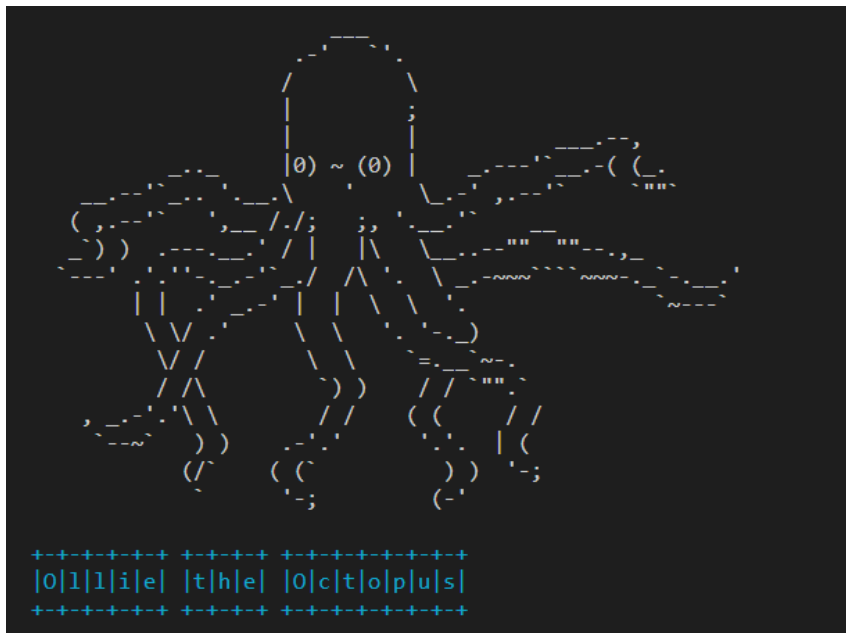
Expert Mode

You have chosen to play on **Expert Mode**.
There are two players I recommend you challenge to a game.

Firstly there is the intelligent and quick-witted **Ollie the Octopus**. He is currently undefeated having never lost a game. He has won 7 games out of 83.
Secondly there is **Danni the Dolphin**. She is innovative and elegant and also remains undefeated. She has won 48 games out of 111.

You can select any player that you like but if you wish to play on expert mode those are the two I recommend to appropriately test your abilities.

Menu entries can be selected with the arrow or j/k keys.



David Middlehurst. (2022). ASCII.co. Retrieved September 20, 2022, from OCTOPUS - ASCII ART: <https://ascii.co.uk/art/octopus>



Kwasniewski, M. '. (2022). *Dolphins*. Retrieved September 20, 2022, from Injosoft ASCII Art Archive: <https://www.asciiart.eu/animals/dolphins>

```
def minimax(self, state, is_maximising):
    """A recursive minimax method used to return a utility score
    for each possible position.

    The minimax method will continue until one of the following
    terminal conditions are met:
    1. The maximising player wins:
        Returns:
            dict: showing the position previously determined
            through recursion and a score of 1 * (the
            number of empty squares remaining + 1).
    2. The minimising player wins:
        Returns:
            dict: showing the position previously determined
            through recursion and a score of -1 * (the
            number of empty squares remaining + 1).
    3. There are no positions left on the board:
        Returns:
            dict: showing the position previously determined
            through recursion and a score of 0

    Assumes that the minimising player is also playing optimally.

    Args:
    state(any): an instance of the TicTacToe class
        shows the current board state in that simulation.
    is_maximising(bool): indicates if it is the maximiser's turn in the
    simulation.
        when it is the maximiser's turn they are playing to get the highest
        utility score.
        when it is the minimiser's turn they are playing to get the lowest
        utility score.

    Returns:
        dict: showing the position (with an int value between 0-8)
        and score (an int value) see terminal conditions above.
    """
```

```
    maximising_letter = self.letter
    minimising_letter = "X" if maximising_letter == "O" else "O"
    # Terminal states
    if state.current_winner:
        return {
            "position": None,
            "score": 1 * (state.num_empty_squares() + 1)
            if state.current_winner == maximising_letter
            else -1 * (state.num_empty_squares() + 1)
        }
    if not state.free_positions():
        return {"position": None, "score": 0}

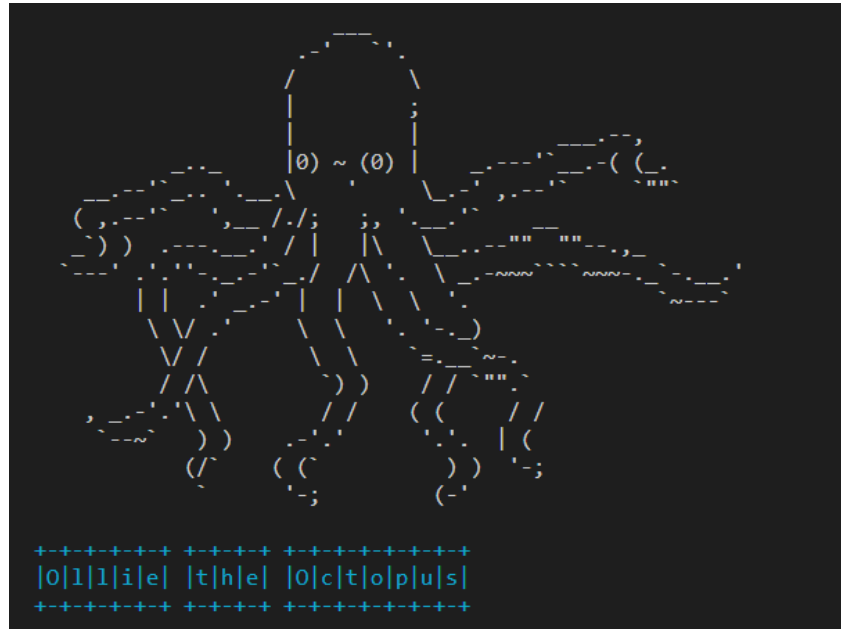
    # When it is the maximisor's turn they are playing to get the
    # highest utility score.
    if is_maximising:
        best_move = {"position": None, "score": -500}
        # Loop through all free moves remaining.
        for possible_move in (state.free_positions()):
            state.make_move(possible_move, maximising_letter)
            # Method calls itself to test that move.
            sim_score = self.minimax(state, False)
            # The board is then returned to it's original position
            state.board[possible_move] = " "
            state.current_winner = None
            sim_score["position"] = possible_move
            if sim_score["score"] > best_move["score"]:
                best_move = sim_score
        return best_move

    # When it is the minimisor's turn in the simulation the
    # following code will be executed:
    # The best move for the maximisor is the lowest utility score.
    best_move = {"position": None, "score": 500}
    for possible_move in state.free_positions():
        state.make_move(possible_move, minimising_letter)
        sim_score = self.minimax(state, True)
        state.board[possible_move] = " "
        state.current_winner = None
        sim_score["position"] = possible_move
        if sim_score["score"] < best_move["score"]:
            best_move = sim_score
    return best_move
```

The Unbeatable AI

A test was conducted to check the accuracy of the ExpertComputerPlayer AI. This set up games where an instance of the ExpertComputerPlayer (Ollie the Octopus) played against an instance of the RandomComputerPlayer (Katie the Koala).

Out of 1000 games Ollie won 902, tied 98 and lost none.



VS



```
• (.venv) day1e07@LAPTOP-07L0PUKG:~/term1/{DayleClarke}_T1A3/src$ python3 check_ai.py
      player_name  win  tie  loss  total_games  percentage_loss_ratio
0    Katie the Koala    0   98   902         1000          90.20%
1  Ollie the Octopus  902   98    0         1000           0.00%
```


Game play order determined by Scissor's-Paper-Rock game

```
You have selected Danni the Dolphin as your opponent today.  Good choice.  
We will begin by playing scissors-paper-rock to determine which player will start.  
Menu entries can be selected with the arrow or j/k keys.  
Please select one of the following hand gestures:
```

```
> rock  
  paper  
  scissors
```

```
-----  
---'-----  
      -----  
      -----  
      -----  
      -----  
      -----  
---.-----  
      -----
```

Dayle has chosen to play paper.

```
-----  
---'-----  
      -----  
      -----  
      -----  
      -----  
      -----  
---.-----  
      -----
```

Dayle has chosen to play scissors.

```
-----  
---'-----  
      -----  
      -----  
      -----  
      -----  
      -----  
---.-----  
      -----
```

Dayle has chosen to play rock.

[illegible]

Karlsson, V. (2022). Hand Gestures- Nonverbal Communication - Signals. Retrieved September 19, 2022, from Injosoft ASCII Art Archive:
<https://www.asciiart.eu/people/body-parts/hand-gestures>

4. User Registration/ Log in Option

```
src > user_credentials.csv
1  username,password
2  Dayle>Password1
3  Gillian,catDog1
4  Joanne,Alex123
5  Jackie,Easter121
6  Katie,Pancake1
7  Bec,AppleB1
8  Tilley,Mellon1
9  DeeDee,Dalma1
10 Darma,Darma12
11 Jill>Password134
12
```

User data is maintained
in a separate csv file
(user_credentials.csv)

Users can register as a new
user, log in with an existing
username or to play as a
guest.

Welcome to ...

THE TAG TAG

+++++

How would you like to begin today?

Menu entries can be selected with the arrow or j/k keys.

```
> Register as a new user
  Login as an existing user
  Play as a guest (no log-in required)
```

Users must have a
unique username:

How would you like to begin today?
Menu entries can be selected with the arrow or j/k keys.

```
Register as a new user
Username: Dayle
That username is already taken. Please choose a unique username.
Username: Dayle07
You entered Dayle07 for your username.
```

New Passwords must:

be 5-10
characters long.

contain at least one
upper case and one
lower case
character

contain a
number

be confirmed

```
Enter a password to associate with your account.
```

```
Passwords must:
```

- 1) be between 5 and 10 characters in length.
- 2) contain at least one number
- 3) contain one uppercase and one lowercase letter.

```
Password: pass
```

```
That is not a valid password length. Please enter a password that contains between 5 and 10 characters (inclusive).
```

```
Password: password123435687
```

```
That is not a valid password length. Please enter a password that contains between 5 and 10 characters (inclusive).
```

```
Password: password
```

```
That is not a valid password. Please ensure your password contains at least one upper and one lower case character.
```

```
Password: PASSWORD
```

```
That is not a valid password. Please ensure your password contains at least one upper and one lower case character.
```

```
Password: Password
```

```
That is not a valid password. Please ensure your password contains at least one number.
```

```
Password: Password1
```

```
Confirm password: password1
```

```
The passwords entered do not match. Please try again.
```


```
Password: Password1
```

```
Confirm password: Password1
```

```
Registration successfull! Welcome to Animal TicTacToe Dayle07!
```


Logging in as an Existing User

Users can log in with an existing username which is matched against the data in the user_credentials.csv file.



```
How would you like to begin today?  
Menu entries can be selected with the arrow or j/k keys.  
  
Login as an existing user  
Welcome back. Please enter your log_in credentials.  
Username: Bree  
Bree is not an existing username. Please try again.  
Username: Dayle  
You entered Dayle for your username  
  
Password: █
```

If they can't remember their username, the user has 3 chances to get it correct before returning to the main menu:



```
How would you like to begin today?  
Menu entries can be selected with the arrow or j/k keys.  
  
Login as an existing user  
Welcome back. Please enter your log_in credentials.  
Username: Test  
Test is not an existing username. Please try again.  
Username: Test  
Test is not an existing username. Please try again.  
Username: Test  
Test is not an existing username. Please try again.  
You seem to be having difficulty remembering your username.  
  
> Register as a new user  
Login as an existing user  
Play as a guest (no log-in required)
```

Logging in as an Existing User- Passwords

```
Password: test
Incorrect password. Please try again.
Password: test
Incorrect password. Please try again.
Password: test
Incorrect password. Please try again.
You seem to be having difficulty remembering your password.

> Register as a new user
  Login as an existing user
  Play as a guest (no log-in required)
```

If they can't enter their password correctly (if it doesn't match what is stored as their password in user_credentials.csv , the user has 3 chances to get it correct before returning to the main menu:

When they enter the username and password correctly:

```
+++++
How would you like to begin today?
Menu entries can be selected with the arrow or j/k keys.

Login as an existing user
Welcome back. Please enter your log_in credentials.
Username: Dayle
Password: testPass1
Registration successfull! Welcome back to Animal TicTacToe Dayle!
```

Game play data is maintained in a csv file (player_scores.csv).

This csv file records the win, loss, and tie history of each player

After each game scores are updated, and a new loss percentage ratio is calculated.

At the end of each game the top 10 players (as calculated based on the total number of wins) are printed to the console.

```

+---+---+ +---+ +---+---+---+---+
|T|o|p| |1|0| |P|l|a|y|e|r|s|
+---+---+ +---+ +---+---+---+---+

      player_name  wins  ties  losses  total_games  percentage_loss_ratio
Ollie the Octopus    66    79      0         145          0.00%
Danni the Dolphin    48    63      0         111          0.00%
      Guest         13     2       3          18         16.67%
Katie the Koala       8    10     41          59         69.49%
Pete the Panda        7    13     64          84         76.19%
      Dayle         3     0       0           3          0.00%
      Gillian        1     0       0           1          0.00%
      Joanne         1     0       0           1          0.00%
      Jackie         1     0       0           1          0.00%
      Katie          1     0       0           1          0.00%
      Bec            1     0       0           1          0.00%

Thanks for playing today. Would you like to play again? (yes/no): █
```

The following two functions will be created to allow the Expert Computer Player to select the most optimal position to place their marker.

define get_move function
with parameters for self (the instance of the
Expert ComputerPlayer class) and the game

```
Are all positions on
the board free
(board.free_positions
== 9)
```

Return a random choice from one of the board's corners (either 0, 2, 6 or 8)

End (return to it's position in the program where it was first called)

End (return to it's position in the program where it was first called)

- ➔ to select the best position to play in and return the result.

Minimax method invoked

Return a dictionary of the position previously determined through recursion) and a score of 1 * (the number of empty squares remaining + 1)

```
define minimax_function
```

- with parameters for self (the instance of the ExpertComputerPlayer class), the state the board is in in the simulation, and a boolean value indicating it is currently a maximising player's turn)

Define a variable to store the maximising player's letter (O)

Define a variable to store the minimising player's letter (X)

Test 1 to see if a Terminal Condition has been achieved

Is there a current winner?

Test 2 to see if a Terminal Condition has been achieved

—Yes—

Are there no more positions on the board left (number of empty squares == zero)

Return a dictionary of the position previously determined through recursion) and a score of 0.

```
Is current winner ==
    maximising_letter
```

maximising_letter

```
ent winner ==
mising_letter
```

—No

Return a dictionary of the position previously determined through recursion) and a score of $-1 * (\text{the number of empty squares remaining} + 1)$

Yes

11

Continued below (on next slide)



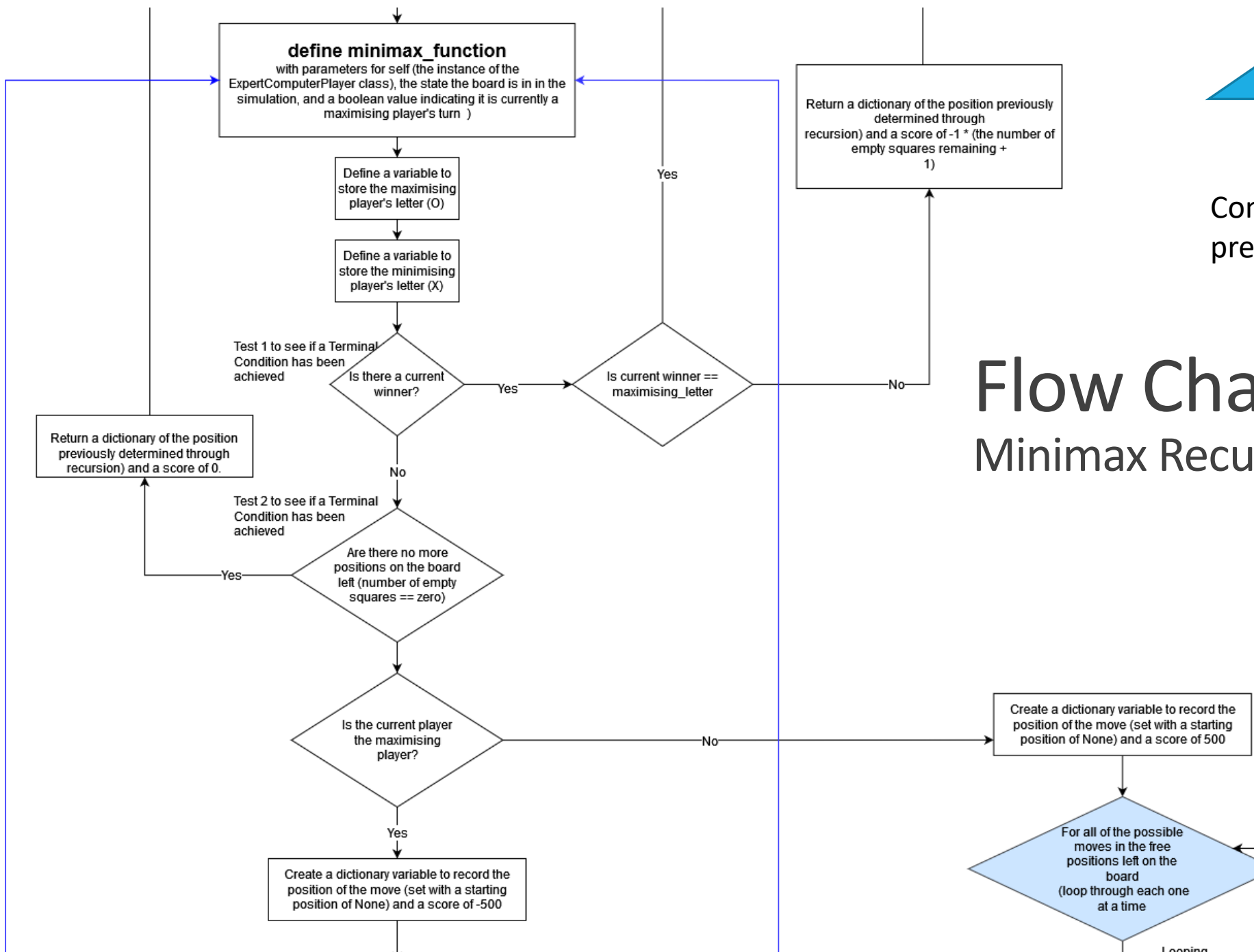


Continued above (on previous slide)

Flow Chart

Minimax Recursive Method Con

Continued below (on next slide)

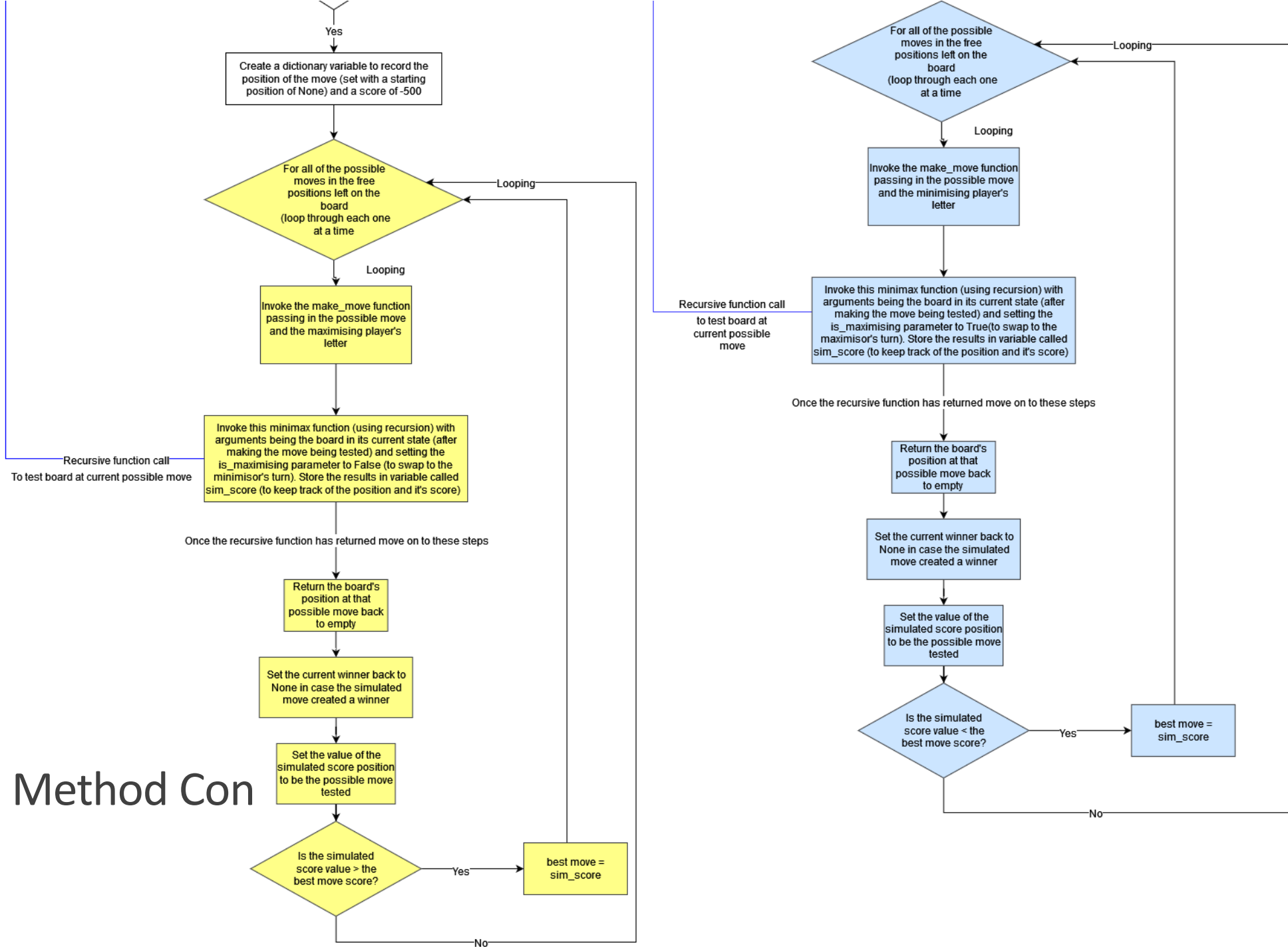




Continued above (on previous slide)

Flow Chart

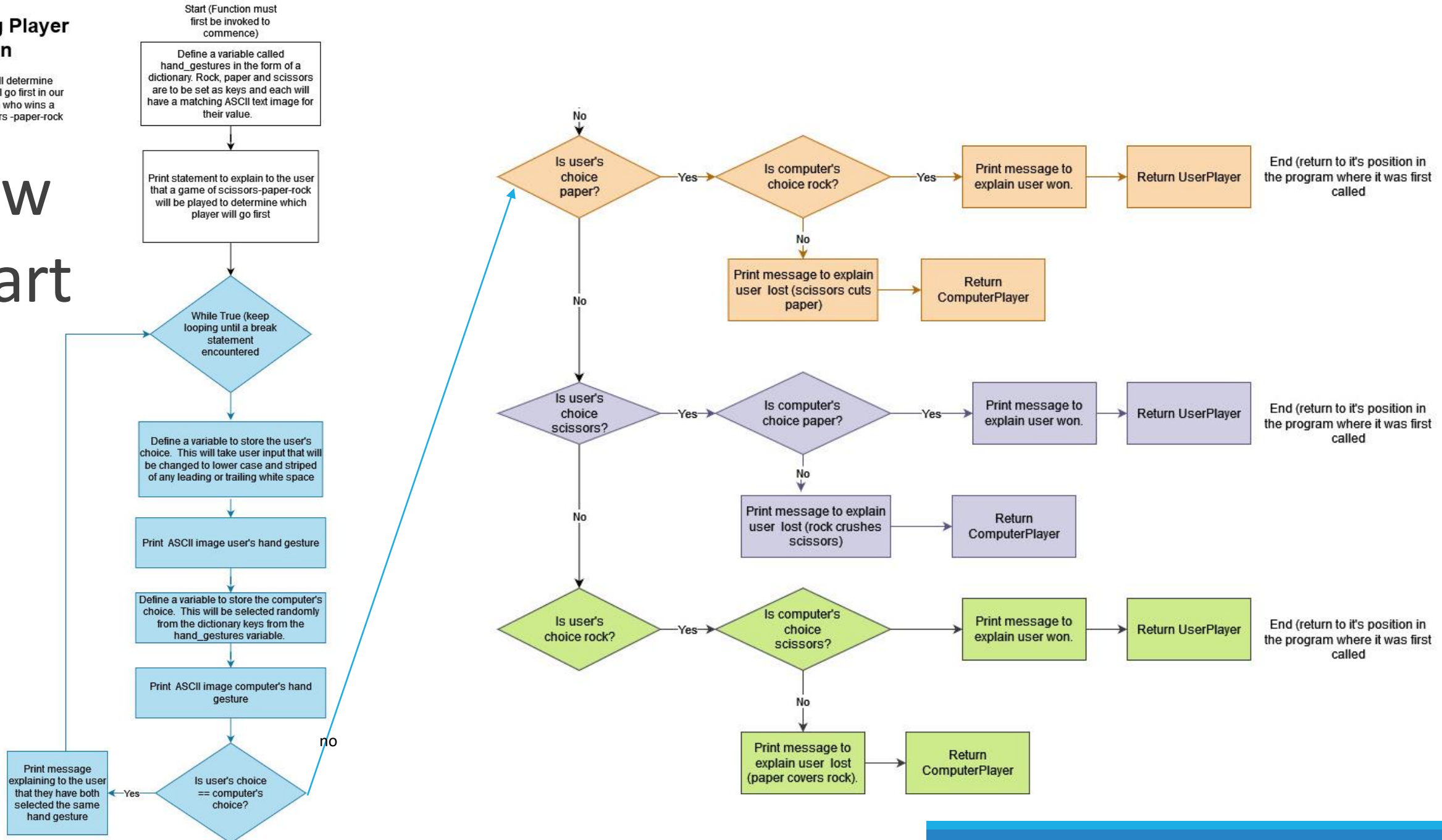
Minimax Recursive Method Con



Starting Player Function

This function will determine which player will go first in our game based on who wins a game of scissors-paper-rock

Flow Chart



Challenges

Deciding on a application idea- delayed starting.

Getting the minimax algorithm to work- a challenging project

Time management and prioritisation

Bash Scripting

Ethical Issues

Ensuring all of the code produced was original

Remembering to maintain a record of all sources used in the application such as ASCII image references.