T3 – LE BAO DUY NGUYEN (102449993)

SUMMARY

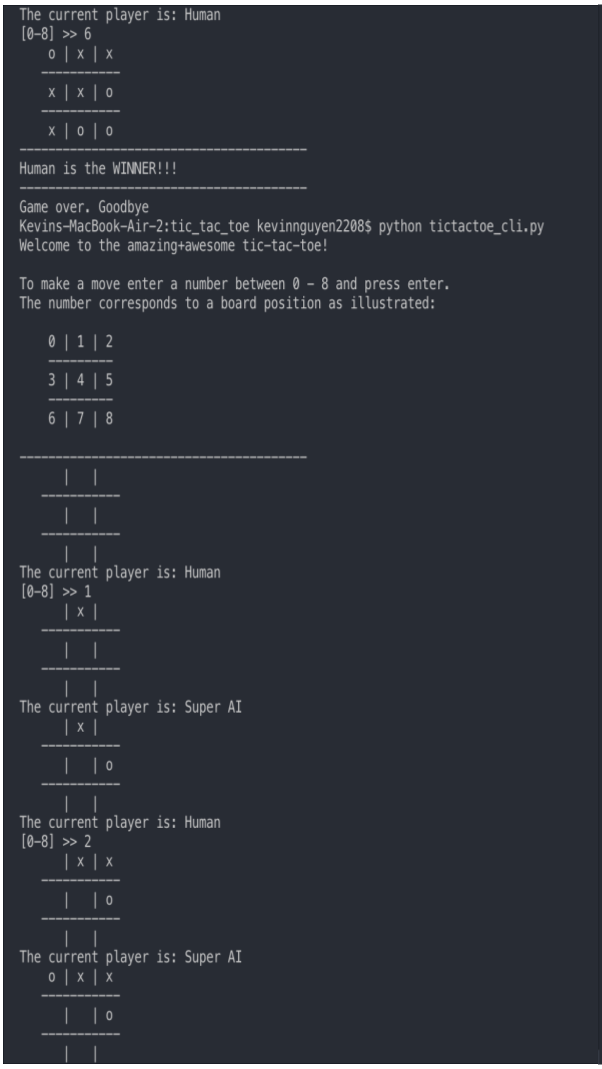
The goal of this game is to create a simple tic tac toe game between human (user) and AI.

```
#==========================================================================================
# agent (human or AI) functions


def get_human_move():
    '''Get a human players raw input. Returns None if a number is not entered.'''
    return input('[0-8] >> ')


def get_ai_move():
    '''Get the AI's next move'''
    # Check the possibilities of winning
    for row in WIN_SET:
        # if certain spaces in each row the ai will return a result
        if board[row[0]] == board[row[1]] and board[row[2]] == ' ':
            return row[2]
        elif board[row[1]] == board[row[2]] and board[row[0]] == ' ':
            return row[0]
        elif board[row[0]] == board[row[2]] and board[row[1]] == ' ':
            return row[1]

    return randrange(9)


def get_ai_move_2():
    #   '''Get the AI's 2 next move'''
    # check the current results.
    if check_for_result():
        # return result.
        return check_for_result()
    # Check what moves have been made.
    elif check_move():
        # return move.
```

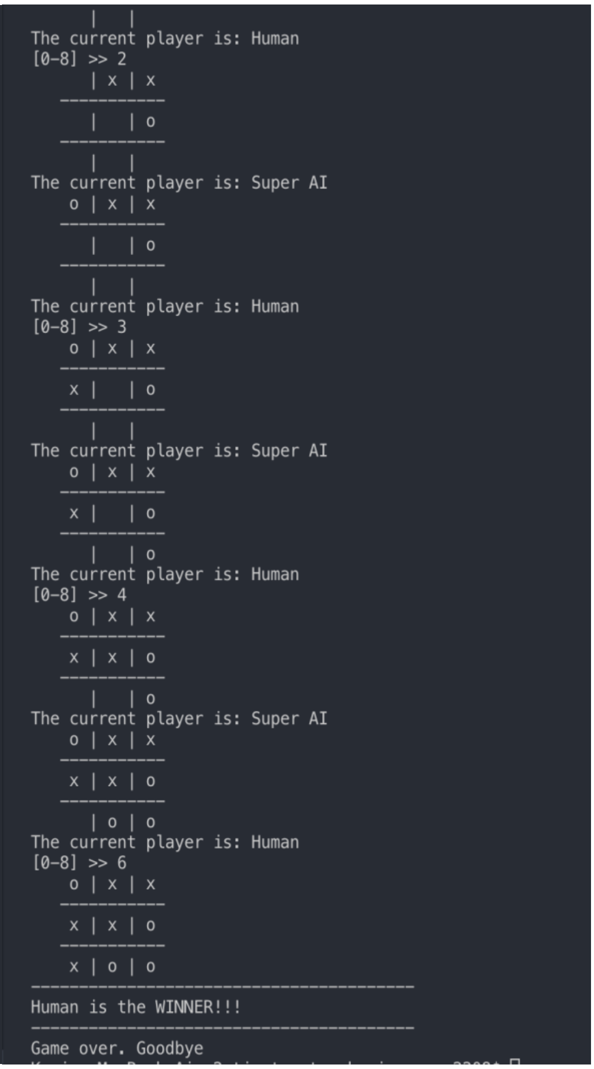
```
    return check_move()
# return random if nothing else works.
else:
    return randrange(9)
```

RESULT

T3 – LE BAO DUY NGUYEN (102449993)



```
The current player is: Human
[0-8] >> 6
   o | x | x
  -----------
   x | x | o
  -----------
   x | o | o
----------------------------------------
Human is the WINNER!!!
----------------------------------------
Game over. Goodbye
Kevins-MacBook-Air-2:tic_tac_toe kevinnguyen2208$ python tictactoe_cli.py
Welcome to the amazing+awesome tic-tac-toe!

To make a move enter a number between 0 - 8 and press enter.
The number corresponds to a board position as illustrated:

   0 | 1 | 2
   ---------
   3 | 4 | 5
   ---------
   6 | 7 | 8

----------------------------------------
     |   |
  -----------
     |   |
  -----------
     |   |
The current player is: Human
[0-8] >> 1
     | x |
  -----------
     |   |
  -----------
     |   |
The current player is: Super AI
     | x |
  -----------
     |   | o
  -----------
     |   |
The current player is: Human
[0-8] >> 2
     | x | x
  -----------
     |   | o
  -----------
     |   |
The current player is: Super AI
   o | x | x
  -----------
     |   | o
  -----------
     |   |
```



```
     |   |
The current player is: Human
[0-8] >> 2
     | x | x
  -----------
     |   | o
  -----------
     |   |
The current player is: Super AI
   o | x | x
  -----------
     |   | o
  -----------
     |   |
The current player is: Human
[0-8] >> 3
   o | x | x
  -----------
   x |   | o
  -----------
     |   |
The current player is: Super AI
   o | x | x
  -----------
   x |   | o
  -----------
     |   | o
The current player is: Human
[0-8] >> 4
   o | x | x
  -----------
   x | x | o
  -----------
     |   | o
The current player is: Super AI
   o | x | x
  -----------
   x | x | o
  -----------
     | o | o
The current player is: Human
[0-8] >> 6
   o | x | x
  -----------
   x | x | o
  -----------
   x | o | o
----------------------------------------
Human is the WINNER!!!
----------------------------------------
Game over. Goodbye
```

EXTENSION

T3 – LE BAO DUY NGUYEN (102449993)

The goal of the extension of this game is to create a more advanced twist to the game. By letting the user the options to choose who they'd like to face , or see other AI's play against each other, adds diversity to the game.

```
from random import randrange

# static game data - doesn't change (hence immutable tuple data type)
#==============================================================================
# Game model functions

def reset_game_data():
    '''Resets the game data in the global variables to the defaults'''
    global board, current_player, ai_choice, ai_choices, ai_vs_ai, players, winner, move, firstTurn, quitting
    board = [' '] * 9
    current_player = ''  # 'x' or 'o' for first and second player
    ai_choices = {'r': 'Random', 'a': 'Average Ai', 's': 'Smart AI'}
    ai_choice = None
    ai_vs_ai = False
    players = {'x': 'Human', 'o': 'Super AI'}
    winner = None
    move = None
    firstTurn = True
    quitting = False


def check_set_for_player(set, player):

    count = 0
    move = -1

    for x, index in enumerate(set):
        if board[index] == player:
            count += 1
        elif board[index] != 'x' and board[index] != 'o':
            move = index
            if x == 2 and count != 2:
                # if the count is at two when we have ennumerated through the set then we set move
                # to -1 to indicate to return false
```

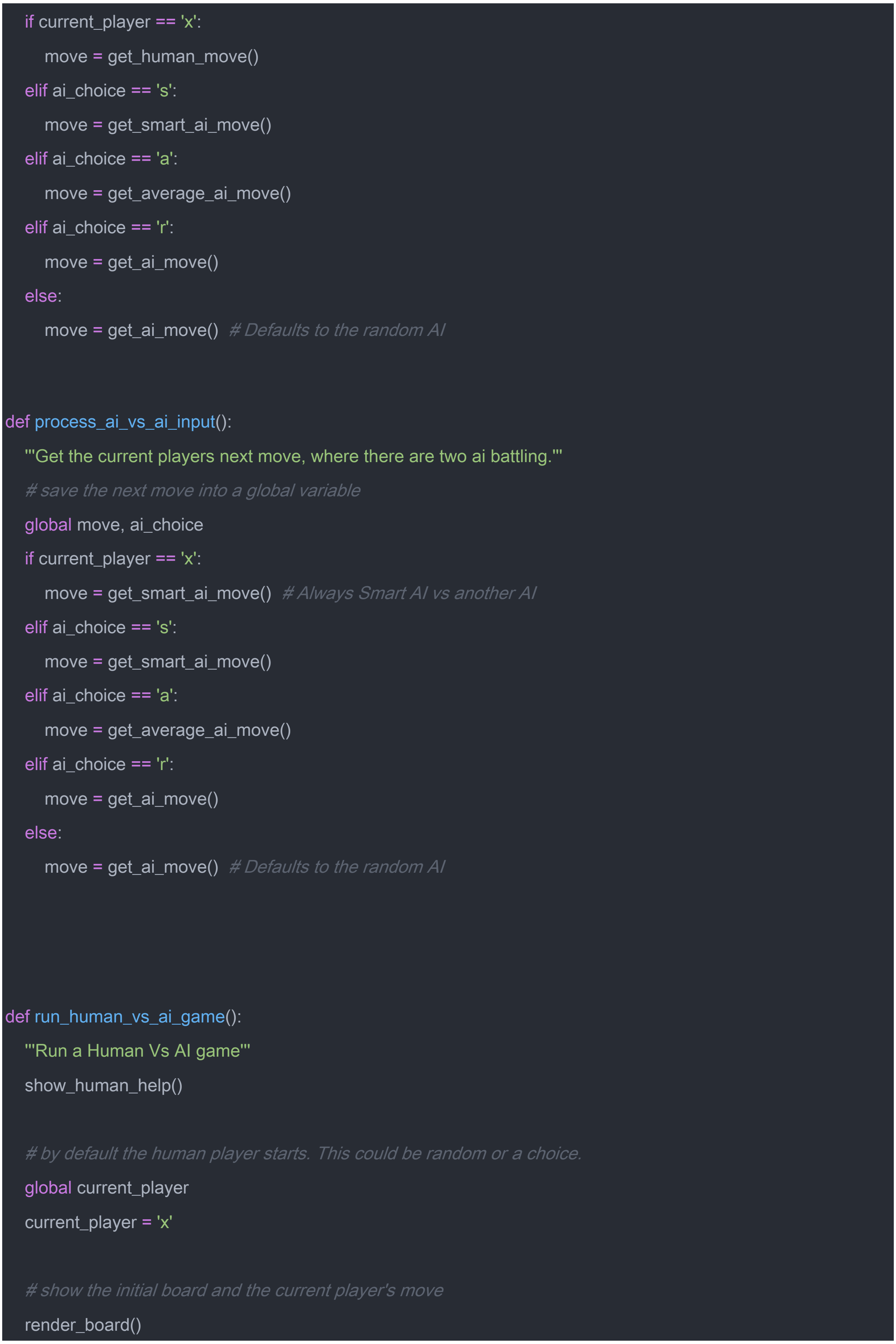
T3 – LE BAO DUY NGUYEN (102449993)

```
                move = -1
        else:
            move = -1

    return move


#==============================================================================
# agent (human or AI) functions
def get_human_move():
    '''Get a human players raw input. Returns None if a number is not entered.'''
    return input('[0-8] >> ')


def get_ai_move():
    '''Get the AI's next move '''
    # A simple dumb random move - valid or NOT!
    # Note: It is the models responsibility to check for valid moves...
    return randrange(9)  # [0..8]


def get_average_ai_move():

    global current_player

    if current_player == 'x':
        otherPlayer = 'o'
    else:
        otherPlayer = 'x'

    for set in WIN_SET:
        # Check if other player is about to win using check_set().
        chk = check_set_for_player(set, otherPlayer)
        if chk != -1:
            return chk
        # Then make the move to stop the other player from winning
    #else choose a random option
    return randrange(9)  # [0..8]
```

T3 – LE BAO DUY NGUYEN (102449993)

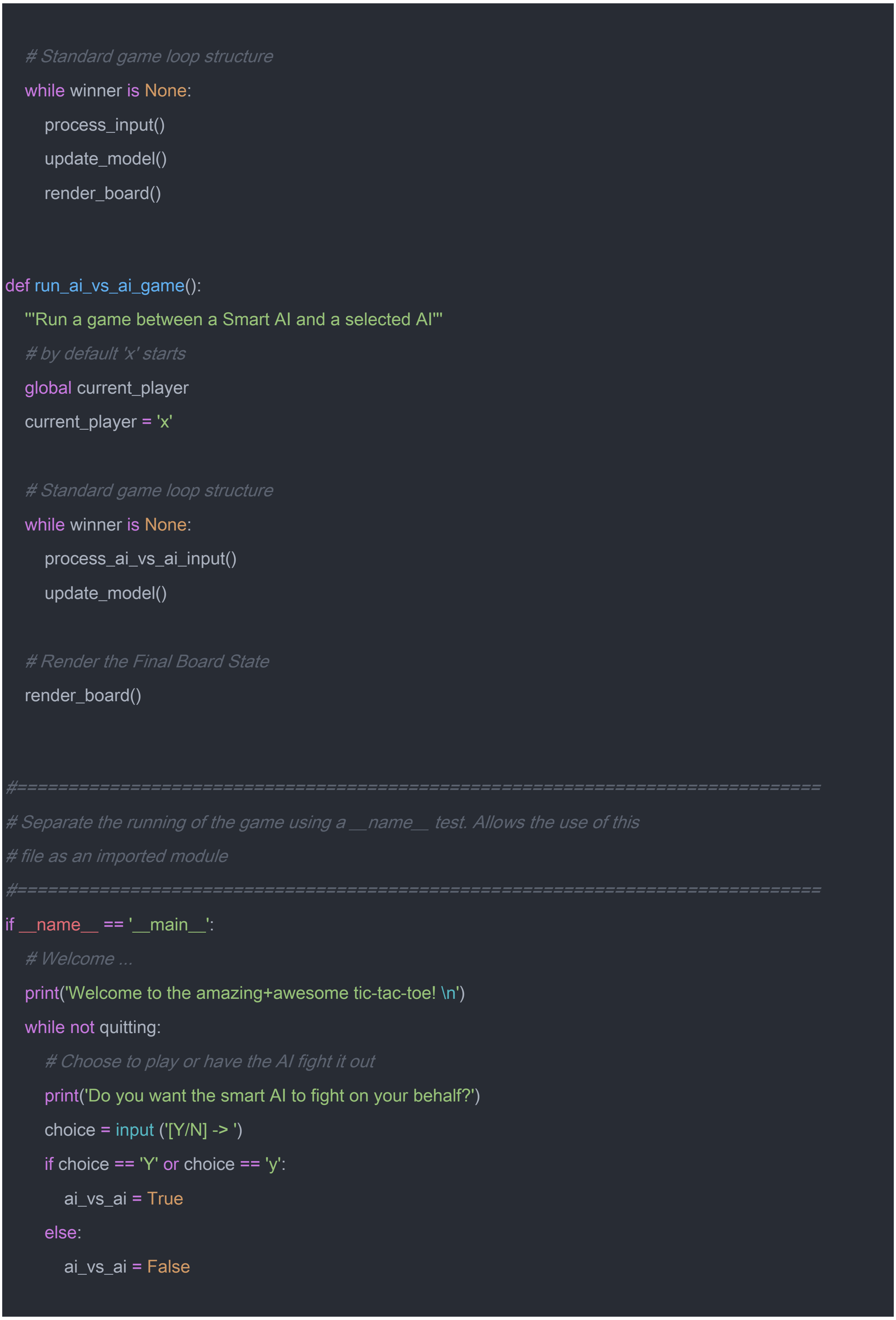
```python
def get_smart_ai_move():


    global current_player, firstTurn

    if current_player == 'x':
        otherPlayer = 'o'
    else:
        otherPlayer = 'x'

    for set in WIN_SET:
        # Check if this player is about to win using check_set()
        chk = check_set_for_player(set, current_player)
        if chk != -1:
            return chk
        # Then make the move that allows you to win
    # Check if other player is about to win using check_set().
        chk = check_set_for_player(set, otherPlayer)
        if chk != -1:
            return chk
        # Then make the move to stop the other player from winning

    # if its the first turn, return the middle
    if firstTurn:
        firstTurn = False  # set this to False so it only tries this once.
        return 4

    # If neither condition
    # Then make a random move from available spaces
    return randrange(9)  # [0..8]


#==============================================================================
# Standard trinity of game loop methods (functions)

def process_input():
    '''Get the current players next move.'''
    # save the next move into a global variable
    global move, ai_choice
```

T3 – LE BAO DUY NGUYEN (102449993)



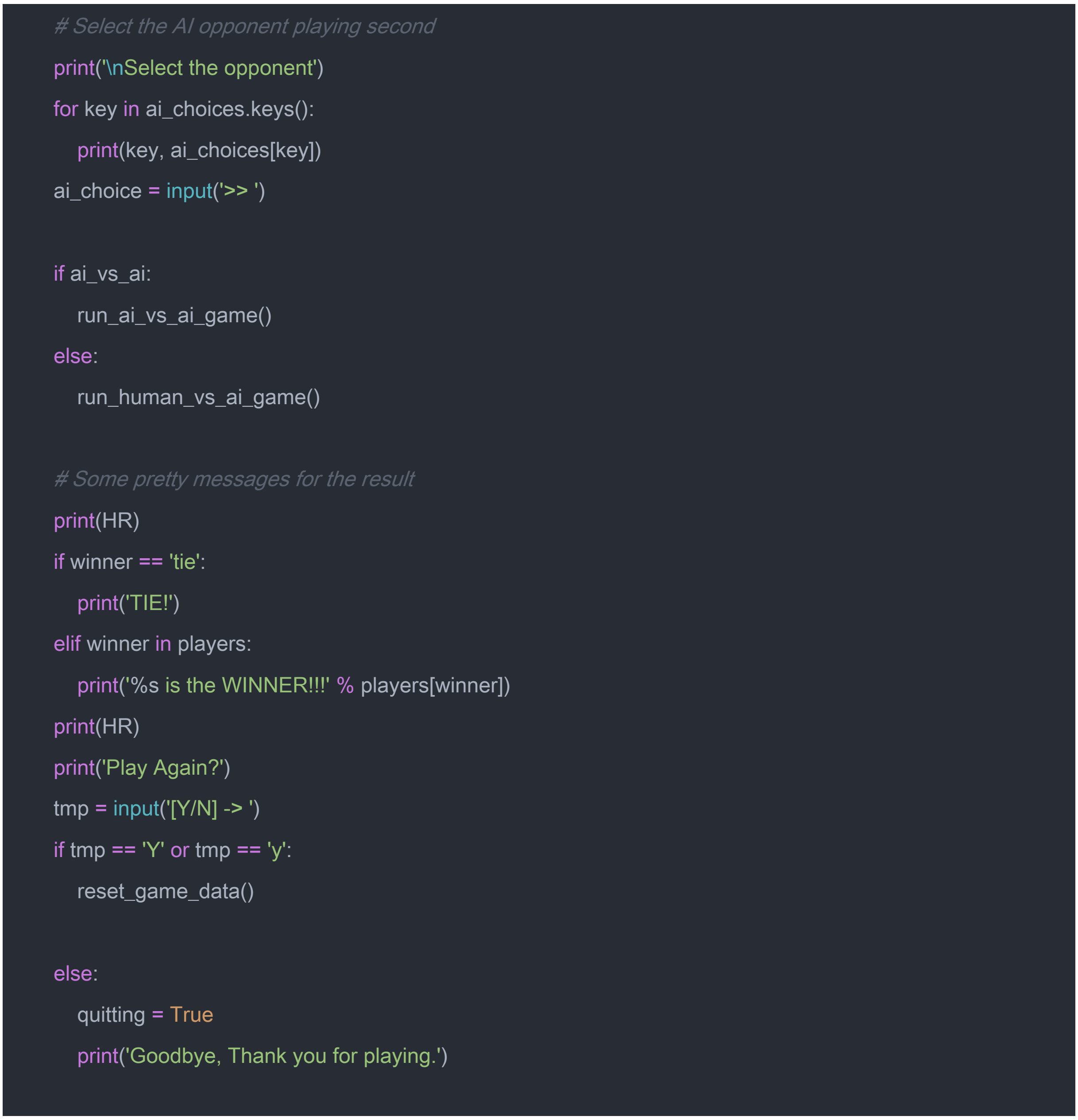
```
    if current_player == 'x':
        move = get_human_move()
    elif ai_choice == 's':
        move = get_smart_ai_move()
    elif ai_choice == 'a':
        move = get_average_ai_move()
    elif ai_choice == 'r':
        move = get_ai_move()
    else:
        move = get_ai_move()  # Defaults to the random AI


def process_ai_vs_ai_input():
    '''Get the current players next move, where there are two ai battling.'''
    # save the next move into a global variable
    global move, ai_choice
    if current_player == 'x':
        move = get_smart_ai_move()  # Always Smart AI vs another AI
    elif ai_choice == 's':
        move = get_smart_ai_move()
    elif ai_choice == 'a':
        move = get_average_ai_move()
    elif ai_choice == 'r':
        move = get_ai_move()
    else:
        move = get_ai_move()  # Defaults to the random AI




def run_human_vs_ai_game():
    '''Run a Human Vs AI game'''
    show_human_help()

    # by default the human player starts. This could be random or a choice.
    global current_player
    current_player = 'x'

    # show the initial board and the current player's move
    render_board()
```

T3 – LE BAO DUY NGUYEN (102449993)



```
    # Standard game loop structure
    while winner is None:
        process_input()
        update_model()
        render_board()


def run_ai_vs_ai_game():
    '''Run a game between a Smart AI and a selected AI'''
    # by default 'x' starts
    global current_player
    current_player = 'x'

    # Standard game loop structure
    while winner is None:
        process_ai_vs_ai_input()
        update_model()

    # Render the Final Board State
    render_board()


#===============================================================================
# Separate the running of the game using a __name__ test. Allows the use of this
# file as an imported module
#===============================================================================
if __name__ == '__main__':
    # Welcome ...
    print('Welcome to the amazing+awesome tic-tac-toe! \n')
    while not quitting:
        # Choose to play or have the AI fight it out
        print('Do you want the smart AI to fight on your behalf?')
        choice = input ('[Y/N] -> ')
        if choice == 'Y' or choice == 'y':
            ai_vs_ai = True
        else:
            ai_vs_ai = False
```

T3 – LE BAO DUY NGUYEN (102449993)

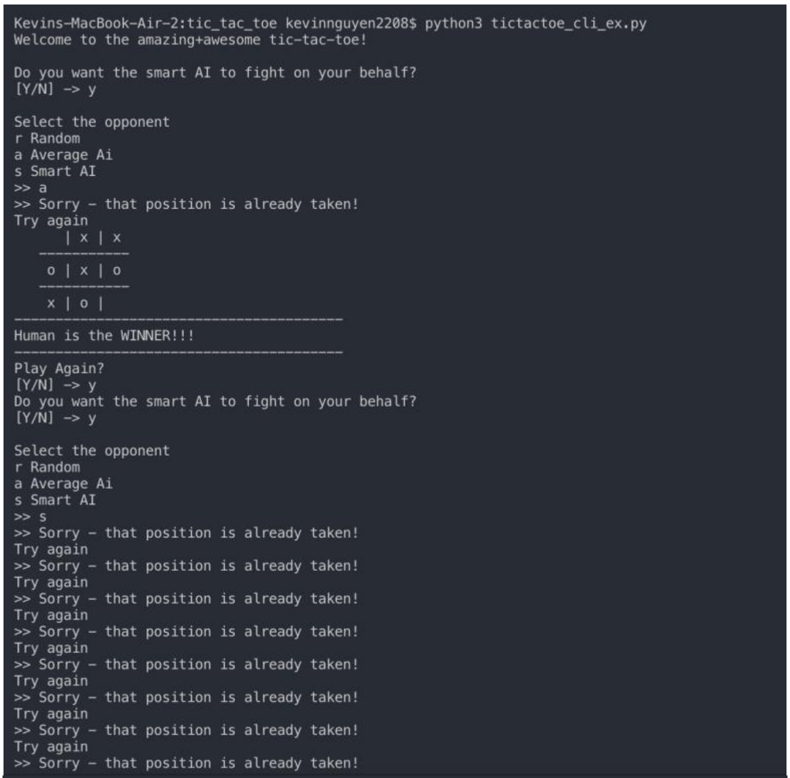


```
# Select the AI opponent playing second
print('\nSelect the opponent')
for key in ai_choices.keys():
    print(key, ai_choices[key])
ai_choice = input('>> ')

if ai_vs_ai:
    run_ai_vs_ai_game()
else:
    run_human_vs_ai_game()

# Some pretty messages for the result
print(HR)
if winner == 'tie':
    print('TIE!')
elif winner in players:
    print('%s is the WINNER!!!' % players[winner])
print(HR)
print('Play Again?')
tmp = input('[Y/N] -> ')
if tmp == 'Y' or tmp == 'y':
    reset_game_data()

else:
    quitting = True
    print('Goodbye, Thank you for playing.')
```

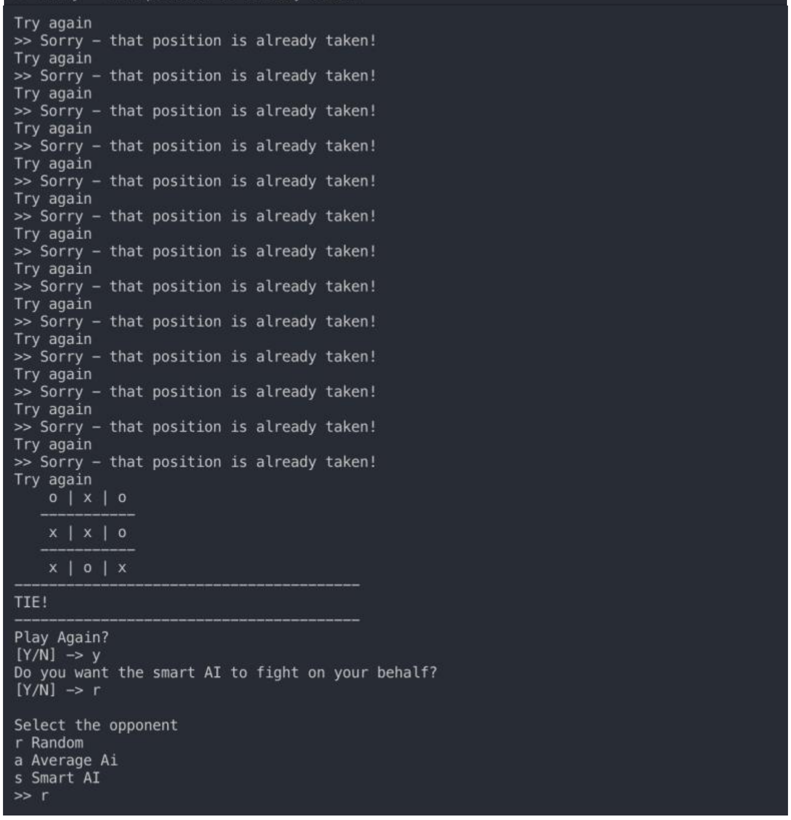
T3 – LE BAO DUY NGUYEN (102449993)

RESULT

T3 – LE BAO DUY NGUYEN (102449993)

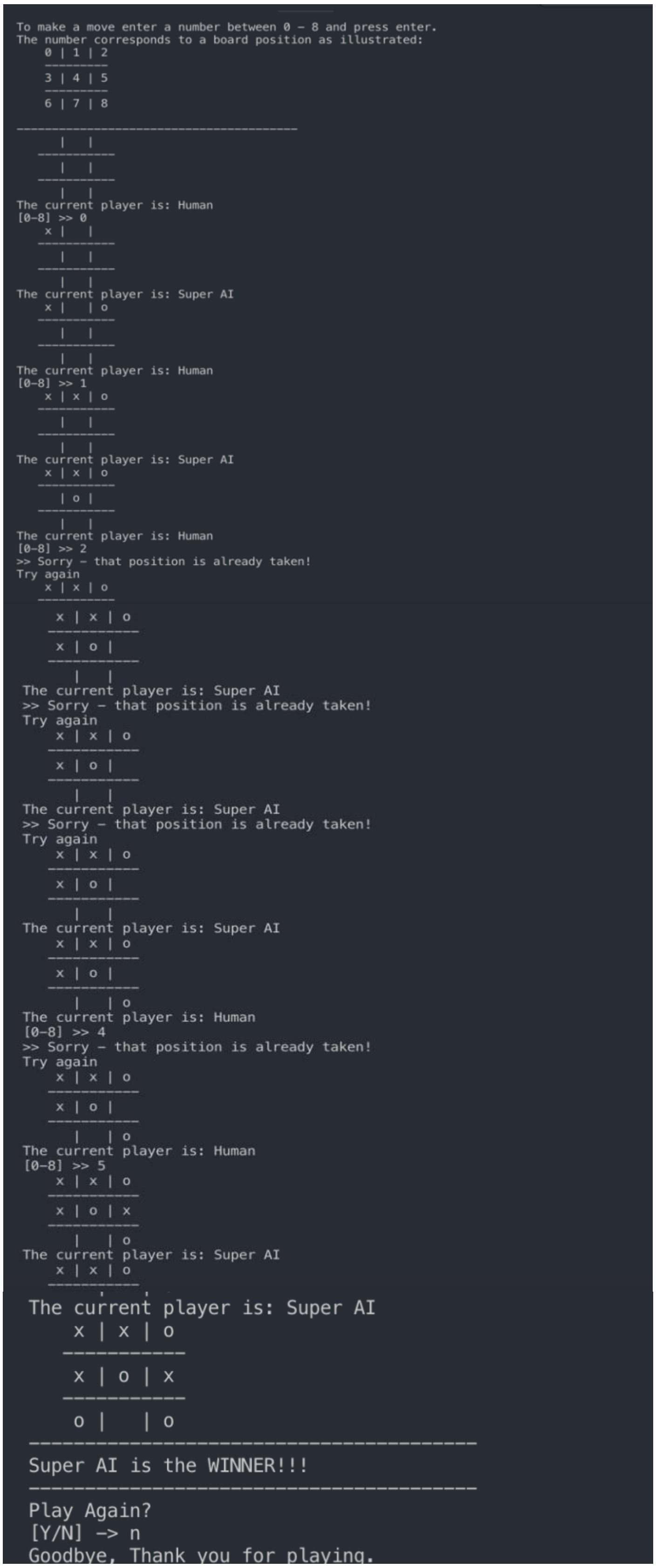


```
Kevins-MacBook-Air-2:tic_tac_toe kevinnguyen2208$ python3 tictactoe_cli_ex.py
Welcome to the amazing+awesome tic-tac-toe!

Do you want the smart AI to fight on your behalf?
[Y/N] -> y

Select the opponent
r Random
a Average Ai
s Smart AI
>> a
>> Sorry - that position is already taken!
Try again
       | x | x
    -----------
     o | x | o
    -----------
     x | o |
----------------------------------------
Human is the WINNER!!!
----------------------------------------
Play Again?
[Y/N] -> y
Do you want the smart AI to fight on your behalf?
[Y/N] -> y

Select the opponent
r Random
a Average Ai
s Smart AI
>> s
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
```



```
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
     o | x | o
    -----------
     x | x | o
    -----------
     x | o | x
----------------------------------------
TIE!
----------------------------------------
Play Again?
[Y/N] -> y
Do you want the smart AI to fight on your behalf?
[Y/N] -> r

Select the opponent
r Random
a Average Ai
s Smart AI
>> r
```

T3 – LE BAO DUY NGUYEN (102449993)



```
To make a move enter a number between 0 - 8 and press enter.
The number corresponds to a board position as illustrated:
    0 | 1 | 2
    ---------
    3 | 4 | 5
    ---------
    6 | 7 | 8

----------------------------------------
      |   |
   -----------
      |   |
   -----------
      |   |
The current player is: Human
[0-8] >> 0
    x |   |
   -----------
      |   |
   -----------
      |   |
The current player is: Super AI
    x |   | o
   -----------
      |   |
   -----------
      |   |
The current player is: Human
[0-8] >> 1
    x | x | o
   -----------
      |   |
   -----------
      |   |
The current player is: Super AI
    x | x | o
   -----------
      | o |
   -----------
      |   |
The current player is: Human
[0-8] >> 2
>> Sorry - that position is already taken!
Try again
    x | x | o
   -----------
    x | x | o
   -----------
    x | o |
   -----------
      |   |
 The current player is: Super AI
 >> Sorry - that position is already taken!
 Try again
    x | x | o
   -----------
    x | o |
   -----------
      |   |
 The current player is: Super AI
 >> Sorry - that position is already taken!
 Try again
    x | x | o
   -----------
    x | o |
   -----------
      |   |
 The current player is: Super AI
    x | x | o
   -----------
    x | o |
   -----------
      |   | o
 The current player is: Human
 [0-8] >> 4
 >> Sorry - that position is already taken!
 Try again
    x | x | o
   -----------
    x | o |
   -----------
      |   | o
 The current player is: Human
 [0-8] >> 5
    x | x | o
   -----------
    x | o | x
   -----------
      |   | o
 The current player is: Super AI
    x | x | o
   -----------
The current player is: Super AI
    x | x | o
   -----------
    x | o | x
   -----------
    o |   | o
----------------------------------------
Super AI is the WINNER!!!
----------------------------------------
Play Again?
[Y/N] -> n
Goodbye, Thank you for playing.
```