EXTENSION REPORT

By Le Bao Duy Nguyen – 102449993

EXTENSION 1 – T03 – TIC TAC TOE

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

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
            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]


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
    if current_player == 'x':
        move = get_human_move()
```

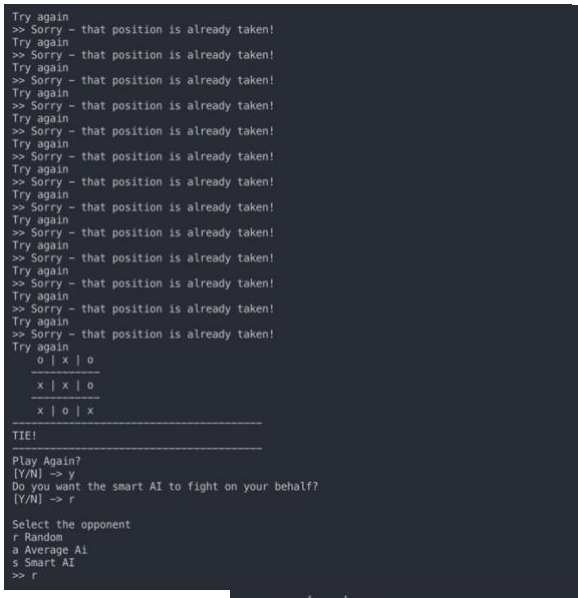
```
    elif ai_choice == 's':
        move = get_smart_ai_move()
    elif ai_choice == 'a':
        move = get_average_ai_move()
    elif ai_choice == 'r':
        move = get_ai_move()
    else:
        move = get_ai_move()  # Defaults to the random AI


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()

    # 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()

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

        # 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()
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.')
```

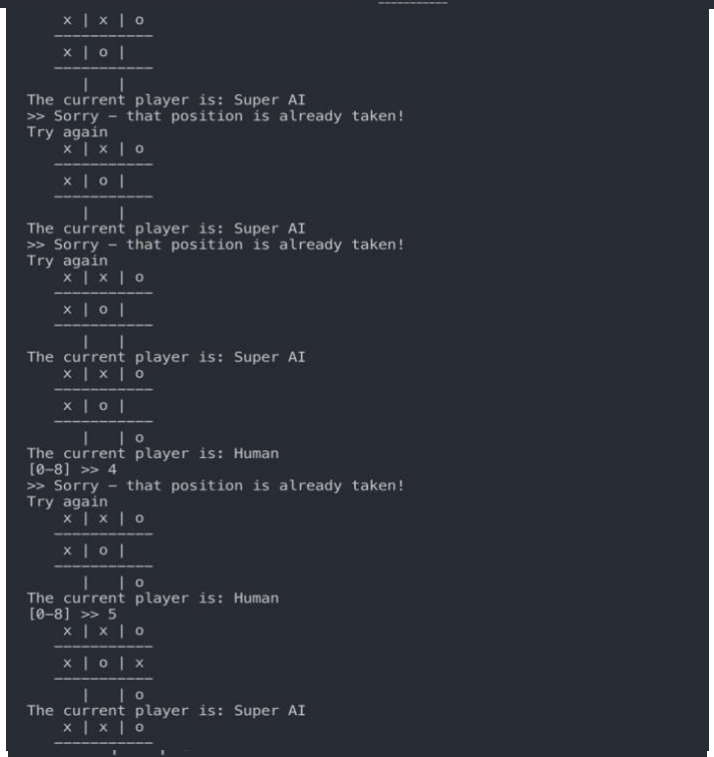
RESULT

```
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!
```

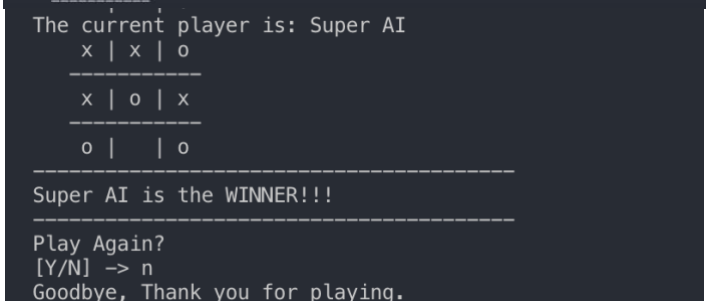


```
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position is already taken!
Try again
>> Sorry - that position 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
```

```
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
   ___________
```

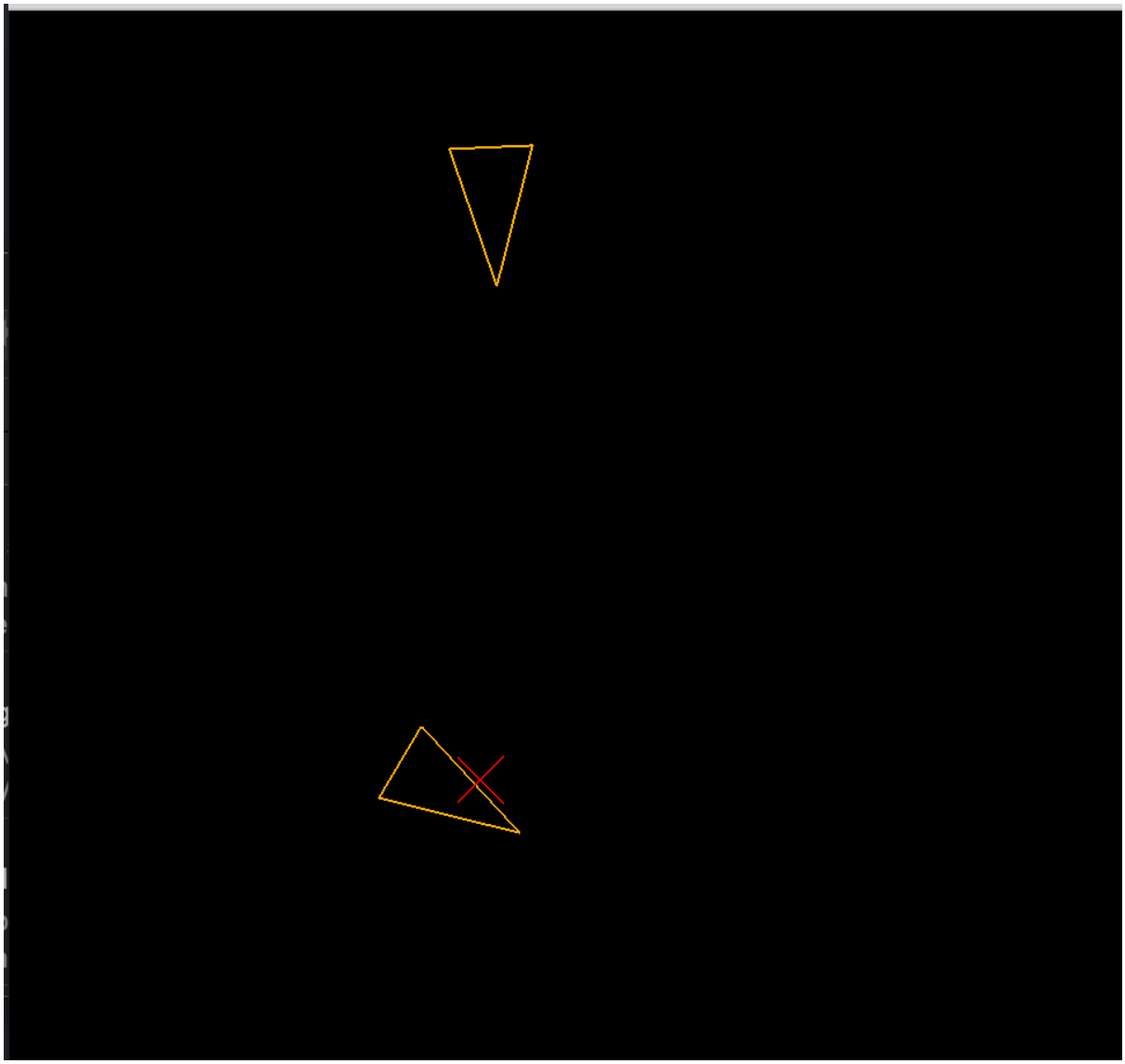


```
The current player is: Super AI
    x | x | o
   ___________
    x | o | x
   ___________
    o |   | o
_________________________________________
Super AI is the WINNER!!!
_________________________________________
Play Again?
[Y/N] -> n
Goodbye, Thank you for playing.
```

EXTENSION 2 – T08 – LAB – STEERING #1 – SEEK ARRIVE FLEE

Pursuit mode is added. Predicting where an agent will be in time T and seeks towards that point to intercept it.

```
def pursuit(self, evader):
self.toEvader = evader.pos - self.pos
    self.relHeading = Vector2D.dot(evader.heading, self.heading)

    if(Vector2D.dot(self.toEvader, self.heading) > 0 and self.relHeading < -0.95):
        return self.seek(evader.pos)

    lookAheadTime = Vector2D.length(self.toEvader) / (self.max_speed + evader.speed())
    return self.seek(evader.pos + evader.vel * lookAheadTime)
```



EXTENSION 3 – T14 – AGENT MARKSMANSHIP

Shotgun mode is added where it sent out waves of controllable number of bullets out. Slow but deadly (guaranteed to hit).

```
if self.world.hunter.aim is True:
            bullet_speed = 20 if self.mode in ['Shotgun'] else 10
            target_pos = self.aim()
```

```
elif self.mode == 'Shotgun':
            for i in range(5):
                self.world.add(ShotgunBullet(self.init_pos, enemy_pos))
```

```
class ShotgunBullet(Bullet):
    def __init__(self, firing_pos, target_pos):
        Bullet.__init__(self, firing_pos, target_pos +
                        Vector2D(randrange(-100, 100), randrange(-100, 100)))
        self.radius = 5
        self.velocity = randrange(18, 22)
```

EXTENSION 4 – T16 - GOAP

A menu is made so that user can choose between options. Each option will guide them on what is their possibility to defeat Thanos. This increases the liveliness of an AI.

```
print(
"""--------------------------------
A: Do you want see the possibility of you saving the universe?
B: Quit --------------------------------""")
option = input("Enter option:")
if option.lower() == "a":
    print("""--------------------------------
A: Steal Gem
B: Steal Gauntlet
C: Kill Thanos
D: Too hard ! Get me out --------------------------------""")
    option = input("Enter your choice:")
    if option.lower() == "a":
        goal_state = 'Has Gem'
        path = agent.plan(goal_state)
        print('Goal: ' + goal_state + "\n")
        for i in range(len((path['Actions']))):
            print(str(i+1) + ') ' + path['Actions'][i].name +' (' + str(path['Actions'][i].cost) + ')')
        print('Win ratio: 1/' + str(path['Ratio']))
        pass

    elif option.lower() == "b":
        goal_state = 'Has Gauntlet'
        path = agent.plan(goal_state)
        print('Goal: ' + goal_state + "\n")
        for i in range(len((path['Actions']))):
            print(str(i+1) + ') ' + path['Actions'][i].name + ' (' + str(path['Actions'][i].cost) + ')')
        print('Win ratio: 1/' + str(path['Ratio']))
        pass

    elif option.lower() == "c":
        goal_state = 'Defeat Thanos'
        path = agent.plan(goal_state)
```

```
        print('Goal: ' + goal_state + "\n")
        for i in range(len((path['Actions']))):
            print(str(i+1) + ') ' + path['Actions'][i].name +' (' + str(path['Actions'][i].cost) + ')')
        print('Win ratio: 1/' + str(path['Ratio']))
        pass

    elif option.lower() == "d":
        print("------------------------------")
        print("Understandable choice!")
        pass
elif option.lower() == "b":
    print("------------------------------")
    print("Understandable choice!")
    pass
else:
    print("You can only selct A or B")
    print("Please try again")
    pass
```

```
Kevins-MacBook-Air-2:16 - Spike - GOAP kevinnguyen2208$ python3 goap_Thanos.py
--------------------------------
A: Do you want see the possibility of you saving the universe?
B: Quit
--------------------------------
Enter option:a
--------------------------------
A: Steal Gem
B: Steal Gauntlet
C: Kill Thanos
D: Too hard ! Get me out
--------------------------------
Enter your choice:a
Goal: Has Gem

1) get Gem (600860)
Win ratio: 1/600860
Kevins-MacBook-Air-2:16 - Spike - GOAP kevinnguyen2208$ python3 goap_Thanos.py
--------------------------------
A: Do you want see the possibility of you saving the universe?
B: Quit
--------------------------------
Enter option:a
--------------------------------
A: Steal Gem
B: Steal Gauntlet
C: Kill Thanos
D: Too hard ! Get me out
--------------------------------
Enter your choice:c
Goal: Defeat Thanos

1) get Gem (600860)
2) get Gauntlet (5412989)
3) fight Thanos (7986756)
Win ratio: 1/14000605
Kevins-MacBook-Air-2:16 - Spike - GOAP kevinnguyen2208$ python3 goap_Thanos.py
--------------------------------
A: Do you want see the possibility of you saving the universe?
B: Quit
--------------------------------
Enter option:b
--------------------------------
Understandable choice!
Kevins-MacBook-Air-2:16 - Spike - GOAP kevinnguyen2208$
```