

בניית משחק
איקס עיגול

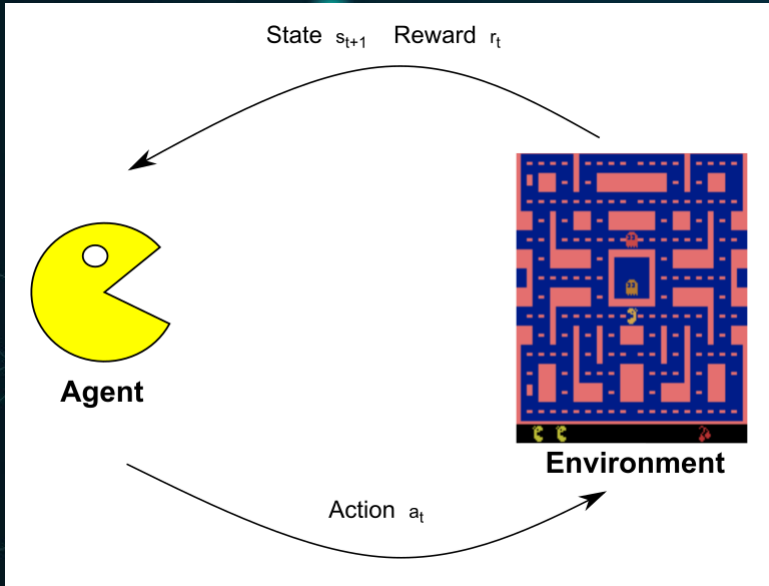
גלעד מרקמן



קריית החינוך
פארק המדע
בית לערכים
למצוינות ולחדשנות

בניית משחק לפי מודל סביבה לקוח

- נדגים בניית משחק בהתאם למודל סביבה לקוח באמצעות בניית משחק איקס עיגול.



- המשחק יכול את המחלקות הבאות:

• סביבה (Environment) : Tic_Tac_Toe

• סוכן (Agent) : Human_Agent

• מצב: state

• פעולה (action) : Tuple (row, col)

• גרפיקה: Graphics

• Game : התוכנית הראשית עם לולאת המשחק.

מחלקות התוכנית

Environment: Tic_Tac_Toe		
מאפיין	מצב – state	המצב הנוכחי של הסביבה
פונקציה	Move(action)	הפונקציה משנה את המצב הנוכחי בהתאם לפעולה שמקבלת.
פונקציה	isLegal(action)	הפונקציה בודקת אם הפעולה חוקית
פונקציה	isEndOfGame(state)	הפונקציה מחזירה אמת אם המצב הינו סוף משחק.
פונקציה	Next_state(state, action)	הפונקציה מקבלת מצב ופעולה ומחזירה את המצב הבא (בדומה ל- move הפועלת על המצב של הסביבה)

state		
מאפיין	Board	מערך המתאר את מצב המשחק
מאפיין	Player	תור של מי לשחק
פעולה	Init_board()	יצירת מצב התחלתי של המחשב

Action = Tuple (row, col)

מחלקות התוכנית - המשך

Human_Agent		
מאפיין	סביבה – env	הסביבה בה פועל הסוכן
פעולה	get-action(state)	הסוכן מקבל מצב מסויים ומחזיר את הפעולה אותה הוא בוחר לבצע.

Graphics		
מאפיין	Screen	מסך ראשי של PyGame
פעולה	Draw(state)	הפעולה מקבלת מצב ומדפיסה אותו על המסך
פעולה	Write(text)	פעולה כותבת טקסט על המסך
פעולה	calc_row_col(pos)	פעולה מקבלת מיקום העכבר ומחזירה את השורה והעמודה המתאימים למיקום
פעולה	Calc_pos(row_col)	הפעולה מקבלת שורה ועמודה בלוח ומחזירה את המיקום במשטח pyGame.

תוכנית ראשית - Game

```
1 import pygame
2 from Graphics import *
3
4 pygame.init()
5 clock = pygame.time.Clock()
6
7 def main ():
8     run = True
9
10    while (run):
11        events = pygame.event.get()
12        for event in events:
13            if event.type == pygame.QUIT:
14                run = False
15
16        clock.tick(FPS)
17
18 if __name__ == '__main__':
19     main()
20
```

```
1 FPS = 60
2
3 WIDTH, HEIGHT = 300, 400
4 ROWS, COLS = 3, 3
5 SQUARE_SIZE = 100
6 LINE_WIDTH = 2
7
8 H_WIDTH, H_HEIGHT = 300, 100
9 M_WIDTH, M_HEIGHT = 300, 300
10
11 #RGB
12 RED = (255, 0, 0)
13 WHITE = (255, 255, 255)
14 BLACK = (0, 0, 0)
15 BLUE = (0, 0, 255)
16 LIGHTGRAY = (211,211,211)
17 GREEN = (0, 128, 0)
18 CADETBLUE1 = (152,245,255)
19
```

לוח המשחק - Graphics

```
import pygame
from Graphics import *

pygame.init()
clock = pygame.time.Clock()
graphics = Graphics()

def main ():
    run = True

    while (run):
        events = pygame.event.get()
        for event in events:
            if event.type == pygame.QUIT:
                run = False

        graphics()
        clock.tick(FPS)
```

```
class Graphics:
    def __init__(self):
        self.screen = pygame.display.set_mode((WIDTH, HEIGHT))
        self.header_surf = pygame.Surface((H_WIDTH, H_HEIGHT))
        self.main_surf = pygame.Surface((M_WIDTH, M_HEIGHT))
        pygame.display.set_caption('Tic Tac Toe')

    def draw (self):
        self.header_surf.fill(CADETBBLUE1)
        self.main_surf.fill(LIGHTGRAY)
        self.screen.blit(self.header_surf, (0,0))
        self.screen.blit(self.main_surf, (0,100))
        pygame.display.update()

    def __call__(self):
        self.draw()
```

לוח המשחק - המשך

```
def draw (self):  
    self.header_surf.fill(CADETBBLUE1)  
    self.main_surf.fill(LIGHTGRAY)  
  
    self.draw_Lines()  
  
    self.screen.blit(self.header_surf, (0,0))  
    self.screen.blit(self.main_surf, (0,100))  
    pygame.display.update()
```

```
def draw_Lines(self):  
    for i in range(ROWS):  
        pygame.draw.line(self.main_surf, BLACK, (i * SQUARE_SIZE, 0),  
                        (i * SQUARE_SIZE , WIDTH), width=LINE_WIDTH)  
        pygame.draw.line(self.main_surf, BLACK, (0, i * SQUARE_SIZE),  
                        (HEIGHT, i * SQUARE_SIZE ), width=LINE_WIDTH)
```

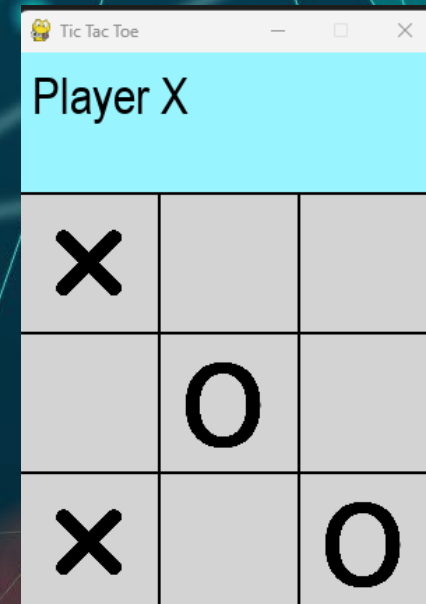
Environment & state

```
import numpy as np

class State:
    def __init__(self, board = None, player = 1):
        if board is not None:
            self.board = board
        else:
            self.board = self.init_board()
        self.player = player
        self.end_of_game = 0

    def init_board(self):
        board = np.zeros((3,3))
        return board
```

```
class TicTacToe:
    def __init__(self, state):
        self.state = state
```





יצירת אובייקט סביבה בתוכנית הראשית

```
import pygame
from Graphics import *
from TicTacToe import TicTacToe
from State import State

pygame.init()
clock = pygame.time.Clock()
graphics = Graphics()
env = TicTacToe(State())

def main ():
    run = True

    while (run):
        events = pygame.event.get()
        for event in events:
            if event.type == pygame.QUIT:
                run = False

        graphics(env.state)
        clock.tick(FPS)
```

הדפסת הלווח בהתאם ל state

```
def __call__(self, state):  
    self.draw(state)
```

```
def __init__(self):  
    self.screen = pygame.display.set_mode((WIDTH, HEIGHT))  
    self.header_surf = pygame.Surface((H_WIDTH, H_HEIGHT))  
    self.main_surf = pygame.Surface((M_WIDTH, M_HEIGHT))  
    pygame.display.set_caption('Tic Tac Toe')  
    self.load_img()
```

```
def load_img (self):  
    x_img = pygame.image.load("img/x_img.png")  
    o_img = pygame.image.load("img/o_img.png")  
    self.x_img = pygame.transform.scale(x_img, (80, 80))  
    self.o_img = pygame.transform.scale(o_img, (80, 80))
```

```
▼ TIC_TAC_TOE_2  
  > __pycache__  
  ▼ img  
    o_img.png  
    x_img.png  
  Game.py  
  Graphics.py  
  State.py  
  Tic_Tac_Toe.py
```

הדפסת הלוח בהתאם ל state

```
def draw_pieces(self, state: State):  
    board = state.board  
    for row in range(ROWS):  
        for col in range(COLS):  
            if board[row,col]!=0:  
                self.draw_piece((row,col), board[row,col])
```

```
def draw_piece(self, row_col, player):  
    if player == 1:  
        img = self.x_img  
    elif player == -1:  
        img = self.o_img  
  
    x, y = self.calc_pos(row_col)  
  
    self.main_surf.blit(img,(x+10, y+10))
```

```
def calc_pos (self, row_col):  
    row, col = row_col  
    x = col * SQUARE_SIZE  
    y = row * SQUARE_SIZE  
    return x, y
```

```
def draw (self, state: State):  
    self.header_surf.fill(CADETBLUE1)  
    self.main_surf.fill(LIGHTGRAY)  
  
    self.draw_Lines()  
  
    self.draw_pieces(state)  
  
    self.screen.blit(self.header_surf, (0,0))  
    self.screen.blit(self.main_surf, (0,100))  
    pygame.display.update()
```

הוספת כיתוב – תור מי לשחק

```
def draw (self, state: State):  
    self.header_surf.fill(CADETBBLUE1)  
    self.main_surf.fill(LIGHTGRAY)  
  
    self.draw_Lines()  
    self.draw_pieces(state)  
  
    if state.player == 1:  
        self.write('Player X')  
    else:  
        self.write('Player O')  
  
    self.screen.blit(self.header_surf, (0,0))  
    self.screen.blit(self.main_surf, (0,100))  
    pygame.display.update()
```

```
def write(self, txt):  
    font = pygame.font.SysFont("Arial", 36)  
    txt_surf = font.render(txt, True, BLACK)  
    self.header_surf.blit(txt_surf, (10,10))
```

