Team 5 [Mahmoud Adel & Yomna Magdy]

Facilitators: [Nadine Essam & Ahmed Emad]

Game:

Mouse Escaping

- **Setup the game:**

```python
import pygame
import random
import time

pygame.display.init()

# Variables
height = 700
width = 700
window = pygame.display.set_mode((height, width))
move = [[100, 0], [0, 100], [-100, 0], [0, -100]]
step = 0
x_cat = 0
x_mouse = 0
y_cat = 0
y_mouse = 0
# Colours
blue = (30, 62, 75)
white = (255, 255, 255)
green = (0, 255, 0)
red = (255, 0, 0)
grey = (128, 128, 128)
window.fill(blue)
pygame.draw.rect(window, blue, (0, 0, 100, 100))
pygame.display.update()
pygame.display.set_caption("Welcome to our first game")
```

In This Code:

1. First, We Import These Libraries [pygame, random, time]
2. We initialized pygame modules
3. We gave some variables to objects to make it easy for us to use them

4. We fill the window with blue
5. We set caption to our game

```python
# Load Cat
cat_img = pygame.image.load("cat.png")
cat = pygame.transform.scale(cat_img, (50, 50))

# Load Mouse
m = pygame.image.load("mouse.png")
mouse = pygame.transform.scale(m, (50, 50))

# Load Background Music
pygame.mixer.init(44100, -16, 2, 2048)
pygame.mixer.music.load('background.wav')
pygame.mixer.music.play(-1)
# Load background image
background_image = pygame.image.load("background.png")
background = pygame.transform.scale(background_image, (height, width))

# Load win img
win_image = pygame.image.load("win.jpg")
win = pygame.transform.scale(win_image, (height, width))

# Load Move Sound and Win Sound
pygame.mixer.init(44100, -16, 2, 2048)
move_sound = pygame.mixer.Sound("laser.wav")
win_sound = pygame.mixer.Sound("win_sound.mp3")

# load Start image
s = pygame.image.load("startgame.png")
start = pygame.transform.scale(s, (height, width))
```

In This Code:

1. We Loaded All media needed to play the game [Cat, Mouse, Music, Images, Sound]
2. We play the background music
3. We gave variables to images to use them

```python
# Animals'position
def animal_position():
    global x_cat, x_mouse, y_cat, y_mouse
    list_x_cat = [100, 200, 300, 400, 500]
    list_y_cat = [100, 200, 300, 400, 500]
    while True:
        x_cat = random.choice(list_x_cat)
        y_cat = random.choice(list_y_cat)
        if x_cat != 500 and y_cat != 300:
            break
    list_x_mouse = [100, 200, 300, 400, 500]
    list_x_mouse.remove(x_cat)
    list_y_mouse = [100, 200, 300, 400, 500]
    list_y_mouse.remove(y_cat)
    x_mouse = random.choice(list_x_mouse)
    y_mouse = random.choice(list_y_mouse)


# Exit
def Exit():
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            exit()
```

In This Code:

1. We make a function for animal position [Cat and Mouse]
2. Mouse and Cat take a random place at the first of the game
3. We made the cat can't take the place that front of the win block
4. We made the mouse can't take the cat place at the first of the game [By removing the positions of the cat]
5. We made a function for exit the game

```python
# Draw board
def draw_board():
    y = 100
    for repeat in range(5):
        for x in range(100, 600, 100):
            pygame.draw.rect(window, white, [x, y, 99, 99])
            pygame.display.update()
        y += 100
    pygame.draw.rect(window, green, [600, 300, 99, 99])
    pygame.display.update()


# Draw Cat rect
def draw_cat():
    pygame.draw.rect(window, red, [x_cat, y_cat, 99, 99])
    window.blit(cat, (x_cat + 25, y_cat + 25))
    pygame.display.update()


# Draw Mouse
def draw_mouse():
    pygame.draw.rect(window, grey, [x_mouse, y_mouse, 99, 99])
    window.blit(mouse, (x_mouse + 25, y_mouse + 25))
    pygame.display.update()
```

In This Code:

1. **We draw the components of our game [Grid, Cat, Mouse] and put them in functions**

```python
# 2 types of mouse movement

def manual_move():
    global x_mouse, y_mouse, step
    key = pygame.key.get_pressed()
    if key[pygame.K_UP]:
        y_mouse -= 100
    if key[pygame.K_DOWN]:
        y_mouse += 100
    if key[pygame.K_RIGHT]:
        x_mouse += 100
    if key[pygame.K_LEFT]:
        x_mouse -= 100
    move_sound.play()

def random_move():
    x_pos, y_pos = random.choice(move)
    global x_mouse, y_mouse, step
    x_mouse += x_pos
    y_mouse += y_pos
    move_sound.play()
    step += 1
    time.sleep(.5)
```

In This Code:

1. **We make the movements of the mouse in functions to use them later [Two options: Random and Manual]**
2. **We add sound after moving**

```python
# start game using 2 ways
def start_random():
    global step
    while True:
        while True:
            window.fill(blue)
            draw_board()
            draw_cat()
            draw_mouse()
            random_move()
            time.sleep(1)
            Exit()
            # Conditions
            # If Mouse out borders
            if (x_mouse > 550 and y_mouse != 300) or (x_mouse < 100) or (y_mouse > 550) or (y_mouse < 100):
                draw_mouse()
                random_move()
                pygame.display.update()
                time.sleep(1)
                break

            if step == 20:
                draw_mouse()
                random_move()
                pygame.display.update()
                time.sleep(1)
                break
```

```python
# If  Mouse Set on Cat
if x_mouse == x_cat and y_mouse == y_cat:
    draw_mouse()
    random_move()
    pygame.display.update()
    time.sleep(1)
    break
```

In This Code:

We make a function to use it later when the user chooses the random scenario.

1. Some function we explain them before.
2. We put some conditions to the mouse:
3. If the mouse went out to border, or moved 20 steps without reaching to the win block, or touched the cat, he will lose and the game will show a loser background.

```python
        # If Set on Block win
        if x_mouse == 600 and y_mouse == 300:
            pygame.display.update()
            time.sleep(2)
            break


    # If Lost
    while not (x_mouse == 600 and y_mouse == 300):
        if not (pygame.key.get_pressed()[pygame.K_SPACE]):
            pygame.mixer.music.stop()
            window.blit(background, (0, 0))
            pygame.display.update()
        else:
            animal_position()
            window.fill(blue)
            draw_board()
            draw_cat()
            draw_mouse()
            random_move()
            pygame.mixer.music.play(-1)
            step = 0
            break


    # Exit
    Exit()
```

In This Code:

1. If the mouse reaches the win block he will win and break this function and start the winning function [we will explain it later]
2. We made a chance for the user to play again by pressing SPACE.

```python
# If Win
while x_mouse == 600 and y_mouse == 300:
    if not (pygame.key.get_pressed()[pygame.K_SPACE]):
        pygame.mixer.music.stop()
        win_sound.play()
        window.blit(win, (0, 0))
        pygame.display.update()
    else:
        animal_position()
        window.fill(blue)
        draw_board()
        draw_cat()
        draw_mouse()
        random_move()
        pygame.mixer.music.play(-1)
        step = 0
        break
Exit()
```

In This Code:

1. If the user win, the winner background will start with its sound.
2. We made a chance to the user to play again by pressing SPACE.

```python
def start_manually():
    global step
    while True:
        while True:
            window.fill(blue)
            draw_board()
            draw_cat()
            manual_move()
            draw_mouse()
            time.sleep(.6)
            Exit()
            # Conditions
            # If Mouse out borders
            if (x_mouse > 550 and y_mouse != 300) or (x_mouse < 100) or (y_mouse > 550) or (y_mouse < 100):
                draw_mouse()
                manual_move()
                pygame.display.update()
                time.sleep(.5)
                break
            if step == 20:
                draw_mouse()
                manual_move()
                pygame.display.update()
                time.sleep(.5)
                break
```

In This Code:

1. **This is function for manual moving.**
2. **All Codes Like Random moving....**

```python
            # If  Mouse Set on Cat
            if x_mouse == x_cat and y_mouse == y_cat:
                draw_mouse()
                manual_move()
                pygame.display.update()
                time.sleep(.5)
                break

            # If Set on Block win
            if x_mouse == 600 and y_mouse == 300:
                pygame.display.update()
                time.sleep(1.5)
                break
```

In This Code:

1. There conditions like Random functions....

```python
# If Lost
while not (x_mouse == 600 and y_mouse == 300):
    if not (pygame.key.get_pressed()[pygame.K_SPACE]):
        pygame.mixer.music.stop()
        window.blit(background, (0, 0))
        pygame.display.update()
    else:
        animal_position()
        window.fill(blue)
        draw_board()
        draw_cat()
        draw_mouse()
        manual_move()
        pygame.mixer.music.play(-1)
        step = 0
        break

    # Exit
    Exit()
```

In This Code:

1. These codes like random function.... [we made a chance to the user to play again]

```
# If Win
while x_mouse == 600 and y_mouse == 300:
    if not (pygame.key.get_pressed()[pygame.K_SPACE]):
        pygame.mixer.music.stop()
        win_sound.play()
        window.blit(win, (0, 0))
        pygame.display.update()
    else:
        animal_position()
        window.fill(blue)
        draw_board()
        draw_cat()
        draw_mouse()
        random_move()
        pygame.mixer.music.play(-1)
        step = 0
        break
Exit()
```

In This Code:

1. **These Codes Like random function.... [winning scenario and made a chance to the user to play again]**

```python
# main function
def main():
    animal_position()
    flag = True
    while True:
        Exit()
        window.blit(start, (0, 0))
        pygame.display.update()
        if pygame.key.get_pressed()[pygame.K_r]:
            break
        elif pygame.key.get_pressed()[pygame.K_m]:
            flag = False
            break

    if flag:
        start_random()
    else:
        start_manually()

if __name__ == "__main__":
    main()
```

In This Code:

1. This is the main function in the game, The program starts from here.
2. This gave the user a chance to choose between [random moving Or manual moving with (up, down, right, left Keys)]
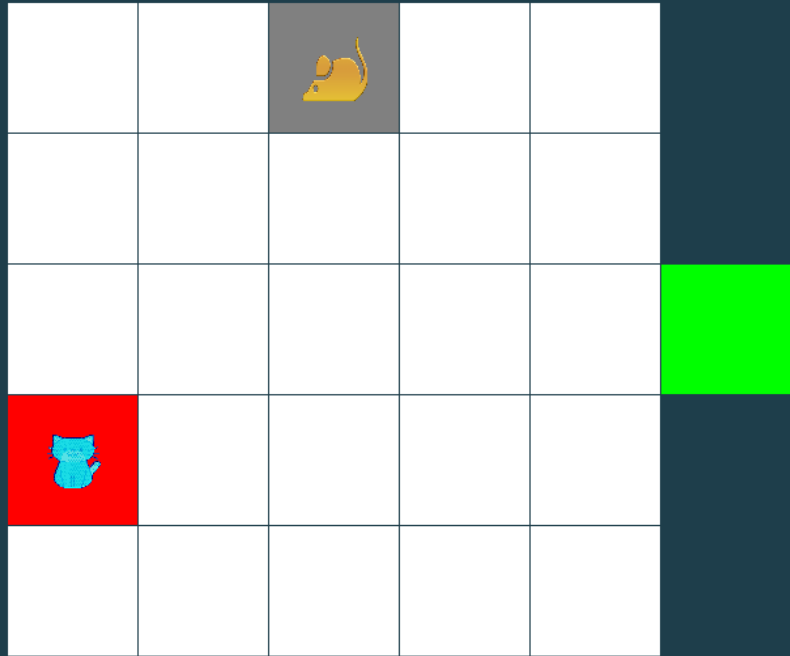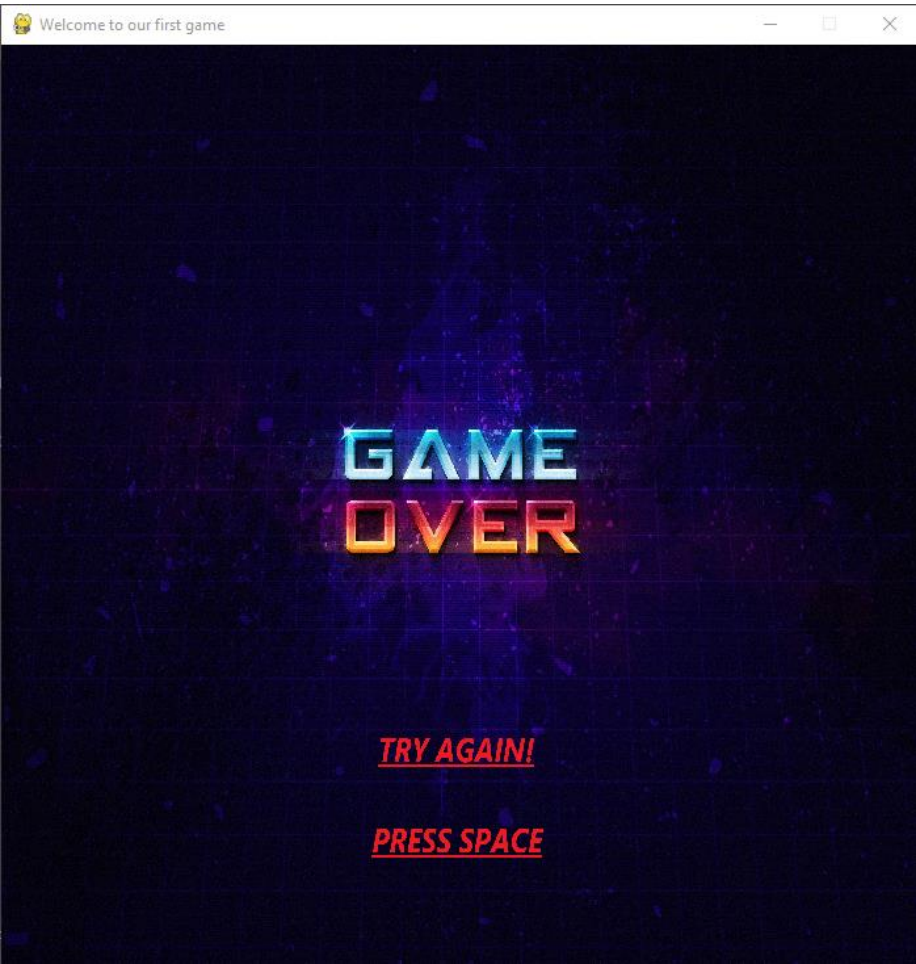
*Some Photos from Game:

GAME
START

Press r to move random
Press m to move manually

Team 5 [Mahmoud Adel & Yomna Magdy]

Facilitators: **[Nadine Essam & Ahmed Emad]**

Game:

Mouse Escaping