9/10/25

Task: 12 Simulate Gaming Concepts using Pygame

Aim:

To create a simple 2D game simulation using the pygame module in python that demonstrates basic gaming concepts such as player movement, enemy motion, collision detection, and event handling

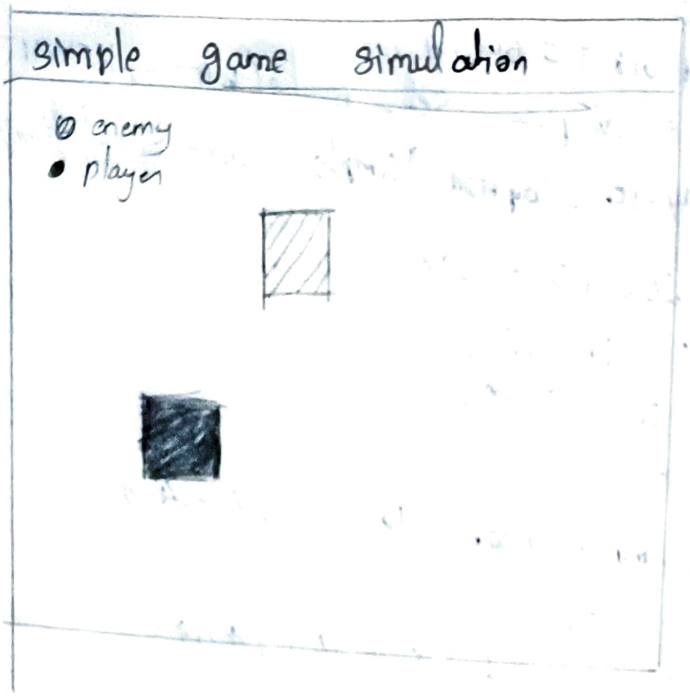
Algorithm:

1. Start the program
2. import pygame and sys module
3. initialize pygame using pygame.init()
4. Create a game window with a title and background color
5. Define player and enemy object using rectangle (pygame.Rect).
6. Handle keyboard input:
   - move player using arrow keys
7. move enemy automatically across the screen
8. Detect collision between players and enemy using colliderect()
   - if they collide, display "Game over". and stop the game loop.
9. Update screen after every frame
10. Control frame rate using pygame.time.Clock().
11. Quit the game properly when closed

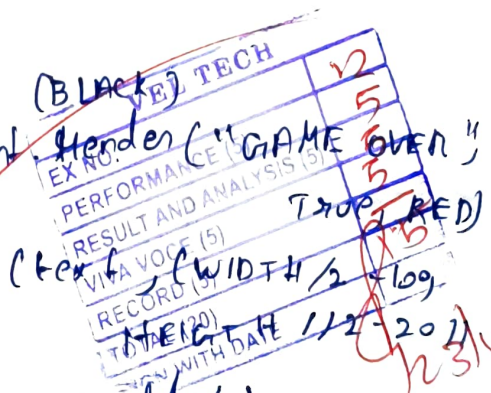
Pygagram:

```
import pygame
import sys

pygame.init()

WIDTH, HEIGHT = 600, 400
win = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("Simple Game Simulation")

WHITE = (255, 255, 255)
BLUE = (50, 150, 250)
RED = (255, 50, 50)
BLACK = (0, 0, 0)
player = pygame.Rect(100, 200, 40, 40)
player_speed = 5
enemy = pygame.Rect(500, 150, 40, 40)
enemy_speed = 4
clock = pygame.time.Clock()
font = pygame.font.SysFont("Arial", 36)

running = True
game_over = False

while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    if not game_over:
        keys = pygame.key.get_pressed()
        if keys[pygame.K_UP] and player.top >
            player.y -= player_speed
```

output :



```
if keys[pygame.K_DOWN] and
    player.bottom < HEIGHT:
    player.x += player_speed
if keys[pygame.K_LEFT] and
    player.left > 0:
    player.x -= player_speed
if keys[pygame.K_RIGHT] and
    player.right < WIDTH:
    player.x += player_speed

enemy.x -= enemy_speed
if enemy.right < 0:
    enemy.left = WIDTH
if player.colliderect(enemy):
    game_over = True

win.fill(WHITE)
pygame.draw.rect(win, BLUE, player)
pygame.draw.rect(win, RED, enemy)

else:
    win.fill(BLACK)
    text = font.render("GAME OVER",
    win.blit(text, [WIDTH/2 - 100,
    pygame.display.update()
    clock.tick(30)
```



Result:

Thus the program successfully demonstrate basic gamic concepts using Pygame