

## Practical Task :- Simulate Gaming concept using concept using Pygame

Aim:- TO simulate Gaming concept using Pygame.

Algorithm:-

- 1) Import Pygame package and initialize it
- 2) Define the window size and title
- 3) Create a snake class which initializes the snake position, color.
- 4) Create a fruit class which initializes the fruit position and color.
- 5) End the game if the user quits or the snake collides with the window

Program:-

```
# import pygame
```

```
import time
```

```
import random
```

Snake - speed = 15

window - x = 720

window - y = 480

black = pygame . color (0, 0, 0)

Score 10

## OUTPUT:-

## cochise

Score: 0

Score: 0

-: 070000014

Digitized by srujanika@gmail.com

মাত্র একটা

мабаня аюзни

sparks - abegg = k

$$OCF = x - w \cancel{+} b \cancel{+} w$$

$\text{CH}_3\text{OH} + \text{H}_2\text{SO}_4 \rightarrow \text{CH}_3\text{OSO}_3\text{H}$

white = pygame.color[255, 255, 255]

red = pygame.color[255, 0, 0]

green = pygame.color[0, 255, 0]

blue = pygame.color[0, 0, 255]

pygame.init()

pygame.display.set\_caption()

FPS = pygame.time.Clock()

snake\_body = [[100, 50],  
[90, 50],  
[80, 50],  
[70, 50]]

fruit\_position = [random.randrange(1, width),  
random.randrange(1, height)]

def game\_over():

my\_font = pygame.font.SysFont("times new roman", 50)

game\_over\_surface = my\_font.render("Game Over", True, red)

"Your Score is: " + str(score), True, red]

game\_window.blit(game\_over\_surface,  
game\_over\_rect)

game\_over\_rect

move-space = move

game-window.fill([black])

for pos in snake-body:

Pygame.draw.rect(game-window, green,  
Pygame.Rect(pos[0], pos[1], 10, 10])

if snake-position[1] < 0 or snake-position[1]  
> window-y-10.,

Pygame.display.update()

Fps.tick(snake-speed)

Result:- Thus ~~A~~ the program has been

verified and proved successful

## Program 2:-

Aim:- Write a Python program to develop a chess board using pygame

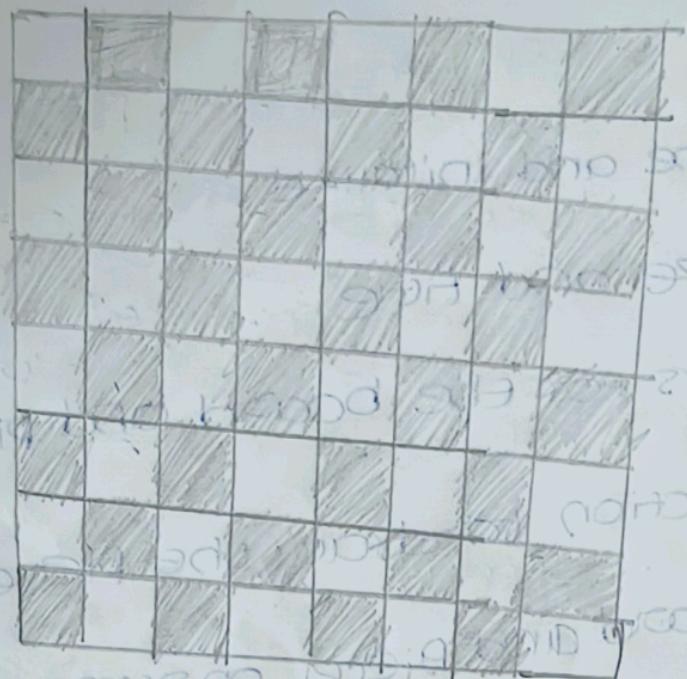
### Algorithm :-

- 1) Import pygame and initialize
- 2) Set screen size and title
- 3) Define colors for the board and pieces
- 4) Define a function to draw the pieces
- 5) Draw the board and pieces on screen
- 6) Start the game loop.

### Program:-

```
import pygame  
pygame.init()  
  
Screen_size = [640, 640]  
Screen = pygame.display.set_mode(Screen_size)  
pygame.display.set_caption('Chess Board')  
  
black = [0, 0, 0]  
white = [255, 255, 255]  
brown = [153, 76, 0]
```

Output :-



```
def draw_board():
    for row in range[8]:
        for col in range[8]:
            square_rect = pygame.Rect(col * 80, row * 80, 80, 80)
    }  
    for row in range[8]:
        for col in range[8]:
            piece = board[row][col]
            if piece != '':
                piece_image = piece_images[piece]
                piece_rect = pygame.Rect(col * 80, row * 80, 80, 80)
    board = [
        ['r', 'n', 'b', 'q', 'k', 'b', 'n', 'r'],
        ['p', 'p', 'p', 'p', 'p', 'p', 'p', 'p'],
        ['.', '.', '.', '.', '.', '.', '.', '.'],
        ['.', '.', '.', '.', '.', '.', '.', '.'],
        ['.', '.', '.', '.', '.', '.', '.', '.'],
        ['.', '.', '.', '.', '.', '.', '.', '.'],
        ['.', '.', '.', '.', '.', '.', '.', '.'],
        ['p', 'p', 'p', 'p', 'p', 'p', 'p', 'p'],
        [r', 'n', 'b', 'q', 'k', 'b', 'n', 'r']
```

draw - piece [board]

- while true:

    For event in pygame.event.get():  
if event.type == pygame.quit:  
    pygame.quit()  
    quit()  
pygame.display.update()

Completed

VELTECH	
EX No.	12
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	15
SIGN WITH DATE	

✓

Result:- Thus, the program has been verified  
And executed successfully