

### \* Program:-

```
#import libraries
import Pygame
import time
import random
snake_Speed = 15
window_x = 720
window_y = 480
black = Pygame.Color(0,0,0)
white = Pygame.Color(255, 255, 255)
red = Pygame.Color(255, 0, 0)
green = Pygame.Color(0, 255, 0)
blue = Pygame.Color(0, 0, 255)
snake_body = [(0, 50), (10, 50), (20, 50), (30, 50)]
fruit_position = (random.randrange(0, window_x // 10), random.randrange(0, window_y // 10))
direction = 'Right'
score = 0
def show_Score(choice, color, front):
    game_over_rect = game_over.get_rect()
    by game_over_rect
    while True:
        if event.type == Pygame.KEYDOWN:
```

Date: 13/10/25

### TASK-12 Simulate gaming Concepts using program

Aim:- To simulate gaming concepts using program

#### Snake game:-

problem:- write a python program to Create a snake game using Pygame package.

- ① set the window size.
- ② Create a snake
- ③ when the snake hits the fruit increase the directions when left, down and key is pressed.
- ④ when the hits the window Game over.

#### Algorithm:-

- ① Import Pygame package and initialize it.
- ② Define the window size and title.
- ③ Create a fruit class which initializes the fruit position and color.
- ④ Create a function to check if Snake Collides with the fruit and increases the score.
- ⑤ Create a function to check if Snake Collides with window end the game.
- ⑥ Create a fun to create update the snake position based on user input.

if event key == Pygame.K\_UP:

change\_to = 'up'.

if even\_key == Pygame.K\_DOWN:

change\_to = 'Down'.

if direction == 'up':

Snake\_position[1] = 0

if direction\_snake\_dobt

Graving mechanism:

Snake\_body.insert(0, list(Snake\_position) Sceave\_to.

fruit = sprout = False

else:

    snake = body.pop()

if not fruit == Sprout:

    fruit\_position = (random.random() \* width) \* 10, (random.random() \* height) \* 10,

    random.randomrange((width - 10) // 10) \* 10)

    if snap\_position @ block ==

        pass

            if

            else



### Program:-

```
import pygame  
pygame.init()  
screen_size = (800, 600)  
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)  
  
def draw_pieces(board):  
    piece_image = {  
        'k': pygame.image.load('images/rook.png'),  
        'n': pygame.image.load('images/knight.png'),  
        'b': pygame.image.load('images/bishop.png'),  
        'q': pygame.image.load('images/queen.png')  
    }  
  
    for row in range(8):  
        for col in range(8):  
            piece = board[(row, col)]  
            if piece != None:  
                screen.blit(piece_image[piece], (col * 100, row * 100))
```

### Problem-2:-

write a Python program to develop a chess board using Pygame

### Algorithm:-

- ① Import Pygame and initialize it.
- ② set screen size and title.
- ③ Define colors for board and pieces.
- ④ Define a function to draw the board by looping over rows and columns and drawing squares.
- ⑤ Define the initial state of board as list.
- ⑥ Draw the board & pie.

piece\_image = piece\_image(piece)

piece\_rect = Pygame\_rect(101, 80, 100, 80, 101, 80)  
screen.blit(piece\_image, piece\_rect)

board = [

C, 'N', 'B', 'K', 'B', 'N', 'R'

(P, P, P, P, P, P, P)

(., "R", "N", "B", "K", "B", "R")

(., "N", "B", "K", "B", "R", ".")

(., "B", "K", "B", "R", "N", ".")

(P, R, P, R, P, R, P, P, P)

(P, N, B, K, B, N, R)

draw\_board()

draw\_pieces(board)

while True:

for event in Pygame.event.get():

if event.type == Pygame.QUIT:

Pygame.quit()

quit()

Pygame.display.update()

| VEL TECH                |    |
|-------------------------|----|
| EX NO.                  | 12 |
| PERFORMANCE (5)         | 8  |
| RESULT AND ANALYSIS (5) | 8  |
| VIVA VOCE (5)           | 8  |
| RECORDS (5)             | 8  |
| TOTAL (20)              | 8  |
| 5.GRADE WITH DATE       | 8  |

Result:- Thus, the program for Pygame is executed and verified successfully.