

* 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 = [(100,50), (20,50), (80,50), (60,50)]

fruit_position = (random.randrange(1), (window_y/10)*10,
                 random.randrange(1), (window_x/10)*10)

direction = 'Right'

score = 0

def show_score(choice, color, font, size):
    game_over_text = 'Game Over'
    pygame.quit()

    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 direction 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 increase 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 event key == Pygame.K_DOWN:

change_to = 'Down'

if direction == 'up'

snake_position[1] = 10

if direction snake_body

growing mechanism:

snake_body.insert(0) list(snake_position) score += 10

fruit_spawn = False

else:

snake_body.pop()

if not fruit_spawn:

fruit_position = (random.randrange(1, window // 10) * 10,

random.randrange(1, window // 10) * 10)

if snake_position[0] == block:

pygame.quit()

Output

score = 0

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)

def draw_pieces(board):
    piece_image = {
        'k': pygame.image.load('images/king.png'),
        'n': pygame.image.load('images/knight.png'),
        'b': pygame.image.load('images/bishop.png'),
        'a': pygame.image.load('images/queen.png')
    }
```

```
for row in range(8):
    for col in range(8):
        piece = board[row][col]
```

```
if piece != "":
```

```
    for col in range(8):
        for row in range(8):
```

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 piece.
- 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, 80, 80)
screen.blit(piece_image, piece_rect)

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_board()

draw_pieces(board)

while True:

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

break

pygame.display.update()

VEL TECH	
Roll No.	
PERFORMANCE (%)	
RESULT AND ANALYSIS (%)	
VIVA VOCE (%)	
RECORD (%)	
TOTAL (%)	
SIGNATURE	
DATE	

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