

## Task No:12

### Simulate Gaming Concepts using pygame

Aim:- To Simulate Gaming Concepts using pygame

Problem:- write a python program to create a Snake Game using pygame package.

Condition:-

1. Set the window size.
2. Create a snake
3. Make the snake to move the directions when left, right and up key is pressed
4. when the snake hits the fruit increase the score by 10
5. if the snake hits the window Game over.

Algorithm:-

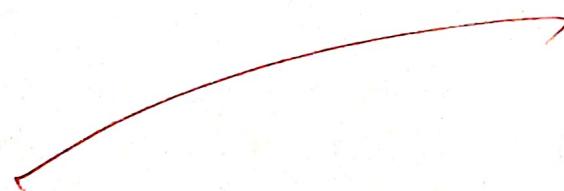
1. Import pygame and initialize it
2. Define the window size and title
3. Create a snake class which initializes the snake position, color, and movement.
4. Create a fruit class which initializes the fruit position and colour.
5. Create a function to check if the snake collides with the fruit and increase the score.
6. Create a function to check if the snake collides with window and the end game.
7. Create a function to update the snake position based on user input
8. Create a function to update the game display and draw the snake and fruit.
9. Create a game loop to continuously update the game display, snake position.
10. End the game if the user quits or the snake collides with the window.

```
program
# importing libraries
import pygame
import time
import random
Snake_Speed = 15
# window size
window_x = 720
window_y = 480
# defining Colors
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)
# initialising pygame
pygame.init()
# initialise game window
Pygame.display.set_caption('Greeks for Greeks Snakes')
game_window = pygame.display.set_mode((window_x,window_y))
# FPS (frame per second) controller
fps = Pygame.time.Clock()
# defining Snake default position
Snake_body = [[100,50],
              [90,50], [80,50], [70,50]]
# fruit position
fruit_position = [random.randrange(1,(window_x/10))*10, random.randrange(1,(window_y/10))*10]
fruit_spawn = True
# Setting default snake direction towards
# right
direction = 'right'
Change_to = direction
# initial score
Score = 0
# display Score function
def show_score(choice,color,font,Size):
    score = font.render("Score : " + str(Score),True,color)
    game_window.blit(score,[10,10])
```

```
# creating font object score-font
score-font = pygame.font.SysFont(font, size)
# Create the display surface object
# SCORE-Surface
Score-Surface = Score-front.render('Score: ' + str(score), True, color)
# Create a rectangular object for the text
# Surface object
Score-rect = Score-Surface.get_rect()
# display text
game-window.blit(Score-Surface, Score-rect)
# game over - function
def game-over():
    # creating front object my-font
    my-font = pygame.font.SysFont('times new roman', 50)
    # creating a text surface on which text
    # will be drawn
    game-over-Surface = my-front.render('Your Score is: ' + str(score), True, red)
    # Create a rectangular object for the text
    # Surface object
    game-over-rect = game-over-Surface.get_rect()
    # Setting position of the text
    game-over-rect.midtop = (window-x/2, window-y/2)
    # blit will draw the text on screen
    pygame.display.flip()
    # after 2 Seconds we will quit the program
    time.sleep(2)
    # deactivity pygame library
    pygame.quit()
    # quit the program
    quit()
    # Main function
    while True:
```

```
# handling key events
for event in pygame.event.get():
    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_UP:
            Change_to = 'Up'
        if event.key == pygame.K_DOWN:
            Change_to = 'Down'
        if event.key == pygame.K_LEFT:
            Change_to = 'left'
        if event.key == pygame.K_RIGHT:
            Change_to = 'Right'
# if two keys pressed simultaneously
# we don't want snake to move into two
# direction simultaneously
if Change_to == 'Up' and direction != 'Down':
    direction = 'Down'
if Change_to == 'Down' and direction != 'Up':
    direction = 'Up'
if Change_to == 'Left' and direction != 'Right':
    direction = 'Right'
if Change_to == 'Right' and direction != 'Left':
    direction = 'Left'
# Moving the snake
if direction == 'Up':
    Snake_position[1] -= 10
if direction == 'Down':
    Snake_position[1] += 10
if direction == 'Left':
    Snake_position[0] -= 10
if direction == 'Right':
    Snake_position[0] += 10
# Snake body growing mechanism
# if fruits and snakes collide then score
# will be increased by 10
Snake_body.insert(0, list(snake_position))
if snake_position[0] == fruit_position[0] and snake_position[0] == fruit_position[1]:
    Score += 10
```

```
fruit-spawn = True  
else:  
    snake-body.pop()  
    if not fruit-spawn:  
        fruit-position = [random.randrange(1, (window-x/10)) * 10, random.  
                           randrange(1, (window-y//10) * 10)]  
    fruit-spawn = True  
game-window.fill(black)  
for pos in snake-body:  
    pygame.draw.rectangle(game-window, green, pygame.Rect(  
                           pos[0], pos[1], 10, 10))  
pygame.draw.rect(game-window, white, pygame.Rect(fruit-  
                           position[0], fruit-position[1] 10, 10))  
# Game over Conditions  
if snake-position[0] < 0 or snake-position[0] > window-x-10;  
    game-over()  
if snake-position[1] < 0 or snake-position[1] > window-y-10;  
    game-over()  
# Touching the Snake body  
for block in snake-body[1:]:  
    if snake-position[0] == block[0] and snake-position[1] == block[1]:  
        game-over()  
# displaying score Continuously  
show-score(1, white, 'times new roman', 20)  
# Refresh game screen  
pygame.display.update()  
# frame per second / Refresh Rate  
fps.tick(snake-speed)
```



## Output :-

Score: 0

problem? 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 the board and pieces  
Define function to draw the board by looping over rows and columns and drawing squares of different colors
4. Define a function to draw the pieces on the board by loading images for each images for each piece and placing them on the corresponding square
5. Define the initial state of the board as the list of lists containing the pieces
6. Draw the board and pieces on the screen

```
# program
import pygame
# Initialize program
pygame.init()
# Set screen size and title
pygame.init() ScreenSize=(640,640)
# Set screen size
Screen=pygame.display.set_mode(ScreenSize)
pygame.display.set_caption("Chess Board")
# Define colors
black=(0,0,0)
white=(255,255,255)
brown=(153,76,0)
# Define function to draw the board
def draw_board():
    for row in range(8):
        for col in range(8):
```

Square-Color = white if (row+col) / 2 == 0 else brown

Square-rect = pygame.Rect(col\*80, row\*80, 80, 80)

pygame.draw.rect(screen, Square-Color, Square-rect)

# Define function to draw the pieces

def draw-pieces (board):

piece-images = {

'r': 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'),

'k': pygame.image.load('images/king.png'),

'p': pygame.image.load('images/pawn.png'),

}

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)

Screen.blit(piece-image, piece-rect)

# Define initial state of the board

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']]

]

output value will be displayed.

1. What does (some) bring about?

1960-1961 学年第一学期

卷之三

10. *On the other hand, the author's* [unclear]  
[unclear] *is not mentioned.*

*1903-04-03* *1903-04-03*

~~1923-24~~ 1924-25

~~As a rule, the best way to get rid of a bad habit is to replace it with a good one.~~

2014-03-07 10:22:45

1942-1975

```
#Draw board and pieces  
draw_board()  
draw_pieces(board)  
# Start game loop  
while True:  
    for event in pygame.event.get():  
        if event.type == pygame.quit():  
            pygame.quit()  
            quit()  
    pygame.display.update()
```

H

VELTECH	
EX No.	12
PERFORMANCE (5)	5
RESULT AND ANALYSIS (3)	5
VIVA VOCE (3)	5
RECORD (4)	
TOTAL (15)	
SIGN WITH DATE	15

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