

Task 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.

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, colour, and movement
4. Create a fruit class which initializes the fruit position and colour.
5. Create a function to check if snake collides with the window and end the game
6. Create a function to update the snake position based on the user input
7. Create a function to update the game display and check for collision.
8. End the game if the user quits or the snake collides with window.

Program:

#importing libraries

import time

import pygame

import random

snake_speed = 15

#window size

window_x = 720

window_y = 480

#defining colour

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('Grecks for Geel's Snakes')
game_window = pygame.display.set_mode((window-x, window-y))
```

```
#FPS (Frames per second) controller.
```

```
fps = pygame.time.Clock()
```

```
#defining snake default position
```

```
snake_position = [100, 50]
```

```
#defining first 4 blocks of snake body
```

```
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
```

```
#displaying score function
```

```
def show_score(choice, colour, font, size)
```

```
#Creating font object score-font
```

```
score_font = pygame.font.SysFont(font, size)
```

```
#create the display surface object
```

```
#score-font = pygame.font.SysFont(font, size)
```

#displaying text

game_window.blit(score_surface, score_rect)

#game over function

def game-over():

#creating font 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-font.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/4)

#blit will draw the text on screen

game_window.blit(game-over_surface, game-over_rect)

pygame.display.flip()

#after 2 seconds will quit the program

time.sleep(2)

#deactivating pygame library

pygame.quit()

#quit the program

quit()

#Main function

while True:

 #handling key events

 for event.type == pygame.KEYDOWN

 if event.key == pygame.K_UP:

 if event.key == pygame.K_UP

 change_to = 'UP'

 if event.key == pygame.K_DOWN

 change_to = 'DOWN'

Score: 6

1902-1903

* (*) very good job

Hand für Hand Hand für Hand

(12.1) $\text{Hom}(V, W) \cong \text{Hom}(W, V)$ if and only if $V \cong W$.

but I do not want to be a woman.

Monday 21 Nov 11

1) $\frac{1}{2} \log \frac{1}{2} = -0.5$

$$(b_{11}, \dots, b_{1n}, (m_{11}, \dots, m_{1n})) + (2, \dots, 2) \in m_{11} \dots m_{1n}$$

~~1. The first step is to identify the problem.~~

the case against

Just as the military police

10-20-32 in text on page 11

(1) gift, paid, gift, money

...the ...

(1992-2000)

11/10/1911

(1) Fig. 2000

nothing at first

() 41

North and North E

Lead slide

shows that pollution is

Handwritten:

[illegible]

Chapman, J. W.

And the answer is 9.

$\frac{1}{2} \times 10 = 5$

```
if event.key == pygame.K_LEFT  
    change_to = 'LEFT'  
if event.key == pygame.K_RIGHT  
    change_to = 'RIGHT'
```

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 fruit and snake collide then scores

will be increment by 10

```
snake_body.insert(0, list(snake_position))
```

```
if snake_position[0] == fruit_position[0] and snake_  
    position[1] == fruit_position[1]:
```

```
    score += 10
```

```
    fruit_spawn = False
```

else:

```
    snake_body.pop()
```

```
if not fruit_spawn:
```

```
    fruit_spawn = True
```

```
game_window.fill(black)
```


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 the board and pieces
- 4) Define a function to draw the pieces on the board by loading images for each piece and placing them on the corresponding square.
- 5) Define the initial state of the board as a list of lists containing the pieces.
- 6) Draw the board and pieces on the screen
- 7) Start the game loop.

Program:

```
import pygame
```

```
# initialize pygame
```

```
pygame.init()
```

```
# set screen size and title
```

```
screen_size = (640, 640)
```

```
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):
```

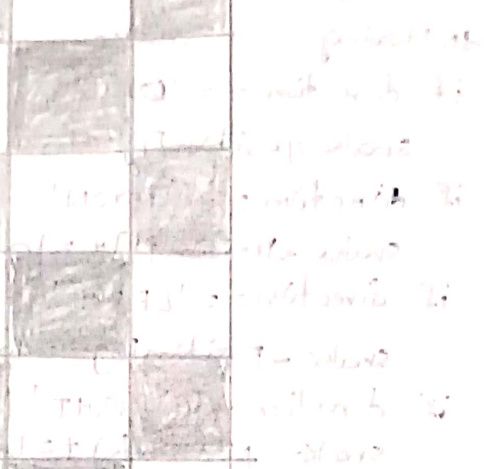
```
            square_color = white if (row + col) % 2 == 0 else brown
```

```
            square_rect = pygame.Rect(col * 80, row * 80, 80, 80)
```

```
            pygame.draw.rect(screen, square_color, square_rect)
```

```
pygame.display.update()
```

[Faint handwritten notes at the bottom of the page, likely bleed-through from the reverse side.]



1. $\{x \in X \mid x \text{ is a point}\}$
 2. $\{x \in X \mid x \text{ is a line}\}$
 3. $\{x \in X \mid x \text{ is a plane}\}$
 4. $\{x \in X \mid x \text{ is a solid}\}$

Scanned with OKEN Scanner

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

Result:

Thus the program for Pygame is executed and verified successfully.