Task-12 - Simulate Gaming Concepts using Pygame

Aim:- To simulate Gaming concepts using Pygame.

SnakeGame:-

Problem 1. write a python program to create a snake Game using pygame Package

Conditions:-

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

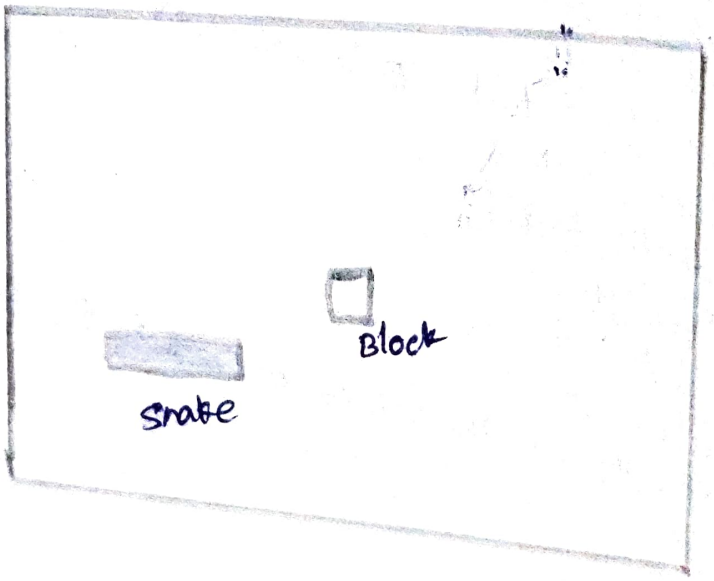
Algorithm:-

1. Important pygame package and initialize it
2. Define the window size and title.
3. create a snake class which initialize the snake position, color, and movement.
4. create a fruit class which initialize the fruit position and color.
5. Create a function to check if the snake collides with the fruit and increase the score.
6. Create a function to update

Progam:-

```
# importing libraries
import pygram
import time.
import random
snake-speed=15
# window size
  window-x=720
  window-y=480
  # defining colors
  # display score function
    def show-score (choice, color, font, size):
  # creating font objects score-front
    score-font objects = pygame. font. sysFont (font, size)
    # Create the display surface object
    # score-surface
    score-surface = score-font. render ('score:' +str(score), the, color)
    # create a rectangular object for the text
    # surface object
```

Output :-



```
score_rect = score_surface.get_rect()
# 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 we 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 in pygame.event.get():
    if event.type == pygame.KEYDOWN:
        change_to = 'DOWN'
        if event.key == pygame.K_LEFT
            change_to = 'LEFT'
        if event.key = pygame.K_RIGHT
            change_to = 'RIGHT'
```

```
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('Geeks for Geeks 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_position = True
# setting default snake direction towards
# right
direction = 'RIGHT'
change_to = direction
# initial score
score = 0
```

```
#If two keys pressed simultaneously
# we don't want snake to move into two
#directions simultaneously.
if change_to == 'UP' and direction != 'DOWN':
    direction = 'UP'
if change_to = 'DOWN' and direction != 'UP':
        direction = 'DOWN'.
if change_to = 'LEFT' and direction != 'RIGHT':
        direction = 'LEFT'
# snake body growing mechanism
# if fruits and snakes collide then scores
# will be incremented 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.insert (0, list, (snake_position))
if snake_position [0] == fruit_position[0] and snake_position [1]
== fruit_position[1]
    serecet = 10
    fruit spawn = False

else:
    snake_body.pop()
if not fruit spawn
    fruit_position = [random.randrange (1, (window_x [1]0))* 10
    fruit_spawn = True
    game_window.fill (black)
    for pos in snake_body:
     pygame.draw rect (game_window, green).
     # Game over conditions
      if snake_position [0] < 0 or snake_position [0] > window.
        game_over()
     # Frame per second / Refresh Rate
        fps.tick (snake_speed).
```

Problem 2:-

write a program to develop a chess boards using pygame.

Algorithm:-

1) import pygame and initialize it.
2) set screen size and title.
3) define colors for the board and peices.
4) define the initial state of the board as a list of lists
5) start the game loop.

Program:-

```
# initialize program
  pygame.init()
# set screen size and title
  screen-size = (640,640)
  screen = pygame.display.set_mode(screen-size)
  pygame=display.set caption('chess board')

  # Define colors
  black = (0,0,0)
  white = (255,255,255)
  brown = (153,760)
  # Define function to draw the board
  def draw_board():
    for row in range(8):
  # define function to draw the pieces
      def drawpieces(board):
    piece-images = {
      'r' pygame.image.load('image(rook.prg'),
      'n' : pygame.image.load('image/knight-prg'),
      'b' : pygame.image.load('image/bishop.png'),
      'q' : pygame.image.load('image/queen.png'),
      'k' : pygame.image.load('images/king.png'),
```

Output:-

o/p

```
# define initial state of 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'],
]
# 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()
```

Result:- Thus program for program is executed and verified sucessfully.

VEL TECH	
EX No.	
PERFORMANCE (5)	12
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	3
TOTAL (20)	
SIGN WITH DATE	