

8/10/23

Task 12. Simulate Gaming Concepts using pygame

Aim:- To Simulate Gaming Concepts using Pygame.

Snake Game

Problem :-

Conditions:-

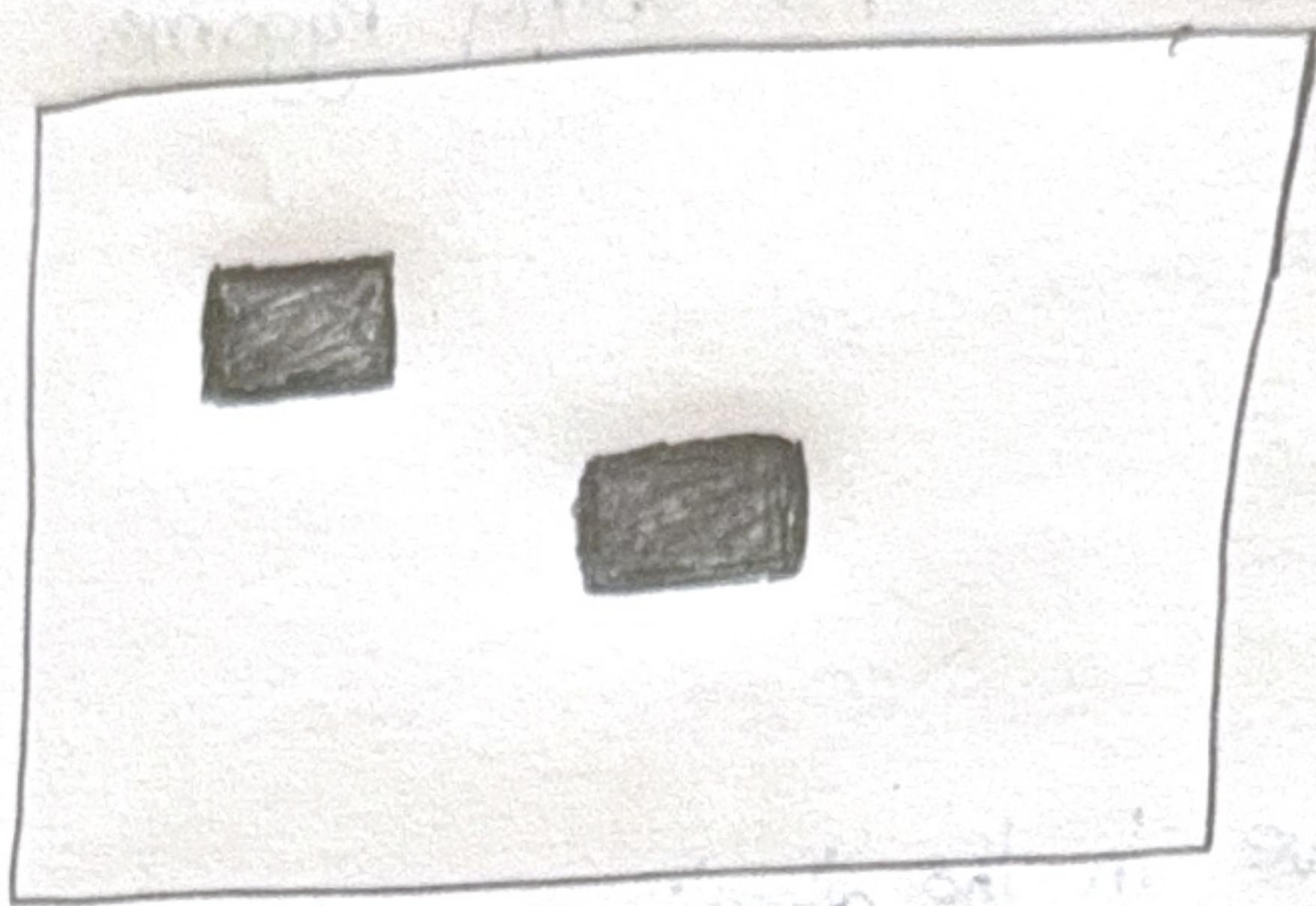
1. Set the window size
2. Create a Snake
3. Make the Snake to move in the directions when Left, right, down and up key is pressed
4. When the Snake hits the fruit, increase Score by 10
5. If the Snake hits the window, Game over

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, color.
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 check if the snake collides with the window and end the game
7. Create a function to update the snake position based on the user input
8. Create a function to update the game display and draw the snake.
9. Create a game loop to continuously update the game display, snake position, and check for collisions
10. End the game if the user quits or the snake collides with the window.

Program

```
# importing libraries  
import pygame  
import time
```



~~OK~~

```
import random
Snake-speed = 15
# Window Size
window-x = 720
window-y = 480
# defining colors
black = pygame.colour(0,0,0)
white = pygame.colour(255,255,255)
red = pygame.colour(255,0,0)
green = pygame.colour(0,255,0)
blue = pygame.colour(0,0,255)
# Initialising pygame
pygame.init()
# Initialise game window
pygame.display.set_caption('Greens for Greens Snakes')
game_window = pygame.display.set_mode((window-x, window-y))
# FPS Controller
fps = pygame.time.Clock()
# defining snake default position
Snake-position = [100,50]
[90,50]
[80,50]
[70,50]
J
# 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,color,font,size):
```

creating font object score-font
score_font = pygame.font.SysFont(font, size):

create the display surface object

score - surface

```
score_surface = score_font.render("Score:" + str(score),  
True, color)
```

create a rectangular object for the text

Surface object

```
score_rect = score_surface.get_rect()
```

displaying text

```
game_window.blit(score_surface, score_rect)
```

game over function

```
def game_over():
```

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_width / 2, window_height / 4)
```

Result:

Thus the program for pygame is executed and verified

VEL TECH		
EX No.	1 (1)	12
PERFORMANCE	100	5
INTERVIEW	95	5
RESULT AND ANALYSIS	95	5
DISCUSSION	95	5
VIVA VOCE	95	5
REPORT	95	5
FIGURES	95	5
DATE	95	5
REMARKS	95	5
GRADE	95	5
DATE OF ISSUE	95	5
RECORDED BY	95	5
VERIFIED BY	95	5
DATE	95	5