“SNAKE GAME”

A

Project Report

Submitted in partial fulfillment

For the award

Degree of Bachelor of technology

in Department of Computer Engineering

****

Aakash paswan (Roll no:19EGKCS001)

Deepak singh (Roll no:19EGKCS011)

Hritik Bhagiya (19EGKCS015)

B TECH (CSE)

RAJASTHAN TECHNICAL UNIVERSITY

Session:2022-2023

Under the supervisior

Of

MRS POOJA SHARMA MAM

Department of computer science & engineering

Gurukul institute of engineering & technology (kota)

CERTIFICATE

This is certify that the project entitled “SNAKE GAME” has been submitted to the Department of Computer Engineering. Gurukul Institute of Engineering & Technology,kota (Rajasthan Technical University ,Kota) for the fulfillment of the requirement for the award of the degree of Bachelor of Technology in “computer Engineering” by Mr. Aakash paswan(19EGKCS001) , Deepak Singh(19EGKCS011), Hritik Bhagiya(19EGKCS015) of final year B TECH (Computer Engineering).

Student Name(with roll no)

Aakash paswan (RNo:19EGKCS001)

Deepak singh (RNo:19EGKCS011)

Hritik Bhagiya(RNo:19EGKCS015

SUPERVISOR NAME HEAD of Department

Mrs Pooja sharma Mam Mrs. Swati jadon mam

ACKNOWLEDGEMENT

We like to share our sincere gratitude to all those who help us in completion of this project. During the work we faced many challenges due to our lack of knowledge and experience but these people help us to get over from all the difficulties and in final compilation of our idea to a shaped sculpture.

We would like to thank Mrs pooja Mam for his governance and guidance, because of which our whole team was able to learn the minute aspects of a project work.

We would also like to show our gratitude to our Project Coordinators Mrs. Swati mam for their continuous help and monitoring during the project work.

In the last we would like to thank the management of Gurukul Institute of Engineering & Technology,Ranpur Kota for providing us such an opportunity to learn from these experiences.

All of our team is thankful to all the Faculties and Staff of Department of Computerin Engineering, GIET, for their help and support towards this project and ourteam.

We are also thankful to our whole class and most of all to our parents who have inspired us to face all the challenges and win all the hurdles life.

Thank you All

Student Name(with roll no)

Aakash paswan (RNo:19EGKCS001)

Deepak singh (RNo:19EGKCS011)

Hritik Bhagiya(RNo:19EGKCS015)

DECLARATION

We,hereby declare that the discussion entitled “SNAKE GAME” being submitted by us towards the partial fulfillment of the degree of bachelor of Technology,in the Department of Computer Engineering is a project work carried by us under the supervision of Mrs. pooja mam,and have not been submitted anywhere else.

Student Name

Aakash paswan (RNo:19EGKCS001)

Deepak singh (RNo:19EGKCS011)

Hritik Bhagiya(RNo:19EGKCS015)

ABSTRACT

The efforts required in achieving the desired output can be effectively and economically be decreased by the implementation of better designs. Power screws are used to convert rotary motion into translator motion. A screw jack is an example of a power screw in which a small force applied in a horizontal plane is used to raise or lower a large load. The principle on which it works is similar to that of an inclined plane. The Computer advantage of a screw jack is the ratio of the load applied to the effort applied. The screw jack is operated by turning a lead screw. The height of the jack is adjusted by turning a lead screw and this adjustment can be done either manually or by integrating an electric motor.

Screw jack plays an important role in changing the wheel of an automobile, also a huge amount of force is required in opening and tightening the nuts to fix the wheel. By the help of this project we want to reduce both of these manual jobs, to a less complicated – more efficient motor driven and Semi-Automatic Job.

The significance and purpose of this work is to provide a user friendly support for the existing car jack in order to make the operation easier, safer and more reliable in order to reduce health risks. The designed kit will also save time and requires less human energy to operate.

CONTENTS

Certificate ……….. |

Acknowledgement……….. ||

Declaration ……….. |||

Abstract ……….. |\/

Chapter 1 : INTRODUCTION ….. page no 7

1 introduction to the project

Chapter 2:LITERATURE REVIEV ….. page no 8

Literature survey

Chapter 3: DESIGN PROCEDURE ….. page no 9

3.1 System Design

3.2 Software Design

3.3 FUNCTIONAL REQUIREMENT

4. TESTING AND VALIDATION page no 10

1. INTRODUCTION

Snake game is one of the most popular arcade games of all time. In this game, the main objective of the player is to catch the maximum number of fruits without hitting the wall or itself. Creating a snake game can be taken as a challenge while learning Python or Pygame. It is one of the best beginner-friendly projects that every novice programmer should take as a challenge. Learning to build a video game is kinda interesting and fun learning.

We will be using Pygame to create this snake game. Pygame is an open-source library that is designed for making video games. It has inbuilt graphics and sounds libraries. It is also beginner-friendly, and cross-platform.

Installation

To install Pygame, you need to open up your terminal or command prompt and type the following command:

pip install pygame

2. LITERATURE SURVEY

LITERATURE SURVEYThe history of the Snake game goes back to the 1970's, the concept originated inthe 1976 arcade game Blockade, and its simplicity has led to many implementations. However, it was the 1990's when the game took on the look that we will be using. It was sold under numerous names and many platforms but probably gained widespread recognition when it was shipped as standard on Nokia mobile phones in the late 1990' The first published Nokia, for monochrome phones. It was programmed in 1997 by Taneli Armanto of Nokia and introduced on the Nokia 6110,

The game involves controlling a single block or snakehead by turning only left or right by ninety degrees until you manage to cut a block. When you get the block, the Snake grows an extra block or body segment.

If, or rather when, the snake bumps into the edge of the screen or accidentally casts himself the game is over. The more blocks the snake eats the higher the score.

PY Game Py game is a cross-platform set of python modules designed for writing video games.It includes computer graphics and sound libraries designed to be used with the PythonProgramming language.• To install the library, you can use pip installer from the command line:

pip install pygame

Import pygame

3.1 SYSTEM DESIGN

To create a Snake game that allows users to control the movement of a snake on a screen, to get points for eating food and avoiding runninginto the walls or the growing tail of the snake itself. In this problem, we want to write a game where a graphical representation of a snake moves across the screen. When it encounters a piece of food, the snake grows longer and we gain a point. If it hits the wall we die.

To write this program we are going to need:

1. A way of representing the sake of representing the food.

2. A way to display the score,

3. A way for our instructions to reach the snake

4. A way to know when we've run into something and

3.2 SOFTWARE DESIGN

We are going to use an object-oriented approach and provide some detail here. We have to think about the Classes that we want to build, with the associated variables and functions that will make sense for the development

Let's start by looking at the snake itself, the hero of the game. The stake has a location on the screen, and contains multiple visual elements, as it can grow, and the snake's head is connected to the rest of snake and the stake’s body follows it around the screen. If the snake “eats” food,it grows.

3.3 FUNCTIONAL REQUIREMENTS

Here are the requirements (functional requirements for how the snake moves.

1. The snake must appear to move around the screen.

2. The snake must turn in response to user input.

3. The snake will increase in length if it eats food.

4. The snake will die if it runs to the walls.

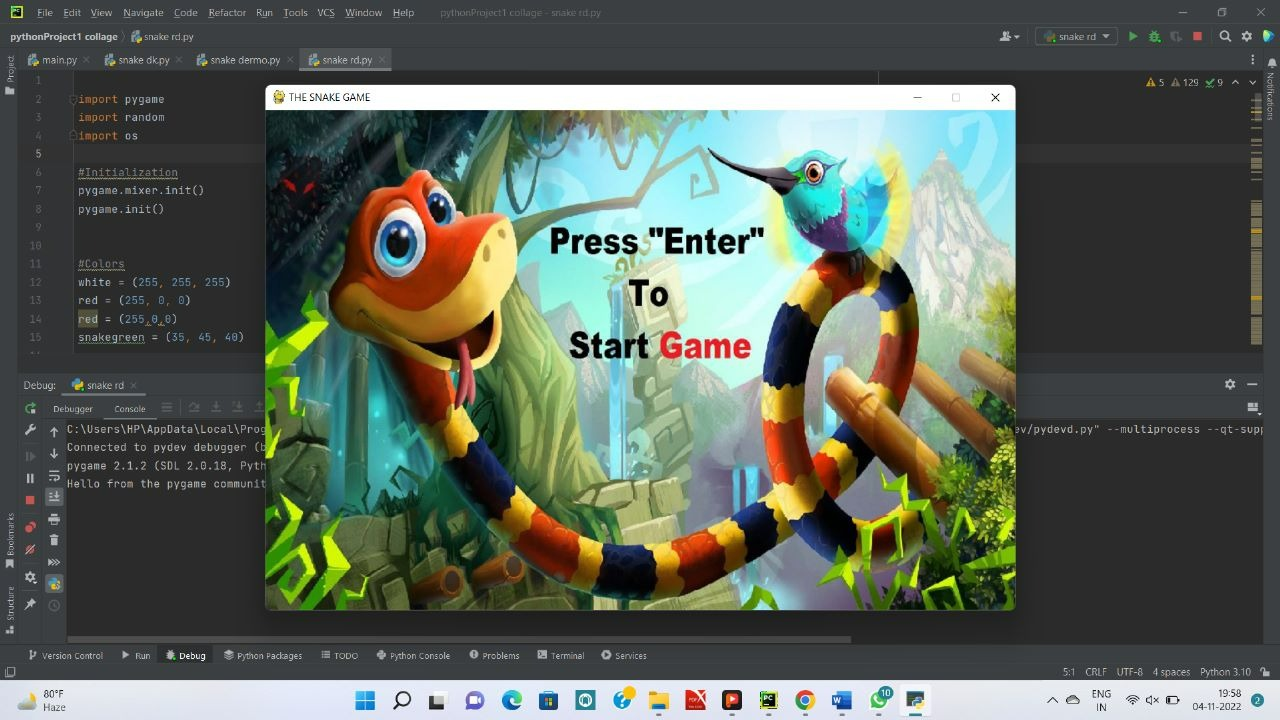
5. The snake never stops moving.

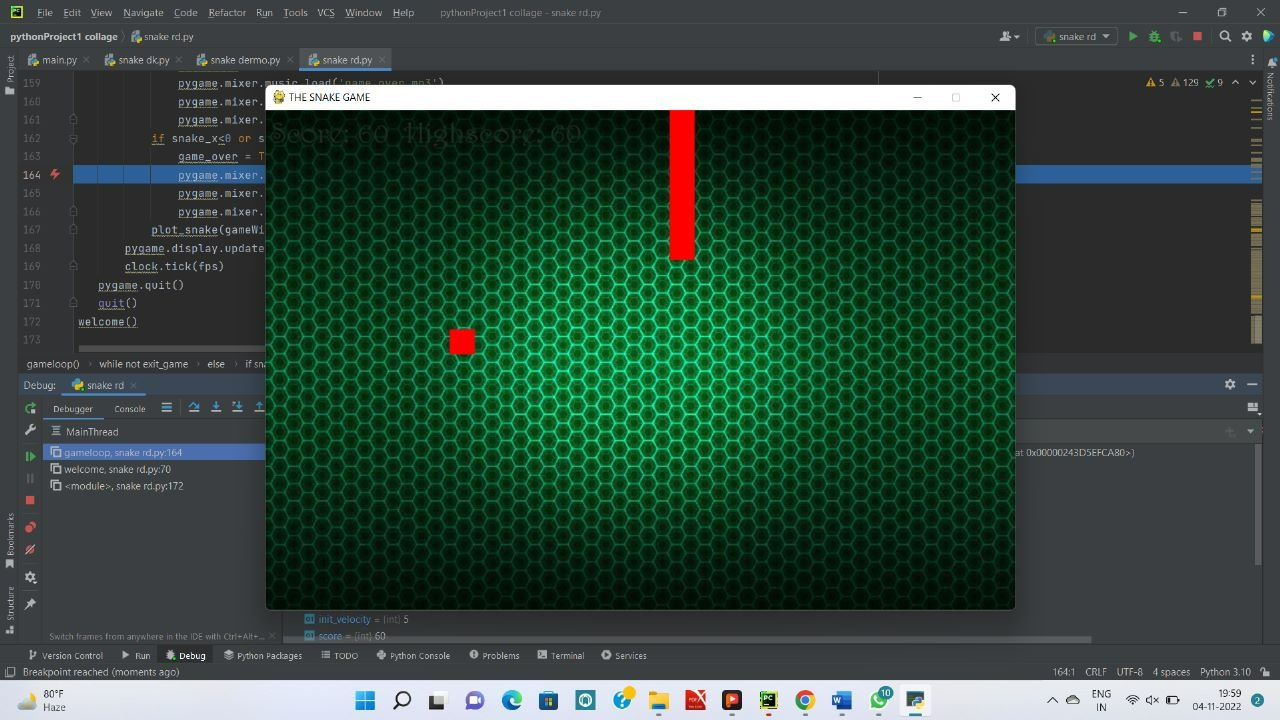
4. TESTING AND VALIDATION

To ensure that during operation the system will perform as per specification. TO make sure that system meets the user requirements during operation

To make sure that during the operation, incorrect input, processing and output will be detected. To see that when correct inputs are fed tothe system the outputs are correct

To verily that the controls incorporated in the same system as intended.Testing is aprocess of executing a program with the internet of finding error.





SOURCE CODE

import pygame

import random

import os

#Initialization

pygame.mixer.init()

pygame.init()

#Colors

white = (255, 255, 255)

red = (255, 0, 0)

red = (255,0,0)

snakegreen = (35, 45, 40)

#Creating The window

screen\_width = 900

screen\_height = 600

gameWindow = pygame.display.set\_mode((screen\_width, screen\_height))

#Game Backgrounds

# bg1 = pygame.image.load("bg.jpg")

bg2 = pygame.image.load("background.jpg")

bg2 = pygame.transform.scale(bg2, (screen\_width, screen\_height)).convert\_alpha()

intro = pygame.image.load("game start.jpg")

intro = pygame.transform.scale(intro, (screen\_width, screen\_height)).convert\_alpha()

outro = pygame.image.load("outro.png")

#Game Title

pygame.display.set\_caption("THE SNAKE GAME" )

pygame.display.update()

#Music

pygame.mixer.music.load('background.mp3')

pygame.mixer.music.play(100)

pygame.mixer.music.set\_volume(.6)

#Variables For The Game

clock = pygame.time.Clock()

font = pygame.font.SysFont('Harrington', 35)

def text\_screen(text, color, x, y):

screen\_text = font.render(text, True, color)

gameWindow.blit(screen\_text, [x,y])

def plot\_snake(gameWindow, color, snk\_list, snake\_size):

for x,y in snk\_list:

pygame.draw.rect(gameWindow, color, [x, y, snake\_size, snake\_size])

#Welcome Screen

def welcome():

exit\_game = False

while not exit\_game:

gameWindow.blit(intro, (0,0))

for event in pygame.event.get():

if event.type == pygame.QUIT:

exit\_game = True

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_RETURN:

pygame.mixer.music.fadeout(200)

pygame.mixer.music.load('background.mp3')

pygame.mixer.music.play(100)

pygame.mixer.music.set\_volume(.6)

gameloop()

pygame.display.update()

clock.tick(60)

# Game Loop

def gameloop():

# Game specific variables

exit\_game = False

game\_over = False

snake\_x = 45

snake\_y = 55

velocity\_x = 0

velocity\_y = 0

snk\_list = []

snk\_length = 1

#Highscore Build

if(not os.path.exists("highscore.txt")):

with open("highscore.txt", "w") as f:

f.write("0")

with open("highscore.txt", "r") as f:

highscore = f.read()

#Food

food\_x = random.randint(20, screen\_width / 2)

food\_y = random.randint(20, screen\_height / 2)

#Game Variables

score = 0

init\_velocity = 5

snake\_size = 30

fps = 60

while not exit\_game:

if game\_over:

with open("highscore.txt", "w") as f:

f.write(str(highscore))

#GameOverScreen

gameWindow.blit(outro, (0, 0))

text\_screen("Score: " + str(score ), snakegreen, 385, 350)

for event in pygame.event.get():

if event.type == pygame.QUIT:

exit\_game = True

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_RETURN:

welcome()

else:

for event in pygame.event.get():

if event.type == pygame.QUIT:

exit\_game = True

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_RIGHT:

velocity\_x = init\_velocity

velocity\_y = 0

if event.key == pygame.K\_LEFT:

velocity\_x = - init\_velocity

velocity\_y = 0

if event.key == pygame.K\_UP:

velocity\_y = - init\_velocity

velocity\_x = 0

if event.key == pygame.K\_DOWN:

velocity\_y = init\_velocity

velocity\_x = 0

if event.key == pygame.K\_q:

score +=10

snake\_x = snake\_x + velocity\_x

snake\_y = snake\_y + velocity\_y

if abs(snake\_x - food\_x)<12 and abs(snake\_y - food\_y)<12:

score +=10

food\_x = random.randint(20, screen\_width / 2)

food\_y = random.randint(20, screen\_height / 2)

snk\_length +=5

if score>int(highscore):

highscore = score

gameWindow.blit(bg2, (0, 0))

text\_screen("Score: " + str(score) + " Highscore: "+str(highscore), snakegreen, 5, 5)

pygame.draw.rect(gameWindow, red, [food\_x, food\_y, snake\_size, snake\_size])

head = []

head.append(snake\_x)

head.append(snake\_y)

snk\_list.append(head)

if len(snk\_list)>snk\_length:

del snk\_list[0]

if head in snk\_list[:-1]:

game\_over = True

pygame.mixer.music.load('game over.mp3')

pygame.mixer.music.play(100)

pygame.mixer.music.set\_volume(.6)

if snake\_x<0 or snake\_x>screen\_width or snake\_y<0 or snake\_y>screen\_height:

game\_over = True

pygame.mixer.music.load('game over.mp3')

pygame.mixer.music.play(100)

pygame.mixer.music.set\_volume(.6)

plot\_snake(gameWindow, red, snk\_list, snake\_size)

pygame.display.update()

clock.tick(fps)

pygame.quit()

quit()

welcome()