Summer training report/ synopsis/ minor project

on

CLASSIC SNAKE GAME

A Training Report submitted in partial fulfillment of
the requirements for the award of

Bachelor of Engineering

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

SAGAR

(Roll no: CO19355)

Under the supervision of

Dr. Ankit Gupta

CHANDIGARH COLLEGE OF ENGINEERING AND TECHNOLOGY

(DEGREE WING)

Government Institute under Chandigarh (UT) Administration, Affiliated to Panjab University , Chandigarh

Sector-26, Chandigarh. PIN-160019

<u>CANDIDATE'S DECLARATION</u>

I hereby declare that the work presented in this report entitled "**CLASSIC SNAKE GAME**" in fulfillment of the requirement for the award of the degree Bachelor of Engineering in Computer Science & Engineering, submitted in CSE Department, Chandigarh College of Engineering & Technology(Degree wing) affiliated to Punjab University, Chandigarh, is an authentic record of my/our own work carried out during my degree under the guidance of Dr. Ankit Gupta. The work reported in this has not been submitted by me for award of any other degree or diploma.

Date : 02/08/2020 Sagar

Place : Chandigarh CO 19355

CERTIFICATE

This is to certify that the Project work entitled "**CLASSIC SNAKE GAME**" submitted by Sagar in fulfillment for the requirements of the award of Bachelor of Engineering Degree in Computer Science & Engineering at Chandigarh College of Engineering and Technology (Degree Wing), Chandigarh is an authentic work carried out by him/her under my supervision and guidance. To the best of my knowledge, the matter embodied in the project has not been submitted to any other University / Institute for the award of any Degree.

Date : 2/08/2020

Place : Chandigarh

Dr.Ankit Gupta

Deptt of CSE

CCET(Degree Wing)

Chandigarh

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Dr. Ankit Gupta and Chandigarh College Of Engineering and Technology for supporting me throughout my project **CLASSIC SNAKE GAME**. First, I wish to express my sincere gratitude to my mentor Dr. Ankit Gupta, for his enthusiasm, patience, insightful comments, helpful information, practical advice and unceasing ideas that have helped me tremendously at all times in my project. His immense knowledge, profound experience and professional expertise has enabled me to complete this project successfully. Without his support and guidance, this project would not have been possible. I could not have imagined having a better mentor for my project.

I also wish to express my sincere thanks to the Chandigarh College of Engineering and Technology for accepting me into the under graduate program. In addition, I am deeply indebted to the Chandigarh College of Engineering and Technology for sponsoring my Coursera courses. This financial support has enabled me to complete project successfully.
Thanks for all your encouragement!

ABSTRACT

The concept of a snake game originated in the 1976 arcade game Blockade, and its simplicity has led to many implementation (some of which have the word snake or worm in the title). After a variant was preloaded on Nokia mobile phones in 1998, there was a resurgence of interest in the Snake concept as it found a larger audience.

Snake - The first published by Nokia, for monochrome phones. It was programmed in 1997 by Taneli Armanto of Nokia and introduced on the Nokia 6110.

Python is an interpreted, high-level, general-purpose programming language. Created by Guido Van Rossum and first released in 1991, Python Has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

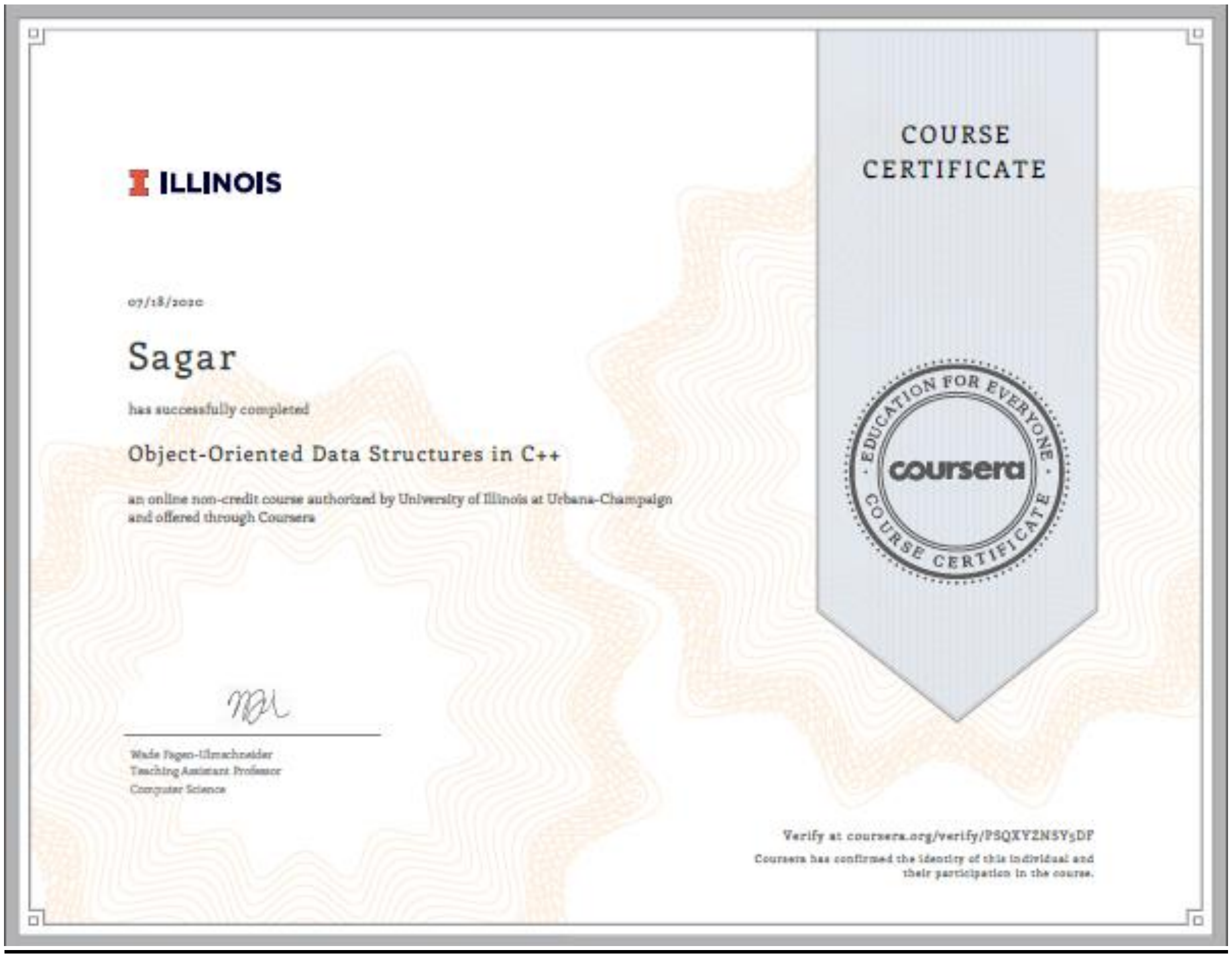
Pygame is a cross-platform set of Python Modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language.

List of tables

List of figures

Contents

1. COURSE CERTIFICATES



ILLINOIS

COURSE CERTIFICATE

07/18/2020

Sagar

has successfully completed

Object-Oriented Data Structures in C++

an online non-credit course authorized by University of Illinois at Urbana-Champaign and offered through Coursera

EDUCATION FOR EVERYONE coursera COURSE CERTIFICATE

[illegible]

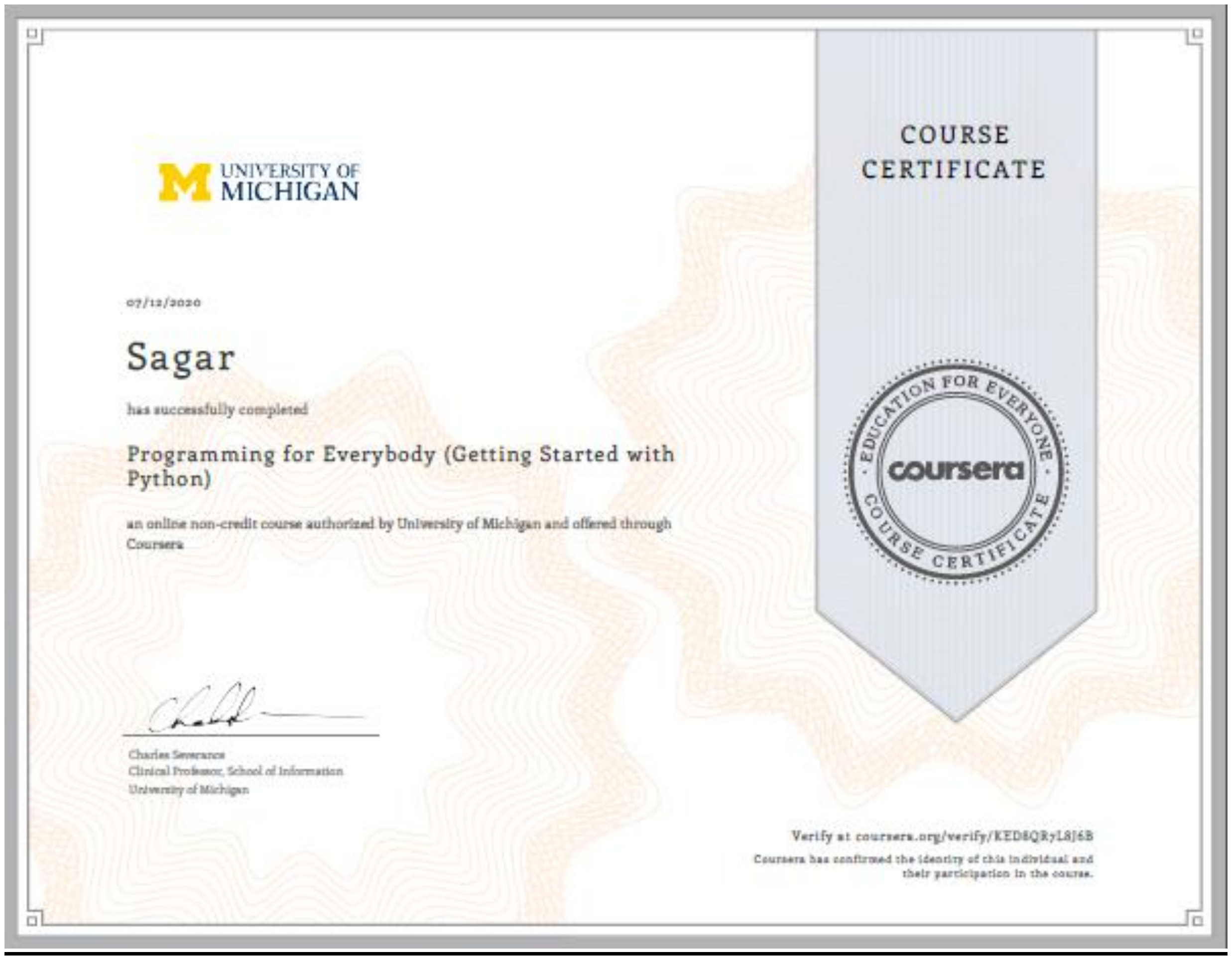
Wade Fagen-Ulmschneider
Teaching Assistant Professor
Computer Science

Verify at coursera.org/verify/PSQXYZNSY5DF

Coursera has confirmed the identity of this individual and their participation in the course.

Figure 1 Course Certificate

Verify at coursera.org/verify/PSQXYZNSY5DF



UNIVERSITY OF MICHIGAN

COURSE CERTIFICATE

07/12/2020

Sagar

has successfully completed

Programming for Everybody (Getting Started with Python)

an online non-credit course authorized by University of Michigan and offered through Coursera

EDUCATION FOR EVERYONE · coursera · COURSE CERTIFICATE

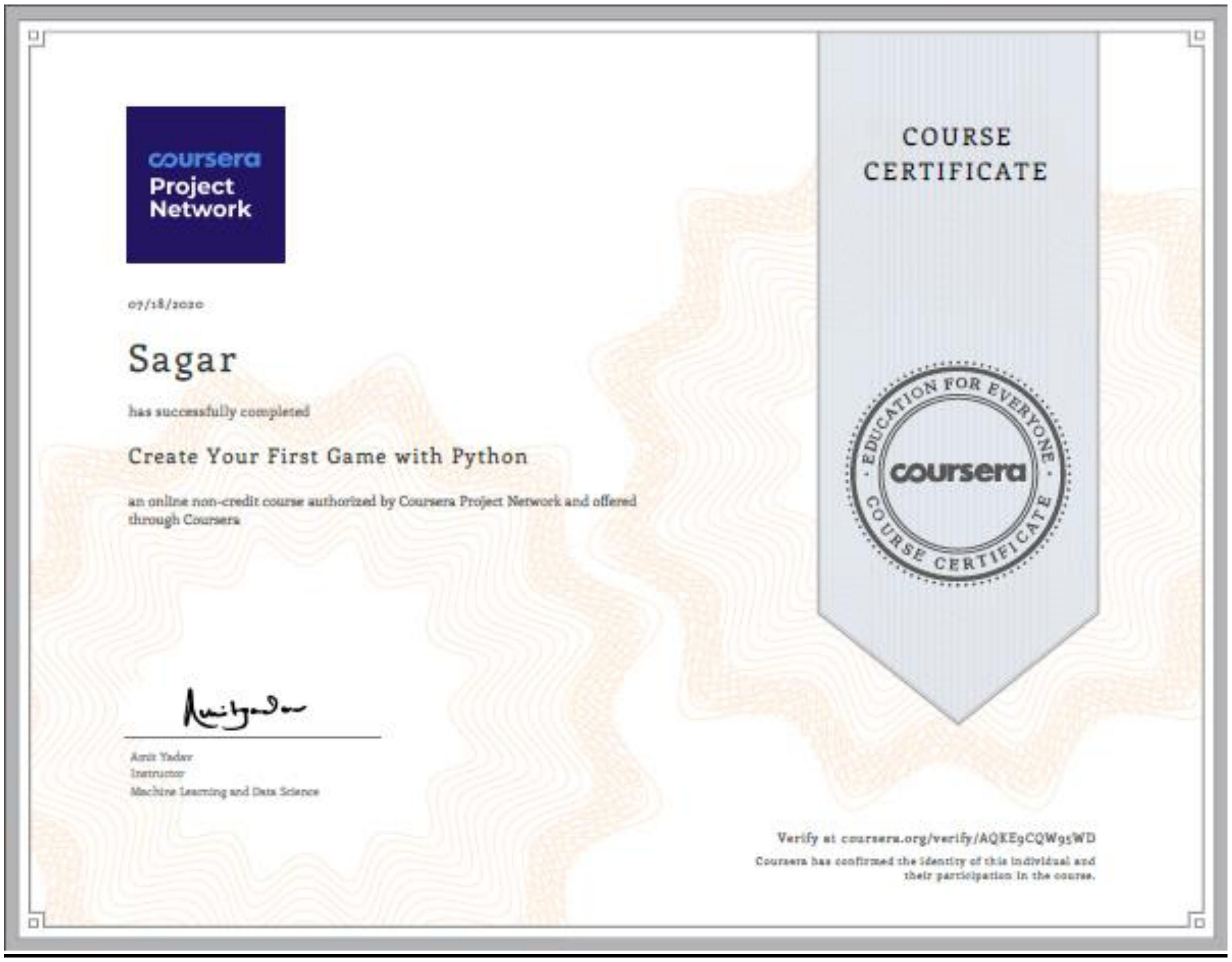
[illegible]

Charles Severance
Clinical Professor, School of Information
University of Michigan

Verify at coursera.org/verify/KED8QR7L8J6B
Coursera has confirmed the identity of this individual and their participation in the course.

Figure 2 Course Certificate

Verify at coursera.org/verify/KED8QR7L8J6B



coursera Project Network

COURSE CERTIFICATE

07/18/2020

Sagar

has successfully completed

Create Your First Game with Python

an online non-credit course authorized by Coursera Project Network and offered through Coursera

[illegible]

Amit Yadav
Instructor
Machine Learning and Data Science

EDUCATION FOR EVERYONE · coursera · COURSE CERTIFICATE

Verify at coursera.org/verify/AQKE9CQW95WD

Coursera has confirmed the identity of this individual and their participation in the course.

Figure 3 Course Certificate

Verify at coursera.org/verify/AQKE9CQW95WD

2. Project Details

2.1 Development Environment :

Language: Python 3.8

Operating System: Windows 7

Processor: Intel Pentium

2.2 Specifications :

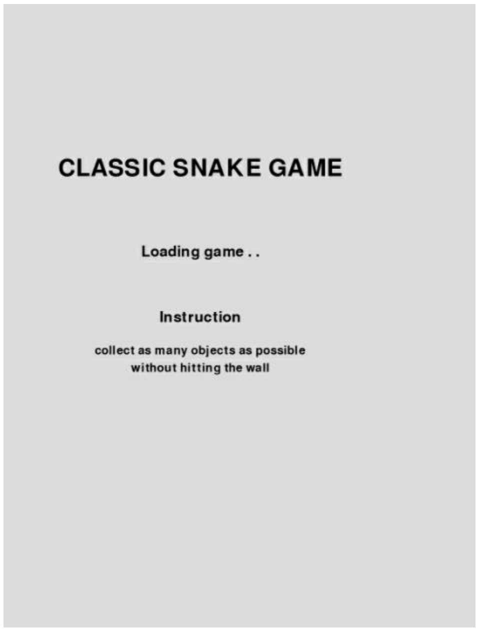
Operating System: Windows 7,8,10

Processor: Intel Pentium onwards.

RAM: 12 Mb

3. <u>How to use</u>

1. Run the application _snake_
2. The game starts with a window displaying out the name of our game, i.e., CLASSIC SNAKE GAME and displaying some instruction for the player.



CLASSIC SNAKE GAME

Loading game . .

Instruction

collect as many objects as possible
without hitting the wall

Figure 4 Game logo

3. After remaining on the screen for 4 seconds another window opens displaying "Get Ready!" and pauses for 2 seconds.

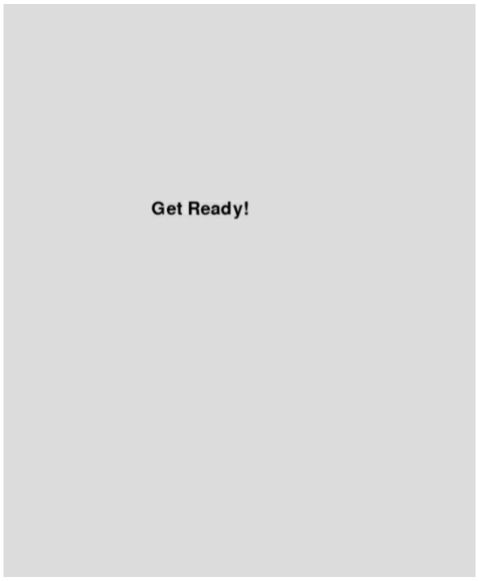


Figure 5 Get ready!

4. Now our game starts displaying a black window surrounded by a grey colour window which contains the score and level of the player.
5. On the black window there will be two things a green colour snake moving towards right side and a red colour food.

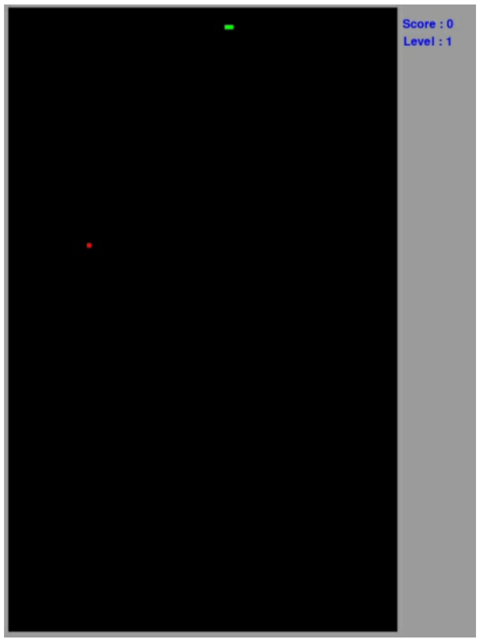
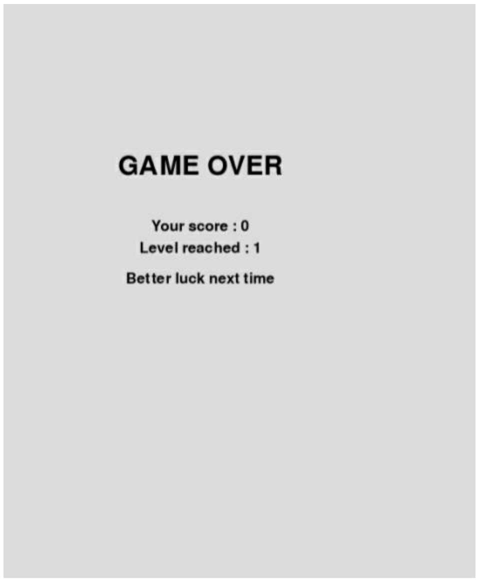


Figure 6 Game starts

6. The player has to control the direction of the snake. The primary task of the player is to protect the snake from hitting the wall and at the same time collecting the food.
7. The final score is the number of food items collected by the player. The score increases by one when a item is collected and the level increases by one when score increase by five. Also the speed increases as the level increases.
8. When the snake hits the wall, the game overs displaying "GAME OVER", final score, level reached and a comment for the player.



```
GAME OVER

Your score : 0
Level reached : 1
Better luck next time
```

Figure 7 Game over

4. **Flow chart**

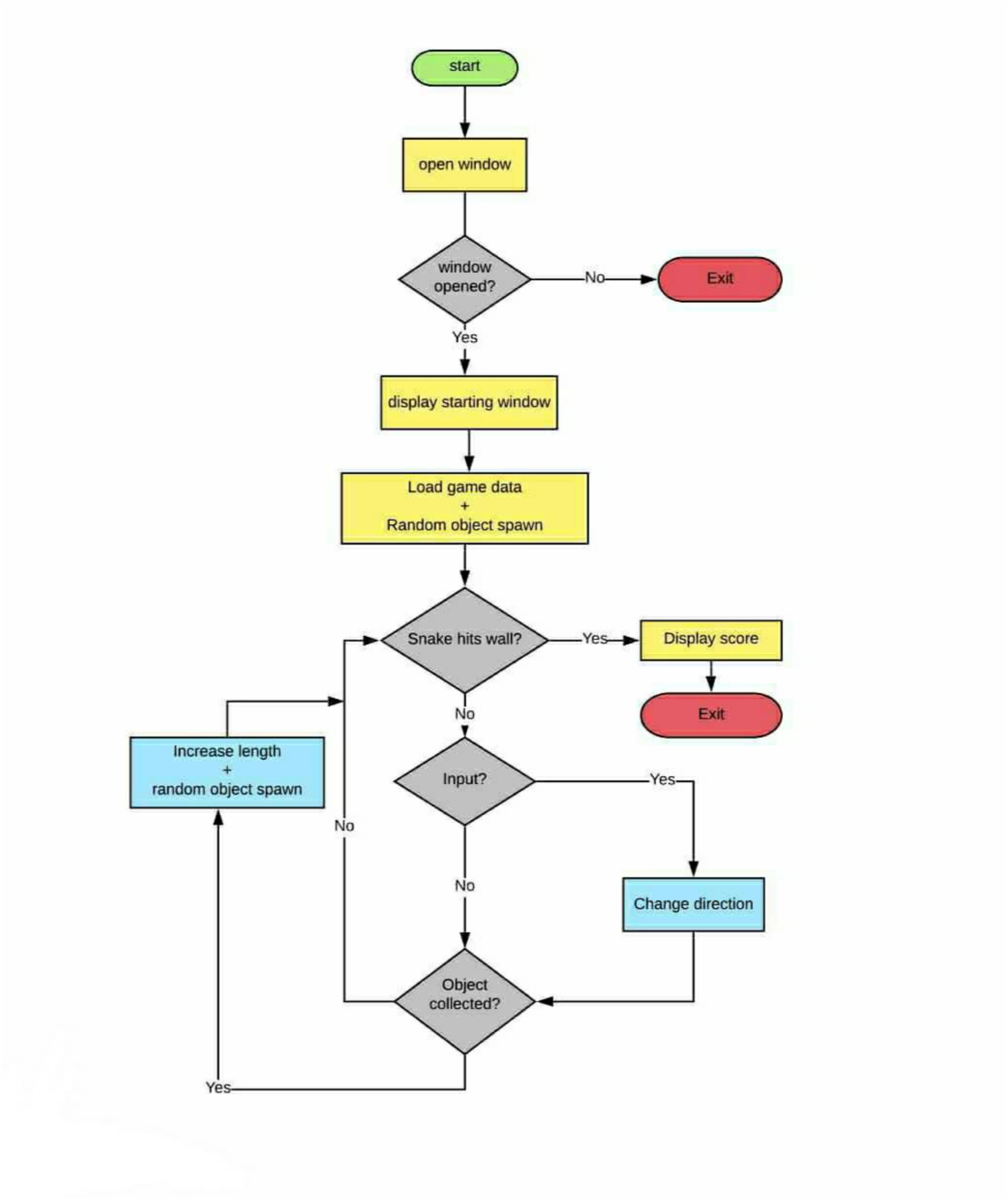


Figure 8 flowchart

5. Explaining code

5.1 Opening window:

```
xsize=1000
ysize=540
pygame.display.set_caption('CLASSIC SNAKE GAME')
window=pygame.display.set_mode((xsize+150,ysize))
```

This opens a window of size 1150x540 pixels

5.2 Starting instructions before starting game:

```
def start_window():
        window.fill((220,220,220))

        display_text('consolas',60,'CLASSIC SNAKE GAME',(0,0,0),xsize/2,ysize/4)
        display_text('consolas',30,'Loading game .',(0,0,0),xsize/2,ysize/4+100)
        display_text('consolas',30,'Instruction :',(0,0,0),xsize/2,ysize/4+170)
        display_text('consolas',25,'collect as many objects as
possible',(0,0,0),xsize/2,ysize/4+240)
        display_text('consolas',25,'without hitting the wall',(0,0,0),xsize/2,ysize/4+310)
        pygame.display.update()
        time.sleep(1)

        window.fill((220,220,220))
        display_text('consolas',60,'CLASSIC SNAKE GAME',(0,0,0),xsize/2,ysize/4)
        display_text('consolas',30,'Loading game . .',(0,0,0),xsize/2,ysize/4+100)
        display_text('consolas',30,'Instruction :',(0,0,0),xsize/2,ysize/4+170)
        display_text('consolas',25,'collect as many objects as
possible',(0,0,0),xsize/2,ysize/4+240)
        display_text('consolas',25,'without hitting the wall',(0,0,0),xsize/2,ysize/4+310)
        pygame.display.update()
        time.sleep(1)

        window.fill((220,220,220))
        display_text('consolas',60,'CLASSIC SNAKE GAME',(0,0,0),xsize/2,ysize/4)
        display_text('consolas',30,'Loading game . . .',(0,0,0),xsize/2,ysize/4+100)
        display_text('consolas',30,'Instruction :',(0,0,0),xsize/2,ysize/4+170)
        display_text('consolas',25,'collect as many objects as
possible',(0,0,0),xsize/2,ysize/4+240)
        display_text('consolas',25,'without hitting the wall',(0,0,0),xsize/2,ysize/4+310)
        pygame.display.update()
        time.sleep(1)

        window.fill((220,220,220))
        display_text('consolas',60,'CLASSIC SNAKE GAME',(0,0,0),xsize/2,ysize/4)
        display_text('consolas',30,'Loading game . . . .',(0,0,0),xsize/2,ysize/4+100)
        display_text('consolas',30,'Instruction :',(0,0,0),xsize/2,ysize/4+170)
        display_text('consolas',25,'collect as many objects as
possible',(0,0,0),xsize/2,ysize/4+240)
        display_text('consolas',25,'without hitting the wall',(0,0,0),xsize/2,ysize/4+310)
        pygame.display.update()
        time.sleep(1)
        ()
        window.fill((220,220,220))
        display_text('consolas',40,'Get Ready!',(0,0,0),xsize/2,ysize/4+50)
        pygame.display.update()
        time.sleep(2)
```

5.3 Direction control keys:

You can use

Figure 9 Arrow control

or

Figure 10 WASD controls

5.4 Direction control method:

```
elif event.type == pygame.KEYDOWN:
    if event.key == pygame.K_UP or event.key == ord('w'):
        change_direction = 'UP'
    if event.key == pygame.K_DOWN or event.key == ord('s'):
        change_direction = 'DOWN'
    if event.key == pygame.K_LEFT or event.key == ord('a'):
        change_direction = 'LEFT'
```

```
            if event.key == pygame.K_RIGHT or event.key == ord('d'):
                change_direction = 'RIGHT'
            if event.key == pygame.K_ESCAPE:
                pygame.event.post(pygame.event.Event(pygame.QUIT))

    if change_direction == 'UP' and direction != 'DOWN':
        direction = 'UP'
    if change_direction == 'DOWN' and direction != 'UP':
        direction = 'DOWN'
    if change_direction == 'LEFT' and direction != 'RIGHT':
        direction = 'LEFT'
    if change_direction == 'RIGHT' and direction != 'LEFT':
        direction = 'RIGHT'



    if direction == 'UP':
        snake_pos[1] -= 5
    if direction == 'DOWN':
        snake_pos[1] += 5
    if direction == 'LEFT':
        snake_pos[0] -= 5
    if direction=='RIGHT':
        snake_pos[0] += 5
```

The above code takes input from the user and accordingly changes the direction of the snake. It also checks that it doesnt Change its position immediately from right to left or up to down or vice-versa.

5.5 Random food spawn method:

```
    if not food_spawn:
        food_pos = [random.randrange(1, (xsize//10)) * 10, random.randrange(1, (ysize//10)) * 10]
    food_spawn = True
```

This code chooses a random location on the game window if the food_spawn have a False value, i,e., if there is no food on the screen.

5.6 Over function:

```
def over():
        window.fill((220,220,220))
        display_text('consolas',50,'GAME OVER',(0,0,0),xsize/2,ysize/4)
        display_text('consolas',30,'Your score : '+str(score),(0,0,0),xsize/2,ysize/4+80)
        display_text('consolas',30,'Level reached : '+str(level),(0,0,0),xsize/2,ysize/4+120)

        if score<2:
                display_text('consolas',30,'Better luck next time',(0,0,0),xsize/2,ysize/4+200)
        elif score<6:
                display_text('consolas',30,'Nice try',(0,0,0),xsize/2,ysize/4+200)
        elif score<10:
                display_text('consolas',30,'Well played',(0,0,0),xsize/2,ysize/4+200)
        else:
                display_text('consolas',30,'You are Excellent',(0,0,0),xsize/2,ysize/4+200)

        pygame.display.flip()
        time.sleep(100)
        pygame.quit()
        sys.exit()
```

5.7 Game over conditions:

```
if snake_pos[0] < 10 or snake_pos[0] > xsize-30:
    over()
 if snake_pos[1] < 10 or snake_pos[1] > ysize-10:
    over()
 for block in snake_body[1:]:
    if snake_pos[0] == block[0] and snake_pos[1] == block[1]:
        over()
```

Whenever the snake hits the wall or itself, the game ends by calling defined function over()

5.8 Functions defined:

S. No.	Function name
1.	display_text()
2.	start_window()
3.	over()

Table 1 Defined Functions