

**Program Name : Masters of Computer Application**

**Semester : 2**

**Paper Title : Software Engineering**

**Submitted by : Rishi Raj Chaurasia**

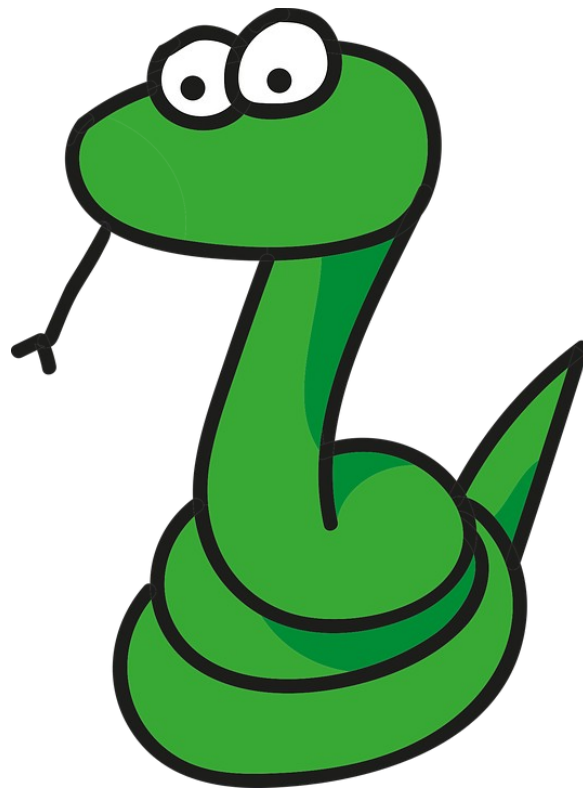
**Examination Roll Number : 19234757031**

**Department Roll Number : 31**

Software Requirements Specifications

# Snake-Swipe

A Classic Revisited



Version 1.0

Dept. of Computer Science  
University of Delhi  
20 March, 2020

Prepared by  
Aditya Ranjan(2)  
Rishi Chaurasia(31)  
Yetendra Bhandari(48)

# Table of Contents

<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions.....	1
1.4 Product Scope.....	1
1.5 References.....	1
<b>2. Overall Description.....</b>	<b>2</b>
2.1 Product Perspective.....	2
2.2 Product Functions.....	3
2.3 User Classes and Characteristics.....	3
2.4 Design and Implementation Constraints.....	3
2.5 User Documentation.....	3
<b>3. External Interface Requirements.....</b>	<b>4</b>
3.1 User Interfaces.....	4
3.1.1. Start Screen:.....	4
3.1.2. Gameplay Screen:.....	5
3.1.3. Video-Feed:.....	6
3.1.4. Processed Image Screen:.....	7
3.1.5. Game-Over Screen:.....	8
3.2 Hardware Interfaces.....	8
3.3 Software Interfaces.....	8
<b>4. Application Features.....</b>	<b>9</b>
4.1 Application Environment.....	9
4.2 Functional Requirement Specifications.....	9
4.2.1 Player use case.....	9
4.2.2 Description and Priority.....	11
4.2.3 Stimulus/Response Sequences.....	11
4.2.4 State Diagram.....	12
4.2.5 Functional Requirements.....	12
<b>5. Other Non-functional Requirements.....</b>	<b>13</b>
5.1 Performance Requirements.....	13
5.2 Safety Requirements.....	13
5.3 Security Requirements.....	13
5.4 Performance Requirements.....	13
<b>6. Other Requirements.....</b>	<b>14</b>

# Table of Figures

S. No.	Figure	Page No.
Figure 1	Context Diagram for 'Snake-Swipe' Application	2
Figure 2	Start Screen	4
Figure 3	Gameplay Screen	5
Figure 4	Video-Feed	6
Figure 5	Processed Image Screen	7
Figure 6	Game-Over Screen	8
Figure 7	Use Case Diagram For Snake-Swipe Application	9
Figure 8	Sequence Diagram for Snake-Swipe	11
Figure 9	State Transition Diagram for Snake-Swipe	12

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to present a detailed description of 'Snake-Swipe: A Classic Revisited'. It will explain the purpose and features, interfaces of this application, what the application will do, the constraints under which it must operate and how the application will react to external stimuli. This document is intended for both the stakeholders and the developers of the application.

## 1.2 Document Conventions

Font used : 'Bellota Text'.

Subheading : in bold

## 1.3 Intended Audience and Reading Suggestions

The document is intended for developers, project managers, end user, testers, and document writers.

## 1.4 Product Scope

This application will be used to demonstrate image-processing using the 'Python' a programming language and 'OpenCV' an image processing library.

## 1.5 References

IEEE Std. 830-1998 IEEE Recommended Practice for Software Requirements Specifications, IEEE Computer Society, 1998.

## 2. Overall Description

### 2.1 Product Perspective

This software is a modern twist on a classic video-game of 'Snake' where the player controls a virtual snake on-screen and attempts to complete the objective of eating the maximum number of food-items before the snake dies. This product allows the player to control the on-screen snake not only with normal keyboard input but also through swiping gestures with the help of the computer's webcam.

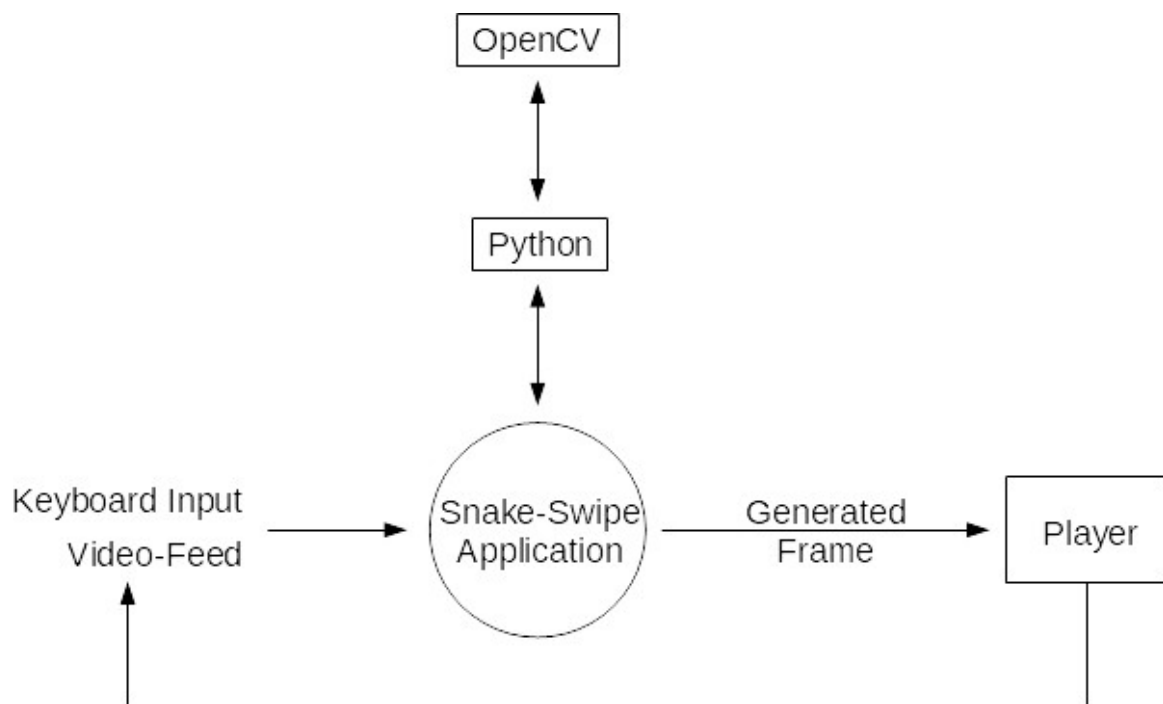


Figure 1: Context Diagram For 'Snake-Swipe' Application

## 2.2 Product Functions

- Players can use the keyboard or swiping gestures to control the snake.
- Once the game is over the player can see the score that they obtained.

## 2.3 User Classes and Characteristics

- Player : any user who intends to play the video-game.

## 2.4 Design and Implementation Constraints

- Must have Python installed.
- Must have OpenCV library installed.
- Webcam required in order to detect gestures.

## 2.5 User Documentation

- Shall deliver user manual for installation instructions.
- Shall install online help for users via the web interface.
- Shall deliver users guide book and operations & maintenance manual.
- Shall provide tutorials for users.

# 3. External Interface Requirements

## 3.1 User Interfaces

Graphical User Interface is used. Upon running the application, user is asked to press any key in order to start the game.

### 3.1.1. Start Screen:

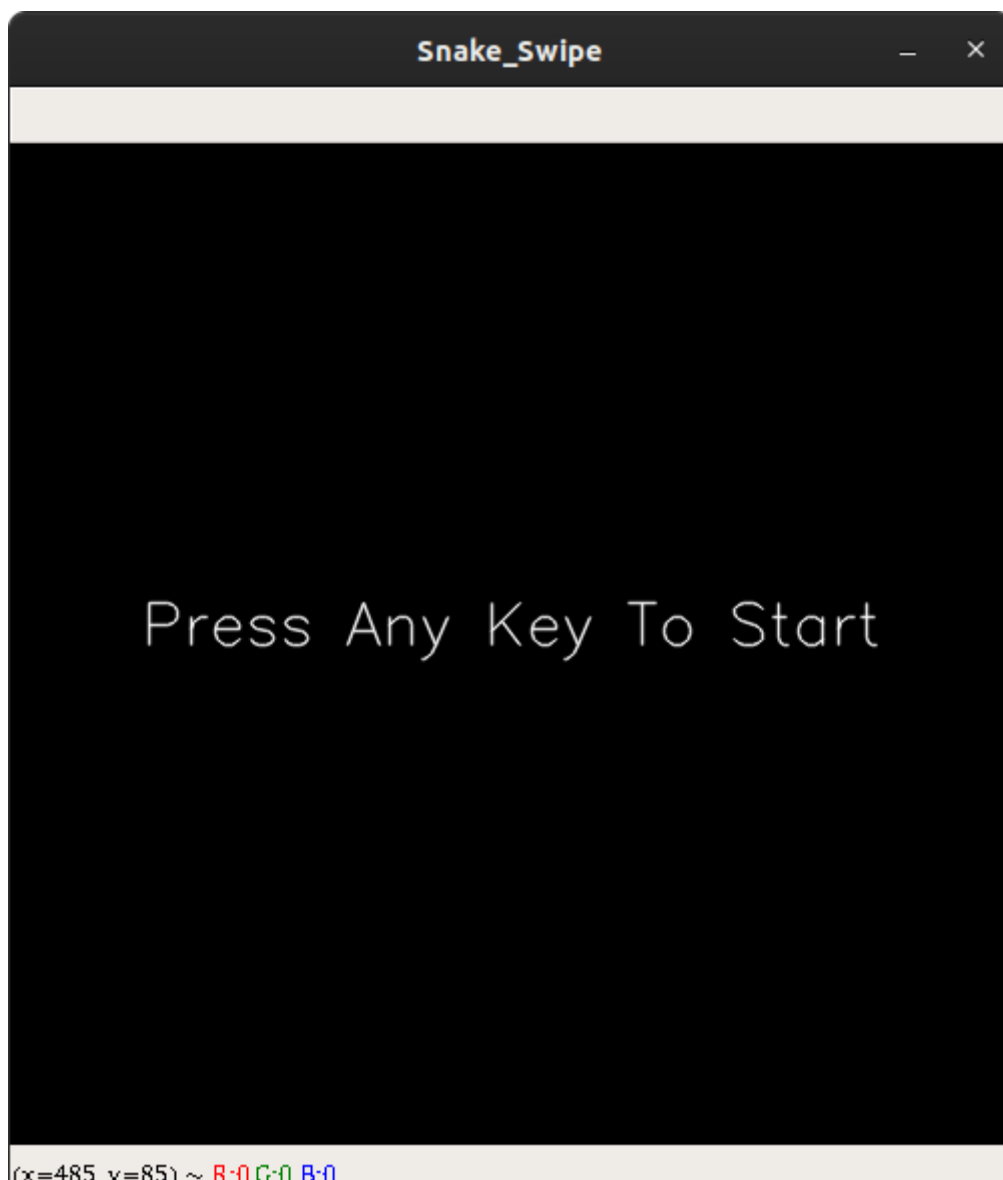


Figure 2: Start Screen

- This is the first screen shown to the player.
- It asks the user to press any to start the game.



### 3.1.2. Gameplay Screen:

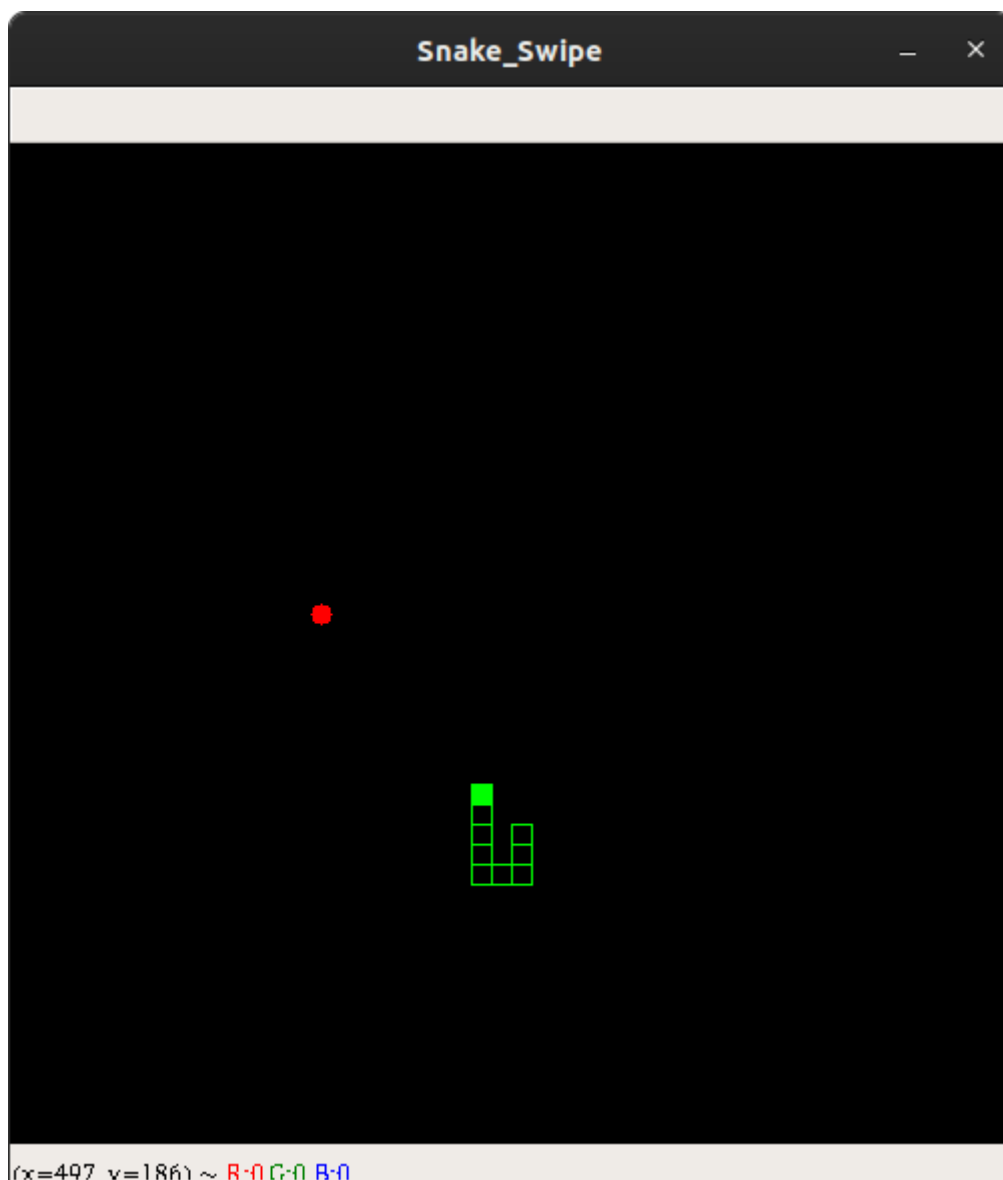


Figure 3: Gameplay Screen

- Once the player presses a key, the gameplay screen appears and the game begins.
- The player can control the on-screen snake using either the keyboard controls i.e. 'W', 'A', 'S', 'D' respectively for Up, Left, Down and Right directions or they can use swiping gestures i.e. move a green in front of the webcam in any of the four directions.
- Once the player presses the keyboard controls or swipes, the direction of the snake's movement is changed appropriately.

### 3.1.3. Video-Feed:

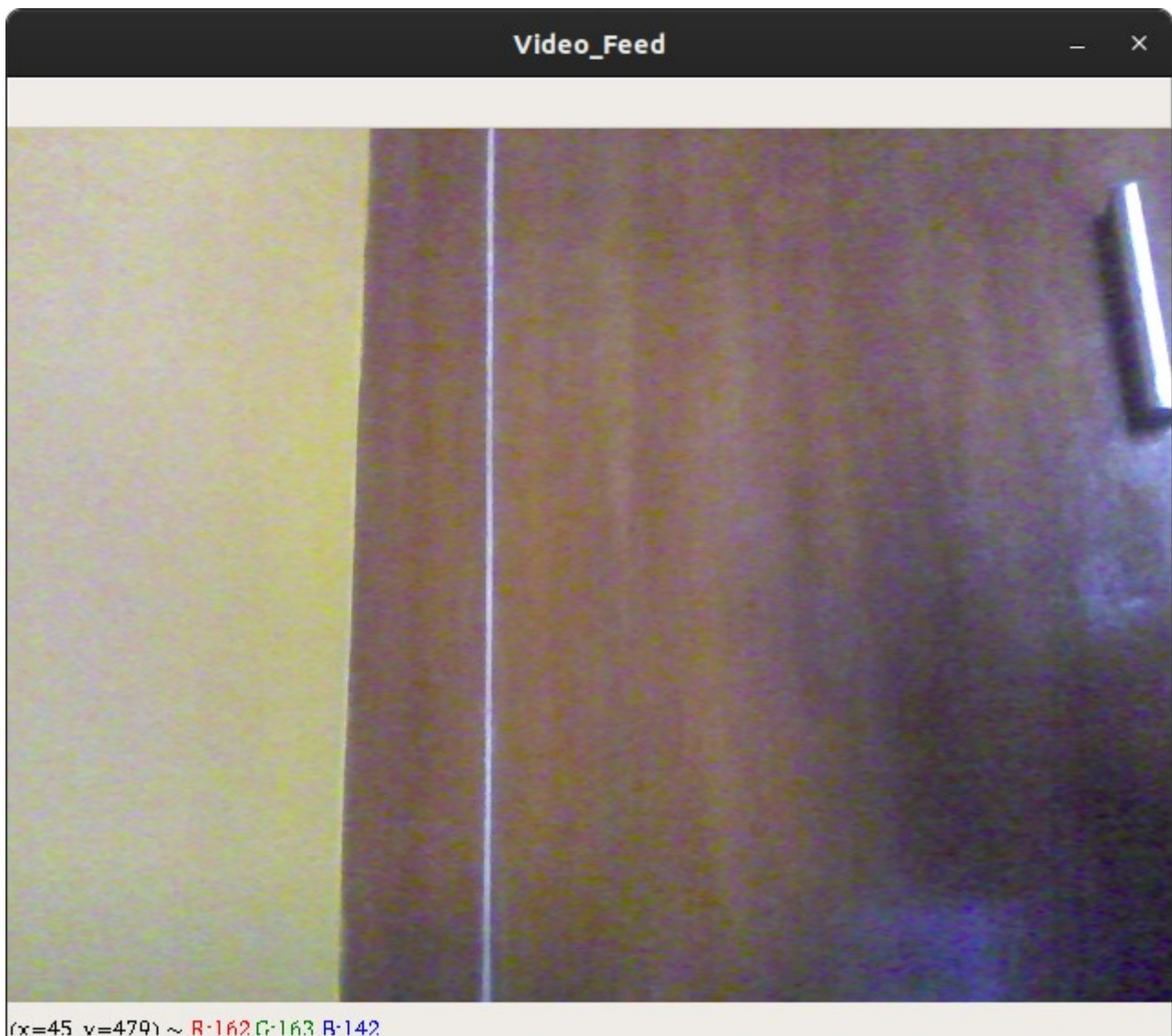


Figure 4: Video-Feed

- This window displays the original video-feed from the webcam.
- It also displays green edges around any green objects detected in the video-feed.

#### 3.1.4. Processed Image Screen:

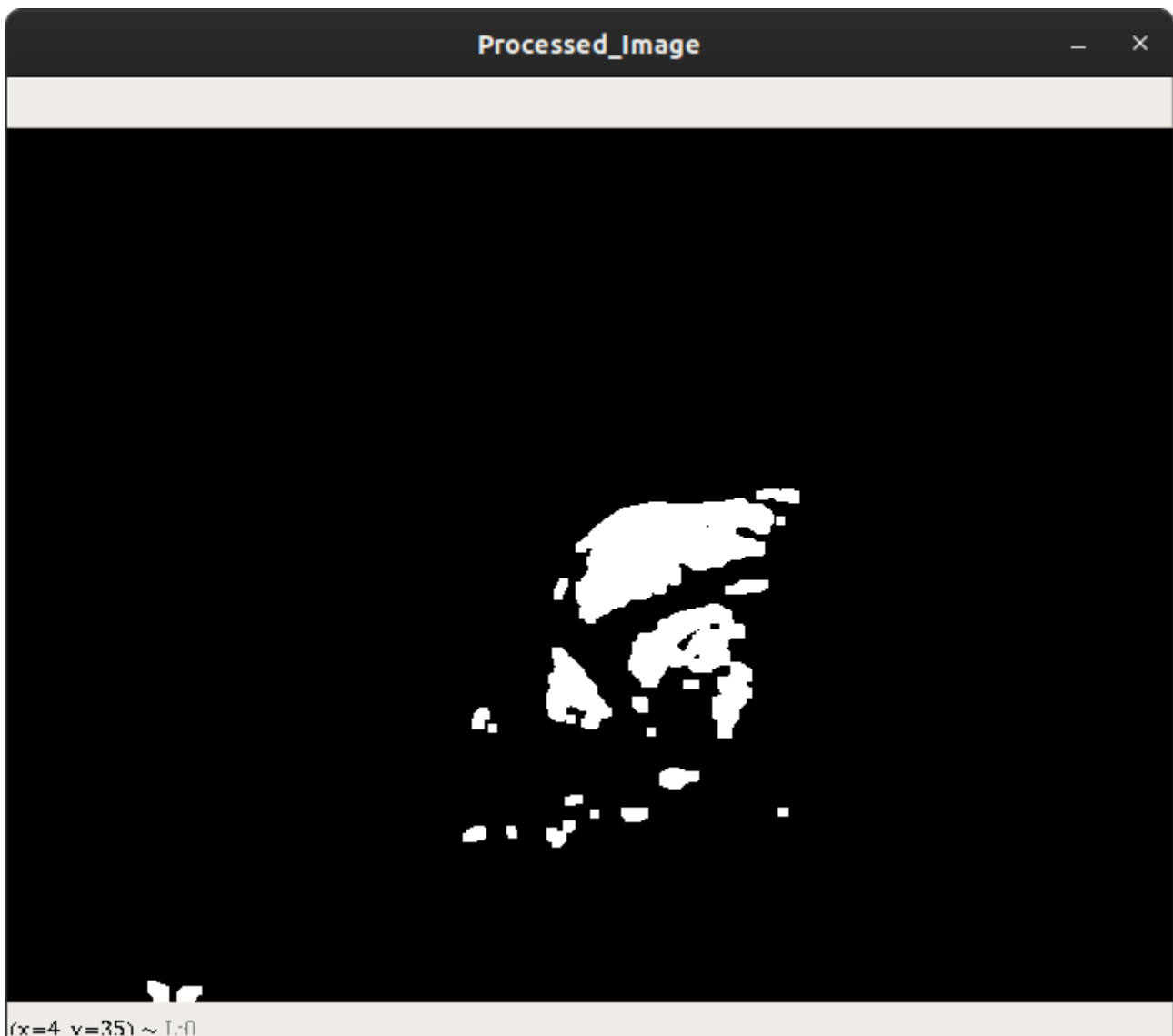


Figure 5: Processed Image Screen

- This window displays the processed video-feed used to detect the swiping gesture.
- It is only in black and white due to the image-processing algorithms used underneath the application.

### 3.1.5. Game-Over Screen:

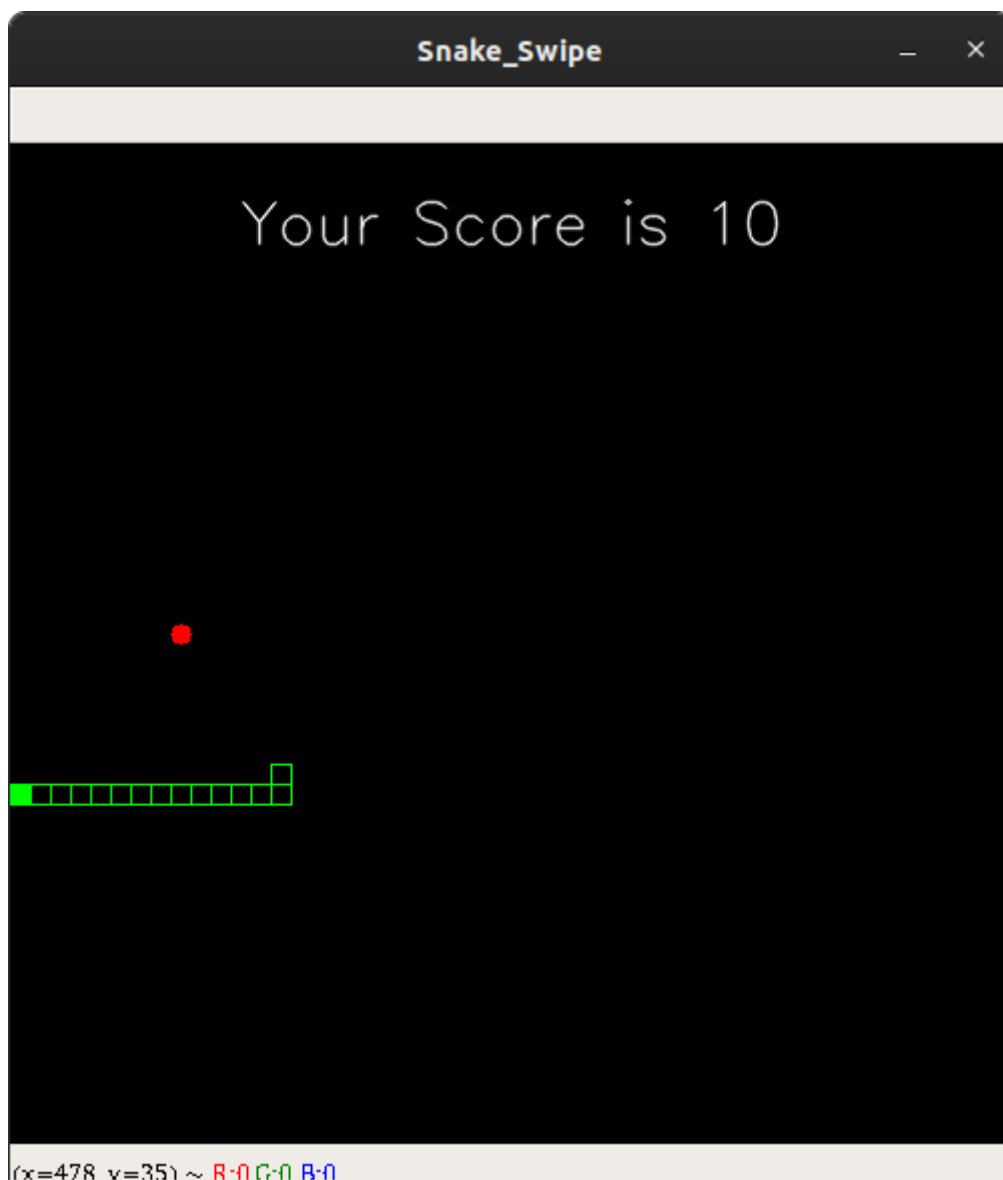


Figure 6: Game-Over Screen

- Once the player loses the game, the game-over screen is shown along with the score that the player obtained.
- At this screen, any key can be pressed in order to exit the application.

## 3.2 Hardware Interfaces

- Webcam, Keyboard

## 3.3 Software Interfaces

- Python Programming Language
- OpenCV Image Processing Library

# 4. Application Features

## 4.1 Application Environment

Being basically a video-game, 'Snake-Swipe' consists of only one actor i.e. the player and therefore the player can independently run this application.

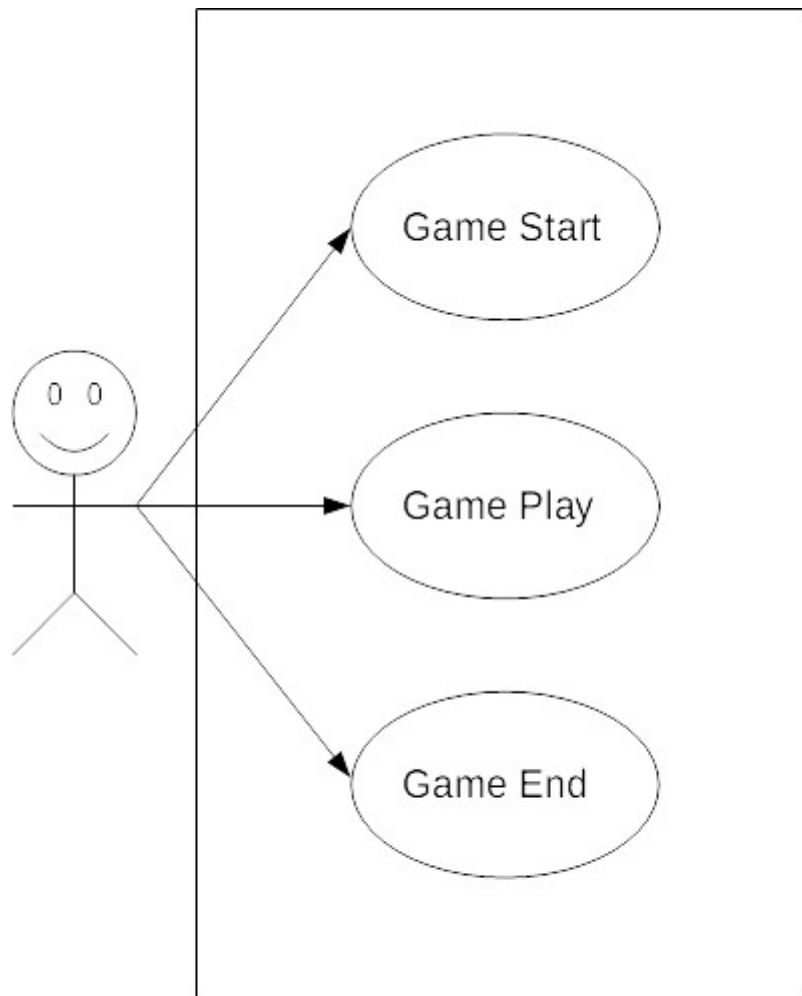


Figure 7: Use Case Diagram For 'Snake-Swipe' Application

## 4.2 Functional Requirement Specifications

### 4.2.1 Player use case

---

#### Game Start

---

Introduction	Player starts the application.
Actors	Player
Pre-Condition	Player must have the webcam on.
Post-Condition	The game begins.
Basic Flow	The player presses a key and the game begins.
Alternate Flow	The player does not press a key.

---

---

#### Game Play

---

Introduction	Player plays the game.
Actors	Player
Pre-Condition	Player must have started the game.
Post-Condition	The game ends.
Basic Flow	The player plays and tries to maximize their score.
Alternate Flow	The player loses or presses the exit key i.e. 'E'.

---

---

#### Game End

---

Introduction	Player sees their score on the game-over screen.
Actors	Player
Pre-Condition	Player must have lost or exited the game.
Post-Condition	The application closes.
Basic Flow	The player presses a key and the application closes.
Alternate Flow	The player does not press a key.

---

#### 4.2.2 Description and Priority

Players can use the keyboard or swiping gestures to control the snake. Once the game is over the player can see the score that they obtained. The score is not stored in the application.

#### 4.2.3 Stimulus/Response Sequences

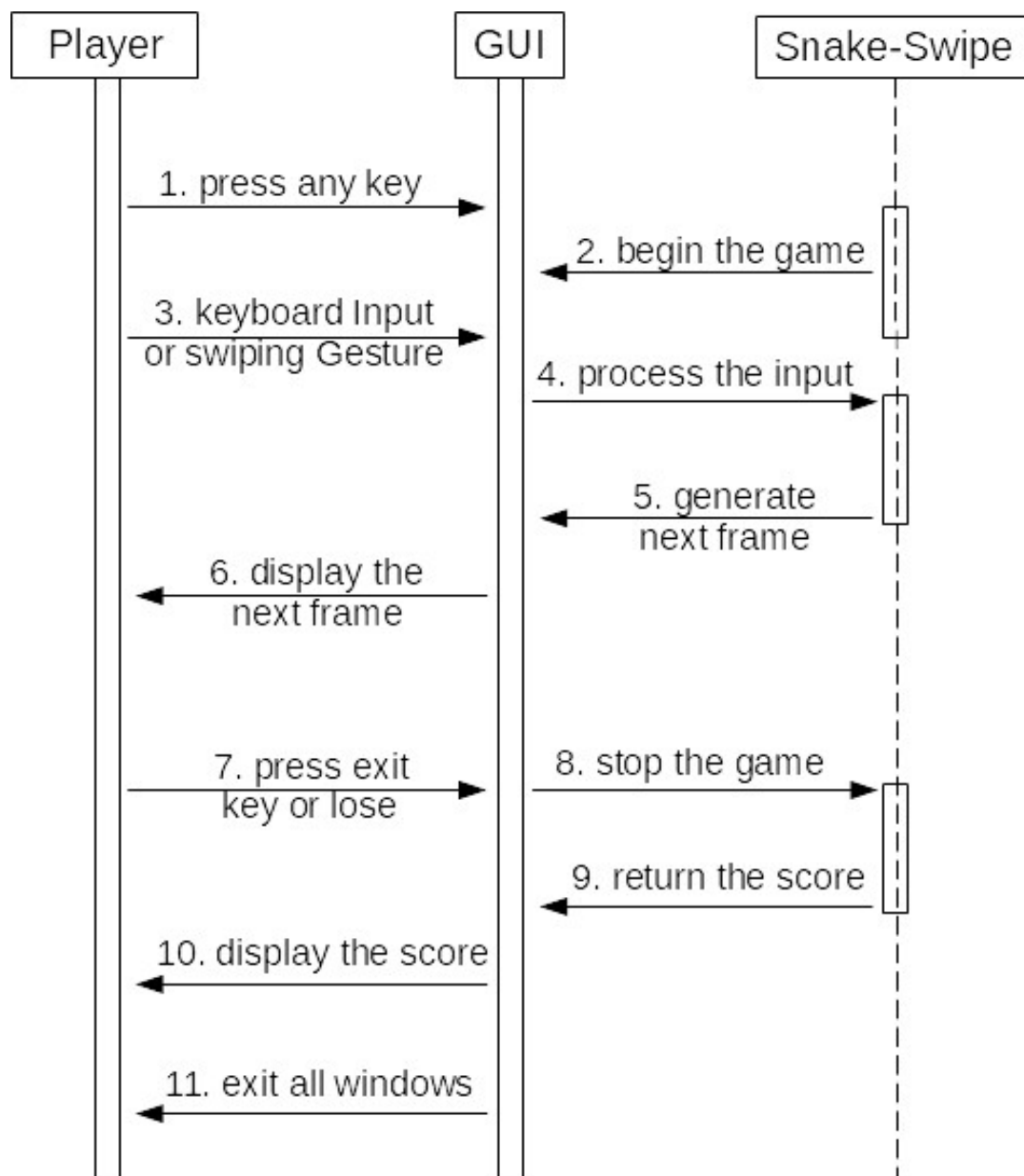


Figure 8: Sequence Diagram for Snake-Swipe

#### 4.2.4 State Diagram

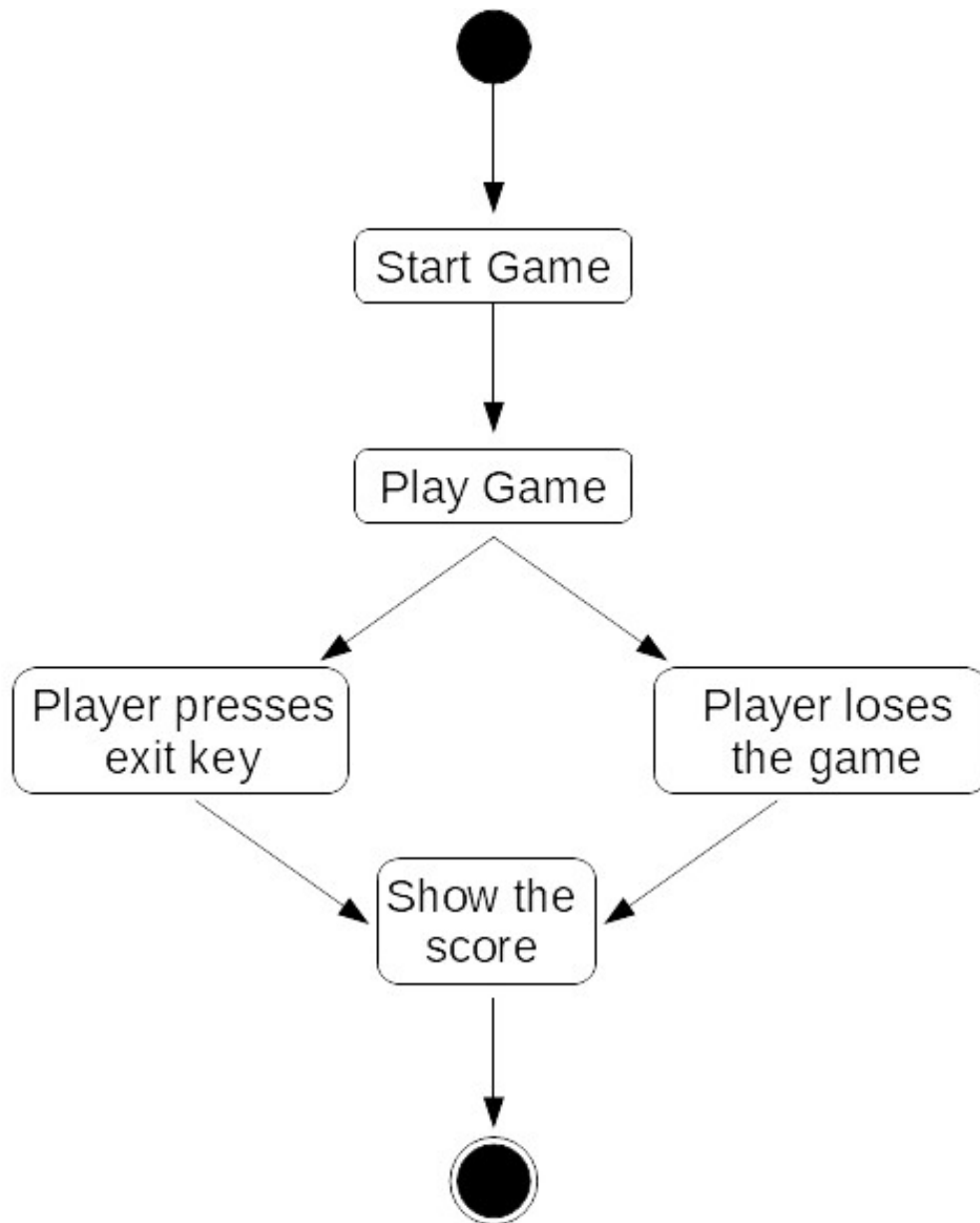


Figure 9: State Transition Diagram for Snake-Swipe

#### 4.2.5 Functional Requirements

Player will play the game and try to maximize their score obtained.



# 5. Other Non-functional Requirements

## 5.1 Performance Requirements

- The application should not cause any delay or lag between the user input and the frame obtained.
- The object detection algorithm should be able to detect green objects in a range of lighting conditions not just in ideal conditions.

## 5.2 Safety Requirements

Being a video-game, no sensitive data is being input in the application and therefore no backups are needed.

## 5.3 Security Requirements

Being a video-game, no sensitive data is being input in the application and therefore no security is required.

## 5.4 Performance Requirements

- Availability: the application shall have an availability of 99.99%
- Usability: application shall be easy to use and learn.
- Maintainability: application shall be easy to upgrade and easily updatable.
- Portability: application shall be extremely portable via USB drive.
- Testability: application should be able to run under debug mode and should be able to create test environment.

## 6. Other Requirements

The system shall adhere to following requirements:

- Python Programming Language.
- OpenCV Image Processing Library.
- Webcam required in order to detect gestures.