PROJECT REPORT FOR BACHELOR OF COMPUTER SCIENCE AND

INFORMATION TECHNOLOGY

**"LOOP HOLE"**



**Aabash Bashnet [86792]**

**Amshu Man Maharjan [86795]**

**Kamal Saud [86802]**

**Milan Gyawali [86806]**

**SAMRIDDHI COLLEGE**

**DEPARTMENT OF SCIENCE AND TECHNOLOGY**

**Nov, 2021**

PROJECT REPORT FOR BACHELOR OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

**"LOOP HOLE"**

**SUPERVISED BY**

**Mohan Bhandari**

FACULTY, SAMRIDDHI COLLEGE

A REPORT SUBMITTED

FOR

FIRST SEMESTER C-PROJECT

**Aabash Bashnet [86792]**

**Amshu Man Maharjan [86795]**

**Kamal Saud [86802]**

**Milan Gyawali [86806]**

**SAMRIDDHI COLLEGE**

**DEPARTMENT OF SCIENCE AND TECHNOLOGY**

**APRIL, 2021**

**DECLARATION**

We hereby declare that this project entitled "LOOP HOLE" is based on my original research work. Related works on this project by other researchers have been duly acknowledged. We owe all the liabilities relating to the accuracy and authenticity of the data and any other information included hereunder.

………………………… ………………………...

Amshu Man Maharjan [86795] Milan Gyawali [86806]

………………………… ….………………………...

Aabash Bashnet [86792] Kamal Saud [86802]

Date: 2021-01-25

**RECOMMENDATION**

This is to certify that this project entitled **"LOOP HOLE"** prepared and submitted by **Aabash Bashnet, Amshu Man Maharjan, Kamal Saud, Milan Gyawali** for First Semester C Project of Bachelor of Computer Science and Information Technology awarded by Tribhuvan University, has been completed under my supervision.

……………………..

Mr. Mohan Bhandari

Faculty, Samriddhi College

Date: February 2021

# **ACKNOWLEDGMENT**

First of all, the party that we would like to thank is Samriddhi College, for providing this excellent opportunity to participate in this project. Also, we would also like to thank the faculty for utilizing this minor project model to consolidate our knowledge on structured Programming and many concepts that are associated with it. We would also like to extended our thanks to our faculty members of Structured Programming Mr. Mohan Bhandari and Mr. Jeeban Dhungel for sharing their invaluable knowledge and providing guidance on this particular topic. Furthermore, we would also like to thank our principal Mr. Sandeep Shrestha, along with Arpan Maskey sir for providing us with the environment and facility so that we could complete the project on time without any time constraints and any pressures. Lastly, we would also like to thank our seniors and friends who have contributed their time to our project by helping us proof read, debug and guide us along the way. This endeavor wouldn’t be successful without the involvement of these parties and although not named we would also like to thank anyone remotely associated with this project and we would like them to know that their helps and advice have been invaluable in this endeavor.

# **ABSTRACT**

Loop hole is a simple game on C-Programming created by using graphics. In this game an object (we created a circle) is moved left right, up and down on the screen and if it hits an obstacles or enters in an obstacle (hole) then game will be over. But if we put the object in its destination then we will win the game. It is a very simple game and any one can play it in a computer.

In this game we moved a circle as an object in the screen. There are many obstacles on the way to the obstacles and reach to its destination. If the object can avoid the obstacles and reach to its destination the player will win otherwise the player will not win. We created a function winner if the object reaches in this destination then this function will be called. But if object doesn’t reaches in destination then we have created a function over. This function will be called if the object go into or hits the obstacle.

Hence by using graphs and creating some functions we have created this game.

Keywords: *Loosing holes, Winning hole, Walls, Winning part, Loosing part, Moving an object up and down.*

**Table of Contents**

[**ACKNOWLEDGMENT** i](#_Toc94564384)

[**ABSTRACT** ii](#_Toc94564385)

[**LIST OF ABBREVIATIONS** vi](#_Toc94564386)

[**1. INTRODUCTION** 1](#_Toc94564387)

[**1.1** **INTRODUCTION** 1](#_Toc94564388)

[**1.2** **PROBLEM STATEMENT** 1](#_Toc94564389)

[**1.3 OBJECTIVE** 2](#_Toc94564390)

[**1.4 SCOPE** 2](#_Toc94564391)

[**1.5 OVERVIEW OF REPORT** 2](#_Toc94564392)

[**2.** **BACKGROUND STUDY AND LITERATURE REVIEW** 3](#_Toc94564393)

[**2.1** **BACKGROUND STUDY** 3](#_Toc94564394)

[**2.2** **LITERATURE REVIEW** 3](#_Toc94564395)

[**3. SYSTEM ANALYSIS AND DESIGN** 5](#_Toc94564396)

[**3.1 SYSTEM ANALYSIS** 5](#_Toc94564397)

[**3.1.1** **REQUIREMENT ANALYSIS** 5](#_Toc94564398)

[**3.1.2** **FEASIBILITY ANALYSIS** 7](#_Toc94564399)

[**3.2** **SYSTEM DESIGN** 8](#_Toc94564400)

[**3.2.1 ALGORITHM** 8](#_Toc94564401)

[**3.2.2 FLOWCHART** 9](#_Toc94564402)

[**4. RESULT AND DISCUSSION** 10](#_Toc94564403)

[**4.1 RESULT** 10](#_Toc94564404)

[**4.2 TEST CASES** 12](#_Toc94564405)

[**5. LIMITATION** 17](#_Toc94564406)

[**6. FUTURE WORK** 18](#_Toc94564407)

[**7.RECOMMENDATION AND CONCLUSION** 19](#_Toc94564408)

[**7.1** **RECOMMENDATION** 19](#_Toc94564409)

[**7.2** **CONCLUSION** 19](#_Toc94564410)

[**REFERENCES** 20](#_Toc94564411)

**List of figure**

[Figure 1: use case diagram for the project 13](#_Toc94267315)

[Figure 2: flow chart of the project 17](#_Toc94267316)

[Figure 3: screen shot of home screen 18](#_Toc94267317)

[Figure 4: screenshot of the game screen. 18](#_Toc94267318)

[Figure 5: screenshot of the lose result screen. 19](#_Toc94267319)

[Figure 6: screen shot of won result screen 19](#_Toc94267320)

**List of Table**

Table 4.2.1:testcase1…………………………………….………………………………………12

Table 4.2.2: test case2……………………………...……………………………………………13

Table 4.2.3: test case 3……………………………………………..……………………………14

Table 4.2.4: test case 4…………………………………………………………………………..15

Table 4.2.5: test case 5………………………………..…………………………………………16

# **LIST OF ABBREVIATIONS**

2D : Two Dimensional

3D : Three Dimensional

API : Application Programming Interface

DT : Digital Trends

EXEC : Executable

FPS : Frames Per Second

GL : Graphics Library

GUI : Graphical User Interface

SPL : Structured Programming Language

# **1. INTRODUCTION**

## **INTRODUCTION**

LOOP HOLE has been enjoyed all around the world. It is a fun game that use concept of endless obstacles to make people try complete the level this makes the game fun even when it is repetitive.

This game has very clear and simple objective to avoid obstacle, thus, is very simple to take up and start playing without hassle. This game releases hormones like adrenaline and dopamine because of the constant frustration of failing at the game and euphoria of going just a few obstacles farther. It provides the same kind of physical and emotional reactions that an artist gets when they are creating art.[1] This game is very great break from monotony of work and also in those times when you just don’t feel like doing anything and just want to sleep, playing this game for some time will release dopamine and adrenaline which will wake you up pretty fast.

This version of the game is ported using the C language thus it can be played on many devices due to the flexibility of the C language.

## **PROBLEM STATEMENT**

The current reality of the world is that everyone is doing their very best to make life better for themselves and their loved ones. This means running in the rat races of the world, constantly trying to do better, to be better, to earn more and such. But in this hectic lifestyle and among the rat races, people are forgetting to take regular breaks of those that do take breaks are not able to think anything other than all the things they have yet to accomplish. This is giving people mild depressions and pessimism towards life and society and also impacting harshly on their mental health.

Thus, we are trying to make an easily accessible game that can take people’s mind off their problems and give themselves a break, a true break with excitement that this game provides. The release of the high excitement hormone adrenaline and satisfaction hormone dopamine is an added bonus that will make them feel good for a while.

## **1.3 OBJECTIVE**

The objectives of our project are the following:

i. To make loop hole more easily accessible using a highly versatile language.

ii. To provide people with a source of good clean fun that will alleviate their stress.

iii. To utilize our knowledge in Structured Programming to make something useful.

## **1.4 SCOPE**

Our project is applicable in any place that has access to a compute regardless of its specifications and its hardware capabilities.

This is made possible by the versatility of C language.

## **1.5 OVERVIEW OF REPORT**

This report altogether contains four chapters. The first chapter “Introduction” contains overall information of the game “LOOP HOLE”.

The chapter is further sub-headed into ten topics. Contents of this chapter are background, problem statements, objectives, scopes and overview of our project. Similarly, chapter two contains literature review of the report. In addition, chapter three is all about the methods by which the project is being completed. Under the third chapter “Methodology”, system flow diagram of loop hole along with its working mechanism is given.

# **2.** **BACKGROUND STUDY AND LITERATURE REVIEW**

## **BACKGROUND STUDY**

Our team decided to port loop hole in C because first off, it is a well like game and thus our program is familiar to the users as there is a high chance that user has at least seen the game. Having familiarity with the game is important because it will be easier to convince people to give a chance to our program instead of trying to make them give a chance to something that they have no idea what it is and no idea whether they will like it [1].

Secondly, we are also starting out with very beginner level knowledge of SPL. Thus, we wanted our project to be moderately challenging without hopeless levels of difficulty that would put us off SPL.

Thirdly, we wanted our project to incorporate technology that was traditional to us but learning it could prove to be very useful to us in the future. the graphics.h graphics library that is inbuilt with the C compilers is quiet old but still very useful [1].

Thus, because of these reasons, we decided that our project would be porting the popular mobile game loop hole to PC using C graphics.

## **LITERATURE REVIEW**

For any game to prove itself irresistible to play, it should have a perfect amalgam of hassle and reward. Psychologist Mihaly Csikszentmihalyi terms this feeling “flow” and swaying below this feeling leads to sense of ennui and above it leads to anxiety. Our game should provide exactly that, the sense of joy on completing after battling yourself to get the circle reach the destination or drop in the hole.

Codes that have been written by someone else is often difficult to understand and horrible to maintain and extend. Countless program units, like classes, or functions, are very large, some of them with thousands of lines of code. Too many dependencies between these units lead to unwanted side effects if something changed. While our team might not be a team of strong software architects we’ll try to implement what they do so that our program would not look like a “big ball of mud”. Understanding basics of software entropies is a very good start and yes, we’ve already implemented that in our project [1].

Mastering how to work with vectors in game development is one of the core building blocks to becoming a professional games programmer and gives you the ability to implement most of the mechanics found in modern games. Positioning of objects simply requires vectors and it will be used quite commonly in out project.

The Broken Window Theory was developed in U.S.-American crime research. The theory states that a single destroyed window at an abandoned building can be the trigger for the dilapidation of an entire neighborhood. The broken window sends a fatal signal to the environment: “Look, nobody cares about this building!” This attracts further decay, vandalism, and other antisocial behavior.

The Broken Window Theory has been used as the foundation for several reforms in criminal policy, especially for the development of Zero-Tolerance strategies. In game development, this theory can be taken up and applied to the quality of code Hacks and bad implementations, which are not compliant with the software design, are called “broken windows”. If these bad implementations are not repaired, more hacks to deal with them may appear in their neighborhood. And thus, the dilapidation of code is set 4 into motion. And to prevent that, we can nullify any broken windows that might appear during the development of our project.

One of the important aspect of good optimal gameplay is clear goal. And yes in our game, the rulw is very simple. Which means that the ones playing would not need to spend large chunks of their precious time to know what’s going around in the game [1].

# **3. SYSTEM ANALYSIS AND DESIGN**

# **3.1 SYSTEM ANALYSIS**

# **REQUIREMENT ANALYSIS**

1. **FUNCTIONAL REQUIREMENT**

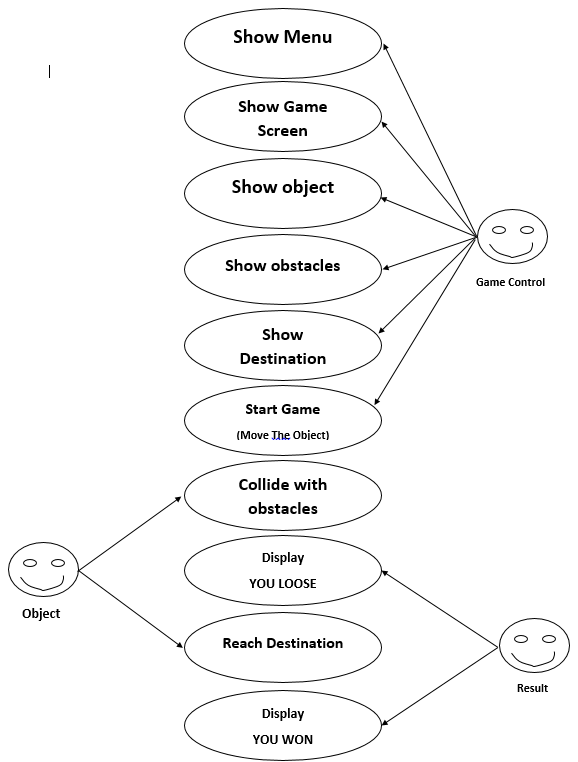


Figure use case diagram for the project

There are three factors in our project; object, result and and controller. Main objective of game is to get rid of all the obstacles by avoiding it moving left right up and down and die if it get into or hits the obstacles. There are two types of obstacles walls and holes.

Talking about our vital actor game controller, it does most of the heavy lifting like showing game screen and most importantly starting the game. All these tangle up to meet to make a fine recipe for an exciting game.

Some major entities of our program with their functions are:

a) Show Menu: It shows menu for user to choose level.

b) Show obstacles: It shows the obstacles that the player has to avoid.

c) Show destination: It shows the destination that the player has to reach.

d) Moving the object: It moves the object in the screen.

e) Collide with object: It detects the collision between objects and the obstacles.

f) Reach destination: It detects if the object has reached its destination.

g) Show Result: It shows the result if the player has won or loose.

**ii. NON-FUNCTIONAL REQUIREMENT**

## 

**a, ACCESSIBILITY**

One can easily gain access to our game on downloading it via internet. Users won’t have a problem downloading it from devices such as a pen drive. It is free of cost and provides a great entertainment to lure away user’s boredom.

**b. PERFORMANCE**

The game that we’ve built is restricted to 30 FPS because we believed that physics at higher or lower frame rates would feel slightly off. Other than that, it should run fine in your computer screen without lags or glitches.

**c. APPEARANCE**

LOOP HOLE might give you vibes about Super Mario because of the obstacles that look similar. Users who have grown playing Nintendo games can get the stimulus and environment of it and maybe even get back their happy good old moments.

## **FEASIBILITY ANALYSIS**

1. **TECHNICAL FEASIBILITY**

Our project is technically feasible because the code is written in C so it doesn’t require much investment

1. **OPERATIONAL FEASIBILITY**

Our project is operationally feasible because it doesn’t require any investment to run normally it only requires that you have the file for the game which is easily available.

1. **ECONOMIC FEASIBILITY**

Our game doesn’t cost anything to either play or run so it is economically feasible to anyone who is interested in our game.

1. **SCHEDULE FEASIBILITY**

Our game only required a week to program so it doesn’t require much time investment hence it is feasible in any schedule.

# **SYSTEM DESIGN**

## **3.2.1 ALGORITHM**

Step 1: Start

Step 2: Show the Welcome Screen

Step 3: Press the arrow keys to move the ball

Step 4: Does the ball collide?

Step 5: If yes, go to step 6,If No, go to step 7.

Step 6: Display “You Lose”.

Step 7: Does the ball reach the end point?

Step 8: If yes, go to step, If No, go to step 3.

Step 9: Display “You Win”.

Step 7: Stop.

## **3.2.2 FLOWCHART**

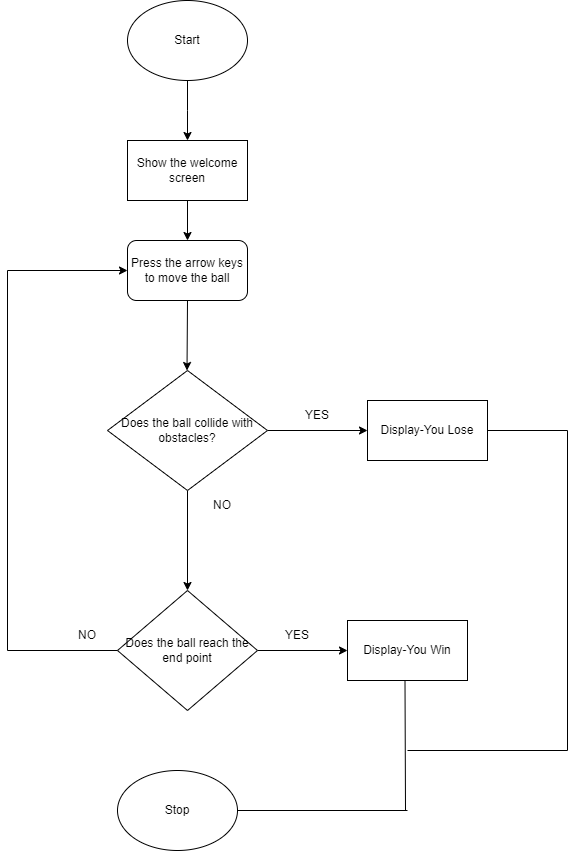


Figure flow chart of the project

# **4. RESULT AND DISCUSSION**

## **4.1 RESULT**

The final output of our program has following outputs:

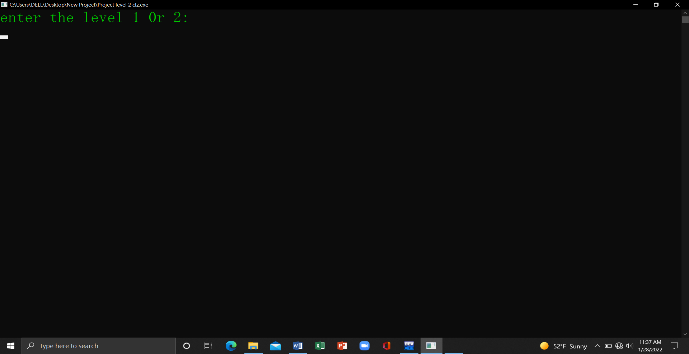


Figure screen shot of home screen

The figure above shown is the menu screen of the game. It is asking for the level of the game that the user wants to play.

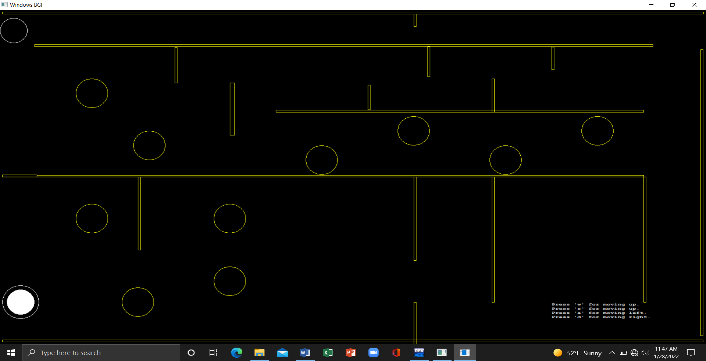


Figure screenshot of the game screen.

In this screen we can see an abject (White circle) on the top left corner and its destination in the top right corner. We can see there are many obstacles on the way to its destination in yellow color.

If we press 'W' then the object (White Circle) on the top left corner will move up. If we press 'S' then it will move down, if we press 'A' then it will move left and if we press 'D' then it will move right. If we go into the holes and collide with walls then result will be displayed on screen as

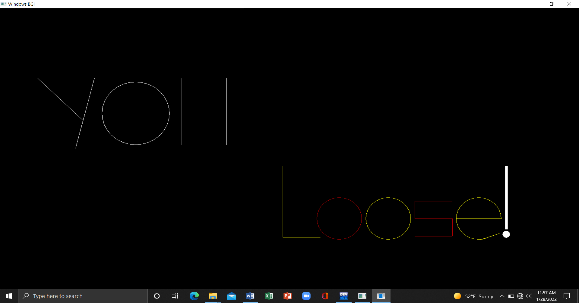


Figure screenshot of the lose result screen.



Figure screen shot of won result screen

## **4.2 TEST CASES**

Table 4.2.1: Test case 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | |  | | LOAD\_001 | | | | | | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
|  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| TEST CASE | |  | | PRESENTING WELCOME SCREEN BEFORE STARTING THE GAME AND | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DESCRIPTION | |  | | LEVEL SELECTION | | | | | | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
|  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| PRE REQUISITE | |  | | -Windows Terminal | | | | | | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
|  | |  | | G++ Compiler | | | | | | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
|  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| TEST SCENARIO | |  | | On Opening executable file | | | | | | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
|  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| TEST DATA | |  | | DESIRED SCREEN SIZE(1920\*1080) | | | | | | | | | | | | | |  | |  | |  | |  | |  | |  | |  | |
|  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| Steps | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
|  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | | | |
|  | |  | | Expected Result | | | | | |  | | Obtained Result | | | | | |  | | Result | | | | | | | | | |  | |
|  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
|  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| -Open executable file | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| -Provide Desired | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| Main screen and level selection | | | | | | | |  | |  | | Main screen and level selection | | | | | | | | Pass | | | | | |  | |
| screen size | |  | |  | |  | |  | |
|  | | should be shown | | | | | | | |  | |  | | was shown. | |  | |  | |  | |  | |  | |  | |  | |
|  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
|  | |  | |  | |  | |  | | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
|  | |  | |  | |  | |  | | | |  | | | |  | |  | |  | |  | |  | |  | |  | |  | |
| Table 4.2.2: Test case 2 | | | |  | |  | |  | | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| Test Case ID | | | |  | |  | | LOAD\_002 | | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
|  | | | | | |  | |  | | | |  | | | |  | |  | |  | |  | |  | |  | |  | |  | |
| TEST CASE DESCRIPTION | | | | | | Movement of ball on pressing A keys | | | | | | | | | | | |  | |  | |  | |  | |  | |  | |  | |
|  | | | |  | |  | |  | | | |  | | | |  | |  | |  | |  | |  | |  | |  | |  | |
| PRE REQUISITE | | | |  | |  | | Running instance of the game | | | | | | | | | |  | |  | |  | |  | |  | |  | |  | |
|  | | | |  | |  | |  | | | |  | | | |  | |  | | | |  | | | |  | |  | | | |
| Test Scenario | | | |  | |  | | Checking the direction of the ball while pressing the A key | | | | | | | | | | | | | | | | | | | |  | | | |
|  | | | |  | |  | |  | | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| Test Data | | | |  | |  | | “A” key | | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
|  | | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| Steps | | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
|  | |  | |  | |  | |  | | | |  | | | |  | |  | | | |  | | | | | |  | | | |
|  | |  | |  | |  | | Expected Result | | | | Obtained Result | | | | | | Result | | | | | | | |  | |  | |  | |
|  | | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
|  | | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| -Open the executable file | | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| -Select any level | | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
|  | |  | | Ball should move in left | | | |  | |  | | Ball moves in left | |  | |  | | Pass | | | |  | |  | |  | |
| -Press the A key | | | |  | |  | |  | |  | |  | |  | |  | |  | | | |
|  | |  | | direction. | | | |  | |  | | direction | |  | |  | |  | |  | |  | |  | |  | |
|  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
|  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
|  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 4.2.3: Test case 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Test Case ID |  |  | LOAD\_003 | | | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| TEST CASE |  |  | Movement of ball on pressing S key | | | | | | | |  |  |  |  |  |
| DESCRIPTION |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| PRE REQUISITE |  |  | Running instance of the game | | | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| Test Scenario |  |  | Checking the direction of the ball while pressing the S key | | | | | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Test Data |  |  | “S” key | | | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Steps |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | Expected Result | |  | Obtained Result | | | |  | Result | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| -Open the executable file |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| -Select any level |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Ball should move in downward | | |  |  | Ball moves in downward | | | |  | Pass |  |
| -Press the S key |  |  |  |  |  |  |  |
|  |  |  | direction. | | |  |  | direction | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | | |  | |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 4.2.4 : Test case 4 | |  |  |  |  | |  |  |  |  |  |  |  |  |  |
| Test Case ID | | |  |  | LOAD\_004 | |  |  |  |  |  |  |  |  |  |
|  | |  |  |  |  | | |  | |  |  |  |  |  |  |
| TEST CASE | | |  |  | Movement of ball on pressing D keys | | | | | |  |  |  |  |  |
| DESCRIPTION | | |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | |  |  |  |  | | |  | |  |  |  |  |  |  |
| PRE REQUISITE | | |  |  | Running instance of the game | | | | | |  |  |  |  |  |
|  | |  |  |  |  | | |  | |  |  |  |  |  |  |
| Test Scenario | | |  |  | Checking the direction of the ball while pressing the D key | | | | | |  |  |  |  |  |
|  | |  |  |  |  | |  |  |  |  |  |  |  |  |  |
| Test Data | | |  |  | “D” key | |  |  |  |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Steps | | |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | |  |  | |  |  |  |  |  |  |
|  |  |  |  |  | Expected Result | |  | Obtained Result | | |  | Result | |  |  |
|  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| -Open the executable file | | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| -Select any level | | |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Ball should move in right | |  |  |  | Ball moves in right |  |  |  | pass |  |
| -Press the D key | | |  |  |  |  |  |  |  |  |  |
|  |  | direction. | |  |  |  | direction. |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | | | | |  |  |  |  |  |  |
|  |  |  |  |  |  | | | | | |  |  |  |  |  |

Table 4.2.5 : Test case 5

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | | |  |  |  |  |  |  |  |  |  |  |  |  |
| Test Case ID |  | LOAD\_006 | | |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | | |  |  |  |  |  |  |  |  |  |  |  |  |
| TEST CASE |  | Displaying the result | | |  |  |  |  |  |  |  |  |  |  |  |  |
| DESCRIPTION |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | | |  |  |  |  |  |  |  |  |  |  |  |  |
| PRE |  | Running instance of the game | | |  |  |  |  |  |  |  |  |  |  |  |  |
| REQUISITE |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | | |  |  | | |  | |  |  |  |  |  |  |
| Test Scenario |  | Checking if ball collides with the obstacles and display the result | | | | | | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Test Data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Steps |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | | |  |  | | |  | |  | |  |  |  |  |
|  |  | Expected Result | | | Obtained Result | | | |  | | Result | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| -Open the |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| executable file |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| When the ball moves without | | |  |  | When the ball moves without | | | | | |  | Pass | |  |
| -Select any |  |  |  |  |  |
| level |  | colliding with the obstacles and | | |  |  | colliding with the obstacles and | | | | | |  |  |  |  |
| -Play the game |  | reaches end point, it should | | |  |  | reaches end point, it displays- | | | | | |  |  |  |  |
|  |  | display-“You Win” if not “You | | |  |  | “You Win” if not “You Lose”. | | | | | |  |  |  |  |
|  |  | Lose”. | | |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# **5. LIMITATION**

“Imperfection is sometimes way too perfect.”, The saying had to be brought up while casting some light on our not so flawless game, LOOP HOLE Although the game was built to be better than the original, with the newly added effect, improved controls to make users more addictive by maintaining science of flow, animation on death and so on, it still lacks few things.

1.Vsync not enabled: The game mode used to implement VSync could not find required extension in most of our test devices which had intel HD driver. With this, we could have watched the monitor’s max refresh rate. If the FPS of our game was equal to or higher than the refresh rate, VSync would have been enabled.

2. Scre tearing on Nvdia GPU: This project also failed to prevent screen tearing on Nvidia GPU, and that explains why it didn’t run on Mac book Air 2010 (one of our test device) which had Nvidia GPU. Not only that, also due to unavailability of free Glut, the game failed to run in the terminal.

3. Screen resolution: We also faced the problem of screen resolution as the game run perfectly on the pc where we code the program but does not show the complete screen to the another device having less screen resolution.

4. Absence of Wakelocks: The game fails to prevent the processor from sleeping or dimming because there is no addition of Wakelocks. Other than these flaws we don’t know of any other limitations of our game LOOP HOLE.

# **6. FUTURE WORK**

After working with loop hole, we’ve come a long way to get some understanding about c graphics and game development. We can now go ahead to work with some game engines to add more animations, and produce a 3D version of the game.

1. 3D version of the game: Previously, players of this game played the game on third person view and had a very little feeling of what it means to be the circle moved. The game in the future can promise you with the perspective of the circle which would further more enhance your experience with the game.

2. Networking involved: Furthermore, we can create a community of the game lovers after we work on the networking so that you could know about the global leaders of the game.

3. Addition of Wakelocks: We can also work with PowerManager Wakelocks to prevent your device’s processor from sleeping or dimming. The game runs on terminal and this might prevent the processor from dimming for now but we can’t be satisfied here.

4. Add more levels : we can add more levels as the gamer will get more and complex cases to complete the game which will enhance the performance of game and make it more compitative. 5. Vsync enabled: Implementation of VSync would let us to know if FPS of our game was higher than monitor’s max refresh rate or not. We are therefore also thinking about implementation of VSync.

# **7.RECOMMENDATION AND CONCLUSION**

## **RECOMMENDATION**

To those people who might take our report as reference we have some suggestions and recommendations:

First recommendation: Note down the coordinates of all the shapes. Because we need to know the coordinates of the shapes to add new shapes

Second recommendation: Make delay high otherwise the computer screen will blink continously.

## **CONCLUSION**

Those who are willing to start game development with C have a lots to look into from game engines to APIs, normal physics to vast camera movements and so much more. Just getting a grasp of basics of simple 2D games is a good way to go for it. We would advice you to get a firm knowledge on the concept of graphics before starting a project like this. This is a long journey and in it comes a lesson on each turns. In a nutshell, it was an exceptional project for us as we were working with graphics in C for the very first time.

There were a lot of frustrations as we hit the dead end a lot of times on various aspects of the development. However, all those grinding gave us some knowledge about API and object oriented programming. This was indeed a great experience for our team.

# **REFERENCES**

|  |  |
| --- | --- |
| [1] | E. BALAGURUSAMY, PROGRAMMING IN ANSI C, MC GREW HILL EDUCATION, 2018. |
| [2] | AYUSH BHATTARAI , DEWANAND GIRI, OM KC , UDAY SHRESTHA, "FLAPPY BIRD," SAMRIDDHI COLLEGE, BHAKTAPUR, 2021. |