TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING PULCHOWK CAMPUS

PROJECT REPORT ON

SINGLE PLAYER PONG GAME

A COURSE PROJECT SUBMITTED TO THE **DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING** IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE PRACTICAL COURSE ON COMPUTER PROGRAMMING

Submitted by:

Darshan Koirala 077BCT024
Aashish Shrestha 077BCT003
Avahan Tamrakar 077BCT015
Anmol Chalise 077BCT011
Bachelor of Computer Engineering, 1st SEM, 077, Pulchowk Campus, IOE, TU

Submitted to:

Department of Electronics and Computer Engineering, Pulchowk Campus
Institute of Engineering, Tribhuvan University Lalitpur, Nepal
Bhadra, 2078

ABSTRACT

A computer can be used in a variety of ways. People use computers in accordance with their needs. People's lifestyles have also changed dramatically as a result of technological advancements. Games have become popular on computers as well. Kids, As teenagers, everyone loves playing games on the computer.
A programming language can be used to create a variety of useful apps and software on a computer. C, C++, C#, Java, Python, QBASIC, and others are examples of programming languages. We are particularly interested in the C programming language out of all of these computer languages. Dennis M. Ritchie created the C programming language, which is widely used to create a variety of system and application software. This programming language has a wide range of applications including game development.

We have developed a simple game in c-programming language. The game is titled '**Single Player Pong Game**'. In this game, the player controls the bat by moving the mouse in an up-down direction. Unlike classical pong game, this game has a wall obstacle that can be hit by the player. The player scores when he protects it.

ACKNOWLEDGEMENTS

We would like to express our deep gratitude to **Mr. Ganesh Gautam**, our computer programming instructors who taught us C-programming language.

We would also like to thank all our dear friends and seniors for their advice, encouragement, and support.

TABLE OF CONTENTS

Chapter One:

INTRODUCTION

This C-programming mini-project requires only a rudimentary understanding of the high-level programming language. This initiative is primarily focused on gameplay and recording of high scores. This game also allows the user to view and delete scores. To To execute this project, we used a variety of C-programming ideas.

1.1. Background and problem statements: -

Practical performance, rather than theoretical understanding, is the most essential aspect in determining a technical student's ability and proficiency. As a result, students in the Computer Engineering Department are given an assignment to produce a project using the C programming language in order to improve their practical capacity to construct programs and software using the C programming language according to specifications.

This project is a part of the subject Computer Programming –I, first year and first part of the Computer Engineering course.

This project we have worked on is titled ‘Single Player Pong Game’.

The project necessitated familiarity with the C programming language's conditions, looping, structure, typedefs, enumeration, and characteristics.

We received assistance from lecture classes offered throughout the semester. We also consulted numerous books on c programming and raylib cheatsheets and different websites, which proved to be really helpful in learning computer programming fundamentals.

We began the project from scratch after gaining valuable concepts. Each member of our team contributed to the project by writing different aspects of the program

Chapter Two:

PROBLEM ANALYSIS

Understanding the problem

When we first came up with the concept for this application, we ran into a lot of roadblocks. Some of the difficult questions that came up throughout the discussion were:

- *How do we get started with the program?*
- *What operations are possible?*
- *What is the best way to preserve the data?*

Input Requirements

KeyBoard: SpaceBar → Start/Restart the game

Mouse: Up and Down Motion in Y direction → To move the Bat/Pad

Left Click → To start the movement of the ball.

2.3. Output Requirements

There are few output requirements in our game, the first being being able to display graphics on the screen so as to draw the different elements of our game on the screen so that users can play the game. The Inputs are to be taken with mouse and keyboard and manipulate the elements in accordance with application logic.

2.4. Processing Requirements

We used the basic science that we learn in physics like speed, coordinate system and logical analysis to operate this game. Here we make full use of pointers and structures to control the game states.

2.5. Technical Feasibility

There were some technical difficulties or boundaries while making the game. Out of complexity, the low level OpenGL graphics library could be very difficult to use, so we used the right tool for the job which is Raylib, a high level graphics engine that provides APIs (Application programming Interface) for displaying graphics and getting user inputs.

Chapter Three

REVIEW OF RELATED LITERATURES

What is C?

The C programming language is a procedural programming language. It was created primarily as a system programming language for developing operating systems. Low-level memory access, a small collection of keywords, and a clean style are all qualities that make C language excellent for system programming, such as operating system or compiler development.

Framework of a C program

Documentation Section

This section contains remark lines that provide the programmer's name, the author's name, and other information such as the time and date the program was written.
The documentation part assists anyone in getting a general understanding of the program.

Link Section

The header files for the functions used in the application are found in the link section. It tells the compiler how to connect functions from the system library together.

Definition Section

The defining section contains all of the symbolic constants.

Global Declaration Section

Global variables are declared in the global declaration section and can be used elsewhere in the program. The user-defined functions are also declared in this section.

main() Function Section

Every C program must have one main() function section. This section is divided into two sections: declaration and executable. Between the opening and closing braces, these two portions must be written. A semicolon must be used at the end of each statement in the declaration and executable sections (;).

Subprogram Section

All of the user-defined functions that are utilized to complete a certain task are found in the subprogram section. The main() method calls these user-defined functions.

1. Data Input and Output

A computer software often accepts input from a keyboard (standard input device) or other devices (e.g. data files). After that, the program processes the data and delivers the output to the appropriate devices (i.e. computer screen or monitor or data file).

Example:

```
character_variable = getch();
```

2. Control Statements

Control statements are statements that change the flow of execution of a program. Decision Making and Loop statements are the two most common types of control statements.

2.1 Decision Making Statements

The decision-making statements test a condition and allow some statements to be executed based on the test's outcome (i.e. either true or false).

if Statement

An if statement is used to control the flow of execution of statements.

Syntax:

```
if (test_expression)
{
statement-block;
}
```

Nested if Statement

When one if statement is written within body of another if statement, it is called Nested if Statement.

Loop construct

A loop can be defined as a group of statements that are performed repeatedly for a set number of times or until a given condition is met.

For Loop

The for loop allows to execute block of statements for a number of times.

Syntax:

```
for (counter_initialization; test_condition; increment or
decrement)
{
statements or body of loop;
}
```

while Loop

In this loop, the test condition is evaluated first and if the condition is true, the body of the loop is executed.

Syntax:

```
while(test_condition)
{
body of loop;
}
```

do-while Loop

do-while loop executes a body (i.e. block of statements) first without checking any condition and then checks a test_condition is checked to determine whether the body of the loop is to be executed for the next time or not.

Syntax:

```
do
{
   body of loop;
}  while(test_condition);
```

Break statement:

The break statement terminates the execution of the loop and the control is transferred to the statement immediately following the loop.

Syntax:

```
break;
```

switch Statement

When there are a number of options available and one of them is to be selected On the basis of some criteria, switch statement is used.

Syntax:

```
switch (variable or expression)
{
case caseConstant1:
statements;
break;
case caseConstant2:
statements;
break;
.......................
.......................
default:
statements;
}
```

Function

In a program, a function is defined as a self-contained set of statements that performs a certain job. User-defined functions and library functions are the two types of C functions.

Library Functions (Built-in Functions)

These are functions that have already been written, compiled, and placed in the C library and do not need to be written again by the programmer.
For example: printf(), scanf() etc.

User-defined Functions

These are the functions that the user specifies while writing a program.

Array and Strings

An array is a data structure that allows you to store a large number of data objects as a single entity (object). Individual data pieces are referred to as elements, and they all have the same data types.

Pointer

A pointer is a special type of a variable that contains a memory address instead of values (i.e. data).

Structure

A structure is a collection of variables under a single name. The variables may be of different types, and each has a name which is used to select it from the structure.

Syntax:

```
struct structure_name
{
data_type member_variable1;
}
```

Data Files

A data file is a location on the hard drive where a collection of linked data is saved For producing and processing data files, the programming language C includes a number of library functions.

Opening and Closing a file

Before performing reading and writing operations on a file, a software must first open it. When you open a file, it creates a connection between the software and the operating system.

The buffer area is established as

```
FILE *ptr_variable;
```

A data file is opened using syntax:

```
ptr_variable=fopen (file_name, file-mode);
```

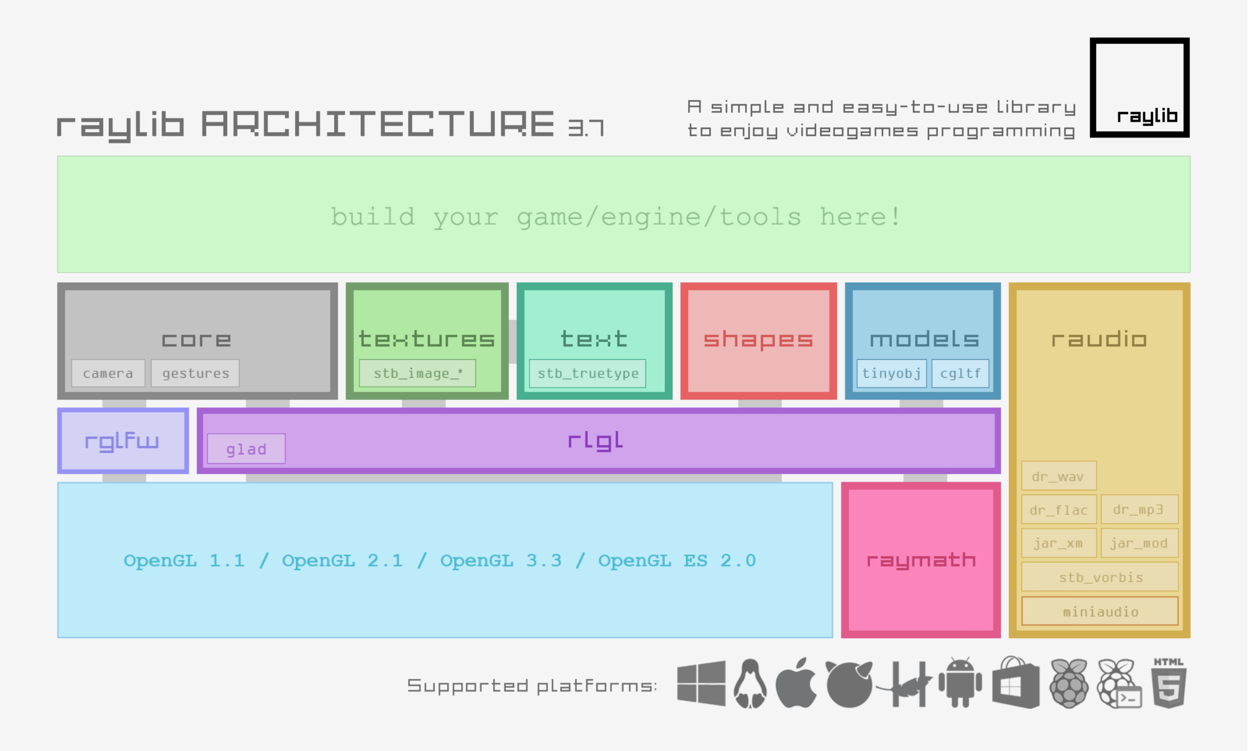
The file is closed using other library function fclose() as below:

```
fclose (ptr_variable);
```

Chapter 4

RAYLIB LIBRARY

Raylib (stylized in lowercase as raylib) is a cross-platform open-source software development library. The library is meant to create graphical applications and games. The official website introduces it as "a simple and easy-to-use library to enjoy video games programming."[3][4] The library is highly inspired by the Borland BGI graphics library and by the XNA framework and it's especially well suited for prototyping, tooling, graphical applications, embedded systems and education. The source code is written in plain C (C99) and it's primarily intended for beginners, distributed under open-source license. It supports compilation to several target platforms, including Windows, Linx, macOS, FreeBSD, Android, Raspberry Pi and HTML5.



Chapter 5

ALGORITHM

In this game we use a simple algorithm to build it. Following are our key algorithm:

```
Step 1: START
Step 2: Prepare a window with proper aspect ratio for the game
Step 3 : Create a never ending loop so that the screen will not terminate until the user presses Esc key on keyboard.
Step 4 : Creating an intro type scenario with the help of time functions of raylib library.
Step 5: Create a home screen where once the user presses the space bar, the game starts
Step 6: If the escape Key is not being pressed..
Step 7: Game starts once the user presses the left mouse button.
Step 8: The ball starts with a constant velocity, it collides with the walls and if
          Ball collides with pad
          Increment Score by 15
    Else
          Show Game Over Screen
Step 9: Listen to Input and accordingly go Step 6
Step 10: Goto step 5
Step 11: STOP
```

Chapter 6

LOGICAL APPROACH

Here in the game we have used a lot of logical approaches such as movement of the ball,its collision with wall and pad as well as movement of the pad. Likewise increment of the scores and termination of the game once the ball touches the left screen.

For Pad Movement: We just adjust the movement of the pad with the mouse Y-coordinates.
For Ball Movement: For ball movement we used some constants *speedX* and *speedY* for the speed in the concerned axis. The ball moves at a constant speed until it reaches the width or height of the screen and the ball bounces, for bouncing of the ball we multiply the speed of the respective axis by -1.
For Score: Once the ball touches the pad the score gets increased by 15 points.

Chapter 7

IMPLEMENTATION AND CODING

At the beginning of the project we thought working with the team would be difficult as we all have different methods to do the same things so it was hard for us to coordinate at the beginning but later on as time passed we got to know each other's skills and faults and improved our coordinations. We completed this challenge only with our team work because building a game is one of the hardest things in programming so working in a team is best for building a game.In our project we just divide our work in an effective way. After dividing our work our first priority was to look for the prototype so that we could replicate it. So we researched deeply throughout the web as well as raylib site to complete this game. Though our project targets have not been achieved completely but our first functional game prototype is prepared and presented here.

Chapter 8

RESULTS,DISCUSSIONS AND CONCLUSION

This slightly modified version of the famous *pong game* was created as a project work, successfully written in the C-programming language. An external library named Raylib was implemented into the program mainly for the purpose of displaying graphics, then working on it.

The basic logic of the program was to input the y-coordinates from the user's mouse and adjust the pads accordingly in each frame. While doing this, the program also simultaneously calculates the movement of the ball. Everytime the ball touches the pads, the score increases by 15

Various functions defined in the Raylib library, along with user defined functions and built-in functions were properly utilized while writing the program. Pointers, structures and arrays were also heavily involved while writing the program - which while running, has to continuously input data from keyboard and mouse.

Nowadays, the C-programming language is considered old and even obsolete by many, we can see that it is not true. It is close to machine level language than most other languages which allows for faster execution time. It is suitable for beginners to learn about the basics of programming.

We would like to express our utmost gratitude to the reader for reading this project report.

----------------------------------x--------------------------------x-------------------------------------

Our Game can be downloaded for following link:

Click to download

Only access with pulchowk account

If you have any queries please ask with this email address:

077bct003.aashish@pcampus.edu.np

077bct011.anmol@pcampus.edu.np

077bct015.avahan@pcampus.edu.np

077bct024.darshan@pcampus.edu.np

THANK YOU