A Project Report

On

Pong game

For The Course

“ Project-I”

By

Afrin

IT23039

Supervised by

Dr. Ziaur Rahman

Associate professor

Department of ICT, MBSTU



DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

SANTOSH, TANGAIL-1902, Dhaka, Bangladesh

Declaration

This is to certify that the work presented in this project is carried out by the candidate under the supervision of Dr. Ziaur Rahman in the department of Information and Communication Technology, MBSTU, Tangail, Bangladesh. It is also declared that neither of this project has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Signature of Supervisor Dr.Ziaur Rahman

Associate professor

Dept. of ICT, MBSTU

Acknowledgements

I must sense grateful to the Almighty Allah to complete the dissertation. At the outset, I would like to express gratitude to my supervisor Dr. Ziaur Rahman, Associate professor, Dept. of Information and Communication Technology, MBSTU who has supported our plan to continue Software Development Project-I. I also like to express gratitude to our supervisor, for his valuable guidance and insight, encouragement, support and reliance throughout the project. However, it is not possible acknowledge properly the effort of our honorable teacher in writing words. We are, as always, indebted to our family. The love and support of our parents remain bedrock of our life.

Contents

Page No.

Cover Page----------------------------------------------------------------- 01

Declaration----------------------------------------------------------------- 02

Acknowledgements------------------------------------------------------- 03

Chapter -1-------------------------------------------------------- 5

Introduction of project

Chapter – 2------------------------------------------------------- 6-10

2.1 – Programming Language

-- Types of programming language

2.2 – C++ Programming Language

-- Features of C++ Language

-- Advantage

– Disadvantage

2.3 – IDE details

Chapter – 3----------------------------------------------------- 10

3.1 –Header File

Chapter – 4 ---------------------------------------------------- 11

4.1 -- Function Name & Details

Chapter – 5 ---------------------------------------------------- 12-18

5.1 – Source Code

Chapter – 6 ---------------------------------------------------- 18-20

6.1 – Output console

Chapter – 7 ---------------------------------------------------- 21

7.1 – Conclusion

7.2 – Limitation

7.3 – Future Work

Chapter – 1

Introduction of Project

This console-based Pong game is a simple adaptation of the classic arcade game. The player controls a paddle at the bottom of the screen, using the left and right arrow keys to keep a bouncing ball from falling off the screen. Points are earned each time the ball hits the paddle, while missing the ball results in losing a life. The game features walls around the play area, adding complexity as the ball bounces off them unpredictably. The player’s goal is to achieve the highest possible score before running out of lives. It’s a fun, fast-paced game that challenges reflexes and timing.

Chapter 2

2.1 - Programming Language

Programming Language : As we know, to communicate with a person, we need a specific language, similarly to communicate with computers, programmers also need a language is called Programming language.

Before learning the programming language, let's understand what is language?

What is Language?

Language is a mode of communication that is used to share ideas, opinions with each other. For example, if we want to teach someone, we need a language that is understandable by both communicators.

What is a Programming Language?

A programming language is a computer language that is used by programmers (developers) to communicate with computers. It is a set of instructions written in any specific language ( C, C++, Java, Python) to perform a specific task.

A programming language is mainly used to develop desktop applications, websites, and mobile applications.

Types of programming language:

1. Low-level programming language: Low-level language is machine-dependent (0s and 1s) programming language. The processor runs low- level programs directly without the need of a compiler or interpreter, so the programs written in low-level language can be run very fast.

Low-level language is further divided into two parts –

i.Machine Language: Machine language is a type of low-level programming language. It is also called as machine code or object code. Machine language is easier to read because it is normally displayed in binary or hexadecimal form (base 16) form. It does not require a translator to convert the programs because computers directly understand the machine language programs.

The advantage of machine language is that it helps the programmer to execute the programs faster than the high-level programming language.

ii. Assembly Language:

Assembly language (ASM) is also a type of low-level programming language that is designed for specific processors. It represents the set of instructions in a symbolic and human-understandable form. It uses an assembler to convert the assembly language to machine language.

The advantage of assembly language is that it requires less memory and less execution time to execute a program.

2.High-level programming language:High-level programming language (HLL) is designed for developing user-friendly software programs and websites. This programming language requires a compiler or interpreter to translate the program into machine language (execute the program). The main advantage of a high-level language is that it is easy to read, write, and maintain.

High-level programming language includes Python, Java, JavaScript, PHP, C#, C++, Objective C, Cobol, Perl, Pascal, LISP, FORTRAN, and Swift programming language.

A high-level language is further divided into three parts-

i.Procedural Oriented programming language:

Procedural Oriented Programming (POP) language is derived from structured programming and based upon the procedure call concept. It divides a program into small procedures called routines or functions.

Procedural Oriented programming language is used by a software programmer to create a program that can be accomplished by using a programming editor like IDE, Adobe Dreamweaver, or Microsoft Visual Studio.

The advantage of POP language is that it helps programmers to easily track the program flow and code can be reused in different parts of the program. Example: C, FORTRAN, Basic, Pascal, etc. 03.Middle-level programming language: Middle-level programming language lies between the low-level program.

Example: C, FORTRAN, Basic, Pascal, etc.

03.Middle-level programming language:

Middle-level programming language lies between the low-level programming language and high-level programming language. It is also known as the intermediate programming language and pseudo-language. A middle-level programming language's advantages are that it supports the features of high-level programming, it is a user-friendly language, and closely related to machine language and human language.

Example: C, C++, language.

2.2 - C ++ Programming Language

C++ is a programming language that is an extension of an earlier language, C. For the most part, we will use the C subset of C++ in this course because it provides the tools that we need to explore physical data structures. A few of the language features that we will use are part of C++ but not of C. These notes make no attempt to offer a com

Most Important Features of C Language:

• Simple

• Abstract Data types.

• Machine Independent or Portable.

• Mid-level programming language.

• Structured programming language.

• Rich Library.

• Memory Management.

• Quicker Compilation.

Advantages:

• Object-Oriented. C++ is an object-oriented programming language which means that the main focus is on objects and manipulations around these objects. ...

• Speed. ...

• Compiled. ...

• Rich Library Support. ...

• Pointer Support. ...

• Closer to Hardware

Disadvantages:

• Object-orientated programming languages have several security issues which means that programs written in C++ aren't as safe as others.

• The pointers that are used in C++ take up a lot of memory which is not always suitable for some devices.

• Cannot support built-in code threads.

2.3 IDE details

Code::Blocks:

Code::Blocks is a free, open-source cross-platform IDE that supports multiple compilers including GCC, Clang and Visual C++. It is developed in C++ using wxWidgets as the GUI toolkit. Using a plugin architecture, its capabilities and features are defined by the provided plugins. Currently, Code::Blocks is oriented towards C, C++, and Fortran. It has a custom build system and optional Make support.

Features

Compilers

Code::Blocks supports multiple compilers, including GCC, MinGW, Digital Mars, Microsoft Visual C++, Borland C++, LLVM Clang, Watcom, LCC and the Intel C++ compiler. Although the IDE was designed for the C++ language, there is some support for other languages, including Fortran and D. A plug-in system is included to support other programming languages.

Code editor

The IDE features syntax highlighting and code folding (through its Scintilla editor component), C++ code completion, class browser, a hex editor and many other utilities. Opened files are organized into tabs. The code editor supports font and font size selection and personalized syntax highlighting colors.

Debugger

The Code::Blocks debugger has full breakpoint support. It also allows the user to debug their program by having access to the local function symbol and argument display, user-defined watches, call stack, disassembly, custom memory dump, thread switching, CPU registers and GNU Debugger Interface.

Chapter – 3

3.1 - Header files

These header files provide access to different libraries and functions in C++:

1. **#include <iostream>**
   * Provides the functionality for input and output operations.
2. **#include <conio.h>**

* Provides access to console input and output functions specific to the Windows operating system.

1. **#include <windows.h>**

Provides Windows-specific functions and data types.

Chapter – 4

4.1 - Function Name & Details

1. **The gotoxy Function**

The gotoxy function is a custom function that moves the cursor to a specified position in the console window. It uses Windows-specific API functions to control the console cursor.

void gotoxy(int x, int y) {

COORD coord; // COORD is a structure that holds coordinates

coord.X = x; // Set X-coordinate (column)

coord.Y = y; // Set Y-coordinate (row)

// Set the console cursor position

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coord);

}

1. system(“cls”): This is a function which clear the screen.The cursor is moved to upper left corner of the screen.

Chapter – 5

5.1 – Source Code

#include <iostream>

#include <conio.h>

#include <windows.h>

using namespace std;

void gotoxy(int x, int y)

{

COORD coord;

coord.X = x;

coord.Y = y;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coord);

}

void drawBoard( int paddleX, int ballX, int ballY, int score, int lives){

system("cls");

// Draw paddle

gotoxy(paddleX, 20);

cout << "======";

// Draw ball

gotoxy(ballX, ballY);

cout << "O";

// draw wall

for (int i = 0; i < 42; i++) {

gotoxy(i, 0);

cout << "#";

gotoxy(i, 24);

cout << "#";

}

for (int i = 0; i < 25; i++) {

gotoxy(0, i);

cout << "#";

gotoxy(41, i);

cout << "#”;

}

// Display score and lives

gotoxy(45, 5);

cout << "Score: " << score;

gotoxy(45, 7);

cout << "Lives: " << lives;

}//function

int main(){

system("mode con: lines=30 cols=60");

int paddleX = 20;

int ballX = 32;

int ballY = 19;

int ballDirX = 1;

int ballDirY = -1;

int score = 0;

int lives = 3;

cout<<"Instructions: "<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"1.Press right and left arrow keys for move paddle."<<endl;

cout<<"2.Press Esc key for quit game."<<endl;

system("pause");

while(lives > 0){

if(\_kbhit()){

char key = \_getch();

if (key == 75 && paddleX > 1){

paddleX--;

}

else if (key == 77 && paddleX < 34){

paddleX++;

}

else if (key == 27){

break;

}

}// keyboard hit

ballX += ballDirX;

ballY += ballDirY;

// Reverse ball direction if it hits the paddle

if (ballY == 20 && ballX >= paddleX && ballX <= paddleX + 5) {

ballDirY = -ballDirY;

score++;

}

// Reverse ball direction if it hits the walls

if (ballX == 1 || ballX == 40)

ballDirX = -ballDirX;

if (ballY == 1)

ballDirY = -ballDirY;

if (ballY == 23) {

lives--;

ballX = 20;

ballY = 19;

ballDirY = -ballDirY;

}

drawBoard(paddleX, ballX, ballY, score, lives);

Sleep(150);

}//while

system("cls");

gotoxy(15, 8);

cout<<"Game Over!"<<endl;

gotoxy(15, 9);

cout << "Final Score: " << score << endl;

gotoxy(15, 10);

cout << "Press any key to exit..." << endl;

\_getch();

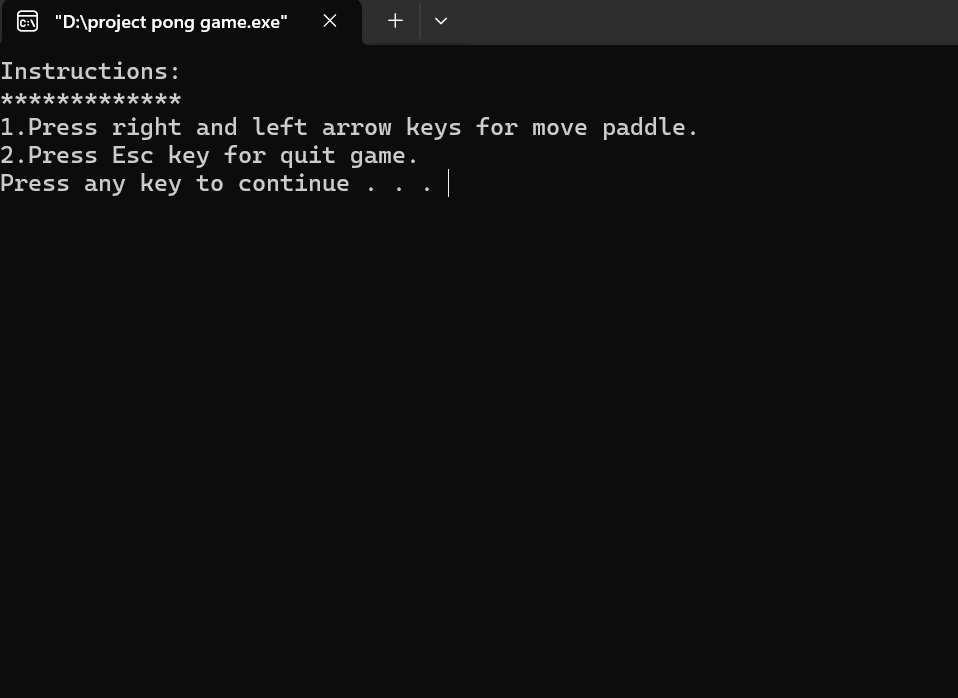
return 0;

}

Chapter – 6

6.1 – Output

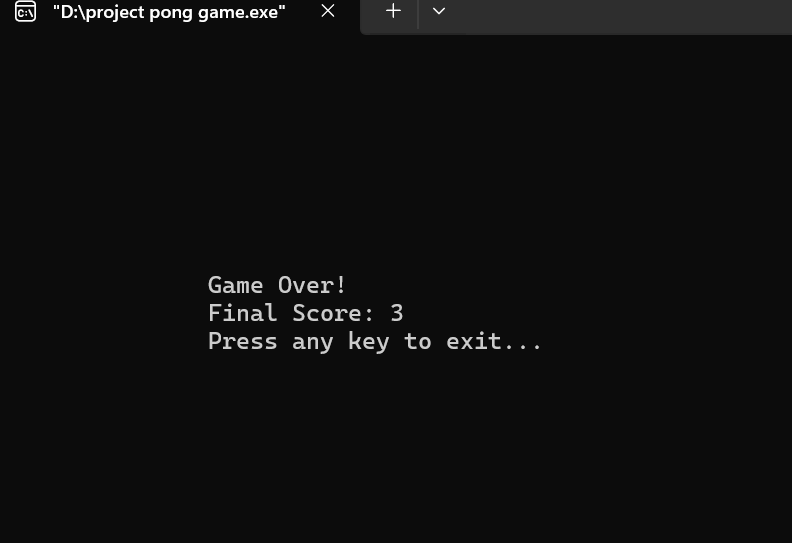
Output:



Start Game:



Result:



Chapter – 7

7.1 – Conclusion

For developing this project we faced some difficulties which are solved by the directions of our honorable supervisor sir. We are still working it for adding some additional features to make this project more user friendly.

7.2 – Limitation

This Pong game has limitations such as basic graphics confined to console display, lack of increasing difficulty or levels, and no multiplayer option. Additionally, it only runs on Windows due to its reliance on Windows-specific libraries, restricting its portability to other operating systems.

7.3– Future work

I know the limitation of our project. So I can improve it in future. I will upload out code at github and open to edit. So that others can contribute to our code and make is more efficient