MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR (M.P)
(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV ,Bhopal)

(SESSION : 2021 – 2023)

Skills Based Project Report

on

FLAPPY BIRD

Submitted By:-
SONALI MAGRIYA
(0901CA211060)

Mentor:-
DR. ANSHU CHATURVEDI
(Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE,
Gwalior – 474005 (MP) est.1957

July-December 2021

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Sonali Magriya** (0901CA211060) has submitted the project report titled FLAPPY BIRD under the mentorship of **Dr. Anshu Chaturvedi** (Professor) as the skills based mini project in 1st year of Master of Computer Application in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

DR. ANSHU CHATURVEDI
Professor
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR
(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV ,Bhopal)

DECLARATION

I have declare that the work being presented in this project report, for the fulfilment of partial requirement of the skills based mini project in 1st year of Master of Computer Application in Computer Science and Engineering at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **DR. ANSHU CHATURVEDI(Professor**), MITS Gwalior.

I declare that I have not submitted the matter embodied in this report anywhere else.

SONALI MAGRIYA
(0901CA211060)
(1st Year)
Master of Computer Application,
Computer Science and Engineering

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my Institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary project. I extend my gratitude to the Director of the institute**, DR. R.K. PANDIT** and Dean Academics, **DR. MANJAREE PANDIT** for this.

I would sincerely like to thank my department, **Department of Computer Science and Engineering**, for allowing me to explore this project. I humbly thank **DR. MANISH DIXIT**, Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my Faculty coordinator. I am grateful to the guidance of **DR. ANSHU CHATURVEDI**(Professor), Computer Science and Engineering, for her continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Sonali Magriya(0901CA211060)

(1st year)

MCA(Master of Computer Application)

DEPARTMENT OF CSE

ABSTRACT

Our mini project tittle is “FLAPPY BIRD” C++, we students of master of computer application has done a C++ program for FLAPPY BIRD game. We have done this using the feature in c++ called class. We have used features of c++. We have created different functions for different menus in the games the bird based on the speed at which a player flaps her arms. The FPGA will render an image of the bird flying through an environment, and display the flapping motion of the wings according to the player’s speed. contains several technical and logical problems. We have done this for mini project

COVER PAGE

CERTIFICATE

DECLARATION

ACKNOLEDGEMENT

ABSTRACT

CONTENTS

TITLE **PAGE NO.**

Chapter 1: INTRODUCTION

Introduction

Our project is a new version of Flappy Bird Game. It is a game program written in cpp. Users can play this game play it in their computer. It contains two play modes, single player and dual player modes. In single player mode, each player have 3 lives and final scores will be written to a score board, which keeps track of top 10 scores. The dual player mode gives users the opportunity to compete with their friends. We also implemented two background settings, daytime and night modes. The users could also choose either one background mode before playing the game. One difficulty for users is that the horizontal shifting speed will increasing as time goes on. We also implemented a hidden trick in the single player mode. When user achieves 10 scores, the flappy bird will evolve into flappy Joe.

The goal

Using learning algorithms we want to allow player to do a training set with different definitions of states (more details about those differences will discuss later), and learn the flappy bird world. After the training set we want the player to successfully continue playing as long as we want without any hit. We can examine the learning success using different parameters as number of training runs, learning rate, discount factor.

The Approach

We choose to use learning against search methods because of Flappy Bird is a continuous and dynamic game, the environment is not predefined, and we wanted to create agent that can take a first phase of learning and then play the game fluently. In order to make effective learning we needed to defined the state space of the problem. We found that there are 3 elements that we want to include in the state space:

- ❖Vertical distance from the lower pipe.
- ❖Horizontal distance from the lower pipe.
- ❖Bird's velocity. We assumed that those 3 elements are the most important to describe a current state of the bird.

Chapter 2: OBJECTIVE

Our objective for this project is to implement the game . We try to create user-friendly interfaces to allow user better understand the game instructions and workflows. The game allows users to choose playing modes and background settings, and to control bird using touchscreen or keyboard buttons. It also contains a score board function to track previous history.

Chapter 3: CODE

```
#include<iostream>
#include<conio.h>
#include<dos.h>
#include<stdlib.h>
#include<string.h>
#include <windows.h>
#include <time.h>

#define SCREEN_WIDTH 90
#define SCREEN_HEIGHT 26
#define WIN_WIDTH 70
#define MENU_WIDTH 20
#define GAP_SIZE 7
#define PIPE_DIF 45

using namespace std;
//		cout<<"±±±±±±±±";

HANDLE console = GetStdHandle(STD_OUTPUT_HANDLE);
COORD CursorPosition;

int pipePos[3];
int gapPos[3];
int pipeFlag[3];
char bird[2][6] = { '/','-','-','o','\\',' ',
                                        '|',',',',_',' ','>' };
int birdPos = 6;
int score = 0;

void gotoxy(int x, int y)
{
	CursorPosition.X = x;
	CursorPosition.Y = y;
	SetConsoleCursorPosition(console, CursorPosition);
}

void setcursor(bool visible, DWORD size)
```

```
{
	if(size == 0)
		size = 20;

	CONSOLE_CURSOR_INFO lpCursor;
	lpCursor.bVisible = visible;
	lpCursor.dwSize = size;
	SetConsoleCursorInfo(console,&lpCursor);
}
void drawBorder(){

	for(int i=0; i<SCREEN_WIDTH; i++){
		gotoxy(i,0); cout<<"±";
		gotoxy(i,SCREEN_HEIGHT); cout<<"±";
	}

	for(int i=0; i<SCREEN_HEIGHT; i++){
		gotoxy(0,i); cout<<"±";
		gotoxy(SCREEN_WIDTH,i); cout<<"±";
	}
	for(int i=0; i<SCREEN_HEIGHT; i++){
		gotoxy(WIN_WIDTH,i); cout<<"±";
	}
}
void genPipe(int ind){
	gapPos[ind] = 3 + rand()%14;
}
void drawPipe(int ind){
	if( pipeFlag[ind] == true ){
		for(int i=0; i<gapPos[ind]; i++){
			gotoxy(WIN_WIDTH-pipePos[ind],i+1); cout<<"*";
		}
		for(int i=gapPos[ind]+GAP_SIZE; i<SCREEN_HEIGHT-1; i++){
			gotoxy(WIN_WIDTH-pipePos[ind],i+1); cout<<"*";
		}
	}
}
void erasePipe(int ind){
	if( pipeFlag[ind] == true ){
		for(int i=0; i<gapPos[ind]; i++){
```

```
                gotoxy(WIN_WIDTH-pipePos[ind],i+1); cout<<"   ";
            }
            for(int i=gapPos[ind]+GAP_SIZE; i<SCREEN_HEIGHT-1; i++){
                gotoxy(WIN_WIDTH-pipePos[ind],i+1); cout<<"   ";
            }
        }
}
void drawBird(){
        for(int i=0; i<2; i++){
            for(int j=0; j<6; j++){
                gotoxy(j+2,i+birdPos); cout<<bird[i][j];
            }
        }
}
void eraseBird(){
        for(int i=0; i<2; i++){
            for(int j=0; j<6; j++){
                gotoxy(j+2,i+birdPos); cout<<" ";
            }
        }
}
int collision(){
        if( pipePos[0] >= 61  ){
            if( birdPos<gapPos[0] || birdPos >gapPos[0]+GAP_SIZE ){
//                  cout<< " HIT ";
//                  getch();
                    return 1;
            }
        }
        return 0;
}
void debug(){
//      gotoxy(SCREEN_WIDTH + 3, 4); cout<<"Pipe Pos: "<<pipePos[0];
}
void gameover(){
        system("cls");
        cout<<endl;
        cout<<"\t\t--------------------------"<<endl;
        cout<<"\t\t-------- Game Over -------"<<endl;
        cout<<"\t\t--------------------------"<<endl<<endl;
```

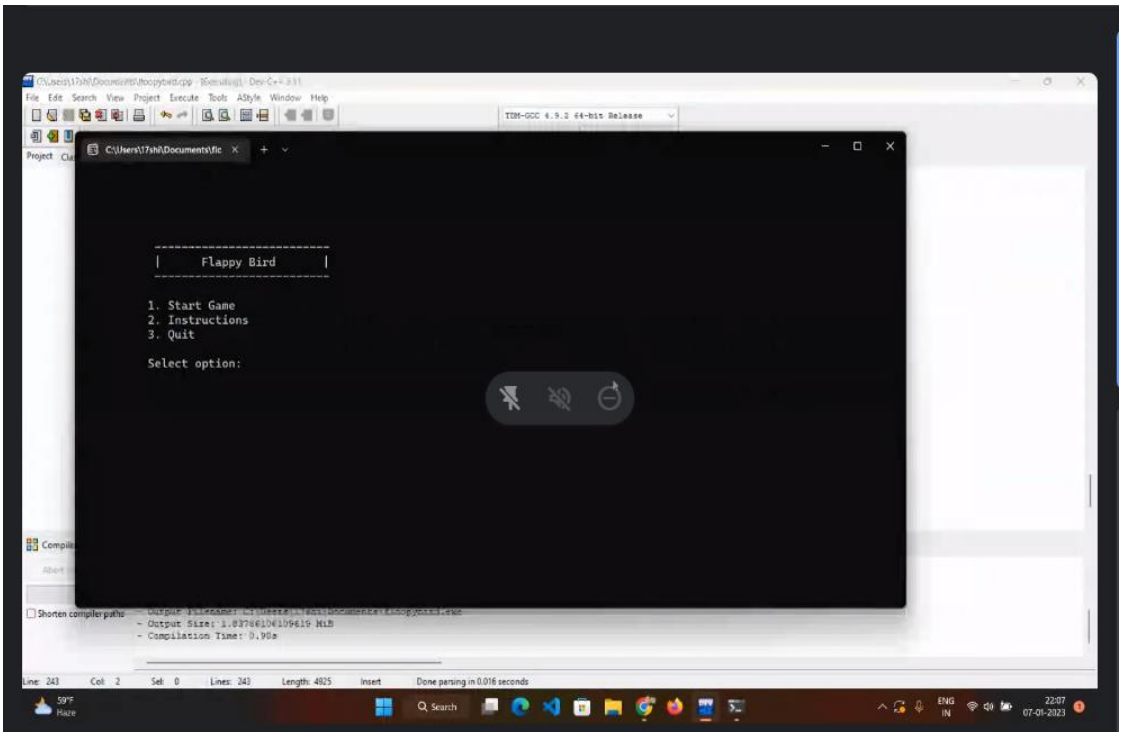
```
	cout<<"\t\tPress any key to go back to menu.";
	getch();
}
void updateScore(){
	gotoxy(WIN_WIDTH + 7, 5);cout<<"Score: "<<score<<endl;
}
void instructions(){
	system("cls");
	cout<<"Instructions";
	cout<<"\n----------------";
	cout<<"\n Press spacebar to make bird fly";
	cout<<"\n\nPress any key to go back to menu";
	getch();
}
void play(){

	birdPos = 6;
	score = 0;
	pipeFlag[0] = 1;
	pipeFlag[1] = 0;
	pipePos[0] = pipePos[1] = 4;

	system("cls");
	drawBorder();
	genPipe(0);
	updateScore();

	gotoxy(WIN_WIDTH + 5, 2);cout<<"FLAPPY BIRD";
	gotoxy(WIN_WIDTH + 6, 4);cout<<"----------";
	gotoxy(WIN_WIDTH + 6, 6);cout<<"----------";
	gotoxy(WIN_WIDTH + 7, 12);cout<<"Control ";
	gotoxy(WIN_WIDTH + 7, 13);cout<<"-------- ";
	gotoxy(WIN_WIDTH + 2, 14);cout<<" Spacebar = jump";

	gotoxy(10, 5);cout<<"Press any key to start";
	getch();
	gotoxy(10, 5);cout<<"                      ";
	while(1){
		if(kbhit()){
			char ch = getch();
```

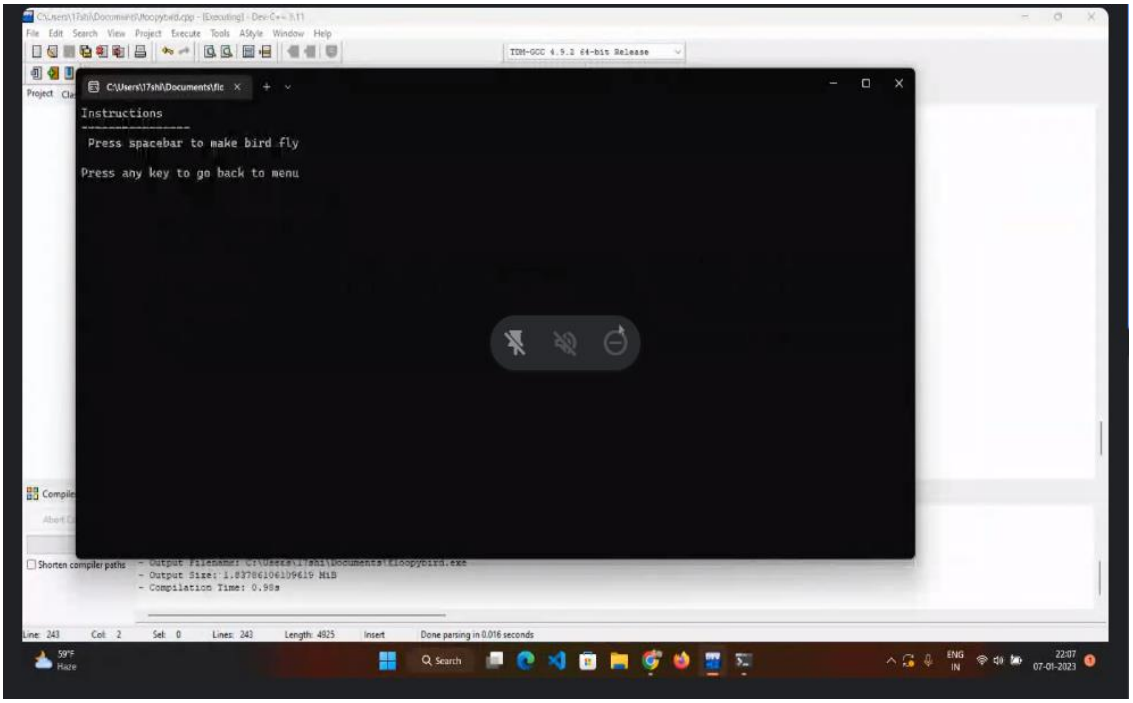
```
        if(ch==32){
                if( birdPos > 3 )
                        birdPos-=3;
        }
        if(ch==27){
                break;
        }
}

drawBird();
drawPipe(0);
drawPipe(1);
debug();
if( collision() == 1 ){
        gameover();
        return;
}
Sleep(100);
eraseBird();
erasePipe(0);
erasePipe(1);
birdPos += 1;

if( birdPos > SCREEN_HEIGHT - 2 ){
        gameover();
        return;
}
if( pipeFlag[0] == 1 )
        pipePos[0] += 2;

if( pipeFlag[1] == 1 )
        pipePos[1] += 2;

if( pipePos[0] >= 40 && pipePos[0] < 42 ){
        pipeFlag[1] = 1;
        pipePos[1] = 4;
        genPipe(1);
}
if( pipePos[0] > 68 ){
        score++;
```

```
                updateScore();
                pipeFlag[1] = 0;
                pipePos[0] = pipePos[1];
                gapPos[0] = gapPos[1];
            }
        }
}
int main()
{
        setcursor(0,0);
        srand( (unsigned)time(NULL));
//      play();
        do{
            system("cls");
            gotoxy(10,5); cout<<" -------------------------- ";
            gotoxy(10,6); cout<<" |      Flappy Bird       | ";
            gotoxy(10,7); cout<<" --------------------------";
            gotoxy(10,9); cout<<"1. Start Game";
            gotoxy(10,10); cout<<"2. Instructions";
            gotoxy(10,11); cout<<"3. Quit";
            gotoxy(10,13); cout<<"Select option: ";
            char op = getche();

            if( op=='1') play();
            else if( op=='2') instructions();
            else if( op=='3') exit(0);

        }while(1);

        return 0;
}
```

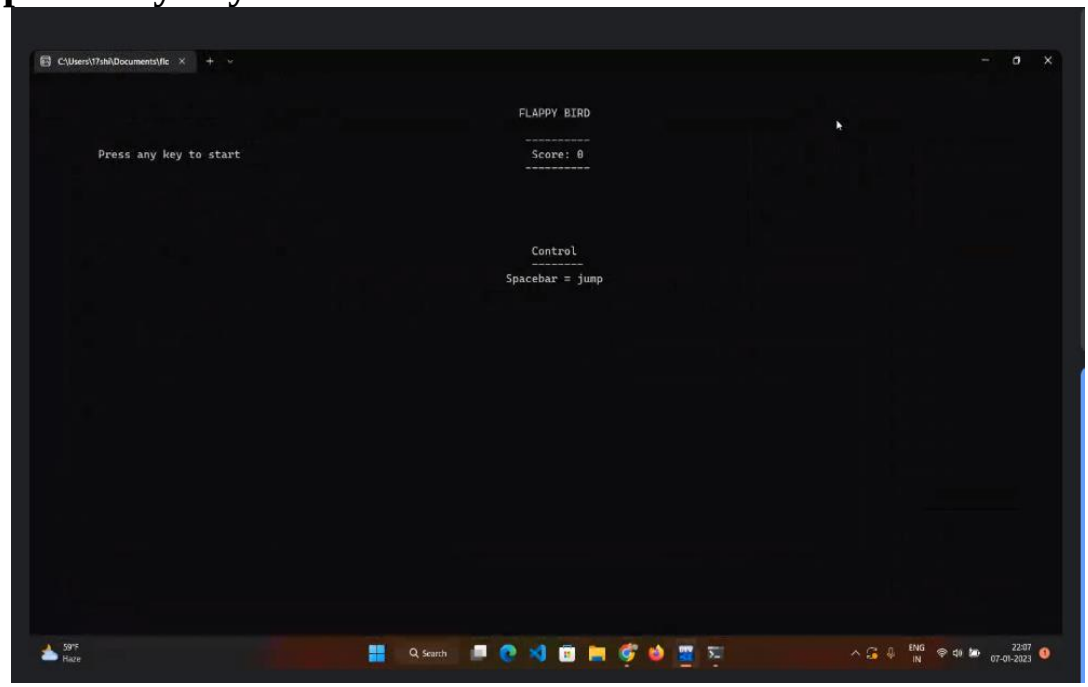
Chapter 4: SCREENSHOTS

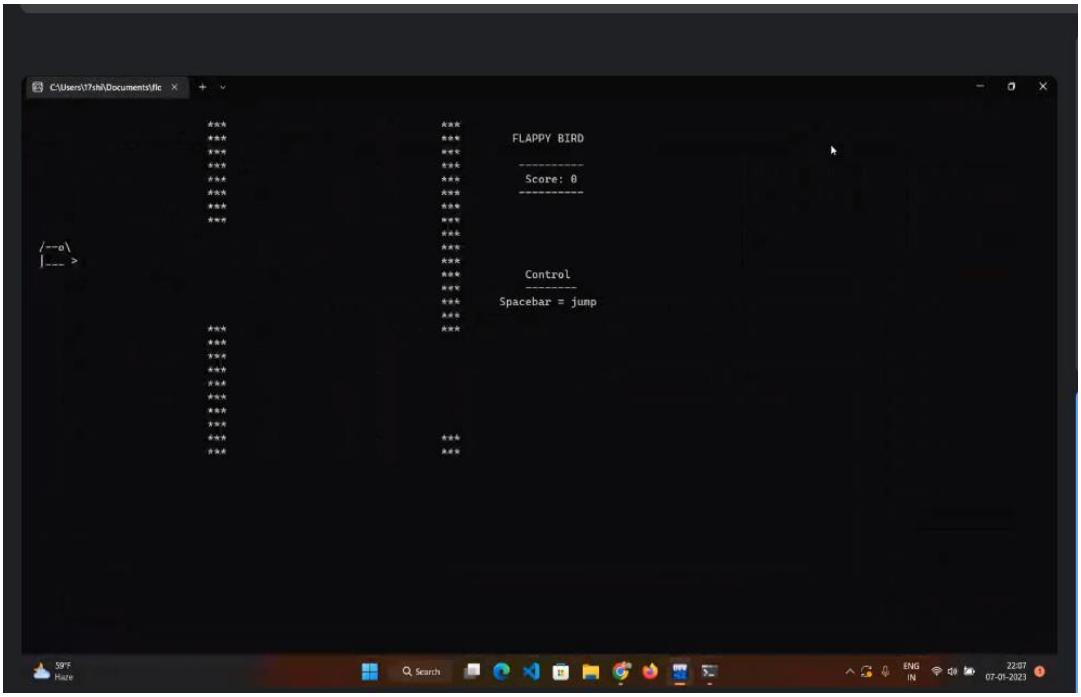
start here



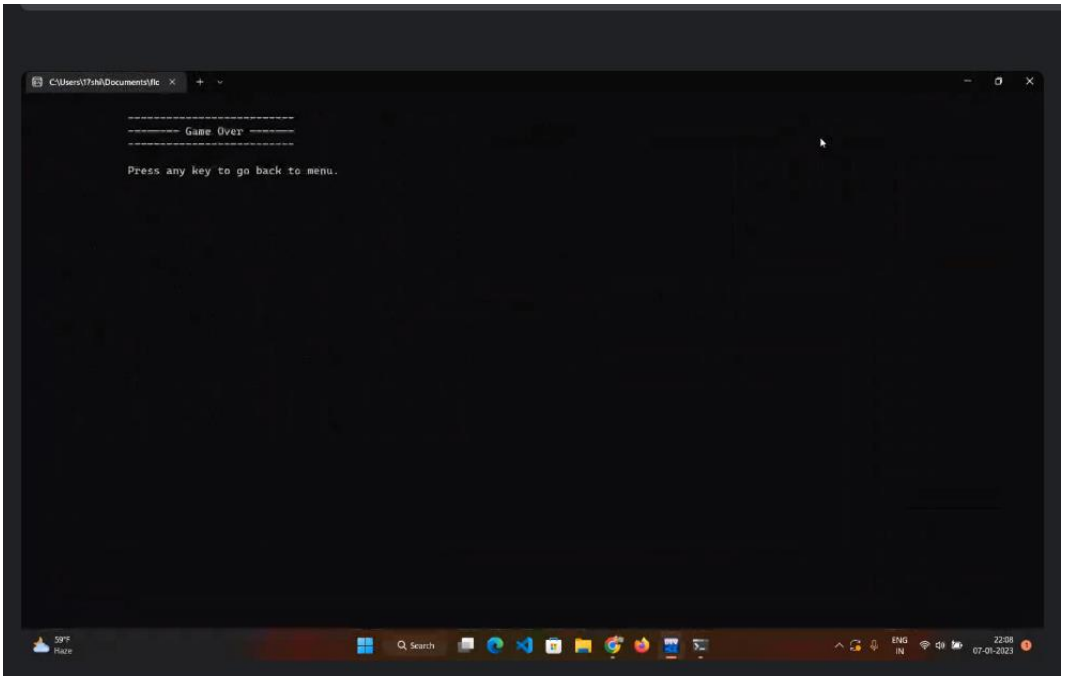


press any key to start





Over here



Chapter 5: CONCLUSION

Our final project was planned, developed and demonstrated as expected. We designed a new version of Flappy Bird Game written in CPP, which could be played personal computer. Firstly, a user-friendly interface was implemented. Secondly, single player mode was realized. In single player mode, each player have 3 lives and final scores will be written to a score board, which keeps track of top 10 scores. Then the code of dual player mode was written. It gives users the opportunity to compete with their friends, which will bring a lot of fun. We also implemented two background settings, daytime and night modes. The users could also choose either one background mode before playing the game. One difficulty for users is that the horizontal shifting speed will increasing as time goes on. We also implemented a hidden trick in the single player mode. When user achieves 10 scores, the flappy bird will evolve into flappy Joe.

Chapter 6: REFERENCES

https://www.w3schools.com/

https://www.thestudygenius.com/