COMPUTER GRAPHICS MINI PROJECT

PROJECT NAME: ENDLESS FLYER

GROUP MEMBERS: PRATHAMESH THAKUR: D7A -63

SHREYAS SAWANT: D7A-55

HARIHARAN IYER: D7A-22

ABSTRACT:

This project explores the use of physical law (gravitational law) for game development. The game in its exactness follows rules similar to that of the original game known as flappy bird.
The rules and regulations of the game are very basic and easy to understand. The game scenario puts the player in control of hovering a bird on the screen with moving obstacles of walls.
The goal of the game is to get through the opening in the obstacles without colliding with them. The game is implemented in C++. The case study discusses the use of simpler motion and modification to build the abstraction. The game is in single-player mode.
The user can interact with the game using the keyboard. The position of the obstacles will also be discussed. The game is a 2D game with the purpose to keep the bird moving forward and avoid a collision.

TABLE OF CONTENTS

ABSTRACT..... I
Chapter 1: Introduction.....3-4
1.1 Background.....3
1.2 Features.....3
1.3 Objectives.....4

Chapter 2: System Study..... 4
2.1 System requirements……4

Chapter 3: Results.....4-9
3.1 Code screenshots.....4-8
3.2 Output screenshots……8-9

Chapter 4: Conclusion.....10
4.1 Conclusion.....10

Chapter 1: Introduction

1.1 Background

History

Flappy Dot is an abstraction of the original game Flappy Bird. **Flappy Bird,** first seen in 2013, was developed by Nguy n Hà Đông (Dong Nguyen) a small, independent game developer based in Vietnam and published by GEARS Studios, also based in Vietnam. The game is where the player controls a bird, attempting to fly between rows of green pipes without hitting them. *Flappy Bird* was removed from both and by its creator on February 10, 2014, due to guilt over what he considered to be its addictive nature and overuse. The game's popularity and sudden removal caused phones with it pre-installed to be put up for sale for high prices over the Internet. Games similar to *Flappy Bird b*ecame popular on the iTunes App Store in the wake of its removal, and both Apple and Google have removed games from their app stores for being too similar to the original. The game has also been distributed through unofficial channels on multiple platforms.

1.2 Features:

This project is the replica of flappy bird with some changes and some of the features of this project are:

- GUI Interface
- Single player game
- Keyboard interaction
- No graphics

1.3 Objectives:

We made this game or project so that we could give our best in possible ways and show what we learned. The objectives of this project are:

- To play the famous game flappy bird on the computer.
- To make it user-friendly.
- To provide an easy interface.
- To entertain people in their leisure time.

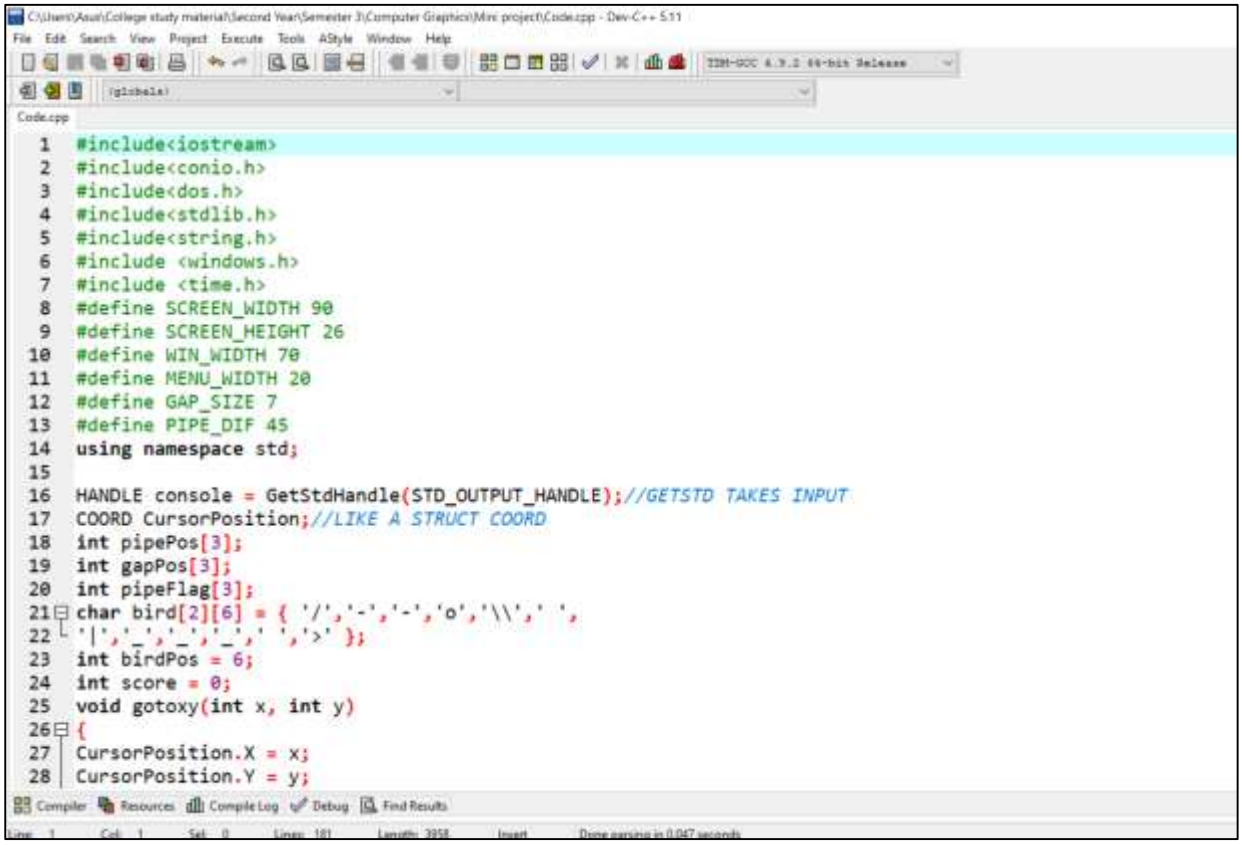
Chapter 2: System Study

2.1 System requirements:

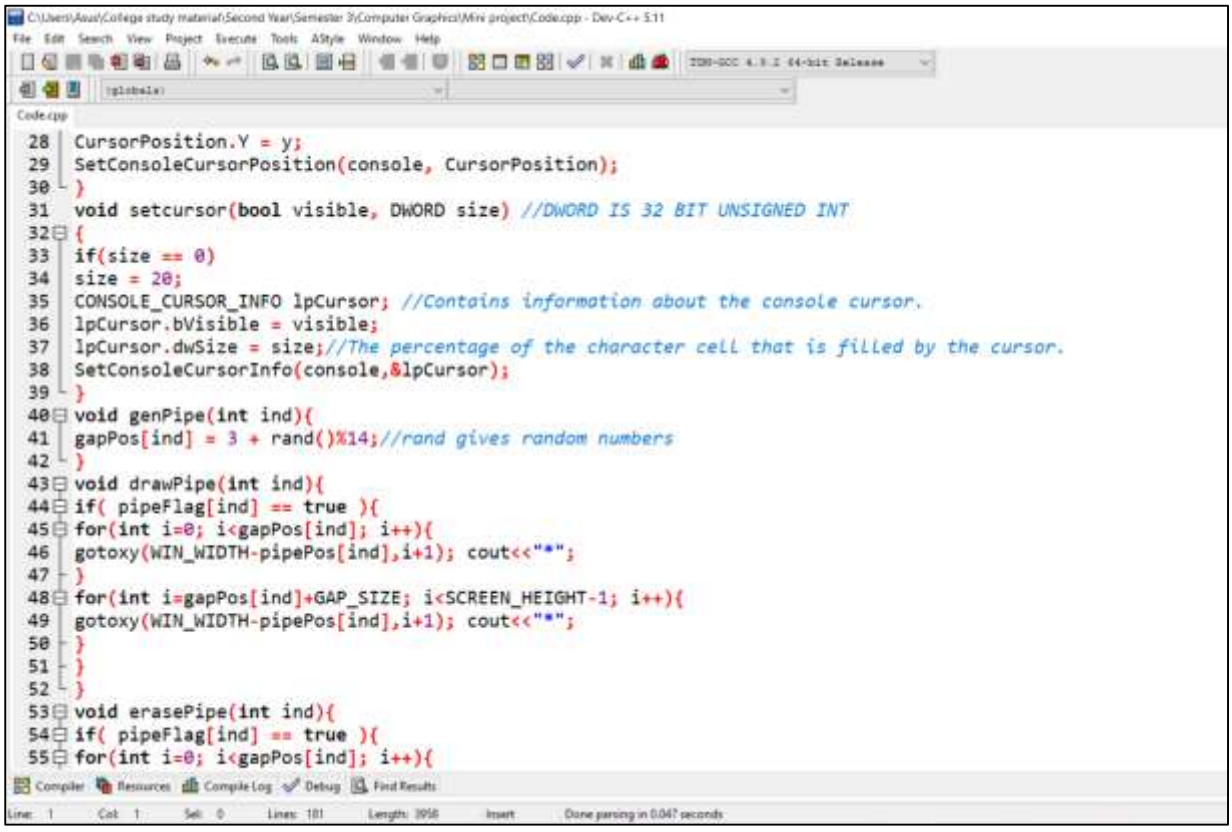
HARDWARE:
Operating System: Windows XP or more
Software: Turbo C/C++ OR DEV C++

Chapter 3: Results

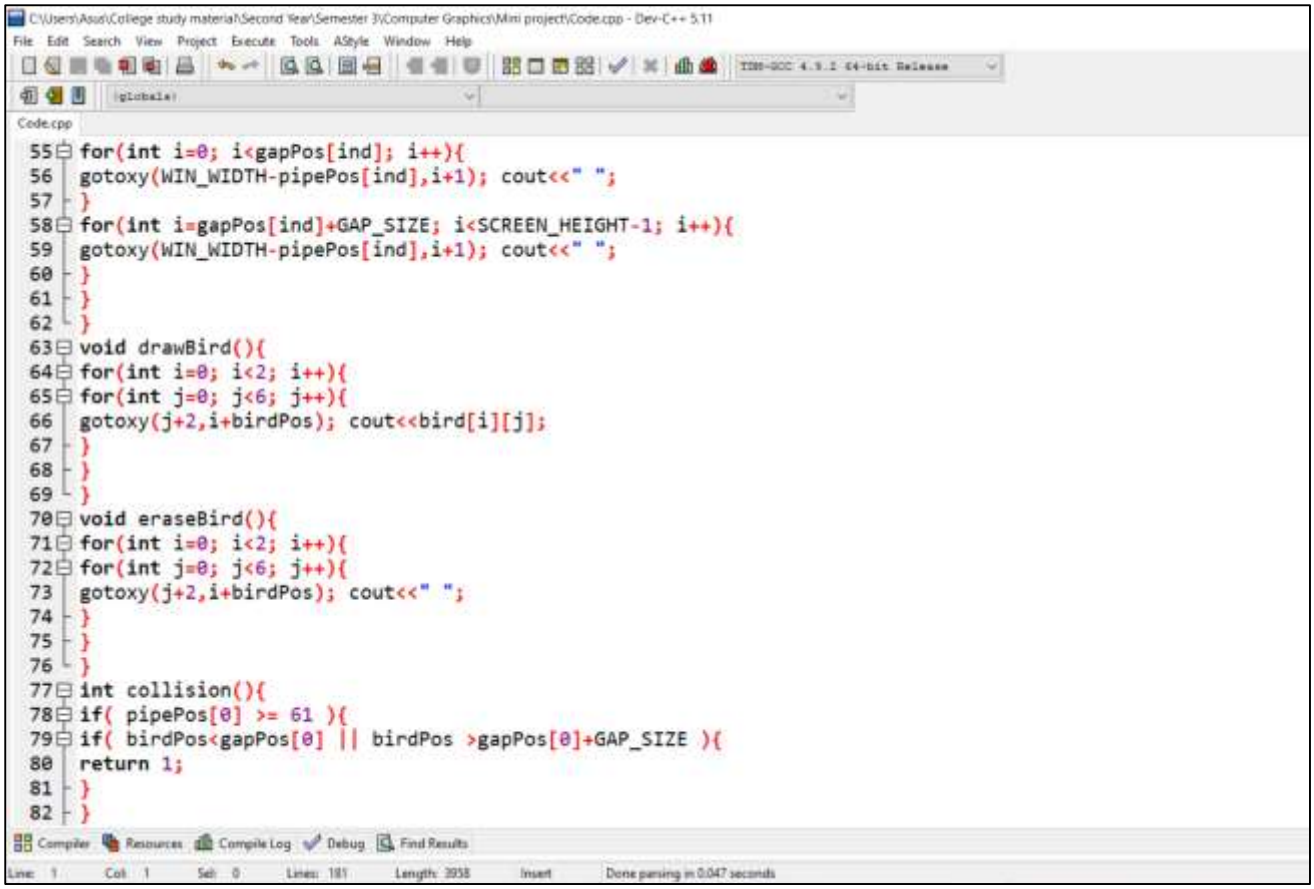
3.1 Code Screenshots:



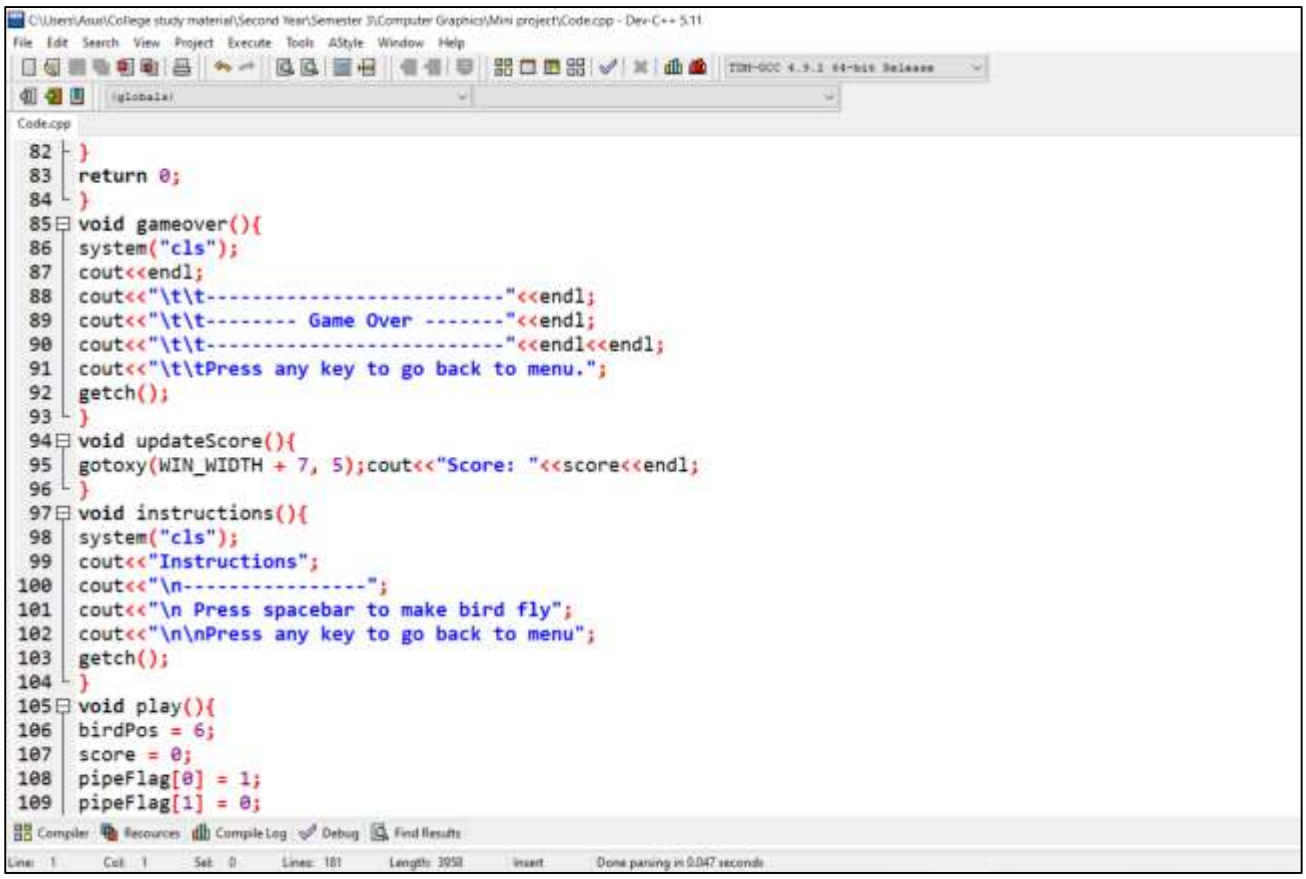
```
#include<iostream>
#include<conio.h>
#include<dos.h>
#include<stdlib.h>
#include<string.h>
#include <windows.h>
#include <time.h>
#define SCREEN_WIDTH 90
#define SCREEN_HEIGHT 26
#define WIN_WIDTH 70
#define MENU_WIDTH 20
#define GAP_SIZE 7
#define PIPE_DIF 45
using namespace std;

HANDLE console = GetStdHandle(STD_OUTPUT_HANDLE);//GETSTD TAKES INPUT
COORD CursorPosition;//LIKE A STRUCT COORD
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;
```



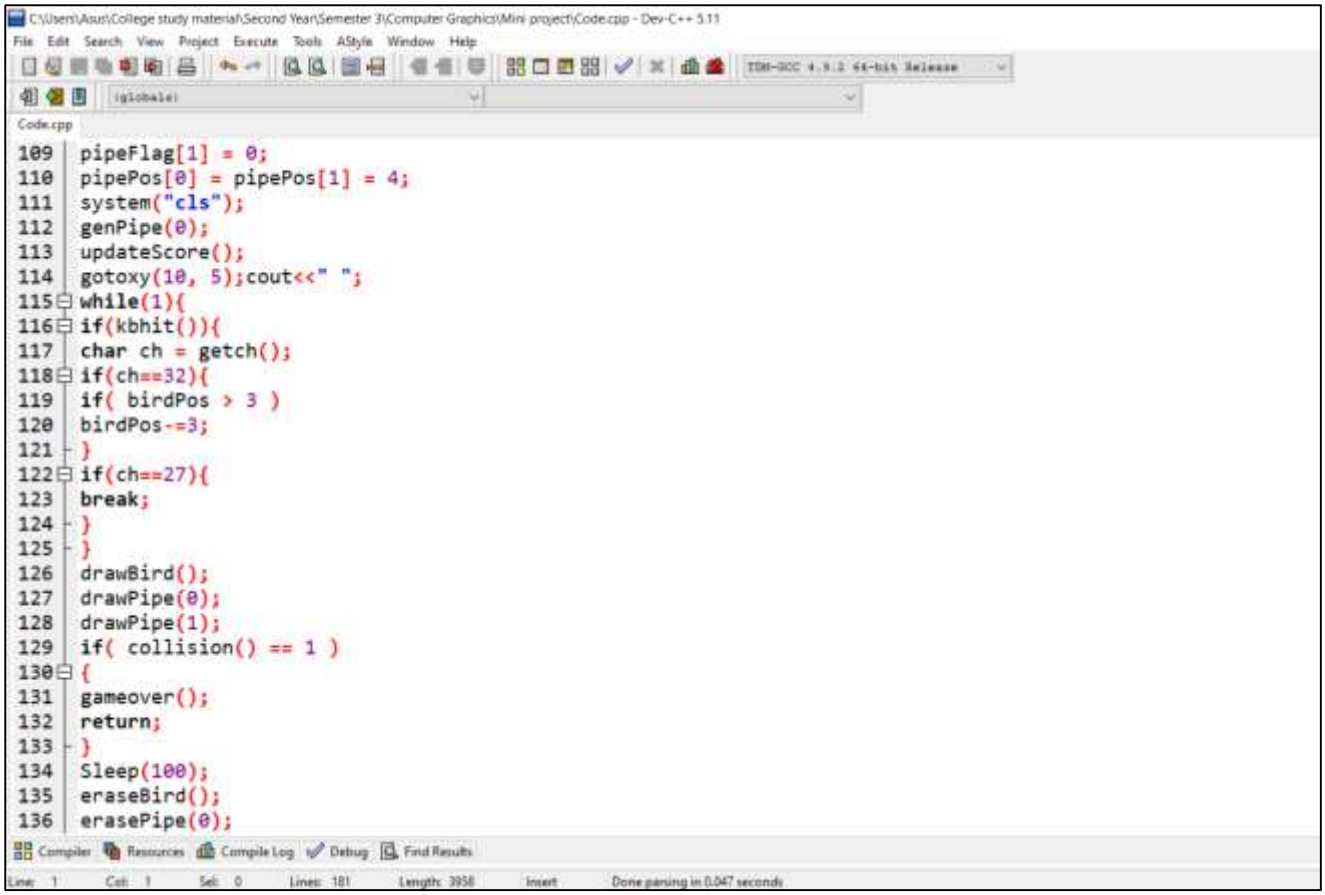
```
CursorPosition.Y = y;
SetConsoleCursorPosition(console, CursorPosition);
}
void setcursor(bool visible, DWORD size) //DWORD IS 32 BIT UNSIGNED INT
{
if(size == 0)
size = 20;
CONSOLE_CURSOR_INFO lpCursor; //Contains information about the console cursor.
lpCursor.bVisible = visible;
lpCursor.dwSize = size;//The percentage of the character cell that is filled by the cursor.
SetConsoleCursorInfo(console,&lpCursor);
}
void genPipe(int ind){
gapPos[ind] = 3 + rand()%14;//rand gives random numbers
}
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++){
```

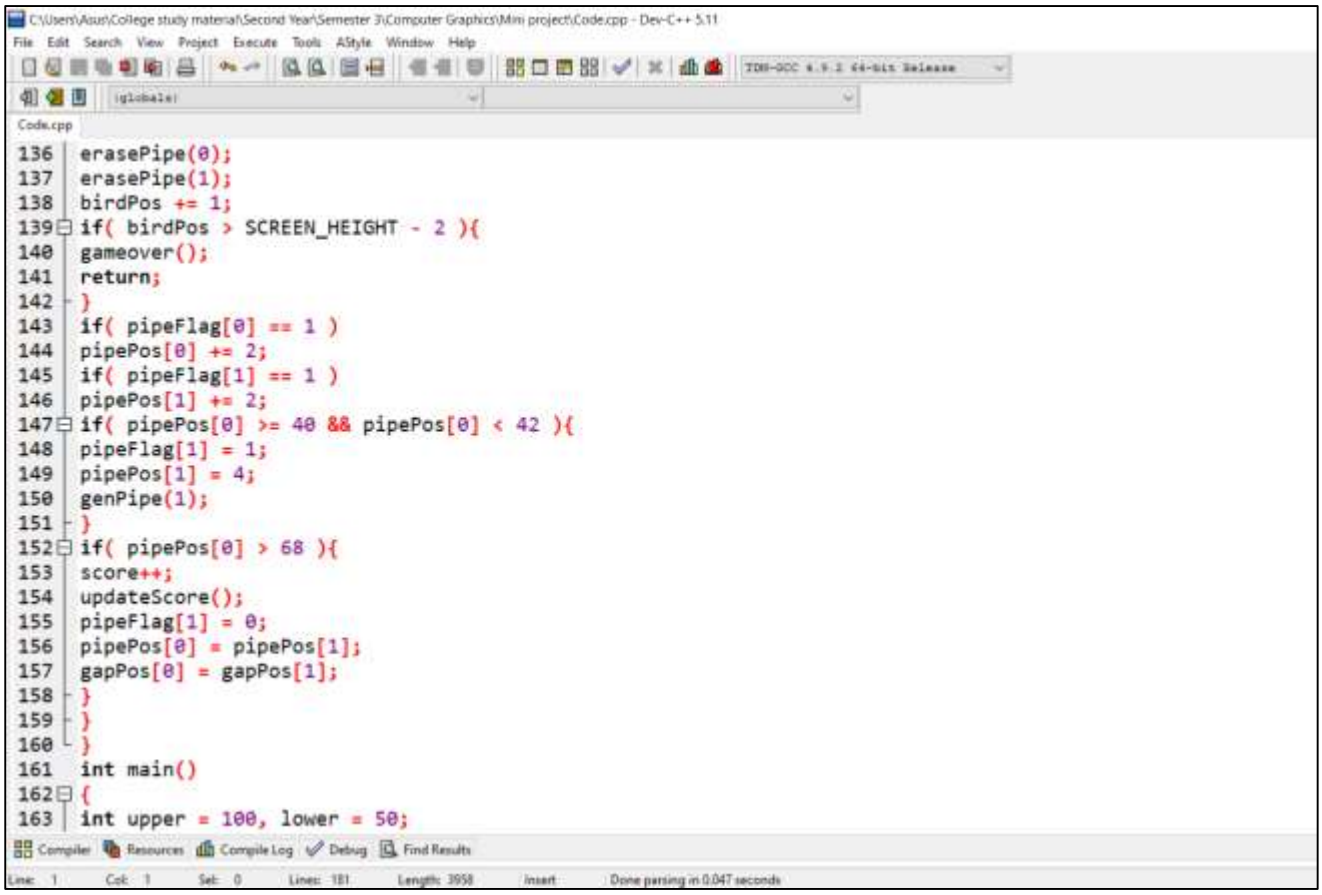


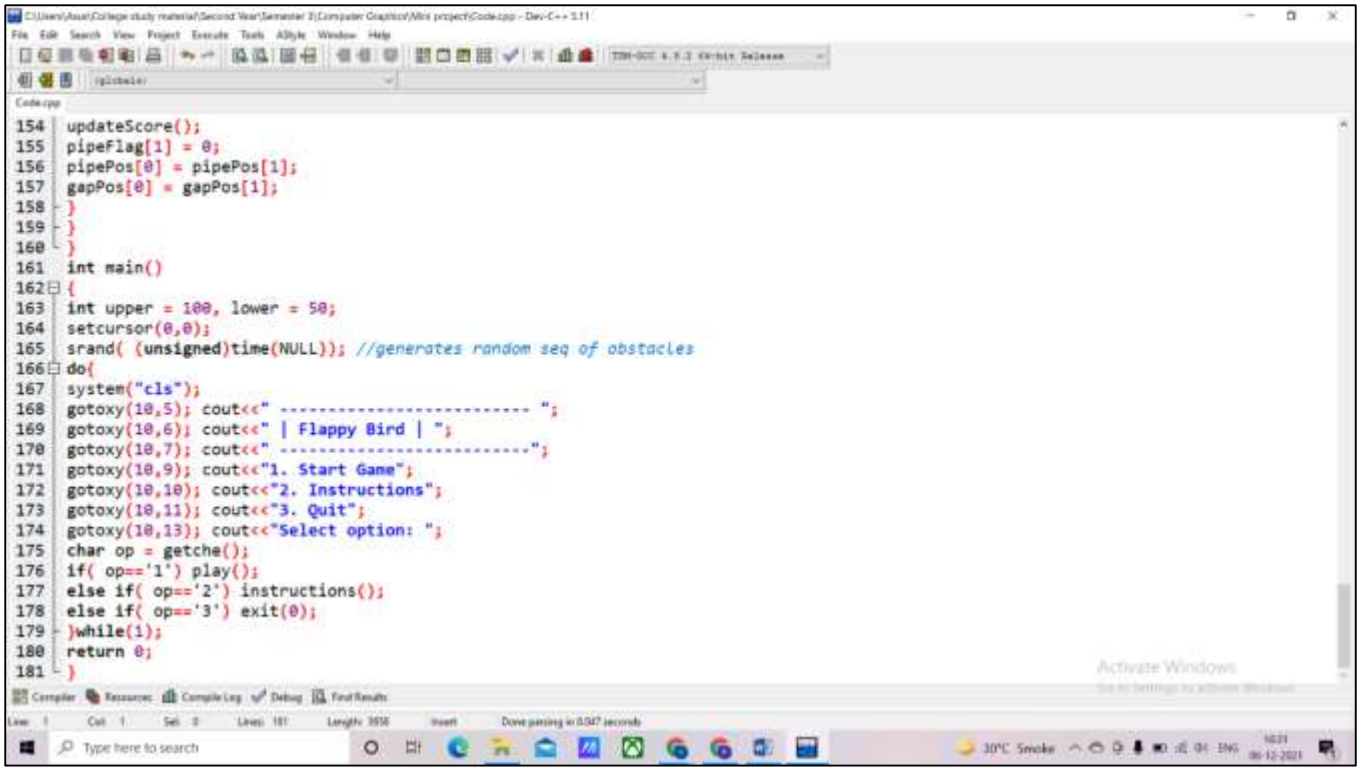
```
for(int i=0; i<gapPos[ind]; i++){
gotoxy(WIN_WIDTH-pipePos[ind],i+1); cout<<" ";
}
for(int i=gapPos[ind]+GAP_SIZE; i<SCREEN_HEIGHT-1; i++){
gotoxy(WIN_WIDTH-pipePos[ind],i+1); cout<<" ";
}
}
}
void drawBird(){
for(int i=0; i<2; i++){
for(int j=0; j<6; j++){
gotoxy(j+2,i+birdPos); cout<<bird[i][j];
}
}
}
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 ){
return 1;
}
}
```



```
}
return 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;
```

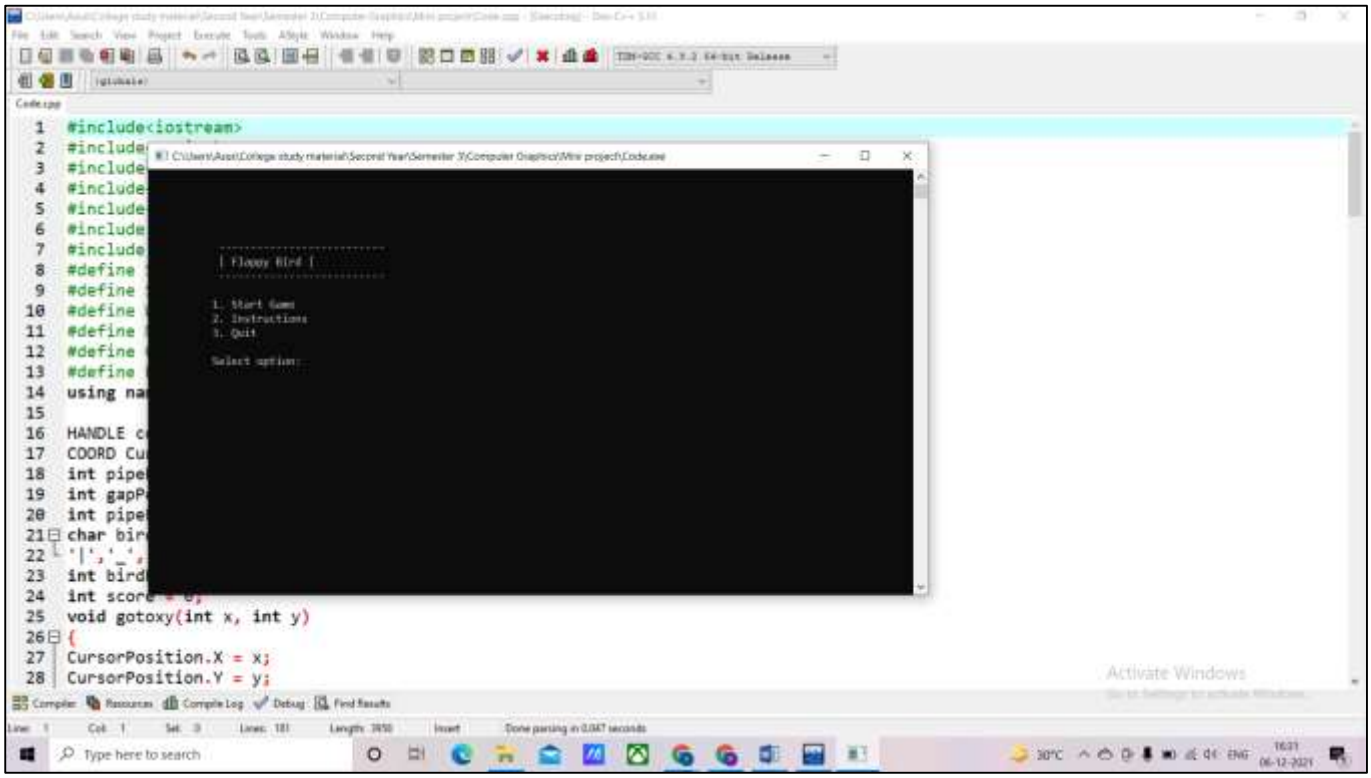

C:\Users\Asus\College study material\Second Year\Semester 3\Computer Graphics\Mini project\Code.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Code.cpp
109 pipeFlag[1] = 0;
110 pipePos[0] = pipePos[1] = 4;
111 system("cls");
112 genPipe(0);
113 updateScore();
114 gotoxy(10, 5);cout<<" ";
115 while(1){
116 if(kbhit()){
117 char ch = getch();
118 if(ch==32){
119 if(birdPos > 3)
120 birdPos-=3;
121 }
122 if(ch==27){
123 break;
124 }
125 }
126 drawBird();
127 drawPipe(0);
128 drawPipe(1);
129 if(collision() == 1)
130 {
131 gameover();
132 return;
133 }
134 Sleep(100);
135 eraseBird();
136 erasePipe(0);
Compiler Resources Compile Log Debug Find Results
Line: 1 Col: 1 Sel: 0 Lines: 181 Length: 3958 Insert Done parsing in 0.047 seconds


C:\Users\Asus\College study material\Second Year\Semester 3\Computer Graphics\Mini project\Code.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Code.cpp
136 erasePipe(0);
137 erasePipe(1);
138 birdPos += 1;
139 if(birdPos > SCREEN_HEIGHT - 2){
140 gameover();
141 return;
142 }
143 if(pipeFlag[0] == 1)
144 pipePos[0] += 2;
145 if(pipeFlag[1] == 1)
146 pipePos[1] += 2;
147 if(pipePos[0] >= 40 && pipePos[0] < 42){
148 pipeFlag[1] = 1;
149 pipePos[1] = 4;
150 genPipe(1);
151 }
152 if(pipePos[0] > 68){
153 score++;
154 updateScore();
155 pipeFlag[1] = 0;
156 pipePos[0] = pipePos[1];
157 gapPos[0] = gapPos[1];
158 }
159 }
160 }
161 int main()
162 {
163 int upper = 100, lower = 50;
Compiler Resources Compile Log Debug Find Results
Line: 1 Col: 1 Sel: 0 Lines: 181 Length: 3958 Insert Done parsing in 0.047 seconds



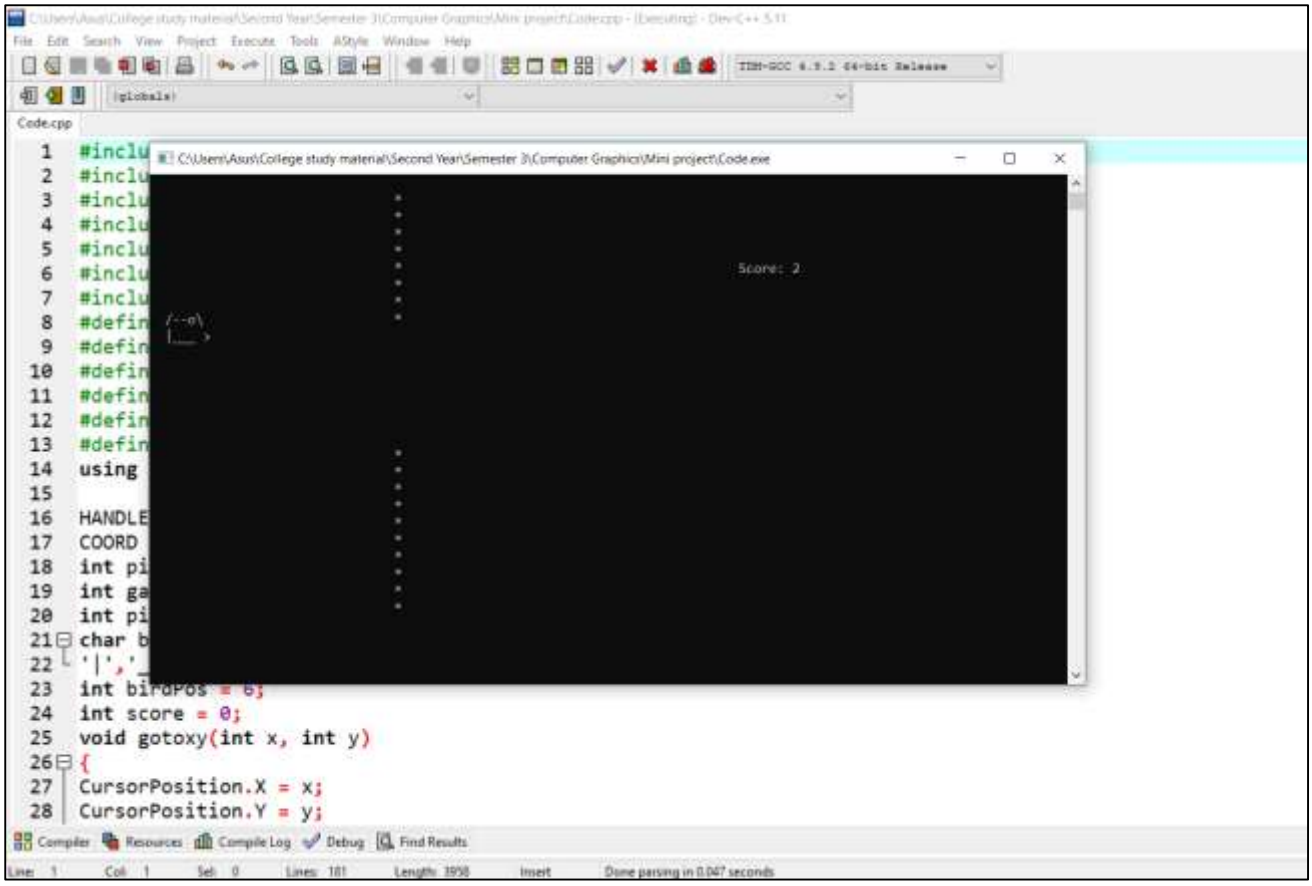
```
updateScore();
pipeFlag[1] = 0;
pipePos[0] = pipePos[1];
gapPos[0] = gapPos[1];
}
}
}
int main()
{
int upper = 100, lower = 50;
setcursor(0,0);
srand( (unsigned)time(NULL)); //generates random seq of obstacles
do{
system("cls");
gotoxy(10,5); cout<<" ------------------------- ";
gotoxy(10,6); cout<<" | 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;
}
```

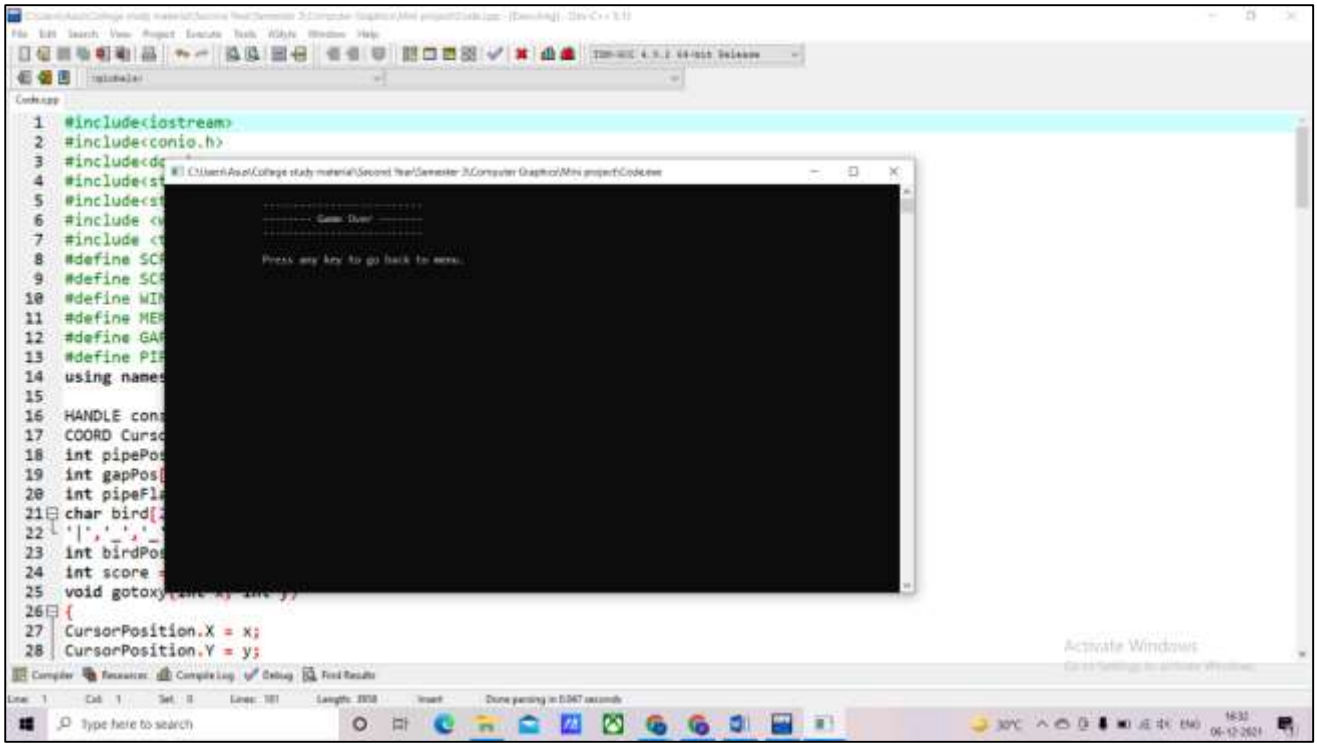
3.2 Output screenshots:



```
#include<iostream>
#include
#include
#include
#include
#include
#include
#define
#define
#define
#define
#define
#define
using na

HANDLE c
COORD Cu
int pipe
int gapP
int pipe
char bir
'|',' ',
int bird
int score = 0;
void gotoxy(int x, int y)
{
CursorPosition.X = x;
CursorPosition.Y = y;
```

```
 -------------------------
 | Flappy Bird |
 -------------------------

1. Start Game
2. Instructions
3. Quit

Select option:
```


C:\Users\Asus\College study material\Second Year\Semester 3\Computer Graphics\Mini project\Code.exe
Score: 2
/--o\
|___ >
Code.cpp
#include
#define
using
HANDLE
COORD
int pi
int ga
int pi
char b
'|','_
int birdPos = 6;
int score = 0;
void gotoxy(int x, int y)
{
CursorPosition.X = x;
CursorPosition.Y = y;
Compiler
Resources
Compile Log
Debug
Find Results
Done parsing in 0.047 seconds


#include<iostream>
#include<conio.h>
#define
using
HANDLE
COORD
int pipePos
int gapPos
int pipeFl
char bird
int birdPos
int score
void gotoxy
{
CursorPosition.X = x;
CursorPosition.Y = y;
Game Over
Press any key to go back to menu.
Activate Windows

Chapter 4: Conclusion

4.1 CONCLUSION

Throughout this project, we aimed to develop a flappy bird game that allows users to interact with the game using a keyboard with ease.

The game has no graphics library but has a graphical interface for ease of interaction. The player has to go through all the pipes to win or obtain a high score. The concept is to create a flappy bird game with a graphical interface on the computer without using any graphics library.