



University of Dhaka  
Department of Computer Science and Engineering

**Project report:**

Fundamentals of Programming Lab(CSE-1211)

**Project Name:**

Flappy Bird

**Project Mentor:**

Md Mahmudur Rahman

**Team Members**

- 1) Sidratul Muntaher Meheraj (SK-57)
- 2) Md. Sifat Hossain (SH-17)

## **Table of Contents**

Chapter 1- Introduction .....	3
1.1 Game Overview .....	3
Chapter 2 - Game Objectives .....	3
Chapter 3 - Project Feature .....	4
3.1 Images .....	4
3.2 Audio .....	4
3.3 Keyboard Keys .....	5
3.4 Game Design.....	5
3.4.1 Game Rules.....	5
3.4.2 Game Play .....	12
3.5 Attempts to make the game more fun .....	12
Chapter 4 - Project Module .....	13
4.1 List of classes and their responsibilities .....	13
Chapter 5 - Team member responsibilities .....	13
Chapter 6 - Platform,Library & Tools .....	14
Chapter 7 - Limitations .....	14
Chapter 8 - Conclusion .....	14
Chapter 9 - Future Plan .....	14
Chapter 10 - Repositories .....	15
Chapter 11 - References .....	15

## **1. Introduction**

This report describes the process involved in making a game named “Flappy Bird”. This report discusses the game overview including its Objectives, features, limitations And also the future plan of this game.

### **1.1 Game Overview:**

In this project, we design and implement a “Flappy Bird” game adding some new features. Flappy Bird is a very popular mobile game on the Android platform, driving a lot of people crazy. In this game, the player can control the vertical or horizontal movement of the bird. As soon as the game begins, walls will keep appearing from the right side of the screen and moving leftwards.(so that it seems like the bird flying forward). The goal of this game is to control the bird, dodging and passing the incoming walls, and gaining hearts and coins as many as possible as the heart increases its life and coins increase its score. The game is endless until the bird’s life becomes zero.

## **2. Game Objectives**

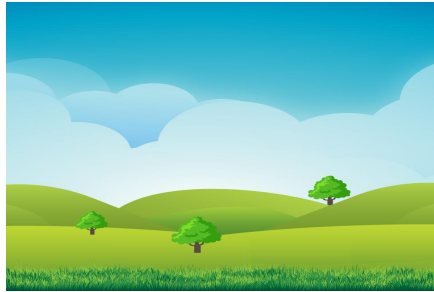
1. To implement a moving bird that can move both horizontally and vertically.
2. To add some objects like “hearts and coins” rather than just flying escaping the walls.
3. To design the game simple yet beautiful
4. To make a user friendly interface that will be pleasant to look at
5. To add animations to all sorts of game objects like birds, walls including UI transitions.

### 3. Project Feature

#### 3.1 Images :



**Coin**



**background**



**Heart**



**Bird**

#### 3.2 Audio:

The following table shows the list of audio clips with descriptions we have used to develop this game.

Name	Category	Description
Melodyloops	Background Music	Plays During the game
Arcade score interface	Coin gaining sound	Plays when the player gets coin
Arcade heart interface	Heart gaining sound	Plays when the player gets heart
Arcade hit interface	Hit sound	Plays when the player gets Collided with the wall and life becomes less than zero
Crazy-game-over	Game over sound	When the game is over It plays

### 3.3 Keyboard Keys:

Name	Description
SDLK_ESCAPE	Helps player to go back to menu board
SDLK_0	Turns off the music while playing
SDLK_1	Turns on the music while playing

### 3.4 Game Design

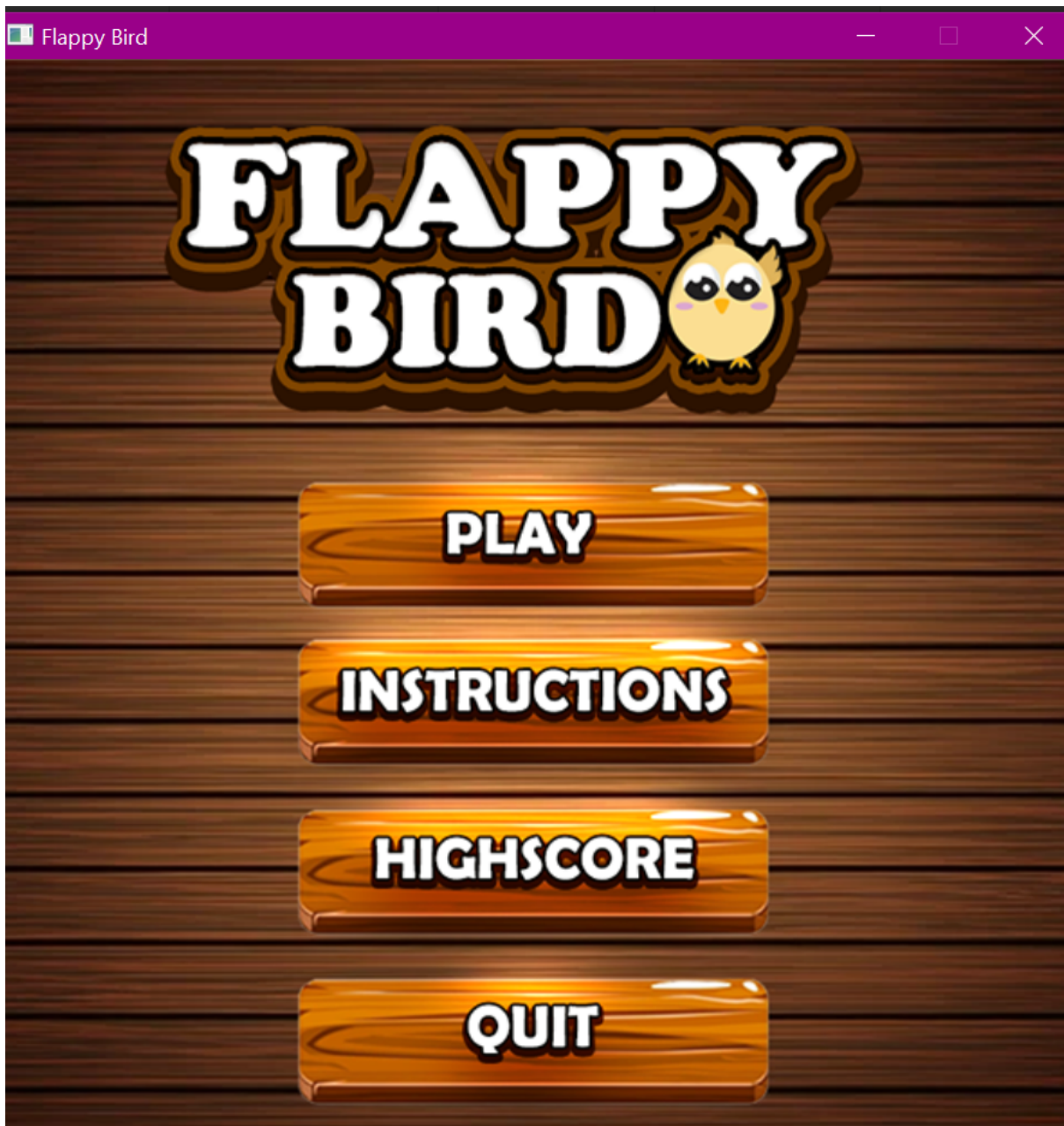
In this Game design part there is a vivid study about how to make the game functional by setting game rules, game mechanics, gameplay which is already briefly discussed in Chapter 1 but in this section it is discussed more clearly with some screenshots from the game.

#### 3.4.1 Game Rules

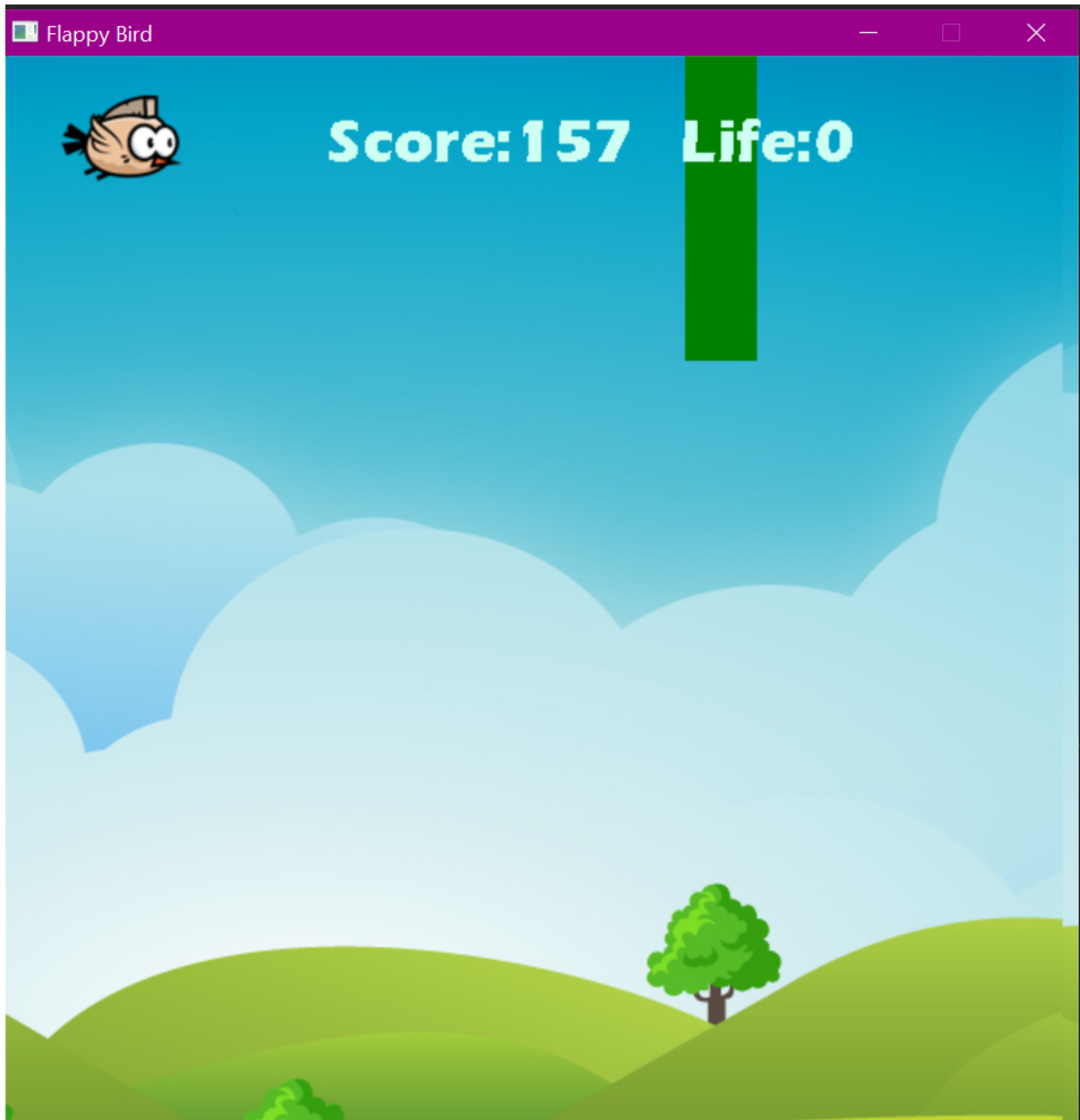
1. The game starts with a bird flying from the top left corner.
2. By using **arrow keys** we can move the bird any side we want. For instance, if we press the **up arrow** the bird will move up, and pressing the **down arrow** will move it down.
3. Similarly the **left and right arrow keys** will move the bird consecutively backward and forward.
4. In the **instruction section** a vivid description about how the game can be played is described.
5. The **highscore section** contains the record of the score of top 3 players.
- 6 . Also the **quit button** closes the window we created thus stops the game.

## Game Instructions

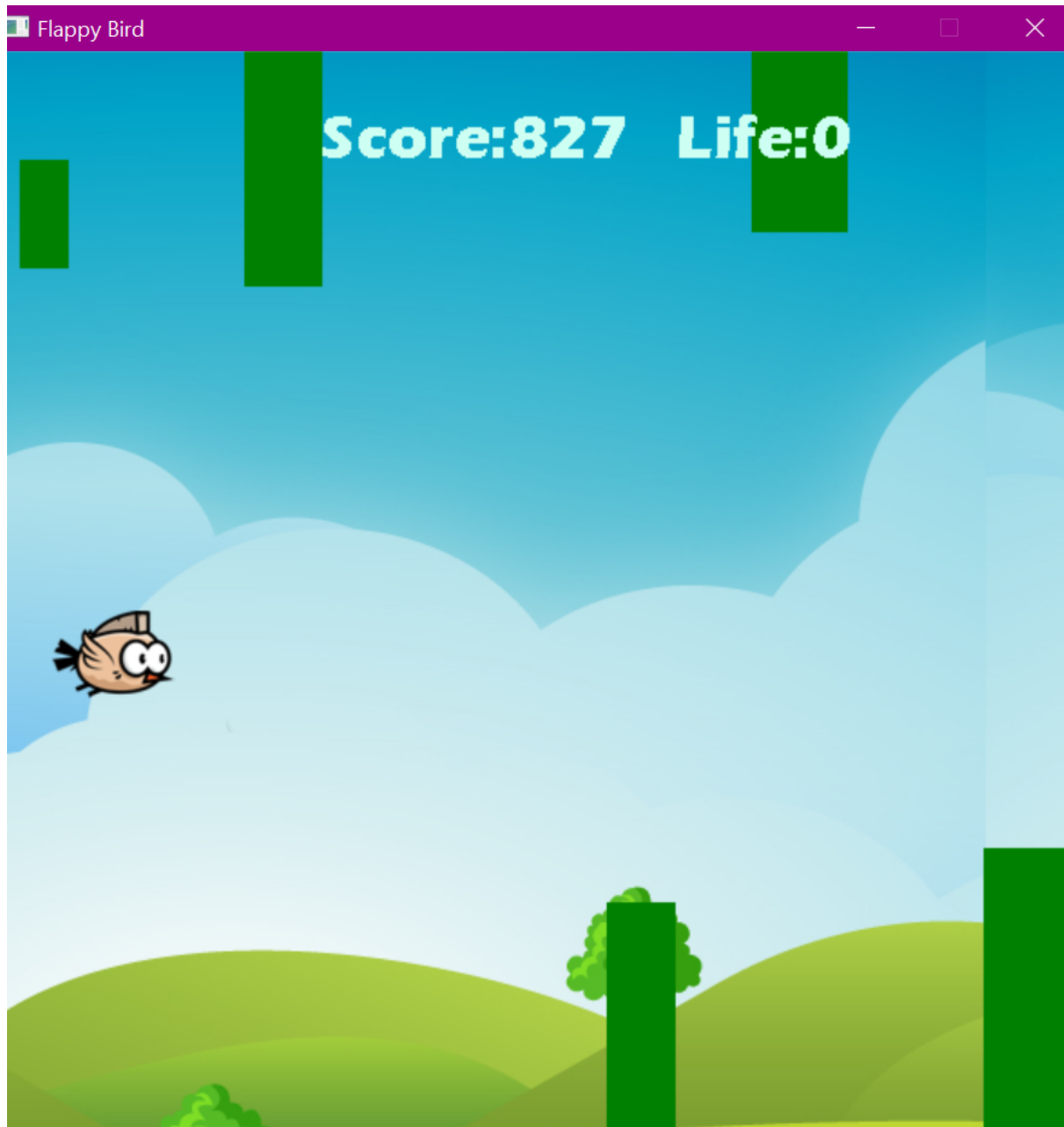
1. The game starts popping up its menu bar. In this menu bar it can be seen there are 4 options including play which initializes the game and the music.



2. The game starts with the birds flying from the top left corner .Initially “ score” and “life” both will be zero.

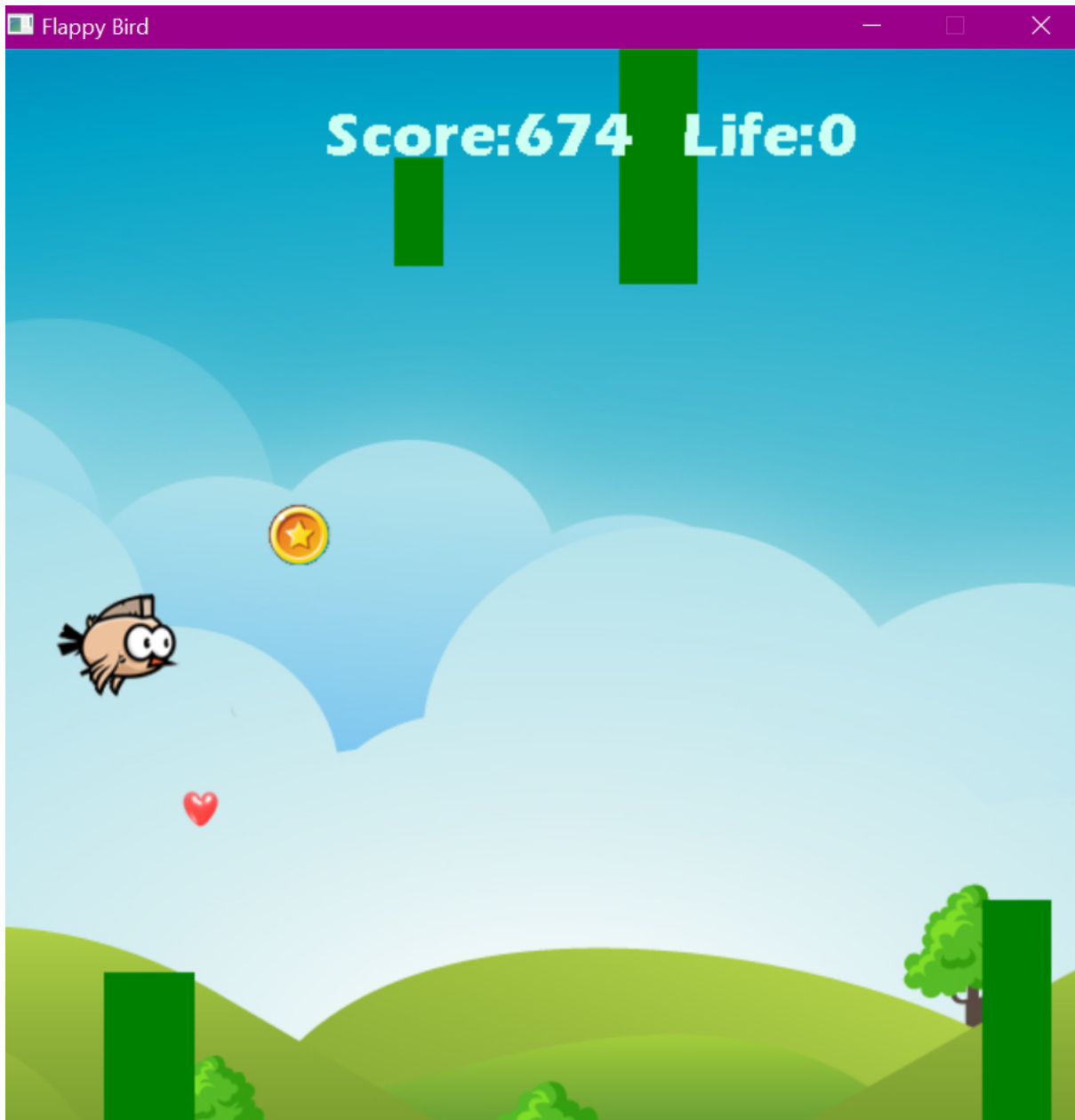


3. There will be pillar shaped obstacles at different heights. The speed of the screen  
With the moving of the bird.

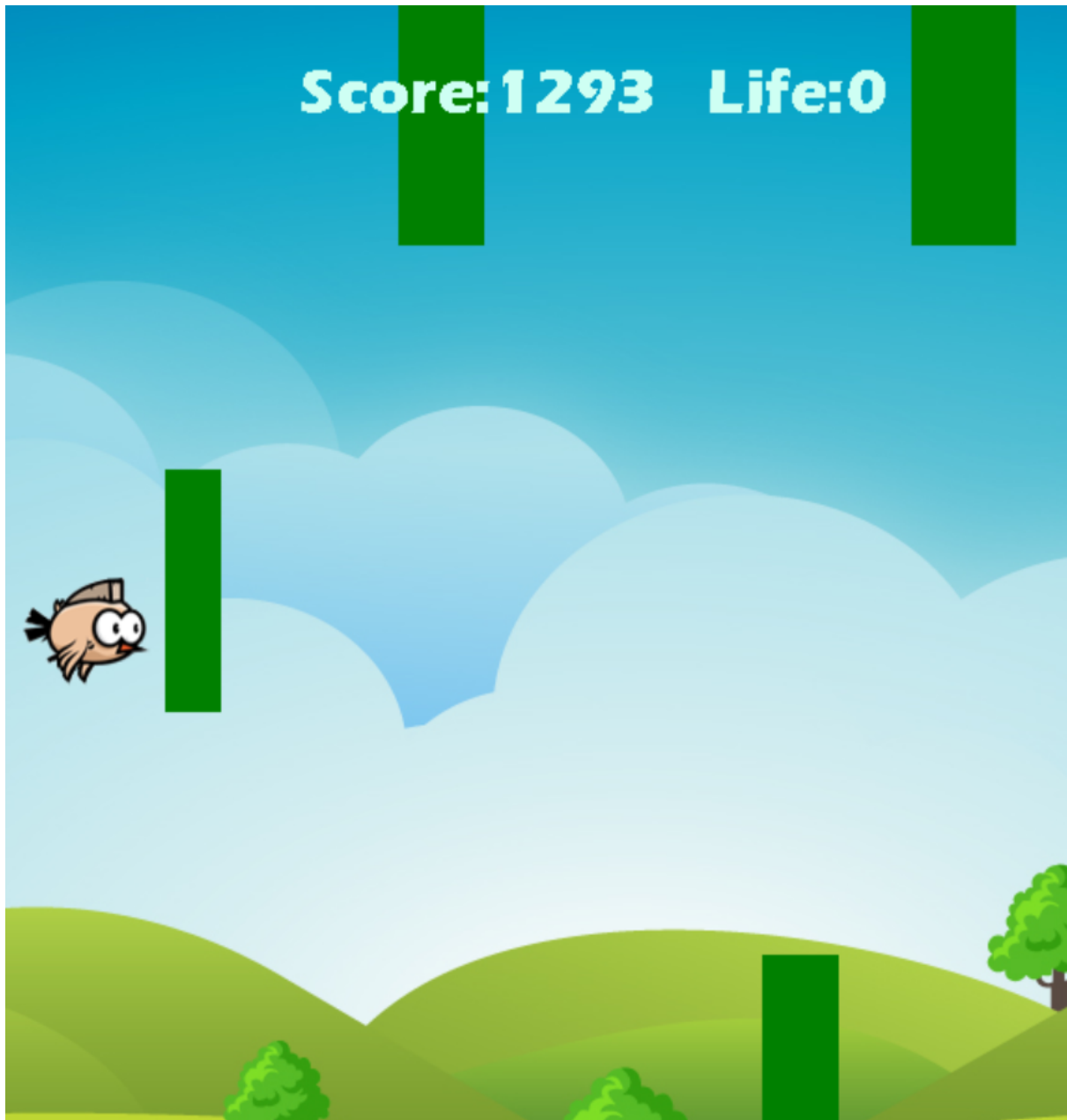




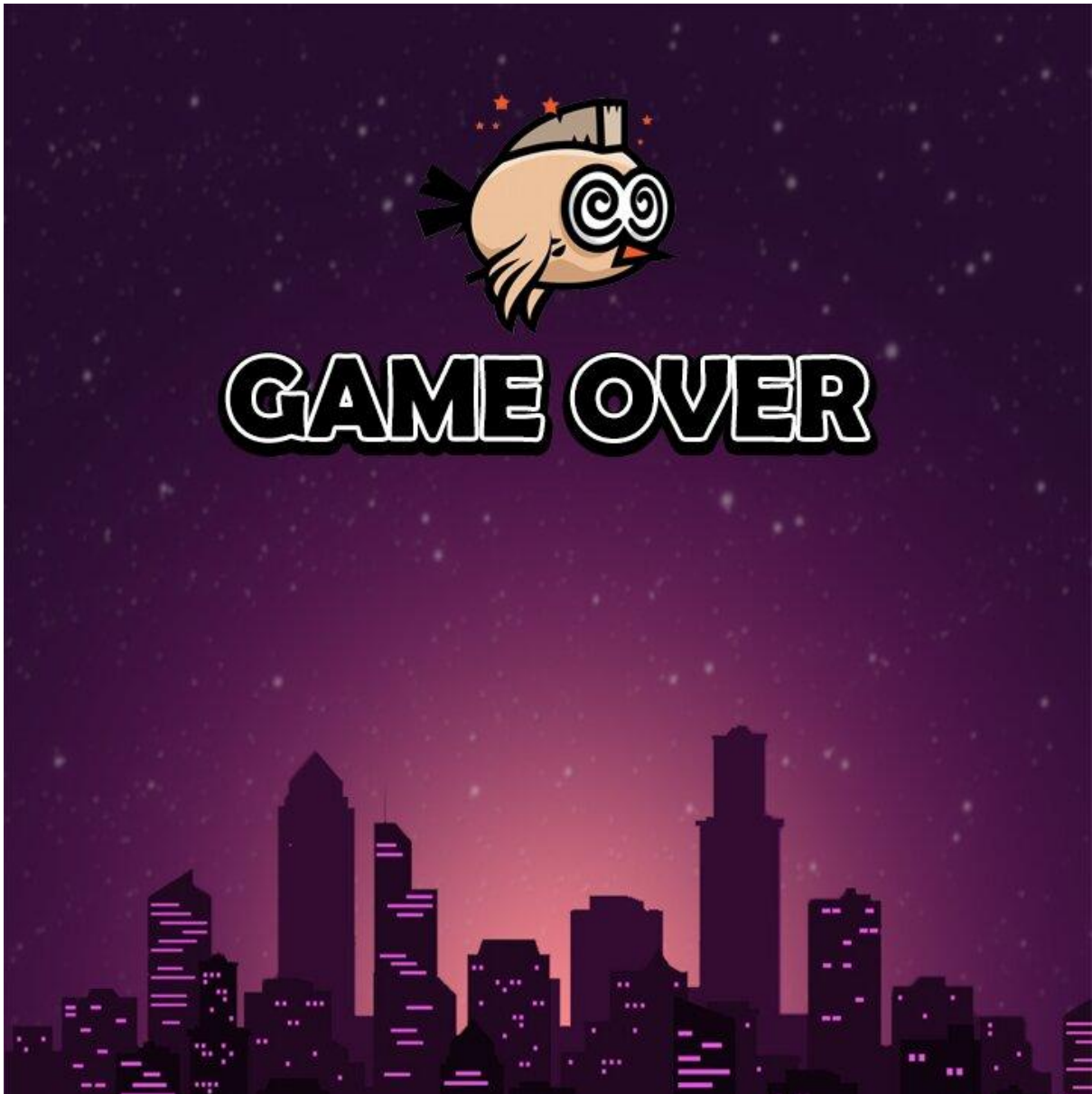
4. There will be coins and hearts. By obtaining a heart the bird can increase its Life. The score portion will be increasing always with the bird moving and also by getting coins. We have used sound at the time of getting coins.



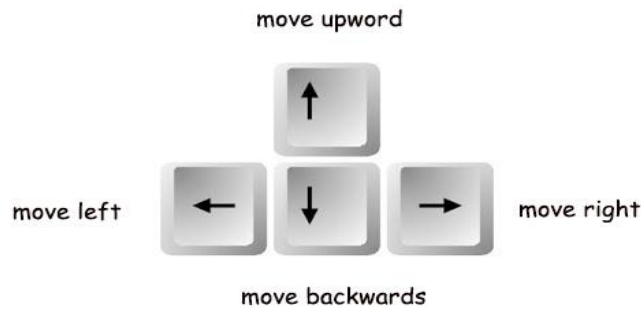
5. If the bird collides with the pillars ahead, it will decrease its life. If Life gets zero, the game is over.



6. And the game ends showing a picture written “Game Over”.



### 3.4.2 Game Play:



### 3.5 Attempts to make game more fun:

Although common mechanics can make the game functional, that's not enough for entertainment purposes, to entertain the users we need to pay attention to other parts of the game. With keeping that mindset we have included the following features in the game to make the more fun

**Background Music:** Instead of just using one background music we have added multiple music such as while the player is Playing we have used one sound and when it is game over We have used another one.

**Sound Effects:** Instead of using one sound clip we used here various clips to distinguish the heart, coin or collision.

## 4. Project Module

### 4.1 List of Classes and their responsibilities:

#### Texture.h :

All the user defined functions used in texture.cpp are declared here.

Number	Class Name	Responsibility
1	LTexture	Responsible for loading all the images that are used in the game.
2	Dot	Responsible for detecting collisions, moving the bird , controlling the background music and rendering Some objects.
3	LButton	Holds the constants for the button and controls them

#### Struct.h :

Number	Structure Name	Responsibility
1	A	Responsible for playing the background music and also various types of sound effects.
2	B	Holds the window for playing the game and the renderer to render various objects.
3	C	Responsible for creating all the SDL_Rect structures.

**Defs.h :** Defines all the constants for the game.

## 5. Team member responsibilities

#### Sidratul Muntaher :

1. Implemented texture.h, def.h, play.cpp, game.cpp, scoreboard.cpp.
2. Graphics.

#### Md. Sifat Hossain :

1. Implemented texture.h, init.cpp, struct.h, game.cpp, dot.cpp.

## 6. Platform, Library & Tools

1. Programming Language : C++.
2. Code Editor : Visual Studio Code.
3. Library : Simple DirectMedia Layer (SDL).
4. Image Editor : Adobe Photoshop CC 2017
5. Audio Editor : Audacity 2.1.3

These technologies mentioned above are completely free for students. There were no other additional tools required to make this game.

## 7. Limitations

- 1) There are no username features and as a result the player may get confused about his score.
- 2) Failed to add timing feature.
- 3) Failed to implement any “pause options” So players may feel some difficulties.

## 8 . Conclusion

After finishing this project we now have a clear idea about how structs are used. A little bit about Modular programming. A pretty much better idea how SDL library is used and its functions and events. Though we faced some problems during building up this project such as setting up SDL libraries and how to use them. Next time we are hopeful that It would be much better than this project.

This game is solely made for the user’s entertainment purpose and its principle objective was to make a game that is similar to Flappy Bird games but yet to have a modern vibe.

## 9. Future plan

This game is made for PC at the present. There are many scopes available for the improvement of this game including the additions of other functionalities.

Some of the plans for the future of this application could possibly be:

- 1) To make an IOS version.
- 2) To add timing feature
- 3) To change the game art entirely .
- 4) Adding game levels.
- 5) Add more different enemies in the game.

## 10. Repositories:

Github: <https://github.com/muntaher1025/Flappy-Bird>

Youtube link: <https://www.youtube.com/watch?v=tj9n3z5B4Ck>

## 11.References:

1. [https://lazyfoo.net/SDL\\_tutorials/](https://lazyfoo.net/SDL_tutorials/)
2. <https://www.parallelrealities.co.uk/tutorials/shooter/shooter6.php>
3. <https://www.pngegg.com/en/png-bbzmt>
4. [https://www.kindpng.com/imgv/TwJmoho\\_flying-bird-10-clip-arts-flappy-bird-png/](https://www.kindpng.com/imgv/TwJmoho_flying-bird-10-clip-arts-flappy-bird-png/)
5. <https://www.101soundboards.com/boards/10178-flappy-bird-sounds>