



University of Dhaka

Department of Computer Science and Engineering

Project Report:

Fundamentals of Programming Lab (CSE-1211)

Project Name:

DX-BALL

Team Members

1. Syed Mumtahin Mahmud, Roll: 50
2. Nazira Jesmin Lina, Roll 55

Introduction

Dx ball is a single player game. The main goal is breaking all the bricks without losing the ball. As it is a common Breakout-style game, the object is to clear the screen of all bricks in order to advance to the next level. The player controls a paddle at the bottom of the screen by using the mouse/keyboard, keeping one ball in play by bouncing it into a field of bricks to clear them. Some bricks will take multiple hits before they clear, while other bricks will appear to be unbreakable. As bricks are being cleared, random Power-Ups will occasionally release upon impact. The player may choose to catch these with the paddle, or avoid them, depending on the specific Power-Up and its effect. For instance, some Power-Ups may speed up game progress by enhancing the paddle with guns. While other Power-Ups may increase the difficulty by speeding up the ball or shrinking the paddle. If the player misses the ball in play, a paddle will be lost. The game will end once all spare paddles have been lost, or after the selected level has been completed. The game keeps the scores of a player, if a player can score a point which is in the range of maximum scores of all time.

Objective

Our main objective is to implement a simple graphics game project by using C/C++ language and SDL (Simple DirectMedia Layer) library in the practical field. However, we wanted to expand our knowledge of C/C++ for developing and applying it.

Project Features

The project DX-BALL's features can be divided into two sections. One is UI(User Interface) features and the other one is Game-play features. User interface (UI) features are the series of screens, pages, and visual elements like buttons and icons that enable a person to interact with the game. On the other hand game-play features are those which can be used or achieved or visualized while playing the game. Many features are added in this game to improve the user experience and make the game more flexible and interesting. The features are:

- User Interface Features:

1. Main Menu Features:

The user will experience this page after entering the game and also after completing the game. In this page the player can see a page with the icons named New Game, High Score, Options, Help and Exit. In the lower corner of the page images of all kinds of power-up features are shown alongside their names. If the “Music” option is on from the “Option” menu, then there will be music. From this page the player can choose to start a new game, see the previous high scores, go to the option menu for controlling the sounds of the game, choose the help section for instructions and control features and to close the game player can choose the exit menu from the main menu page. The icons of the main menu page are described below:

1.1 New Game:

This option will take a player to the level-1 of the game. All the previous game data will be lost after that, and the player has to start again. To start a new game player has to select the “New game” icon.

1.2 High Score:

This option will show the top 20 highest scores of previous games in descending order. First page will show the first top 10 highest score and after clicking the “Next” button, the second page will appear and show the other 10 highest scores.



Figure 1: Main Menu Page



Figure 2: High Score Page

1.3 *Option:*

This icon will take the player to the page from where the player will be able to turn ON/OFF the game music and also turn ON/OFF the sound (In game effect sounds. Such as: paddle and bar collision, ball and bricks collision).

1.4 *Help:*

After choosing this option the player will see a page with three icons named Instructions, Controls and Power-up. “Instruction” section will show the details about the game mechanics and game module. “Controls” section will show the player how to control the ball, paddle and power-up’s features. On the other hand, the “Power up” section will show the details about all the power ups in detail.

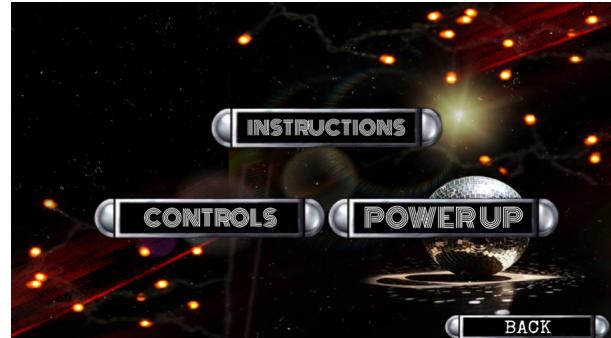


Figure 3: Help Page

1.5 *Exit:*

After choosing the option the player will be able to see a confirmation message with the options “YES” and “NO”. Choosing “YES” will close the game and “NO” will take the player back to the main menu page.

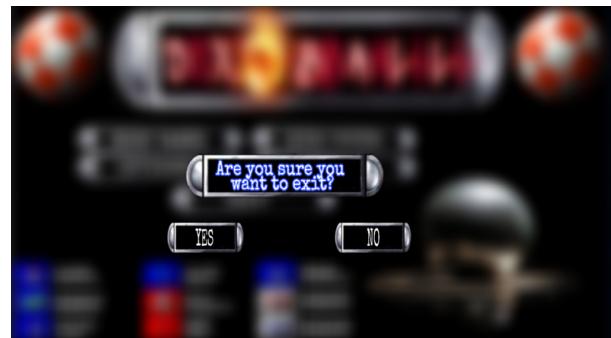


Figure 4: Exit Page

2. Pause Menu Features:

This page appears when a player pauses a game while playing the game. This page appears by pressing the “ESC” button on the keyboard while playing the game. This page is almost similar to the Main Menu Page. The common icon between the pause menu page and the main menu page works the same. The exception is the visual representation of the page and the pause menu page contains the “Continue” icon and doesn’t contain the “High Score” icon. By selecting the “Continue” icon player can get back to the game and play the game. This same thing happens when a player presses the button “ESC”. In the lower corner of the page pictures of all kinds of power-up features are shown alongside their names. If the “Music” option is on from the “Option” menu, then there will be music. From this page the player can choose to start a new game, go to the option menu for controlling the sounds of the game, choose the help section for instructions and control features and to close the game player can choose the exit menu from the pause menu page.



Figure 5: Pause Menu Page

3. Game Over Features:

This page will appear after completing the game or losing all life. It will ask for the player’s name. Player had to type his/her name in 10 characters. After typing the player name, the player has to press the “Enter” button. After that a new page will appear showing the player’s score along with the player’s name. Then it takes the player to the main menu page.



Figure 6: Game Over Page

- ***Game-Play Features:***

1. ***Power-Up Features:***

There are nine types of power ups that can be achieved in every level randomly. Players can achieve these power up(s) in every level. Each time after achieving a power up, the player will get 5 points . Details about the power-ups are also available in the "Help" menu, which can be accessed from both the main menu page and pause menu page. Every power up has different types of ability or effect in the game play. Details about the power ups are here:

1.1 ***Laser Paddle:***

This power up will give the ability to fire from the two corner sides of the paddle.If a player achieves this power-up once, he/she will be able to fire 25 times(50 lasers). And each fire has the ability to destroy a single brick.

1.2 ***Grab Paddle:***

This power up will give the ability to grab the ball whenever it touches the paddle and after that the player can move the paddle with the ball and shoot the ball again from a suitable position.

1.3 ***Extra Life:***

This power up will give an extra life (Paddle) .

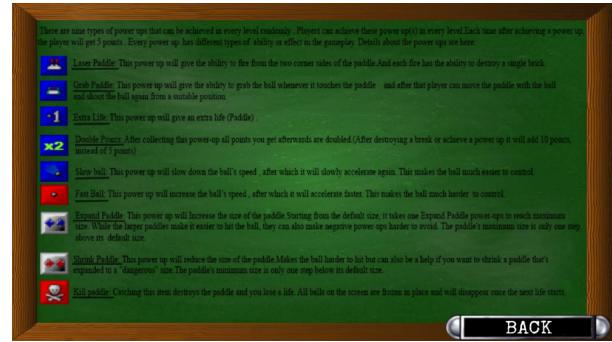


Figure 7: Power Up Information Page

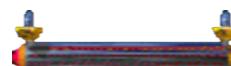


Figure 8: Laser Paddle

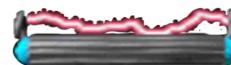


Figure 9: Grab Paddle

1.4 Double Points:

After collecting this power-up all points a player get afterwards are doubled.(After destroying a break or achieve a power up it will add 10 points, instead of 5 points)

1.5 Slow ball:

This power up will slow down the ball's speed , after which it will slowly accelerate again. This makes the ball much easier to control.

1.6 Fast Ball:

This power up will increase the ball's speed , after which it will accelerate faster. This makes the ball much harder to control.

1.7 Expand Paddle:

This power up will Increase the size of the paddle.Starting from the default size, it takes one Expand Paddle power-ups to reach maximum size. While the larger paddles make it easier to hit the ball, they can also make negative power-ups harder to avoid. The paddle's maximum size is only one step above its default size.

1.8 Shrink Paddle:

This power up will reduce the size of the paddle.Makes the ball harder to hit but can also be a help if you want to shrink a paddle that's expanded to a "dangerous" size.The paddle's minimum size is only one step below its default size.

1.9 Kill paddle:

Catching this item destroys all the paddle and the player looses all the lives. All balls on the screen are frozen in place and the game over page will appear with the current score of the player.

2. Multiple Levels:

DX-BALL has a total of three different levels. The difficulty of those increases proportionally. Details about the game's levels are given below:

2.1 Level 1:

This level contains only single hit bricks. Various types of power up will appear in this level randomly. This level's difficulty is much easier than other levels. As this is the first level of the game we tried to make it easier for the users.



Figure 10: Power Up Information Page

2.2 Level 2:

This level contains single hit bricks and multi hit bricks. Various types of power up will appear in this level randomly too. This level's difficulty is harder than level 1. As this is the second level of the game we tried to increase the difficulty for the users.

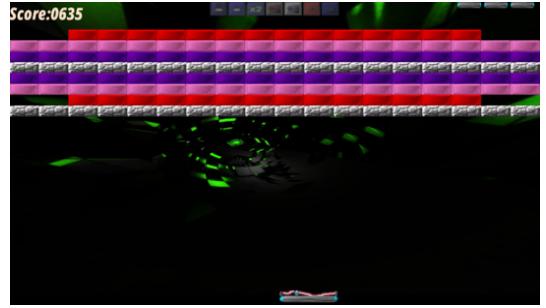


Figure 11: Power Up Information Page

2.3 level 3:

This level is the last level of the game. This level contains a combination of single hit bricks , multi hit bricks and invisible bricks. Various types of power up will appear in this level randomly too. This level's difficulty is harder than any other level of the game because of the invisible bricks as the players won't be able to know where the invisible bricks are. When a player hits an invisible brick the bricks will appear and the ball will move to the opposite. So, the player will have to face more difficulties at this level.



Figure 12: Power Up Information Page

3. Special Bricks:

Some special bricks are used in the game to increase the difficulty of level-2 and level-3 which will increase the difficulty of those level.

Those are:

3.1 Multi Hit Bricks:

These bricks will appear in the level-2 and level-3 and takes 2 hits to destroy.



3.2 Invisible Bricks:

These bricks will appear in the level-3. When the ball hits those bricks for the first time, then those bricks will be visible.

Figure 13: Multi Hit and Invisible Bricks

4. Sound Effects:

Different kinds of sound effects are included in this game. These sound effects will appear whenever the ball hits the paddle, ball or fire hits the bricks, life is lost, power-up is achieved, start firing from the laser paddle, starting a new level. If a player wants to turn off the sound effects then it can be done from "Option" menu and turning off the "Sound" icon from the main menu page or the pause menu page.

5. Score and Life Indicator:

Score and Life Indicator: After starting the game the player will be able to see the score and life indicator separately at the top left corner and the top right corner of the page (Figure 8). Score indicator is made with the help of SDL TTF (True Type Font). Score indicator will show the achieved score. On the other hand, a life indicator will show how many lives(paddles) are left. These indicators will update real time.

6. Power-up Indicator:

After starting the game the player will be able to see the power-up indicator in the top middle corner of the page(Figure 8). This indicator contains the blurred image of the power-ups. After achieving a power-up, the blurred image of the specific power-up will become bright. This indicator will help the player remember what special ability he/she has in the game. If a player loses an ability then the bright image will become blurry.



Figure 14: Power-Up Indicator, Score-Life Indicator

Project Module:

The total game code is divided into multiple header files and source code files. In this game we have implemented 10 custom header files. Details about those custom header files are given below:

1. #include "pre_declare.h"

This header file contains all pre-processors which were used in the code. It also contains the renderer, the renderer flag, window and game over flags. It also contains some important global variables with extern and all the textures of the level background.

2. #include "sound.h"

This header file contains all the sound related variables and functions. It also contains some sound related boolean variables which are used globally in the code. Details of the functions:

2.1 `void music_load();`

This function is used for loading all the music and sound effects of the game. Such as main menu music, bar paddle collision sound effect, ball bricks collision sound effect, game over music, expand paddle music etc. All the power up related music is loaded in this function.

3. `#include "struct.h"`

This header file is used for declaring all the struct(s) that we have used in the code. In the whole code we have used many kind of struct(s). These strut(s) are used for many different purposes such as brick's coordinate storing, power up information storing, paddle rendering etc purposes. These struct's variables or struct's arrays are also declared in that custom header file.

4. `#include "utils.h"`

This header file contains some global variable related to game,brick's textures and some function related to initialize the game and initialize the level. Details about those functions:

4.1 `int INITIALIZE();`

This function is used for initialize the SDL libraries. This user defined function initialize SDL, SDL ttf, SDL mixer etc libraries. This function don't take anything as a parameter but returns an integer.If this initialization process is completed successfully then it will return 1 or if the initialization process is not completed properly then it will return 0. This function also create and initialize the window and renderer.

```
4.2 SDL_Texture *DISPLAYING_SCORE(char ch[ ], int colour,  
                                int make_null);
```

This function is for printing the score. This function takes a string a variable named "colour" and a flag as parameters and returns a texture. This function actually takes a string and make it a image using SDL TTF with specific colour and font which needed to be declare.

```
4.3 int levelup_bricks_initialization(int level);
```

This function is for every level's bricks distribution. In this function we have generated all the brick's coordinates using FILES. This function takes a variable which is the current level as a parameter and returns an integer which contains the total number of bricks in that specific level. In this function bricks distribution is made by using FILES.

```
4.4 void name_write();
```

This function is for writing the player name after the end of the game. In this function we have used SDL TTF to write the player's name. After writing the name, the player will be able to see his/her name alongside his/her score.

```
4.5 void mainmenu_level_renderer();
```

This function is for rendering bricks for each level. At the starting of a new level this function is always called.

```
4.6 void ball_music_and_powerup_load();
```

This function loads the ball's image, some music of the game and all the power-up related images. The images we have used in the power-up indicator are also being loaded here.

5. #include "mainmenu_resources"

This custom header file contains all the UI related variables, texture and functions. More specifically, this header file has SDL Rect, Textures and functions which are related to the main menu page and pause menu page. Details about the functions:

5.1 void mainmenu_load();

This function loads all the main menu related images and creates textures from surfaces. In this function we have generated the coordinates of those images. This function also loads some pause menu related images and creates textures from surfaces too.

5.2 int mainmenu_render();

This function is for rendering the main menu page. This function doesn't take any parameters but returns an integer which is 0 or 1. If someone closes the game from the main menu page it returns 0 else it returns 1. This function also takes a player to a different menu such as high score menu, help menu, option menu , end menu etc. Mainly this function visualizes the whole main menu page. This page can be controlled with both mouse and keyboard.

5.3 void pause_menu_renderer();

This function is for rendering the pause menu page. This function also takes a player to a different menu such as help menu, option menu , end menu etc. Mainly this function visualizes the whole pause menu page. This page can be controlled with both mouse and keyboard.

```
5.4 void option_render();
```

This function renders the “Option” menu. This function controls turning on or turning off the music. Also game play sound effects turning on or turning off are controlled by this function depending on the user’s input.

```
5.5 void highscore_render();
```

This function renders the “High Score” menu which shows the top 20 high scores. This function takes the top 20 scores from a file and renders them in two pages.

```
5.6 void help_render();
```

This function renders the “Help” menu which shows another three pages named Instruction, Controls and Power-up. These sub pages are also being rendered from this function.

```
6. #include "gameplay.h"
```

This custom header file contains all the Game play related variables, textures and functions. These functions are being used for loading game play data and rendering them. Details about the functions:

```
6.1 void score_and_life_print(int score,int life);
```

This function is for printing the current score and lives that a player has. Actually this function is for creating and rendering the score indicator and life indicator in time of game play.

```
6.2 void bricks_and_bar_load();
```

This function is for loading the paddles and bricks. There are 4 types of paddles and many types of bricks are used in the game. All of those paddles and bricks are being loaded from here and we are creating their textures from their surfaces.

```
6.3 void firerender(int fbar);
```

This function is for rendering the laser. After achieving the laser paddle this function renders the laser from the corner sides of the bar depending on the player's input.

```
6.4 int bar_and_bricks_render(int cnt,int l);
```

This function is for rendering the paddles and the bricks according to their coordinates. The animation of the paddle has been done in this function. This function takes two integers as parameters and returns an integer value which is the total count of bricks. Mainly, this function visualises the paddles and bricks.

```
7. #include "game_physics.h"
```

This header file contains variables and functions related to the ball and paddle movement of the game. This part is the main part of this DX-BALL project. Details about the functions:

```
7.1 void game_physics();
```

This function controls the ball's and power-up's movement. This function contains the direction and vectors of ball's and power-up's. Also the ball's and grab paddle's combined movement is controlled by this function.

Initially, we assume the angle to be 30° for the ball's initial movement. Here Δx is the horizontal change and Δy is the vertical change of the ball's coordinates. So the change of the ball's coordinate is, ,

$$\theta_0 = 30, v = BALL_SPEED,$$

$$\Delta x = v \cdot \sin\left(\frac{\pi\theta_0}{180}\right)$$

$$\Delta y = v \cdot \cos\left(\frac{\pi\theta_0}{180}\right)$$

.

7.2 void collision();

This function controls the collision between the ball and bricks. After collision the bricks will disappear. This function also controls the collision between laser and bricks. Power up start ups are also being done from here and the sound effect of collision also starts from here.

7.3 void ball_fall_paddle_collision();

This function controls the fall effect of the ball and also controls the paddle and ball collision. Bar and paddle collision sound effect and fall musics are also being controlled from here. The main part of the function is the ball's deflection in after hitting the paddle.

Here, we assume the angle to be θ for the ball's deflection. Here Δx is the horizontal change and Δy is the vertical change of the ball's coordinates,"ball.x" and "ball.y" is the x and y coordinate of ball, "paddle.x" and "paddle.y" is the x and y coordinate of paddle. So the change of the ball's coordinate is,

$$v = BALL_SPEED, w = Paddle's_width$$

$$\theta = \frac{(w - [ball.x - paddle.x])}{w} \cdot 180;$$

$$\Delta x = v \cdot \cos\left(\frac{\pi\theta}{180}\right)$$

$$\Delta y = v \cdot \sin\left(\frac{\pi\theta}{180}\right)$$

8. #include "powerup.h"

This custom header file contains power up related variables and functions. Details about the functions:

8.1 void powerup_achieve();

This function is for selecting which power up a player has achieved. This power up is generated randomly.

8.2 void powerup_renderer(int p, int type);

This function is for rendering the specific power up. This function also contains the movement, velocity and vectors of the power up. Mainly, this function visualizes the achieved power up icon and also controls the movement of the power up icon.

9. #include "end.h"

This custom header file contains some functions which are used for resetting the game or freeing up the space before the game. Details about the functions:

9.1 void reset_game(int flag);

This function takes an integer variable as a flag. Basically this function is used for resetting the game when a player loses a life, is promoted to a new level or starts a new game.

```
9.2 void quit();  
9.3 void font_closing();  
9.4 void level_destroy();
```

These functions are used for freeing up the space by destroying the window , destroying the textures, closing the fonts and this function also closes the SDL, SDL TTF, SDL IMAGE , SDL MIXER etc libraries.

10. #include "header.h"

This custom header file doesn't contain any variables or functions. Instead, it contains all the library header files and all the custom header files. As this header file contains all the header files we have included only this specific header in all custom header files and source code files.

Team Member Responsibilities:

Team Member 1:

Syed Mumtahin Mahmud, Roll: 50

1.1 Paddle Animation:

General paddle, grab paddle, laser paddle and combined grab and laser paddle are included in this module. All the paddles are rotating on their own horizontal axis. Grab paddle will catch balls that fall on it in an electric force field that can be released with the left mouse button.

1.2 Paddle Movement::

Paddle can be controlled by moving the mouse and keyboard. If the player moves the mouse left/right, the paddle moves left/right.

1.3 Ball Movement:

When the ball bounces on a paddle or collides with the upper corner side of the screen/right corner side of the screen / left corner side of the screen,

the ball moves according to its incoming angle. The angle at which the ball will bounce off is determined by the side of the paddle it lands on. The further left or right the paddle is hit, the more the ball will be angled in that direction precisely. Ball speed's change is dependent on the power up.

1.4 Game level 1 and 2's Bricks Distribution:

The game contains three levels and different assemblies of bricks. The coordinates of the bricks are generated and then the bricks images are rendered in the selected coordinates.

1.5 Ball and Bricks Collision:

When the ball collides with brick, the brick will vanish and the ball starts to move to the opposite direction with the geometric angle which depends on its incoming angle. A sound will appear when the ball hits a brick.

1.6 All the Power-up Related Works:

Power-ups will appear when the ball hits some specific bricks. These bricks are selected randomly. By default when a player achieves a power up, it starts to move towards up with the speed and the angle of the ball. When a player catches the power up icon with the paddle, he/she will get the ability of the power up. There are a total nine types of power-ups. Some of them have positive impacts and some of them have negative impacts in the game.

1.7 Laser Paddle (fire) Related Works:

When the player achieves laser paddle, he can fire from the two corner sides of the paddle by pressing the left mouse button or the space button. If a laser hits a brick, the bricks will be destroyed. To destroy the multi hit bricks or the invisible bricks more than one laser hit is needed. There will be a collision sound if a player hits a brick with a laser. After achieving this laser paddle, a player will be able to fire the laser 25 times (50 lasers).

1.8 Grab Paddle Related Works:

When the player achieves grab paddle he/she will be able to grab the ball whenever it touches the paddle. After that the player can move the paddle with the ball and shoot the ball from a suitable position.

1.9 Combined Laser and Grab Paddle Related Works:

When the player achieves grab paddle after achieving the laser paddle or vice-versa, the player will get the ability of both grab paddle and laser paddle.

1.10 Power-up Indicator:

From the beginning of the game, the power-up's blurred images will be shown at the top of the screen. When the player achieves a power-up, the specific power-up image will become bright from the blurriness at the top of the screen.

Team Member 2:

Nazira Jesmin Lina, Roll: 55

2.1 Main Menu Page Design:

All the main menu page related works such as new game, high-score, options, help, exit button and rendering the power up images in the front page bottom of the screen. If a player selects a button, it will take him/her to the corresponding page.

2.2 Score Print:

When the ball or laser collides with the brick or power up is achieved, the player gets 5 points. The score is shown in the top left corner of the screen.

2.3 Life Print:

The remaining paddles (lives) in the game are displayed in the top right corner of the screen.

2.4 Paddle Movement with Keyboard:

Paddle can be moved by right and left keys of the keyboard.

2.5 Game Level 3's Bricks Distribution:

In game level 3, the text “THE END” is created by various colours of bricks. First the coordinates of the bricks are generated and then the bricks images are rendered in the selected coordinates.

2.6 All the Sound Related Works:

In this game there are two types of sound. They are UI music and game sound. Those sound effects will appear at different times. Those can be played or turned off by clicking on the icon which is included in the “Option” part in the main menu page or the pause menu page.

2.7 High Score Orientation:

In this section the scores are saved in a file sorted in descending order. When a new score is achieved , it will be compared with the previous data. If the score is greater than the lowest score, then all the scores will be sorted again and those scores will be saved in a file for future uses.

2.8 Game's Help and Instructions and Exit Section:

In the game's help section controls, instruction and power-up button are included. By clicking those button player will be able to know about the game's details. If a player wants to exit he/she has to click the exit button and then a confirmation message will appear, if the player confirms then it will close or it will take the player back to the main menu.

2.9 Game over Page Presentation:

When a player loses all the lives or completes all the levels, a game over page will be displayed with the text “Write your name”. After writing the players name, the score will be shown with the written name. Then the screen will back to the main menu.

Platform, Library and Tools:

- Platform: Linux.
- Language: C/C++.
- Library: SDL (Simple DirectMedia Layer) library.
- Tools : Adobe Photoshop, Pics Art Photo Editor, Mp3 Cutter, Online Music Converter, Online Image Resizer.
- LaTex editor: Overleaf
- Video editor: OBS Studio, Kdenlive

Limitations:

There are some limitations that a player may face while playing the game.

- The project DX-Ball is made for Linux platforms.
- The name of the player which is kept for high scores record will be confined into 10 characters.
- There will no longer be the existence of previous game records after a new starting except high scores.

Conclusions:

In this project we have gained lots of experience and had to face many obstacles too. Making a game is completely a new and practical experience to us. So, We had to learn things from the beginning. By this work we have introduced ourselves with the SDL (Simple DirectMedia Layer) library. Now, We know how a model is constructed and how it is animated. Our thinking and imagination capability have grown. We have developed our communication skills by co-operating between group members. Actually it was a whole new experience for us.

The obstacle we have faced is that making a game is completely a new experience to us and it is different from the programming that we used to. We had to learn things from the beginning which were fully new to us. We have learnt things by video tutorials, internet and learning materials. Overall, it's not an easy task. It's a matter of hard work, time and patience. It can be said that making games is a very sensible work to do because through the game we try to connect the game environment with the real world. Creating a user friendly model is a difficult task because we had to work with each and every point of the model.

At first we thought making a DX-BALL game would be easy. But later we realized that it's not that easy. Because all the work had to be done by ourselves. Every single thing needed to be implemented by us. From the ball's movement to the score count, we had to do it by ourselves. So, from the UI to game play we had to do a lot of work. So, that experience or realisation was far beyond our expectations. Overall, it was a great and new experience and we are happy that we have implemented our idea and teacher's suggestion in our own codes.

Future plan:

- Improve the graphical representation of the game.
- Game level extension.
- Introduce and implement new game features.
- Take user feedback and improve features according to their opinion.
- Introduce new environment and scenes.
- Make the game suitable for Windows and Mac OS platform.

Repositories:

GitHub Repository: <https://github.com/Dip-to/DX-BALL.git>

YouTube Video: <https://youtu.be/Jks3s1dX2RU>

References:

SDL Wiki: <https://wiki.libsdl.org/Tutorials>

Lazy Foo: https://lazyfoo.net/SDL_tutorials/

Steam Community: <https://steamcommunity.com/sharedfiles/dx-ball>

Lib SDL(TTF section): https://www.libsdl.org/projects/SDL_ttf/

Lib SDL(Mixer section): https://www.libsdl.org/projects/SDL_mixer/

GeeksforGeeks: <https://www.geeksforgeeks.org/write-header-file-c/>

Mixkit: <https://mixkit.co/free-sound-effects/game/>

Typesetting(Overleaf): <https://www.overleaf.com/read/jftsqwsdkmzj>