University of Dhaka

Department of Computer Science & Engineering

Project Report:

Course Title: Fundamentals of Programming Lab

Course Code:CSE-1211

Project Name: DX-BALL

Submitted BY:

1. Joty Saha (Roll: 51)
2.Asef Sami Chowdhury (Roll: 53)
3. Joty Saha (Roll: 45)

Introduction:

Dx ball is a single player game.The main goal is breaking all the bricks without losing the ball**.** The goal of the game is to remove all of the bricks from the screen in order to move on to the next board, which is a common concept in Breakout clones. Using the mouse or keyboard, the player manoeuvres a paddle at the bottom of the screen to keep one or more balls by deflecting them into a field of bricks while avoiding missing the only ball that is currently in play. Some bricks might have a solid appearance, require multiple hits to remove them, be invisible, or be explosive. A random power-up will occasionally be released onto the screen as bricks are cleared.

Depending on whether they will have a positive, neutral, or negative effect, the player can decide whether to catch them with the paddle or avoid them.Positive power-ups typically make things harder.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 goal is to use the SDL (Simple DirectMedia Layer) library and C/C++ language to implement a straightforward graphics game project in the real world. However, in order to create and use it, we wanted to increase our C/C++ knowledge.

Project Features : The project DX-BALL can be divided into two parts

>User Interface
>Game-play features.

• User Interface Features:

1. Main Menu Features: It is the entering page of the game . On this page the player can see some option like

- New game
- High score
- Option
- Help and
- Exit

1.1 New Game: A player can access level 1 of the game by selecting this option. After that, the player must restart with all of their previous game data lost. The player must choose the "New game" icon to begin a new game.

1.2 High Score: This option will show the top 20 highest scores of previous games in descending order.

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 clicing the help button it will help the player to know the game feature

1.5 Exit: After clicing the exit button it will show two option either the player want to exit yes or no.

• Game-Play Features:

1. Power-Up Features: There are nine different kinds of power ups that can be acquired at random in every level. These powerups are accessible to players in every level. The player receives 5 points each time they obtain a power up. The "Help" menu, which can be accessed from both the main menu page and the pause menu page, also contains information about the power-ups.

1.1 Laser Paddle: This power-up will grant the ability to shoot from the paddle's two corner sides. A player can fire 25 times (50 lasers) if they obtain this power-up once. A single brick can be destroyed by each fire.

1.2 Grab Paddle: This power up will enable the player to grab the ball whenever it touches the paddle, move the paddle with the ball still on it, and then shoot the ball again from an appropriate position.

1.3 Extra Life: This power up will give an extra life (Paddle) .

1.4 Double Points: All points a player earns after obtaining this power-up are doubled. (When a break is destroyed or a power-up is obtained, it will add 10 points rather than 5 points.)

1.5 Slow ball: The ball's speed will decrease as a result of this power-up before gradually increasing again. The ball is now much simpler to control as a result.

1.6 Fast Ball: The ball's speed will increase as a result of this power up, and it will then accelerate more quickly. The ball is now much trickier to handle as a result.

1.7 Expand Paddle: The paddle's size will increase thanks to this power-up. It takes one Expand Paddle power-up to increase the paddle's

size from the default size to its maximum size. The bigger paddles make it easier to hit the ball, but they can also make it more difficult to avoid negative power-ups. Only one step separates the paddle's default size from its maximum size.

1.8 Shrink Paddle: The size of the paddle will be decreased by this power-up. makes the ball more difficult to strike, but it can also be useful for shrinking a paddle that has grown to a "dangerous" size. Only one step separates the paddle's minimum size from its default size.

1.9 Kill paddle: When this item is caught, all the paddles are destroyed, and the player loses all of their lives. The game over page will appear with the player's current score after all balls on the screen have frozen in place.

2. Sound Effects:

In this game, various sound effects are present. When the ball hits the paddle, the ball or fire hits the bricks, a life is lost, a power-up is obtained, the laser paddle begins to fire, or a new level is begun, these sound effects will play. The "Option" menu's "Sound" icon can be disabled from the main menu page or the pause menu page if a player wants to disable the sound effects.

3. Score and Life Indicator:

There are many different sound effects in this game. These sound effects will play when the ball strikes the paddle, when the ball or fire strikes the bricks, when a life is lost, when a power-up is acquired, when the laser paddle starts to fire, or when a new level is started. If a player wants to turn off the sound effects, they can do so from the main menu page or the pause menu page by disabling the "Sound" icon in the "Option" menu.

4. Special Bricks:

Levels 2 and 3 of the game's difficulty can be increased by using some special bricks, which will make those levels harder. Which are:

4.1 Multi Hit Bricks:

These bricks will appear in levels 2 and 3 and require two hits to eliminate.

4.2 Invisible Bricks:

The third level will contain these bricks. These bricks will become visible when the ball strikes them for the first time.

5. Power-up Indicator:

The power-up indicator can be seen by the player in the top-middle corner of the page after the game has begun. The power-ups' hazy image is displayed in this indicator. When a power-up is obtained, its previously blurry image transforms into a bright one. The player can use this indicator to help them remember what special abilities they have in the game. When a player loses an ability, the vivid picture changes.

6. 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:

6.1 Level 1:

Bricks can only be hit once in this level. In this level, various power ups will sporadically appear. The difficulty of this level is much lower than that of other levels. We made an effort to make this game's first level as simple as possible for players.

6.2 Level 2:

Both single-hit and multiple-hit bricks are present in this level. In this level, various power ups will also appear at random.
The difficulty of this level is greater than level 1.
We made an effort to make the game's second level more challenging for the players.

6.3 level 3:

The game's final level is this one. Invisible bricks, multi-hit bricks, and single hit bricks are all present in this level. In this level, various power ups will also appear at random. Because the players won't be able to see the invisible bricks, this level's difficulty is greater than that of any other level in the game. A player will hit an invisible brick, which will cause bricks to appear and move the ball in the opposite direction. As a result, at this level, the player will encounter more challenges.

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 haeader 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, ,

$$\theta_0 = 30, v = BALL\_SPEED,$$

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

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

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.

$$v = BALL\_SPEED, w = Paddle's\_width$$

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

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

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

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:
Joty Saha, Roll: 51

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, 17the 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 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:
Asef Sami Chowdhury, Roll: 53

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.

Team Member 3:
Farhan Tanvir, Roll: 45

3.1 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.

3.2 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.

3.3 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.

3.4 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

3.5 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

3.6 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.

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:

We have tried to reduce the limitations as much as we can.Yet users have to face these limitations.

- ☐ Game can be played only on the Linux Operating system.
- ☐ High scores of only 20 players can be recorded.

Conclusion:

We gained a lot of knowledge from working on this project, but we also encountered several challenges. For us, creating a game is a completely new and useful experience. As a result, we had to start from scratch. We have been introduced to SDL (Simple DirectMedia Layer) library through this work. We are now aware of the creation and animation processes for models. Our capacity for thought and creativity has increased. We have improved our communication abilities through group collaboration.

Actually, for us it was a completely new experience.The challenge we've encountered is that creating a game is entirely foreign to us and differs from the programming we're used to. We had to start from zero by learning things that were entirely new to us. The internet, learning materials, and video tutorials have all helped us learn new things. Overall, it's a difficult task. It requires perseverance, time, and hard effort. Because we attempt to link the game environment with the actual world through the game, it can be claimed that creating games is a very sensible activity to engage in. It was challenging to deal with each and every aspect of the model to make it user-friendly.

At first, we believed it would be simple to create a DX-BALL game. But we later discovered that it's not that simple. since we had to do everything by ourselves. Everything has to be put into action by us. We had to do everything on our own, from moving the ball to keeping track of the score. We therefore had a lot of work to do, from the Ui to the game play. Therefore, the realisation or experience went much beyond what we had anticipated. Overall, it was a wonderful and novel experience, and we are pleased to have incorporated our own suggestions and those of the teacher into our own codes.

Future Plan:

- Improve the graphics of the game
- Introduce new levels
- Imply new unique features
- Make different versions of the the game
- Make it suitable for other operating systems like Windows, Mac etc.

Repositories:

GitHub Repository:
YouTube Video:

References:

- SDL Wiki: *https://wiki.libsdl.org/Tutorials*
- Lazy Foo: *https://lazyfoo.net/SDL tutorials/*
- SteamCommunity: *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*