1. OBJECTIVES

The objectives of this project are summarized as below:

- To familiarize with the Object-Oriented Programming (OOP) with respect to C++ language.
- To be able to use various features of C++ language.
- To be able to implement the knowledge gained through the lectures of our C++ lecturer into real life programming.
- To develop a game using almost all the features of C++ and using graphics libraries such as SFML.
- To develop the logic for good programming practice.
- To learn to work in a group and other project management skills.

2. INTRODUCTION

2.1. Background

Pinball game is simulated collection of pinball table. Pinball is a type of arcade game, in which points are scored by a player manipulating one or more steel balls on a play field inside a glass-covered cabinet called a pinball machine (or "pinball table"). The primary objective of the game is to score as many points as possible. Points are earned when the ball strikes different targets on the play field. A drain is situated at the bottom of the play field, partially protected by player-controlled plastic bats called flippers. A game ends after all the balls fall into the drain a certain number of times. Secondary objectives are to maximize the time spent playing (by earning "extra balls" and keeping the ball in play as long as possible) and to earn bonus games (known as "replays").

2.2. Overview

Creating games is always exciting as it is, playing. We wanted to revamp this classic pinball game by using its core rules and features and also adding our own customizations for better gameplay and user experience.

As an important part of our objective, we have made a multiplayer pinball game, named **"Pinball Versus",** where points are scored by trying to push the ball in opponents drain, and a player gets a point when the ball finally ends up in it. It can be played in two different modes viz. highest scorer in two minutes to win, and player to score first 5 points to win .In this project, we have used the SFML, a C++ graphics library to build the game.

3. APPLICATION

This project is a desktop game, and its sole purpose is for entertainment, passing good time with your friends, family, relatives, and so on. Games are popular worldwide in every possible format as it ensures entertainment, and a great refreshment when one is going through hard times. This project delivers not less than any other game, so that users can enjoy themselves whenever they want.

4. LITERATURE SURVEY

Since project is based on C++, we referenced "The Secrets of Object Oriented Programming in C++" by D.S. Baral. For SFML, there are various resources available in internet. We went through SFML documentation available in its official website and SFML series in youtube by Sonar System. We also referenced youtube video series by Tig, javidX9 and Coding Math for physics programming.

5. EXISTING SYSTEM

There has been many versions of computer simulations of pinball. The version of the game that everyone has heard of is the space cadet version which was included in past versions of windows. 3D Pinball for Windows – Space Cadet is a version of the Space Cadet table bundled with Microsoft Windows. It was originally packaged with Microsoft Plus! 95 and later included in Windows NT 4.0, Windows ME, Windows 2000, the 32-bit editions of Windows XP and Windows XP Professional x64 Edition. Windows XP was the last client release of Windows to include this game. Many open source modified and fancy pinball games are also available in internet. Our project has been inspired from Pinball Versus developed by PinballBox.

6. METHODOLOGY

6.1. Overview

Every project has a development stage and processes, and we have also tried to develop our project in a managed way so that its incorporation and merging doesn't become tedious. The work has been equally divided among each team member, supporting in each other's work if needed. Several discussions and suggestions have been made among us to ensure a quality final product.

The project development methodologies and stages to complete Pinball Versus can be summarized in points below:

- Analyzing the concept
- Installing SFML
- Familiarization with core concepts of SMFL

- Finding game resources, designing project structure
- Initial logic, gameplay physics and features planning
- Coding for the main project
- Execution and testing the program
- Debugging
- Additional features and improvements
- Program Documentation

Game's component wise description goes as follows:

6.2. Menu Design

Menu has been designed making separate class for its component design and drawing. The game contains two levels of menu, the second one comes when the player clicks on Play option to select the game mode.

6.3. Board Design and Assets Selection

Pinball board/arena was designed using photoshop and various images available in internet. The board design was finalized after numerous trials and visualizations, so that the game looks attractive as possible. The obstacles like bumpers, pins, and other assets like ball and flippers were selected as per our requirement. These were the most important assets to be collected to maintain the concept of pnball game.

6.4. Ball Physics

As much as it is attractive and interesting game to be played, physics behind it is even more interesting. In order to program ball physics, we had to take into account the gravity and corresponding change in its velocity. We had to make it update ready for the possible collision, and resulting magnitude and direction collisions. For this, we have used the concept of components of vector, and update it as requirement. We have reversed the velocity with certain losses when hit the wall or factor when hit by the flipper. The ball roles down the flipper when the player doesn't hit the strike key.

6.5. Collision Detection

This is the most important part in programming pinball physics, as ball moves her and there in the board and collides with sides, obstacles and flippers. To detect ball-bumper and ball-pin collisions, we have used circle-circle collision detection technique of finding the distance between the circles, and comparing with their radii. For ball- wall collision, we have used the concept of distance between point and line and comparing the distance with the radius if it is equal or shorter. We used two-point form of equation of line by finding out two extreme points of a segment of a wall. For ball-flipper collision, we have used the concept of dot product and projection to find the closest point between line (side of rectangle) and center of circle, and comparing the distance between them with the radius, if it is equal or shorter.

6.6. Flipper Physics

Flipper is another important component of the game. Its collision detection has been discussed above. For the response part, the ball's velocity is changed according to the point on the flipper, where the ball strikes, longer the distance from its axis of rotation, ball returns with more velocity. The ball's velocity is independent of collision point, if player hasn't flipped the flipper, as it is made to roll on the flipper in such situation.

6.7. Game Updates and Game Modes

We made it to be played in two different modes. The highest scorer in two minutes time to win, and the five points scorer to win. The game goes back to menu when a round is completed. We have made the score to be updated when the ball ends up in opponent's drain. If it ends up in one's own drain, it is made to be considered as suicide points.

6.8. Sound Effects

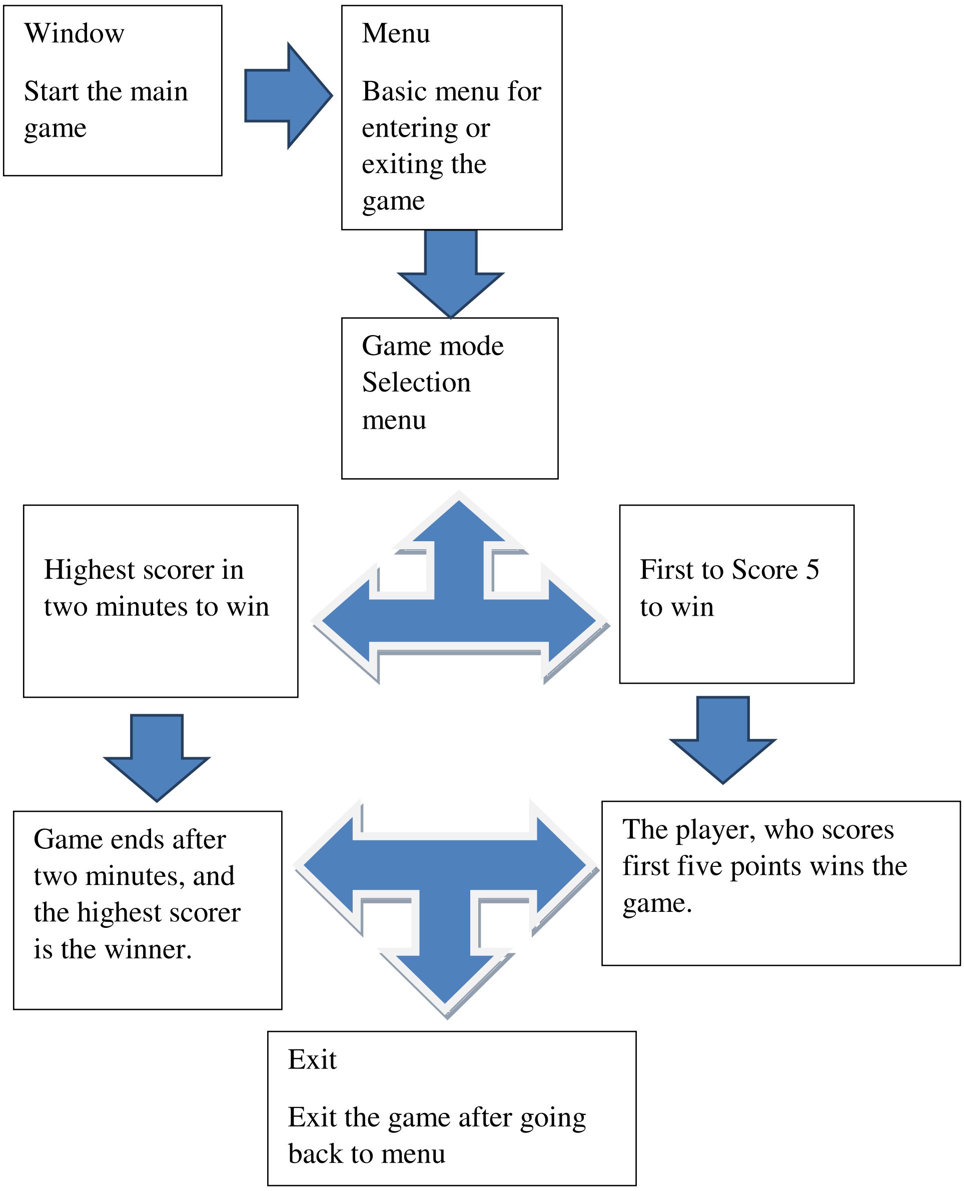
Various sound effects have been used for events like rolling, striking and so on.

7. IMPLEMENTATION

7.1. Overview

The game starts with basic menu to enter the game. When the player chooses Play option, game mode selection menu occurs where players can choose either of the option as mentioned above to enter into the game. Then the game goes back to menu when any player scores 5 goals first, or after 2 minutes depending upon the game mode. It is summarized in following block diagram.

7.2. BLOCK DIAGRAM


Window
Start the main game
Menu
Basic menu for entering or exiting the game
Game mode Selection menu
Highest scorer in two minutes to win
First to Score 5 to win
Game ends after two minutes, and the highest scorer is the winner.
The player, who scores first five points wins the game.
Exit
Exit the game after going back to menu

8. RESULTS

By applying aforementioned methodologies, the project was successfully completed, and the implementation is clearly shown in block diagram above. The results were similar to what we had planned, and players can easily play the game with good gaming experience. The ball, flipper movements were fluid, no glitches were observed during transition between menus and games, or during the game. We had aim for sound UI and gameplay, which we achieved by the end of project completion.

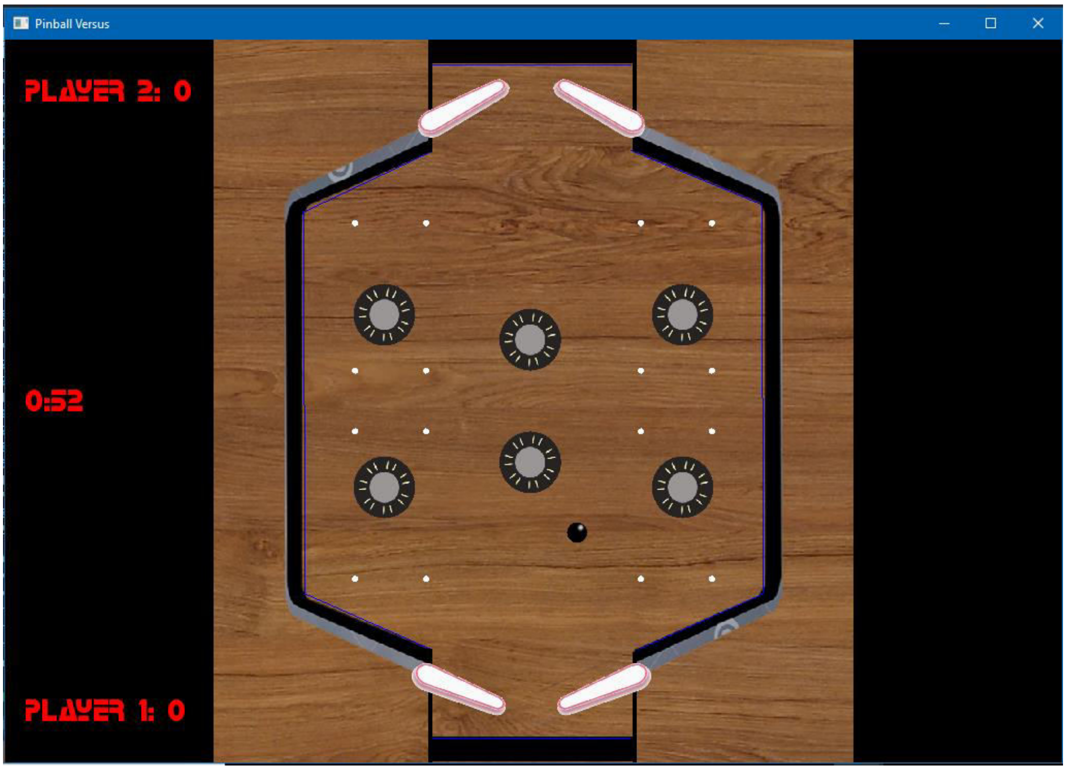


Fig: A glimpse at Pinball Versus

9. PROBLEM FACED AND SOLUTIONS

It is unlikely that the project is completed without facing certain problems and hurdles. So, we also faced several problems, which are listed below:

- We found it hard to find the proper collision detection and response of ball-flipper collision.

- Initially, we faced problems while designing the board, due to lack of image editing skills.
- We also struggled for some time for boundary collision.
- We faced problem to mirror the texture and rotate the sprite about new origin.
- We faced problem to implement directed line segment (vector) properly.

These problems were respectively solved as follows:

- Proper collision detection was achieved through dot product and projection concept, and proper response was achieved through projection length and with the help of perpendicular vector to the flipper.
- We learned few photoshop skills to overcome the designing problem.
- Concept of equation of line was used for boundary collision.
- Mirroring texture was solved using setScale() member function of SFML RectangleShape class.
- We solved the problem of vector by taking difference of two points of which we wanted a vector, and translating the vector to initial point.

10. LIMITATIONS AND ENHANCEMENTS

Current project has only multiplayer option. Single player option can be added in future. Similarly, the game can only be played between two human players, not with computer. This feature can also be added in future using Artificial Intelligence. It is only a simple prototype, which can be enhanced by including tournament modes, pinball powerups, including rewards, playing in different maps and so on.

11. CONCLUSION

Hence, in this way, the project **Pinball Versus** was successfully completed. We not only learned to manage and develop a real world project, but also the importance of team work, and learned how to work in a group. We also got hands-on experience of using OOP concept to model real world objects like ball, board flipper, and how it makes large program management and maintenance easier.

12. REFERENCES

- Daya Sagar Baral, Diwakar Baral, *"The Secrets of Object Oriented Programming in C++"*,Bhundipuran Prakashan, Bagbazar, lalipur
- https://www.sfml-dev.org/
- Sonar System SFML series: https://www.youtube.com/playlist?list=PLRtjMdoYXLf776y4K432eL_qPw4na_py3
- Tig's Intersection Between Line and Circle | 2D Segment Collision Algorithm: https://www.youtube.com/watch?v=_3dRFu3k8Nw&t=315s
- javidX9's Programming Balls #1 Circle Vs Circle Collisions C++: https://www.youtube.com/watch?v=LPzyNOHY3A4&t=153s
- https://pinballbox.itch.io/pinballversus