Course Project Documentation

Project Report

POOL GAME

TEAM ID-115

Darshit Jain	14D070039
Abhishek	140040062
Vivek Turkar	140040052
Seema Kumari	140040059

Submitted for CS 101 (Prof. KAVI ARYA)

Table of Contents

1.0. Introduction

1.1. Purpose of Document

The purpose of this document is to present a detailed description of the Pool game that our team made. The following sections of the document explain the purpose and features of the game, the interface of the game, and the constraints under which it must operate.

1.2. The Project

We have made a straight Pool game. It is not the replica of the original game but stands out as a ‘new’ game based on the original idea. A ball and table is used to serve this purpose and the trajectory of the ball is decided by the user controlled movement.

1.3. Scope of Project

This game is meant for pure entertainment for people who enjoy playing mini-games and will attract more fans like people on the go. This game can be even made an online (free to try) in the future. It can be also replicated easily onto any mobile platform. For now it is employed as a PC mini-game. The Pool game has been there for years now but we made it a bit different by using simplecpp graphics introducing simple rules in the game.

2.0. PROBLEM STATEMENT

The aim of the project is to code a user-friendly pool game using simplecpp graphics.The aim of game is to pocket all the balls(not the cue ball) with minimum number of strokes.(Straight Pool).

Other aims are calculating the high score(no. of strokes),creating playing modes i.e. hard and easy.The only difference between these options is speed of cue ball. In easy mode speed of cue ball increases and hence random motion increases and probability of pocketing goes up.

The game can not be completed without the basic functions of collision,motion,reflection,friction and some basic physics.

3.0. REQUIREMENT SPECIFICATIONS

3.1 Functional Requirements / Modules

Starting the game

1. Game starts upon clicking the executable equivalent file of the game.
2. Window appears showing logo of the developer (team) involver
3. Main Menu appears with the following options
    a. NEW GAME
    b. OPTIONS
    c. INSTRUCTIONS
    d. HIGH SCORES
    e. QUIT

In Game

1. Using the mouse the user will be able to control the cue ball to strike the ball to other balls and thus complete the object of pushing all the balls in the holes.

2. The velocity of the ball is proportional to the distance of click from cue ball by user and ball moves in the direction of click.

Exiting/Ending the Game

1. The user can leave the game by pressing the QUIT option in the main menu.
2. While playing a game, clicking the 'MAIN MENU' key takes the user to the Main Menu from where he can end the game by following the above procedure.

3.2 Non-Functional Requirements

1. **General Guidelines:** Priority should be given to performance, adaptability, maintainability, and usability.
2. **Operating Constraints:** This program requires the simplecpp library to be present on the target machine and for the computer to have C++ Redistributable package in case of Windows and C++ Runtime Environment in case of Linux. Codeblocks compiler can work in both cases.

3. **Documentation:** The User Manual will be made for the application in addition to the INSTRUCTIONS included inside the 'Main Menu' of the game.
4. **Portability:** The program will run on Windows XP/Vista/7, and Ubuntu 10.04/11.04/11.10.The speed is bit slow in Windows environment.
5. **Reliability:** As the game is for pure recreation and involves no user data, reliability is of low importance.
6. **Deployment:** The game will be uploaded online in the form of a C++ executable file.

3.3 User Characteristics

The Player is expected to be literate in the computer basics and be able to use a keyboard or a mouse.

4.0. IMPLEMENTATION

Let's start with beginning of the project and step by step see the implementation.
So the first task was to code for the table and balls. Rectangle ,circle features in simplecpp were used. Structure was defined for Ball having elements as Circle ,Velocity initialized to (0,0).After generating the playing arena next step is motion. Getclick was used to accomplish this.Final click is stored as second point and a velocity proportional to distance is given to ball.
Other functions such as reflection ,friction .collision were implemented using simple laws of physics.
Other functions are distance between ball,a function that checks whether all balls are stationary so as to prompt the user for next click and no. of balls pocketed.
HighScore was implemented by creating a text file "highscore.txt" and using I/O stream to keep count of highscore.
Everything else is just the basic Simplecpp Graphics. ☺

4.1 Assumptions and Dependencies

I. The application is not internationalized; all menus and messages are in English.

II. The application will run on Windows XP/Vista/7 and Linux platforms with the C++ Runtime Environment installed.

III. The application does not require an internet connection, other than to download the application from the product website.

5.0. TESTING STRATEGY AND DATA

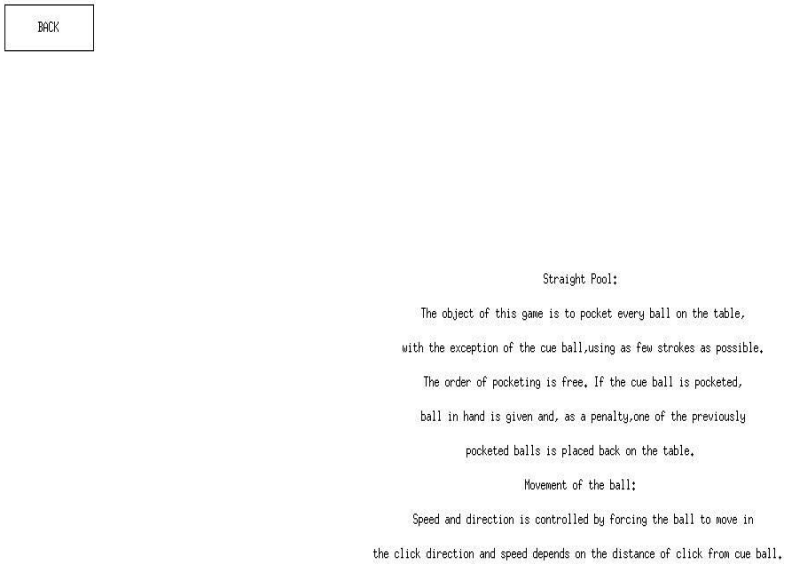
This section is primarily to notify the TA and the Professor in charge about the test cases that have been done by the developers.

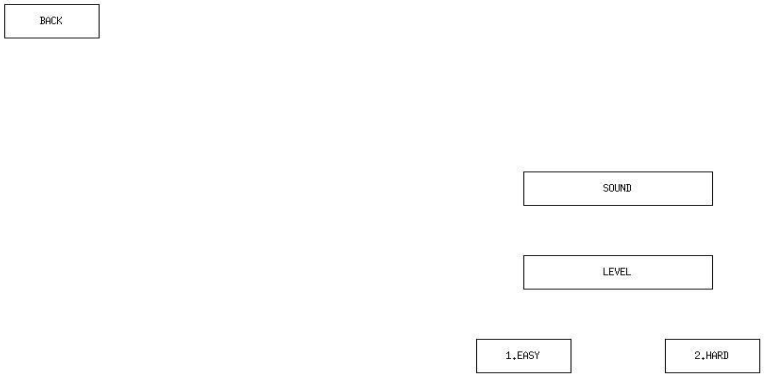
Functions such as distance between ball ,collsion were tested by taking output in terminal window for known cases.

Other objectives such as table ,reflection ,motion were tested in the canvas window and adjusted suitably.

There is the main window on the next page-:

The instuctions and option window is like-:





And finally here comes the playing arena:-

6.0. DISCUSSION OF SYSTEM

6.1. Work as per plan

We have completed all the stuff that was promised in the SRS.The only difference from SRS is that we have switched our assigned tasks among ourselves.

The project is completed using simplecpp library and graphics as promised.

6.2. Features more than mentioned in SRS

Nothing more as such is added in the game apart from that in the SRS but we have tried to show a line where the ball will move using mouse click drag event and such functions but at last gave it up as it wasn't functioning in UBUNTU.

6.3. Changes in plan

No changes in plan were there.

We have successfully completed our project using simplecpp library and graphics.

7.0. FUTURE PROSPECTS

After the completion of the product, various errors may be reported by the users. Post-production stage will thus involve maintaining the product for a certain period of times. Fixes to errors in the form of updates will be issued and will be available on a suitable website. The mini-game can also be ported as a mobile game and made available on tablets after suitable changes are made for increased adaptability.

Changes for the pc game involves implementing 3D graphics and sound.
Apart from this common prospect to modify a game what one can do is addition of different modes such as-

Free Training is, as the name says, completely free. There are no rules; even the break stroke does not have to hit the ball on the foot spot, as in all the other pool.

Rotation:

This game is quite similar to Pocketing, except that the cue ball must first hit the lowest-numbered ball on the table on every shot. If the cue ball hits one of the other balls first, all the balls pocketed with the shot will be put back on the table. If the cue ball is pocketed, ball in hand is given and, as a penalty, one of the previously pocketed balls is placed back on the table.

Straight Pool 99:

Straight pool 99 is like straight pool, except you have to pocket 99 balls. When there is only one ball (or none) left on the table, all balls are returned onto the table so you always have balls to pocket. The goal is still to pocket these 99 balls with as few strokes as possible.

Continuous:

Continuous is single player game where the game lasts as long as you shoot at least one ball (not the cue ball of course) in a pocket. Just like in Straight pool 99, balls are returned onto the table when you run out, so there is no limit on how long the game can last. The game ends when you fail to pocket any ball. Your score is the amount of strokes you've made. (Note: your score is **not** the amount of balls you've pocketed. So no matter how many balls you pocket, only one point is given per stroke.)

8.0. CONCLUSION

Ours was a relatively simple project with nice applications of the taught topics in the course CS-101 .We have come up with correct usage of functions , structures , classes and made the most out of the simplecpp graphics. Concluding , it was enjoyment to work on such a nice project and the project helped learn a lot and apply practically what was taught.

10.0. REFERENCES

We have used just a single source for the whole project and that is:

"AN INTRODUCTION TO PROGRAMMING THROUGH C++"

BY ABHIRAM RANADE

Those two chapters on graphics are enough to make a nice project.