

Computer Science Project

Game

Daniyal | XII -A| Roll No – 14

Index

|  |  |
| --- | --- |
| **Description** | **Page Number** |
| **Certificate** |  |
| **Acknowledgement** |  |
| **Introduction** |  |
| **Features** |  |
| **Classes Used/ File(s) Used** |  |
| **Functions Used** |  |
| **Sample Run** |  |
| **Bibliography** |  |
| **Program Code** |  |

Acknowledgement

I'd like to express my greatest gratitude to the people who have helped & supported me throughout my project. I’m rather grateful to our school’s Computer Science faculty.

I thank Mrs. Vandana Sharma for her continuous support for the project, from initial advice and encouragement to this day. Special thanks of mine goes to my colleagues who helped me in completing the project.

I wish to thank my parents for their undivided support & interest who inspired me & encouraged me to go my own way without which I would be unable to complete my project. At last but not the least I want to thanks my friends who appreciated me for my work & motivated me.

CERTIFICATE

This is to certify that Daniyal Abbasi of class XII has completed the C++ project on “Runjee Jump (Game)” under the guidance of Mrs. Vandana Sharma, for his class XII Computer Science Practical Examination of Central Board of Secondary Education in the academic year 2016 -2017.

It is further certified that the project is the individual work of student.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Vandana Sharma

(HOD Computer Science, Somerville School NOIDA)

Introduction

This program is an example of using graphics concept to develop an application using C++. This game an interactive graphical game which has been prepared by which users will able to get graphical mode on command line interface. Users will have to just run the program and program will guide the users how to play the game and exit from it.

The game has a simple objective. The player has to use the keyboard in order to make the ball jump over the incoming buildings. When the ball collides with the building, the game is over.

High score of the user is saved in his account and the leader-board shows the top 5 high scores of different users.

Features

The program greets its users with an attractive welcome screen made with the help of graphics.

Following the welcome screen, a menu comes where the user can either sign-in or sign-up. There is also a provision for administrator login as well.

A new user will first sign-up where he is required to enter his Full Name, Username and Password. The username has to be unique and should not be redundant.

The password typed is hidden in order to maintain privacy.

The objective of the game is, to make the ball jump over the incoming buildings. Each building successfully jumped over adds to the score. When the ball collides with the building, the game is over.

The high score of each player is stored in his account details and the top five high scores are displayed. The high scores are sorted using bubble sort.

The users can edit their account by changing their username or password. Also the have an option to delete their account as well.

The administrator has the option to check the leader board, view all accounts and also has the option of deleting all the accounts.

The admin is a dynamically created object whose details can not be changed. The objects of user are stored in a binary file named, “Game\_data.dat” which can be edited and the edited information is stored in the file.

Classes Used

A class named “player” is used in the program. It is the main account frame for each player.  
The class has the following data member :

* Character array named “name” which stored the name of the user.
* Character array named “username” which stored the username of the user.
* Character array named “password” which stored the password of the user.
* An integer named “userid” which stored the user ID of each player.
* A unsigned integer named “highscore” which stored the high score of the user.

The class has the following member functions:

* A parameterized constructor player which creates an object using the given name username and password.
* A function “compare” of return type void which takes score as it parameter, compares it with the high score of the user and replaces it if score it greater than high score.
* A function “change” of return type void which takes username and password as it parameter and changes the username and password of the account.
* A function retuserid of return type integer which returns the userid of the user.
* A function rethigh of return type integer which returns the highscore of the user.
* A function retusername of return type char\* which returns the username of the user.
* A function retpassword of return type char\* which returns the password of the user.
* A function retname of return type char\* which returns the name of the user.

Files Used

A binary file named “game\_data.dat” is used in order to store all the details of different users.

Functions Used

|  |  |
| --- | --- |
| **Name of Function** | **Purpose** |
| void welcome() | Attractive welcome screen with graphics |
| void login() | Login menu, with sign in, sign up and admin login. |
| void usermenu() | Menu for the user with play, leader board, instructions, edit account and delete account. |
| void game() | The main function for the game |
| void jumphigh() and void jumplow() | Functions used to jump the ball. |
| void showdetails() | Displays the details of object of class player. |
| void leaderboard() | Shows personal high score and top four scores of different users. |
| void displayall() | Displays all the details of all account, only accessible by the administrator. |
| void funcwrite() | Function to write any changes of details to the file for the signed in player. |
| void deleteall() | Deletes all the user details. |
| void adminmenu() | Menu for administrator. |
| void signin() | Function used to input details and sign in the user. |
| void signup() | Function used to input details and sign up the user. |
| void adminlogin() | Function used to input details and sign in the administrator. |
| void editaccount() | Function used to edit username and password of an account. |
| void deletep1() and void deleteaccount() | Function used to delete an account by the user themselves. |
| void instructions() | Function to display instructions about the game and controls. |
| void endgame(int score) | Function to display final score after each turn. |
| void exitscreen() | Displays goodbye and closes the game. |

Bibliography

* Computer Science –Sumita Arora – ClassXI
* Computer Science –Sumita Arora – ClassXII
* stackoverflow.com
* ioscpp.org
* cplusplus.com