International Institute Of Information Technology, Bhubaneswar

Information Technology(IT)

Department Of Computer Science

Multiplayer Quizzing Game using C++

B316011	Nawed Imroze
B316033	Ankur Kumar

November 8, 2017

IIIT BHUBANESWAR

IIIT Bhubaneswar

Imagine, Innovate, Inspire

Report

1. Introduction

The project is a feature rich, single as well as multiplayer quiz game. It is written in C++ language, compiled using GCC compiler. The project is text based Graphical User Interface (GUI).

The project uses the Object Oriented Programming features of C++ and concepts of file handling. The whole source code is divided into module that makes the program readable and easy to understand.

2. Basic Outline

2.1 Header files used:

a) ncurses.h
b) iostream
c) fstream
d) string

2.2 Classes defined:

a) display

It creates splash screen which is the entry point into the game. It has options like new game, help and exit. The class consists of three function are splashScreen(), help(), exit() which are produce the required screen.

b) playerInfo

This class has data member as array of string variables to store names of the players. The game is designed to have maximum 6 players so array can accommodate names of maximum 6 players. The functions included are getPlayerName() which is used to obtain the names of the players. The other function is showPlayer(int) which takes player number as input and returns the name of the player.

c) round

This class has integer data member named roundNumber which is used to store the round number that is currently being played. It has 3 functions. The function updateRound() is used to increment the round number. The function getRoundNumber() is used to return the current round number. The displayRound(int) function is used to show the information about the round like round name before the beginning of the round.

d) score

This class has array of 6 integer type data types named as points which is used to store the scores of the players playing the game. It has the constructor which initializes the score of all players to zero. The function scoreUpdateDirect(int, bool) takes player number as integer and whether the answer is correct or not as bool data type. It is used to update the score of the player on direct question. There is another similar function namedscoreUpdatePass(int, bool) used to update the score of the player it is pass question.

e) newGame

This class publicly inherits the classes playerInfo, round, score. It has one integer type data member called players which stores the number of players. The playerCount() function is used to take input from user about how many players are playing the game. The function getPlayerCount() returns the number of players. The function showAllScore() displays the current score of the players. The function finalScreen() shows the final standings in terms of score of the players playing.

2.3 main() function:

An object named disp is created to display the splash screen, help and exit art. Object game of class newGame is declared. It takes the player count and the names of the players. Then the game starts. The questions and the four options are read from the questions file and displayed on the screen using mvprintw(row, col, string) which is used to display anything at the coordinate row, col. The function getmaxyx(stdscr, row, col) is used to get the width and height of the current console or terminal where the program is running. Then the user enter option from A, B, C, D. The user input is checked with correct answer and if the answer is correct then the scores are updates and next question is asked. If the answer is wrong then scores are decremented and the correct answer is displayed. The user can opt to pass the question without attempting. In that case the question will be asked to next player in the sequence. After the end of all round the final standings of all the players is displayed and the winner's name is shown. Then again the game goes to the main menu.

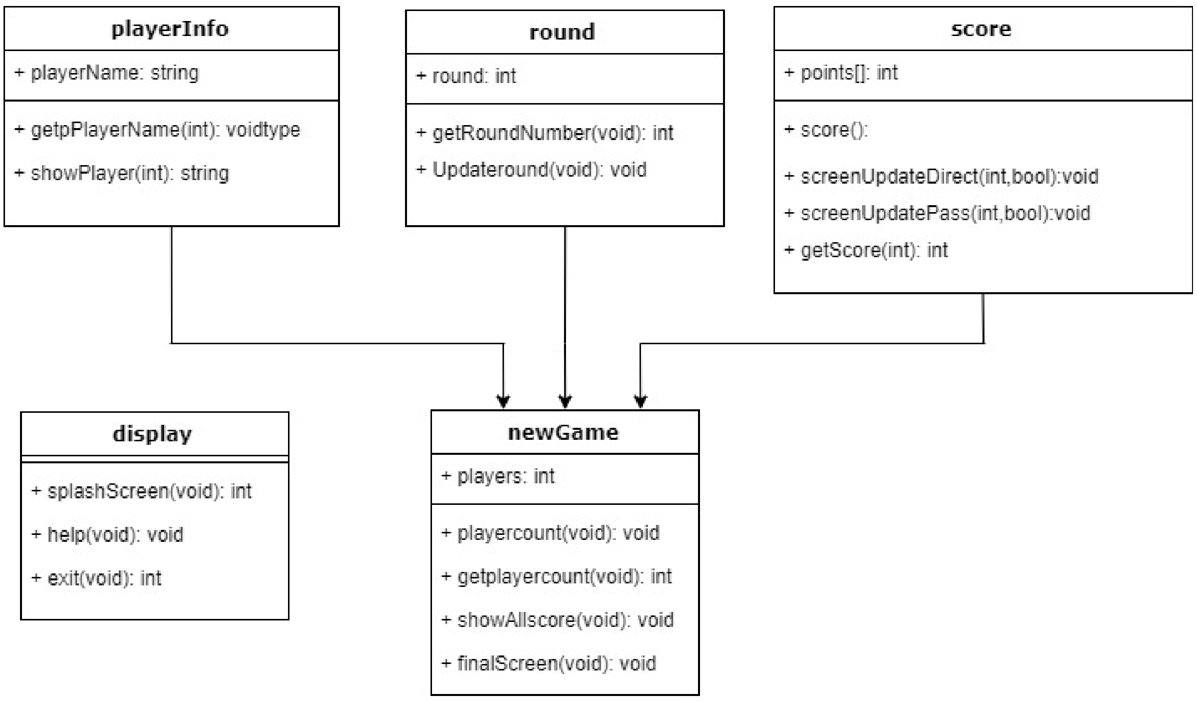
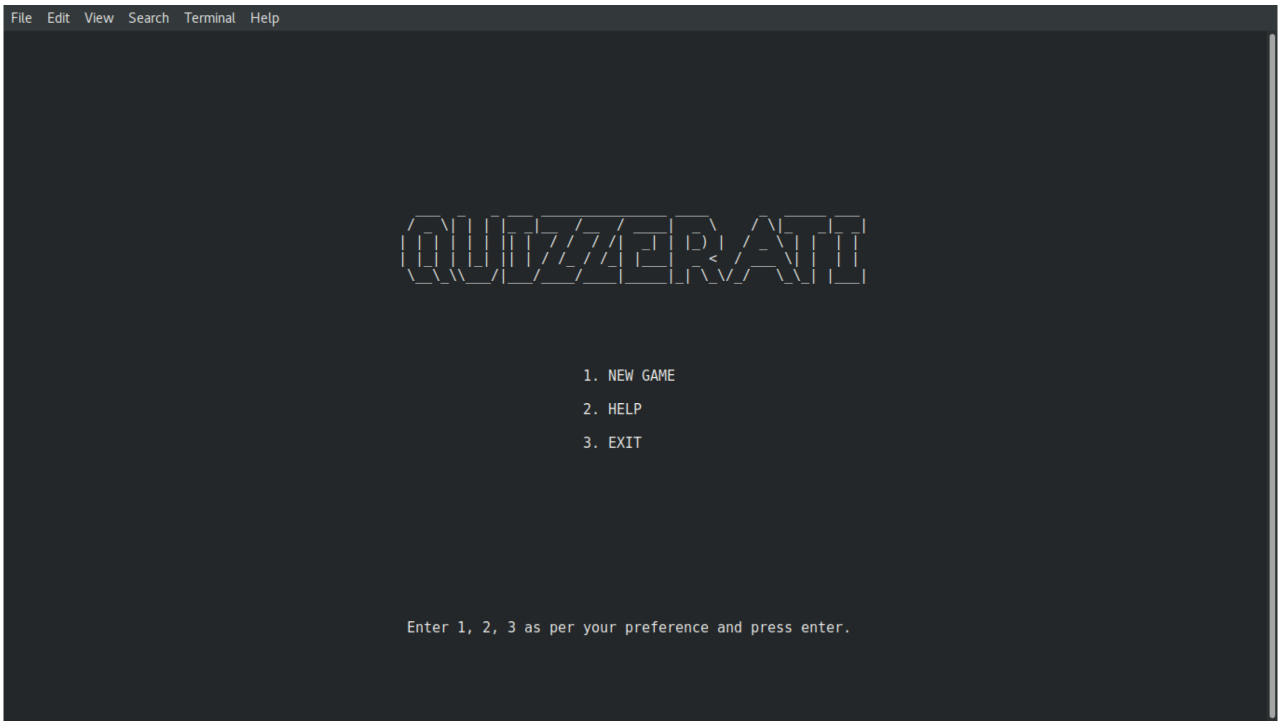

playerInfo
+ playerName: string
+ getpPlayerName(int): voidtype
+ showPlayer(int): string
round
+ round: int
+ getRoundNumber(void): int
+ Updateround(void): void
score
+ points[]: int
+ score():
+ screenUpdateDirect(int,bool):void
+ screenUpdatePass(int,bool):void
+ getScore(int): int
display
+ splashScreen(void): int
+ help(void): void
+ exit(void): int
newGame
+ players: int
+ playercount(void): void
+ getplayercount(void): int
+ showAllscore(void): void
+ finalScreen(void): void

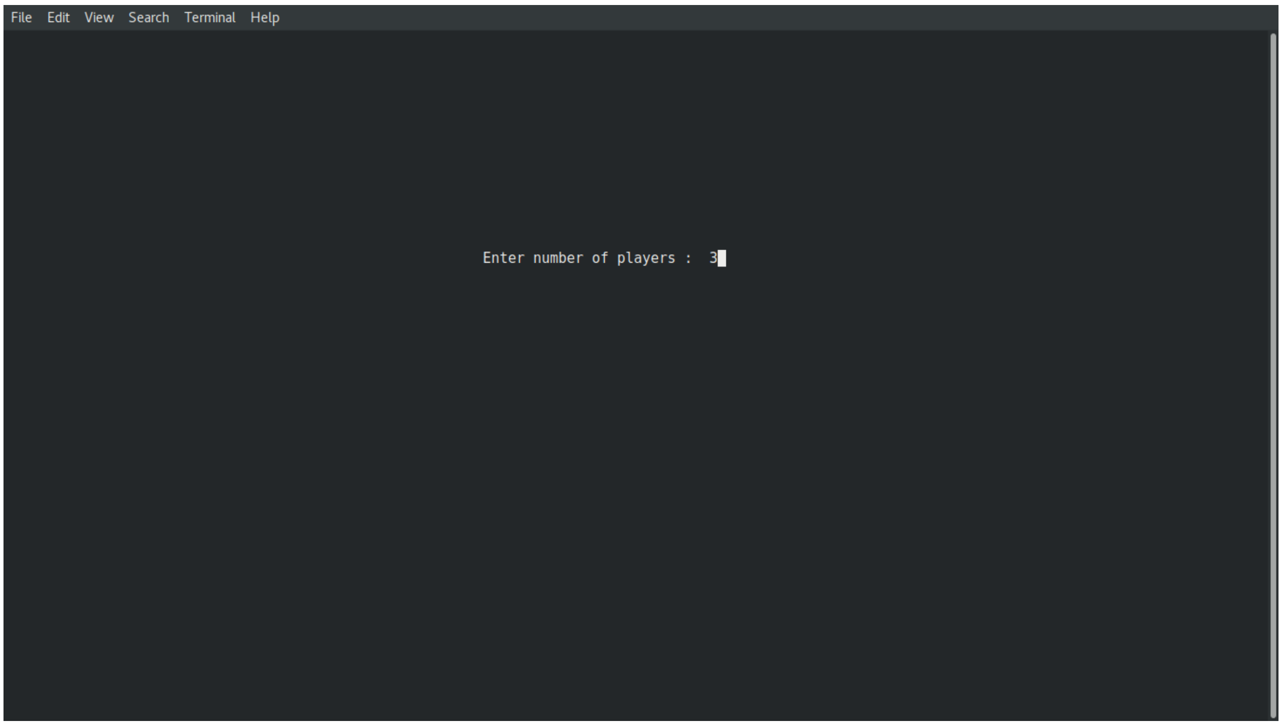
Figure 1: This is a flow chart showing inheritance structure of classes

2.4 Walkthrough:



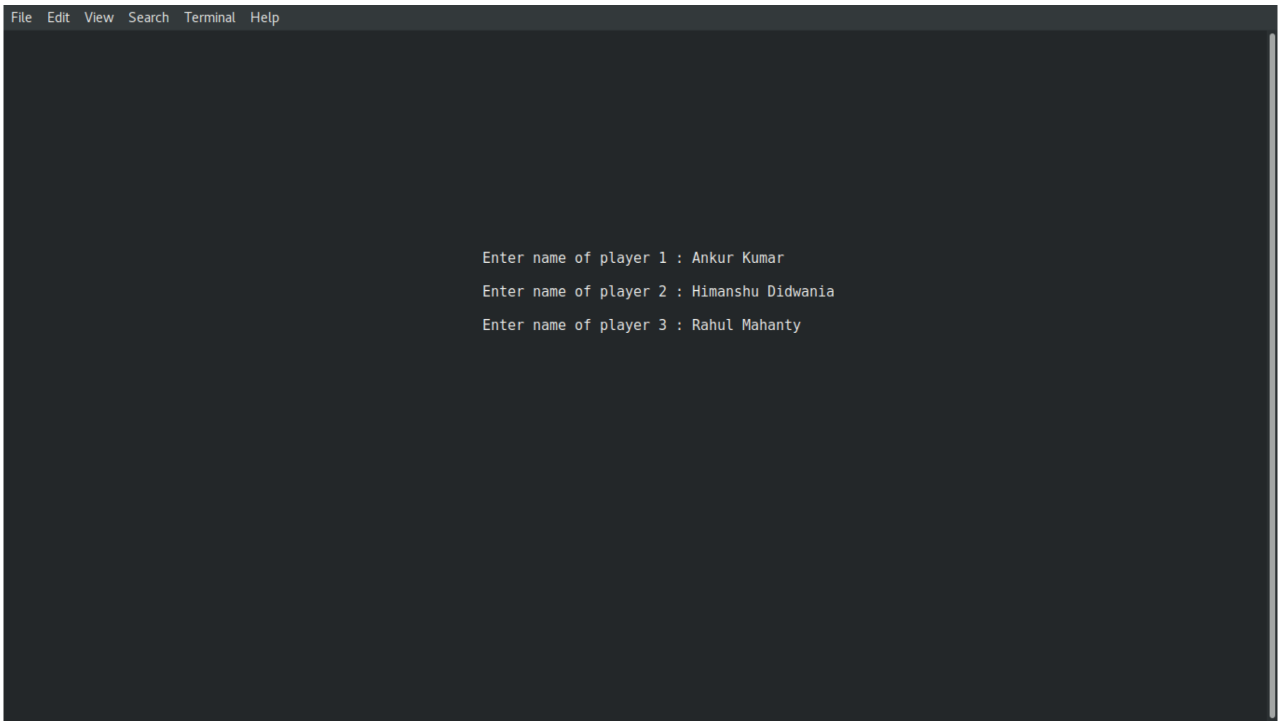
```
  ___  _   _ ___ __________ _____ ____      _  _____ ___
 / _ \| | | |_ _|__  /__  /| ____|  _ \    / \|_   _|_ _|
| | | | | | || |  / /  / / |  _| | |_) |  / _ \ | |  | |
| |_| | |_| || | / /_ / /_ | |___|  _ <  / ___ \| |  | |
 \__\_\\___/|___/____/____||_____|_| \_\/_/   \_\_| |___|


                    1. NEW GAME

                    2. HELP

                    3. EXIT



       Enter 1, 2, 3 as per your preference and press enter.
```

Figure 2: This is the splash screen or main menu of game



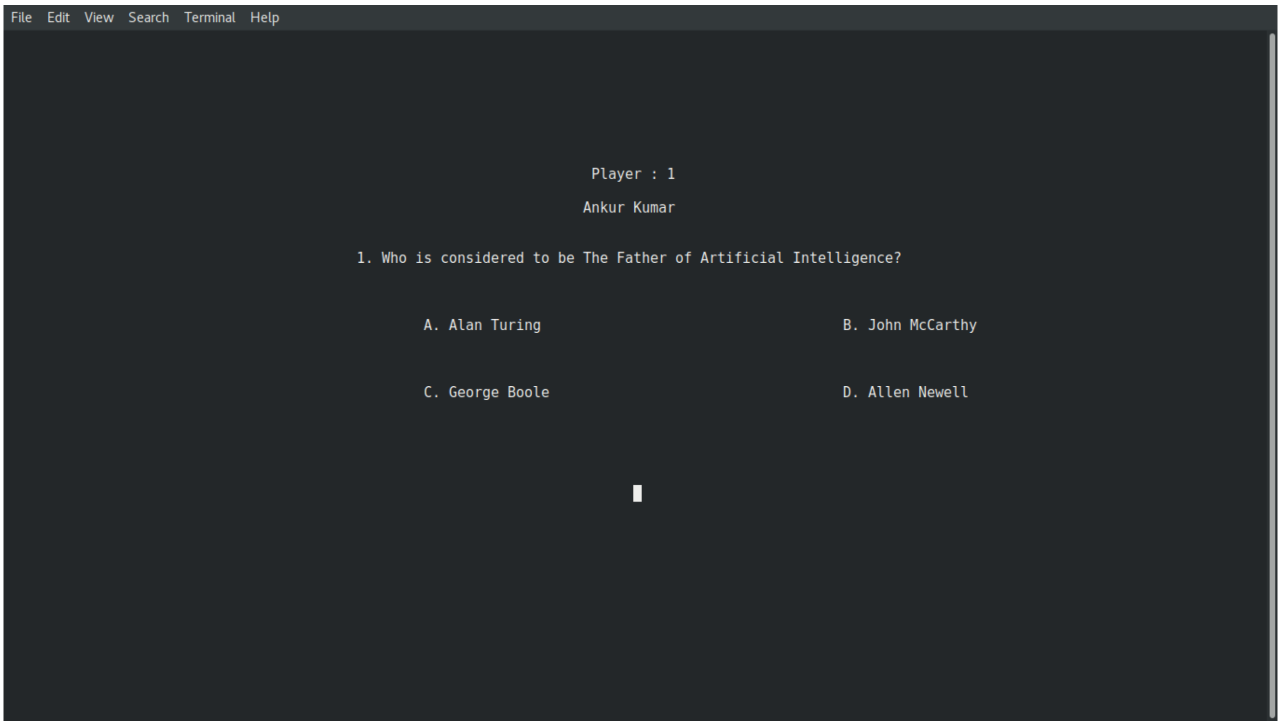
```
Enter number of players :  3
```

Figure 3: Prompts user to enter number of players



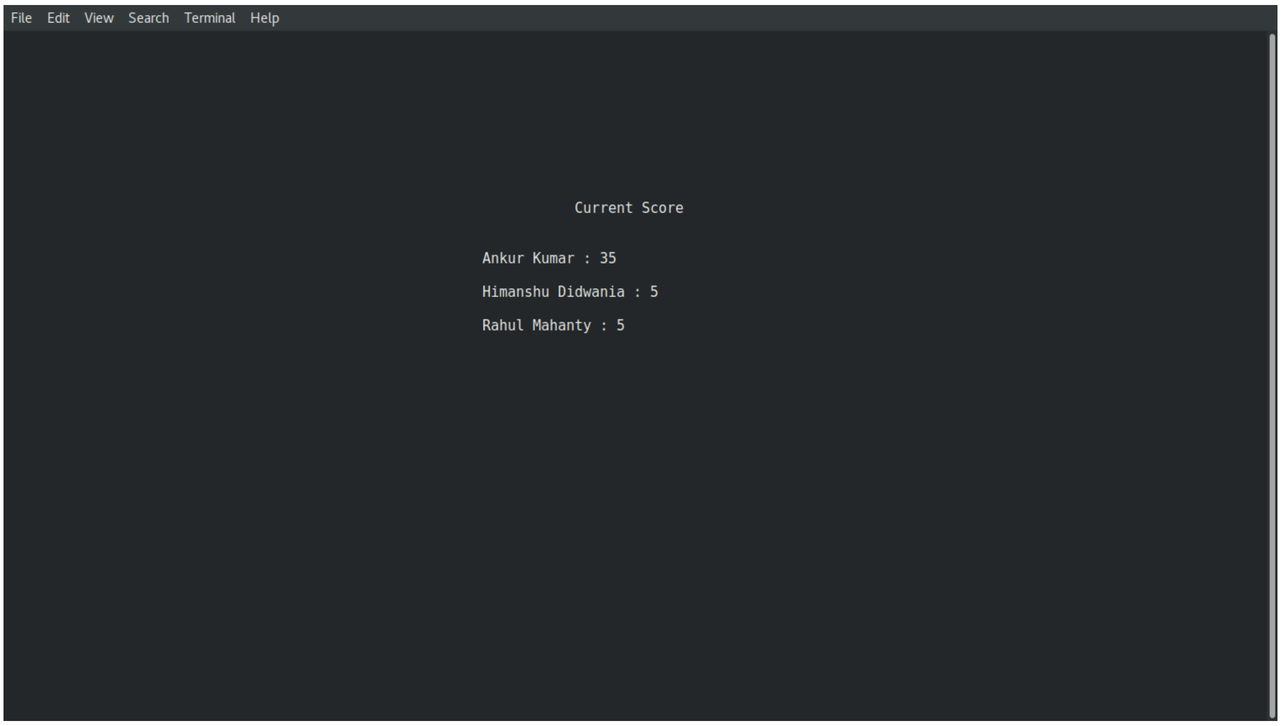
```
Enter name of player 1 : Ankur Kumar

Enter name of player 2 : Himanshu Didwania

Enter name of player 3 : Rahul Mahanty
```

Figure 4: Prompts user to enter names of players



```
                         Player : 1

                        Ankur Kumar

  1. Who is considered to be The Father of Artificial Intelligence?


        A. Alan Turing                      B. John McCarthy


        C. George Boole                     D. Allen Newell
```

Figure 5: Shows question to the player



```
                    Current Score

        Ankur Kumar : 35

        Himanshu Didwania : 5

        Rahul Mahanty : 5
```

Figure 6: Shows current score of all players after each round

Figure 7: Shows the exit screen of the game

3. Reference:

1. www.geeksforgeek.com
2. www.cppreference.com
3. www.stackoverflow.com
4. **Book::E. Balaguruswamy (Object Oriented Programming using C++)**

4. Source Code:

```cpp
#include <iostream>
#include <ncurses.h>
#include <string>
#include <fstream>
using namespace std;
string roundList[] = {"Technology", "Sports", "Science", "Politics", "Literature"};
string questionsFileList[] = {"questions/technology.txt", "questions/sports.txt", "questions/science.txt", "questions/politics.txt", "questions/literature.txt"};

string getstring()
{
    string input;
    // let the terminal do the line editing
    nocbreak();
    echo();
    // this reads from buffer after <ENTER>, not "raw"
    // so any backspacing etc. has already been taken care of
    int ch = getch();
    while ( ch != '\n' )
    {
        input.push_back( ch );
        ch = getch();
    }
    // restore your cbreak / echo settings here
    return input;
}

class display{
public:
    int splashScreen()
    {
        int i, row, col, response;
        initscr();
        getmaxyx(stdscr, row, col);
        ifstream file;
        file.open("resources/ascii-art/splashscreen.txt");
        string s;
        for(int i=0; getline(file, s); i++)
            mvprintw(row/4+i, (col-s.size())/2, "%s", s.c_str());
        file.close();
        mvprintw(row/2, (col-11)/2, "1. NEW GAME\n");
        mvprintw(row/2+2, (col-11)/2, "2. HELP\n");
        mvprintw(row/2+4, (col-11)/2, "3. EXIT\n");
        mvprintw(row-row/6, (col-53)/2, "Enter 1, 2, 3 as per your preference and press enter.");
        refresh();
        cin>>response;
        clear();
        refresh();
        return response;

    }
    void help()
    {
        int i, row, col, response;
        initscr();
        getmaxyx(stdscr, row, col);
        ifstream file;
        file.open("resources/ascii-art/help.txt");
        string s;
        for(int i=0; getline(file, s); i++)
            mvprintw(row/4+i, (col-s.size())/2, "%s", s.c_str());
        file.close();
        getch();
        clear();
        refresh();
    }
    int exit()
    {
        int i, row, col, response;
        initscr();
        getmaxyx(stdscr, row, col);
        ifstream file;
        file.open("resources/ascii-art/exit.txt");
        string s;
        for(int i=0; getline(file, s); i++)
            mvprintw(row/3+i, (col-s.size())/2, "%s", s.c_str());
        file.close();
        getch();
        clear();
        refresh();
        return 0;
    }
};

class playerInfo{
protected:
    string playerName[6];
public:
    void getPlayerName(int playerCount)
    {
        int i, row, col;
        initscr();
        getmaxyx(stdscr, row, col);
        for(int i=0; i<playerCount; i++)
        {
            mvprintw(row/3+(2*i), (col-25)/2-5, "Enter name of player %d : ", i+1);
            playerName[i] = getstring();
            refresh();
        }
        clear();
              refresh();
          }
          string showPlayer(int playerNumber)
          {
              return playerName[playerNumber-1];
          }
      };

      class round{
          int roundNumber;
      public:
          void updateRound()
          {
              roundNumber++;
          }
          int getRoundNumber()
          {
              return roundNumber;
          }
        void displayRound(int j)
        {
            int i, row, col, response;
            initscr();
            getmaxyx(stdscr, row, col);
            mvprintw(row/3, (col-9)/2, "Round : %d", j+1);
            mvprintw(row/3+4, (col-roundList[j].size())/2, "%s", roundList[j].c_str());
            getch();
            clear();
            refresh();
        }
    };

    class score{
    protected:
        int points[6];
    public:
        score()
        {
            for(int i=0; i<6; i++)
                points[i]=0;
        }
        void scoreUpdateDirect(int playerID, bool responseStatus)
        {
            if(responseStatus==true)
                points[playerID]+=10;
            else if(responseStatus==false)
                points[playerID]-=5;
        }
        void scoreUpdatePass(int playerID, bool responseStatus)
        {
            if(responseStatus==true)
                points[playerID]+=5;
            else if(responseStatus==false)
                points[playerID]-=2;
        }
        int getScore(int playerID)
        {
            return points[playerID];
        }
    };

    class newGame : public playerInfo, public round, public score{
        int players;

    public:
        void playerCount()
        {
            int i, row, col, response;
            initscr();
            getmaxyx(stdscr, row, col);
            mvprintw(row/3, (col-25)/2-5, "Enter number of players : ");
            mvscanw(row/3, col/2+9, "%d", &players);
            clear();
            refresh();
        }
        int getPlayerCount()
        {
            return players;
        }
        void showAllScore()
        {
            int i, row, col, response;
            initscr();
            getmaxyx(stdscr, row, col);
            mvprintw(row/3-3, (col-13)/2, "Current Score");
```

```cpp
        for(int i=0; i<players; i++)
        {
            mvprintw(row/3+(2*i), (col-25)/2-5, "%s : %d", playerName[i].c_str(), points[i]);
            refresh();
        }
        getch();
        clear();
        refresh();
    }
    void finalScreen()
    {
        int i, row, col, response;
        initscr();
        getmaxyx(stdscr, row, col);
        mvprintw(row/3-3, (col-15)/2, "Final Standings");
        for(int i=0; i<players; i++)
        {
            mvprintw(row/3+(2*i), (col-25)/2-5, "%s : %d", playerName[i].c_str(), points[i]);
            refresh();
        }
        int large=points[0], winner;
        for(i=1; i<players; i++)
        {
            if(points[i]>large)
            {
                large=points[i];
                winner=i;
            }
        }
        mvprintw(row/3+20, (col-6)/2, "Winner");
        mvprintw(row/3+22, (col-playerName[winner].size())/2, "%s", playerName[winner].c_str());
        getch();
        clear();
        refresh();
    }
};

int main()
{
    display disp;
    int response=1;
    while(response)
    {
        response=disp.splashScreen();
        if(response==1)
        {
            newGame game;
            game.playerCount();
            //int num=game.getPlayerCount();
            game.getPlayerName(game.getPlayerCount());
            for(int i=0; i<5; i++)
            {
                game.updateRound();
                ifstream fin;
                fin.open(questionsFileList[i]);
                game.displayRound(i);
                int row, col;
                initscr();
                getmaxyx(stdscr, row, col);
                for(int j=0; j<game.getPlayerCount(); j++)
                {
                    string ques, a, b, c, d, correctOption, givenOption;
                    getline(fin, ques);
                    getline(fin, a);
```

```cpp
                            getline(fin, b);
                            getline(fin, c);
                            getline(fin, d);
                            getline(fin, correctOption);
                            //cout<<ques<<endl<<a<<endl<<b<<endl<<c<<endl<<d<<endl<<correctOption<<endl;
                            for(int k=j, m=0; m<game.getPlayerCount()-1; k++, m++)
                            {
                                clear();
                                k=k%game.getPlayerCount();
                                mvprintw(row/5, (col-9)/2, "Player : %d", k+1);
                                mvprintw(row/5+2, (col-game.showPlayer(k+1).size())/2, "%s", game.showPlayer(k+1).c_str());
                                mvprintw(row/3, (col-ques.size())/2, "%s", ques.c_str());
                                mvprintw(row/3+4, col/3, "%s", a.c_str());
                                mvprintw(row/3+4, 2*(col/3), "%s", b.c_str());
                                mvprintw(row/3+8, col/3, "%s", c.c_str());
                                mvprintw(row/3+8, 2*(col/3), "%s", d.c_str());
                                mvprintw(row/3+14, col/2-1, " ");
                                givenOption = getstring();
                                //cout<<givenOption<<endl;
                                refresh();
                                if(givenOption[0]=='P')
                                    continue;

                                if(givenOption[0]==correctOption[0])
                                {
                                    clear();
                                    mvprintw(row/3, (col-14)/2, "Correct answer");
                                    refresh();
                                    game.scoreUpdateDirect(k, true);
                                    getch();
                                    break;
                                }
                                else{
                                    clear();
                                    mvprintw(row/3, (col-12)/2, "Wrong answer");
                                    mvprintw(row/3+3, (col-correctOption.size())/2-11, "The Correct answer is %s", correctOption.c_str());
                                    refresh();
                                    game.scoreUpdateDirect(k, false);
                                    getch();
                                    break;
                                }
                                refresh();
                            }
                        }
                        fin.close();
                        clear();
                        game.showAllScore();
                        refresh();
                    }
                    game.finalScreen();
                }

                if(response==2)
                    disp.help();
                if(response==3)
                    response=disp.exit();
            }
            endwin();
            return 0;
        }
```

****