

CSCI2120 Introduction to Software Engineering

Assignment 2

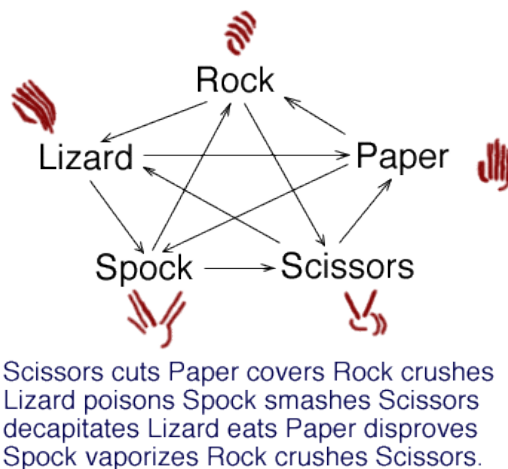
Deadline: 20th Mar 2016 (Sun) 23:59

Objective

The objective of this assignment is to practice Object-Oriented Programming (OOP) in C++ using inheritance.

Task

In this assignment, your task is to write a game to allow two players (a human player or a computer player) to play a game called "rock-paper-scissors-Spock-lizard" invented by Sam Kass and Karen Bryla. The game rule is shown as follows:



(Image adapted from <http://www.samkass.com/theories/RPSSL.html>)

Your program consists of the following steps.

Step 1: Display the Game Menu

When your program starts, it first displays a menu with the game title and asks the player to press '1' for "Player vs. Computer", '2' for "Player vs. Player", or '3' for "Exit". All other inputs are ignored. For example, if the user presses 'q' or '4', nothing will happen. Please use `_getch()` to get the input, which will not be echoed on the screen. Your program has to include `<conio.h>` to use this function. An example of using `_getch()` is shown as follows:

```
char c;  
c = _getch();  
if(c == 'a')  
    cout << "You entered the letter a" << endl;
```

After receiving a valid input, clear the screen by using `system("cls")`. If the player presses '1' or '2', go to Step 2. If the player presses '3', exit the program.

Step 2: Select the Number of Rounds

Your program displays another menu for selecting the number of rounds, which is either three or five. The player should press '1' for "3 Rounds" or '2' for "5 Rounds". All other inputs are ignored. Please use `_getch()` to get the input. After receiving a valid input, clear the screen and go to Step 3.

Step 3: Ask for the Player Information

Your program asks the human player(s) to input the name(s) and the winning statement(s), which may contain space characters. (Hint: use `getline(cin, str)` to read the input, where `str` is a string variable. You have to type `#include<iostream>` and `#include<string>` in the beginning of the code. After using `getline(...)`, please avoid using `"cin >>"` in your program as they are incompatible with each other.)

- If the player chose "Player vs. Computer", Player 1 will be a human player and Player 2 will be a computer player, whose name and winning statement are "Computer" and "You cannot beat a computer!" respectively. The name and the winning statement of Player 1 must contain at least one alphabet character and the name cannot be "Computer" (case sensitive). If the name (or the winning statement) is invalid, ask the player to input the name (or the winning statement) again. After receiving a valid name and a valid winning statement for Player 1, clear the screen and go to Step 4.
- If the player chose "Player vs. Player", both Player 1 and Player 2 are human players. The names and the winning statements of both Player 1 and Player 2 must contain at least one alphabet character and Player 2 cannot have the same name (case sensitive) as Player 1 (in this case, the program only needs to ask Player 2 to input the name again). In this mode, the name "Computer" can be used. If the name (or the winning statement) is invalid, ask the player to input the name (or the winning statement) again. After receiving a valid name and a valid winning statement for Player 1, clear the screen and then ask Player 2 for input. After receiving a valid name and a valid winning statement for Player 2, clear the screen and go to Step 4.

Step 4: Start the Game

For each round:

1. Display the round number (1, 2, 3, 4, or 5).
2. Ask the players to choose a gesture (rock, paper, scissors, Spock, or lizard).
 - If it is a human player, ask him/her to press '1' for rock, '2' for paper, '3' for scissors, '4' for Spock, or '5' for lizard. All other inputs are ignored. Please use `_getch()` to get the input. After receiving a valid input, clear the screen and go on to the next player.
 - If it is a computer player, randomly generate a gesture for it. Your program does not need to display anything on the screen for this. (Hint: you can use `srand(time(0))` to initialize the random number sequence [execute once only in the program], and then each time a random number is needed, use `rand().`)

3. Display the gestures chosen by both players.
4. Determine who wins this round:
 - If it is a draw round:
 - i. Display an appropriate message and ask the player to press any key to continue (use `_getch()`).
 - ii. Repeat this round (without increasing the round number).
 - If there is a winner in this round:
 - i. Display the name of the winner of this round.
 - ii. Display how many rounds both players win and ask the player to press any key to continue (use `_getch()`).
 - iii. If the player wins the whole game, i.e., the player wins 2 out of 3 rounds, or wins 3 out of 5 rounds:
 - a. Display the name and the winning statement of the winner.
 - b. Ask the player whether he/she wants to play again (press 'y') or exit to the main menu (press 'n'). All other inputs are ignored. Please use `_getch()` to get the input.
 - If the player chose to play again, clear the screen and repeat Step 4.
 - If the player chose to exit, clear the screen and go to Step 1.

Additional Requirements

1. You are required to implement this assignment in OOP style with the principles of encapsulation, information hiding, inheritance, and dynamic binding.
2. At least four classes (**Player**, **HumanPlayer**, **ComputerPlayer**, and **Game**) should be involved in your project and your main function should not contain more than 30 lines of code.
3. For dynamic binding, please use pointers of **Player** as the member variables in the **Game** class.
4. There should be no memory leakage in your program. Especially when you start a new game with new instances of **Player**, the **Player** objects pointed by the **Player** pointers should be deleted at the end of the game.
5. The random seed in your program should be system time to ensure variance of random number.

Two sample runs are shown as follows (underlined characters are input from the user, and characters with (resp. without) strikethrough means input with (resp. without) echo on the screen):

Sample run 1:

Rock - paper -scissors - Spock - lizard

1. Player vs. Computer

2. Player vs. Player

3. Exit

1

Please select the Number of Rounds:

1. Three rounds

2. Five rounds

2

Please input your name:

123

Invalid player name! Your name should contain at least one alphabet.

Wizard

Please input your winning statement:

I know what you are thinking about!

Round 1

Wizard, please choose your gesture:

1. rock

2. paper

3. scissors

4. Spock

5. lizard

1

Wizard, your gesture is rock.

Computer, your gesture is paper.

Computer wins in this round!

Wizard wins 0 round(s).

Computer wins 1 round(s).

Press any key to continue...

Round 2

Wizard, please choose your gesture:

1. rock

2. paper

3. scissors

4. Spock

5. lizard

3

Wizard, your gesture is scissors.

Computer, your gesture is Rock.

Computer wins in this round!

Wizard wins 0 round(s).

Computer wins 2 round(s).

Press any key to continue...

Round 3

Wizard, please choose your gesture:

1. rock
2. paper
3. scissors
4. Spock
5. lizard

3

Wizard, your gesture is scissors.

Computer, your gesture is Spock.

Computer wins in this round!

Wizard wins 0 round(s).

Computer wins 3 round(s).

Computer: You cannot beat a computer!

Would you like to play again? (y/n)

n

Rock - paper -scissors - Spock - lizard

1. Player vs. Computer
2. Player vs. Player
3. Exit

3

Sample run 2:

Rock - paper -scissors - Spock - lizard

1. Player vs. Computer
2. Player vs. Player
3. Exit

2

Please select the Number of Rounds:

1. Three rounds
2. Five rounds

1

Player 1, please input your name:

Bob

Please input your winning statement:

345...

Invalid winning statement! Your statement should contain at least one alphabet.
haha!

Player 2, please input your name:

Alice

Please input your winning statement:

I am so lucky!

Round 1

Bob, please choose your gesture:

1. rock
2. paper
3. scissors
4. Spock
5. lizard

1

Alice, please choose your gesture:

1. rock
2. paper
3. scissors
4. Spock
5. lizard

1

Bob, your gesture is rock.

Alice, your gesture is rock.

Draw! Press any key to repeat this round...

Round 1

Bob, please choose your gesture:

1. rock
2. paper
3. scissors
4. Spock
5. lizard

1

Alice, please choose your gesture:

1. rock
2. paper
3. scissors
4. Spock
5. lizard

2

Bob, your gesture is rock.

Alice, your gesture is paper.

Alice wins in this round!

Bob wins 0 round(s).

Alice wins 1 round(s).

Press any key to continue...

Round 2

Bob, please choose your gesture:

1. rock
2. paper
3. scissors
4. Spock
5. lizard

5

Alice, please choose your gesture:

1. rock
2. paper
3. scissors
4. Spock
5. lizard

2

Bob, your gesture is lizard.

Alice, your gesture is paper.

Bob wins in this round!

Bob wins 1 round(s).

Alice wins 1 round(s).

Press any key to continue...

Round 2

Bob, please choose your gesture:

1. rock
2. paper
3. scissors
4. Spock
5. lizard

4

Alice, please choose your gesture:

1. rock
2. paper
3. scissors
4. Spock
5. lizard

5

Bob, your gesture is Spock.

Alice, your gesture is lizard.

Alice wins in this round!

Bob wins 1 round(s).

Alice wins 2 round(s).

Alice: I am so lucky!

Would you like to play again? (y/n)

y

Round 1

Bob, please choose your gesture:

1. rock
2. paper
3. scissors
4. Spock
5. lizard

....

Declaration

As required by the University, please place the following declaration as the comment in the beginning of the code:

```
' CSCI2120 Introduction to Software Engineering
'
' Assignment <assignment_number>
'
' I declare that the assignment here submitted is original except for source
' material explicitly acknowledged. I also acknowledge that I am aware of
' University policy and regulations on honesty in academic work, and of the
' disciplinary guidelines and procedures applicable to breaches of such policy
' and regulations, as contained in the following University website:
' http://www.cuhk.edu.hk/policy/academichonesty/.
'
' Student Name: <your_name>
' Student ID: <your_student_ID>
' Date: <date>
```

Assessment Scheme

Correctness: 70%

Class Design: 20%

Program readability (e.g. indentation, comments, meaningful variable names, etc.): 10%

Please make sure that your program can compile successfully. Otherwise, you will get a very low score. Please also be reminded that your TA will read your code and can identify similar submissions easily. Consequences of confirmed plagiarism cases can be failing of the course and receiving a demerit.

Testing Platform

Your program will be tested in the following platform:

- Windows 7
- Visual Studio 2013

Submission

Please follow the following steps to submit your work.

- Step 1: If you only have one .cpp source file, please save your source file as <your_student_id>_assg2.cpp. (e.g. 1155012345_assg2.cpp). If you have more than one source files, please put them into a compressed file (zip, rar, or 7z) and name the compressed file as <your_student_id>_assg2.zip. (or .rar or .7z) (e.g. 1155012345_assg2.zip)
- Step 2: Go to CU eLearning (<http://elearn.cuhk.edu.hk/>) and login.
- Step 3: Go to “Assignment Submission” category.
- Step 4: Upload your file prepared in Step 1.

Resubmissions are allowed. But only the latest one will be counted. Late submissions within three days can only receive 70% of the marks. Late submissions more than three days will not be graded.