

COMP2045 Coursework 2022

Introduction

This coursework is worth **30% of the module mark**. It requires you to write a C++ program that solves the task described below. The deadline for this exercise is **23:55pm on Friday 29th April 2022**.

Read the entire document before beginning the exercise.

If you have any questions about this exercise, please ask in the Q&A forum on Moodle, after a lecture, in a lab, or during the advertised office hours. Do not post your program or parts of your program to Moodle as you are not allowed to share your coursework programs with other students. If any questions require this exercise to be clarified then this document will be updated and everyone will be notified via Moodle.

Task

Your task is to implement the game of Goofspiel in C++. Goofspiel is a card game for two players, using cards from a standard deck of cards. Each suit is ranked, i.e. ace is worth one point, 2 is two points, etc., jack 11, queen 12, and king 13 points.

One suit is singled out as the "prizes"; each of the remaining suits becomes a hand for one player, with one suit discarded. The prize cards are shuffled and placed between the players with one card turned up.

Play proceeds in a series of rounds. The players make sealed bids for the top (face up) prize by selecting a card from their hand (keeping their choice secret from their opponent). Once these cards are selected, they are simultaneously revealed, and the player making the highest bid takes the prize card. If there is a tie, the prize card value is equally split between the tied players. The cards used for bidding are discarded, and play continues with a new upturned prize card.

After 13 rounds, there are no remaining cards and the game ends. The player who has gained the highest sum of the prize cards wins the game.

See reference [1-3] for more details of Goofspiel and strategies for the game.

Your program should implement a Goofspiel game that the user plays against an automatic algorithm *repeatedly*. Your program should give a score (or scores) on how well the user has performed in the game playing.

Some other requirements of your C++ program include:

1. It must use C++ Object Oriented Programming (OOP) techniques, rather than C style sequential programming.
2. It must be an interactive game that the user plays against computer.
3. It must allow input from the user (by means of keyboard or mouse).
4. It must show results (win, lose, tie, or scores).

You can use either Linux X2Go or Windows Visual Studio 2019 to develop your game. Code developed on other platforms must compile on either X2Go or VS2019. Graphic interface is *NOT* compulsory.

Submission

You must submit a zip file that contains **a report** and **all your code** for this exercise. The code files should compile and run without warning and error messages on either X2Go or Visual Studio 2019.

In your report, you should clarify on what platform your program compiles and runs. You should explain the details of your program, include the hierarchy of classes used in your game and the strategy of Goofspiel - it is deterministic, random, hybrid, or adaptive (See [3] for a classification of Goofspiel strategies). You should also evaluate the performance of the strategy – how well it performs against some deterministic and random strategies and how well it performs against human players.

Late submissions: Late submissions will lose 2 percentage points per hour, rounded up to the next whole hour. No late submissions will be accepted more than 48 hours after the exercise deadline. If you have extenuating circumstances you must file them before the deadline.

Marking

The marking scheme will be as follows:

- *Testing (40% or 12 marks):* Your program should satisfy the hard requirements. As noted in the submission section, **if your program does not compile then you will lose all testing marks.**
 - Correctness. The program always gives correct output according to the rule of game.
 - Robustness. The program handles some incorrect user inputs.
 - No runtime error.
 - No memory leak.
- *Satisfaction of constraints (20% or 6 marks):*
 - Does the program use C++ object oriented techniques correctly?
 - Is the program well formatted and does it contain appropriate comments?
 - Does the program contain an appropriate interface?
- *Quality of the algorithm (20% or 6 marks):*
 - Does your algorithm perform better than the deterministic and random strategies?
 - Does your algorithm perform well competing against human players in repeated game playing?
- *Report (20% or 6 marks):*
 - Appropriate description of the program and algorithm.
 - UML expression of classes involved
 - Appropriate evaluation of the algorithm

Plagiarism

You should complete this coursework on your own. Anyone suspected of plagiarism will be investigated and punished in accordance with the university policy on plagiarism (see your student handbook and the University Quality Manual). This may include a mark of zero for this coursework.

You should write the source code required for this assignment yourself. You must not copy code from Internet. You must not copy or share source code with other students. You must not work together on your solution. You can informally talk about higher-level ideas but not to a level of detail that would allow you all to create the same source code.

Remember, it is quite easy for experienced lecturers to spot plagiarism in source code. We also have automated tools that can help us identify shared code, even with modifications designed to hide copying. If you are having problems you should ask questions rather than plagiarize. If you are not able to complete the exercise then you should still submit your incomplete program as that will still get you some of the marks for the parts you have done (but make sure your incomplete solution compiles and partially runs!).

Reference

[1] <https://en.wikipedia.org/wiki/Goofspiel>

[2] <http://46.101.185.98:8001/introduction.html>

[3] Mark Grimes and Moshe Dror (2013) Observations on strategies for Goofspiel, IEEE Conference on Computational Intelligence in Games.