COMP41670: Software Engineering

2024 / 2025

Individual Project

Dr Avishek Nag

UCD School of Computer Science,
University College Dublin,
Belfield, Dublin 4.

avishek.nag@ucd.ie

Introduction

The assignment solution should be submitted using Brightspace via the submission links. Assignment grades and feedback will be available in Brightspace.

All students should be familiar with the UCD Late Submission Policy.

All students should be familiar with the UCD Plagiarism Policy.

Patience Game

Read the rules of Patience at

https://simple.wikibooks.org/wiki/Card_Games/Patience

Play some Patience games!!!

https://g.co/kgs/YCdqP8

Implement and verify a Java program that allows a user to play Patience.

The program should have the following features:

- Displays the cards on the console and takes input from the keyboard (i.e. no GUI).
- The display lists the score and number of moves so far. The scoring should be as follows for successful moves:
    - 10 points per card moving from the uncovered pile to one of the suits
    - 20 points per card moving from one of the lanes to one of the suits
    - 5 points per card moving between lanes
- The uncovered pile is labelled "P", the card "lanes" are numbered 1-7, and suite piles are labelled D (diamonds), H (hearts), C (clubs) and S (spades).
- Prompt the user to enter a command.
- The user can enter the following commands:
    - Q = quit
    - D = uncover a new card from the draw pile
    - EITHER:
        - <label1><label2> = move one card from <label1> to <label2>. E.g. "P2" or "2D"
        - <label1><label2><number> = move <number> cards from <label1> to <label2>. E.g. "413".
    - OR:
        - <label1><label2> = move card(s) from <label1> to <label2> detecting the number of cards that need to move so as to make a legal move if at all possible. E.g. "P2" or "2D"
- Allow uppercase and lowercase commands.
- Display an error message and ask for a new command if an invalid command is entered.
- Display an error message and ask for a new command if a play cannot be made.
- Display a game over message if all cards are in their correct suites.

Submit your program as per the instructions in the Submission section below.

Submission

For all assignments, submit a zip file named with your student number containing following items.

- A report
    - Name,
    - Your self-assessment checklist that lists for the FINAL CODE all of the features with a comment as to whether the feature is fully working, partially working, or not working. Explain partially working features in more detail.
- A video (max. 5 minutes) in .mp4 format of a screen recording with voice over. The video should:
    - Show the working features
    - Explain the structure of your code
    - Explain how you tested the code
- A directory containing the source code.
- A directory containing an executable JAR file for the program (see Brightspace – Learning Materials - About the Tools for more information on creating the JAR).

Grading Scheme

Functionality is worth 80% of the grade.

Code quality is worth 20% of the grade.

Both are marked according to the scoresheets below. The numerical value is converted to a grade using the Alternative Linear Conversion Grade Scale 40% Pass.

https://www.ucd.ie/students/exams/gradingandremediation/understandinggrades/

Functionality

Functionality is marked as the percentage of features working correctly. Each feature is scored by its number of points, which indicates the difficulty of implementing the feature.

Feature	Points	Comment
Display cards and labels	4	
Prompt for user input	1	
Accept user input	1	
Error message for invalid commands	2	
Error message for invalid play	2	
Draw card from pile working	4	
Move card working	6	
Game over working	2	
Quit working	2	
TOTAL	24	

Programs are marked against the functionality described in this document. It is OK to include extra functionality to your game in addition to the items listed above. However, any extra functionality should be in addition to the functionality listed above, not instead of. For example, you might choose to add a feature whereby the player can click icons using the mouse. That's OK but you should also allow the user to type in the commands as specified.

Code Quality

Code quality is marked according to the quality checklist below.

Quality Check	Points	Comment
Is the code readable?	2	
Is the project well structured into classes?	2	
Does each class model a logical entity?	2	
Does each method perform a single clearly defined function?	2	
Are appropriate data structures used?	2	
Is scope minimised?	2	
Is data single source?	2	
Is inheritance used wisely?	2	
Are naming conventions used consistently?	2	
Are constants well used?	2	
Are names clear and meaningful?	2	
Are comments useful?	2	
Is the testing approach shown in the video robust?	2	
TOTAL	26	