

Final individual coursework Q&A

1. Is it feasible to implement a 2-player version of Risk?

Yes, you could have two players where a person plays against a computer. This will limit the scope of the project according to the available time, which is one of the four duties explained in the brief (see Introduction section).

2. What is the minimum number of regions and countries that needs to be included to maintain a balanced and enjoyable game? Would focusing on a single continent be sufficient? Any limitations

There are two aspects here, the first one is that your implementation should be compatible with the map files of the Risk "Domination" version. Hence, you should use those map files instead of creating your own. You can find those map files in the map directory after downloading the game on your machine (see The Problem section). The second aspect is related to the startup phase, where the user should define a map (e.g. selecting one of the available maps). After that, all the countries in the defined map are randomly assigned to the players.

3. What are the essential game phases (e.g., setup, attack, fortify) that must be implemented?

As explained in The Game section in the brief, you have five phases to consider, the startup phase and three turn-based play phases: The reinforcement phase, the attack phase, and the fortifications phase. Lastly, the End game phase, where you end the game whenever one of the players owns all the countries on the map.

4. do you recommend any design pattern for the implementation? Eg Abstract Factory pattern be applied to manage the creation of game objects (e.g., players, countries)?

We expect your code to include many of the Java features presented during the module. Hence, you are free to choose any design patterns from the given materials that you find suitable.

5. what programming paradigm do you expect us to follow ? OOP? functional ?

Both programming paradigms are covered by the materials, and you can use whichever is suitable for the design of your code.

6. Which core algorithms are necessary for basic game functionality (e.g., turn handling, combat resolution), and how can they be simplified?

This coursework involves the building of a challengingly extensive Java program, this includes the formulation all the requirements. To better understand the requirements, we highly recommend playing the game and reading its guidelines.

7. What features constitute the minimum viable product (MVP) for this project? What are the must-have elements versus nice-to-have enhancements?

The coursework description is purposely incomplete and it's your duty to elicit all the details. Perhaps playing the game would help you with constituting the minimum features of the game.

8. Is a graphical user interface (GUI) necessary for the MVP, or would a text-based interface be sufficient for demonstrating the core game mechanics?

Using command-line is sufficient. You don't have to build any GUI.

9. What are the primary learning objectives for this project (e.g., applying design patterns, understanding game development basics, algorithms...)? How can the project be structured to maximize these learning outcomes?

The learning objectives of this coursework align with the Module learning outcomes that could be found [here](#).

10. What minimal set of tests should be conducted to ensure the game functions correctly with the reduced scope? Can you provide a list of potential scenarios you want us to ensure is bulletproof?

We expect you to have sufficient unit testing that covers all main features in your implementation. For instance, you could have several test cases for the input provided by a player in each phase. You could also have a test case for ending the game.

11. Are there any restrictions, minimum requirements or guidelines that need to be considered when selecting and implementing design patterns and algorithms for this project?

The coursework description is purposely incomplete and it's your duty to elicit all the details. Perhaps playing the game would help you with formulating all the requirements.

12. What libraries can be used for this project, and what are their limitations, if any? Can we use freely Spring or Guava or shall we do DI at hand?

You can use either Spring or Guava but not any other library.

13. What are the essential components of project documentation (e.g., code comments, user manual) needed for this project?

Your report should follow the given proforma, which can be downloaded from [here](#). It should also explain how to run your code and play the game.

14. Is there any code base to start or shall we start from scratch ?

In this coursework, no code base is given and you should start from scratch.