AntiPlague Coronavirus Game

Write a program that is a game like "Plague Inc.". A world map will be presented during the game and divided into countries (select a minimum of 10 countries). This map has various ways of travelling between countries (airlines, buses, boats, trains, cars, etc.). Travelling methods are optional, but there must be at least 3 (not all must be implemented between all countries, but at least 2 travelling methods). The game is to prevent the infection of all people in all countries with a virus.

Each country has different criteria for hanging each of the transport routes between countries (criteria according to your creativity - e.g. according to the number of infected). Each of the criteria must be achievable and be able to be modified with purchased upgrades. You must create at least 9 upgrades (the selection and functionality depend on your imagination).

Upgrades can be purchased for points collected for anyone saved from infection or cured. The virus starts in the selected country at the beginning of the program (randomly or by the user). The manner, speed, and spread effects depend on the difficulty level and are implemented at your discretion.

The game ends when we protect the entire world population and stop the infection or when everyone is infected. Of course, progressive virus infection and its spread worldwide should also be implemented.

A fully functional graphical interface should be provided. The command line console (*CLI*) can only be used to help, but no user interaction with the program can occur there.

After starting, the program should display the main menu consisting of the options:

- *New Game*
- *High Scores*
- *Exit*

After starting a new game, the player will be asked in a separate window about the game difficulty level (at least 3 levels - the implementation of difficulty levels depends on you). After selecting the difficulty level, the game window is displayed in the new window, and the time counter starts (it is worth noting that the time counter, like the virus behaviour and others, must be ***implemented in separate threads***). During the game must be visible points and a time counter, which are constantly updated.

The game is played according to the rules mentioned above. It should be possible to interrupt the game at any time through the ***compound keyboard shortcut*** - ***Ctrl+Shift+Q***, which will return you to the main menu.

Transport connections between countries should be visualized using simple animations (e.g. for air transport, animation of an aeroplane icon moving along the path). The project should use graphic files and try to create a nice visual effect for the entire application, taking into account the appearance of windows such as *Menu*, *High Score* and *main game window.*

After finishing the game, a player is asked for his name in the new window, under which he will be saved in the high scores.

High scores are calculated based on the time, obtained effects and difficulty level (any implementation). You should save the ranking, so you do not lose saved records after closing the application. You must use the interface ***Serializable***.

After selecting the *High Score* option from the main menu, it is displayed to the user. There may be a relatively large number of saved results, so you should take care of scrollbars in case it does not fit in a window of a reasonable size.

Hints:

- Take care of exceptions in the program. If any occurs, display its message to the user.
- High Scores list must be implemented using the ***JList*** component.
- Take care of the appearance of the application
- You should take care of the scalability of application windows
- Countries can be implemented using buttons with the appropriate graphics, but you can also design your component
- Not all windows need to be implemented via the JFrame class. Dialogs can be used

The project is based on GUI material. Use Swing framework. You are NOT allowed to use JavaFX in this project.

The MVC design pattern must be used in the project.

Attention:

- ***In the case of receiving a project with significant deficiencies in implementation or a non-compiling solution, the result for such a project will be 0 points.***
- ***It is impossible to use WYSIWYG tools to generate windows (e.g. Window Builder).***
- ***Lack of knowledge of any line of code or plagiarism will result in obtaining 0 points for this project with the possibility of failing the entire subject.***
- ***Not only the practical and substantive correctness of the solution will be assessed, but also the optimality, quality and readability of the code written by you.***
- ***An important part of the project is the use of: inheritance, collections, interfaces or abstract classes, lambda expressions, Java Generics, additional functionalities or structures and other characteristic elements presented in the classes and lecture (but only in a natural way, nothing by force).***