GUI

Project

Write an application that implements a game in the style of "Pacman". Make the project using Swing.

In the game, every 5 seconds, enemies have a 25% chance to create upgrades (e.g. +50% movement speed, etc.) that the player can collect. Implement at least 5 different upgrades. After launching the application, it displays the main menu consisting of the following options:

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

When starting a new game, the player is asked for the size of the board. Implement at least 5 different boards of different sizes. After selecting the board size, the generated game board is displayed in a new window.

A fully functional graphical interface should be provided. *CLI* can only help the programmer, but there can be no interaction with the user.

During the game, the score counter, time counter, life counter, and other necessary elements of the graphical interface must be visible and will be constantly updated during the game.

You also need to use graphic files and create a cohesive look of the entire application, including all application windows.

Create stop motion animations for character movement and performed actions (e.g., walking animation, eating animation, etc.).
Implement animations yourself based on threads and images, not e.g. through GIF files.

All things related to the passage of time must be done using the Thread class (the `Timer`, `Executor` and other classes are not allowed). The careful and correct synchronization of all threads must be ensured. Different functionalities cannot be combined into one thread.

The game is played according to the rules mentioned above. It should be possible to interrupt the game at any time, which will return you to the main menu.

At the end of the game, the player is asked for the name under which they want to be saved in high scores. Provide high scores persistence using the ***Serializable*** interface.

After selecting the High Scores option from the main menu, it is displayed to the user. The ranking should be implemented using the ***JList*** component. Because the High Score window can be large, take care of the scrollbars.

Implement the application using good programming practices with complete event handling implemented by the delegated event handling model

Hints:

- Take care of exceptions in the application. If any occurs, display appropriate messages to the user.
- Take care of the scalability of application windows.
- Dialog windows can be used for smaller and informative windows.

Remember to adhere to the formal requirements regarding the rules for submitting and sending projects described in the course rules!

Attention:

- ***It is impossible to use WYSIWYG tools to generate windows (e.g. Window/Scene Builder).***

- ***The use of any AI tools in the implementation of the project is prohibited***

- ***Lack of knowledge of any line of code or high similarity of the submitted solution to another solution will result in failing the 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 and other elements discussed during the semester.)***