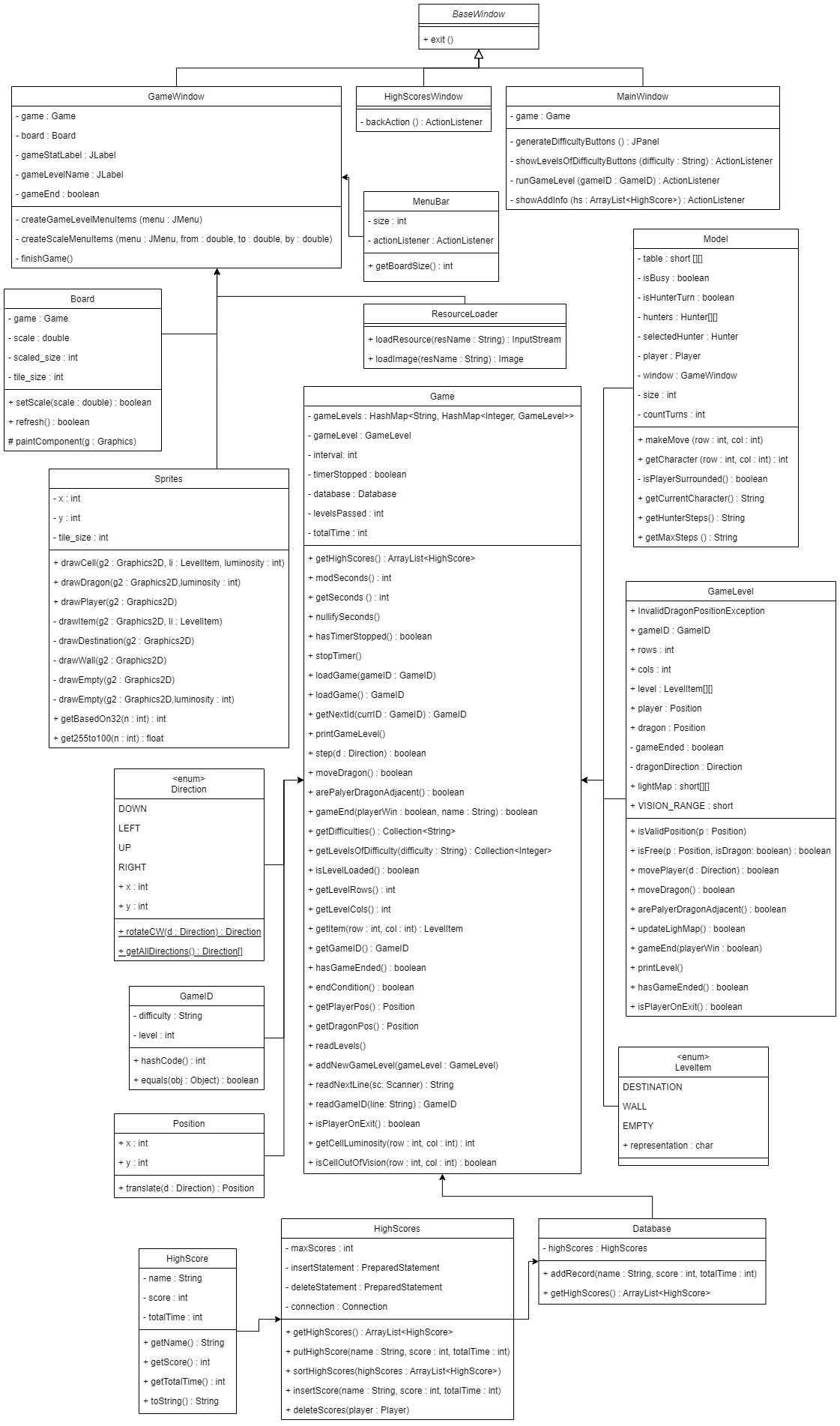
Documentation

# Description

## 3. Labyrinth

# Create the Labyrinth game, where objective of the player is to escape from this labyrinth. The player starts at the bottom left corner of the labyrinth. He has to get to the top right corner of the labyrinth as fast he can, avoiding a meeting with the evil dragon. The player can move only in four directions: left, right, up or down. There are several escape paths in all labyrinths. The dragon starts off from a randomly chosen position, and moves randomly in the labyrinth so that it choose a direction and goes in that direction until it reaches a wall. Then it chooses randomly a different direction. If the dragon gets to a neighboring field of the player, then the player dies. Because it is dark in the labyrinth, the player can see only the neighboring fields at a distance of 3 units. Record the number of how many labyrinths did the player solve, and if he loses his life, then save this number together with his name into the database. Create a menu item, which displays a highscore table of the players for the 10 best scores. Also, create a menu item which restarts the game. Take care that the player and the dragon cannot start off on walls.

# Class Diagram



# Method descriptions

I can highlight the “paintComponent” from the class “Board”. It draws the main interface using a JPanel, and all the components are 2D graphics written in the code.

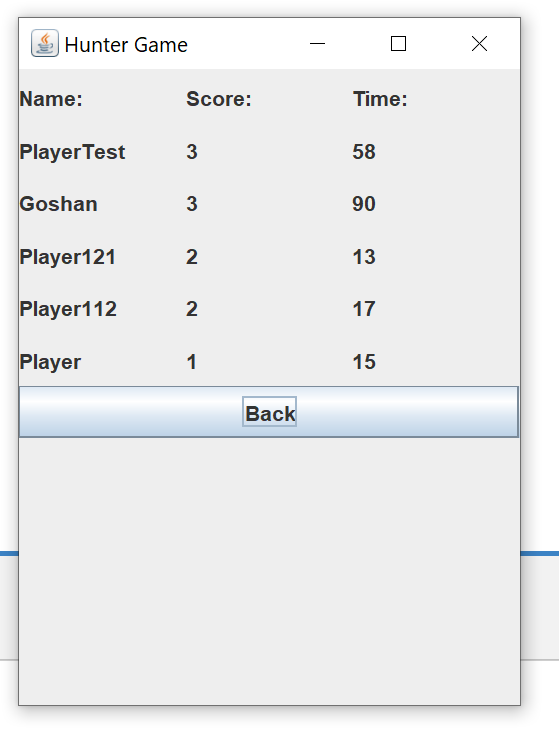
“GameWindow” also manages the logic of the dragon and the winning conditions, which are sent to separate threads in order to continue working while the main program runs.

# Connections between events and event handlers

The game has a lot of events and handlers. A good example is the player movement. It is done by the method “addKeyListener”, that reacts to the pressed key. Also the main menu contains buttons which are also implemented with event-handling.

# Testing

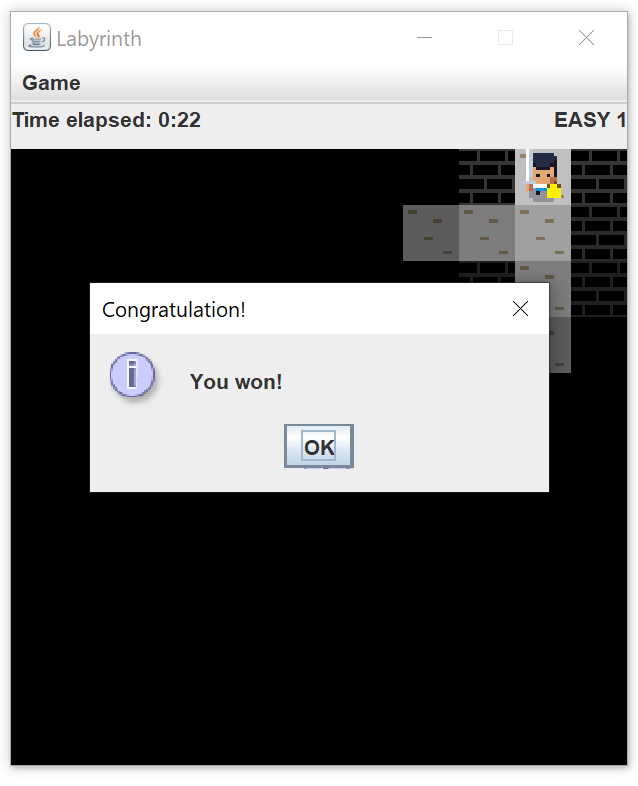
The menu works.



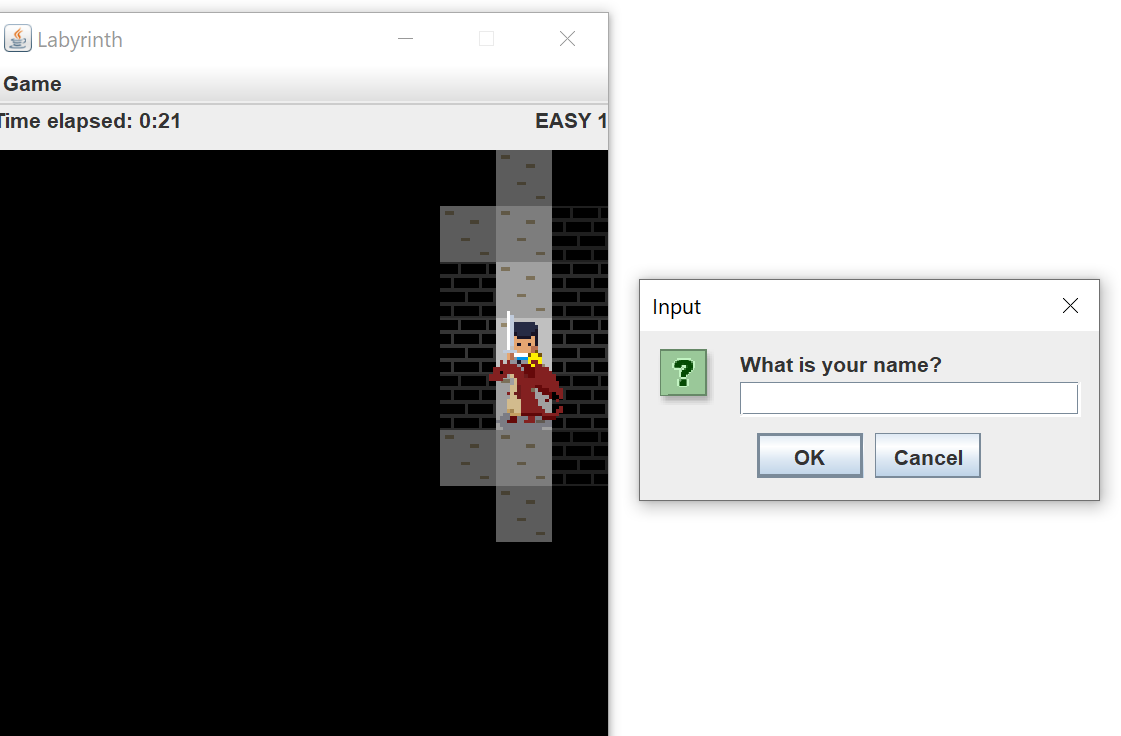
We can see our leader board.



The game runs.



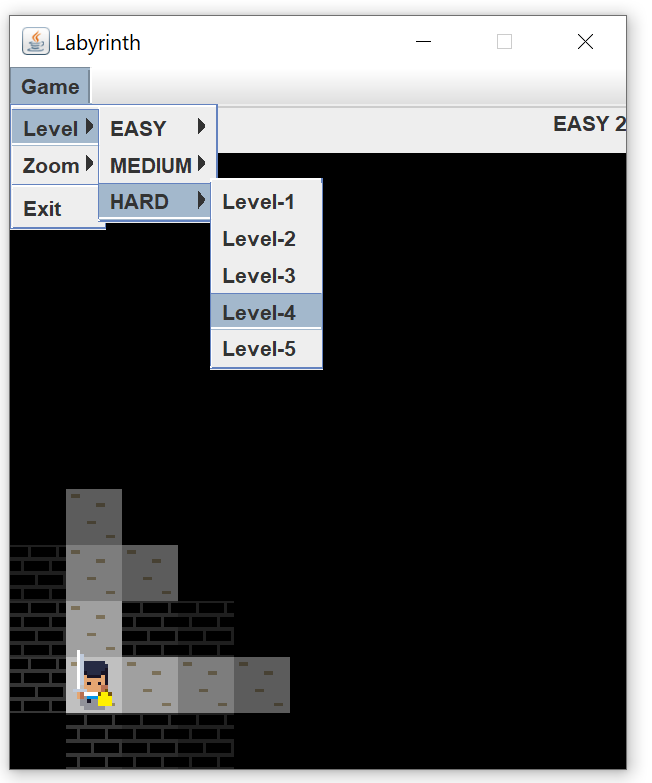
We can win.



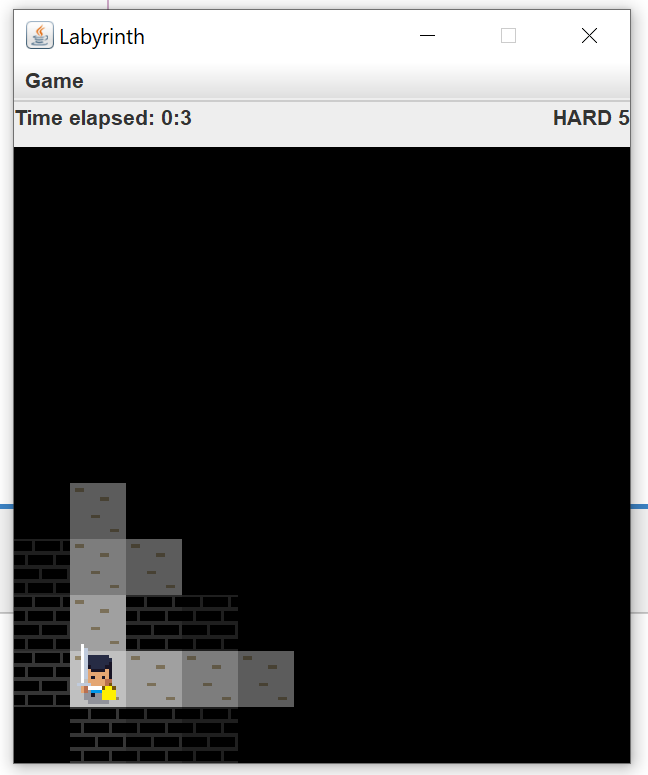
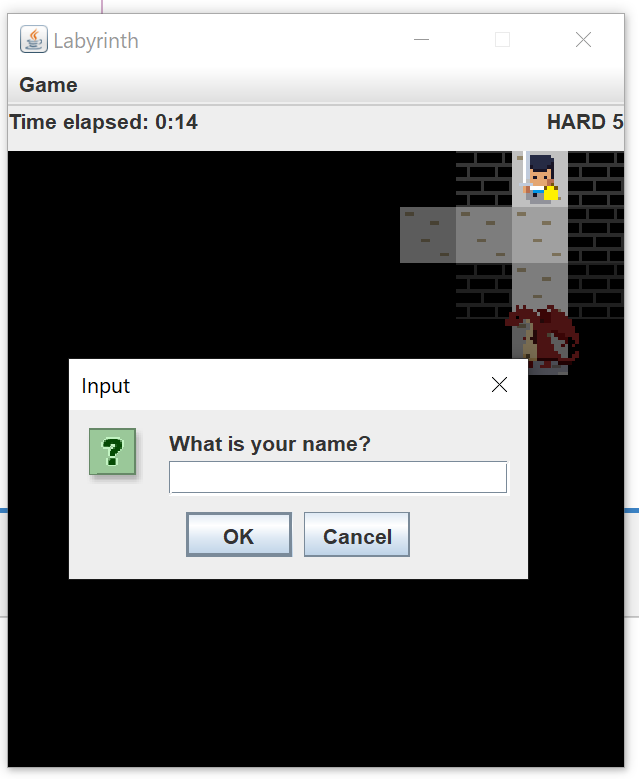
We can lose.



After winning the next level loads.



We can change the level from the bar.



After winning the last level, we complete the game.