

Documentation

1. Used technologies

For GUI was used Swing. For logging — log4j. For testing — JUnit 5, for creating executable jar — maven-assembly-plugin, for load/save — Java Serialization.

2. Logger

One of the conditions of the project was the ability to turn the logger off/on using arguments when starting the program. I've created the class MyLogger, which is a wrapper class for log4j2 logger. It has static field isLogOn, which equals false by default. To set it to true first argument in args should be equal to "-log". Also the level of logging can be changed in configuration file (log4j2.xml), it is set to "trace" by default.

3. JUnit 5

More unit tests will be added later, now there are only simple tests and parameterized tests.

4. GUI

Swing "no layout" is used. Game has one window, which is an instance of GameWindow class (Wrapper for JFrame), and one panel (an instance of the GamePanel class that inherits from JPanel). Also my own Button class is used instead of JButton.

There is also a constant SCALE, which allows you to change the scale of the game window and can be changed in the code without losing functionality, but can not be changed from the game. This feature will be added later with the settings menu.

5. Game loop

Game loop is separated to two threads: render and update. Render is responsible for drawing on the game panel and update — for updating state of the game objects.

6. Gamestates

All gamestates are implemented as Singletons.

7. Program structure

