

SPSE Ječná

Field 18–20-M/01 Information Technology

Ječná 30, 121 36, Prague 2

Tanks in the Labyrinth

Lazurko Danylo

Information and Communication Technologies

2025

Content

1. The goal of the work	2
2. Work description	2
2.1 Algorithm:	2
2.2 Mechanics:.....	2
3. System requirements.....	2
4. Fundamental structure	3
5. Testing data	3
6. User handbook	4
7. Conclusion	4
8. Sources.....	4

1. The goal of the work

The goal of the work is to create a simple window game for 2-4 players on one computer. Each player should be able to control their own tank. The tank shouldn't be able to go through walls of the labyrinth. Each tank should be able to fire up to 7 bullets. Bullets, on the other hand, should bounce off the walls and have a lifetime of 10 seconds. The game should implement a simple menu, a score system, and a simple interface for displaying controls of each tank. The game should be easily adjustable for the future changes.

2. Work description

This project is a simple local multiplayer tank game with up to four players. Each tank has a unique color, and the match takes place in a randomly generated 8x8 labyrinth. Players choose between three modes based on the number of participants.

2.1 Algorithm:

The maze is built using a recursive backtracking algorithm, ensuring a connected layout. After generation, a few random walls are removed to add alternate paths and reduce linearity. Each tile is marked as visited during the process, and walls are removed between neighboring tiles to create the maze structure.

2.2 Mechanics:

Tanks can move in four directions and shoot bullets that travel until they hit something. If a tank is hit, it's eliminated. The last tank standing wins the round, and the player's score increases. Scores are shown next to each tank and update after every game.

3. System requirements

The project is written on version Java SE 23. To run it, you must have the appropriate JDK installed. Apart from the JDK itself, no external libraries or frameworks are required to run the program - all functionality is handled using standard Java libraries. The program can be run both in a regular command line and in any development environment that supports Java, such as IntelliJ IDEA.

4. Fundamental structure

The program is designed using an object-oriented approach and is divided into several basic classes that communicate with each other. The Main class serves as the entry point for launching the application and initializes the necessary components. The MyFrame class is the main window. The MenuPanel is a panel where you can choose a mode. The GamePanel is a panel with all controls, score, and the labyrinth itself.

MazePanel is the most significant class in this project. This panel contains all tanks, bullets, walls and animations. Tank class is a representation of the tanks themselves. Each tank has an X and Y coordinate and the angle, which are used to move the tank. Bullet class is used to make bullets, each bullet X and Y coordinate and the angle, which are used to move the bullet.

The main game logic —processing moves, checking win conditions, and managing player turns — is concentrated in the MazePanel class. Components communicate through methods that ensure the progression of the game, such as setMapGrid(), checkRoundEnd(), render(), drawBackground(), createObjects().

The program structure is designed to be clear, easily maintainable, and extensible in the future, for example, to support playing against a computer.

5. Testing data

The program can be tested manually through different gameplay scenarios that cover key mechanics such as movement, shooting, collisions, and win detection.

It's also important to observe how the game handles edge cases — for example, tanks attempting to move into walls, bullets hitting obstacles at various angles, or players spawning too close to each other. These tests help ensure stability and correctness during gameplay.

Basic functional behaviors can also be validated with unit tests, particularly for tank movement, bullet updates, and collision logic. Combined with manual playtesting in 2, 3 and 4 player modes, this provides confidence that the core systems work as expected.

6. User handbook

When the game starts the user will be met with a menu. There you can choose the number of players that will be playing. From now on the game will start. To check controls of each player you can press the “Controls” button on the left of the labyrinth. You can see your score at the bottom of the labyrinth.

The round ends when only one tank remains alive. Then the new round starts after 2.5 seconds.

To return back to the menu you have to press escape (Esc) button.

7. Conclusion

I enjoyed working on this project. There weren't any major problems during the development of the project. I've got to train making user interface and animate tanks. Also, I made an FPS system in this project and of course to train OOP.

8. Sources

1) Online. In: DendyTanchiki.ru – Игра Танки в лабиринте. [online].

[2025-05-27]. Available at: <https://dendytanchiki.ru/igra-tanki-v-labirinte/>

2) Online. In: YouTube – Java Game Development - Full Course. [online].

[2025-05-27]. Available at:

<https://www.youtube.com/watch?v=Kmgo00avvEw&t=12312s>

3) Online. In: YouTube – Java Tank Game Devlog (Playlist). [online].

[2025-05-27]. Available at:

https://www.youtube.com/watch?v=OdWEhPyp5_o&list=PLyt2v1LVXYS2SSQkm1bIn1FRsg3BPez06