

Game Manual – ThinkOutsideTheRoom

WISA

April 28, 2025

1 Introduction

ThinkOutsideTheRoom is a cooperative Java-based multiplayer game featuring a client-server architecture using UDP communication. This project showcases synchronized gameplay where multiple clients control different aspects (actions) of a shared game character.

2 Game Objective

Players collaborate to solve puzzles by controlling various game objects in a shared environment. Each player has specific controls for a character, such as moving, jumping, grabbing, or throwing. Effective teamwork is required to progress through levels and challenges.

3 Launching the Game

Build Step

```
./gradlew build
```

This command compiles your project and packages it into a runnable JAR file.

Run the Server

```
java -jar .\build\libs\Think_Outside_The_Room-0.0.1-ALPHA.jar server 8888
```

Launches the server on port 8888. You may replace '8888' with any desired port.

Run a Client

```
java -jar .\build\libs\Think_Outside_The_Room-0.0.1-ALPHA.jar client 25.36.51.25:9876 [Username]
```

Connects to the server at IP **IP-adress** on port **Port**.

Note: The **Username** parameter is optional; if omitted, a default name is used by the operationg system.

4 Controls

Depending on the role assigned, a player may only control a subset of available actions. Below are the default keybindings:

Action	Keybinding
Move Left / Right	arrow left / arrow right
Jump	arrow up
Grab	E
Throw	R (red arrow, which indicates the direction)

Note: press the F key to go into throwing mode, then press R to throw.

5 Game Mechanics

- Game sessions are created using the **CREATEGAME** command.
- Players join sessions via **JOINGAME**.
- Objects such as players, platforms, and doors are instantiated using **CREATEGO**.
- Concealed parameters (automatically appended) help identify messages:

S:ObjectId,S:GameSessionId,S:UsernameId,S:MessageId

- The game supports in-game chat (global, Lobby and whisper).
- Multiple clients can simultaneously control the same object.

6 Game Flow

1. Start the server.
2. Clients connect and choose a username.
3. A game session is created by writing a name and press **CREATEGAME**.
4. A game session is joined by selecting the lobby and press **JOIN GAME**
5. To start the level press **START Level**
6. To control a player select **Alfred** or **Gerald**

7 User Interface

The JavaFX-based GUI includes:

- A visual rendering of the game world
- A chat box for communication
- A sidebar listing all connected players

8 Technical Overview

- **Language:** Java
- **Build Tool:** Gradle
- **GUI:** JavaFX
- **Networking:** UDP, custom protocol
- **Type Prefixes:** S:, F:, etc. to indicate parameter types

9 Known Issues

- UDP is not guaranteed to be reliable; packets may be dropped.
- Shared control might introduce conflicts if not handled carefully.
- Duplicate or missing object IDs can cause undefined behavior.

10 Credits

- WISA – Game Design and Programming
- JavaFX – UI Framework
- Additional dependencies – see `build.gradle`

11 Exiting the Game

- Type `EXIT{REQUEST}[]` in the in-game console to leave.
- Alternatively, exit the JavaFX GUI window.