



User Manual – Režim Vytěžit

1. Objective of the Game

The goal of the game is to mine all the rocks on the game board and avoid dangerous tiles. Once both players have mined all the rocks, they advance to the next level. If either player loses all their lives, the game ends.

2. Number of Players

The game is designed for 2 players who play on the same keyboard.

3. Controls

Player 1: W, A, S, D – movement | B – mining

Player 2: arrow keys – movement | M – mining

4. Running the Game

The game is launched using the Main.java class. Before starting, you must have:

- JDK installed (version 22 or newer)
- JavaFX libraries on the correct --module-path
- A folder images inside src/main/resources containing all necessary images (Character, Gold, Blocks, Dangerous block)

5. Game Objects

K – mineable rock

Z – dangerous block (takes away a life)

X – impassable wall

[space] – walkable tile



6. End of Game

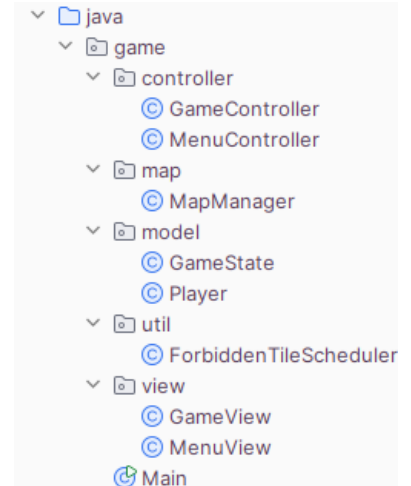
- After winning, the message “**You Won!**” appears
- After losing, the message “**Game Over**” appears

Technical Documentation – Režim Vytěžit

1. Project Structure

The project is organized into the following packages:

- **game.controller**: Contains game logic and level switching.
- **game.model**: Data classes representing players, game state, and tile types.
- **game.map**: Loads maps from JSON files and generates dynamic dangerous tiles.
- **game.util**: Helper tools like the timer for generating forbidden tiles.
- **game.view**: Renders the game using JavaFX and handles keyboard input.



2. Classes

GameController

Controls game flow. Includes:

- startGame(Stage) – game start or state load
- checkHazardCollision() – deducts life if standing on “Z”
- checkWin() – checks if the level is complete
- saveGame() and loadGame() – serializes/deserializes GameState

GameState

Holds information about both players, maps, current level, and remaining rocks.

Player

Represents a player: position, movement direction, number of lives.

MapManager

Loads maps from level1.json, level2.json. Adds dynamic forbidden tile (“Z”).

ForbiddenTileScheduler

Timer that adds a new forbidden tile every 5 seconds.

GameView

Renders the game: maps, players, lives. Uses images from *resources/images*.

MenuController, MenuView

Manages the start screen with game start button and win/lose messages.

JavaDoc

All major public methods and classes are documented using JavaDoc comments.

Serialize

The game state (GameState) is saved using ObjectOutputStream to a file named save.ser and loaded on game start.

3. Running the Project

The project requires JavaFX. Use --module-path and add --add-modules javafx.controls,javafx.fxml in VM options.

4. Resources

- maps: level1.json, level2.json – level designs.
- images: *Postavicka64.png*, *Zlato64.png*, *Blok64.png*, *Nebezpecny_blok64.png*.

5. Testing

The project includes unit, process and integration tests using JUnit and Mockito.

