SPŠE Ječná

IT

Ječná 517, 120 00 Nové Město

City Builder

Dominik Svoboda

IT

2025

City Builder Game Documentation

---

1. Project Goal

The goal of this project is to create a strategy city-building game in Java, where the player manages resources, builds buildings, and reacts to random events.
The key features of the game include:

- Placing and upgrading different types of buildings
- Managing materials and population growth
- Surviving and adapting to random events
- Real-time simulation with periodic updates
- Resource purchasing system

---

2. Game Description

2.1 Story/Logic

The game runs in real-time cycles (each representing a "day") and includes challenges such as storms, economic events, or robberies. The player must balance the growth of the city with the availability of food, money, and materials. If food runs out for too long, the city population will perish.

2.2 Mechanics

- **Building System**: Place buildings (houses, shops, power plants, warehouses) on a 20×20 grid.
- **Upgrading**: Buildings can be upgraded up to level 3, increasing their benefits.
- **Resources**: The player manages money, food, power, iron, concrete, and glass.
- **Events**: Random events (e.g., storms, treasure discoveries, robberies) influence gameplay.
- **Shops**: Buy materials when needed.

- **Game Loop**: Each "day", the game recalculates income, food production/consumption, triggers events, and checks for win/loss conditions.
- **Loss Condition**: If the population starves for 20 days, the game ends.

---

3. System Requirements

- **SDK:** Oracle OpenJDK 24
- **Libraries**: Standard Java libraries only.
- **Development Environment**: The game can be compiled and run using any Java-compatible IDE such as IntelliJ IDEA, Eclipse, or NetBeans.
- **Execution**: Run the `Main` class to start the application.

---

4. Basic Structure

The application is structured using object oriented principles:

- **`MainWindow`**: Initializes the game, handles game state switching, and manages save/load options.
- **`GameManager`**: Core logic and game state (resources, buildings, tiles, events).
- **`GameScreen`**: Main interface for interacting with the game grid and buildings.
- **`Tile`**: Represents one grid square that may contain a building.
- **`Building`**: Represents individual buildings with upgrade logic and icons.
- **`EventManager` + `GameEvent` subclasses**: Define and trigger random events like storms or economic booms.
- **`ResourceShop` / `BuildingShop`**: Dialogs for purchasing buildings and materials.
- **`StatusPanel`**: UI panel showing real-time game stats.

- `FileUtils`: Utility class for saving and loading the game.
- `BuildingTest`: Unit test for the building logic using JUnit.

---

5. Test Data

The program can be tested using JUnit tests and manually through the UI.

Sample Tests Include:

- Testing building creation and initial levels
- Validating correct level upgrades
- Ensuring resource limits are enforced during upgrades
- Verifying that population bonuses update correctly
- Checking event behavior like money changes from robberies or treasure

Manual testing includes simulating starvation, resource shortage and event triggers by letting the game run through its loop.

---

6. User Manual

- **Placing a Building**: Click an empty tile to open the shop and choose a building if affordable.
- **Upgrading a Building**: Click an existing building and confirm the upgrade.
- **Buying Materials**: Use the button on the right to open the resource shop and purchase materials.
- **Progression**: Every 5 seconds is a new "day". Resources are calculated, population is updated, and events may occur.
- **Losing the Game**: If food runs out and population starves for 20 days, the game ends.
- **Menu Options**: Use the top menu to start a new game, save, load, or exit.

---

7. Conclusion

The development of the City Builder game made me realize just how unskilled I was when I first started this project. In the first week, I didn't even want to push it to GitHub because I was embarrassed by it. Then I kind of gave up for the rest of the week. I watched some tutorials and went through my old projects with ChatGPT, who explained them to me. That left me with only this last week to finish.

I encountered a couple of problems that were usually fixable pretty quickly, but there was one I just couldn't solve, it was the ImageIcons for the upgraded buildings, so I asked ChatGPT to help fix it. It took me about three days to finish the first 500 lines, and I finally made the initial commit to GitHub.

These last three days of non-stop programming, when I was writing the final 500 lines, made me realize that I really need to start sooner and not leave everything until the last possible moment.

---

8. Sources

[1] Icons used in the game are sourced from **https://www.flaticon.com**.
[2] Some code logic was assisted by ChatGPT.