Pacman Game Documentation

By Mohamed Eldeeb

1. Introduction

Welcome to the Pacman Game documentation! This document provides information about the game's main features, gameplay, important design choices and the interaction with the game in general; Once clicking on the game, the player will be greeted by the main menu.

2. Main Menu

The main menu serves as the gateway to various features and options in the game. It lets the player choose from 3 given options.

Load Map

The "Load Map" option allows you to load a predefined map, providing different challenges and layouts for Pacman. And of course, start the game.

Create Map

With the "Create Map" option, you can design your custom playable map. Once clicking on this, the player will get a grid window where they can place coins/walls/empty spaces on the grid for different gaming experiences.

Scoreboard

Access the "Scoreboard" to view the highest all-time scores. See how your skills compare to other players in the Pacman community.

3. Gameplay

Pacman

Pacman moves in a tile-based system, requiring strategic planning to navigate through mazes and collect all the coins by moving with arrows. To win, the player has to be cautious of the ghosts that roam the game.

Ghosts

Ghosts move on a grid in a turn based system, the player needs to be cautious of getting too close to them to avoid dying.

Objective

The objective of the game is to collect all coins in the map without getting caught by the ghosts. Navigate through mazes, plan your moves carefully, and collect all the coins to win.

Class design

This technical documentation provides an overview of the classes in the Pacman game, explaining their structure, purpose, and key functionalities.

1. MainMenu Class

Description

The `MainMenu` class represents the main menu of the Pacman game. It extends `JFrame` and provides options to load predefined maps, create custom maps, and view the high scores.

Methods

- `MainMenu()` : Constructor method to initialize the main menu window.
- `createMainMenu()` : Method to create the main menu components, including buttons for loading maps, creating maps, and accessing the high scores.
- `createMenuButton(String buttonText)` : Method to create a styled button for the main menu.
- `main(String[] args)` : Main method to launch the main menu.

Functionality

- **Load Map**: Loads predefined maps for the game.
- **Create Map**: Opens the map builder to create custom playable maps.
- **Scoreboard**: Displays the highest all-time scores.

2. MapBuilder Class

Description

The `MapBuilder` class represents the map-building interface for creating custom playable maps. It extends `JFrame` and allows users to design maps by toggling tiles. This is the main class responsible for the map designer componenet of the game.

Methods

- `MapBuilder()` : Constructor method to initialize the map builder window.
- `mapSave()` : Method to serialize and save the created map to a file.
- `toggleTile(int row, int col)` : Method to toggle the state of a tile between different stats.
- `updateTileColor(int row, int col)` : Method to update the color of a tile based on its state.
- `setInitialTileColor(int row, int col)` : Method to set the initial color of a tile.
- `MapBuild()` : Static method to launch the map builder.

Functionality

- **Tile Toggle**: Allows users to toggle between different tile states.
- **Save Map**: Serializes and saves the created map to a file.

3. PacMan Class

Description

The `PacMan` class represents the main gameplay of the Pacman game. It extends `JFrame` and handles Pacman movement, ghost behavior, and game logic.

Methods

- `PacMan()` : Constructor method to initialize the Pacman game window.
- `handleKeyPress(KeyEvent e)` : Method to handle keyboard inputs for Pacman movement.
- `checkNearbyGhost()` : Method to check if Pacman is caught by nearby ghosts.
- `respawnPacMan()` : Method to respawn Pacman after losing a life.
- `checkWin()` : Method to check if the player has won the game.
- `countRemainingCoins()` : Method to count the remaining coins on the map.
- `checkCoinCollection()` : Method to check if Pacman collects a coin.
- `loadScaledImage(String path, int width, int height)` : Method to load and scale images for Pacman, ghosts, coins, and tiles.
- `moveSprite(int deltaX, int deltaY)` : Method to move the Pacman sprite.
- `isValidMove(int x, int y)` : Method to check if a move is valid.

Functionality

- **Pacman Movement**: Moves Pacman based on user input.
- **Ghost Movement**: Ghosts move randomly on the map.
- **Collision Detection**: Checks for collisions with ghosts and coins.
- **Win/Loss Conditions**: Determines win or loss conditions.

4. PacManMap Class

Description

The `PacManMap` class represents the game map and is responsible for loading the map from a file. It extends `JPanel` .

Methods

- `PacManMap()` : Constructor method to initialize the map panel.
- `loadMapFromFile()` : Static method to load the map from a serialized file.

Functionality

- **Map Loading**: Loads the game map from a serialized file.

5. Score Class

Description

The `Score` class handles score-related functionalities, including saving scores to a file, displaying top scores, and loading scores.

Methods

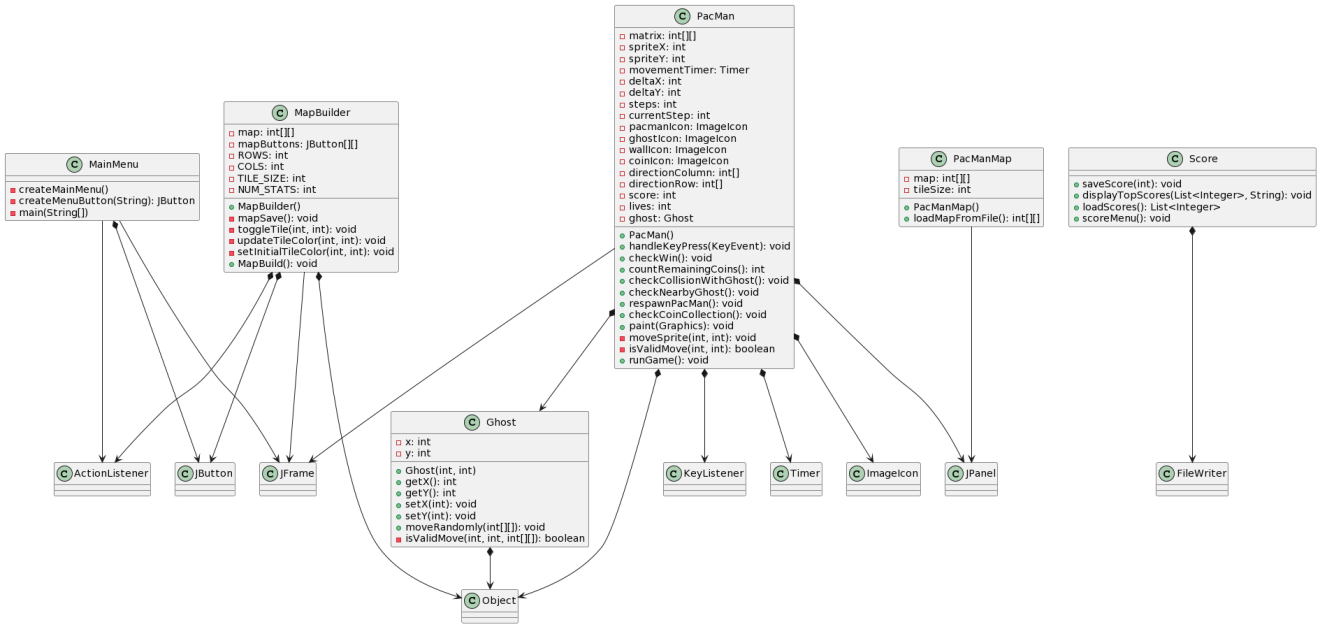
- `saveScore(int score)` : Method to save a score to the scoreboard file.
- `displayTopScores(List<Integer> scores, String title)` : Method to display the top scores in a JOptionPane.
- `loadScores()` : Method to load scores from the scoreboard file.
- `scoreMenu()` : Method to display the top scores in a menu.

Functionality

- **Score Saving**: Saves the player's score to the scoreboard.
- **Score Loading**: Loads scores from the scoreboard file.
- **Top Scores Display**: Displays the top scores in a menu.

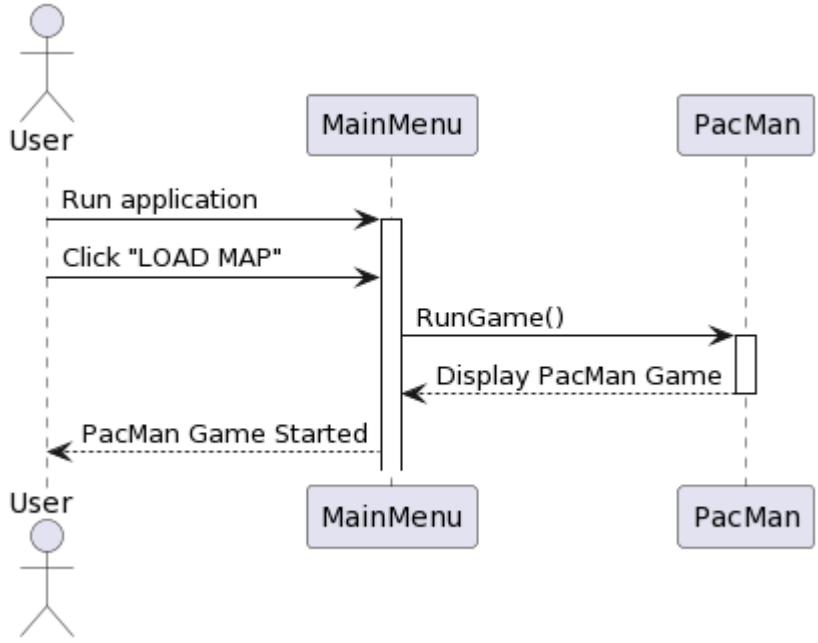
This technical documentation provides an overview of the key classes and their functionalities in the Pacman game. It serves as a reference for developers and contributors to understand the code structure and implement enhancements or modifications.

-

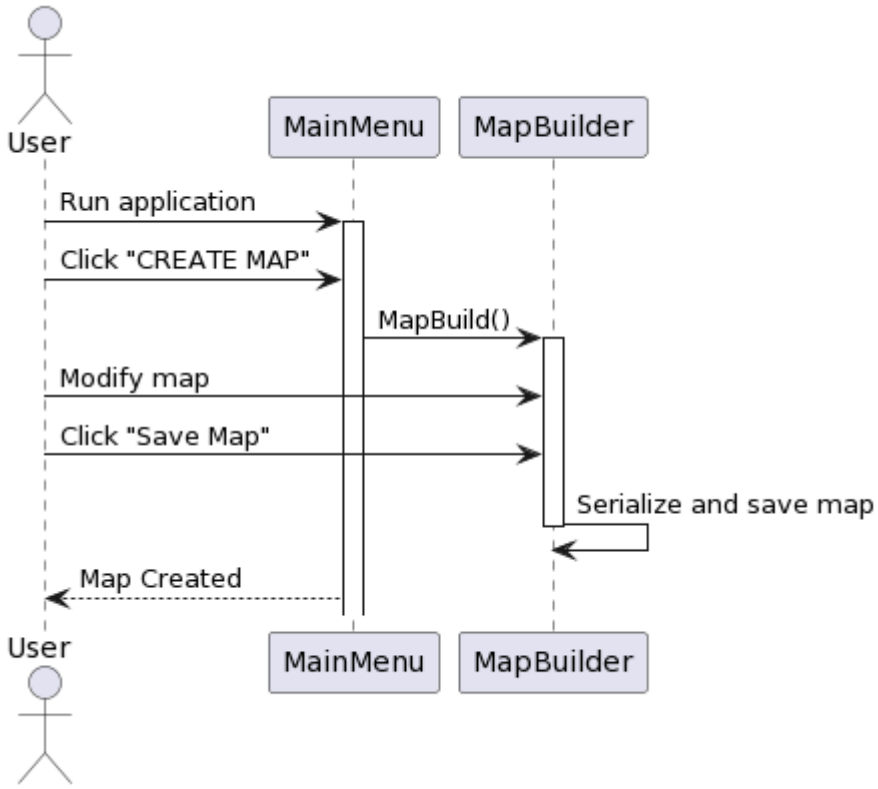


Class diagram for the software

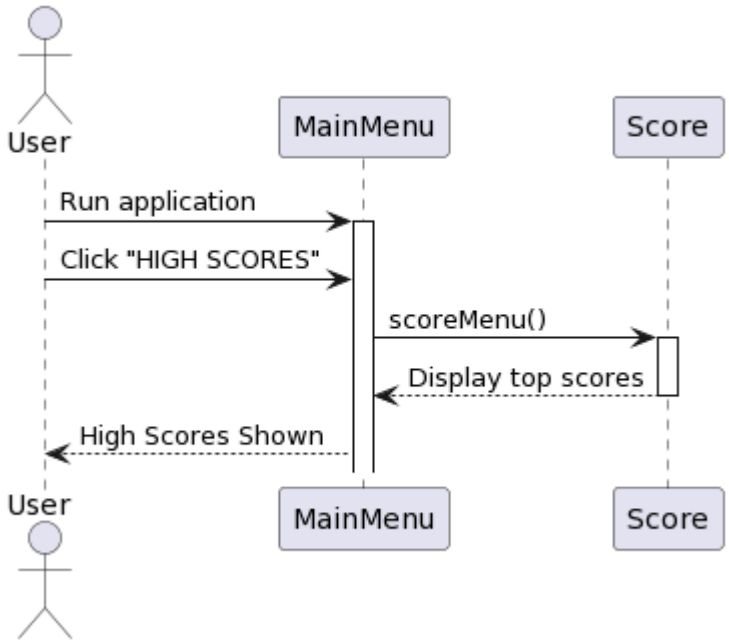
Sequence diagrams for different use cases



Starting the game



Creating a map



Loading scoreboard

Test cases

MainMenuTest

Class: `MainMenuTest`

Purpose: Tests the functionalities of the `MainMenu` class.

`testFrameSetup()`

Description: Ensures that the main menu frame is set up correctly with the expected title and default close operation.

Test Case: Verifies that the title of the frame is “Pacman Game” and the default close operation is `JFrame.EXIT_ON_CLOSE`.

`testButtonsAdded()`

Description: Checks if the main panel in the main menu frame contains the correct number of buttons.

Test Case: Asserts that the main panel has three components (assumed to be buttons).

MapBuilderTest

Class: `MapBuilderTest`

Purpose: Tests the functionalities of the `MapBuilder` class.

`testMapInitialization()`

Description: Verifies the initialization of the map in the `MapBuilder`.

Test Case: Checks if the map array is not null and has the correct dimensions.

`testToggleTileColorChange()`

Description: Tests the `toggleTile` method’s ability to change a tile’s state and color.

Test Case: Toggles a tile and checks if its color changes accordingly.

ScoreTest

Class: `ScoreTest`

Purpose: Tests the functionalities of the `Score` class.

`testSaveAndLoadScores()`

Description: Ensures that scores can be saved and loaded correctly.

Test Case: Saves a score, loads scores, and checks if the saved score is present in the loaded scores.

`testTopScoresCalculation()`

Description: Tests the calculation and sorting of top scores.

Test Case: Provides a list of scores, sorts them, and verifies if the top three scores are as expected.