Developer manual

for

MonopolyCMD

Version 1.0

This document provides instructions for developers on how to set up, build, and run the Monopoly game project. It assumes the reader is familiar with setting up a Python development environment and emphasizes project-specific details.

# 1 Development Platform

The Monopoly game project was developed and tested on **Windows 11**.

While the game may work on other platforms, this manual is specifically intended for Windows-based development and usage.

# 2 Development Tools

## 2.1 Initial Development:

* The project was initially developed using **PyCharm** Community Edition by JetBrains.
* PyCharm was used for its powerful debugging tools, project structure management, and seamless integration with Python.
* You can download PyCharm from <https://www.jetbrains.com/pycharm/download/>.

## 2.2 Later Development:

* During the later stages of development, the project was migrated to **GitHub Codespaces** for cloud-based coding and collaboration.
* Codespaces provided a lightweight, browser-accessible development environment with pre-configured settings, ensuring consistent project setup across team members.
* For more information on GitHub Codespaces, visit: <https://github.com/features/codespaces>.

# 3 Project Setup

## 3.1 Clone the Repository

To retrieve the project files, clone the Git repository:

git clone <https://github.com/754cac/COMP3211-MonopolyCMD.git>

# 4 Open the Project

* **For PyCharm**:

1. Launch PyCharm and select Open from the welcome screen.
2. Navigate to the directory where the project was cloned, and open the folder.
3. PyCharm will automatically detect the **main.py** file and other relevant files in the project structure.

* **For GitHub Codespaces**:

1. Open the repository in GitHub.
2. Click on the Code button and select Create Codespace on Main.
3. Wait for the Codespace to initialize. Once ready, the project files will automatically load in the online editor.
4. Use the built-in terminal in **Codespaces** to run and debug the project.

# 5 Running the Game

## 5.1 Launching the Game

To run the Monopoly game, follow these steps:

1. Open the main.py file in your development environment (PyCharm or Codespaces).
2. Run the game:

* In PyCharm, click the Run button in the top-right corner.
* In Codespaces, use the terminal and run the following command:

python main.py

1. The game will start, and you will see the command-line interface prompting you for player actions.

# 6  Debugging the Game

To debug the game:

* **In PyCharm:**

1. Set breakpoints in the desired locations within the code (e.g., in game.py or player.py for key logic).
2. Click the Debug button in PyCharm to launch the game in debugging mode.
3. Use PyCharm’s debugging tools to step through the code, inspect variables, and analyze program flow.

* **In Codespaces:**

1. Set breakpoints in the VS Code-like interface provided by Codespaces.
2. Use the built-in debugging tools to analyze the program.
3. Run the game in debugging mode using the terminal or the Run/Debug panel.

# 7 Key Files and Directories

Below is an overview of the key files and directories in the project:

* main.py: The entry point for running the game.
* model.py: Contains the logic for entering game or entering gameboard design.
* game.py: Contains the main game logic, including turn handling and game state management.
* player.py: Defines the Player class and handles player-specific logic such as money management and property ownership.
* gameboard.py: Manages the gameboard, including the definition of squares and their effects.
* functions.py: Defines the functions to be used in the Monopoly game and exported as a dictionary to be used in the game initilization
* vars.py: Defines all default player, gameboard parameters, dynamic working directories to be used in the game initilization
* tests/: Contains unit tests for the game logic. Run these tests to ensure functionality.
* README.md: A high-level overview of the project, including instructions for setting up and running the game.

# 8 Running Unit Tests

Unit tests have been written to validate the core game logic. To run the tests:

1. Open a terminal in your development environment (PyCharm or Codespaces).
2. Navigate to the project directory.
3. Run the following command:

python -m unittest discover Test\_functions.py

python -m unittest discover Test\_game.py

python -m unittest discover Test\_gameboard.py

python -m unittest discover Test\_player.py

python -m unittest discover Test\_vars.py

1. The test results will be displayed in the terminal. Ensure all tests pass before deploying any changes.