**Project: Monopoly Game**

**Created by: Aseel Ahmad  
Monopoly Game - README**

**Overview**

This project is a Monopoly game implementation developed in C++ for Ubuntu 22.04 LTS. It includes features such as player management, game board logic, property buying, rent payments, and more. The game can be run using both clang++ and g++ compilers, and test cases are provided to ensure the correct functionality of the game.

**Development Environment**

* **Operating System**: Ubuntu 22.04 LTS
* **Compilers**:
  + clang++ version 14
  + g++ version 11
* **Testing Framework**: doctest (included in the project)

**Build and Run Instructions**

To build and run the game or the test cases, use the following commands in your terminal:

1. **Build and run the game using clang++:**

make

This will compile the game with clang++ and start the Monopoly game.

1. **Build and run test cases using clang++:**

make test

This will compile and run test cases with clang++ using doctest.h for unit tests.

1. **Build and run the game using g++:**

make g++

This will compile the game with g++ and start the Monopoly game.

1. **Build and run test cases using g++:**

make testg++

This will compile and run test cases with g++ using doctest.h.

**Recommended Screen Resolution**

* **Best Experience**: Full HD (1920x1080)
* **Supported**: HD (1360x768) and other resolutions
* **Note**: Full HD resolution is recommended for the best visual experience.

**Game Rules**

**Rent Fees for Properties**

The rent fee depends on the number of houses or hotels on the property:

* **0 Houses**: 10% of the street price
* **1 House**: 20% of the street price
* **2 Houses**: 40% of the street price
* **3 Houses**: 80% of the street price
* **4 Houses**: 160% of the street price
* **Hotel**: 320% of the street price

**Upgrade/Build Costs**

* **Adding a house**: 50% of the property's initial price per house
* **Building a hotel**: Price of 4 houses + ₪100

**Property Sale Price**

When selling properties, the sale price is always 50% of the purchase price. This applies to:

* Streets
* Companies
* Railroads
* Houses and hotels

**Chance Cards**

Some examples of Chance cards include:

* **Advance to Go**: Collect ₪200
* **Bank pays you dividend**: Collect ₪50
* **Go back 3 spaces**
* **Go directly to Jail**: Do not pass Go, do not collect ₪200
* **Pay poor tax**: ₪15
* **Advance to nearest Utility or Railroad**: If owned, pay rent, otherwise buy it from the bank

**Community Chest**

Some examples of Community Chest cards include:

* **Bank error in your favor**: Collect ₪200
* **Income tax refund**: Collect ₪20
* **You inherit**: ₪100
* **Life insurance matures**: Collect ₪100

**Class and File Descriptions**

**Definitions.hpp**

This header file contains essential definitions and constants used throughout the game.

* **Key Features**:
  + Defines constants such as the maximum number of players
  + Contains enumerations for cell types and game states
  + Includes a Helper class for utility functions
* **Importance**:
  + Acts as a dictionary for common game terms
  + Centralizes important definitions for easy adjustments across the program

**GameCell Class**

This class represents a single space on the Monopoly board.

* **Key Features**:
  + Stores information about the cell type (e.g., street, railroad, chance)
  + Tracks ownership of properties
  + Manages the visual representation of the board cell
* **Importance**:
  + Forms the core of the game board
  + Handles the rules for each space and updates the game visually

**Player Class**

This class manages individual player data and actions.

* **Key Features**:
  + Stores player name, money, position on the board, and jail status
  + Manages "Get Out of Jail Free" cards
* **Importance**:
  + Represents each player and tracks their status throughout the game

**GameBoard Class**

This is the main class responsible for managing the game.

* **Key Features**:
  + Manages all GameCell and Player objects
  + Keeps track of game state, player actions, and logs game events
* **Importance**:
  + Acts as the central controller, enforcing game rules and tracking progress

**GameWindow Class**

This class handles the graphical user interface and user input.

* **Key Features**:
  + Creates and manages the game window size
  + Draws the game board, players, and other visual elements
  + Handles user input and updates the display accordingly
* **Importance**:
  + Provides the visual representation of the game and processes user interactions

**Monopoly.cpp (Main Game Logic)**

This file contains the entry point and main loop of the game.

* **Key Features**:
  + Initializes the game window and runs the game loop
* **Importance**:
  + Starts and maintains the game, updating the state and redrawing until the game ends

**Test Cases (TestMonopoly.cpp)**

Contains unit tests to ensure the game behaves as expected.

* **Key Features**:
  + Uses the doctest framework for testing various game scenarios
* **Importance**:
  + Helps ensure the game is functioning correctly and follows the rules