User Manual for

MonopolyCMD

Version 1.0

1. **Introduction**

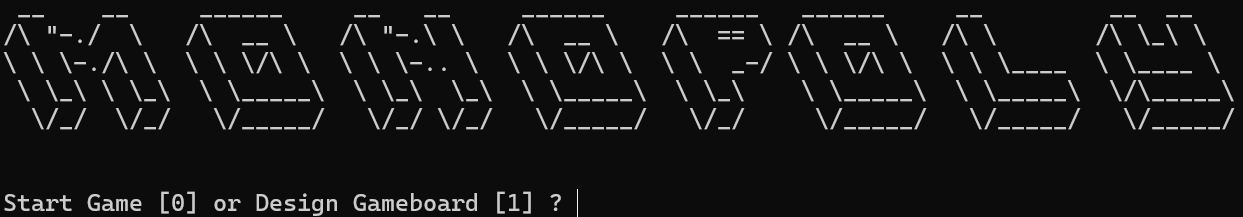
MonopolyCMD (MCMD) is a command-line version of the classic multiplayer board game Monopoly implemented in Python, intended for Windows machines. This is a user manual intended for assisting players and gameboard designers on using MCMD.

1. **Getting Started**

Users should first make sure that they have Python already installed on their PC. Users should first unzip the MCMD zipped source code downloaded from Github into a folder of their choice. Then, users can open up the Windows Command Prompt or Windows Powershell and make sure that the current directory is the MCMD folder they just created. After that, MCMD should launch by simply executing the following command in Windows CMD or Powershell:

python main.py

Users will be greeted by the logo of the game and their first choice. If you are a player, enter 0 here and press enter to start the game (Read section 3 of this manual for guides regarding gameplay). Gameboard designers can instead enter 1 and then press enter to start making their gameboard (Read section 4 of this manual for board designing instructions). Note that navigating this program will all be similar to this and users are always expected to press enter after inputting their choices. Invalid inputs will usually just cause the same prompt to repeat until the user gives an accepted input.



After choosing to start the game, users will be prompted to choose between starting a new game (0) or loading an existing save file (1) to play with.

**Start a new game (0)**

The user will then need to choose between using the default gameboard design (0) or an existing design (1). After successfully loading a gameboard for play, the user will need to enter the number of players and decide the names of all of them. Note that the players will always take turns according to the order that they input their name. This should be the end of the game creation process, and users will be asked once if they want to save the game. Answering this will start the main game of Monopoly.

**Playing with existing save files (1)**

Users should make sure that the save data is located inside the “\data\save\_state” folder inside the MCMD folder, and will be asked to input the complete file name (including “.json”) of the save file. A successful load will instantly resume the save game.

1. **Gameplay**

The players’ primary choices include checking various different kinds of status (0), querying the next player (1) and rolling the dice (input nothing and press enter).

**Show status (0)**

Here, players can choose to show their own status (0), check specific player status (1), all players status (2), or game status (3). Player status (0, 1 and 2) will include if the player is retired, player ID, player name, order, the player’s current amount of money, the player’s owned properties, if the player is currently in jail or not and if so, how many more rounds they still need to stay in the jail.

Choosing to view game status (3) will show the game id, current round number, ID of the current player and the gameboard status which includes information of all locations that exist on the gameboard.

**Query next player (1)**

Simply shows the name and player ID of the next player.

**Rolling the dice (empty input)**

Upon choosing this option, if the player is not currently in jail, the dice will automatically be rolled and the current player will also be automatically moved the rolled number of steps. Jailed players cannot move through dice rolling and will instead have a special prompt for other choices (see part vii. of this section). The square which the current player landed on will be shown, and according to the type of the square, further events might happen.

1. Property square

If the property is currently owned by no one, the current player will be shown its price and asked whether they want to buy it or not.

If the property is currently owned by someone else, the current player will be forced to pay money to the current owner. In the case that the current player doesn’t have enough money to pay the rent, the current player will instantly be retired.

1. Go square

Every time a player lands on or pass through this square, they will receive $1500.

1. Chance Square

Players who land on this square might gain or lose a random amount of money.

Gain: $10 - $200

Lose: $10 - $300

1. Income Tax

Players who land on this square will have 10% of their current money (rounded down to multiples of $10) taken away just like tax collection. If the player does not have enough money to pay this tax, they will instantly be retired.

1. Free Parking

This square has no effect.

1. Go to Jail

Players who land on this square will instantly go to the “In Jail/Just Visiting” square.

1. In Jail/Just Visiting Square

This square has no effect for players who just lands on it and didn’t go here through the “Go to Jail” mentioned above.

For players who went here through “Go to Jail”, in the first round of staying here, they will be asked if they want to pay a fine of $150 to instantly be freed, before rolling the dice. If they choose to pay(input “y”), they can move forward by the number of squares rolled. Else(input “n”), they will be forced to get lucky and roll for the same number on both dices to get out of jail, also for the next two rounds. In case of getting out of jail through the roll mentioned above, the player can also instantly move forward by the number they rolled. If a player isn’t lucky enough and can’t get out of jail through dice rolling for 3 rounds, they will be forced to pay the $150 fine to get out and again, get to move forward according to their throw. If the jailed player don’t have enough money to pay the fine, they will be instantly retired.

1. **Designing Gameboards**

After choosing Design Gameboard (1) in the very first prompt of the program, the user will next be asked if they want to start a new design (0) or to load an existing design (1) to edit.

**Start a new design (0)**

Before proceeding, the user should first prepare an empty dummy json file inside the “\data\gameboard \_design” folder inside the MCMD folder so that the program can later save the gameboard into this json file.

The user would be asked to first enter the name of the json file (including “.json”) they prepared for saving the gameboard, then the gameboard size. After that the user can start editing the gameboard.

**Load an existing design (1)**

The user would first need to input the file name of the existing gameboard design (including “.json”). After that, the user should be able to edit the gameboard.

**Editing the gameboard**

1. Edit a property (0)

Here, the user can choose to perform the following actions:

1. Insert a property (0)

The user will be asked to decide (in the following order,)the location, name, price, rent and whether the property can be owned or not (enter “true” or “false”).

1. Update a property (1)

Similar to the steps for inserting a property, except here the user can choose to keep the old values by an empty input.

1. Delete a property (2)

The user can choose to delete one of the existing properties on the gameboard. Please note that this action is irreversible, so please proceed with caution.

1. Up a level

Go back.

1. Edit a function (1)
2. Insert a function (0)

The user will be asked to decide (in the following order,) the location, name and function type.

1. Delete a function (1)

The user can choose to delete one of the existing function squares on the gameboard. Please note that this action is irreversible, so please proceed with caution.

1. Up a level

Go back.

1. change gameboard size (2)

The user can simply input the new gameboard size to their liking by inputting the total number of squares that will be present in the new gameboard.

1. View current design (3)

Here the current size, and information of all the existing squares will be shown to the user.

1. Check design for validity (4)

A handy option which shows the user what the gameboard is currently missing before it can be played in this program.

1. Discard design (5)

Instantly quit the program without saving the design.

1. Save & exit (empty input)

Quit the program after saving the gameboard.

If you have further enquiries, please don’t hesitate to send us an email at teamMCMD@gmail.com.