Project 1

Title

Monopoly\*

\*A text-based computer replica of the

well-known classic boardgame –

for educational purposes only

Course

**CSC-5 Programming Concepts and Methodology I: C++**

Section

**40514**

Due Date

**May 3, 2020**

Author

**Sonya Smalley**

# Basic Gameplay

The original Monopoly game (currently trademarked and published by Hasbro) on which this game program is based, is suitable for 2 to 8 players. However, due to time constraints and restrictions on the use of functions and arrays, as well as for the consideration of having readable code, this game program is restricted to two players: the human user (player 1) and the computer (player 2).

This game is more limited than the actual game of Monopoly.

# Player Actions

There are a limited number of actions that the player can take. Other than choosing their game tokens, most actions are automatic or require only that they press the Enter key.

# Properties

In the original Monopoly game, players may buy and sell properties, mortgage them when additional funds are needed, build houses and hotels on them, obtain regional monopolies, and charge rents for owned properties that vary accordingly with each of the aforementioned conditions. These abilities are not enabled in the game I have produced. Instead, in this game, all properties are “randomly” assigned to one or the other of the players prior to the start of gameplay. During the game, players will charge rent fees to other players who land on their owned properties. The value of each property is different and the rent to be charged varies according to location.

# Utilities

Utilities in the original Monopoly game are a special type of property. Instead of paying a set rent that is based on monopolies and how extensively properties are built out, the fees that players pay when they land on either Water Works or Electric Company are determined by the roll of the dice: a fee of $4 (or $10 if both utilities are owned by one player) multiplied by the number shown after a roll of the dice. For simplicity and consistency with the fee system being used for the other properties, I have chosen not to use that system (if I decide to use this game for the next project, I will use the proper system). Instead, a flat rate fee of $25 is charged when players land on the utilities.

# Go to Jail & Get Out of Jail

There are three ways to end up in jail: landing on the Go to Jail space, rolling three doubles in a row during one turn, and drawing a Chance or Community Chest card that tells you to Go to Jail. As soon as a player lands in jail, their turn is over. While they are in jail, they continue to collect rent fees as appropriate.

There are similarly three ways that a player can get out of jail: pay $50 on their next turn, use a Get Out of Jail Free card, or wait for up to 3 turns. If they choose to pay $50 or use a Get Out of Jail Free card, they will roll the dice and move out of jail the number of spaces indicated on their dice. If they choose to wait for 3 turns and try to get out of jail without spending money or a Get Out of Jail Free card, they must roll the dice at each turn. If they get doubles, they are free from jail. They may then roll again and move their token as usual. However, if they do not get doubles by the end of their third turn, they are required to pay $50 to get free from jail and then continue their turn as usual (roll, then move).

Due to the subtle nuances (as they appear to me as a novice programmer) of choosing the right places to place various pieces of code, I had a difficult time with this part of the game, particularly getting the player out of jail. I spent several hours trying to get this to work right. However, there are no versions to show prior to the “Go to Jail v1” code that will be available for you to peruse. This is because I did not have a working code until I finished writing “Go to Jail v1”, which, as I said, was the product of several hours of work (~5).

Chance and Community Chest Cards

When players land on the Chance or Community Chest spaces, a random card is chosen for them. The player must then immediately follow the instructions given by the card. If a fee is charged or gifted to the player, then the requisite funds are automatically deducted from or added to their accounts. If they are instructed to move to a specific space, their game piece is immediately moved to the specified location. If they receive a Get Out of Jail Free card, they may keep the card for later use. I have added the restriction that players may only hold one Get Out of Jail Free card at a time. Therefore, if they already have one, the program will select from the cards again until a different card is chosen.

# Pseudocode

Global Constant

Game Board = 40 spaces

Set Random Number Seed

Declare Variables and Initialize

InFile and OutFile – for reading from and writing to files

Players – one unique variable per player for indicating whose turn it is

Game Pieces – players will choose their game pieces for play

Player Turn – (Boolean) to see whose turn it is

Round – keep track of how many times round the board they go

Die1 and Die2 – two dice to be shared

Sumdie – sum the value of two rolled dice

Doubles – count of doubles per turn

Space – one variable per player to mark their positions on the board

Jail – (Boolean) one variable per player to indicate if they’re in jail or not

Money – one variable per player to track their money

Jail Turns – keep track of how many turns in a row a player is in jail (max 3)

Jail Choice – a player may choose how to get out of jail

Owner – to indicate who owns a property

Card – variables for choosing Chance and Community Chest cards

Jail Card – (Boolean) players may keep one Get Out of Jail Free card each

**Display Game Piece User Menu**

Game pieces are chosen using the gamePieces.dat file

Ask Player 1 to choose their game piece token

Player 1 selects a number and chooses their game piece token

Loop to pull Player 1’s choice from the file and then verify their choice

Loop to randomly assign a different game piece token to Player 2

**Randomly Assign Ownership to Players**

Create a file named owners2Players.dat to hold ownership values

Loop to assign ownership values to all 40 spaces

0 denotes properties that cannot be owned by a player

1 denotes properties that are owned by Player 1

2 denotes properties that are owned by Player 2

**Roll to See Who Goes First** – Highest Value Goes First

Loop the whole process

Tell user what is going on and prompt them to start the process

Roll die1 for Player 1 and die2 for Player 2

Output results to user

If-else statements to see who rolled highest

Output who goes first

If they rolled the same value, then repeat the entire process again

**Loop to play the game**

Test to see if player is in jail

If in jail, do **Get Out of Jail activities**

Track how many turns in a row the player is in jail (max 3)

Prompt user to choose pay, card, or roll (randomize the choice for computer player)

Validate user input and loop if there’s an error

If player pays, money is deducted and they’re freed

If player uses a card, the card is no longer in their possession and they are freed

If player rolls for doubles, there are three possible outcomes

If they roll doubles, they are freed

If it is their first or second turn in jail and they do not roll doubles, then they must try again on their next turn

If is their third turn in jail and they do not roll doubles, they pay the fee, and then are freed

If not in jail, proceed with **Regular Game Play**

Play is on a loop as long as the player rolls doubles and is not in jail

Roll two dice

Sum the dice and move the token the number of spaces indicated by the sum

If the token will move off the board, go round the board instead

If they land on Go (space 1) or pass Go, output that information and add $200 to their account

**Loop to check which space the player landed on**

If they land on chance or community chest, player is prompted to choose a card

Open cards file and loop to randomly select one

Output the card to the user

Test value of the card and take appropriate action dependent on which card it is

Some cards award money or take it away

Some send the player to different spaces on the board

Some put the player in jail

Some are Get Out of Jail Free cards, to be kept for later

If they land on Income Tax or Luxury Tax, money is deducted from their account

If they land on an ownable property, open properties file and loop to see who owns the property

If they own the property, then output that information and move on

If another player owns the property, open rent file

Loop to determine how much rent is owed

Deduct the required funds from their account and add them to the appropriate player’s account

If the player rolled doubles, they roll again

Count how many times in a row a player rolls doubles

If the player rolls three doubles in a row, they go to jail and their turn ends

Else, the loop starts at the top of regular game play