This is a fully playable, full-length Monopoly game for one or two players.

The game can be played in multiple ways: in its current setup, a Slick 2D visual game is instantiated in the Simple Slick Game class between a human player and the computer. If you comment out this game, a text-based version can be played through the Main class, which currently is set to instantiate a game between two computer players.

In this game, players start at Go with $1500, roll 2 random dice to move them around the board, and have the option of buying properties they land on. Once they accumulate a monopoly, they can buy houses and then hotels. Players can sell houses, sell hotels, or mortgage properties to raise money. Chance and Community Chest cards will either give/take away players’ money or transport them around the board. Landing in jail requires a player to roll doubles to exit, or pay $50, or possess a Get Out Of Jail Free card. In this program, players are unable to trade properties, which means that it is possible for the program to infinite loop (which happens 5-10% of the time) when neither player can secure a monopoly and bankrupt the other.

A further breakdown of the program’s classes:

**Game:** Whenever a new Game is instantiated (whether from Main or Simple Slick Game), players start on the board and take turns until someone’s turn ends in a negative balance. On every turn, a player rolls dice (unless he has chosen to pay to get out of jail) and lands on a property (or board property: Jail, Go, etc). If the property is available he may buy it, otherwise he may have to pay rent on it (mortgaged properties do not charge rent). At the end of his turn, the player is checked to see if he owns any monopolies, whether he wants to buy houses/hotels on any monopolies he already owns, if he has any mortgaged properties he wants to unmortgage, and for his ending balance.

If a human player would end his turn in the negative, he is given an opportunity to sell or mortgage all he owns in order to avoid bankruptcy. By opting ‘no’ on these choices, a human player can concede the game. A computer player is forced to always avoid bankruptcy if possible. Players whether human or computer are also prompted to sell/mortgage when they’d like to buy a property/house they can’t afford, or when they don’t have enough money to pay the opponent player.

When buying houses, a player must buy them evenly across a monopoly, and no hotels may be bought until all properties in that monopoly have four houses. A player may not buy houses or hotels on a monopoly where any of the properties are mortgaged.

The Game class also has a section detailing possible Chance and Community Chest cards, their logic, and their actions upon the player.

Jail, in this program, is represented by -1 when a player is not in jail, a number which increments to 0 when a player first lands in jail. The count increments as a player attempts to roll for doubles – if the count reaches 3 (meaning a player has unsuccessfully rolled three times), the player is freed and his Jail count goes back to -1.

**Simple Slick Game** contains the methods for a Slick 2D game. We can render images for the two board tokens moving around the board, houses and hotels on a property, a color-coded “possession marker” which appears on each owned property, and the interface for a human player. For games between two computer players, a sleep thread is included so that movement can be observed at a reasonable speed.

Methods describing image coordinates are included in order to make the game interactable by mouse. Properties, players, houses, hotels, possession markers, mortage indicators, and the interface for a human player are all mapped out by coordinates (the Position class allows us to set and get). Houses, hotels, and possession markers have a different orientation depending on whether they are on a “vertical” property or a “horizontal” property (if one is facing the board on the stretch between “Jail” and “Go”).

The balance of both players is also drawn onto the board (in the middle).

**Board Creator and Rent Schedule**. Board Creator is where we create all of the properties and add them into sets according to color, utilities, or railroads. Color sets also represent monopolies. Board properties such as Go, Luxury Tax, Free Parking, etc., are also created here, though they are not added to a set. The Rent Schedule helps us to keep track of how much a player must pay when they land on another player’s property, dependent on how many houses or hotels are on that property. (For railroads, how much a player pays is dependent on how many railroads are owned by the opponent; utilities are not in the rent schedule because how much a player pays is mostly dependent on a dice roll.)

**Property**. There are four constructors in Property, one each for board properties, non-board properties, utilities, and railroads. Theses constructors contain (when applicable) the property’s name, buy cost, mortgage cost, owner, number of houses, whether they are available for buying, whether they are mortgaged, what type of property they are, and what type of rent schedule they are on. Setters and getters for these parameters are included.

A location index method is included as an easy way to refer to any given property. “Go” is property 0, while “Boardwalk” is property 39.

The Get Rent Cost method breaks a non-board property down into whether it is a regular property, a railroad, or utility, assesses how many houses/railroads/utilities are owned, then returns the proper rent schedule cost.

Methods for adding a house, subtracting (selling) a house, or changing a mortgage status are included.

**Player**. The player handles what player token a player is represented by, his monetary balance, whether or not he’s in jail , the possession of any Get Out Of Jail Free cards, the player’s current position on the board, and a record of what properties he owns, what monopolies he owns, and which owned properties are mortgaged. Setters and getters for all these are included, as well as methods for counting up all of a player’s owned houses and hotels, and a method for determining the player’s “net worth” – the sticker value of all properties/houses/hotels owned, plus all cash on hand. Methods for buying houses, selling houses, mortgaging properties, and unmortgaging properties are also in this class.

The Player class has three inherited player types: Console Player, Gui Player, and Random Player. The first two represent human players (the difference is whether they are playing the text-based or Slick 2D game). The human player is always given a choice on whether or not they want to buy a property, whether or not they want to buy a house/hotel, where on a monopoly they want to buy that house/hotel (the list of options automatically only includes properties which are affordable, and which are eligible for house-buying based on evenness), whether they want to sell a house and where, whether they want to mortgage a property and where, and whether they want to unmortgage a property and where. Different methods are included depending on whether the player is making a yes/no decision or picking an option from a list.

The Random Player (computer player) faces these same choices and makes them based on a heuristic. If the player lands in Jail and owns fewer than seven properties or has more than $200, he must attempt to roll out of jail rather than pay to exit. The Random Player always buys the property he lands on, if available. If he is eligible to buy houses, and has more than twice the amount of cash that the property costs, he buys the house on a random property within the monopoly (again, the list of options only presents those available based on price and evenness). As mentioned before, both human and computer players are prompted to sell/mortgage properties and houses if their balance goes negative, if they’d like to buy something they can’t afford, or if they don’t have the money to pay another player. The Random Player must say yes to these prompts if he has less than $100 (in the case of whether or not he should sell houses), less than $500 (whether or not he should mortgage), or more than $600 (whether he should unmortgage).

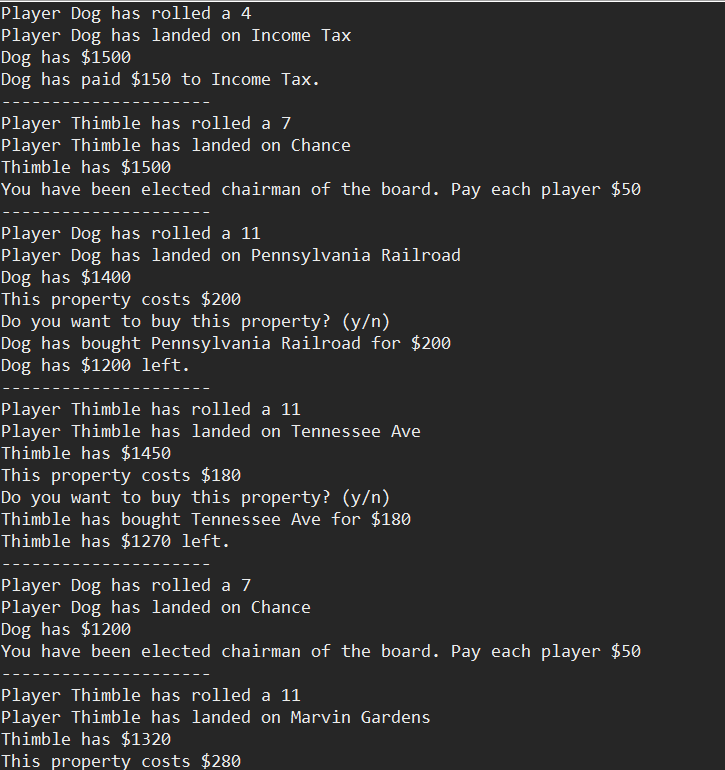
Here are some images from the game:







Text version, beginning stage:



End stage:

