Monopoly Programmer's Guide

**Main, default package**

StartGame.java - initializes the game from main and holds all objects in the Players and Properties class.

**Menu package**

MainMenu.java - initiated by StartGame and the first thing a user will see, this allows a person to start a new game, see the help menu or exit completely.

HelpMenu.java - an interactive JMenu with alt+[letter] functionality and a drop down menu on top.

**Pieces Package**

Board.java - this is the hub of everything. It initiates all other classes using buttons and action listeners. There are two JFrames. One holds all images and buttons. The other, stats holds all information for current events happening in the game. The top updates based on the current player, then below it updates the properties that are bought, mortgaged and have Villas or Pantheons. The last bit is a status text that outputs from the individual class that has been activated by clicking a button.

Dice.java - simple class with random integers 1-6, this allows Board to find dice images and to increment the player position upward. Dice total is used for utilities in the properties class.

**Events Package**

Turn.java - starts in Board when JButton rolling is clicked, this activates if statements based on where the token moves and updates the current player based on the turnCounter in Board, it also activates chance and community chest classes and pulls them back into Board.

Rent.java - this looks for the current player, the positon they landed on from the player object updated by Board and Dice classes, finds the owner in the properties class and takes away the rent amount in the property object found and increments up to the owner of the property. This is automatic and prints a response in the stats jframe.

Jail.java - searches dice for doubles, checks token for property object 31 and puts player token into jail for certain number of rounds or until doubles is found.

Player.java - a setter and getter that links to the StartGame class. It holds the array of objects 0-7, for players I-VIII. It can update the Balance and position. It gets most other integers and strings.

**Own Package**

Buy.java – starts in Board when JButton buying is clicked, this sets the Properties array item to switch owners and increments the player’s balance up and the owner that it switched to, gets their balance incremented down. This will make a confirmBuy button visible to complete the action.

Purchase.java - starts in Board when JButton purchase is clicked, this sets the Properties array item to bought 1-8 for each player and increments the player's balance down and Board updates by searching through the current Player object.

Sell.java - starts in Board when JButton sell is clicked, allows a player to sell item for full price only to a second player, changing the getOwner() integer to the new owner's number.

Trade.java - starts in Board when JButton trade is clicked, allows a property to be swapped between the current player and the second player. This will make a confirmTrade button visible to complete the action.

Mortgage.java - starts in Board when JButton mortgage is clicked, this allows a player to mortgage a Properties array item that they own to the bank and to later unMortgage it from the bank with interest, this is in the player object as a boolean.

Unmortgage.java - starts in Board when JButton unmortgage is clicked, this allows a player to buy back their property and rent can be collected again.

Improvements.java - starts in Board when JButton improve is clicked, holds all updates to villas and pantheons allowing the current player to update a owned property with a Villa or Pantheon, this outputs to the stats jFrame in the Board class.

SellImprovements.java – starts in Board when JButton simprove is clicked, holds the functions to remove updates to villas and pantheons allowing the current player to sell their Villa or Pantheon to the bank.

**Cards Package**

Chance.java - starts in Turn when landing on the Chance space, rolls a random dice to pull a chance card out of the stack and to move the player's token position, give a get out of jail card, or to increment the player balance up or down.

CommunityChest.java - starts in Turn when landing on the Community Chest space, rolls a random dice to pull a chest card out of the stack and to move the player's token position, give a get out of jail card, or to increment the player balance up or down.

**Tile Package**

Properties.java - this is the getter and setter for the Property objects in StartGame. This is just as much of a hub of the game as Board. Each property defines a space to move. There are 0-40 spaces. There are only 27 properties that can be bought, mortgaged, unmortgaged, sold, traded, or rented.

**Location of Images**

* Board - /pieces/boardrotate.jpg
* Villas, Pantheons, & Dice - /pieces/images/[imagetitle].png
* Chance & Chest cards - /cards/images/[imagetitle].png
* Title Deed spaces - /cards/images/[image title defined in Properties class].PNG
* Main Menu image - /menu/menu.jpg