Monopoly Documentation

Created by

Polapob Ratanachayoto 6331328521
Pisit Wongsripisant 6331332021

2110215 Programming Methodology
Semester 2 Year 2020
Chulalongkorn University

Object of the Game

Become the wealthiest player through buying, renting and upgrading property and force other players into bankruptcy.

How to win the Game

Player who is bankrupt will lose the game and another player will win the game immediately. Bankruptcy occurs when a player owes more than he or she can pay.

MainMenu scene

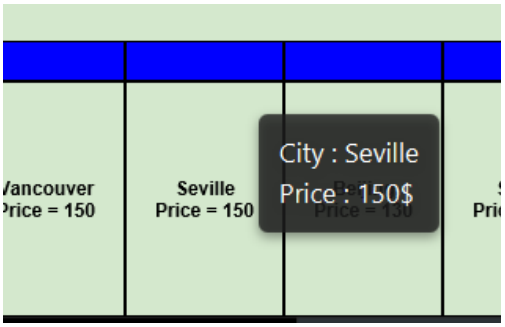
You can click the exit button to quit the game and click play button to start the game (Game Scene)!

Game scene

GameScene is a HBox that contains GameBoard on the left side and DicePane on right side.

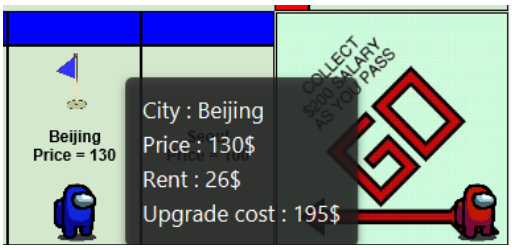
GameBoard is a board that contains squares. There are property squares which players can buy, start square which is the start point, jail square which will make people can't roll the dice for 1 turn, free parking square which does nothing to players, and go to jail square which forces players to jail square.

DicePane consists of a status text on the top which will give the player information about what is going on in the game. Dice picture which will change when people press the roll button to roll the dice. Buy area button to buy area, upgrade button to upgrade your property from area -> house -> hotel, end turn button to end your turn and exit button to exit the game.

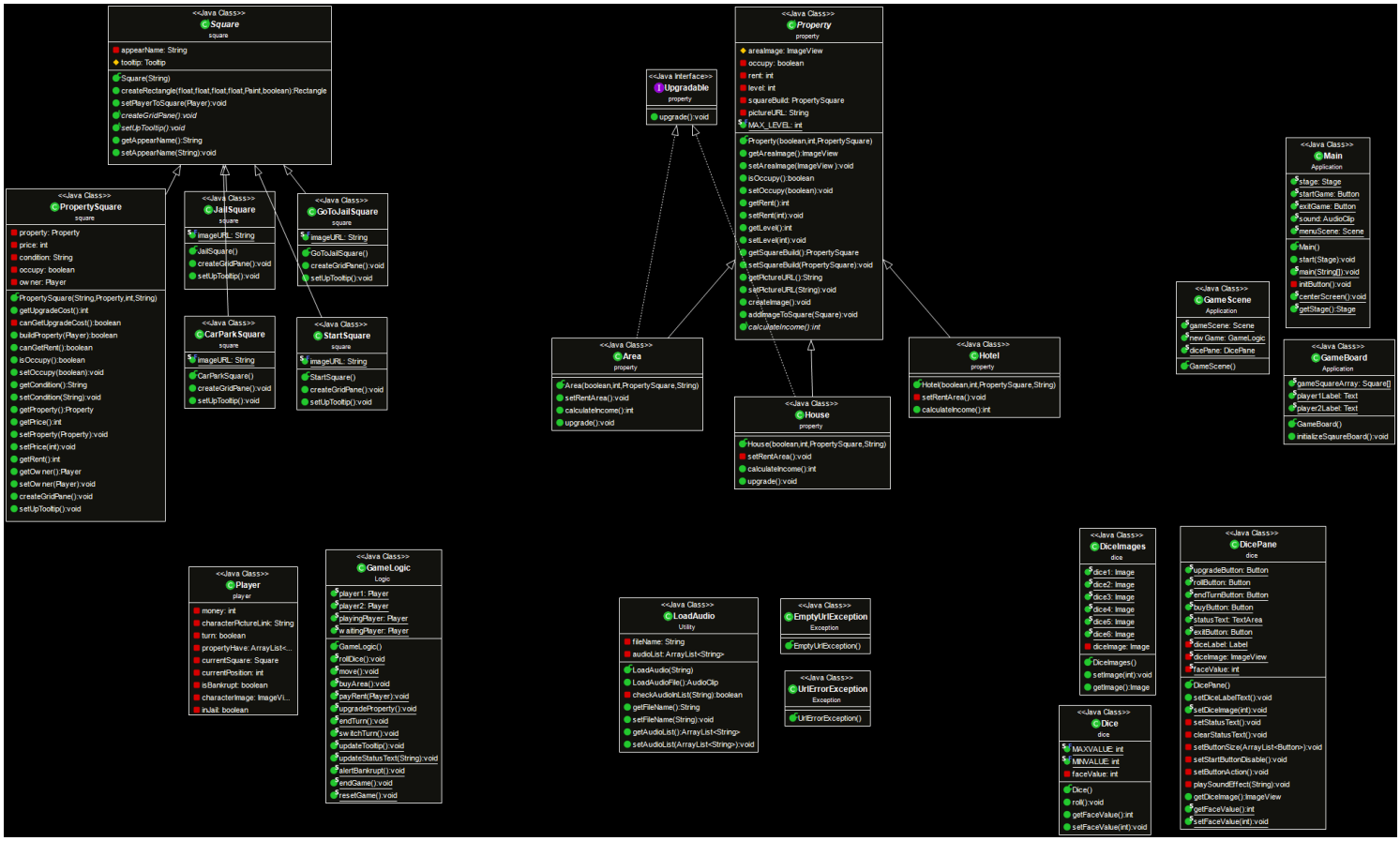


When you move your mouse to a property square which doesn't have any property, the tooltip will give you information of city name and price.

If a property square has a property square, the tooltip will have rent and upgrade cost.



Class diagram



1. Package player

1.1 Class Player

1.1.1 Fields

- int money	Player’s money
- String characterPictureLink	URL of player’s picture
- boolean turn	Is it player’s turn
- ArrayList<Property> propertyHave	Player’s properties
- Square currentSquare	Player’s square
- int currentPosition	Player’s position
- boolean isBankrupt	To check if the player has already gone bankrupt or not.
- ImageView characterImage	Player’s picture
- boolean inJail	To check if the player is in jail or not.
+ final int START_MONEY	Start money = 2000$
+ final int MAX_POSITION	Max Position = 28
+ final int JAIL_POSITION	Jail Position = 7
+ final int GOTOJAIL_POSITION	Go to jail position = 21

1.1.2 Constructors

+ Player(String characterPictureLink)	- set initial money with START_MONEY. - set the player's image by using characterPictureLink. - set bankrupt to false. - initialize array propertyHave.

	- set current position (default) to 0. - set the current square to the starting point. - set turn to false. - set character image.

1.1.3 Methods

+ boolean checkUnOccupyArea(Square square)	If square is a propertySquare and property of that square is null, return true; otherwise, return false.
+ void move(int FaceValue)	- If a player passes the start point, add money 200$ and call GameLogic to update the status text. - If player move to “jail Square” or “go to jail square”,then set inJail to true , change current position to jail and call GameLogic to update status text. - set player’s current position , player’s current square and set player’s picture to new square. ==== Now Check what destination square is and take action by calling doDestinationSquareAction method ===
- void doDestinationSquareAction()	- if destination square is property square, >> player can buy that square if player has enough money and square hasn't been occupied yet so set enable to buyButton in DicePane. >> if the square is already owned by other players, call GameLogic to pay rent to the owner. >> if the square is owned by this player, the player has enough money to upgrade and this

	property hasn't reached max level yet, set enable to upgradeButton in DicePane.
- void setInitialMoney(int money)	If money < 0 , set money to 0 ; otherwise, set initialMoney to money.
+ void setMoney(int money)	If money < 0 , set money to 0, set Bankrupt to true, call GameLogic to updateStatusText, alertBankrupt and endGame. otherwise, set initialMoney to money. Finally, call GameBoard to update the money text of every player.
+ void setCharacterImage()	Set by using characterPictureLink. Set Width, Height to 50.
+ String getCharacterPictureLink() + ImageView getCharacterImage() + int getMoney() + boolean isTurn() + ArrayList<Property> getPropertyHave() + Square getCurrentSquare() + boolean isBankrupt() + int getCurrentPosition() + boolean isInJail()	Getter methods.
+ void setCharacterPictureLink(String characterPictureLink) + void setTurn(boolean turn) + void setPropertyHave(ArrayList<Property> propertyHave) + void setCurrentSquare(Square currentSquare) + void setBankrupt(boolean isBankrupt) + void setCurrentPosition(int currentPosition) + void setInJail(boolean inJail)	Setter methods.

2. Package square

2.1 Abstract Class Square extends GridPane

2.1.1 Fields

- String appearName	Square's name
# Tooltip tooltip	Tooltip

2.1.2 Constructors

+ Square(String appearName)	Set AppearName.

2.1.3 Methods

+ Rectangle createRectangle(float xPos, float yPos, float boxHeight, float boxWidth, Paint color, boolean isStroke)	- initialize rectangle - setX, setY by using xPos, yPos - set width, set height and set color fill. - if isStroke is true, set stroke to black color.
+ void setPlayerToSquare(Player player)	- create arrays of int that contains the position of corner square in board (0,7,14,21) - if player's current position is in corner, add player image to the square grid pane (column 0 , row 0) - else add a player image to a grid pane which row and column will depend on what player's position is. (Just to make it look nice)
+ abstract void createGridPane() + abstract void setUpTooltip()	Abstract method.
+ String getAppearName()	Getter methods.
+ void setAppearName(String appearName)	Setter methods. If appearName is empty, set name to "Default Name"

2.2 Class PropertySquare extends Square

2.2.1 Fields

- Property property	Property that contain in this square
- int price	Price to buy this property square
- String condition	To check that this square is in first, second, third or fourth row on the game board.
- boolean occupy	To check that this square is already occupied or not.
- Player owner	Property's owner

2.2.2 Constructors

+ PropertySquare(String Name, Property property, int price, String condition)	- Call super constructor (super(Name)) - set other fields - call createGridPane and setUpTooltip method.

2.2.3 Methods

+ int getUpgradeCost()	If property's level = 0, upgradeCost is 1.5 times of price. If property's level = 1, upgradeCost is 2.0 times of price. Else return 0. // which means cannot upgrade
- boolean canGetUpgradeCost()	If property is null or reach max level, return false; otherwise, return true
+ boolean buildProperty(Player player)	If property in this square is null, create an area with propertyPicture.
+ void createGridPane()	Create small rectangle, large rectangle, text. Then, decorate all of it by checking condition fields. Finally, add to the grid pane.
+ boolean canGetRent()	If this square doesn't have property, return false; otherwise, return true.

+ void setUpToolTip()	>> When the mouse moves Over this square, show the city name, price. If this square can get rent, also add rent information. If this square can get upgrade cost, also add upgrade information. >> When the mouse exited this square, hide the tooltip.
+ boolean isOccupy() + String getCondition() + Property getProperty() + int getPrice() + int getRent() + Player getOwner()	Getter methods.
+ void setOccupy(boolean occupy) + void setCondition(String condition) + void setProperty(Property property) + void setPrice(int price) + void setOwner(Player owner)	Setter methods.

2.3 Class StartSquare extends Square

2.3.1 Fields

+ final String imageURL	imageURL = "/startPoint.png"

2.3.2 Constructors

+ StartSquare()	- Call super constructor (super(“Game Start”)) - Call createGridPane and setUpTooltip method.

2.3.3 Methods

+ void createGridPane()	Create a large rectangle, text. Then, fill the rectangle with an image from imageURL. Finally, add to the grid pane.

+ void setUpToolTip()	>> When the mouse moves Over this square, show the text “Start Point! Get 200$ every round.” >> When the mouse exited this square, hide the tooltip.

2.4 Class JailSquare extends Square

2.4.1 Fields

+ final String imageURL	imageURL = "/inJail.png"

2.4.2 Constructors

+ JailSquare()	- Call super constructor (super(“Jail”)) - Call createGridPane and setUpTooltip method.

2.4.3 Methods

+ void createGridPane()	Create a large rectangle, text. Then, fill the rectangle with an image from imageURL. Finally, add to the grid pane.
+ void setUpToolTip()	>> When the mouse moves Over this square, show the text “Jail! You can't roll the dice for 1 turn.” >> When the mouse exited this square, hide the tooltip.

2.5 Class CarParkSquare extends Square

2.5.1 Fields

+ final String imageURL	imageURL = "/freeParking.png"

2.5.2 Constructors

+ CarParkSquare()	- Call super constructor (super("Car parking lot")) - Call createGridPane and setUpTooltip method.

2.5.3 Methods

+ void createGridPane()	Create a large rectangle, text. Then, fill the rectangle with an image from imageURL. Finally, add to the grid pane.
+ void setUpToolTip()	>> When the mouse moves Over this square, show the text "Take a rest." >> When the mouse exited this square, hide the tooltip.

2.6 Class GoToJailSquare extends Square

2.6.1 Fields

+ final String imageURL	imageURL = "/goToJail.png"

2.6.2 Constructors

+ CarParkSquare()	- Call super constructor (super("Go To Jail")) - Call createGridPane and setUpTooltip method.

2.6.3 Methods

+ void createGridPane()	Create a large rectangle, text. Then, fill the rectangle with an image from imageURL. Finally, add to the grid pane.
+ void setUpToolTip()	>> When the mouse moves Over this square, show the text "You Are Under Arrest! Go back to jail." >> When the mouse exited this square, hide the tooltip.

3. Package property

3.1 Abstract Class Property

3.1.1 Fields

# ImageView areaImage	Image of property.
- boolean occupy	This property is occupied or not.
- int rent	The amount of money that the payer has to pay to the owner.
- int level	Level of property >> level 0 is area >> level 1 is house >> level 2 is hotel
- PropertySquare squareBuild	Property's square that this property is created.
- String pictureURL	Picture of property.
+ final int MAX_LEVEL	MAX_LEVEL = 2 (hotel)

3.1.2 Constructors

+ Property(boolean isOccupy, int level, PropertySquare squareBuild)	Set fields with setter methods.

3.1.3 Methods

+ void createImage()	Create image from pictureURL field. Set areaImage field. Set width and height to 50.
+ void addImageToSquare(Square currentSquare)	Add image to square with checking condition to make it different in each row. (For good look purpose)

+ abstract int calculateIncome()	Abstract methods.
+ boolean isOccupy() + int getRent() + int getLevel() + PropertySquare getSquareBuild() + String getPictureURL() + ImageView getAreaImage()	Getter methods.
+ void setOccupy(boolean occupy) + void setRent(int rent) + void setLevel(int level) + void setSquareBuild(PropertySquare squareBuild) + void setPictureURL(String pictureURL) + void setAreaImage(ImageView areaImage)	Setter methods.

3.2 Interface Upgradable

3.2.1 Methods

+ void upgrade()	Upgrade >> area -> house >> house -> hotel

3.3 Class Area extends Property implements Upgradable

3.3.1 Constructors

+ Area(boolean isOccupy, int level, PropertySquare squareBuild, String pictureURL)	Set fields by calling super constructor. Call the setRentArea method. Set fields with setter methods. Call createImage method. Call addImageToSquare method.

3.3.2 Methods

+ void setRentArea()	Set rent by calling the calculateIncome method.
+ int calculateIncome()	Calculate rent income. Multiple property’s

	price by 0.2
+ void upgrade()	Delete old property by setting it to null. Then, create a new property which is a house. (Consider who upgrade it to set a different house picture color)

3.4 Class House extends Property implements Upgradable

3.4.1 Constructors

+ House(boolean isOccupy, int level, PropertySquare squareBuild, String pictureURL)	Set fields by calling super constructor. Call the setRentArea method. Set fields with setter methods. Call createImage method. Call addImageToSquare method.

3.4.2 Methods

+ void setRentArea()	Set rent by calling calculateIncome method.
+ int calculateIncome()	Calculate rent income. Multiple property's price by 0.35
+ void upgrade()	Delete old property by setting it to null. Then, create a new property which is a hotel. (Consider who upgrade it to set a different hotel picture color)

3.4 Class Hotel extends Property

3.4.1 Constructors

+ Hotel(boolean isOccupy, int level, PropertySquare squareBuild, String pictureURL)	Set fields by calling super constructor. Call setRentArea method. Set fields with setter methods.

	Call createImage method. Call addImageToSquare method.

3.4.2 Methods

+ void setRentArea()	Set rent by calling calculateIncome method.
+ int calculateIncome()	Calculate rent income. Multiple property's price by 0.5

4. Package Dice

4.1 Class Dice

4.1.1 Fields

+ final int MAXVALUE	Set the max value to 6.
+ final int MINVALUE	Set the min value to 1.
- int faceValue	Face value of dice.

4.1.2 Constructors

+ Dice()	

4.1.3 Methods

+ void roll()	Random number 1-6 and set the number to face value field.
+ int getFaceValue()	Getter methods.
+ void setFaceValue(int faceValue)	Setter methods. >> If faceValue is greater than MAXVALUE, set faceValue to MAX VALUE.

	>> If faceValue is less than MINVALUE, set faceValue to MIN VALUE. >> else set faceValue with the parameter.

4.2 Class DiceImages

4.2.1 Fields

+ Image dice1	Initialize dice 1 image.
+ Image dice2	Initialize dice 2 image.
+ Image dice3	Initialize dice 3 image.
+ Image dice4	Initialize dice 4 image.
+ Image dice5	Initialize dice 5 image.
+ Image dice6	Initialize dice 6 image.
- Image diceImage	Default is dice 1 image

4.2.2 Constructors

+ DiceImages()	

4.2.3 Methods

+ void setImage(int value)	Setter methods. Set image of diceImage field with suitable picture to match with value in parameter.
+ Image getImage()	Getter methods.

4.3 Class DicePane extends VBox

4.3.1 Fields

+ Button upgradeButton	Initialize button with “Upgrade Area” text.
+ Button rollButton	Initialize button with “Roll” text.
+ Button endTurnButton	Initialize button with “End Turn” text.
+ Button buyButton	Initialize button with “Buy Area” text.
+ TextArea statusText	Initialize text area.
+ Button exitButton	Initialize button with “Exit” text.
- Label diceLabel	Show dice information text.
- ImageView diceImage	Initialize diceImage.
- int faceValue	Set default to 1.

4.3.2 Constructors

+ DicePane()	- Set alignment to center. - Set width to 300. - Set spacing to 40. - Set border stroke with light gray color. - Set status text - Set dice label text - Initialize an array list that contains all buttons then called setButtonSize method. - call setStartButtonDisable method. - call setButtonAction method. - call setDiceImage method. - create buttonPane VBox, set spacing to 15, set alignment to center, add buttons in button pane. - add statusText, diceLabel, diceImage, buttonPane, exitButton to this class (dicePane)

4.3.3 Methods

+ void setDiceLabelText()	- Initialize labels.

	- Set font to “Arial” and font size to 20. - Set text to “DICE (1-6)” - Set top padding to 60.
+ void setDiceImage(int faceValue)	- Initialize image with DiceImages class. - Set image with face value in parameter. - Set width and height to 100.
- void setStatusText()	- Set initial text to “==== Player 1 Turn ====\n\n” - Set font to “Arial” and font size cto 15 - Set wrap text to true.
- void clearStatusText()	- If it’s player1 turn, set statusText to "==== Player 1 Turn ====\n\n" - else, set statusText to "==== Player 2 Turn ====\n\n" - This method will be called when switching turns.
- void setButtonSize(ArrayList<Button> allButton)	- For each button set width to 150 and height to 50. - Set top padding for exitButton to 70.
- void setStartButtonDisable()	- This is initial condition of all button - Only rollButton enabled to be clicked.
- void setButtonAction()	- Roll button called GameLogic to roll dice and move players. Disabled roll button and enabled end turn button. Play sound effects. - Buy button called GameLogic to buy area. Disabled buy button. Play sound effects. - Upgrade button called GameLogic to upgrade property. Disabled upgrade button. Play sound effects. - End turn button called GameLogic to end turn. Called clearStatusText method. Disabled all buttons except the roll button. - When you click the exit button, exit the program.
- void playSoundEffect(String url)	- Play sound - Use try catch to catch exceptions when the url is incorrect.

+ ImageView getDiceImage() + int getFaceValue()	Getter methods.
+ void setFaceValue(int faceValue)	Setter methods.

5. Package Utility

5.1 Class LoadAudio

5.1.1 Fields

- String filename	Destination place of file.
- ArrayList<String> audioList	List of audioPath which audio is stored

5.1.2 Constructors

+ LoadAudio(String filename)	setFileName

5.1.3 Methods

+ AudioClip LoadAudioClip() throws UrlErrorException,EmptyUrlException	Load an Audio from file path if the path is empty, call EmptyUrlException and play default audio. Else If the path is not in audioList, call UrlErrorException and play default audio. Otherwise play the song from the path.
+ boolean checkAudioInList(String fileName)	Iteratively check fileName from audioList if fileName is in audioList return true else False.
+ String getFileName() + ArrayList<String> getAudioList()	Getter method
+ void setAudioList(ArrayList<String> audioList) + void setFileName(String fileName)	Setter method

6. Package Exception

6.1 Class EmptyUrlException extends Exception

6.1.1 Constructors

+ EmptyUrlException()	- Called super constructor "Your URL is empty" message.

6.2 Class UrlErrorException extends Exception

6.2.1 Constructors

+ EmptyUrlException()	- Called super constructor "URL ERROR PLEASE TRY AGAIN." message.

7. Package Logic

7.1 Class GameLogic

7.1.1 Fields

+ Player player1	Initialize player1 with player1 picture.
+ Player player2	Initialize player2 with player2 picture.
+ Player playingPlayer	This field is a current player.
+ Player waitingPlayer	This field is a waiting player.

7.1.2 Constructors

+ GameLogic()	=== Set start game condition === - Set player1 ‘s turn to true. - Set player2 ‘s turn to false. - Set playing player to player 1 - Set waiting player to player 2

7.1.3 Methods

+ void rollDice()	This method is to random dice from 1-6 then, change dice image and update status text to show face value.
+ void move()	Move the playing player by the face value.
+ void buyArea()	Check the current square of the playing player to get the price for buying. Set the area property in that square, deduct money, update tooltip, and update status text.
+ void payRent(Player payer)	Get the current square of the payer. Then, calculate income for the owner. After that, deduct money from payer to owner and update status text.
+ void updateProperty()	Get the current square of the playing player. Then, call the getUpgradeCost method to get the cost. After that, deduct money, upgrade property, update tooltip and update status text.
+ void endTurn()	>> If the waiting player is in jail and the playing player is in jail, switch turns. >> If only the waiting player is in jail, don’t switch turn and set the waiting player’s in jail to false so next round he/she can move now. >> else, (no one in jail) switch turns normally.
+ void switchTurn()	- Set the playing player’s turn to false. - Set waiting player’s turn to true. - Switch playing player and waiting player.
+ void updateTooltip()	Get the current square of the playing player. Then, call setUpToolTip from the current square if it is a property square.

+ void updateStatusText(String addText)	Set status text from DicePane class by adding addText.
+ void alertBankrupt()	- Set alert type to warning. - Set the alert title to “BANKRUPT !” - Set the alert content text to “GAME OVER !\nYou are bankrupt.” - Wait for click ok.
+ void endGame()	- Get the stage from the main class and set scene to menu scene. - Called the centerScreen method from main class. - Called reset game method.
+ void resetGame()	- Initialize new player to player1 and player2 - Set player 1 ‘s turn to true. - Set player 2 ‘s turn to false. - Set the playing player to player1 and set the waiting player to player 2.

8. Package Application

8.1 Class GameScene

8.1.1 Fields

+ Scene gameScene	Game scene.
+ GameLogic newGame	GameLogic
+ DicePane dicePane	DicePane

8.1.2 Constructors

+ GameScene()	>> Create an HBox, set spacing to 10, padding

	every side to 10, and set height to 400. >> Initialize gameBoard, gameLogic, dicePane. >> Add gameBoard and dicePane to HBox. >> Set HBox to gameScene. >> Add stylesheet2.css to gameScene.

8.2 Class GameBoard extends GridPane

8.2.1 Fields

+ Square[] gameSquareArray	Initialize array with array size = 28.
+ Text player1Label	Initialize text (To show player's money)
+ Text player2Label	Initialize text (To show player's money)

8.2.2 Constructors

+ GameBoard()	- Called initializeSquareBoard method. - Set font to player1Label and player2Label with "Cordia New '' font and size = 20. - add every square from gameSquareArray in gridPane. - add a player picture to start position. - add player1Label and player2Label. - Set background fill to color #d4e8cd.

8.2.3 Methods

+ void initializeSqaureBoard()	For each index in array, initialize square for game board.

8.3 Class Main extends Application

8.3.1 Fields

+ Stage stage	Stage.
+ Button startGame	Initialize button with "Play" text.
+ Button exitGame	Initialize button with "Exit" text.
+ AudioClip sound	Background music.
+ Scene menuScene	Menu scene.

8.3.2 Methods

+ void start(Stage primaryStage) throws UrlErrorException, EmptyUrlException	- Set stage field to primaryStage. - Play background music with an infinite loop. - Create VBox, set background image and set alignment to center. - Initialize image (monopoly logo), set width to 300 and height to 700. - Create an Hbox button pane, called initButton method, set alignment to center, set spacing to 50. - Add the startGame button and exitGame button to buttonPane. - Add logo and buttonPane to root (VBox). - Initialize menu scene - Add stylesheet.css to the menu scene. - Set stage's title to "Monopoly Game". - Set menuScene to stage. - Show stage and setResizable to false. - Called centerScreen method.
+ void main(String args[])	launch(args)
- void initButton()	- Set on action startGame button. Initialize GameScene and set gamescene to stage. Called the centerScreen method. - Set on action exitGame button. Close the program. - Set Ariel font to button and set font size to 50. - Set button width to 300 and height to 80.
+ void centerScreen()	Set the stage to the center of the screen.

+ Stage getStage()	Getter Methods.