Java Project Report

on

Dora Cafe

Submitted by

Melvin Macaranas 6030478021

Yodsavee Lertthirathanaphong 6030482421

Subject

Course Programming Methodology Semester 1

Story behind the game : DORA Cafe

Figure 1 : The manga series "Doraemon"

Doraemon is a Japanese manga series written and illustrated by Fujiko F. Fujio. as shown is *figure 1.* The series has also been adapted into a successful anime series and media franchise. The story revolves around a robotic cat named Doraemon, who travels back in time from the 22nd century to aid a boy named Nobita Nobi. In this game, you will become Nobita's Mother or Nobita's father which open a cafe for the kids: Doraemon, Nobita, Suneo, Shizuka and Giant.

1. User Manual

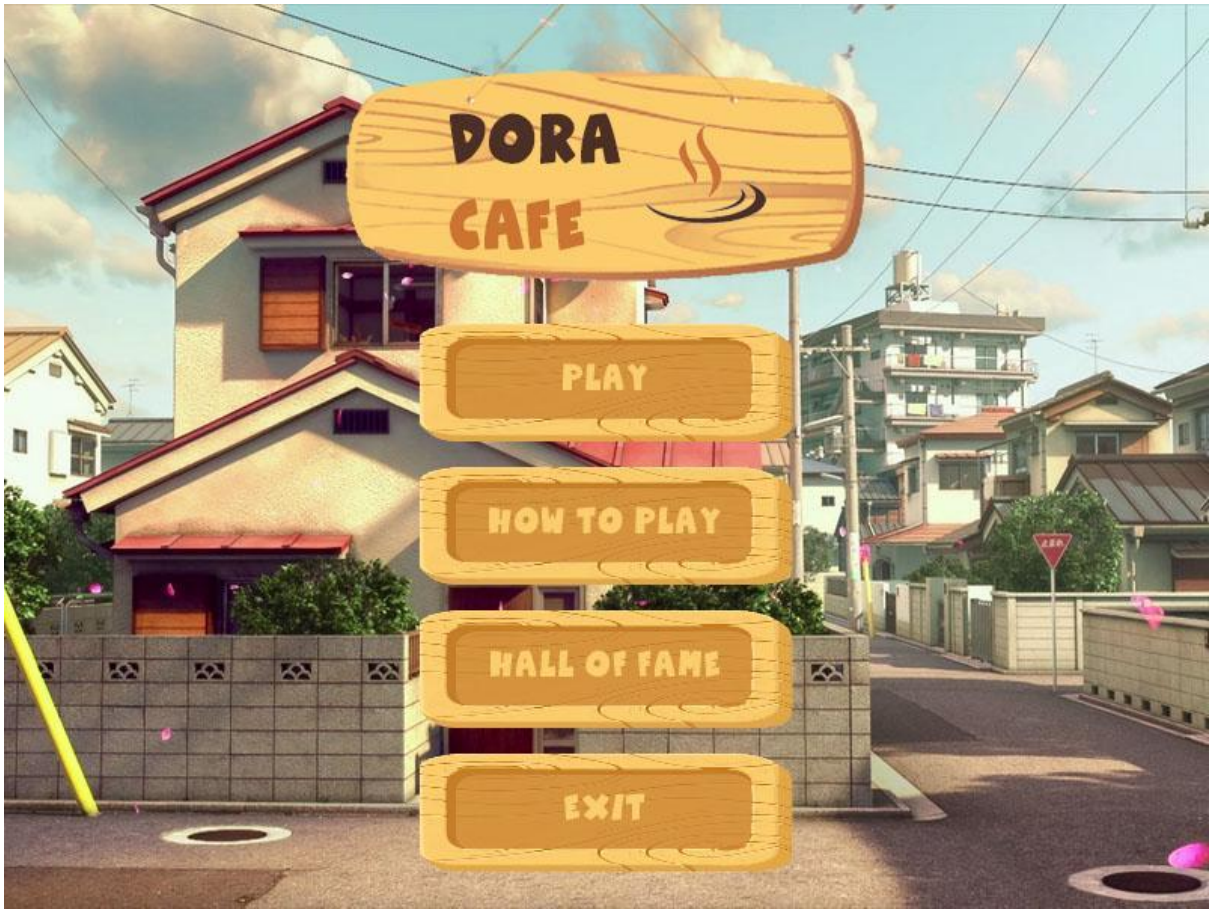

Figure 2 : Welcome screen of the game

When start the program, you will see a welcome screen that have 4 buttons : PLAY, HOW TO PLAY, HALL OF FAME and EXIT as show in *figure 2*.

When you click EXIT:

- Your game will exit.

Figure 3 : Hall of Fame screen

When you click HALL OF FAME:

- You will see the TOP 3 player of the game as shown in figure 3.

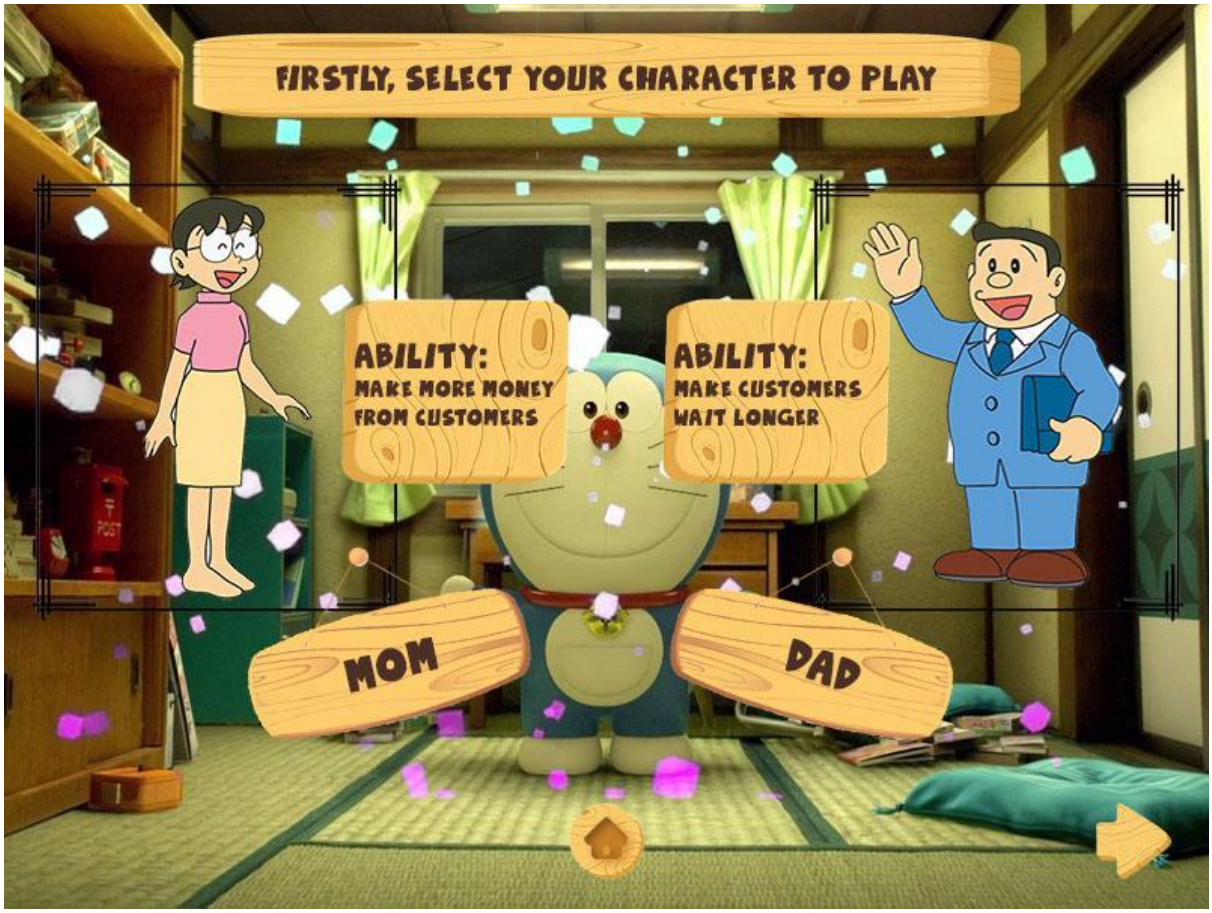

Figure 4.1 : How to play screen 1

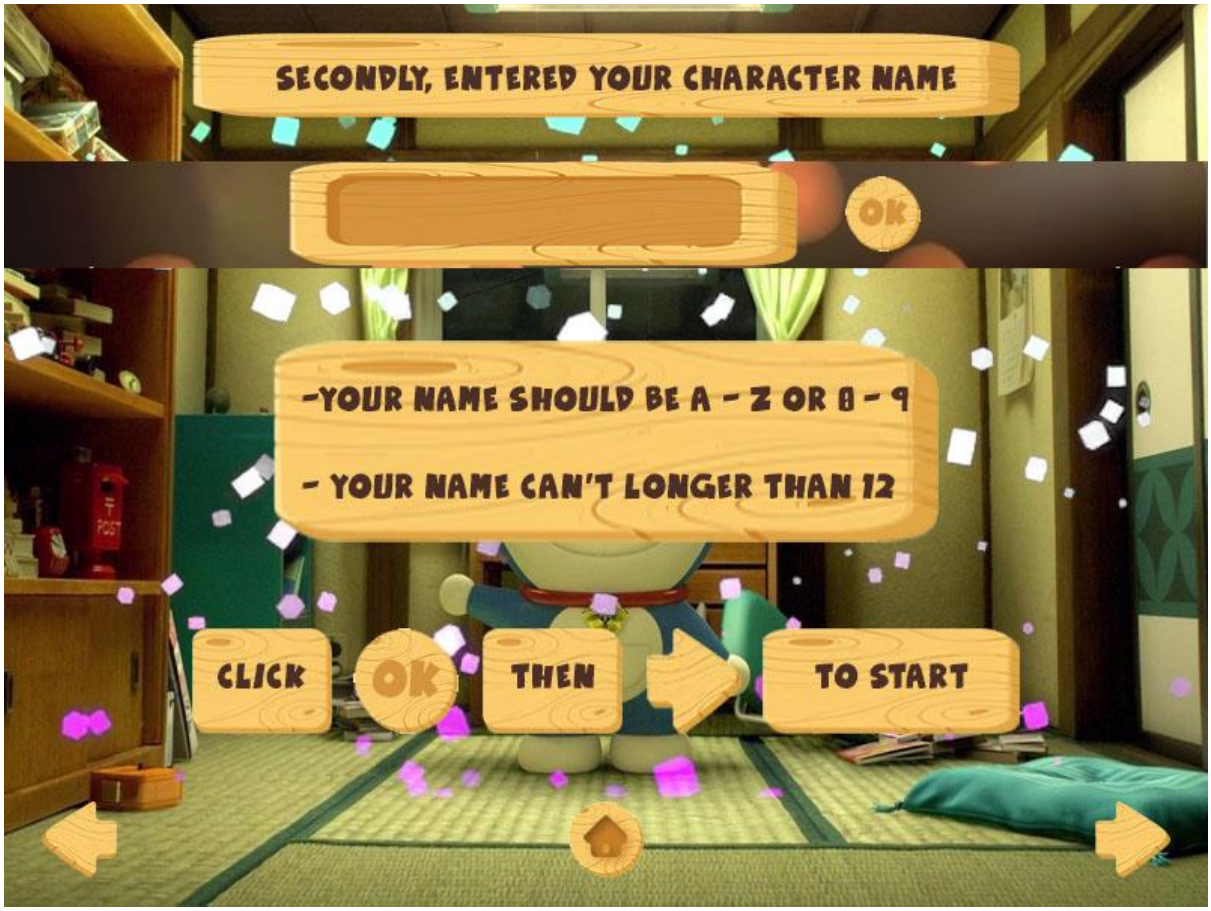

Figure 4.2 : How to play screen 2

Figure 4.3 : How to play screen 3

Figure 4.4 : How to play screen 4

Figure 4.5 : How to play screen 5

Figure 4.6 : How to play screen 6

Figure 4.7 : How to play screen 7

Figure 4.8 : How to play screen 8

Figure 4.9 : How to play screen 9

When you click HOW TO PLAY:

- You will see an instruction of the game as shown in *figure 4.1- figure 4.9*.

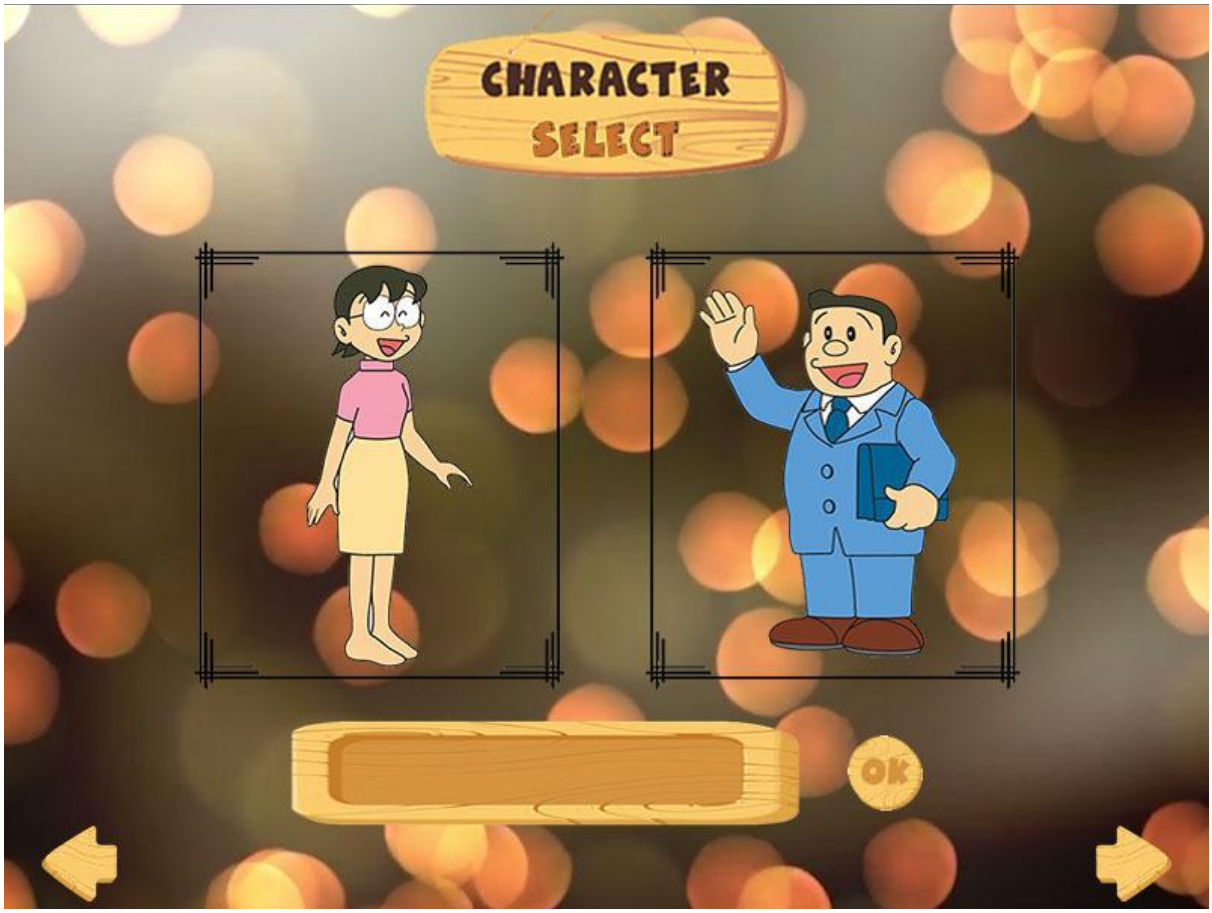

Figure 5 : Character select screen

When you click PLAY:

- You will see a character selecting screen as shown in *figure 5*.
- You need to choose your character which is ‘Mom’ or ‘Dad’.
- ‘Mom’ can collect more money.
- ‘Dad’ can make your customer wait longer.
- You also need to type your name which can be an alphabet A-Z or number 0-9 and your name must be 1-12 characters long.
- You need to click ‘OK’ after you type your name and you can’t change your name after you click ‘OK’.
- If you wish to go back to the welcome screen, click the back button.
- If you have chosen your character, typed your name, clicked ‘OK’ and you wanted to play, click the next button and you will go to the game screen.

Figure 6 : Game screen

Game Screen:

- In the game screen, it is separated into 2 part which are the main game part and the status bar as shown in *figure 6*.
- the status bar will show the character you have selected, the time left, the money you have earned, the heart score you have left and the message which will remind you every time you make your customer mad or you do something wrong for example, you put your customer to the table which is not available.
- If you have no more heart score left or the time is up, you will see a game over screen.

Figure 7 : Game over screen

Game Over Screen:

- You will see a conclusion of an amount of money you have earned this time and the rank you are in among every other player as shown in *figure 7*.
- If you click the 'OK' button on the bottom of the screen, you will reach the 'HALL OF FAME' screen to see the top 3 player of the game.

2. Implementation Detail

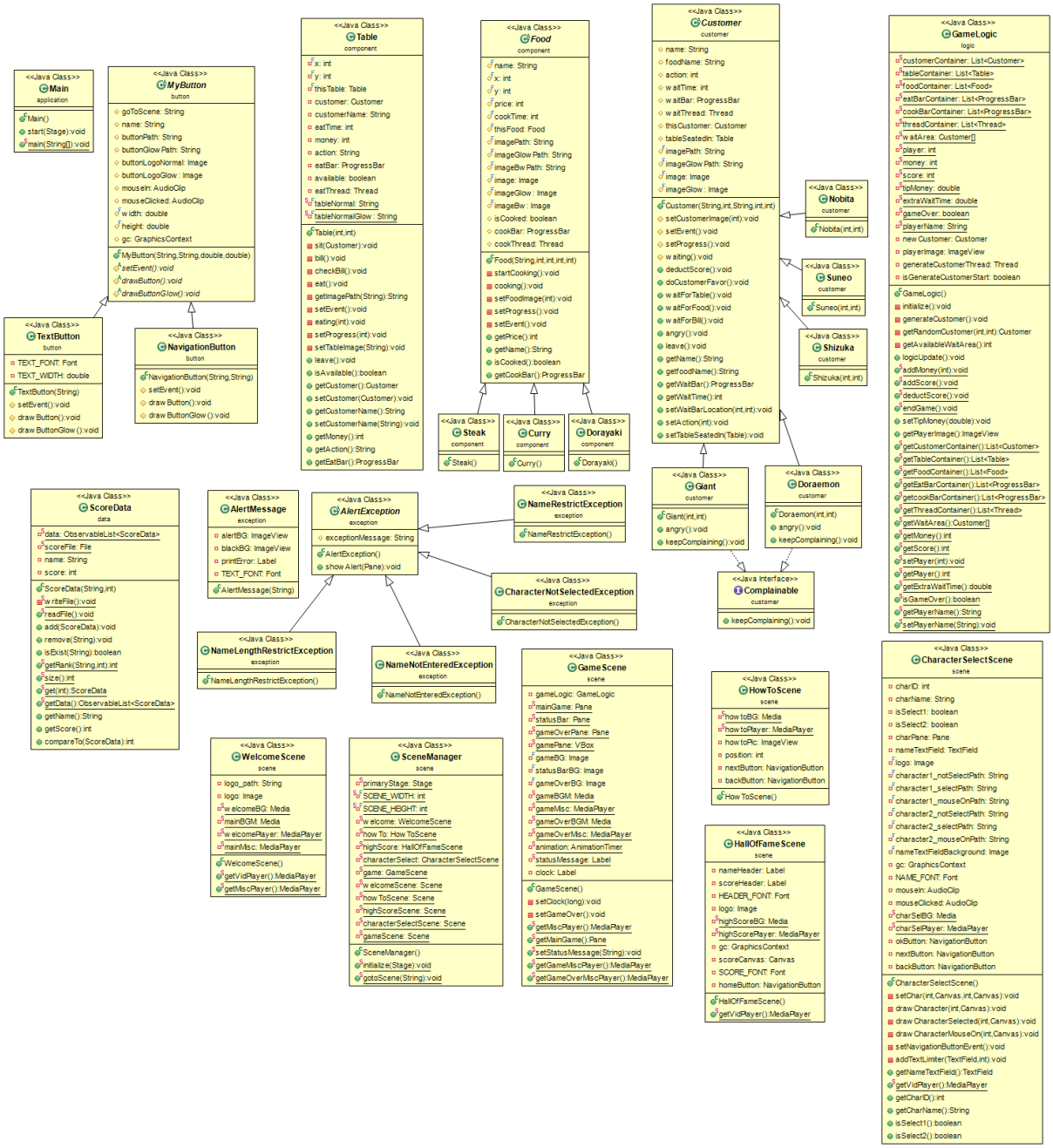

Figure 8 : UML Diagram of the application

Access Modifier Notations can be listed below

+ (public)

(protected)

- (private)

2.1 Package application

2.1.1 class Main

2.1.1.1 Method

+ void start	The main entry point for the JavaFX applications.
+ void main	An entry point of the application

2.2 Package button

2.2.1 abstract class MyButton extends Canvas

2.2.1.1 Field

# String goToScene	The scene that the button will navigate to.
# String name	The name of the button
# String buttonPath	Image path of the button
# String buttonGlowPath	Image path of the button (glow)
# Image buttonLogoNormal	Button image normal
# Image buttonLogoGlow	Button image glow
# AudioClip mouseIn	Sound when mouse enter button
# AudioClip mouseClicked	Sound when mosue click
# final double width	Width of the button
# final double height	Height of the button
# GraphicsContext gc	GraphicsContext of this button's canvas

2.2.1.2 Constructor

+ MyButton(String name, String toScene, double width, double hright)	Set width and height of the canvas and set the name of the button and set the goToScene of the button and set the image and audio

2.2.1.3 Method

# abstract void setEvent()	abstract method for setting event of the button
# abstract void drawButton();	abstract method for draw normal button
# abstract void drawButtonGlow();	abstract method for draw glow button

2.2.2 class NavigationButton extends MyButton

2.2.2.1 Constructor

+ NavigationButton(String name, String toScene)	Set super constructor and draw normal button and set button event

2.2.2.2 Method

# abstract void setEvent()	Override method for setting event of the button
# abstract void drawButton();	Override method for draw normal button
# abstract void drawButtonGlow();	Override method for draw glow button

2.2.3 class TextButton extends MyButton

2.2.3.1 Field

- Font TEXT_FONT	Font for text on button
- double TEXT_WIDTH	The width of the text

2.2.3.2 Constructor

+ TextButton(String name)	Set width of the button then draw and setEvent of the button

2.2.2.2 Method

# abstract void setEvent()	Override method for setting event of the button
# abstract void drawButton();	Override method for draw normal button
# abstract void drawButtonGlow();	Override method for draw glow button

2.3 Package component

2.3.1 abstract class Food extends ImageView

2.3.1.1 Field

# final String name	The name of the food
# final int x	The x-axis position of food image
# final int y	The y-axis position of food image
# final int price	The price of food
# final int cookTime	Time to cook the food
# final Food thisFood	It is the food itself
# final String imagePath	Image path of food image
# final String imageGlowPath	Image path of glow food image
# final String imageBwPath	Image path of black and white food image
# final Image image	Food image normal
# final Image imageGlow	Food image glow
# final Image imageBw	Food image black and white

# final boolean isCooked	Check that is the food cooked yet
# final ProgressBar cookBar	Progress bar to show cookTime
# final Thread cookThread	Thread to run each cookBar

2.3.1.2 Constructor

+ Food(String name, int price, int cookTime, int x, int y)	Set name , price and cook time of food , set image normal , glow and black and white , set image position ,set cook bar width, position and style , set event and start cooking

2.3.1.3 Method

- void startCooking	Set cook bar progress to 0 , visible and disable and set food image then cooking
- void cooking()	Set the cookThread and run it
- void setFoodImage(int status)	Set the image of the food in each state (isCooked, normal, glow)
- void setProgress()	Add the progress of the cookBar
- void setEvent()	Set the event handler of the
+ int getPrince()	Getter for price
+ String getName()	Getter for name
+ boolean isCooked()	Return true if the food is cooked, return false otherwise
+ ProgressBar getCookBar()	Getter for cookBar

2.3.2 class Curry extends Food

2.3.2.1 Constructor

+ Curry()	Set super constructure with curry option

2.3.3 class Dorayaki extends Food

2.3.3.1 Constructor

+ Dorayaki()	Set super constructure with dorayaki option

2.3.4 class Steak extends Food

2.3.4.1 Constructor

+ Steak()	Set super constructure with steak option

2.3.5 class Table extends ImageView

2.3.5.1 Field

- final int x	Position on x-axis of the table
- final int y	Position on y-axis of the table
- final Table thisTable	It is the current table itself
- Customer customer	The customer sitting on that table. if no customer it is null
- String customerName	The customer name sitting on that table. if no customer it is “”
- int eatTime	The time that a customer will eat depends on what food
- int money	Money gaining on this table
- String action	Action of the customer on this table which are None, Sit, Eat or Bill.
- ProgressBar eatBar	Progress bar showing progress of eating

- boolean available	True if the table is available, false otherwise
- Thread eatThread	Thread add progress to the eatBar
- static final String tableNormal	Image path of the table normal state
- static final String tableNormalGlow	Image path of the table glow state

2.3.5.2 Constructor

+ Table(int x, int y)	Initialize the table

2.3.5.3 Method

- void sit(Customer customer)	Take the customer to sit to the table
- void bill()	Change the table to the Bill state
- void checkBill()	Check bill your customer and add money to the GameLogic
- void eat()	Change the table to eat state
- String getImagePath(String action)	Return the image path of the table depend on that action
- void setEvent()	Set the Event of the table
- void eating(int eatTime)	Initialize and start the eatThread
- void setProgress(int waitTime)	Set the progress of the eatBar while the eatThread is running
- void setTableImage(String action)	Set the image of the table
+ void leave()	Set that customer is leaving
+ boolean isAvailable()	Getter for available

+ Customer getCustomer()	Getter for customer on the table
+ void setCustomer(Customer customer)	Setter for customer on the table
+ String getCustomerName()	Getter for customer on the table's name
+ void setCustomerName(String customerName)	Setter for customer on the table's name
+ int getMoney()	Getter for money earning from this table
+ String getAction()	Getter for table's action
+ ProgressBar getEatBar()	Getter for the table's eatBar

2.4 Package data

2.4.1 class ScoreData implements Comparable<ScoreData>

2.4.1.1 Field

- static ObservableList<ScoreData> data	Container of the ScoreData file
- static File scoreFile	File that the data are saved in
- String name	Name of the player in this ScoreData
- int score	Score or Money that this customer earned from the game

2.4.1.2 Constructor

+ ScoreData(String name, int score)	Create a ScoreData file and add it to the container

2.4.1.3 Method

- static void writeFile()	Save the data on the file
+ static void readFile()	Read data saved in the file to the List
+ void add(ScoreData data)	Add new ScoreData file to the container
+ void remove(String name)	Remove a scoreData file from the container

+ boolean isExist(String name)	Return true if that player's name is exist
+ static int getRank(String name, int score)	Return rank of that player by inputting player's name and score (money)
+ static int size()	Getter for the container's size
+ static ScoreData get(int i)	Getter for the ScoreData index i in the container
+ static ObservableList<ScoreData> getData()	Getter for the ScoreData's container
+ String getName()	Getter for the customer's name of this ScoreData
+ int getScore()	Getter for the customer's score (money) of this ScoreData
+ int compareTo(ScoreData o)	Compare method inherit from Compareble Interface

2.5 Package exception

2.5.1 abstract class AlertException extends Exception

2.5.1.1 Field

# String exceptionMessage	The scene that the button will navigate to.

2.5.1.2 Method

+ void showAlert(Pane pane)	Show the alert

2.5.2 class AlertMessage extends StackPane

2.5.2.1 Field

- ImageView alertBG	Alert box image
- ImageView blackBG	Black image for increase attractive to alert
- Label printError	Label for show what error occurs

- Font TEXT_FONT	Font for print error label

2.5.2.2 Constructor

+ AlertMessage(String message)	Set print error to message , set width , wrap text alignment , padding and font of printError label and set position of label.

2.5.3 class CharacterNotSelectedException extends AlertException

2.5.3.1 Constructor

+ CharacterNotSelectedException()	print error and set exception message

2.5.4 class NameLengthRestrictException extends AlertException

2.5.4.1 Constructor

+ NameLengthRestrictException()	print error and set exception message

2.5.5 class NameNotEnteredException extends AlertException

2.5.5.1 Constructor

+ NameNotEnteredException()	print error and set exception message

2.5.6 class NameRestrictException extends AlertException

2.5.6.1 Constructor

+ NameRestrictException()	print error and set exception message

2.6 Package logic

2.6.1 class GameLogic

2.6.1.1 Field

- static List<Customer> customerContainer	Container for the customer in the game
- static List<Table> tableContainer	Container for the table in the game

- static List<Food> foodContainer	Container for the food in the game
- static List<ProgressBar> eatBarContainer	Container for the eatBar in the game
- static List<ProgressBar> cookBarContainer	Container for the cookBar in the game
- static List<Thread> threadContainer	Container for the Thread in the game
- static Customer[] waitArea	Container for the waiting area in the game
- static int player	Character that the player has selected which is 1 or 2
- static int money	Total money that the player has earned
- static int score	Heart score of this game start at 100 and the game end when it's <= 0
- static double tipMoney	Tip money to multiply which nomally is 1 except when the player choose Mom as a character which will earned more money
- static double extraWaitTime	More wait time to multiply which nomally is 1 except when the player choose Dad as a character which will make customer wait longer
- static boolean gameOver	True if the game is over, false otherwise
- static String playerName	The name of the player
- Customer newCustomer	When the customer is newly generate, it is add to this which wait for adding to the waitArea
- ImageView playerName	The image of the character that the player has chosen
- Thread generateCustomerThread	Thread for generating new customer

- boolean isGenerateCustomerStart	True if the generateCustomerThread is running, false otherwise

2.6.1.2 Constructor

+ GameLogic()	Initialize the game and the component in the game

2.6.1.3 Method

- void generateCustomer()	Generate new customer
- Customer getRandomCustomer(int x, int y)	Return new customer randomly
- int getAvailableWaitArea()	Get the position of the available space in the waitArea, return -1 if there is no available space
+ void logicUpdate()	Update everything in the game
+ static void addMoney(int money)	Add money to the player
+ static void addScore()	Add more extra Heart Score
+ static void deductScore()	Deduct extra Heart Score
+ static void endGame()	End the game by killing every thread
+ void setTipMoney(double tipMoney)	Setter for tipMoney
+ ImageView getPlayerImage()	Getter for playerImage
+ static List<Customer> getCustomerContainer()	Getter for customerContainer
+ static List<Table> getTableContainer()	Getter for tableContainer
+ static List<Food> getFoodContainer()	Getter for foodContainer

+ static List<ProgressBar> getEatBarContainer()	Getter for eatBarContainer
+ static List<ProgressBar> getcookBarContainer()	Getter for cookBarContainer
+ static List<Thread> getThreadContainer()	Getter for threadContainer
+ static Customer[] getWaitArea()	Getter for waitArea
+ static int getMoney()	Getter for money
+ static int getScore()	Getter for score
+ static void setPlayer(int player)	Setter for player
+ static int getPlayer()	Getter for player
+ static double getExtraWaitTime()	Getter for extraWaitTime
+ static boolean isGameOver()	Return true if the game is over, false otherwise
+ static String getPlayerName()	Getter for playerName
+ static void setPlayerName(String playerName)	Setter for playerName

2.7 Package scene

2.7.1 class SceneManater

2.7.1.1 Field

- static stage primaryStage	The main stage of the application
- static final int SCENE_WIDTH	The width of the stage (800px)
- static final int SCENE_HEIGHT	The height of the stage (600px)
- static WelcomeScene welcome	Root of the WelcomeScene
- static HowToScene howTo	Root of the HowToScene
- static HallOfFameScene highScore	Root of the HallOfFameScene

- static CharacterSelectScene characterSelect	Root of the CharacterSelectScene
- static GameScene game	Root of the GameScene
- static Scene welcomeScene	Scene of the WelcomeScene
- static Scene howToScene	Scene of the HowToScene
- static Scene highScoreScene	Scene of the HallOfFame
- static Scene characterSelectScene	Scene of the CharacterSelectScene
- static Scene gameScene	Scene of the GameScene

2.7.1.2 Method

+ static void initialize(Stage stage)	Initialize the game, going the WelcomeScene
+ static void gotoScene(String scene)	Set the primary stage to the scene the player is going to

2.7.2 class WelcomeScene extends StackPane

2.7.2.1 Field

- String logo_path	Image path for the logo of the game
- Image logo	Logo image of the game
- static Media welcomeBG	Background video of the welcomeScene
- static Media mainBGM	Background song of the welcomeScene
- static MediaPlayer welcomePlayer	MediaPlayer for the background video
- static MediaPlayer mainMisc	MediaPlayer for the music

2.7.2.2 Constructor

+ WelcomeScene()	Initialize the welcomeScene

2.7.2.3 Method

+ static MediaPlayer getVidPlayer()	Getter for background video player
+ static MediaPlayer getMiscPlayer()	Getter for music player

2.7.3 class HowToScene extends StackPane

2.7.3.1 Field

- static Media howtoBG	Background video of the how to scene
- static MediaPlayer howtoPlayer	MediaPlayer for the background video
- ImageView howtoPic	Image that display how to play overlay background
- int position	Number of current page
- NavigationButton nextButton	Button for go to next page
- NavigationButton backButton	Button for go to previous page

2.7.3.2 Constructor

+ HowToScene()	Initialize the howToScene

2.7.4 class HallOfFameScene extends StackPane

2.7.4.1 Field

- Label nameHeader	Label “Name” at header of score board
- Label scoreHeader	Label “Money” at header of score board
- Font HEADER_FONT	Font for header (nameHeader and scoreHeader)

- Image logo	Logo image
- static Media highScoreBG	Backgroud video of hall of fame scene
- static MediaPlayer highScorePlayer	MediaPlayer for background video
- GraphicsContext gc	GraphicsContext for scoreCanvas
- Canvas scoreCanvas	Canvas for display name and score
- Font SCORE_FONT	Font for name and score in scoreboard
- NavigationButton homeButton	Button for go back to welcome scene

2.7.4.2 Constructor

+ HallOfFameScene()	Initialize the hallOfFameScene

2.7.4.3 Method

+ static MediaPlayer getVidPlayer()	Getter for background video player

2.7.5 class CharacterSelectScene extends StackPane

2.7.5.1 Field

- int charID	ID of character that player selected
- String charName	Name of character that player set
- boolean isSelect1	Check if character 1 selected
- boolean isSelect2	Check if character 2 selected
- Pane charPane	This pane itself
- TextField nameTextField	TextField for player to input character name
- final Image logo	Logo image

- final String character1_notSelectPath	Path to image of character 1 normal
- final String character1_selectPath	Path to image of character 1 selected
- final String character1_mouseOnPath	Path to image of character 1 with detail
- final String character2_notSelectPath	Path to image of character 2 normal
- final String character2_selectPath	Path to image of character 2 selected
- final String character2_mouseOnPath	Path to image of character 2 with detail
- final Image nameTextFieldBackground	Background image of nameTextField
- GraphicsContext gc	GraphicsContext for logo canvas
- Font NAME_FONT	Font for nameTextField
- AudioClip mouseIn	AudioClip of sound when mouse in button
- AudioClip mouseClicked	AudioClip of sound when mouse clicked button
- static Media charSelBG	Background video of characterSelectScene
- static MediaPlayer charSelPlayer	Player of background video
- NavigationButton okButton	Button for confirm player name
- NavigationButton nextButton	Button for play
- NavigationButton backButton	Button for go back to welcome scene

2.7.5.2 Constructor

+ CharacterSelectScene()	Initialize the CharacterSelectScene

2.7.5.3 Method

- void setChar(int thisID, Canvas thisCanvas, int otherID, Canvas otherCanvas)	Set game character when either character 1 or character 2 is selected
- void drawCharacter(int ID, Canvas canvas)	Draw normal character
- void drawCharacterSelected(int ID, Canvas canvas)	Draw selected character
- void drawCharacterMouseOn(int ID, Canvas canvas)	Draw character with detail
- void setNavigationButtonEvent()	Override set Navigation button event
- void addTextLimiter(final TextField tf, final int maxLength)	Add input length limit to TextField (nameTextField)
+ TextField getNameTextField()	Getter of nameTextField
+ static MediaPlayer getVidPlayer()	Gerrter of MediaPlayer of background video
+ int getCharID	Getter of charID
+ String getCharName()	Getter of charName
+ boolean isSelect1()	Getter of isSelect1
+ boolean isSelect2()	Getter of isSelect2

2.7.6 class GameScene extends Pane

2.7.6.1 Field

- GameLogic gameLogic	GameLogic of this game
- static Pane mainGame	Pane of the game part (top part of the game)
- static Pane statusBar	Pane of the status bar of the game
- static Pane gameOverPane	Pane when it is game over

- static VBox gamePane	Pane that include both mainGame and statusBar
- final Image gameBG	Background image of the mainGame
- final Image statusBarBG	Background image of the statusBar
- final Image gameOverBG	Background image of the gameOverPane
- static Media gameBGM	Music of the game
- static MediaPlayer gameMisc	Media Player of gameBGM
- static MediaPlayer gameOverMisc	Sound when gameOver
- static AnimationTimer animation	AnimationTimer to update the game
- static Label statusMessage	Message to show at the statusBar
- Label clock	Clock showing time left in the statusBar

2.7.6.2 Constructor

+ GameScene()	Initialize the game when start

2.7.5.3 Method

- void setClock(long timePassed)	Set the time every second
- void setGameOver()	Set the Scene to show gameOver
+ static Pane getMainGame()	Getter for mainGame
+ static void setStatusMessage(String message)	Setter for status message
+ static MediaPlayer getGameMiscPlayer()	Getter for Game Music Player
+ static MediaPlayer getGameOverMiscPlayer()	Getter for Game Over Music Player