HighRich Project Report

Created by Azue Team
Thanawat Jierawatanakanok 6031020321
Nithipud Tunticharoenviwat 6031032921

2110215
Programming Methodology
Semester 1 Year 2018

Introduction

HighRich game is a simulation of being hotel owner. This game allow you to manage your hotel business. You have to make a decision to hire employees or to upgrade the room.One of the important factor that will lead you to win the game is popularity point. The popularity points will affect the number of customers visiting your hotel,so you have to be aware of losing popularity point. To win this game you have to upgrade all the room to highest class and you will lose the game if you can not finish the game in 15 days.

User guide

Start Scene

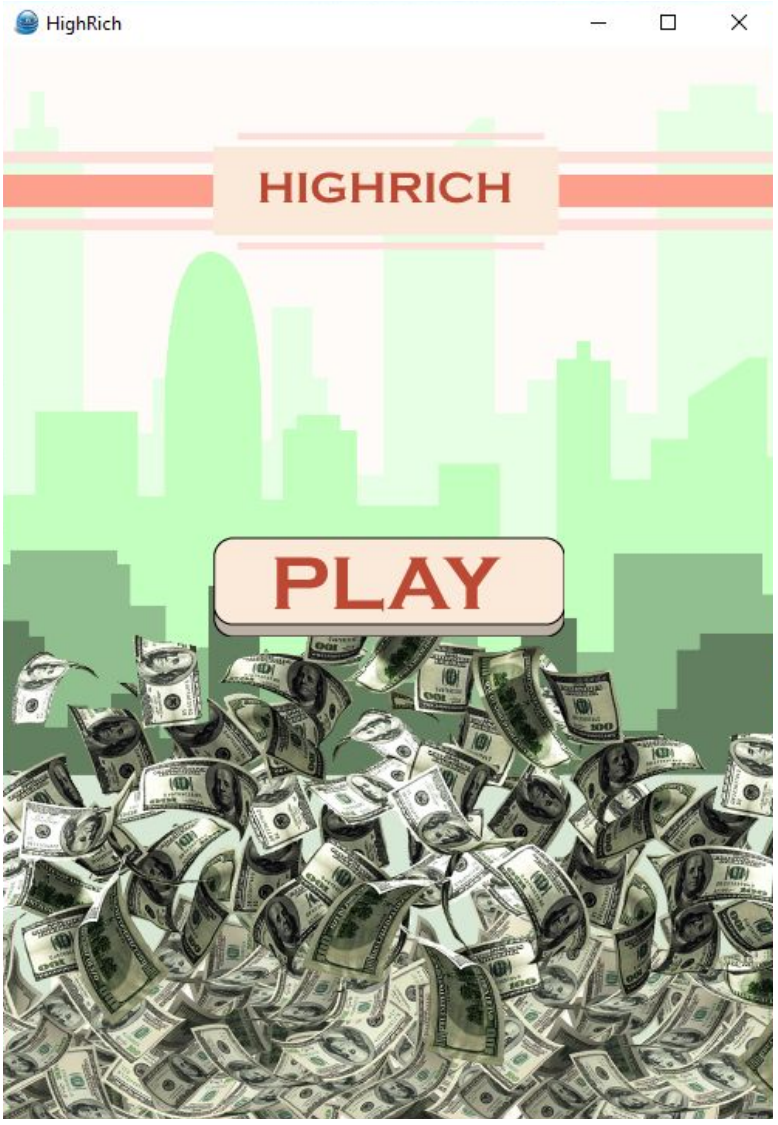

Figure 1: Start scene

To start the game you have to click the play button and then the window will display game scene.

Game scene

consist of two parts that are game screen and control bar.Game screen will be in the top part and control bar will be in the bottom part.

1. Game screen will display the movement of the component in the game and the changes of map.

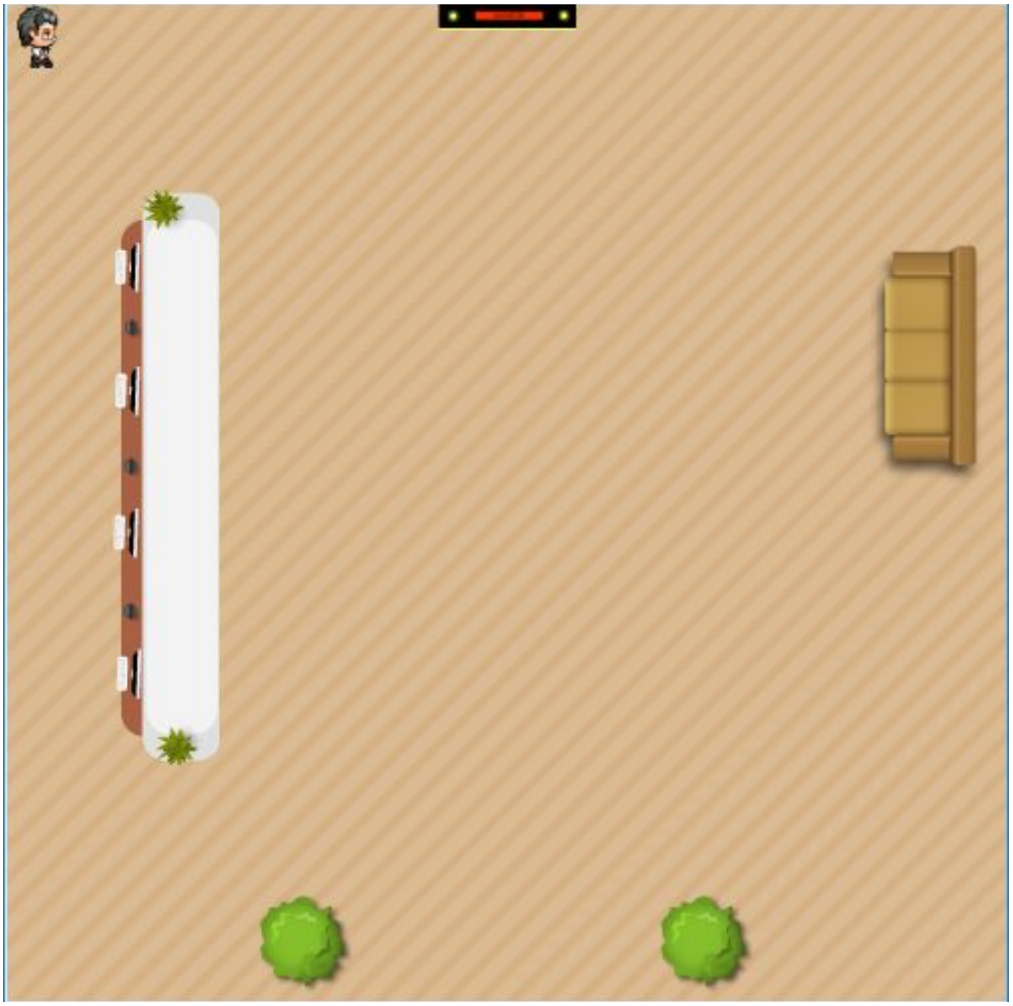

Figure 2: Game screen

2. Control bar will display the status of the player and the status of the game and also contain the button that will help you to control the game.

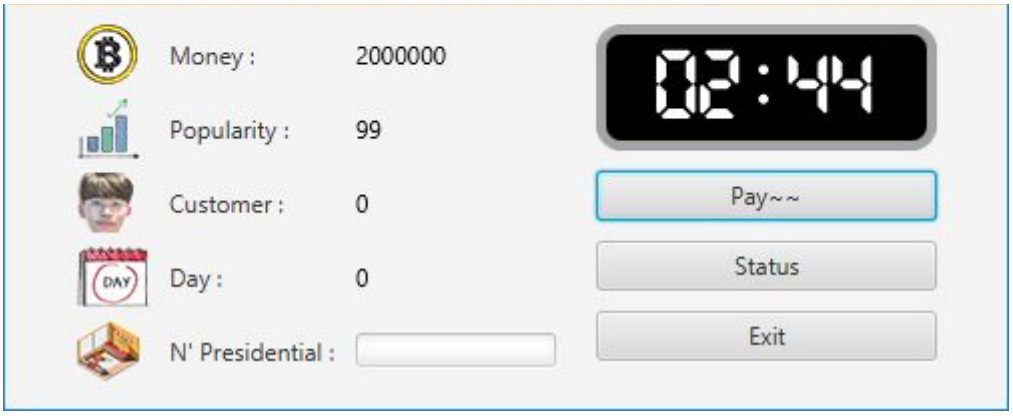

Figure 3: Control bar

2.1) Money

Show the player current money.

2.2) Popularity

Show the popularity point of your hotel. This will affect the number of customer visiting your hotel.When the customer come in to the lobby but you have no room for them or all the receptionists are busy the popularity point will be decreased by 1 but if you can serve them the popularity point will be increased by 1.

Minimum of popularity is 1.

Maximum of popularity is 200.

2.3) Customer

Show the number of customer in your hotel at the moment.

2.4) Day

Show the number of the day past since the game start.

2.5) N' Presidential

Show the progress bar of the number of presidential room(the highest class of the room) per the number of total room. You will win the game when the bar is completely full.

2.6) Time

Show the current time of the game.The customer will checkout at 12.00.The customer will not visit your hotel between 6.00 and 12.00.

2.7) Pay button

When you press pay button, the game will pop up the menu on the game screen and let you choose what to do.

Figure 4: Pop up of pay button menu

When you stand at tractor icon and click “Buy Room” button, the construction dusty will be rendered as Figure 5 after that you can see your new room as Figure 6.

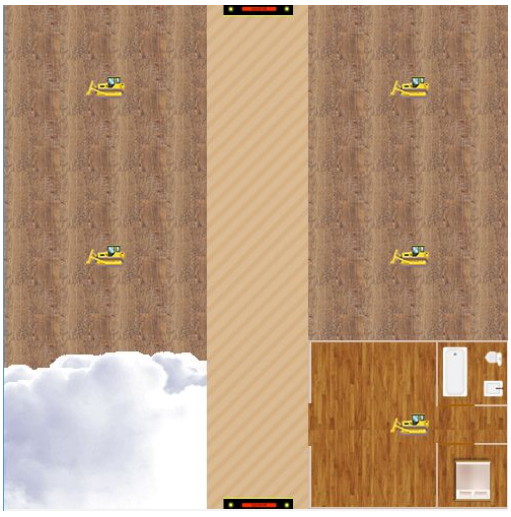

Figure 5: Render dusty

Figure 6: New room

When you click “Buy Receptionist” button then receptionist will be added at reception table.

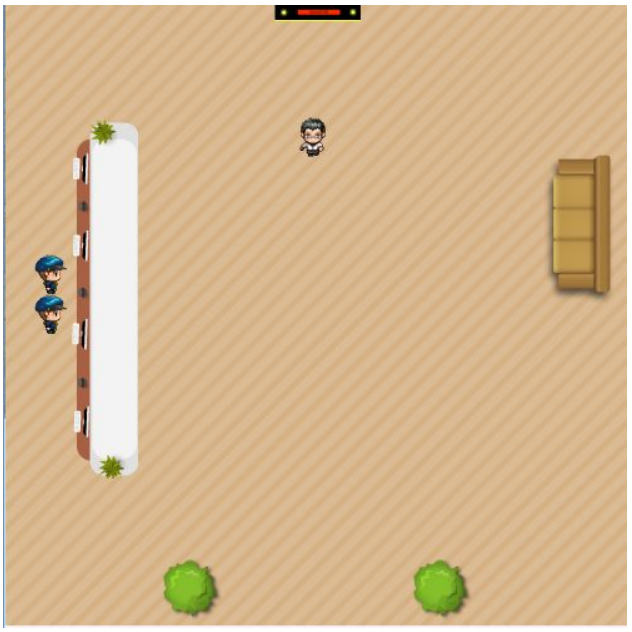

Figure 7: Add receptionist

2.8) Status button

When you press status button, the game will pop up the status on the game screen.

Figure 8: Pop up of status button

2.8.1) N’ Standard

Show the progress bar of the number of standard room.

2.8.2) N’ Executive

Show the progress bar of the number of executive room.

2.8.3) N’ presidential

Show the progress bar of the number of presidential room

2.8.4) Available Room

Show the progress bar of the number of available room.

2.8.5) N' Receptionist

Show the progress bar of the number of receptionist.

2.9) Exit button

When you press exit button, you will exit the game.

Event

There is an alien event, they will steal a half of your money. If you don't have any money they will gain nothing and walk out.

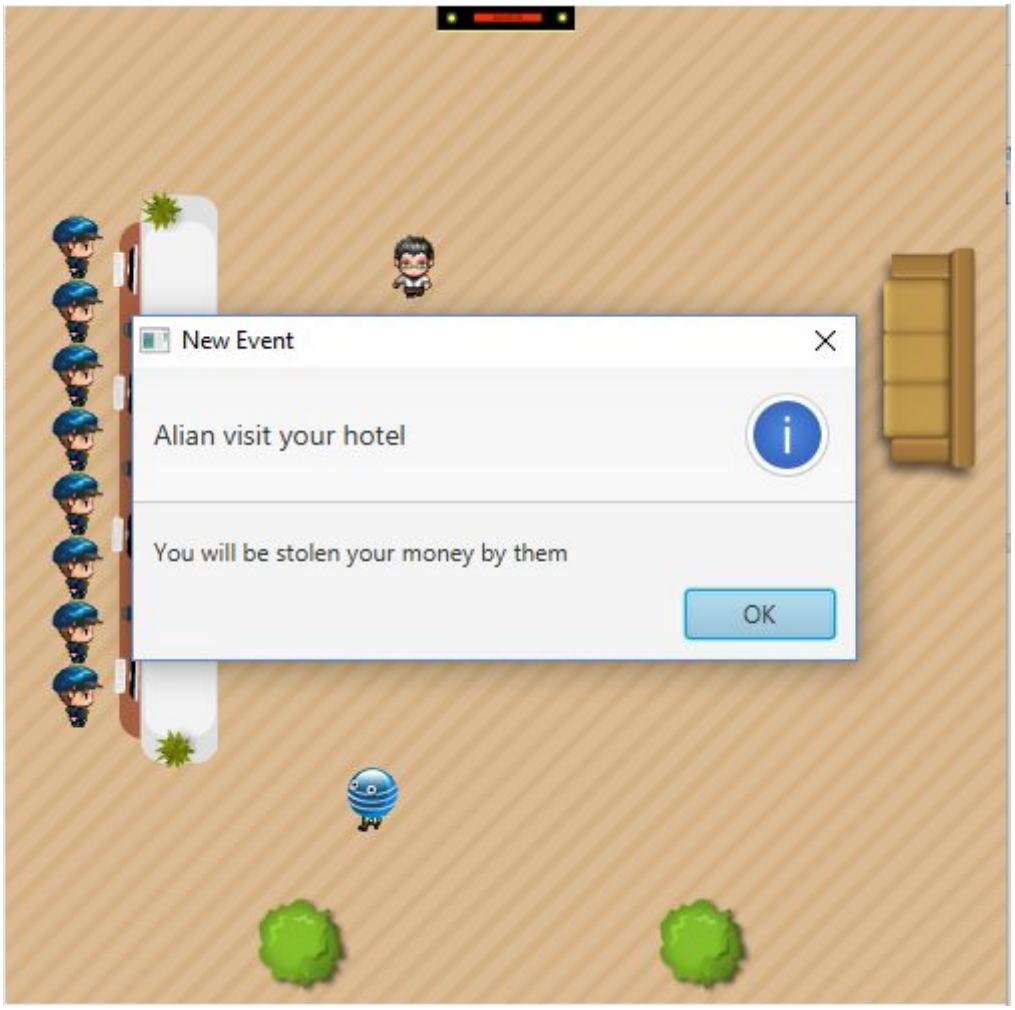

Figure 9: Alain visit your hotel

Moreover, there is a god event for giving you massive money. God will visit your hotel.If you can serve him, you will receive massive money.But if you don't have enough room and receptionist, you will miss the chance to get massive money.

Figure 10: God visit your hotel

Popup - Exception

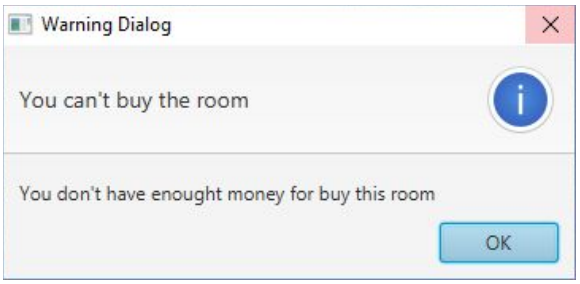

Figure 11: Don't have enough money to buy the room.

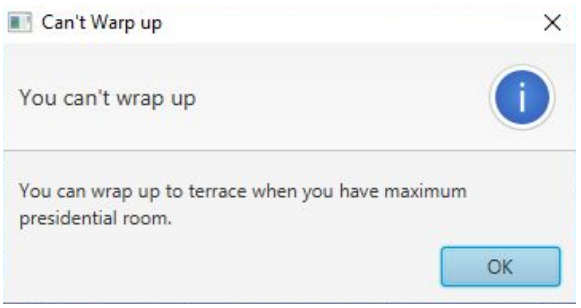

Figure 12: Can not warp to terrace

Figure 13: Don't have enough money to buy new receptionist.

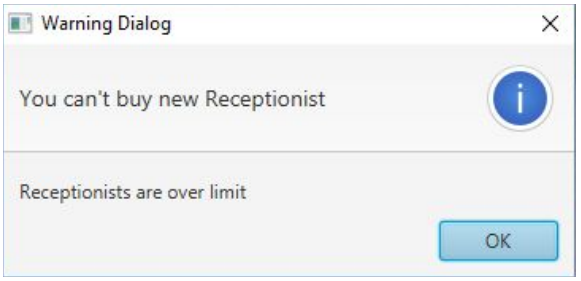

Figure 14: Can not buy more receptionist because the number of receptionists is over limit.

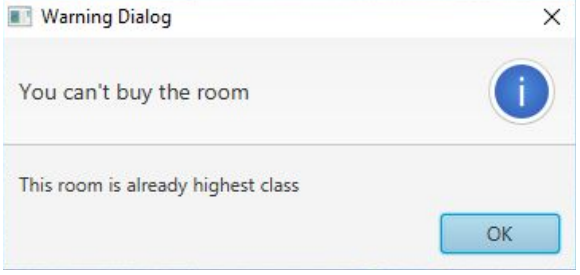

Figure 15: Can not upgraded because this room is highest class.

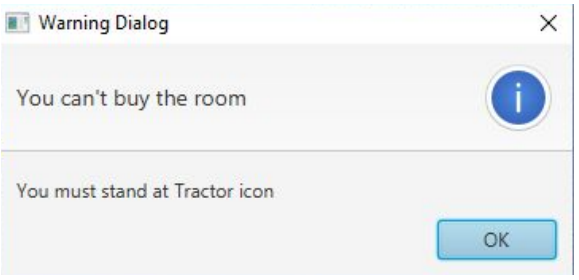

Figure 16: Can not buy or upgrade the room because you don't stand at tractor icon.

Game Win / Over

If you completely upgrade all the room to highest class, terrace map will be unlocked and you can go to see ending scene.

Figure 17: Ending Scene

When you can not end the game in 15 days, you will lose the game. You can click exit button and play again if you want.

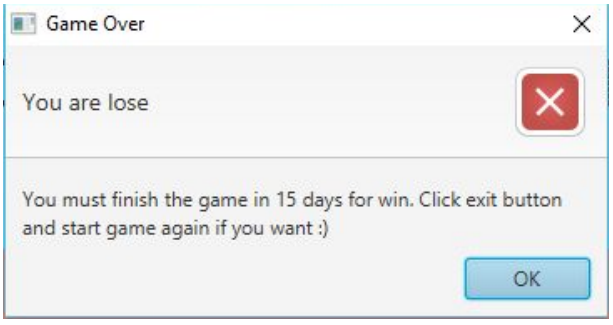

Figure 18: Lose the game

Game Controls

Action	Key Input
Move left	Arrow left
Move right	Arrow right
Move up	Arrow up
Move down	Arrow down
Other control	Press button on control bar

Character

 Player	This is your character.You can control this character and do the duty of hotel owner.
 ProgMeth	When ProgMeth appear,ProgMeth will steal your money for a half.
 God	When God appear and you can served him,you will received the room fee ten times higher than regular.
 Receptionist	Receptionist is a npc served the visitors.The cost to hire receptionist is 5000 Baht.
 Adult	Adult is a visitor.
 Teenage	Teenage is a visitor.
 OldMan	Oldman is a visitor.

Map

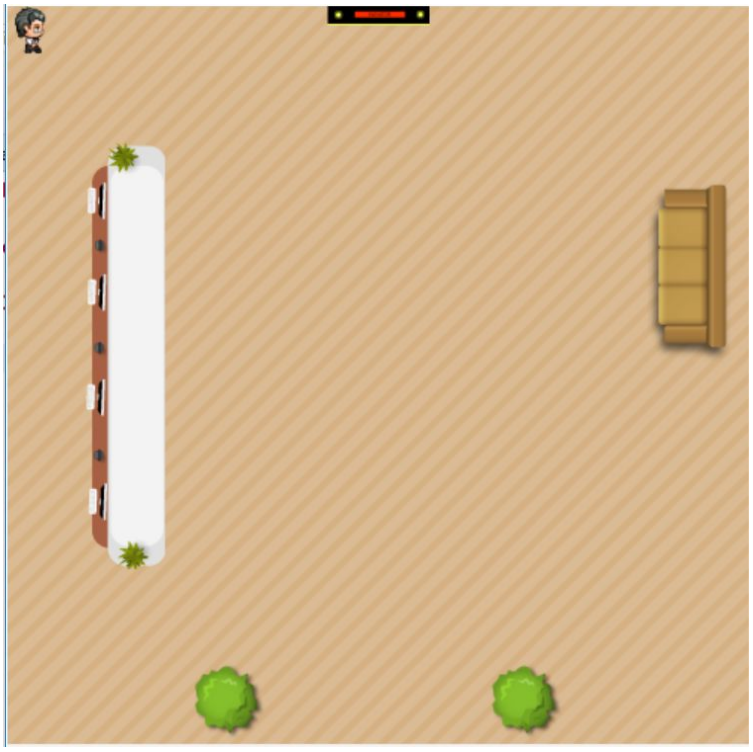

Figure 19: Map welcome

The visitor will come to map welcome first. If there is no receptionist to serve them or the room is full, they will walk out and the popularity point will be decreased by 1. But if you can serve them you will receive money from room fee and the popularity point will increased by 1.

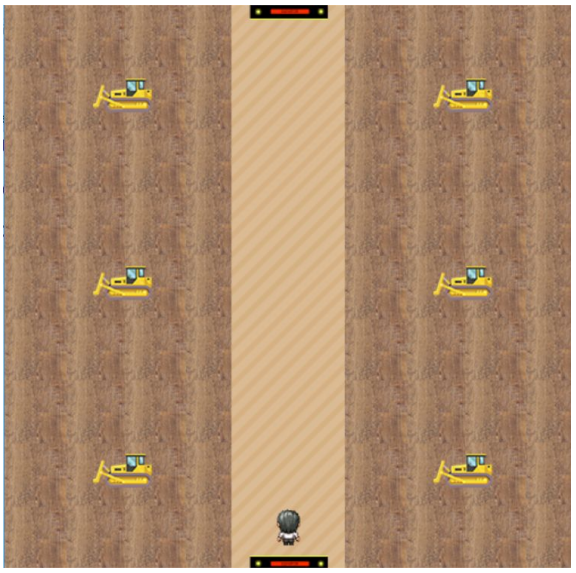

Figure 20: Map upstair

You can construct the room in map upstair by standing at the tractor icon and press pay button. There are 3 floor of map upstair in each floor you can construct the room up to 6 rooms.

Figure 21: Map terrace

Map terrace will be locked until you completely construct the presidential room all the floor.

Room

Room type	Construction cost	Fee
 Standard room	5,000 Baht	1,000 Baht
 Executive room	10,000 Baht	2,000 Baht
 Presidential room	20,000 Baht	4,000 Baht

Implementation Details

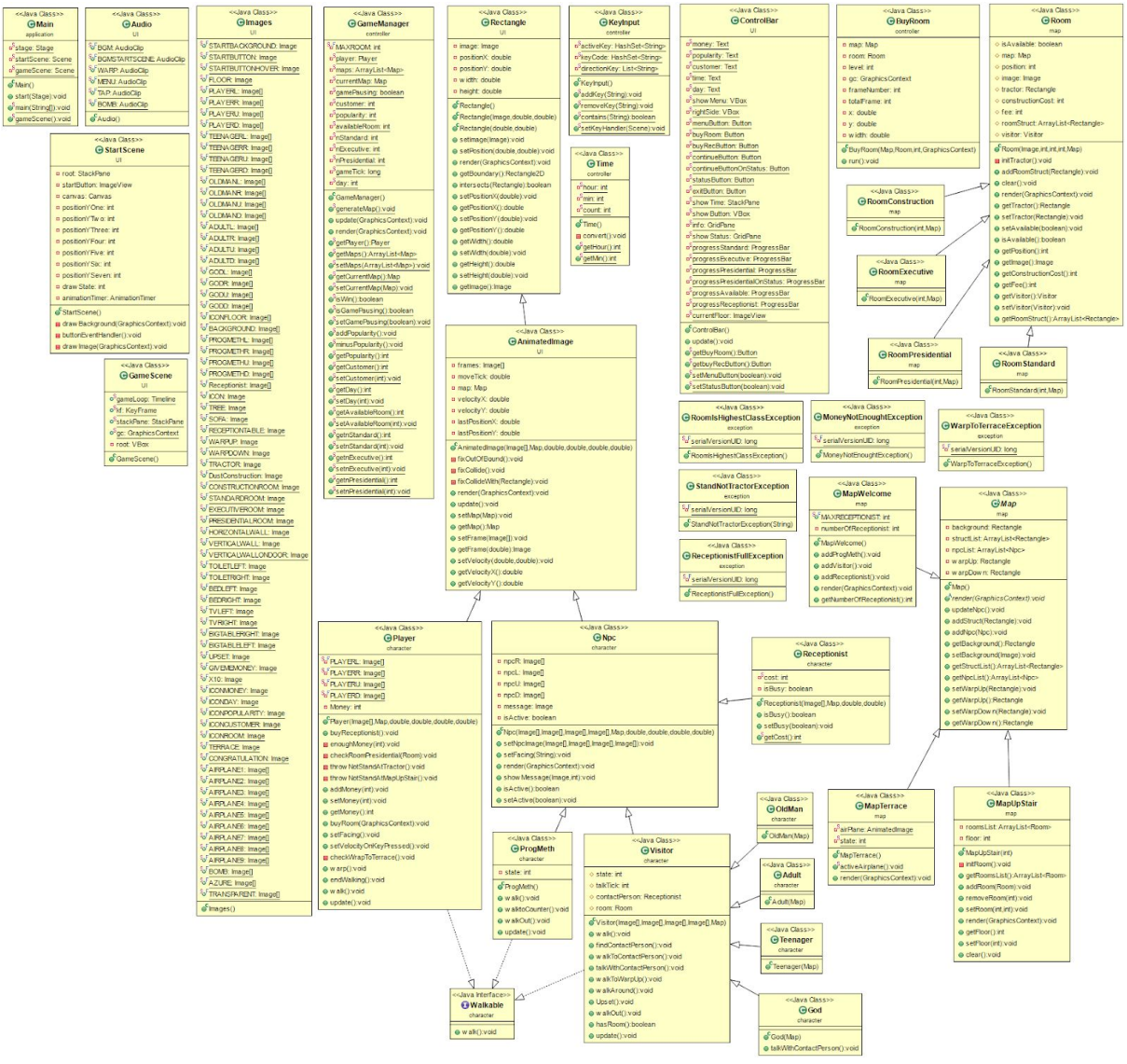

Figure 22: UML
See full image: https://prnt.sc/lr6ymw

* Noted that Access Modifier Notations can be listed below
    + (public)
    # (protected)
    - (private)
    static will be underlined.
    abstract will be italic

1. Package application

1.1 Class Main

1.1.1 Field

- Stage stage	This application's stage.
- Scene startScene	This application's start scene.
- Scene GameScene	This application's game scene.

1.1.2 Method

+ *void main(String[] args)*	An entry point of the application
+ void start(Stage primaryStage)	The main entry point of the JavaFX application.
+ *static void gameScene()*	Construct game scene and assign to stage.

2. Package character

2.1 Interface Walkable

2.1.1 Method

+ void walk()	

2.2 Class Player extends AnimatedImage implements Walkable

2.2.1 Field

- Image[] PLAYERL	Store player left side pictures.
- Image[] PLAYERR	Store player right side pictures.
- Image[] PLAYERU	Store player up side pictures.
- Image[] PLAYERD	Store player down side pictures.
- int Money	Store player current money.

2.2.2 Constructor

+ Player(Image[] frames, Map map, double positionX, double positionY, double velocityX, double velocityY)	Called super class constructor and initialize money for 20000.

2.2.3 Method

+ void buyReceptionist()	Buy the receptionist if it is able to buy.Otherwise,it will pop up alert.
- void enoughMoney(int n)	Check that player's current money is enough if not it will throw MoneyNotEnoughException.
- void checkRoomPresidential(Room room)	Check that the room is already presidential and throw RoomIsHighestClassException.
- void throwNotStandAtTractor()	Method for throw StandNotTractorException with message "tractor icon".

- void throwNotStandAtMapUpStair()	Method for throw StandNotTractorException with message “map up stair”.
+ void addMoney(int m)	Add money to player’s current money by given value.
+ void setMoney(int money)	Set player’s current money by given value.
+ int getMoney()	Get player’s current money.
+ void buyRoom(GraphicsContext gc)	Buy the room and initialise Buyroom class to play animation.
+ void setFacing()	Set facing of player depend on player’s velocity.
+ void setVelocityOnKeyPressed()	Set velocity when the user place the key.
+ void checkWrapToTerrace()	Check that if you can warp to terrace if not it will throw WarpToTerraceException.
+ void warp()	Warp player to another map.
+ void endWalking()	Set walking path of player when the game is end.
+ void walk()	Control the walking of player.
+ void update()	Update position of player.

2.3 Class Npc extends AnimatedImage

2.3.1 Field

- Image[] npcR	Store npc right side picture.
- Image[] npcL	Store npc left side picture.
- Image[] npcU	Store npc up side picture.
- Image[] npcD	Store npc down side picture.
- Image message	Store message image of npc.
- boolean isActive	Active status of npc.

2.3.2 Constructor

+ Npc(Image[] npcL, Image[] npcR, Image[] npcU, Image[] npcD, Map map, double positionX, double positionY, double velocityX, double velocityY)	Called super class constructor , set image of npc and set default value of isActive to true and message to null.

2.3.3 Method

+ void setNpcImage(Image[] npcL, Image[] npcR, Image[] npcU, Image[] npcD)	Set npc's image.
+ void setFacing(String s)	Set facing of npc by the given string such as "LEFT", "UP".
+ void render(GraphicsContext gc)	Draw npc and message on canvas.
+ void showMessage(Image image, int mills)	Set message by given image and show for given time millis.
+ boolean isActive()	Return true if npc is active,otherwise return false.
+ void setActive(boolean isActive)	Set active status of npc.

2.4 Class Receptionist extends Npc

2.4.1 Field

- int cost	Cost to hire receptionist.
- boolean isBusy	Status of receptionist if receptionist is serving customer isBusy is true,otherwise isBusy is false.

2.4.2 Constructor

+ Receptionist(Image[] image, Map map, double positionX, double positionY)	Called super class constructor and set default value of isBusy to false.

2.4.3 Method

+ boolean isBusy()	Return true if receptionist is busy,otherwise return false.
+ void setBusy(boolean isBusy)	Set isBusy by given boolean.
+ int getCost()	Get cost of receptionist.

2.5 Class Visitor extends Npc implements Walkable

2.5.1 Field

# int state	For specify which task the npc should do.
# int talkTick	For count the update loop.
# Receptionist contactPerson	Store which receptionist this npc should contact with.
# Room room	Store which room this npc should be in.

2.5.2 Constructor

+ Visitor(Image[] npcL, Image[] npcR, Image[] npcU, Image[] npcD, Map map)	Called super class constructor and set default value to all fields. - state = 0 - talkTick = 0 - contacPerson = null - room = null

2.5.3 Method

+ void walk()	Control the walking of visitor depend on state. - state 0: Find contact person. - state 1: Walk to contact person. - state 2: Talk with contact person. - state 3: Walk to warp up. - state 4: Walk around. - state -1: Upset - state -2: Walk out
+ void findContactPerson()	Find available receptionist and set contact person,set that receptionist to be busy and state = 1.But if there is no available receptionist state = -1.
+ void walkToContactPerson()	Walk to the front of contact person and state = 2.
+ void talkWithContactPerson()	Talk with contact person for 100 talk tick, pay money, add popularity point of hotel and state = 3.
+ void walkToWarpUp()	Walk to warp up, set isActive false for this visitor, add this visitor to the room and state = 4.
+ void walkAround()	Make visitor walk in their room.
+ void Upset()	Show upset message and state = -2.
+ void walkOut()	Walk out of hotel and disappear.

+ boolean hasRoom()	Return true if there is available room,otherwise return false.
+ void update()	Update this vistor.

2.6 Class Adult extends Visitor

2.6.1 Constructor

+ Adult(Map map)	Call super class constructor.

2.7 Class God extends Visitor

2.7.1 Constructor

+ God(Map map)	Call super class costructor and pop up alert that god has been appeared.

2.7.2 Method

+ void talkWithContactPerson()	Override method pay money 10 times more than usual.

2.8 Class OldMan extends Visitor

2.8.1 Constructor

+ OldMan(Map map)	Call super class constructor.

2.9 Class ProgMeth extends Visitor implements Walkable

2.9.1 Field

- int state	For specify which task the npc should do.

2.9.2 Constructor

+ ProgMeth()	Call super class constructor, set state to 0 and pop up alert that ProgMeth is appear.

2.9.3 Method

+ void walk()	Control the walking of ProgMeth depend on state. - state 0: Walk to counter. - state 1: Walk out.
+ void walktoCounter()	Walk to counter and decrease player money by half.
+ void walkOut()	Walk out and disappear.
+ void update()	Update this ProgMeth.

2.10 Class Teenager extends Visitor

2.10.1 Constructor

+ Teenager(Map map)	Call super class constructor.

3. Package controller

3.1 Class Buyroom

3.1.1 Field

- Map map	Store map upstair which want to construct room
- Room room	Store the room in this floor for construction
- int level	Store type of room level for upgrading
- GraphicsContext gc	Store GraphicsContext for draw dusty after click upgrading or buy.
- int frameNumber	Store current frame of dusty
- int totalFrame	Store total frames of dusty animation
- double x	Store position in x axis of this room
- double y	Store position in y axis of this room
- double width	Store width of dusty picture each frame

3.1.2 Constructor

+ BuyRoom(Map map, Room room, int level, GraphicsContext gc)	set map, room, level, gc and calculate position in x,y axis from room position then set totalFrame, frameNumber and width from dusty picture for draw dusty.

3.1.3 Method

+ void run()	Override method run() that draw dusty animation

3.2 Class GameManager

3.2.1 Field

- int MAXROOM	Constant variable declare the maximum number of rooms equal to 18.
- Player player	Player of the game.
- ArrayList<Map> maps	Store all the map in this game.
- Map currentMap	Current map showing on game screen.
- boolean gamePausing	Value is true when game is paused, otherwise false.
- int customer	The number of customers in hotel at the moment.
- int popularity	The popularity point of the hotel.
- int availableRoom	The number of available rooms at the moment.
- int nStandard	The number of standard rooms have been constructed.
- int nExecutive	The number of executive rooms have been constructed.
- int nPresidential	The number of presidential rooms have been constructed.
- long gameTick	For count the update loop.
- int day	The number of day past since game start.

3.2.2 Constructor

+ GameManager()	Initialise field and set default value to variable.

3.2.3 Method

+ void generateMap()	Generate maps and store in ArrayList.
+ void update(GraphicsContext gc)	Update map, player. npc and generate visitor randomly.
+ void render(GraphicsContext gc)	Draw player, map, components in canvas.
+ Player getPlayer()	Getter of player.
+ ArrayList<Map> getMaps()	Getter of maps.
+ void setMaps(ArrayList<Map> maps)	Setter of maps.
+ Map getCurrentMap()	Getter of currentMap.
+ void setCurrentMap(Map currentMap)	Setter of currentMap.
+ boolean isWin()	Return true if the number of presidential room equal to MAXROOM, otherwise return false.
+ boolean isGamePausing()	Return true if game is pausing, otherwise return false.
+ void setGamePausing(boolean gamePausing)	Setter of gamePausing.
+ void addPopularity()	Increase popularity points by 1.
+ void minusPopularity()	Decrease popularity points by 1.
+ int getPopularity()	Getter of popularity.
+ int getCustomer()	Getter of customer.
+ void setCustomer(int n)	Setter of customer.
+ int getDay()	Getter of day.
+ void setDay(int day)	Setter of day.
+ int getAvailableRoom()	Getter of availableRoom.

+ void setAvailableRoom(int availableRoom)	Setter of availableRoom.
+ int getnStandard()	Getter of nStandard.
+ void setnStandard(int nStandard)	Setter of nStandard.
+ int getnExecutive()	Getter of nExecutive.
+ void setnExecutive(int nExecutive)	Setter of nExecutive.
+ int getnPresidential()	Getter of nPresidential.
+ void setnPresidential(int nPresidential)	Setter of nPresidential.

3.3 Class KeyInput

3.3.1 Field

- HashSet<String> activeKey	HashSet contains input from keyboard that will be active in game.
- HashSet<String> keyCode	HashSet contains all input from keyboard at the moment.
- List<String> directionKey	List contains keyboard code that be able to press.

3.3.2 Method

+ void addKey(String)	Add input from key board to HashSet.
+ void removeKey(String)	Remove keycode from Hashset.
+ boolean contains(String)	Return true if activeKey contains given keycode.
+ void setKeyHandler(Scene)	Set event handler to the given scene.

3.4 Class Time

3.4.1 Field

- int hour	Current hour.
- int min	Current minute.
- int count	For counting the loop.

3.4.2 Constructor

+ Time()	Use thread to run the loop every 100 milliseconds.

3.4.3 Method

- void convert()	Convert count to hour and minute.
+ int getHour()	Getter of hour.
+ int getMin()	Getter of min.

4. Package exception

4.1 Class MoneyNotEnoughtException extends Exception

4.1.1 Field

- long serialVersionUID	-2525558838547083303L

4.1.2 Constructor

+ MoneyNotEnoughtException()	print “not have enough money”.

4.2 Class ReceptionistFullException extends Exception

4.2.1 Field

- long serialVersionUID	-3361896826591043910L

4.2.2 Constructor

+ ReceptionistFullException()	print “full receptionist”.

4.3 Class RoomIsHighestClassException extends Exception

4.3.1 Field

- long serialVersionUID	9045970519910759269L

4.3.2 Constructor

+ RoomIsHighestClassException()	print ”room is highest class”.

4.4 Class StandNotTractorException extends Exception

4.4.1 Field

- long serialVersionUID	4425178354312483007L

4.4.2 Constructor

+ StandNotTractorException(String s)	print “Stand not” concatenated with given string.

4.5 Class WarpToTerraceException extends Exception

4.5.1 Field

- long serialVersionUID	-2191423090320604677L

4.5.2 Constructor

+ WarpToTerraceException()	print “Can’t warp to Terrace”.

5. Package map

5.1 Class Map

5.1.1 Field

- Rectangle background	Background of this map.
- ArrayList<Rectangle> structList	Contain structure of this map that player and npc can't walk through.
- ArrayList<Npc> npcList	Contain npc in this map.
- Rectangle warpUp	Warp up point of this map.
- Rectangle warpDown	Warp down point of this map.

5.1.2 Constructor

+ Map()	Initialize all the fields.

5.1.3 Method

+ *void render(GraphicsContext)*	Abstract method.
+ void updateNpc()	Update all the npc in npcList and remove inactive npc.
+ void addStruct(Rectangle rectangle)	Add structure to structList.
+ void addNpc(Npc npc)	Add npc to npcList.
+ Rectangle getBackground()	getter of background.
+ void setBackground(Image i)	setter of background.
+ ArrayList<Rectangle> getStructList()	getter of structList.
+ ArrayList<Npc> getNpcList()	getter of npcList.
+ void setWarpUp(Rectangle warpUp)	setter of warpUp.
+ Rectangle getWarpUp()	getter of warpUp.
+ void setWarpDown(Rectangle	setter of warpDown.

warpDown)	
+ Rectangle getWarpDown()	getter of warpDown.

5.2 Class MapTerrace extends Map

5.2.1 Field

- AnimatedImage airPlane	Animated image of aiplane.
- int state	For specify movement of airplane.

5.2.2 Constructor

+ MapTerrace()	call super constructor, set background and initialise airPlane.

5.2.3 Method

+ void activeAirplane()	Make airplane move depend on state.
+ void render(GraphicsContext gc)	Draw component in this map in canvas.

5.3 Class MapUpStair extends Map

5.3.1 Field

- ArrayList<Room> roomsList	Store all the rooms in this map.
- int floor	Floor number of this map.

5.3.2 Constructor

+ MapUpStair(int floor)	Call super class constructor, initialise room and set background, warpUp, warpDown.

5.3.3 Method

- void initRoom()	Initialise room and add to roomsList.
+ ArrayList<Room> getRoomsList()	Getter of roomsList.

+ void addRoom(Room o)	Add given room to roomsList.
+ void removeRoom(int position)	Remove given room from roomsList.
+ void setRoom(int position, int level)	Set the room at given position and type of the room will according to given level. - level 0: RoomConstruction - level 1: RoomStandard - level 2: RoomExecutive - level 3: RoomPresidential
+ void render(GraphicsContext gc)	Draw all components in this map on canvas.
+ int getFloor()	Getter of floor.
+ void setFloor(int floor)	Setter of floor.
+ void clear()	Call clear() of all room, remove all npc from npcList and set cutsomer to 0.

5.4 Class MapWelcome extends Map

5.4.1 Field

+ int MAXRECEPTIONIST	Constant variable of the maximum number of receptionist equal to 8.
- int numberOfReceptionist	The number of receptionist you have.

5.4.2 Constructor

+ MapWelcome()	Call super class constructor, add structure to this map and set background and warpUp.

5.4.3 Method

+ void addProgMeth()	Add Progmeth to this map.
+ void addVisitor()	Add visitor to this map by random the type of visitor.

+ void addReceptionist()	Add receptionist to this map if the number of receptionist not exceed maximum.
+ void render(GraphicsContext gc)	Draw all components in this map on canvas.
+ int getNumberOfReceptionist()	Getter of numberOfReceptionist.

5.5 Class Room

5.5.1 Field

# boolean isAvailable	The value is true if the room is available, otherwise false.
# Map map	The map that this room belong to.
# int position	The position of this room, it is the number from 0 up to 6.
# Image image	The image of this room.
# Rectangle tractor	The tractor icon use for check intersection with player when upgrade the room.
# int constructionCost	The construction cost of this room.
# int fee	The fee of this room.
# ArrayList<Rectangle> roomStruct	Store the structure of this room.
# Visitor visitor	The visitor that rest in this room.

5.5.2 Constructor

+ Room(Image image, int position, int constructionCost, int fee, Map map)	Initialise field and assign default value.

5.5.3 Method

- void initTractor()	Generate tractor icon to the center of

	the room.
+ addRoomStruct(Rectangle s)	Add structure to the roomStruct.
+ void clear()	Clear the visitor and set this room available.
+ void render(GraphicsContext gc)	Draw all components of this room on canvas.
+ Rectangle getTractor()	Getter of tractor.
+ void setTractor(Rectangle tractor)	Setter of tractor.
+ void setAvailable(boolean b)	Set isAvailable by the given boolean.
+ boolean isAvailable()	Return true if room is available, otherwise return false.
+ int getPosition()	Getter of position.
+ Image getImage()	Getter of Image.
+ getConstructionCost()	Getter of constructionCost.
+ int getFee()	Getter of fee.
+ Visitor getVisitor()	Getter of visitor.
+ void setVisitor(Visitor visitor)	Setter of visitor.
+ ArrayList<Rectangle> getRoomStruct()	Getter of roomStruct.

5.6 Class RoomConstruction extends Room

5.6.1 Constructor

+ RoomConstruction(int position, Map map)	Call super class constructor and set isAvailable false.

5.7 Class RoomExecutive extends Room

5.7.1 Constructor

+ RoomExecutive(int position, Map map)	Call super class constructor, set isAvailable according to the former room and add room structure.

5.8 Class RoomPresidential extends Room

5.8.1 Constructor

+ RoomPresidential(int position, Map map)	Call super class constructor, set isAvailable according to the former room and add room structure.

5.9 Class RoomStandard extends Room

5.9.1 Constructor

+ RoomStandard(int position, Map map)	Call super class constrctor,set isAvailable true and add room structure.

6. Package UI

6.1 Class Rectangle

6.1.1 Field

- Image image	The image of object.
- double positionX	The position on x-axis of object.
- double positionY	The position on y-axis of object.
- double width	The width of object.
- double height	The height of object.

6.1.2 Constructor

+ Rectangle()	Construct rectangle at (0,0) and width and height equal to 0.
+ Rectangle(Image i, double positionX, double positionY)	Construct rectangle at (positionX, positionY), set image and set width and height according to the size of image.
+ Rectangle(double positionX, double positionY)	Construct rectangle at (positionX, postionY) and width and height equal to 0.

6.1.3 Method

+ void setImage(Image i)	Set image by given image and set width and height according to the size of image.
+ setPosition(double x, double y)	Set the positionX and positionY by given value.
+ void render(GraphicsContext gc)	Draw this image on canvas.
+ Rectangle2D getBoundary()	Return Rectangle2D object that construct from positionX, positionY, width and height value of this class.
+ boolean intersects(Rectangle s)	Return true if this intersect with given Rectangle.
+ void setPositionX(double positionX)	Setter of positionX.
+ double getPositionX()	Getter of positionX.
+ void setPositionY(double positionY)	Setter of positionY.
+ double getPositionY()	Getter of positionY.
+ double getWidth()	Getter of width.
+ void setWidth(double width)	Setter of width.
+ double getHeight()	Getter of height.
+ void setHeight(double height)	Setter of height.

+ Image getImage()	Getter of image.

6.2 Class AnimatedImage extends Rectangle

6.2.1 Field

- Image[] frames	Store frames of image that will use to make animation.
- double moveTick	For count the update loop and use in changing frames.
- Map map	The map that this AnimatedImage belong to.
- double velocityX	The velocity on x-axis of this AnimatedImage.
- double velocityY	The velocity on y-axis of this AnimatedImage.
- double lastPositionX	The previous position on x-axis.
- double lastPositionY	The previous position on y-axis.

6.2.2 Constructor

+ AnimatedImage(Image[] frames,Map map, double positionX, double positionY, double velocityX, double velocityY)	Call super class constructor and set all the field.

6.2.3 Method

- void fixOutOfBound()	Fix the position if this AnimatedImage is out of boundary.
- void fixCollide()	Call fixCollideWith method to fix position if this AnimatedImage collide with structure,npc or player.(Not including itself).
- void fixCollideWith(Rectangle s)	Fix the postion if this AnimatedImage collide with given Rectangle.

+ void render(GraphicsContext gc)	Set the appropriate frame and draw image on canvas.
+ void update()	Update the last position and current postion, fix out of bound, fix collide with other object and then increase moveTick by 1.
+ void setMap(Map map)	Setter of map
+ Map getMap()	Getter of map
+ void setFrame(Image[] frames)	Setter of frames
+ Image getFrame(double moveTick)	Get one of image from frames by using moveTick to specify the index of image in array.
+ void setVelocity(double x, double y)	Setter of Velocity
+ double getVelocityX()	Getter of Velocity on axis X
+ double getVelocityY()	Getter of Velocity on axis Y

6.3 Class Audio

6.3.1 Field

+ AudioClip BGM	Store Audio of background music
+ AudioClip BGMSTARTSCENE	Store background music that play on StartScene
+ AudioClip WARP	Store Warp sound
+ AudioClip MENU	Store audio that play when click on menu button
+ AudioClip TAP	Store audio that play when enter your mouse on Start button in StartScene
+ AudioClip BOMB	Store audio that play after end the game

6.4 Class ControlBar extends HBox

6.4.1 Field

- Text money	Text for amount of player's current money.
- Text popularity	Text for popularity point of hotel.
- Text customer	Text for the number of customer in hotel at the moment.
- Text time	Text for time in game.
- Text day	Text for the number of day past since game start.
- VBox showMenu	VBox pane of pop up menu when press Pay button.
- VBox rightSide	VBox pane for the right hand side of control bar.
- Button menuButton	Button for paying menu.
- Button buyRoom	Button for buying room
- Button buyRecButton	Button for buying receptionist
- Button continueButton	Button for continue the game.
- Button continueButtonOnStatus	Button for continue the game.
- Button statusButton	Button for status menu.
- Button exitButton	Button for exit the game.
- StackPane showTime	StackPane for time of the game.
- VBox showButton	VBox to store button on the right hand side of the control bar.
- GridPane info	GridPane contain information of the game on the left hand side of

	control bar.
- GridPane showStatus	GridPane of pop up menu when press status button.
- ProgressBar progressStandard	ProgressBar for the number of standard room per the number of total room.
- ProgressBar progressExecutive	ProgressBar for the number of executive room per the number of total room.
- ProgressBar progressPresidential	ProgressBar for the number of presidential room per the number of total room.
- ProgressBar progressPresidentialOnStatus	ProgressBar for the number of presidential room per the number of total room.
- ProgressBar progressAvailable	ProgressBar for the number of available room.
- ProgressBar progressReceptionist	ProgressBar show thhe number of receptionist.
- ImageView currentFloor	The image of number corresponding to the current floor.

6.4.2 Constructor

+ ControlBar()	Initialise all field and add to the main screen.

6.4.3 Method

+ void update()	Update the value of text and progress bar.
+ Button getBuyRoom()	get buyRoom button.
+ Button getBuyRecButton()	get buyRec button.

+ setMenuButton(boolean bool)	set MenuButton to disable
+ setStatusButton(boolean bool)	set StatusButton to disable

6.5 Class GameScene extends Scene

6.5.1 Field

+ Timeline gameLoop	Use to process keyframe at specified time interval elapsed.
+ KeyFrame kf	Store keyframe to work with gameloop.
+ StackPane stackPane	StackPane for the game screen(top part of application).
+ GraphicsContext gc	Used to issue draw to a Canvas
- VBox root	Root of GameScene.

6.5.2 Constructor

+ GameScene()	Initialise field, play theme sound and start gameloop.

6.6 Class StartScene extends Scene

6.6.1 Field

- StackPane root	Root of StartScene.
- ImageView startButton	Play button.
- Canvas canvas	Canvas for draw StartScene.
- int positionYOne	position in y axis of background slide number one.
- int positionYTwo	position in y axis of background slide number two.
- int positionYThree	position in y axis of background slide number three.
- int positionYFour	position in y axis of background slide number four.

- int positionYFive	position in y axis of background slide number five.
- int positionYSix	position in y axis of background slide number six.
- int positionYSeven	position in y axis of background slide number seven.
- int drawState	store progress of background slide.
- AnimationTimer animationTimer	Create the loop to play animation in StartScene.

6.6.2 Constructor

+ StartScene()	Initialise field and add event handler for play button.

6.6.3 Method

- void drawBackground(GraphicsContext gc)	move background image to positionYx and set drawState when move complete
- void buttonEventHandler()	Method to set event handler for button.
- void drawImage(GraphicsContext gc)	draw background image on GraphicsContext

6.7 Class Images

6.7.1 Field

+ Image STARTBACKGROUND	Store image of background in start scene.
+ Image STARTBUTTON	Store image of play button in start scene.
+ Image STARTBUTTONHOVER	Store image of play button hover in start scene.
+ Image FLOOR	Store image of floor.
+ Image[] PLAYERL	Store images of player left side.
+ Image[] PLAYERR	Store images of player right side.
+ Image[] PLAYERU	Store images of player up side.
+ Image[] PLAYERD	Store images of player down side.
+ Image[] TEENAGERL	Store images of teenager left side.
+ Image[] TEENAGERR	Store images of teenager right side.
+ Image[] TEENAGERU	Store images of teenager up side.
+ Image[] TEENAGERD	Store images of teenager down side.
+ Image[] OLDMANL	Store images of old man left side.
+ Image[] OLDMANR	Store images of old man right side.
+ Image[] OLDMANU	Store images of old man up side.
+ Image[] OLDMAND	Store images of old man down side.
+ Image[] ADULTL	Store images of adult left side.
+ Image[] ADULTR	Store images of adult right side.
+ Image[] ADULTU	Store images of adult up side.
+ Image[] ADULTD	Store images of adult down side.

+ Image[] GODL	Store images of god left side.
+ Image[] GODR	Store images of god right side.
+ Image[] GODU	Store images of god up side.
+ Image[] GODD	Store images of god down side.
+ Image[] ICONFLOOR	Store images of floor icon.
+ Image[] BACKGROUND	Store images of background start scene.
+ Image[] PROGMETHL	Store images of progmeth left side.
+ Image[] PROGMETHR	Store images of progmeth right side.
+ Image[] PROGMETHU	Store images of progmeth up side.
+ Image[] PROGMETHD	Store images of progmeth down side.
+ Image[] Receptionist	Store images of receptionist.
+ Image ICON	Store image of game icon.
+ Image TREE	Store image of tree.
+ Image SOFA	Store image of sofa.
+ Image RECEPTIONTABLE	Store image of reception table.
+ Image WARPUP	Store image of warp up.
+ Image WARPDOWN	Store image of warp down.
+ Image TRACTOR	Store image of tractor.
+ Image DustConstruction	Store image of dust construction.
+ Image CONSTRUCTIONROOM	Store image of construction room.
+ Image STANDARDROOM	Store image of standard room.
+ Image EXECUTIVEROOM	Store image of executive room.
+ Image PRESIDENTIALROOM	Store image of presidential room.
+ Image HORIZONTALWALL	Store image of horizontal wall.

+ Image VERTICALWALL	Store image of vertical wall.
+ Image VERTICALWALLONDOOR	Store image of vertical wall with door.
+ Image TOILETLEFT	Store image of toilet on left side.
+ Image TOILETRIGHT	Store image of toilet on right side.
+ Image BEDLEFT	Store image of bed on left side.
+ Image BEDRIGHT	Store image of bed on right side.
+ Image TVLEFT	Store image of tv on left side.
+ Image BIGTABLERIGHT	Store image of big table on right side.
+ Image BIGTABLELEFT	Store image of big table on left side.
+ Image UPSET	Store image of “upset” message.
+ Image GIVEMEMONEY	Store image of “give me money” message.
+ Image X10	Store image of “X10” message.
+ Image ICONMONEY	Store image of money icon.
+ Image ICONDAY	Store image of day icon.
+ Image ICONPOPULARITY	Store image of popularity icon.
+ Image ICONCUSTOMER	Store image of customer icon.
+ Image ICONROOM	Store image of room icon.
+ Image TERRACE	Store image of terrace map.
+ Image CONGRATULATION	Store image of “CONGRATULATION” text.
+ Image[] AIRPLANE1	Store image of air plane.
+ Image[] AIRPLANE2	Store image of air plane.
+ Image[] AIRPLANE3	Store image of air plane.
+ Image[] AIRPLANE4	Store image of air plane.

+ Image[] AIRPLANE5	Store image of air plane.
+ Image[] AIRPLANE6	Store image of air plane.
+ Image[] AIRPLANE7	Store image of air plane.
+ Image[] AIRPLANE8	Store image of air plane.
+ Image[] AIRPLANE9	Store image of air plane.
+ Image[] BOMB	Store image of bomb.
+ Image[] AZURE	Store image of AZURE text.
+ Image[] TRANSPARENT	Store transparent image.

6.7.2 Static Block

Set ICONFLOOR, BACKGROUND, TEENAGERL, TEENAGERR, TEENAGERU, TEENAGERD, OLDMANL, OLDMANR, OLDMANU, OLDMAND, ADULTL, ADULTR, ADULTU, ADULTD, PLAYERL, PLAYERR, PLAYERU, PLAYERD, PROGMETHL, PROGMETHR, PROGMETHU, PROGMETHD, GODL, GODR, GODU and GODD images to array of each variables with ordered.