WeLoveProgMeth_NoCap Documentation

Created by

Vorraphan Treepoonsap 6531340121

Sirapatch Thammaleelakul 6532187721

2110215 Programming Methodology

Semester 2 Year 2022

Chulalongkorn University

Crozzy Furry

Introduction

Crozzy Furry is inspired by a popular mobile game named “Crossy Road”. The objective of this game is to guide your character safely across a busy road, avoiding obstacles and vehicles, to reach the other side without getting hit.

Rules

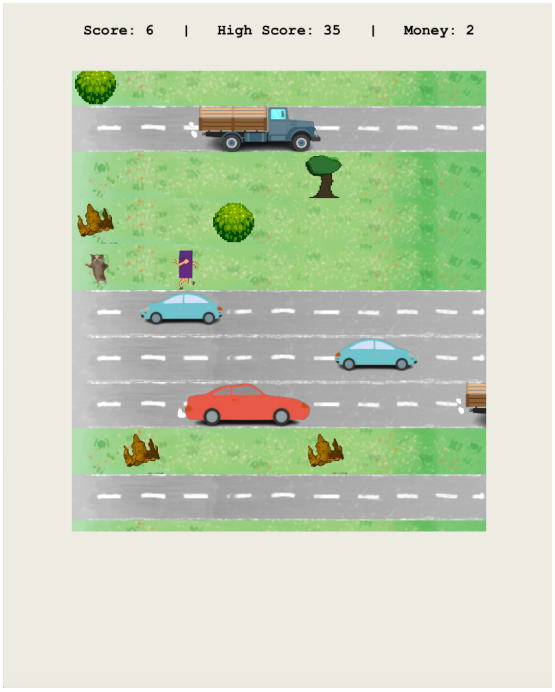
Players must navigate their character avoiding cars, trucks, and other obstacles with randomly generated layouts. If the character gets hit by a vehicle or hits the bottom end of the screen, it is game over. The game continues endlessly, challenging players to achieve high scores by crossing as many roads and obstacles as possible. The score is counted by the number of lanes a character crossed.

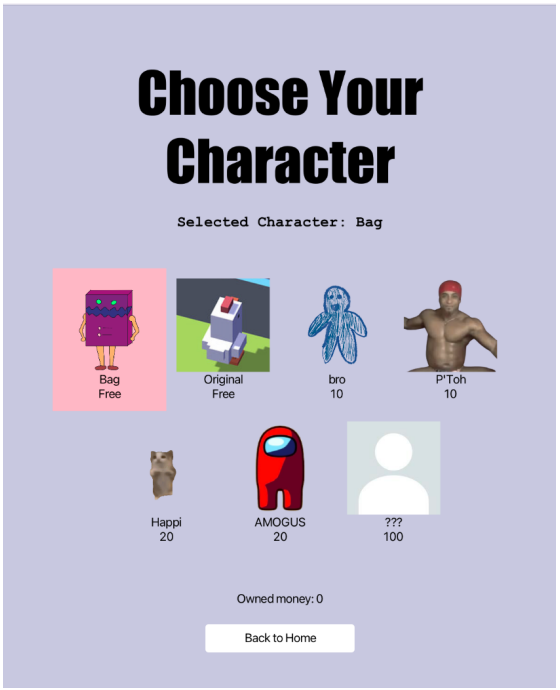
Example

On start, the character is placed near the bottom end of the screen.

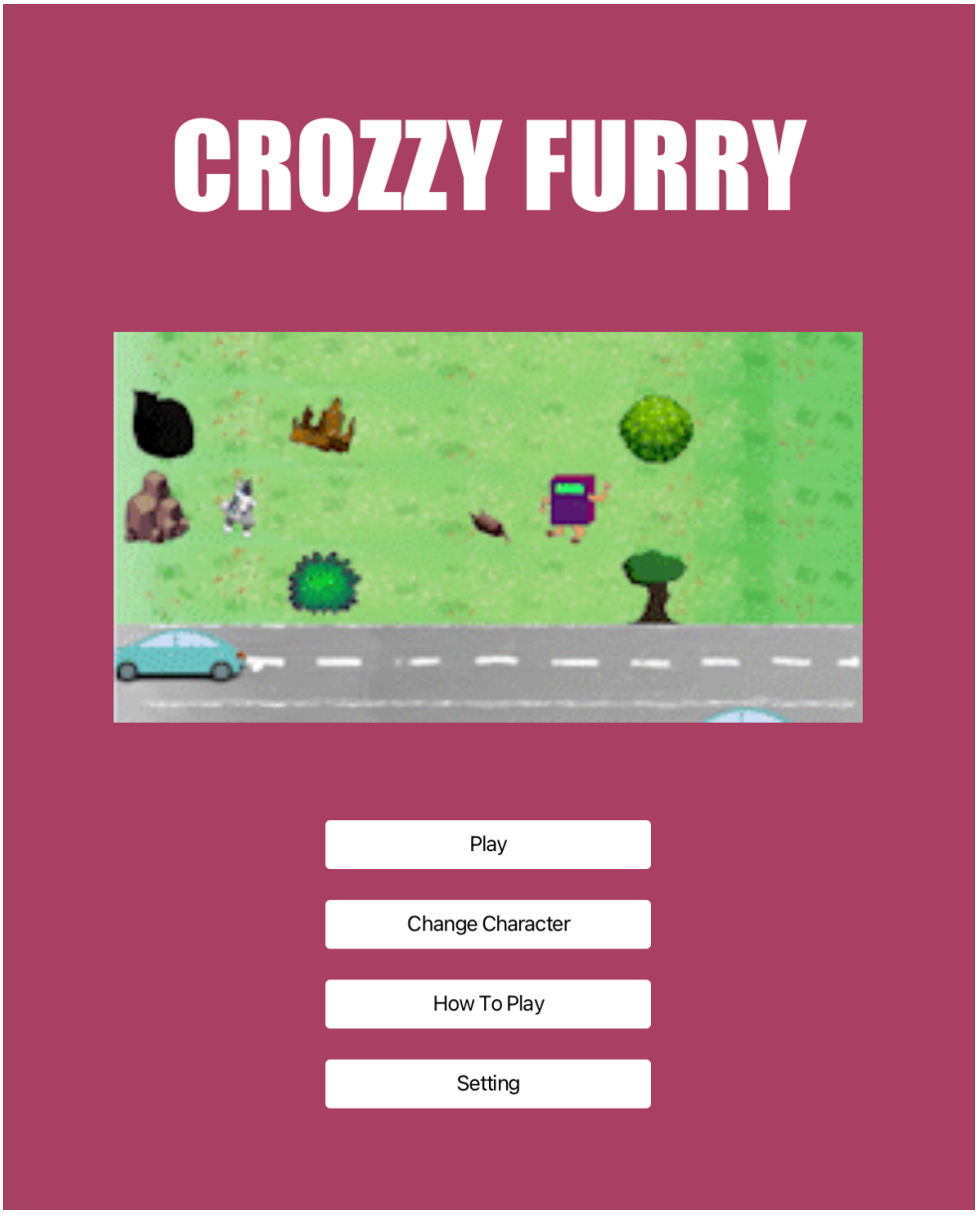
When pressing Start, the game screen continuously moves downward. Players need to move their character upward to avoid it hitting the bottom of the screen or the vehicles. If it does, the game is over.

Players can collect crowns randomly dropped on the map to buy a variety of characters.





Main menu scene



There are four buttons on the main menu.

- Press “Play” to go to the game screen.
- Press “Change Character” to select playable characters from the list.
- Press “How to Play” to see the game instruction.
- Press “Setting” to go to the setting menu.

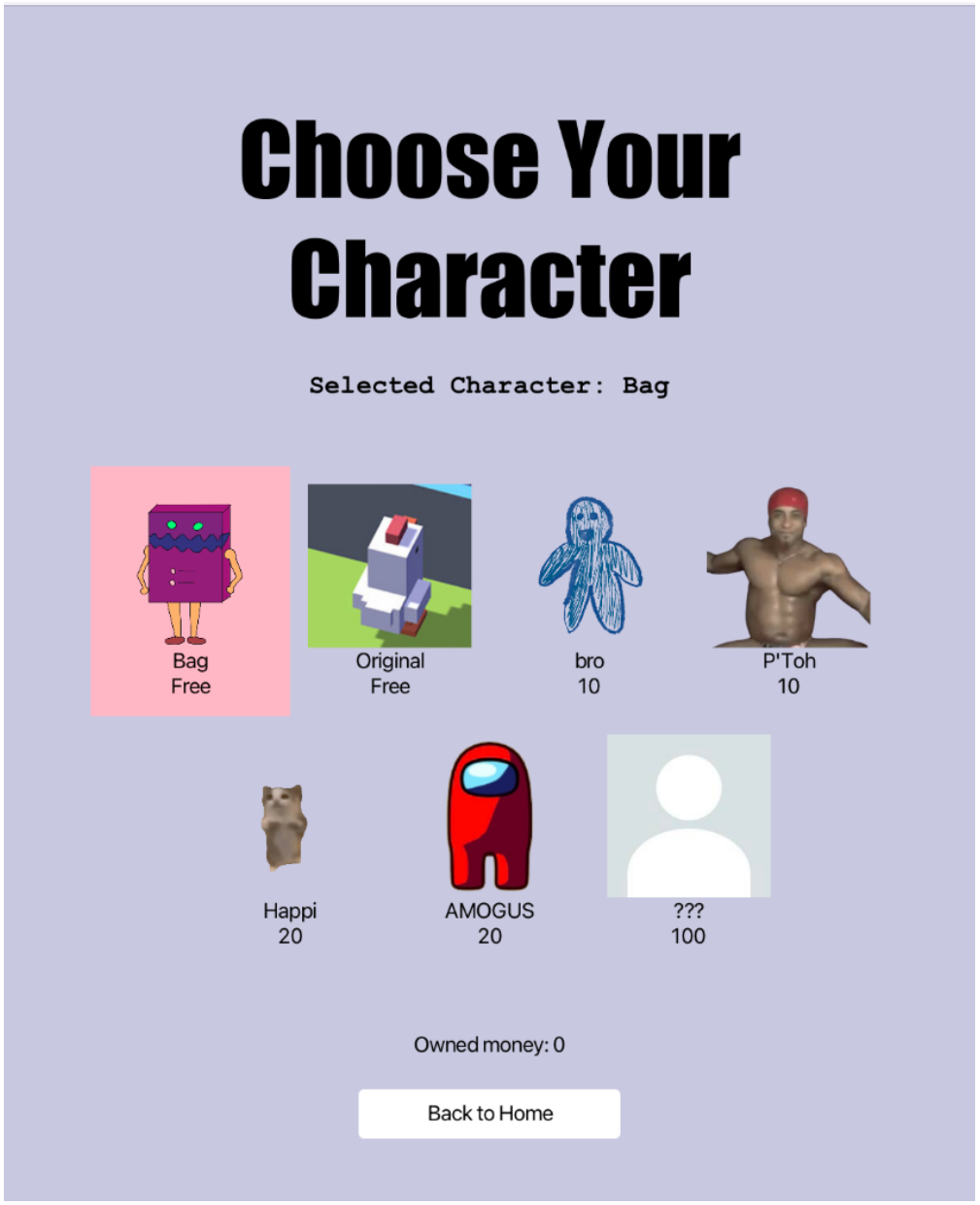
Game screen

There are two buttons on the bottom of the screen.

- Press “Start” to start the game. The game screen will start to go downward, and the character will be controllable.
- Press “Back to Home” to get back to the main menu.

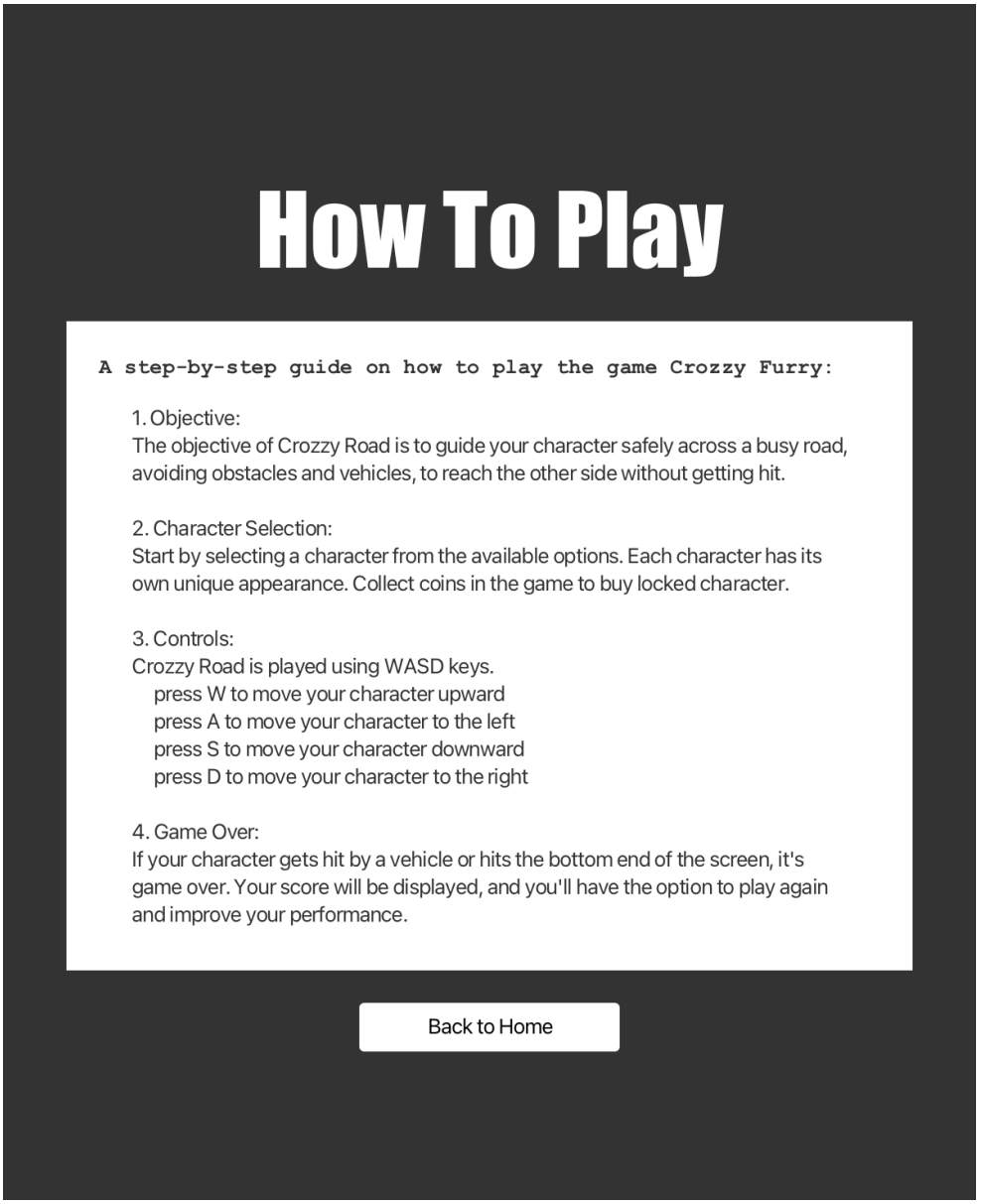
Also, score, high score, and money will be shown on the top of the screen.

Character Selection screen



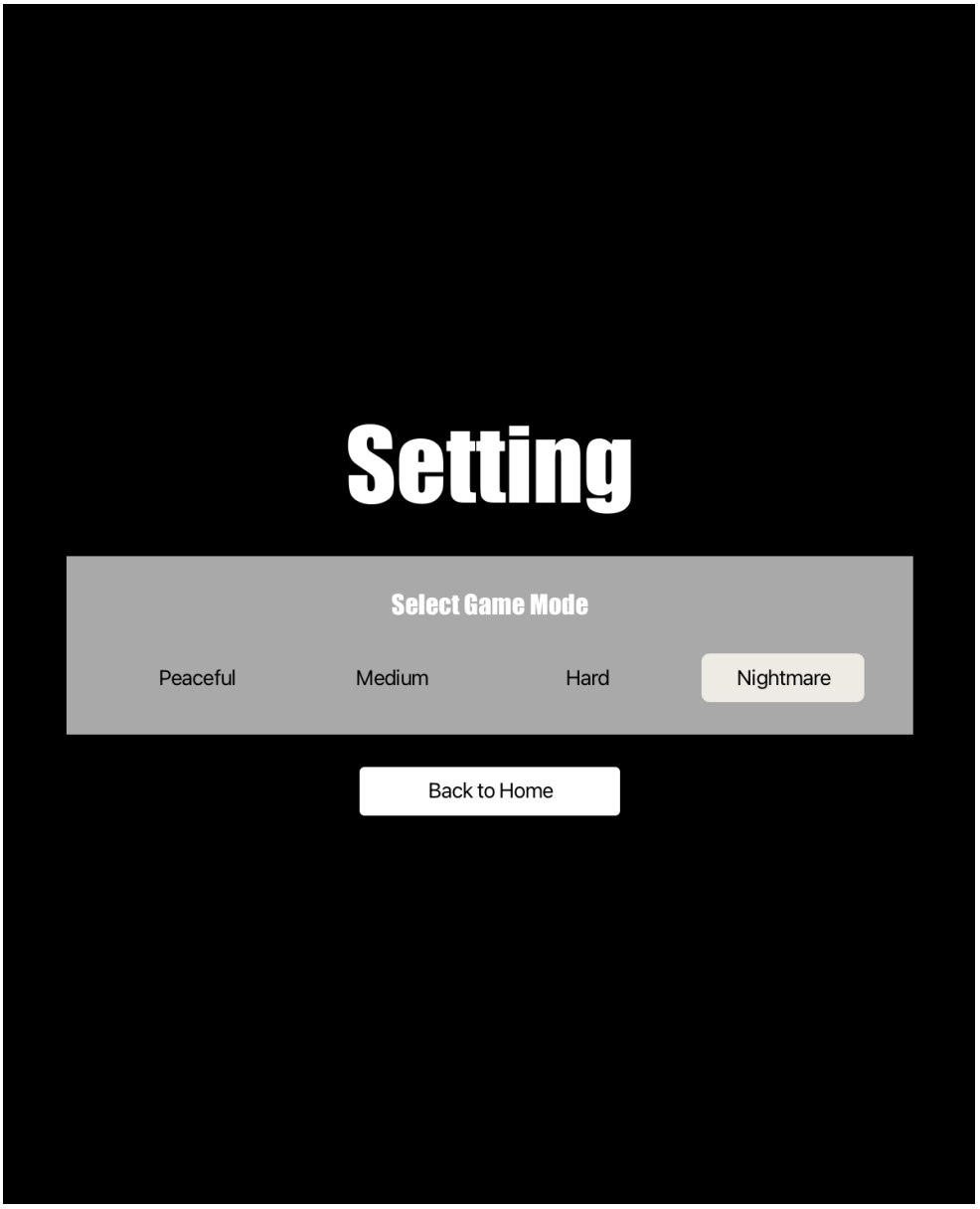
Players can select a character from a list of playable characters from here. Each character icon shows its name and price. The one marked as "Free" can be played without buying. Press "Back to Home" to go back to the main menu.

How to Play screen



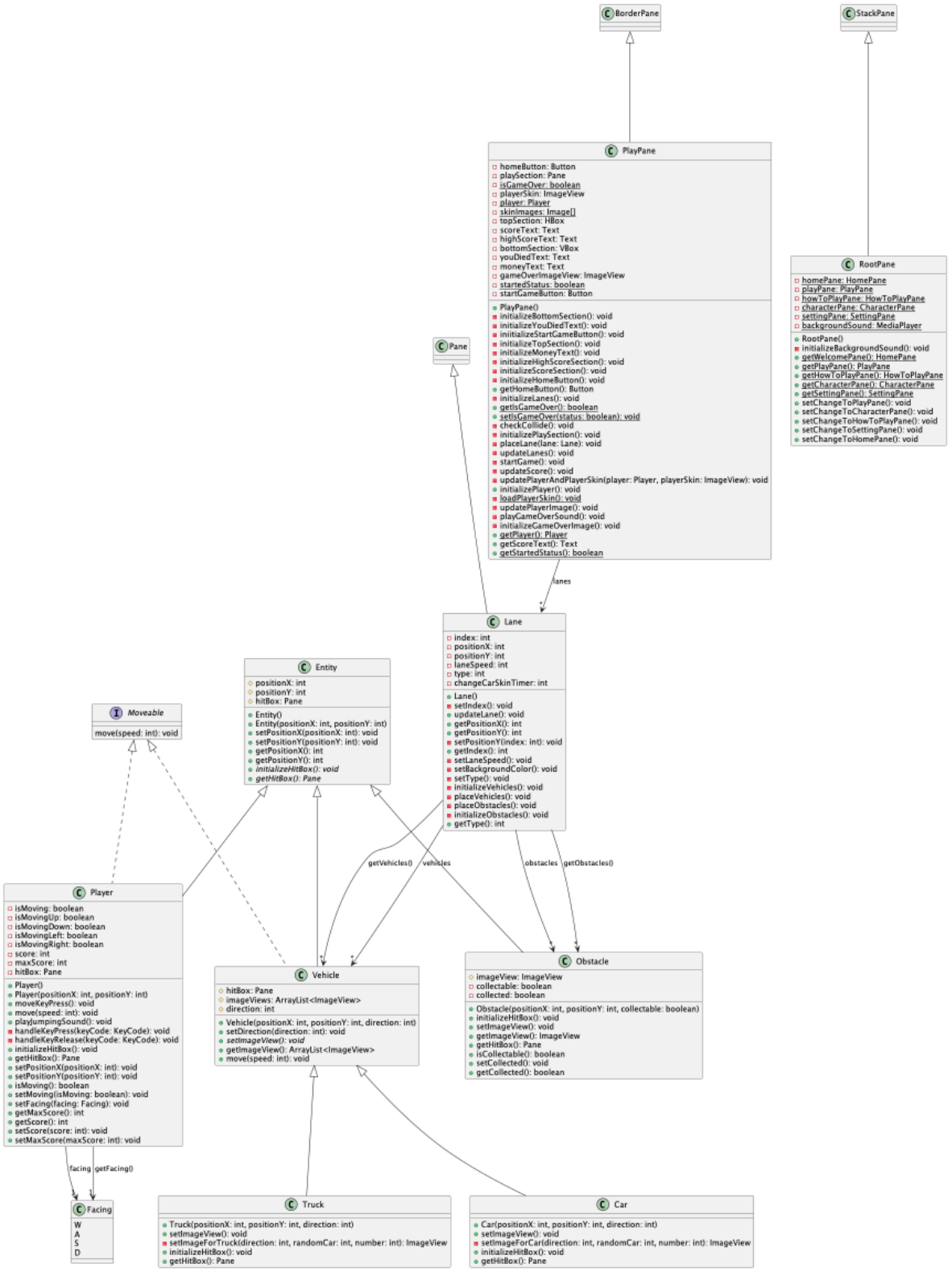
This page shows step-by-step instructions on how to play this game. Press “Back to Home” to get back to the main menu.

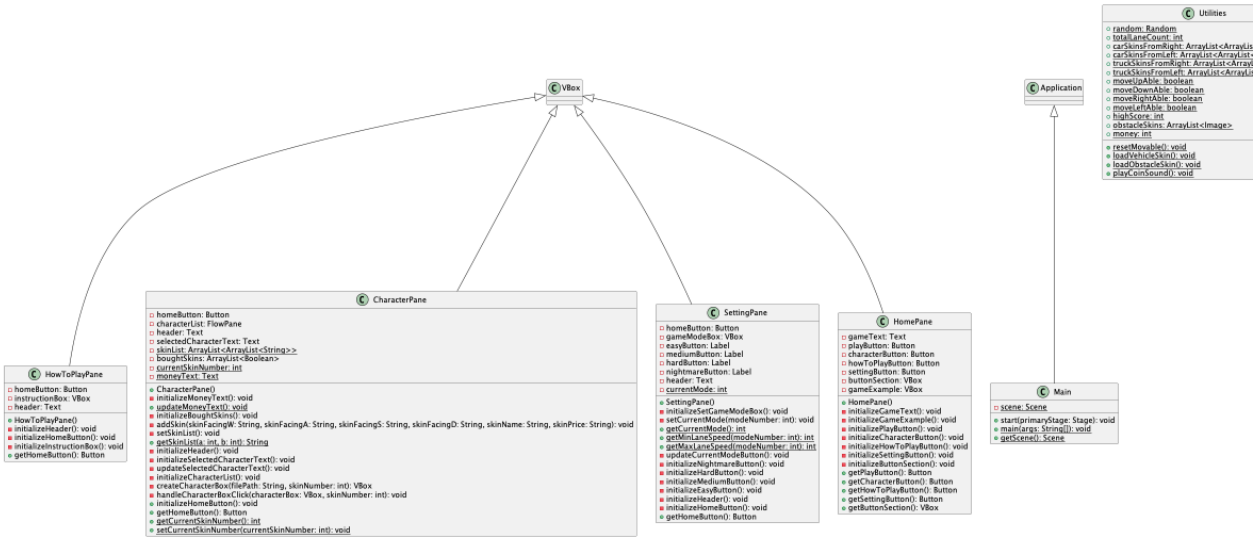
Setting screen



Players can select a game mode from here. Press “Back to Home” to go back to the main menu.

Class diagram




Utilities
- random: Random
- totalLaneCount: int
- carSkinsFromRight: ArrayList<ArrayLis
- carSkinsFromLeft: ArrayList<ArrayList<
- truckSkinsFromRight: ArrayList<Array
- truckSkinsFromLeft: ArrayList<ArrayLi
- moveUpAble: boolean
- moveDownAble: boolean
- moveRightAble: boolean
- moveLeftAble: boolean
- highScore: int
- obstacleSkins: ArrayList<Image>
- money: int
+ resetMovable(): void
+ loadVehicleSkin(): void
+ loadObstacleSkin(): void
+ playCoinSound(): void
VBox
Application
HowToPlayPane
- homeButton: Button
- instructionBox: VBox
- header: Text
+ HowToPlayPane()
+ initializeHeader(): void
+ initializeHomeButton(): void
+ initializeInstructionBox(): void
+ getHomeButton(): Button
CharacterPane
- homeButton: Button
- characterList: FlowPane
- header: Text
- selectedCharacterText: Text
- skinList: ArrayList<ArrayList<String>>
- boughtSkins: ArrayList<Boolean>
- currentSkinNumber: int
- moneyText: Text
+ CharacterPane()
+ initializeMoneyText(): void
+ updateMoneyText(): void
+ initializeBoughtSkins(): void
+ addSkin(skinFacingW: String, skinFacingA: String, skinFacingS: String, skinFacingD: String, skinName: String, skinPrice: String): void
+ setSkinList(): void
+ getSkinList(a: int, b: int): String
+ initializeHeader(): void
+ initializeSelectedCharacterText(): void
+ updateSelectedCharacterText(): void
+ initializeCharacterList(): void
+ createCharacterBox(filePath: String, skinNumber: int): VBox
+ handleCharacterBoxClick(characterBox: VBox, skinNumber: int): void
+ initializeHomeButton(): void
+ getHomeButton(): Button
+ getCurrentSkinNumber(): int
+ setCurrentSkinNumber(currentSkinNumber: int): void
SettingPane
- homeButton: Button
- gameModeBox: VBox
- easyButton: Label
- mediumButton: Label
- hardButton: Label
- nightmareButton: Label
- header: Text
- currentMode: int
+ SettingPane()
+ initializeSetGameModeBox(): void
+ setCurrentMode(modeNumber: int): void
+ getCurrentMode(): int
+ getMinLaneSpeed(modeNumber: int): int
+ getMaxLaneSpeed(modeNumber: int): int
+ updateCurrentModeButton(): void
+ initializeNightmareButton(): void
+ initializeHardButton(): void
+ initializeMediumButton(): void
+ initializeEasyButton(): void
+ initializeHeader(): void
+ initializeHomeButton(): void
+ getHomeButton(): Button
HomePane
- gameText: Text
- playButton: Button
- characterButton: Button
- howToPlayButton: Button
- settingButton: Button
- buttonSection: VBox
- gameExample: VBox
+ HomePane()
+ initializeGameText(): void
+ initializeGameExample(): void
+ initializePlayButton(): void
+ initializeCharacterButton(): void
+ initializeHowToPlayButton(): void
+ initializeSettingButton(): void
+ initializeButtonSection(): void
+ getPlayButton(): Button
+ getCharacterButton(): Button
+ getHowToPlayButton(): Button
+ getSettingButton(): Button
+ getButtonSection(): VBox
Main
- scene: Scene
+ start(primaryStage: Stage): void
+ main(args: String[]): void
+ getScene(): Scene

1. Package gui

1.1 class RootPane extends StackPane

1.1.1 Fields

- static HomePane homePane	homePane for the application
- static PlayPane playPane	playPane for the application
- static HowToPlayPane howToPlayPane	howToPlayPane for the application
- static SettingPane characterPane	settingPane for the application
- static CharacterPane characterPane	characterPane for the application
- static MediaPlayer backgroundSound	background music in the application

1.1.2 Constructor

+ RootPane()	- set the pref width and height to 750 and 600px - initialize all fields and add them to the children - set homePane to the front - initialize all buttons in each pane that can link to other panes - initializeBackgroundSound()

1.1.3 Methods

- void setChangeToPlayPane()	- add event listener to playButton in HomePane to bring playPane to the front
- void setChangeToHomePane()	- add event listener to homeButton in playPane, characterPane, settingPane and howToPlayPane to bring homePane to the front
- void setChangeToCharacterPane()	- add event listener to characterButton in HomePane to bring characterPane to the front
- void setChangeToSettingPane()	- add event listener to settingButton in HomePane to bring settingPane to the

	front
- void setChangeToHowToPlayPane()	- add event listener to howToPlayButton in HomePane to bring howToPlayPane to the front
- void initializeBackgroundSound()	- play Background music from BackgroundSound.mp3
Getter for each fields except backgroundSound	

1.2 class HomePane extends VBox

1.2.1 Fields

- Text gameText	game's name
- Button playButton	button for switching to playPane
- Button howToPlayButton	button for switching to howToPlayPane
- Button SettingButton	button for switching to settingPane
- Button characterButton	button for switching to characterPane
- VBox buttonSection	Box for containing all buttons
- VBox gameExample	Box for containing game example gif

1.2.2 Constructor

+ HomePane()	- set the Padding with Insets(30) - set alignment to center - initialize all fields - set the background color to #A94064 - add gameText and buttonSection to the children

1.2.3 Methods

- void initializeGameText()	- initialize gameText
- void initializeGameExample()	- initialize gameExample with gif file added inside a VBox
- void initializePlayButton()	- initialize playButton

- void initializeCharacterButton()	- initialize characterButton
- void initializeSettingButton()	- initialize settingButton
- void initializeHowToPlayButton()	- initialize howToPlayButton
- void initializeButtonSection()	- initialize ButtonSection with VBox and add every buttons to the children
Getters for each fields except gameText and gameExample	

1.3 class PlayPane extends BorderPane

1.3.1 Fields

- ArrayList<Lane> lanes	lists of each lanes in the playSection
- Button homeButton	button for switching to homePane
- Pane playSection	section for gameplay
- static boolean isGameOver	boolean if game is over yet
- ImageView playerSkin	image for player’s skin
- static Player player	player
- static Image[] skinImages	lists of player’s skin
- HBox topSection	top section
- Text scoreText	text showing score
- Text highScoreText	text showing highscore
- VBox bottomSection	bottom section
- Text youDiedText	text showing you died
- Text moneyText	text showing money
- ImageView gameOverImageView	Image that show when game over
- static boolean startedStatus	boolean for status if started or not
- Button startGameButton	button for starting the game

1.3.2 Constructor

+ PlayPane()	- set pref height and width to 750 and 600 px - initialize all fields - set background color to #EDEBE3 - set top with topSection - set center with playSection - set bottom with bottomSection

1.3.3 Methods

- void initializeBottomSection()	- initialize Bottom section with VBox - initialize startGameButton, youDiedText and homeButton and add them to the children
- void initializeYouDiedText()	- initialize youDiedText and set it to invisible
- void initializeStartGameButton()	- initialize startGameButton and set on action to startGame()
- void initializeHomeButton()	- initialize homeButton
- void initializeTopSection()	- initialize topSection with HBox - initialize scoreSection, moneyText and highScoreSection and add them to the children
- void initializeMoneyText()	- initialize moneyText
- void initializeHighScoreSection()	- initialize highScoreText
- void initializeLanes()	- add new Lane() to lanes 11 times
+ getHomeButton()	- return the homeButton
+ static void setIsGameOver(boolean status)	- set isGameover to status
- void initializePlaySection	- initialize playSection with Pane and place all the lanes in lanes with placeLane(lane) - set clip of this pane with this pane height and width
- void placeLane(Lane lane)	- if lane's position is within the playSection set position of the lane in playSection to lane's position
- void updateLanes()	- if the most top lane has already move

	inside the playSection - create new lane and add it to the lanes - bring playerSkin to the front - update position of each lane - if the lane is already outside the playSection remove it - else move the lane to the current position and move each vehicle's position inside the lane to the current position
- void startGame()	- create new Thread - enable player to move with input keys - set startedStatus to true - while the game is not over - update the lanes - update player and playerSkin - update Score - check if the player collide with anything - sleep the Thread for 20 ms - initialize gameOverImage - play gameOverSound - update money in characterPane - set youDiedText and homeButton to be visible - if the game is over, interrupt the thread
- void updateScore()	- update scoreText with max score from player - if current score is higher than high score in Utilities - set high score in Utilities to current score
- void updatePlayerAndPlayerSkin (Player player, ImageView playerSkin)	- updatePlayerImage() - move player by 1 - set layout position of player's hitbox and playerSkin to the player's position
- void checkCollide()	- if player's position is at the bottom of playSection set isGameOver to true - if player's hitBox intersects with any vehicle's hitBox set isGameOver to true - if there is any obstacle next to the player and it is not collectible set moveable from Utilities in that direction to false

	- if player's hitbox intersect with collectible obstacle
	  - increment money in Utilities by 1
	  - set the obstacle to be collected
	  - play sound from Utilities
	  - set moneyText with the current money

+ void initializePlayer()	- load the playerSkin - initialize playerSkin and player - set playerSkin and player's hitBox layouts to the player's position - add player's hitbox and playerSkin to the children of playSection
- static void loadPlayerSkin()	- get current skin's path from characterPane - set skinImages to the current skin's Image with different side
- void updatePlayerImage()	- check what direction is the player facing and set the playerSkin with corresponding skinImage
- void playGameOverSound()	- play gameOverSound from "GameOverSound.mp3"
- void initializeGameOverImage()	- set Image gameOverImage from "gameOverImage.gif" and add it to the playSection
Getters for gameOver, player, scoreText and startedStatus	

1.4 class CharacterPane extends VBox

1.4.1 Fields

- Button homeButton	button for switching to homePane
- FlowPane characterList	Pane containing every character
- Text header	header Text
- Text selectedCharacterText	Text showing current character
- static ArrayList<ArrayList<String>> skinList	List of skin
- ArrayList<Boolean> boughtSkins	List of boolean showing which skins are bought

- static int currentSkinNumber	index of current skin
- static Text moneyText	Text for showing money

1.4.2 Constructor

+ CharacterPane()	- setSkinList() - set height and width to 750 and 600 px - set spacing, padding and alignment to 20, Insets(20) and center - set background color rgb(200,200,222) - initialize other fields - update money Text - update selected character - add header, selectedCharacterText, characterList, moneyText and homeButton to the children

1.4.3 Methods

- void initializeMoneyText()	- initialize moneyText
- static void updateMoneyText()	- update moneyText with current money from Utilities
- void initializeBoughtSkins()	- initialize boughtSkins if the price is Free then set the boolean for that index to true else set it to false
- void addSkin(String skinFacingW, String skinFacingA, String skinFacingS, String skinFacingD, String skinName, String skinPrice)	- add arrayList containing skinFacingW, skinFacingA, skinFacingS, skinFacingD, skinName and skinPrice to skinList
- void setSkinList()	- initialize skinList - add each skin with addSkin
- static String getSkinList(int skinIndex, int skinFace)	- return skin from the index and facing direction
- void initializeHeader()	- initialize header Text
- void initializeSelectedCharacterText()	- initialize selected character Text
- void updateSelectedCharacterText()	- show the currently selected Text and highlight its box

- void initializeCharacterList()	- initialize characterList with FlowPane and add each characterBox to its children
- VBox createCharacterBox(String filePath, int skinNumber)	- initialize characterBox with VBox and add character's image, name price to its children - set on action to handleCharacterBoxClick()
- void handleCharacterBoxClick(VBox characterBox , int skinNumber)	- if the skin is bought then set current skin and selected character text to the skin - else if has enough money then buy the skin and change the price to "Owned" and set current skin and selected character text to the skin
void initializeHomeButton()	initialize homeButton
Getters for homeButton and currentSkinNumber	
Setter for skinNumber	

1.5 class HowToPlay extends VBox

1.5.1 Fields

- Button homeButton	button for switching to homePane
- VBox instructionBox	box containing instruction
- Text header	header Text

1.5.2 Constructor

+ HowToPlayPane()	- set height and width to 750 and 600 px - set spacing, padding and alignment to 20, Inset(20) and center - set background color to rgb(51,51,51) - initialize other fields and add them to the children

1.5.3 Methods

- void initializeHeader()	- initialize header
- void initializeHomeButton()	-initialzie homeButton
- void initializeInstructionBox()	- initialize instructionBox with VBox containing how to play and rules
Getter for homeButton	

1.6 class Lane extends Pane

1.6.1 Fields

- int index	index of the lane
- int positionX	x position of the lane
- int positionY	y position of the lane
- int landSpeed	lane's vehicle speed
- int type	type of the lane
- ArrayList<Vehicle> vehicles	list containing all vehicles in the lane
- ArrayList<Obstacle> obstacles	list containing all obstacle in the lane
- int changeCarSkinTimer	timer telling when to change vehicle's skin

1.6.2 Constructor

+ Lane()	- set height and width to 50, 450 px and set border to null - initialize all field - place vehicles and obstacle in the children - set background color

1.6.3 Methods

- void setIndex()	- set the index to total laneCount from Utilities

	- increment totalLaneCount by 1
- void setPositionY(this. index)	- if index is equal or less than 10 set position y to 450 - index * 50 - else set it to -99
- void setType()	- if the index is less than 4 set type to 0 - else set it to a random int between 1 and 4
- void setLaneSpeed()	- if the type is between 1 and 3 randomly set speed to [min speed from current mode, max speed from current mode] and randomly choose direction of the lane - else laneSpeed is 0
- void setBackgroundColor()	- if the index is 2 then set the backgroundImage to “specialGrass.png” - else if the type is 1 then set the backgroundImage “grass.png” - else set the backgroundImage to “road.png”
- void initializeVehicles()	- initialize vehicles - add 11 vehicles to the lane by randomly choose between car and truck and randomly choose their distance between each others
- void placeVehicles()	- add all vehicle’s hitbox and imageView to the children at their position
- void initializeObstacle()	- initialize obstacles - if lane type is 4 , randomly add 1 to 4 obstacle(s)
- void placeObstacles()	- add all obstacle’s hitbox and imageView to the children at their position
+ void updateLane()	- increment positionY and changeCarSkinTimer by 1 - if any vehicles is out of the lane remove its hitbox and imageView - else set its layouts to its position and if changeCarSkinTimer is 5 swap visibility of imageViews and set changeCarskinTimer to 0 -if any collectible obstacle is collected

	remove its hitBox and imageView
Getters for all fields	

1.7 class SettingPane extends VBox

1.7.1 Fields

- Button homeButton	button for switching to homePane
- VBox gameModeBox	box containing game mode section
- Label easyButton	easy mode button
- Label mediumButton	medium mode button
- Label hardButton	hard mode button
- Label nightmareButton	nightmare mode button
- static int currentMode	integer for setting current mode (set default to 3) 0 means easy mode 1 means medium mode 2 means hard mode 3 means nightmare mode

1.7.2 Constructor

+ SettingPane()	- set pane pref height to 750 - set pane pref width to 600 - set spacing between items to 20 - set pane padding to 40 - set item alignment to center - set background color to black - initialize header - initialize home button - initialize game mode setting box - add header, game mode setting box, and home button to the pane - update current mode button

1.7.3 Methods

- void initializeSetGameModeBox()	- initialize gameModeBox - set selectGameModeText to “Select Game Mode” - set selectGameModeText font to Impact with font size 16 (bold) - set selectGameModeText color to white - initialize setGameModeBox as HBox - initialize easyButton - initialize mediumButton - initialize hardButton - initialize nightmareButton - add easyButton, mediumButton, hardButton, and nightmareButton to setGameModeBox - set spacing between items in setGameModeBox to 20 - set setGameModeBox item alignment to center - add selectGameModeText and setGameModeBox to gameModeBox - set spacing between items in gameModeBox to 20 - set gameModeBox item alignment to center - set gameModeBox background color to dark gray - set padding of gameModeBox to 20
- void setCurrentMode(int modeNumber)	- set currentMode
+ static int getCurrentMode()	- get currentMode
+ static int getMinLaneSpeed(int modeNumber)	- if modeNumber is 0, min lane speed is 2 - else if modeNumber is 1, min lane speed is 3 - else if modeNumber is 2, min lane speed is 4 - else if modeNumber is 3, min lane speed is 6
+ static int getMaxLaneSpeed(int modeNumber)	- if modeNumber is 0, max lane speed is 4 - else if modeNumber is 1, max lane speed is 6 - else if modeNumber is 2, max lane speed is 10 - else if modeNumber is 3, max lane

	speed is 14
- void updateCurrentModeButton()	- change background color of the button of the current mode to #EDEBE3 and set the others' background to null
- void initializeEasyButton()	- initialize easyButton as "Peaceful" - set the button pref width to 100 - set the button pref height to 30 - set alignment to center - set background to null - on mouse clicked, set current mode to 0 and call updateCurrentModeButton
- void initializeMediumButton()	- initialize easyButton as "Medium" - set the button pref width to 100 - set the button pref height to 30 - set alignment to center - set background to null - on mouse clicked, set current mode to 1 and call updateCurrentModeButton
- void initializeHardButton()	- initialize easyButton as "Hard" - set the button pref width to 100 - set the button pref height to 30 - set alignment to center - set background to null - on mouse clicked, set current mode to 2 and call updateCurrentModeButton
- void initializeNightmareButton()	- initialize easyButton as "Nightmare" - set the button pref width to 100 - set the button pref height to 30 - set alignment to center - set background to null - on mouse clicked, set current mode to 3 and call updateCurrentModeButton
- void initializeHeader()	- initialize header as "Setting" - set text color to white - set font to Impact with font size 60 (bold)
- void initializeHomeButton()	- initialize back to home button
+ getHomeButton()	- get homeButton

2. Package entity.base

2.1 abstract class Entity

2.1.1 Fields

# int positionX	position in the x-axis of the entity
# int positionY	position in the y-axis of the entity
# Pane hitBox	hit box pane of the entity

2.1.2 Constructor

+ Entity()	- set the entity's positionX as 200 and positionY as 300
+ Entity(int positionX, int positionY)	- set the entity's positionX and positionY

2.1.3 Methods

+ abstract void initializeHitBox()	- initialize hit box for the entity
+ abstract Pane getHitBox()	- get hit box of the entity
Getters and Setters for positionX and positionY	

2.2 interface Movable

2.1.1 Methods

+ abstract void move()	- change positionX and positionY of the entity

2.3 enum Facing

W, A, S, D

3 Package entity.derived

3.1 abstract class Vehicle extends Entity implements Moveable

3.1.1 Fields

# Pane hitBox	hit box pane of the entity
# ArrayList<ImageView> imageViews	ArrayList of ImageView of the vehicle
# int direction	direction of the vehicle (0 for going from left and 1 for going from right to left)

3.1.2 Constructor

+ Vehicle(int positionX, int positionY, int direction)	- set the vehicle's positionX, positionY and direction

3.1.3 Methods

+ void setDirection(int direction)	- set the vehicle's direction
+ abstract void setImageView()	- set the vehicle's ImageView
+ ArrayList<ImageView> getImageView()	- get the vehicle's ArrayList of ImageView
+ void move(int speed)	- move the positionX of the vehicle by its speed

3.2 class Car extends Vehicle

3.2.1 Constructor

+ Car(int positionX, int positionY, int direction)	- set the car's positionX, positionY and direction

3.2.2 Methods

+ void setImageView()	- initialize the car's ImageView - set the car's ImageView by randomize from the car skin list
- ImageView setImageForCar(int direction, int randomCar, int number)	- set the car's ImageView according to its direction - set the ImageView's fit height to 50 - set the ImageView's fit width to 100
+ void initializeHitBox()	- initialize the car's hit box - set the hit box's pref height to 50 - set the hit box's pref width to 100

+ Pane getHitBox()	- get the car’s hit box

3.3 class Truck extends Vehicle

3.3.1 Constructor

+ Truck(int positionX, int positionY, int direction)	- set the car’s positionX, positionY and direction

3.3.2 Methods

+ void setImageView()	- initialize the truck’s ImageView - set the truck’s ImageView by randomize from the truck skin list
- ImageView setImageForTruck(int direction, int randomCar, int number)	- set the truck’s ImageView according to its direction - set the ImageView’s fit height to 50 - set the ImageView’s fit width to 150
+ void initializeHitBox()	- initialize the truck’s hit box - set the hit box’s pref height to 50 - set the hit box’s pref width to 150
+ Pane getHitBox()	- get the truck’s hit box

3.4 class Obstacle extends Entity

3.4.1 Fields

# ImageView imageView	ImageView of the obstacle
- boolean collectable	the status that indicates if the obstacle is collectable
- boolean collected	the status that indicates if the obstacle is collected

3.4.2 Constructor

+ Obstacle(int positionX, int positionY, boolean collectable)	- set the obstacle position - set the obstacle’s collectable status - initialize the obstacle’s hit box - set the obstacle’s ImageView - set the obstacle’s collected status to false

3.4.3 Methods

+ void initializeHitBox()	- initialize the obstacle's hit box - set the hit box's pref height to 50 - set the hit box's pref width to 50
+ void setImageView()	- initialize the obstacle's ImageView - set the obstacle's ImageView by randomize from the obstacle skin list - set the ImageView's fit height to 50 - set the ImageView's fit width to 50
+ ImageView getImageView()	- get the obstacle's ImageView
+ Pane getHitBox()	- get the obstacle's hit box
+ boolean isCollectable()	- get the obstacle's collectable status
+ void setCollected()	- set the obstacle's collected status
+ boolean getCollected()	- get the obstacle's collected status

3.5 class Player extends Entity implements Moveable

3.5.1 Fields

- Facing facing	facing direction if the player
- boolean isMovingUp	the status indicating if the player is moving up
- boolean isMovingDown	the status indicating if the player is moving down
- boolean isMovingLeft	the status indicating if the player is moving left
- boolean isMovingRight	the status indicating if the player is moving right
- int score	the player's current score (the number of lanes the player crossed)
- int maxScore	the player's max score (the maximum number of lanes the player crossed)
- Pane hitBox	the player's hit box

3.5.2 Constructor

+ Player()	- set the player's position - set the player's facing to W - set all the player's moving status to false - initialize the player's hit box - set score to 0 - set max score to 0

3.5.3 Methods

+ void moveKeyPress()	- if key is pressed, call HandleKeyPress - if key is released, call HandleKeyrelease
+ void move()	- set the player's positionY by its speed
+ void playJumpingSound()	- play the jumping sound when the player moved
- void handleKeyPress(KeyCode keyCode)	- check if W,A,S, and D key is pressed - if so, move the player in that direction by 50 - also check if the player is in the current play pane, the key is not currently pressed, the game is not over yet, and the player is movable in that direction
- void handleKeyRelease(KeyCode keyCode)	- check if W,A,S, and D key is released
+ void initializeHitBox()	- initialize the character's hit box - set the hit box's pref height to 2 - set the hit box's pref width to 30
+ Pane getHitBox()	- get the player's hit box
+ void setPositionX(int positionX)	- set the player's positionX - if the value is not in between 0 and 450, set to 200
+ void setPositionY(int positionY)	- set the player's positionY - if the value is not in between 0 and 500, set to 500
Setters and Getters for Facing, score, and maxScore	

4. Package utils

4.1 class Utilities

4.1.1 Fields

+ static Random random	Random (set to new Random())
+ static int totalLaneCount	total lanes count (set to 0)
+ static ArrayList<ArrayList<Image>> carSkinsFromRight	ArrayList or ArrayList of Image storing skins for cars that move from the right
+ static ArrayList<ArrayList<Image>> carSkinsFromLeft	ArrayList or ArrayList of Image storing skins for cars that move from the left
+ static ArrayList<ArrayList<Image>> truckSkinsFromRight	ArrayList or ArrayList of Image storing skins for trucks that move from the right
+ static ArrayList<ArrayList<Image>> truckSkinsFromLeft	ArrayList or ArrayList of Image storing skins for trucks that move from the left
+ public static boolean moveUpAble	the status indicating if moving up is available (set to true)
+ public static boolean moveDownAble	the status indicating if moving down is available (set to true)
+ public static boolean moveRightAble	the status indicating if moving right is available (set to true)
+ public static boolean moveLeftAble	the status indicating if moving left is available (set to true)
+ static ArrayList<Image> obstacleSkins	ArrayList storing images of obstacle's skin
+ static int money	amount of money (set to 0)

4.1.2 Methods

+ static void resetMovable()	- set moveUpAble, moveLeftAble, moveRightAble, and moveDownAble to true
+ static void loadVehicleSkin()	- load all vehicle's skin
+ static void loadObstacleSkin()	- load all obstacle's skin

+ static void playCoinSound()	- play coin the sound when a coin is collected

5. Package application

5.1 class Main extends Application

5.1.1 Fields

- static Scene scene	scene for the application

5.1.2 Methods

+ static void main(String[] args)	- launch args
+ void start (Stage primaryStage) throws Exception	- load vehicle skin - load obstacle skin - initialize root pane - initialize scene - set scene title to “Crozzy Furry” - set resizable to false - show scene
+ public static Scene getScene()	- get scene