ANIMAL FALL DOWN

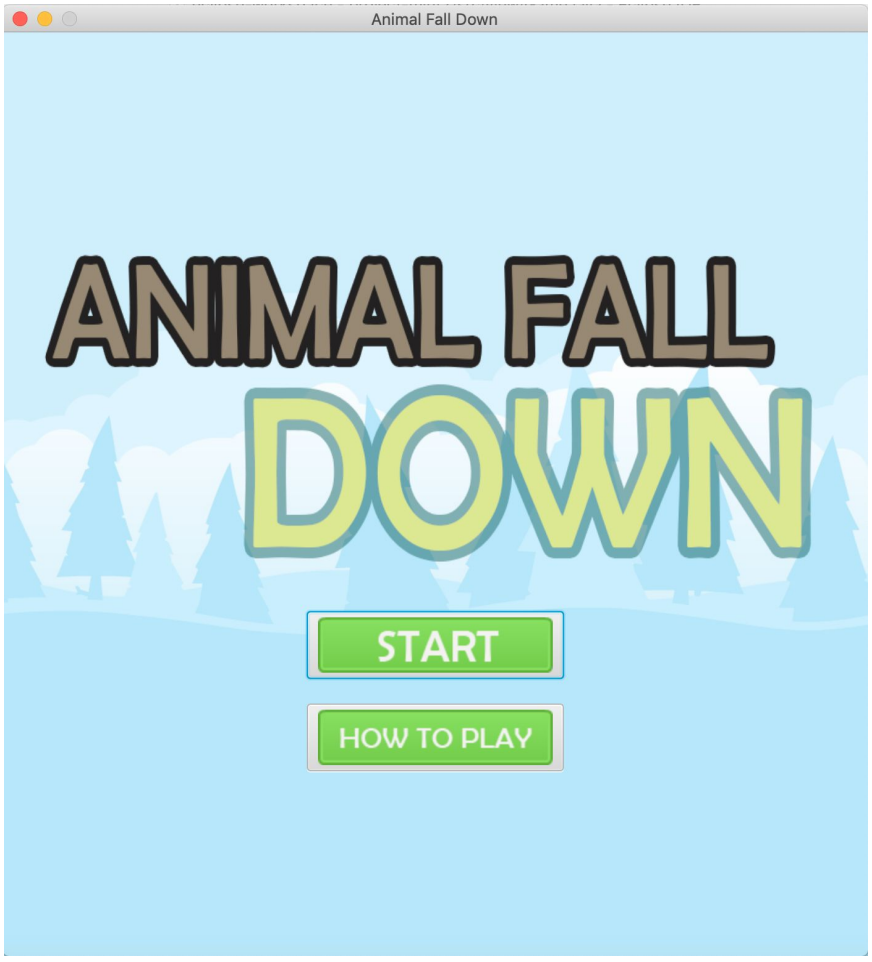
Submitted by
Chaninart Kijrungrattanakul 6131007321
Palita Tearwattanarattikal 6131025621

ABOUT THE GAME

"Animal Fall Down" is created by Java FX. The player controls an animal to drop safely between the platforms to fall down deeper and attempting to collect coins to increase score. Try to control the animal to not collide with any bars. The aim of this game is to get the highest score.

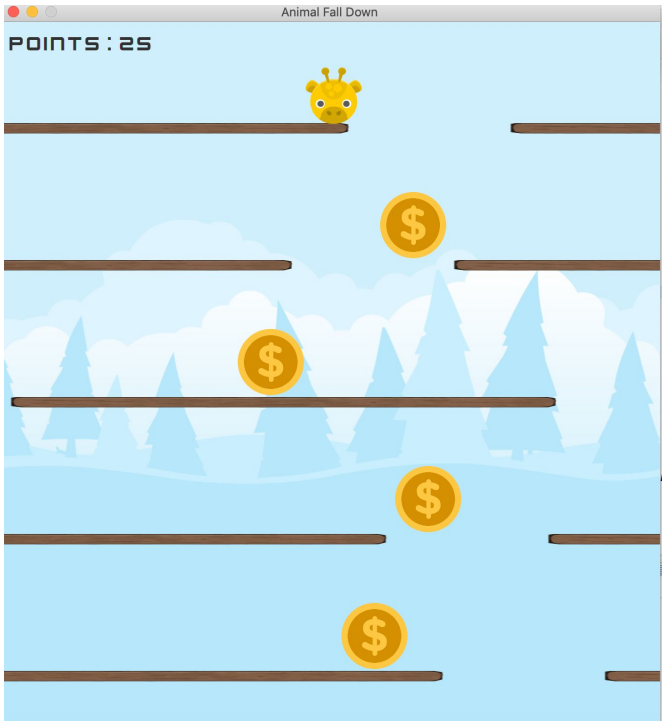
GAME SCREENS

Main menu:



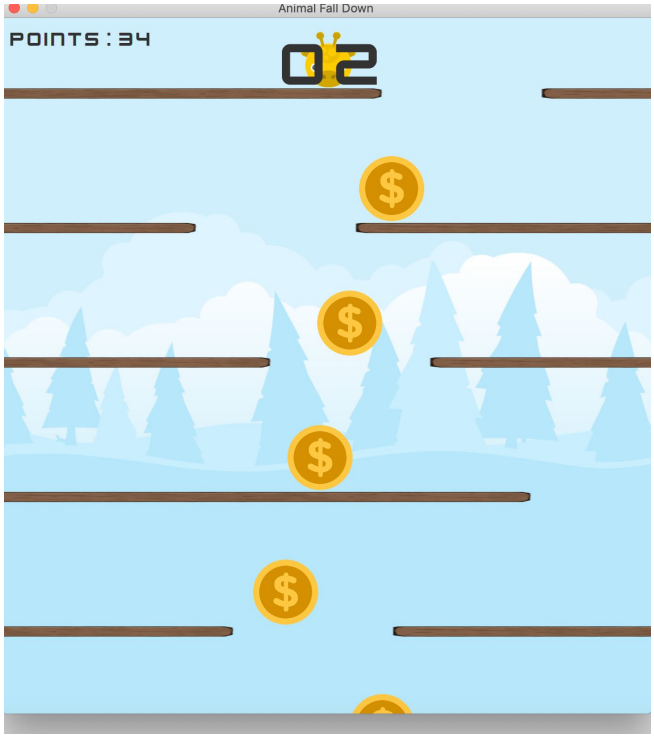
Choose character screen:

Gameplay screen:

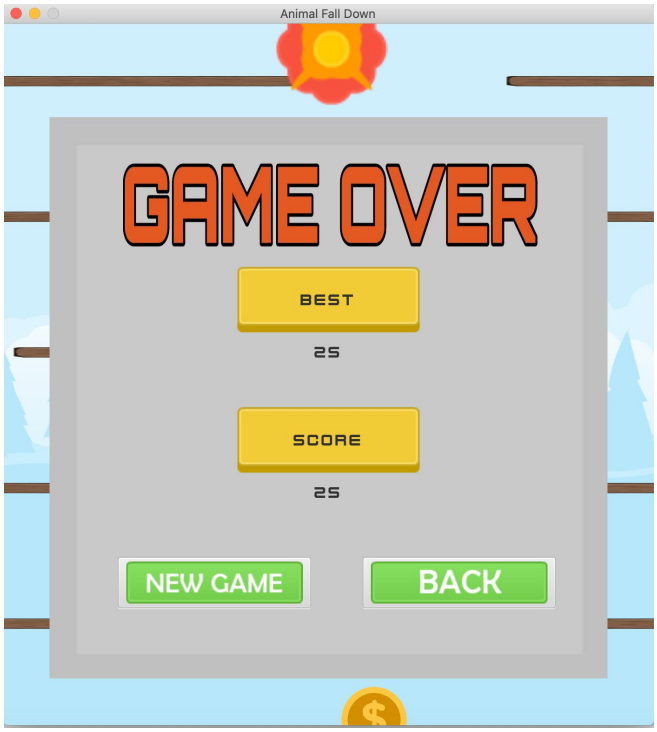


Animal Fall Down

While item activated:

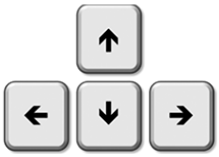


Game over screen:



Animal Fall Down

CONTROLS



press the buttons to move left and right

GAME MODELS

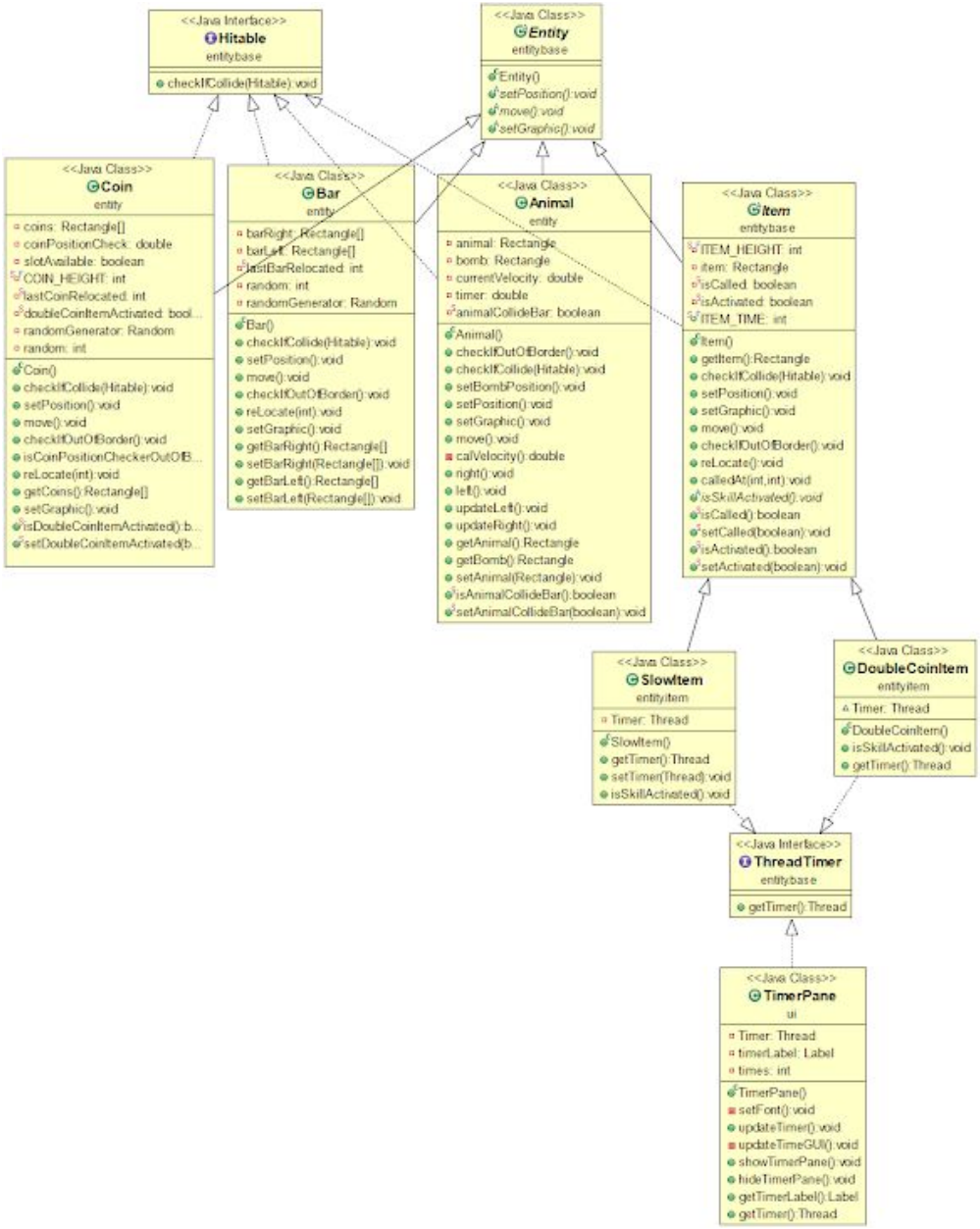
Player:

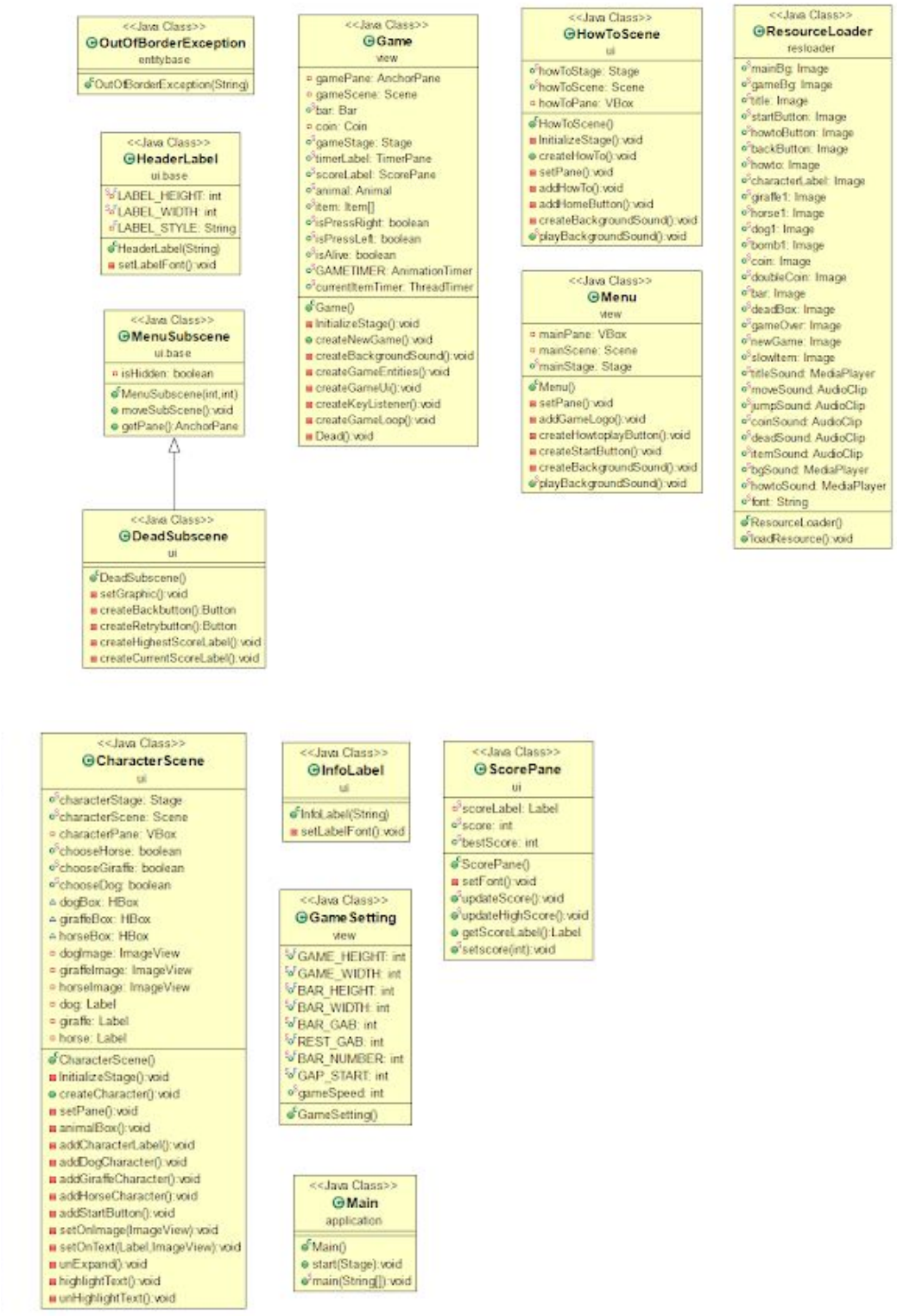
Items:

Obstructions:

Implementation Details

The UML diagram as shown:




<<Java Class>>
OutOfBorderException
entitybase
OutOfBorderException(String)
<<Java Class>>
Game
view
gamePane: AnchorPane
gameScene: Scene
bar: Bar
coin: Coin
gameStage: Stage
timerLabel: TimerPane
scoreLabel: ScorePane
animal: Animal
item: Item[]
isPressRight: boolean
isPressLeft: boolean
isAlive: boolean
GAMETIMER: AnimationTimer
currentItemTimer: ThreadTimer
Game()
InitializeStage():void
createNewGame():void
createBackgroundSound():void
createGameEntities():void
createGameUi():void
createKeyListener():void
createGameLoop():void
Dead():void
<<Java Class>>
HowToScene
ui
howToStage: Stage
howToScene: Scene
howToPane: VBox
HowToScene()
InitializeStage():void
createHowTo():void
setPane():void
addHowTo():void
addHomeButton():void
createBackgroundSound():void
playBackgroundSound():void
<<Java Class>>
ResourceLoader
resloader
mainBg: Image
gameBg: Image
title: Image
startButton: Image
howtoButton: Image
backButton: Image
howto: Image
characterLabel: Image
giraffe1: Image
horse1: Image
dog1: Image
bomb1: Image
coin: Image
doubleCoin: Image
bar: Image
deadBox: Image
gameOver: Image
newGame: Image
slowItem: Image
titleSound: MediaPlayer
moveSound: AudioClip
jumpSound: AudioClip
coinSound: AudioClip
deadSound: AudioClip
itemSound: AudioClip
bgSound: MediaPlayer
howtoSound: MediaPlayer
font: String
ResourceLoader()
loadResource():void
<<Java Class>>
HeaderLabel
ui.base
LABEL_HEIGHT: int
LABEL_WIDTH: int
LABEL_STYLE: String
HeaderLabel(String)
setLabelFont():void
<<Java Class>>
Menu
view
mainPane: VBox
mainScene: Scene
mainStage: Stage
Menu()
setPane():void
addGameLogo():void
createHowtoplayButton():void
createStartButton():void
createBackgroundSound():void
playBackgroundSound():void
<<Java Class>>
MenuSubscene
ui.base
isHidden: boolean
MenuSubscene(int,int)
moveSubScene():void
getPane():AnchorPane
<<Java Class>>
DeadSubscene
ui
DeadSubscene()
setGraphic():void
createBackbutton():Button
createRetrybutton():Button
createHighestScoreLabel():void
createCurrentScoreLabel():void
<<Java Class>>
CharacterScene
ui
characterStage: Stage
characterScene: Scene
characterPane: VBox
chooseHorse: boolean
chooseGiraffe: boolean
chooseDog: boolean
dogBox: HBox
giraffeBox: HBox
horseBox: HBox
dogImage: ImageView
giraffeImage: ImageView
horseImage: ImageView
dog: Label
giraffe: Label
horse: Label
CharacterScene()
InitializeStage():void
createCharacter():void
setPane():void
animalBox():void
addCharacterLabel():void
addDogCharacter():void
addGiraffeCharacter():void
addHorseCharacter():void
addStartButton():void
setOnImage(ImageView):void
setOnText(Label,ImageView):void
unExpand():void
highlightText():void
unHighlightText():void
<<Java Class>>
InfoLabel
ui
InfoLabel(String)
setLabelFont():void
<<Java Class>>
ScorePane
ui
scoreLabel: Label
score: int
bestScore: int
ScorePane()
setFont():void
updateScore():void
updateHighScore():void
getScoreLabel():Label
setscore(int):void
<<Java Class>>
GameSetting
view
GAME_HEIGHT: int
GAME_WIDTH: int
BAR_HEIGHT: int
BAR_WIDTH: int
BAR_GAB: int
REST_GAB: int
BAR_NUMBER: int
GAP_START: int
gameSpeed: int
GameSetting()
<<Java Class>>
Main
application
Main()
start(Stage):void
main(String[]):void

1.Package application

1.1 Class Main extends Application

1.11 Method

+ void start(Stage primaryStage)	Set primaryStage to mainStage Set Resizable false Show stage
+ Void main(String[] args)	An entry point of the application

2.Package entity.base

2.1 Class Entity

2.1.1 Constructor

+ Entity()	Set graphic context and its position

2.1.2 Method

+ setPosition()	Set first position
+ setGraphic()	Set Graphic Context
+ move()	Move the object in each Animation Timer loop

2.2 Class Item extends Entity implements Hitable

2.2.1 Field

- int final ITEM_HEIGHT = 70	Item height
+ int final ITEM_TIME = 5000	Item effect duration (in millis)
- Rectangle Item	Collect item's hitbox and its position
- static boolean isCalled	Check if item is called to the game field or not
- static boolean isActivated	Check if item is activated or not

2.2.2 Constructor

+ Item()	Set both isCalled() and isActivated() to false

2.2.3 Method

+ Rectangle getItem()	Return Item
+ void checkIfCollide(Hitable animall)	Check if item collide with animal, If yes relocate and then set isActivated to true.
+ void setPosition()	Set item's position to be outside of the game field
+ void setGraphic()	Set Graphic Context for item
+ void move()	If isCalled is true, item move by -gameSpeed in Y position and check if collide with animal or not
+ void checkIfOutOfBorder()	Check if item is out of the game field, if yes then relocate
+ void reLocate()	Set item to its position
+ void calledAt(int posx, int posy)	Set item's x position = posx Set item's y position = posy
+ boolean isActivated()	Getter for isActivated
+ void isSkillActivated()	Check if isActivated is true or not If true then apply the item's effect to game
+ boolean isCalled()	Getter for isCalled
+ void setCalled(boolean isCalled)	Setter for isCalled
+ boolean isActivated()	Getter for isActivated
+ void setActivated(boolean isActivated)	Setter for isActivated

2.3 Interface Hitable

2.3.1 Method

+ void checkIfCollide(Hitable x)	Check if collide with x object or not

2.4 Interface ThreadTimer

2.4.1 Method

+ Thread getTimer()	Getter for Timer Thread

2.5 class OutOfBorderException extends Exception

2.5.1 Constructor

+ OutOfBorderException(String e)	Initialize string to Exception

3. Package entity

3.1 Class Animal

3.1.1 Field

- Rectangle animal	Collect rectangle for animal
- Rectangle bomb	Collect rectangle for bomb
- double currentVelocity	Collect current velocity
- double timer	Collect accelerate time
- boolean animalCollideBar	Check if animal collide with x or not

3.1.2 Constructor

+ Animal()	Set animalCollideBar to false

3.1.3 Method

+ void checkIfOutOfBorder()	Check if an animal is out of border If animal out of right or left border then move it to another side
+ void checkIfCollide(Hitable barz)	Check if barz collide with animal if yes then set animalCollideBar is true

Animal Fall Down

+ void setBombPosition()	Initialize bomb animation and set its position
+ void setPosition()	Set animal position
+ void setGraphic()	Set animal and bomb graphic
+ void move()	If animal does not collide with bar, move animal down in +y direction else move animal by -gameSpeed Move left or move right
+ void setSize()	Set animal and bomb size
+ Rectangle getAnimal()	Getter for animal
+ Rectangle getBomb()	Getter for bomb
+ void setAnimal(Rectangle animal)	Setter for animal
+ boolean isAnimalCollideBar	Getter for animalCollideBar
+ void setAnimalCollideBar(boolean animalCollideBar)	Setter for animalCollideBar
- double calVelocity()	Return current velocity + 1.1*timer
+ void right()	If animal still alive then move it right
+ void left()	If animal still alive then move it left
+ void updateLeft()	Update position when move left
+ void updateRight()	Update position when move right

3.2 Class Bar

3.2.1 Field

- Rectangle[] barRight	Collect all of the rectangle for bars right
- Rectangle[] barLeft	Collect all of the rectangle for bars left
- int lastBarRelocated	Collect index of barRight or barLeft that last relocated
- int random	Collect any random int value

Animal Fall Down

- Random randomGenerator	Random generator for this class

3.2.2 Constructor

+ Bar()	Set lastBarRelocated to bar number-1

3.2.3 Method

+ void checkIfCollide(Hitable x)	Check if bar collide to x
+ void setPosition()	Set bars to its position
+ void move()	Move bars in each animation loop and check if it out of border and check if collide with animal
+ void checkIfOutOfBorder()	Check if any bar is out of border then relocate
+ void reLocate(int i)	Relocate bar that index i
+ Rectangle[] getBarRight()	Getter for barRight
+ Rectangle[] getBarLeft()	Getter for barLeft
+ void setBarRight()	Setter for barRight
+ void setBarLeft()	Setter for barLeft
+ void setGraphic()	Set graphic for bars

3.3 Class Coin

3.3.1 Field

- Rectangle[] coins	Collect all rectangles for coins
- double coinPositionCheck	Collect the position of the star for item to spawn
- boolean slotAvailable	Collect boolean for is slot for item to spawn is available
- int final COIN_HEIGHT = 70	Collect the coin height
- int lastCoinRelocated	Set the index of last relocated coin
- boolean doubleCoinItemActivated	Collect if doubleCoinItem is activated

Animal Fall Down

- Random randomGenerator	Random generator for this class
- int random	Collect the random value

3.3.2 Constructor

+ Coin()	Set lastCoinRelocated to bar number -1

3.3.3 Method

+ void checkIfCollide(Hitable animal)	Check if coin collide with animal if yes then add one score and update scorepane
+ void setPosition()	Set coin's position
+ void move()	Move coins and check if star out of border Call isCoinPositionCheckerOutOfBorder()
+ void checkIfOutOfBorder()	Check if coin out of border, if yes then relocate
+ void isCoinPositionCheckerOutOfBorder()	Check if coinPositionCheck is go out of border if yes slotAvailable is true
+ void reLocate(int i)	Relocate coin to its first position and if slotAvailable is true, call item at coin position
+ Rectangle[] getCoins()	Getter for coins
+ void setGraphic()	Set graphic for coins
+ boolean isDoubleCoinItemActivated()	Check if doubleCoinItem is activated or not
+ void setDoubleCoinItemActivated(boolean doubleCoinItemActivated)	Set doubleCoinItemActivated value

4. Package entity.item

4.1 Class DoubleCoinItem

4.1.1 Field

Thread Timer	Timer for item

Animal Fall Down

4.1.2 Constructor

+ DoubleCoinItem()	Set its graphic context

4.1.3 Method

+ void isSkillActivated()	If isActivated is true Set doubleCoinItemActivated to true Start threadTimer and reset to default
+ Thread getTimer()	Getter for timer thread

4.2 Class SlowItem

4.2.1 Field

Thread Timer	Timer for item

4.2.2 Constructor

+ SlowItem()	Set its graphic context

4.1.3 Method

+ void isSkillActivated()	If isActivated is true Set game speed to 2 Start threadTimer and reset to default
+ Thread getTimer()	Getter for timer thread

5. Package resloader

5.1 Class ResourceLoader

5.1.1 Field

+ Image mainBg	Background menu image
+ Image gameBg	Background game image
+ Image title	Title image
+ Image startButton	Start button image
+ Image howtoButton	How to button image

+ Image backButton	Back button image
+ Image howto	How to label
+ Image characterLabel	Character label
+ Image giraffe1	Giraffe image
+ Image horse1	Horse image
+ Image dog1	Dog image
+ Image bomb1	Bomb image
+ Image coin	Coin image
+ Image doubleCoin	Double coin item image
+ Image bar	Bar image
+ Image deadBox	Dead box image
+ Image gameOver	Game over label
+ Image newGame	New game button image
+ Image slowItem	Slow item image
+ MediaPlayer titleSound	Title sound
+ MediaPlayer bgSound	Background game sound
+ MediaPlayer howtoSound	Background how to scene sound
+ AudioClip moveSound	Move sound
+ AudioClip jumpSound	Jump sound
+ AudioClip coinSound	Coin sound
+ AudioClip itemSound	Item sound
+ AudioClip deadSound	Dead sund
+ String font	Font for label

5.1.2 Method

+ void loadResource()	Load all resources

6. Package ui

6.1 Class CharacterScene

6.1.1 Field

+ Stage characterStage	Collect character stage
+ Scene characterScene	Collect character scene
+ boolean chooseHorse	Define if user choose horse
+ boolean chooseGiraffe	Define if user choose giraffe
+ boolean chooseDog	Define if user choose dog
- VBox characterPane	Collect character pane
- HBox dogBox	Show dog icon
- HBox giraffeBox	Show giraffe icon
- HBox horseBox	Show horse icon
- ImageView dogImage	Dog image
- ImageView giraffeImage	Giraffe image
- ImageView horseImage	Horse image
- Label dog	Dog label
- Label giraffe	Giraffe label
- Label horse	Horse label

6.1.2 Constructor

+ CharacterScene()	Initialize stage Set chooseHorse, chooseGiraffe, chooseDog to false Add image to each animal Create label for each animal

6.1.3 Method

- void InitializaStage()	Initialize character stage
- void createCharacter()	Set pane, add all characters and button to this pane Show the stage
- void setPane()	Set character pane
- void animalBox()	Create box for each animal
- void addCharacterLabel()	Create and add character label to this pane
- void addDogCharacter()	Add dog character to this pane
- void addGiraffeCharacter()	Add giraffe character to this pane
- void addHorseCharacter()	Add horse character to this pane
- void addStartButton()	Create and add start button to this pane
- void setOnImage(ImageView image)	Set size for image
- void setOnText(Label label, ImageView image)	Set size, color, font for image and label
- void unExpand()	Set size of animal image which does not chosen
- void highlightText()	Set font and text fill for animal which chosen
- void unHighlightText()	Set font and text fill for animal which does not chosen

6.2 Class DeadSubscene extends MenuSubscene

6.2.1 Field

+ DeadSubscene()	Create MenuSubscene then set graphic

6.2.2 Method

- void setGraphic()	Set graphic of subscene Add highest score and current score label

- Button createBackbutton()	Create back button set on its action, set its position If push this button then game screen will close and menu screen will show
- Button createRetrybutton()	Create retry button set on its action, set its position If push this button then game will restart
- void createHighestScoreLabel()	Create highscore label then add to this pane
- void createCurrentScoreLabel()	Create score label then add to this pane

6.3 Class HowToScene

6.3.1 Field

+ Stage howToStage	Collect how to stage
+ Scene howToScene	Collect how to scene

6.3.2 Constructor

+ HowToScene()	Load all resources and initialize stage

6.3.3 Method

+ void InitializeStage()	Initialize the stage
+ void createHowTo()	Set this pane Add image and home button to this pane
+ void setPane()	Set position of this pane
+ void addHowTo()	Add how to image to this pane
+ void addHomeButton()	Create home button and add to this pane
- void createBackgroundSound()	Initialize Background sound
+ void playBackgroundSound()	Play sound

6.4 Class InfoLabel extends Label

6.4.1 Constructor

+ InfoLabel(String text)	Set label size, padding, alignment, font and

	add text to this label

6.4.2 Method

+ void setLabelFont()	Set font for this label

6.5 Class ScorePane

6.5.1 Field

- Label scoreLabel	Label for score
+ int score	Collect scores
+ int bestScore = 0	Initialize bestScore = 0

6.5.2 Constructor

+ ScorePane()	Create score Label Set label size, padding, alignment, add text to this label Set score = 0, set its font

6.5.3 Method

+ void setFont()	Set font to this label
+ void updateScore()	Update text to current score
+ void updateHighScore()	Update highest score
+ Label getPointsLabel()	Getter for score label
+ void setPoints(int points)	Setter for score

6.6 Class TimerPane implements ThreadTimer

6.6.1 Field

- Thread Timer	Timer
- Label timerLabel	Timer label
- int times	Collect times

6.6.2 Constructor

+ TimerPane()	Create timer Label Set label size, padding, alignment, add text to this label Set times = 0, set font

6.5.3 Method

+ void setFont()	Set font to this label
+ void updateTimer()	Set times = ITEM_TIME/1000 Update timer label If isAlive is true then count down
+ void updateTimeGUI()	Set text in timer label to current time
+ void showTimerPane()	Show timer
+ void hideTimerPane()	Hide timer
+ Label getTimerLabel()	Getter for timer label
+ Thread getTimer()	Getter for timer thread

7. Package ui.base

7.1 Class HeaderLabel extends Label

7.1.1 Field

- final int LABEL_HEIGHT = 70	Define label height to 70
- final int LABEL_WIDTH = 196	Define label width to 196
- final String LABEL_STYLE	Define label style

7.1.2 Constructor

+ HeaderLabel(String text)	Set label size, padding, alignment, add text to this label Set font Set text position in label to center

7.1.3 Method

+ void setLabelFont	Set font to label

7.2 Class MenuButtons extends Button

7.2.1 Field

- final int BUTTON_HEIGHT = 49	Define button height to 49
- final int BUTTON_WIDTH = 190	Define button width to 190
- final String BUTTON_FREE_STYLE	Define button style
- final String BUTTON_PRESSED_STYLE	Define pressed button style

7.2.2 Constructor

+ MenuButtons(String text)	Set button size, padding, alignment, add text to this button Set button style with free button style Initialize button

7.2.3 Method

- void setButtonsFont()	Set font
- void setButtonPressedStyle()	Set button style with pressed button style
- void setButtonReleasedStyle()	Set button style with free button style
- void initializeButton()	Initialize button

7.3 Class MenuSubscene extends Subscene

7.3.1 Field

- Boolean isHidden	Collect if subscene hidden or not

7.3.2 Constructor

+ MenuSubscene(int h, int w)	Create anchorpane to this subscene Set subscene size Fill background image Set isHidden to true Set position

7.3.3 Method

+ void moveSubScene()	Move in subscene to screen and move out
+ AnchorPane getPane()	Getter for pane

8. Package view

8.1 Class Game

8.1.1 Field

- AnchorPane gamePane	Collect gamePane
- Scene gameScene	Collect gameScene
- Coin coin	Collect coins
+ ScorePane scoreLabel	Collect scoreLabel
+ Stage gameStage	Collect gameStage
+ TimerPane timerLabel	Collect timerLabel
+ Bar bar	Collect bars
+ Animal animal	Collect animal
+ Item[] item	Collect all items
+ boolean isPressRight	Define if user press right
+ boolean isPressLeft	Define if user press left
+ boolean isAlive	Define if still alive
+ AnimationTimer GAMETIMER	Collect GAMETIMER
+ ThreadTimer currentItemTimer	Collect ThreadTimer from items

8.1.2 Constructor

+ Game()	Initialize stage and CreateKeyListener

8.1.3 Method

- void InitializeStage()	Initialize game stage

Animal Fall Down

+ void createNewGame()	Hide menu mainStage Load all resources Create Game entities, loop and UI Set isAlive = true Set isPressRight = false Set isPressLeft = false Chow game stage
- void createGameEntities()	Initialize all entities
- void createGameUi()	Initialize all of the UI
- void createKeyListener()	Initialize key listener
- void createGameLoop()	Initialize AnimationTimer
- void createBackgroundSound()	Initialize Background sound
- void Dead()	Stop animation and timer Update highscore

8.2 Class GameSetting

8.2.1 Field

+ final int GAME_HEIGHT = 750	Game height
+ final int GAME_WIDTH = 700	Game width
+ final int BAR_HEIGHT = 600	Bar height
+ final int BAR_WIDTH = 15	Bar width
+ final int BAR_GAP = 150	Bar gap
+ final int REST_GAP = 130	Rest gap
+ final int BAR_NUMBER = 8	Bar number
+ Int gameSpeed = 3	Game speed

8.3 Class Menu

8.3.1 Field

- VBox mainPane	Collect mainPane

- Scene mainScene	Collect mainScene
+ Stage *mainStage*	Collect mainStage

8.3.2 Constructor

- Menu()	Initialize scene, pane, buttons, stage, load resource, play sound and create buttons

8.3.3 Method

- void setPane()	Initialize all entities
- void addGameLogo()	Add logo to the game
- void createHowtoplayButton()	Initialize how to play button
- void createStartButton()	Initialize start button
- void createBackgroundSound()	Initialize Background sound
+ void playBackgroundSound()	Play sound