Programming Report

Created by

Chatree Kurupakorn 6330105721

Charnkij Suksuvankeree 6330104021

2110215 Programming Methodology Semester 1 Year 2020

Chulalongkorn University

1. Game Programming Gameplay : Ha Doz !

1.1 Introduction

You are going back into the game which retrieves your childhood memories because of the theme of this game ! And HaDoz was inspired by a common game on Y8 Platform (ก้าน กล้วย) in a web browser. So, The main objective of this game is to charge and release the power to knock out another player. Finally, for sure to win !

1.2 Gameplay

The rules and elements of this game are stripped down to basic level to make it easier to understand and play. You are bringed back into the street-fighter's tone game place where you can blink and move in 2 cardinal directions (Up - Down). You who charge the power of 3 elements (Fire, Water, Earth) whose power is ordered by fire > earth > water > fire and collect the item (Shield, HpPotion, DmPotion) to release and knock out another player to win the game !

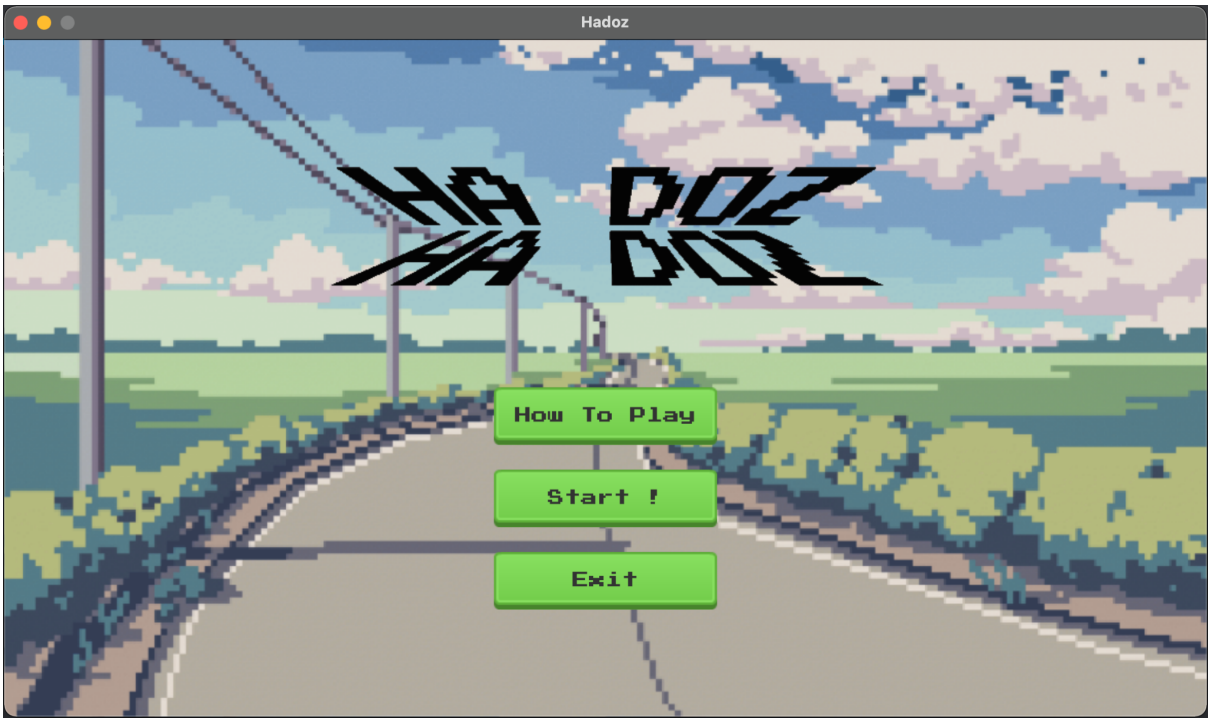


Fig. 1: MainMenu Scene

Fig. 2: HowToPlay Scene

Fig. 3: GamePlay Scene

Blocked Once !

Fig. 4: Item-Shield

Hp up ! +15

Fig. 5: Item-HpPotion

Damage up ! +10

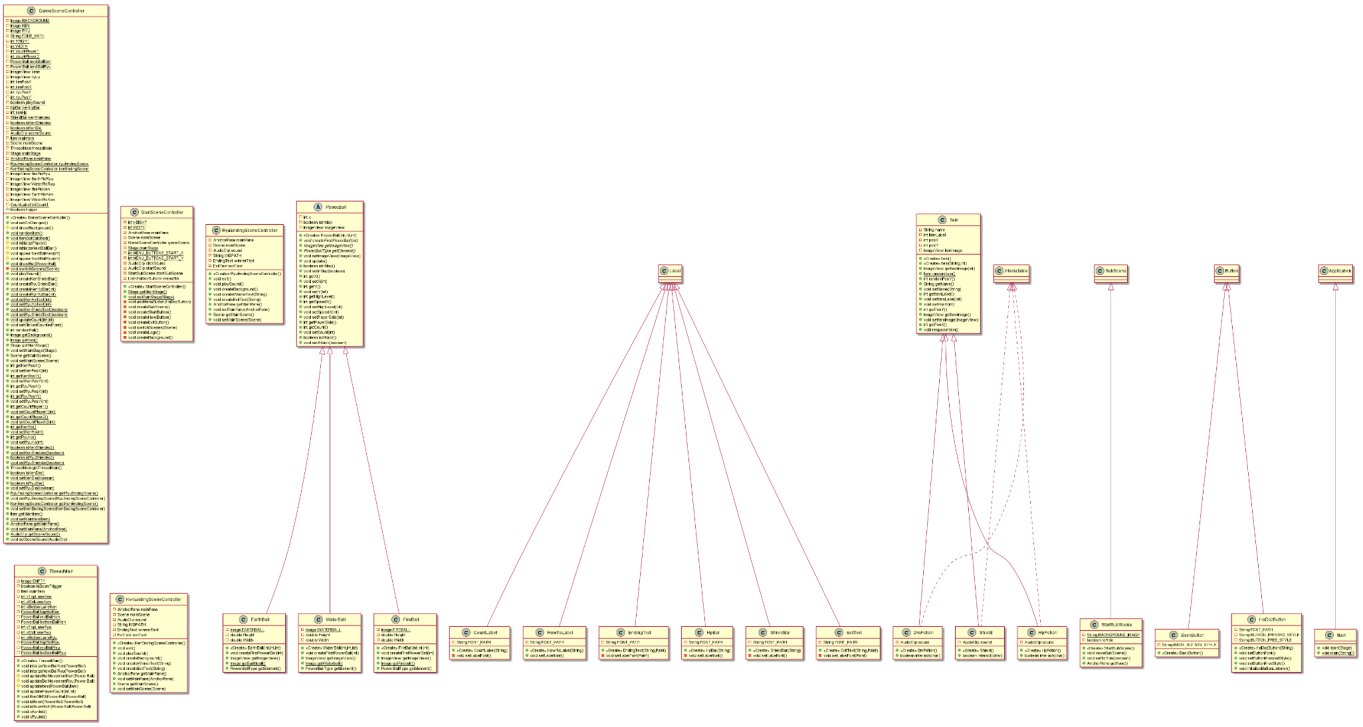
Fig. 6: Item-DmPotion

Fig. 7: Gameplay Scene when ken catched the shield

Fig. 8: Ending Scene when KEN is the winner

Fig. 9: Ending Scene when RYU is the winner

2.Class diagram



3. Code explanation

* Noted that Access Modifier Notations can be listed below

+ (public), underlined (static)

(protected), ALL_CAPS (final)

- (private), *italic (abstract)*

3.1 Package Application

3.1.1 Class Main extends Application

This class represents the game launcher itself, where the method start is in.

3.1.1.1 Methods

Name	Description
+ void start (Stage primaryStage)	- launch up the new StartScreen of the game - show the primaryStage
+ void main (String[ ] args)	the entry point of the application

3.1.2 Class ThreadMain

This class is threads creations for new threads that occur.

3.1.2.1 Fields

Name	Description
- final Image EMPTY	The empty image
- int Item mainItem	Item to keep track of the item on GameScene for checking collidation.

- int xTopLaneKen	Store values in the X-axis of the PowerBall in front of the top lane for Ken.
- int xMidLaneKen	Store values in the X-axis of the PowerBall in front of the mid lane for Ken.
- int xBottomLaneKen	Store values in the X-axis of the PowerBall in front of the bottom lane for Ken.
- PowerBall topBallKen	Keep a reference of the PowerBall in xTopLaneKen.
- PowerBall midBallKen	Keep a reference of the PowerBall in xMidLaneKen.
- PowerBall bottomBallKen	Keep a reference of the PowerBall in xBottomLaneKen.
- int xTopLaneRyu	Store values in the X-axis of the PowerBall in front of the top lane for Ryu.
- int xMidLaneRyu	Store values in the X-axis of the PowerBall in front of the mid lane for Ryu.
- int xBottomLaneRyu	Store values in the X-axis of the PowerBall in front of the bottom lane for Ryu.
- PowerBall topBallRyu	Keep a reference of the PowerBall in xTopLaneRyu.
- PowerBall midBallRyu	Keep a reference of the PowerBall in xMidLaneRyu.

-PowerBall bottomBallRyu	Keep a reference of the PowerBall in xBottomLaneRyu.
- boolean isBoomTrigger	Make sure that isBoom() runs only once.

3.1.2.2 Constructor

Name	Description
+ ThreadMain ( )	

3.1.2.3 Methods

Name	Description
+ void initalizeNewBallKen(PowerBall ball)	Create new Thread(), in thread call methods xKenInit(), xRyuInit() and updateBallMovementKen(ball).
+ void initalizeNewBallRyu(PowerBall ball)	Create new Thread(), in thread call methods xKenInit(), xRyuInit() and updateBallMovementRyu(ball).
# void updateBallMovementKen(PowerBall ball)	Update PowerBall movement in map and check whether the PowerBall collides or not.
# void updateBallMovementRyu(PowerBall ball)	Update PowerBall movement in map and check whether the PowerBall collides or not.
# void updateItem(PowerBall b,Item item)	Update position between the Ball and the Item if on the same position, effect to player

	following item's effect. and call itemGotCatched to delete the item off the map.
+ void updatePlayerCount(int count1,int count2)	Update count display on GameSceneController
+ void BooMMM(PowerBall BallKen, PowerBall BallRyu)	Consider the condition of the element. Fire->Water->Earth->Fire. For example, if WaterBall collide with FireBall then count of WaterBall will +5
+ void isBoom(PowerBall lossBall,PowerBall winBall)	Follow up after the PowerBalls collide. count of winBall -= count of lossBall and lossBall will disappear.
+ void isBoomBoth(PowerBall BallKen, PowerBall BallRyu)	Check if PowerBalls collide and same power. Then both of them will disappear.
+ void xKenInit()	Initialize xTopLaneKen, xMidLaneKen, xBottomLaneKen.
+ void xRyuInit()	Initialize xTopLaneRyu, xMidLaneRyu, xBottomLaneRyu.

3.2 Package component

3.2.1 Class BackButton extends Button

This class represents the back button of Subscene which is clicked then closes the StartSubScene.

3.2.1.1 Fields

Name	Description
- String BACK_BUTTON_STYLE	A string to keep the style of BackButton.

3.2.1.2 Constructor

Name	Description
+ BackButton ( )	Initializes the Button - set the preferred width to 37. - set the preferred height to 36. - set style from BACK_BUTTON_STYLE.

3.2.2 Class CountLabel extends Label

This class represents the count label that will show the strength of your powerBall !

3.2.2.1 Fields

Name	Description
- String FONT_PATH	A string to keep the path of Font.

3.2.2.2 Constructor

Name	Description
+ CountLabel (String text)	Initializes the Label - set the preferred width to 300.

	- set the preferred height to 50 - set the inset padding of 40. - set Text to “Power” + text. - set wrap Text be true. Instantiates property of a node by setLabelFont().

3.2.2.3 Methods

Name	Description
- void setLabelFont ( )	set this Label’s Font by loading from FONT_PATH.

3.2.3 Class EndingText extends Label

This class represents a Label that has properties to be EndingText.

3.2.3.1 Fields

Name	Description
- String FONT_PATH	A string to keep the path of Font.

3.2.3.2 Constructor

Name	Description
+ EndingText (String text, Paint color)	Initializes the Label - set the preferred width to 900.

	- set the preferred height to 75. - set the inset padding of 40. - set Text to text. - set wrap Text be true. Instantiates property of a node by setLabelFont(color).

3.2.3.3 Methods

Name	Description
- void setLabelFont (Paint color)	- set this Label's Font by loading from FONT_PATH. - set this Text fill Color by color.

3.2.4 Class ExitText extends Label

This class represents a Label that has properties to be ExitText.

3.2.4.1 Fields

Name	Description
- String FONT_PATH	A string to keep the path of Font.

3.2.4.2 Constructor

Name	Description
+ ExitText (String text, Paint color)	Initializes the Label - set the preferred width to 800.

	- set the preferred height to 75. - set the inset padding of 40. - set Text to text. - set wrap Text be true. Instantiates property of a node by setLabelFont(color).

3.2.4.3 Methods

Name	**Description**
- void setLabelFont (Paint color)	- set this Label's Font by loading from FONT_PATH . - set this Text fill Color by color.

3.2.5 Class HaDozButton extends Button

This class represents the HaDozButton by interest specific in the style of the button.

3.2.5.1 Fields

Name	**Description**
- String FONT_PATH	A string to keep the path of Font.
- String BUTTON_PRESSED_STYLE	A string to keep the style of HaDozButton when pressed.
- String BUTTON_FREE_STYLE	A string to keep the style of HaDozButton when released.

3.2.5.2 Constructor

Name	Description
+ HaDozButton (String text)	Initializes the Button - set the preferred width to 190. - set the preferred height to 49. - set Text to text. - set style from BUTTON_FREE_STYLE. - complete the handle method by initializeButtonListener ( ). Instantiates property of node by - setButtonFont ( ).

3.2.5.3 Methods

Name	Description
+ void setButtonFont ( )	set this Button's Font by loading from FONT_PATH.
+ void setButtonPressedStyle ( )	- set this Button's Style by BUTTON_PRESSED_STYLE. - set this preferred height to 45. - set this layout Y axis from current + 4.
+ void setButtonFreeStyle ( )	set this Button's Style by BUTTON_FREE_STYLE. - set preferred height to 49.

	- set this layout Y axis from current - 4.
+ void initializeButtonListener ( )	This method is the handler when the button is actioned. If there is a button pressed , set button style by method setButtonPressedStyle(). If there is a button pressed, set button style by method setButtonFreeStyle().

3.2.6 Class HowToLabel extends Label

This class represents a Label that has properties to be HowToLabel.

3.2.6.1 Fields

Name	Description
- String FONT_PATH	A string to keep the path of Font.

3.2.6.2 Constructor

Name	Description
+ HowToLabel (String text)	Initializes the Label - set the preferred width to 750. - set the preferred height to 500. - set the inset padding of 40. - set Text to text. - set wrap Text be true. Instantiates property of a node by

	setLabelFont().

3.2.6.3 Methods

Name	**Description**
- void setLabelfont ( )	- set this Label's Font by loading from FONT_PATH .

3.2.7 Class HpBar extends Label

This class represents a Label that has properties to be HpBar on GameScene.

3.2.7.1 Fields

Name	**Description**
- String FONT_PATH	A string to keep the path of Font.

3.2.7.2 Constructor

Name	**Description**
+ HpBar (String text)	Initializes the Label - set the preferred width to 300. - set the preferred height to 50.. - set the inset padding of 40. - set Text to text. - set wrap Text be true.

	Instantiates property of a node by setLabelFont().

3.2.7.3 Methods

Name	Description
- void setLabelfont ( )	- set this Label's Font by loading from FONT_PATH .

3.2.8 Class ShieldBar extends Label

This class represents a Label that has properties to be Shield Bar on GameScene.

3.2.8.1 Fields

Name	Description
- String FONT_PATH	A string to keep the path of Font.

3.2.8.2 Constructor

Name	Description
+ ShieldBar (String text)	Initializes the Label - set the preferred width to 350. - set the preferred height to 50.. - set the inset padding of 40. - set Text to text.

	- set wrap Text be true. Instantiates property of a node by setLabelFont().

3.2.8.3 Methods

Name	Description
- void setLabelfont ()	- set this Label's Font by loading from FONT_PATH .

3.2.9 Class StartSubScene extends SubScene

This class represents a SubScene of StartScene to show How to play this game.

3.2.9.1 Fields

Name	Description
- String BACKGROUND_IMAGE	A string to keep the path of BACKGROUND_IMAGE.
- boolean toHide	A boolean to keep track whether StartSubScene hides or not.

3.2.9.2 Constructor

Name	Description
+ StartSubScene ( )	call super class constructor (AnchorPane)

	and setSize (750, 500) Initializes the SubScene - set the preferred width to 750. - set the preferred height to 500. - initialize BackgroundImage by loading from BACKGROUND_IMAGE. - set background to subroot which is AnchorPane by Down-casting this root. - set layout x-axis to 150. - set layout y-axis to -500.

3.2.9.3 Methods

Name	**Description**
- void moveSubScene ( )	if not to Hide, translate this node to 538 on y-axis.else translate this node to -538 on y-axis.
- void setToHide(boolean Hide) - AnchorPane getPane ( );	getter/setter method.

3.3 Package entity extends Item implements Interactable

3.3.1 Class Item

This class represents the item in this game and position , name of the item.

3.3.1.1 Fields

Name	Description
- String name	A string to keep the name of Item.
- int itemLabel	An Integer to keep the label of Item.
- int posX	An Integer to keep the x-axis position of Item.
- int posY	An Integer to keep the y-axis position of Item.
- ImageView itemImage	A ImageView to keep the image of Item.
- boolean isCollision	A boolean to keep status whether Item is collided by PowerBall or not.

3.3.1.2 Constructor

Name	Description
+ Item()	- set item name to “none”. - set item label to -1.
+ Item(String name, int Label)	- set item name to parameter name. - set item label to parameter label.

3.3.1.3 Methods

Name	**Description**
+ ImageView getItemImage(int itemLabel)	switch up to case: 0: return image of shield. 1: return image of hpPotion. 2: return image of dmPotion. else throw IllegalArgumentException.
+ void setName(String name)	
+ Item randomItem()	random the label to random initialize the item, switch up to case: 0: return Shield. 1: return HpPotion. 2: return DmPotion. else throw IllegalArgumentException.
+ int randomPosY()	random the position on y-axis to random set position on y-axis of item,switch up to case: 0: setPosY(340) return position on y-axis. 1: setPosY(170) return position on y-axis. 2: setPosY(0) return position on y-axis. else throw IllegalArgumentException
+ void respawnItem()	respawn random items all the time between gameplay scenes.
+ getter/setter all fields	getter / setter method.

3.3.2 Class DmPotion extends Item implements Interactable

This class DmPotion type of Item. It is a solid object. However, it can respawn randomly on the y-axis and affect the player by powering up the power.

3.3.2.1 Fields

Name	**Description**
- AudioClip sound	An AudioClip to keep the audio when drank the dmPotion.

3.3.2.1 Constructor

Name	**Description**
+ DmPotion()	call super constructor to - set name to “dmPotion”. - set label to 2.

3.3.2.2 Methods

Name	**Description**
+ boolean interact(char side)	check if the side is left, set count or power of ken +10 and playsoundeffect.else if the side is right, do the same to ryu.

3.3.3 Class HpPotion extends Item implements Interactable

This class HpPotion type of Item. It is a solid object. However, it can respawn randomly on the y-axis and affect the player by healing the hp.

3.3.3.1 Fields

Name	**Description**
- AudioClip sound	An AudioClip to keep the audio when drank the hpPotion.

3.3.3.2 Constructor

Name	**Description**
+ HpPotion()	call super constructor to - set name to “hpPotion”. - set label to 1.

3.3.3.3 Methods

Name	**Description**
+ boolean interact(char side)	check if the side is left, set hp of ken +15 and playsoundeffect.else if the side is right, do the same to ryu.

3.3.4 Class Shield extends Item implements Interactable

This class Shield type of Item. It is a solid object. However, it can respawn randomly on the y-axis and affect the player by Blocking once.

3.3.4.1 Fields

Name	**Description**
- AudioClip sound	An AudioClip to keep the audio when catched the shield.

3.3.4.1 Constructor

Name	**Description**
+Shield()	call super constructor to - set name to “shield”. - set label to 0.

3.3.4.2 Methods

Name	**Description**
+ boolean interact(char side)	check if the side is left, set isKenShielded of ken to be true and playsoundeffect.else if the side is right, do the same to ryu.

3.3.5 Class PowerBall

This class represents a PowerBall in this game. This class will be used for managing all the balls.

3.3.5.1 Fields

Name	Description
- int x	An Integer to keep the x-axis position of PowerBall.
- int y	An Integer to keep the y-axis position of PowerBall.
- int highLevel	An Integer to keep the current level of PowerBall.(3 level : Top, Mid, Bottom)
- int speedX	An Integer to keep the speed of PowerBall.
- int playerSide	An Integer to keep the side of PowerBall. that come from left : < 0 and right : >0. if playerSide<0 this PowerBall comes from Ryu. if playerSide>0 this PowerBall comes from Ken.
- int count	An Integer to keep the count that means power of PowerBall.
- boolean isInMap	A boolean to track PowerBall whether to be in map or not.

- isAttack	A boolean to track whether PowerBall gets collided or not.
# ImageView imageView	A ImageView to keep the image of PowerBall.

3.3.5.2 Constructor

Name	**Description**
+ PowerBall(int x, int highLevel, int playerSide)	initialize all private fields except count

3.3.5.3 Methods

Name	**Description**
+ void createFirstPowerBall(int count)	Abstract method.
+ void ImageView getImageView()	Abstract method.
+ PowerBallType getElement()	Abstract method.
+ void update()	Update the position of Ball and and check if the ball is still on the map.
+ getter/setter	getter / settet methods.

3.3.6 Class FireBall extends PowerBall

This class represents the fire ball that extends from PowerBall. to configure an ImageView of PowerBall and separate elements.

3.3.6.1 Fields

Name	**Description**
- Image FIREBALL	Image of FIREBALL
- double Height	FIREBALL preference height
- double Width	FIREBALL preference width

3.3.6.2 Constructor

Name	**Description**
+ FireBall(int x, int highLevel, int playerSide)	-> Initialize PowerBall -> Initialize imageView with FIREBALL -> Initialize Height, Width with preference size of imageView

3.3.6.3 Methods

Name	**Description**
+ void createFirstPowerBall(int count)	Initialize size of ball according to count
+ ImageView getImageView()	return imageView
+ PowerBallType getElement()	return PowerBallType.FIRE

+ Image getFireballl()	getter of |FIREBALL

3.3.7 Class EarthBall extend PowerBall

This class represents the earth ball that extends from PowerBall. to configure an ImageView of PowerBall and separate elements.

3.3.7.1 Fields

Name	Description
- Image EARTHBALL	Image of EARTHBALL
- double Height	EARTHBALL preference height
- double Width	EARTHBALL preference width

3.3.7.2 Constructor

Name	Description
+ EarthBall(int x, int highLevel, int playerSide)	-> Initialize PowerBall -> Initialize imageView with EARTHBALL -> Initialize Height, Width with preference size of imageView

3.3.7.3 Methods

Name	Description
+ void createFirstPowerBall(int count)	Initialize size of ball according to count
+ ImageView getImageView()	return imageView

+ PowerBallType getElement()	return PowerBallType.EARTH
+ Image getEarthball()	getter of |EARTHBALL

3.3.8 Class WaterBall extend PowerBall

This class represents the water ball that extends from PowerBall. to configure an ImageView of PowerBall and separate elements.

3.3.8.1 Fields

Name	Description
- Image WATERBALL	Image of WATERBALL
- double Height	WATERBALL preference height
- double Width	WATERBALL preference width

3.3.8.2 Constructor

Name	Description
+ WaterBall(int x, int highLevel, int playerSide)	-> Initialize PowerBall -> Initialize imageView with WATERBALL -> Initialize Height, Width with preference size of imageView

3.3.8.3 Methods

Name	Description
+ void createFirstPowerBall(int count)	Initialize size of ball according to count

+ ImageView getImageView()	return imageView
+ PowerBallType getElement()	return PowerBallType.WATER
+ Image getWaterball()	getter of |WATERBALL

3.3.9 public enum PowerBallType

This enum contains the values that can be used to compare elements of PowerBall. There are three enum types here: PowerBallType.FIRE, PowerBallType.EARTH, PowerBallType.WATER.

3.4 Package entity.base

3.4.1 Interface Interactable

3.4.1.1 Method

Name	Description
+ Boolean interact (char side)	This method is called when the Ball collides with Item and we will check the interaction between Item and Ball to select the effect to each other side. It returns true if Item is used. Otherwise, return false.

3.5 Package scene.controller

3.5.1 Class StartSceneController

This class is the control the start scene.

3.5.1.1 Fields

Name	Description
- int HEIGHT	- set HIGHT to 576.
- int WIDTH	- set WIDTH to 1024.
- AnchorPane mainPane	An AnchorPane to keep childrens of mainPane.
- Scene mainScene	A Scene prepared to show on mainStage.
- GameSceneController gameScene	A GameSceneController prepares to switch when pressed start on mainScene.
- Stage mainStage	A Stage prepared to show on Main.java.
- int MENU_BUTTONS_START_X	- set MENU_BUTTONS_START_X to 417.
- int MENU_BUTTONS_START_Y	- set MENU_BUTTONS_START_Y to 296.

- AudioClip clickSound	An AudioClip to keep the audio when clicked the button.
- AudioClip startSound	An AudioClip to keep the audio when the game started.
- StartSubScene startSubScene	A StartSubScene that will show when clicked on the HowToPlay Button.
- List<HaDozButton> menuBtn	A List contains all buttons in StartScene.

3.5.1.2 Constructor

Name	**Description**
+ StartSceneController()	Initialize all private fields. Instantiates each node and set to fields: - createHowButton( ). - createStartButton( ). - createExitButton( ). - createLogo( ). - createSubScene( ).

3.5.1.3 Methods

Name	**Description**
- void addMenuButton(HaDozButton btn)	add the HaDozButton in the correct ordering on mainPane.
- void createSubScene()	Instantiates each node and adds it to

	StartSubScene. - label is a Title of SubScene. - panel is a Content of SubScene. - Ryu, Ken are characters from GameScene. - bBtn is a BackButton which is clicked then the SubScene will be closed.
- void createStartButton()	Instantiates each node and adds it to StartScene. - sBtn is a start Button which is clicked then plays startSound and switches to GameScene by SwitchScenes(Scene) method.
- void createHowButton()	Instantiates each node and adds it to StartScene. - hBtn is a howtoplay Button which is clicked then plays clickedSound and opens the StartSubScene .
- void createExitButton()	Instantiates each node and adds it to StartScene. - eBtn is an exit Button which is clicked then plays clickedSound and closes the game.
- void switchScenes(Scene scene)	- setScene(parameter scene) to mainStage. - play beginningthegameSound.
- void createLogo()	Instantiates each node and adds it to StartScene.

	- logo is an ImageView which is the name of this game !
- void createBackground()	Instantiates each node and adds it to StartScene. - backgroundImage is an ImageView which is the background of this game ! - setBackground to mainPane.
+ getter/setter	getter / setter method.

3.5.2 Class GameSceneController

This class displays when the game begins. And controller all events that occur on keyEvent.

3.5.2.1 Fields

Name	Description
- Image BACKGROUND	An Image that keeps the image of Background.
- Image KEN	An Image that keeps the image of Ken.
- Image RYU	An Image that keeps the image of Ryu.
- String FONT_PATH	Keeps the path to the font's file location.
- int HEIGHT	Height of the game scene.
- int WIDTH	Width of the game scene.
- int countPlayer1	Count how many times that player1 pressed “A” or “D”.
- int countPlayer2	Count how many times that player2 pressed “RIGHT” or “LEFT”.
- PowerBall nextBallKen	Keep the next random PowerBall of Ken.
- PowerBall nextBallRyu	Keep the next random PowerBall of Ryu.
- ImageView kenn	Keep ImageView from Image KEN.
- ImageView ryuu	Keep ImageView from Image RYU.
- int kenPosX	Keep a position of Ken in the X axis = 70.
- int kenPosY	Keep a position of Ken in the Y axis.
- int ryuPosX	Keep a position of Ryu in the X axis = 900.

- int ryuPosY	Keep a position of Ryu in the Y axis.
- int kenHp	Keep HP of Ken and initialize with 100.
- int ryuHp	Keep HP of Ryu and initialize with 100.
- boolean isKenShielded	Check if Ken has a shield or not.
- boolean isRyuShielded	Check if Ryu has a shield or not.
- HpBar kenHpBar	To display a HP bar of Ken.
- HpBar ryuHpBar	To display a HP bar of Ryu.
- boolean isKenDie	Check if Ken is dead or not.
- boolean isRyuDie	Check if Ryu is dead or not.
- AudioClip sceneSound	Keep an audio clip and play in Background when the game starts.
- Item mainItem	Keep an item that spawns on the map.
- AnchorPane mainPane	Keep an anchor pane and add it to mainScene.
- Scene mainScene	A scene that shows on mainStage.
- Stage mainStage	A stage that displays on the window.
- ThreadMain threadMain	It is a class that collects the usage of threads.
- RyuEndingSceneController ryuEndingScene	Show the ending scene of Ryu when he wins.
- KenEndingSceneController	Show the ending scene of Ken when he

kenEndingScene	wins.
- ImageView firePicRyu	Keep the FIREBALL ImageView of Ryu.
- ImageView EarthPicRyu	Keep the EARTHBALL ImageView of Ryu.
- ImageView WaterPicRyu	Keep the WATERBALL ImageView of Ryu.
- ImageView firePicKen	Keep the FIREBALL ImageView of Ken.
- ImageView EarthPicKen	Keep the EARTHBALL ImageView of Ken.
- ImageView WaterPicKen	Keep the WATERBALL ImageView of Ken.
- CountLabel txtCount1	To display a count number of Ken on display.
- CountLabel txtCount2	To display a count number of Ryu on display.
+ boolean trigger	trigger used to trigger a keypress event

3.5.2.2 Constructor

Name	**Description**
+ GameSceneController()	Initialize all fields, play sceneSound, set background and call setOnCharged().

3.5.2.3 Methods

Name	Description
+ void setOnCharged()	This method will handle all keypresses that occur in this scene. - Pressing “A” or “D” will increase countPlayer1 by 1. - Pressing “W” or “S” will relocate Ken up or down - Pressing “SPACEBAR” will release PowerBall of Ken. and set countPlayer1 to 0. - Pressing “RIGHT” or “LEFT” will increase countPlayer2 by 1. - Pressing “UP” or “DOWN” will relocate Ryu up or down - Pressing “ENTER” will release PowerBall of Ryu and set countPlayer2 to 0.
+ void setClickedCountedFont()	Initailize txtCount1 and txtCount2.
# void drawBackground()	Set background of mainPane to BACKGROUND
+ void randomItem()	This method will randomly pick an item from DmPotion, HpPotion and Shield.
+ void itemGotCatched()	Remove an item when it got catched.
# void initializePlayer()	Initialize Ken and Ryu with a picture, relocate to the right position and add to

	mainPane.
# void initializeNextBallBar()	Initialize a ball on each side. to display the next PowerBall.
# void appearNextBallKen(int r)	Display the next PowerBall of Ken. r=0 is FireBall, r=1 is EarthBall, r=2 is WaterBall.
# void appearNextBallRyu(int r)	Display the next PowerBall of Ryu. r=0 is FireBall, r=1 is EarthBall, r=2 is WaterBall.
+ int randomBall()	Random number in range 0-2. 0=fireBall,1=earthBall,2=waterBall
- void switchScenes(Scene scene)	Used to switch GameSceneController to EndingSceneController of Ken or Ryu.
+ void playSound()	Set the properties of sceneSound. Volume = 10, Cycle count =100 and play it.
+ void createKenHpBar(int khp)	Initialize Ken HP bar and add it to mainPane.
+ void createRynHpBar(int rhp)	Initialize Ryu HP bar and add it to mainPane.
+ void updateCount(int count1, int count2)	This method is called by updatePlayerCount in threadMain to update and display the count of each side in real time.
+ void runBackgroundSound()	Play sceneSound

+ void drawBall(PowerBall ball)	This method use in updateBallMovement in threadMain to draw a movement of PowerBall
+ getter/setter	All getter/setter of fields.

3.5.3 Class KenEndingSceneController

This class shows the ending scene when Ken is the winner.

3.5.3.1 Fields

Name	**Description**
- AnchorPane mainPane	An AnchorPane to keep childrens of mainPane.
- Scene mainScene	A Scene prepared to switch when isKenDie.
- String IMGPATH	Keep IMGPATH = "/scene/controller/res/ken-ending.gif"
- EndingText winnerText	Text that shows the name of the winner !
- ExitText exitText	Text that shows how to exit the game.

3.5.3.2 Constructor

Name	Description
+ KenEndingSceneController()	Initialize all private fields. Instantiates each node and set to fields: - createBackground( ). - createWinnerText(“Winner!! is K E N”). - createExitText(“Press to space exit.”). - exit()

3.5.3.3 Methods

Name	Description
+ void createBackground()	Instantiates each node and adds it to StartScene. - kenEnding is an Image which is the background of this ken-ending-scene ! - setBackground to mainPane.
+ void createExitText(String text)	Instantiates each node and adds it to StartScene. - exitText is a Text which tells how to exit the game !
+ void createWinnerText(String text)	Instantiates each node and adds it to StartScene. - winnerText is a Text which tells who is the winner of this game !

+ getter/setter	All fields getter / setter method.

3.5.4 Class RyuEndingSceneController

This class shows the ending scene when Ryu is the winner.

3.5.4.1 Fields

Name	**Description**
- AnchorPane mainPane	An AnchorPane to keep childrens of mainPane.
- Scene mainScene	A Scene prepared to switch when isRyuDie.
- String IMGPATH	IMGPATH = "/scene/controller/res/ken-ending.gif"
- EndingText winnerText	Text that shows the name of the winner !
- HpBar winnerText	Text that shows how to exit the game.

3.5.3.2 Constructor

Name	Description
+ RyuEndingSceneController()	Initialize all private fields. Instantiates each node and set to fields: - createBackground( ). - createWinnerText(“Winner!! is R Y U”). - createExitText(“Press to space exit.”). - exit()

3.5.3.3 Methods

Name	Description
+ void createBackground()	Instantiates each node and adds it to StartScene. - ryuEnding is an Image which is the background of this ryu-ending-scene ! - setBackground to mainPane.
+ void createExitText(String text)	Instantiates each node and adds it to StartScene. - exitText is a Text which tells how to exit the game !
+ void createWinnerText(String text)	Instantiates each node and adds it to StartScene. - winnerText is a Text which tells who is the

	winner of this game !
+ getter/setter	All fields getter / setter method.