Train to the beat

The Rhythm RPG game

Suthida Sukhonunthikul 5931073121

Arin Trongsantipong 5931075421

Train to the beat

Brief Introduction about Project

This project is a project that made from JavaFX. It's a Rhythm RPG game. The main point of this game is to defeat the boss by leveling up characters through a rhythm gameplay. Heroes and boss also have various effects of skill that made the game more exciting. The difficulty of this game is in normal level. Therefore, it's suitable for all players.

How to play

When you enter a game you will see the welcome screen. You must press [Enter] key to enter main menu screen. If you press the other key the suggestion message will be shown.

Figure 1 : Welcome Screen

Figure 2 : Welcome Screen with message

After you pressed [Enter] the main menu screen will be shown.

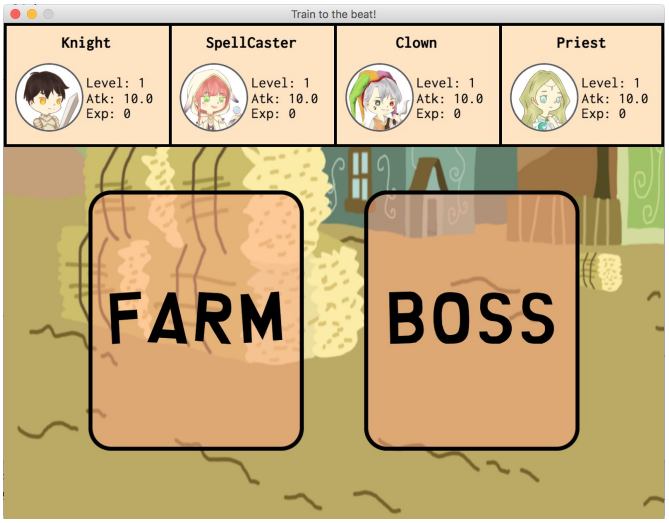


Figure 3 : Main Menu Screen

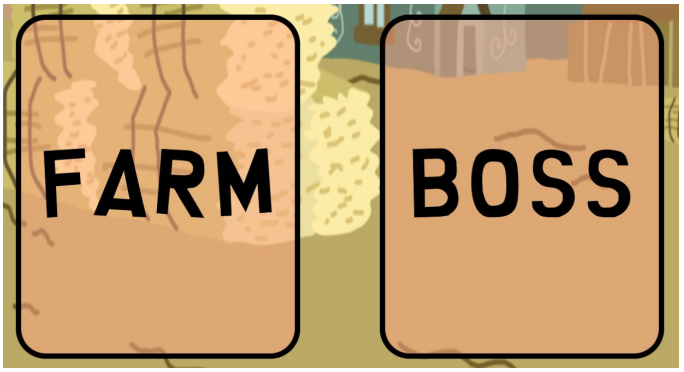


Figure 4 : Select mode buttons

Figure 5 : Hero's status

In Main Menu Screen, it will show current hero status which are level, attack, exp and also show the game mode buttons. There are two modes in this game, "Farm" and "Boss". The objective of this game is to defeat the boss but you have to gain level from the Farm mode first. If you click Farm button, the Character Select Screen will be shown.

Figure 6 : Character Select Screen

Figure 7 : Character Select Screen title

Figure 8 : Heroes select button

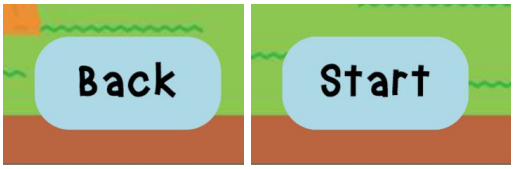


Figure 9 : Back and Start button

In Character Select Screen, you have to choose 1 out of 4 heroes to play which are Knight, SpellCaster, Clown and Priest. Each hero has their own unique skill. The detail is below.

Knight : There is a chance to multiple his attack by 1.2 for 5 seconds (Cooldown : 10 secs)

SpellCaster : There is a chance to guarantee the judge to get at least Perfect for 5 seconds (Cooldown : 15 secs)

Clown : There is a chance to change next 5-15 notes direction to be the same (Cooldown : 13 secs)

Priest : **Boss Mode Only!** There is a chance to nullify and silence the boss to make its unable to activate a skill for 5 seconds (Cooldown : 10 secs)

When you selected a character and press [Start] button you will move to Gameplay Screen. In gameplay, there are 5 types of judge based on your performance the music of each game will be random. If you tap correctly on the beat, each judge is a multiplier for the hero's attack. (No attack for miss)

Critical Perfect : Multiply attack by 1.1

Perfect : Multiply attack by 1

Great : Multiply attack by 0.8

Good : Multiply attack by 0.6

Figure 10 : Game Play Screen

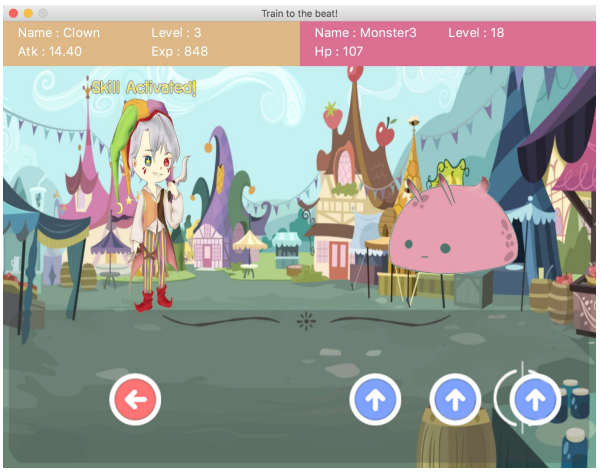


Figure 11 : Skill Activated

Figure 12 : Hero

Figure 13 : Monster

After you gain a level from Farm mode for each character you might go challenge the boss. In Boss mode, the attack will be the summation of all attacks of heroes and every hero can activate his skill in this mode. The Boss is an irregular monster that has its own skill just like the hero.

Boss : There is a chance to reduce one hero's attack to 0 for 5 seconds (Cooldown : 8 secs)

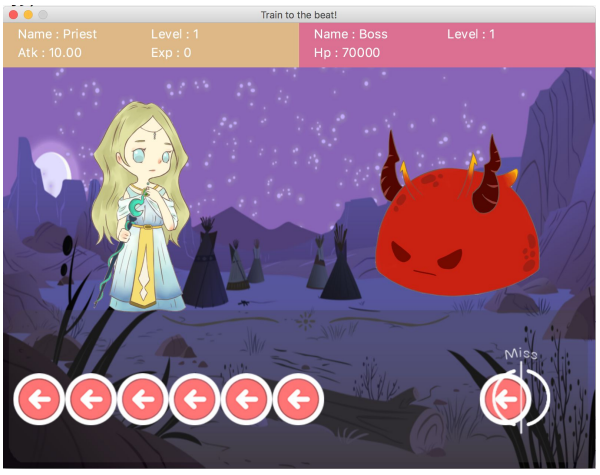


Figure 14 : Boss Mode

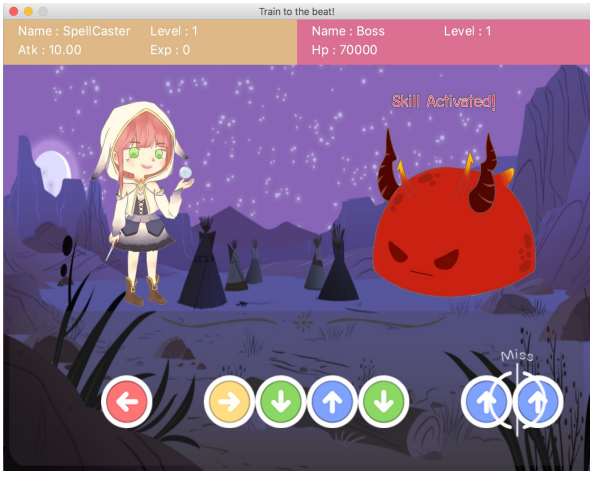


Figure 15 : Boss's skill activated

Figure 16 : Boss

You can manually end the current game by hitting [ESCAPE] key.

When the game end you will move to Result Screen, this screen will show your performance of your last game. The first Result Screen will show how many in each judge type you get and in the second Result Screen will show the before and after status of hero but only in the farm mode.

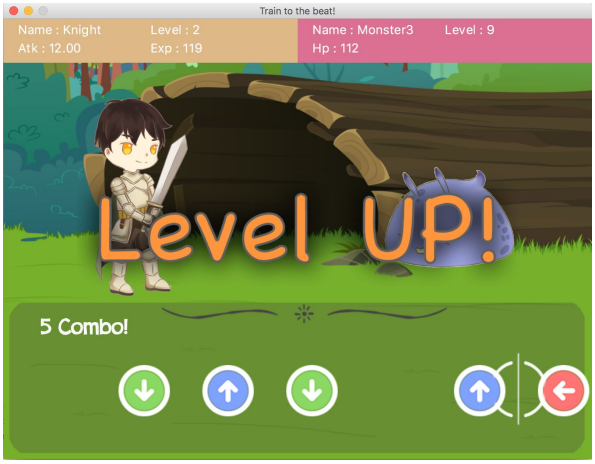


Figure 17 : Level up alert

Figure 18 : Fail alert

Figure 19 : Result Screen 1

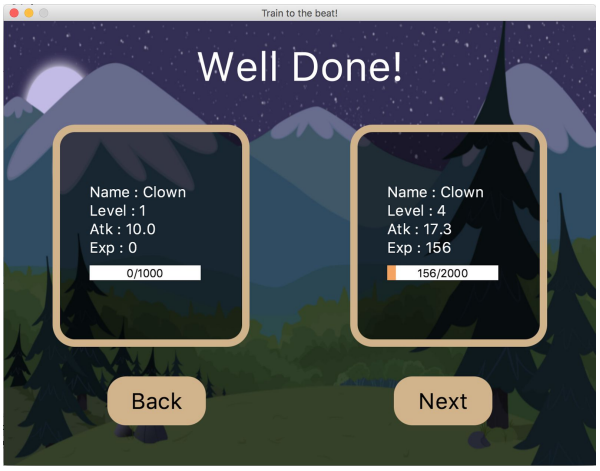
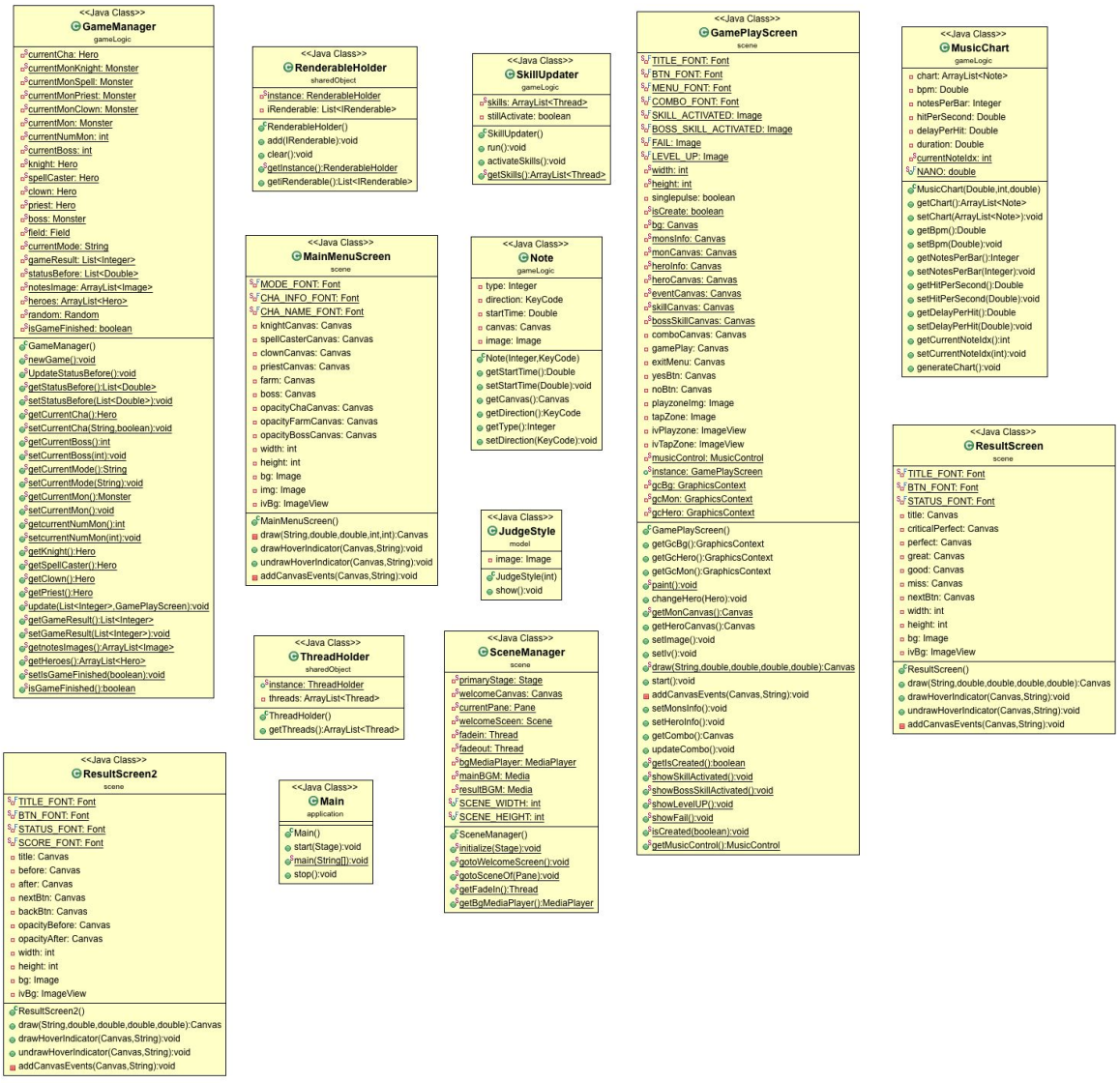
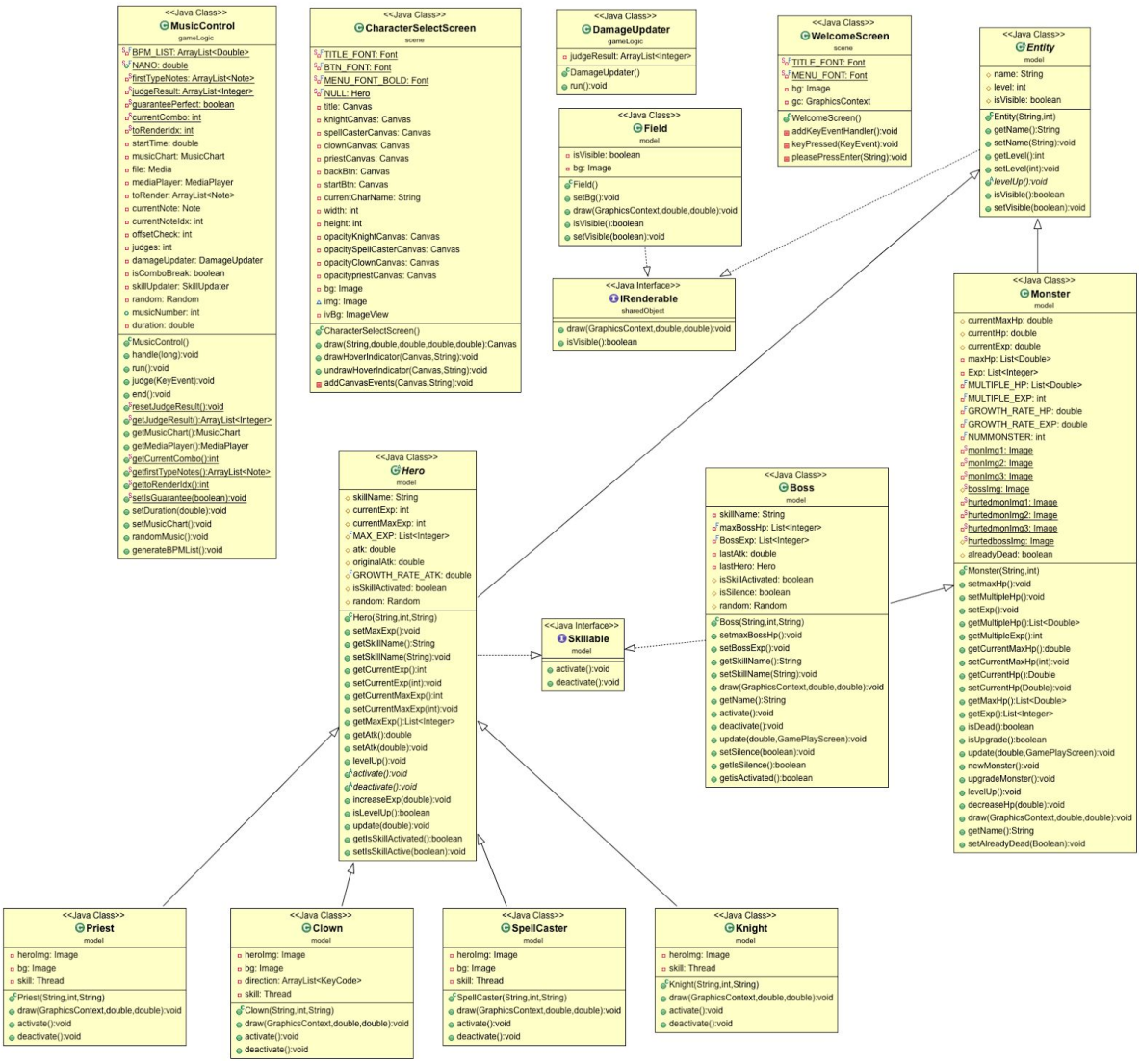


Figure 20 : Result Screen 2 (Farm mode only)

UML Diagram


<<Java Class>>
GameManager
gameLogic
currentCha: Hero
currentMonKnight: Monster
currentMonSpell: Monster
currentMonPriest: Monster
currentMonClown: Monster
currentMon: Monster
currentNumMon: int
currentBoss: int
knight: Hero
spellCaster: Hero
clown: Hero
priest: Hero
boss: Monster
field: Field
currentMode: String
gameResult: List<Integer>
statusBefore: List<Double>
notesImage: ArrayList<Image>
heroes: ArrayList<Hero>
random: Random
isGameFinished: boolean
GameManager()
newGame():void
UpdateStatusBefore():void
getStatusBefore():List<Double>
setStatusBefore(List<Double>):void
getCurrentCha():Hero
setCurrentCha(String,boolean):void
getCurrentBoss():int
setCurrentBoss(int):void
getCurrentMode():String
setCurrentMode(String):void
getCurrentMon():Monster
setCurrentMon():void
getcurrentNumMon():int
setcurrentNumMon(int):void
getKnight():Hero
getSpellCaster():Hero
getClown():Hero
getPriest():Hero
update(List<Integer>,GamePlayScreen):void
getGameResult():List<Integer>
setGameResult(List<Integer>):void
getnotesImages():ArrayList<Image>
getHeroes():ArrayList<Hero>
setIsGameFinished(boolean):void
isGameFinished():boolean
<<Java Class>>
RenderableHolder
sharedObject
instance: RenderableHolder
iRenderable: List<IRenderable>
RenderableHolder()
add(IRenderable):void
clear():void
getInstance():RenderableHolder
getiRenderable():List<IRenderable>
<<Java Class>>
SkillUpdater
gameLogic
skills: ArrayList<Thread>
stillActivate: boolean
SkillUpdater()
run():void
activateSkills():void
getSkills():ArrayList<Thread>
<<Java Class>>
GamePlayScreen
scene
TITLE_FONT: Font
BTN_FONT: Font
MENU_FONT: Font
COMBO_FONT: Font
SKILL_ACTIVATED: Image
BOSS_SKILL_ACTIVATED: Image
FAIL: Image
LEVEL_UP: Image
width: int
height: int
singlepulse: boolean
isCreate: boolean
bg: Canvas
monsInfo: Canvas
monCanvas: Canvas
heroInfo: Canvas
heroCanvas: Canvas
eventCanvas: Canvas
skillCanvas: Canvas
bossSkillCanvas: Canvas
comboCanvas: Canvas
gamePlay: Canvas
exitMenu: Canvas
yesBtn: Canvas
noBtn: Canvas
playzoneImg: Image
tapZone: Image
ivPlayzone: ImageView
ivTapZone: ImageView
musicControl: MusicControl
instance: GamePlayScreen
gcBg: GraphicsContext
gcMon: GraphicsContext
gcHero: GraphicsContext
GamePlayScreen()
getGcBg():GraphicsContext
getGcHero():GraphicsContext
getGcMon():GraphicsContext
paint():void
changeHero(Hero):void
getMonCanvas():Canvas
getHeroCanvas():Canvas
setImage():void
setIv():void
draw(String,double,double,double,double):Canvas
start():void
addCanvasEvents(Canvas,String):void
setMonsInfo():void
setHeroInfo():void
getCombo():Canvas
updateCombo():void
getIsCreated():boolean
showSkillActivated():void
showBossSkillActivated():void
showLevelUP():void
showFail():void
isCreated(boolean):void
getMusicControl():MusicControl
<<Java Class>>
MusicChart
gameLogic
chart: ArrayList<Note>
bpm: Double
notesPerBar: Integer
hitPerSecond: Double
delayPerHit: Double
duration: Double
currentNoteIdx: int
NANO: double
MusicChart(Double,int,double)
getChart():ArrayList<Note>
setChart(ArrayList<Note>):void
getBpm():Double
setBpm(Double):void
getNotesPerBar():Integer
setNotesPerBar(Integer):void
getHitPerSecond():Double
setHitPerSecond(Double):void
getDelayPerHit():Double
setDelayPerHit(Double):void
getCurrentNoteIdx():int
setCurrentNoteIdx(int):void
generateChart():void
<<Java Class>>
MainMenuScreen
scene
MODE_FONT: Font
CHA_INFO_FONT: Font
CHA_NAME_FONT: Font
knightCanvas: Canvas
spellCasterCanvas: Canvas
clownCanvas: Canvas
priestCanvas: Canvas
farm: Canvas
boss: Canvas
opacityChaCanvas: Canvas
opacityFarmCanvas: Canvas
opacityBossCanvas: Canvas
width: int
height: int
bg: Image
img: Image
ivBg: ImageView
MainMenuScreen()
draw(String,double,double,int,int):Canvas
drawHoverIndicator(Canvas,String):void
undrawHoverIndicator(Canvas,String):void
addCanvasEvents(Canvas,String):void
<<Java Class>>
Note
gameLogic
type: Integer
direction: KeyCode
startTime: Double
canvas: Canvas
image: Image
Note(Integer,KeyCode)
getStartTime():Double
setStartTime(Double):void
getCanvas():Canvas
getDirection():KeyCode
getType():Integer
setDirection(KeyCode):void
<<Java Class>>
JudgeStyle
model
image: Image
JudgeStyle(int)
show():void
<<Java Class>>
ResultScreen
scene
TITLE_FONT: Font
BTN_FONT: Font
STATUS_FONT: Font
title: Canvas
criticalPerfect: Canvas
perfect: Canvas
great: Canvas
good: Canvas
miss: Canvas
nextBtn: Canvas
width: int
height: int
bg: Image
ivBg: ImageView
ResultScreen()
draw(String,double,double,double,double):Canvas
drawHoverIndicator(Canvas,String):void
undrawHoverIndicator(Canvas,String):void
addCanvasEvents(Canvas,String):void
<<Java Class>>
ThreadHolder
sharedObject
instance: ThreadHolder
threads: ArrayList<Thread>
ThreadHolder()
getThreads():ArrayList<Thread>
<<Java Class>>
SceneManager
scene
primaryStage: Stage
welcomeCanvas: Canvas
currentPane: Pane
welcomeSceen: Scene
fadein: Thread
fadeout: Thread
bgMediaPlayer: MediaPlayer
mainBGM: Media
resultBGM: Media
SCENE_WIDTH: int
SCENE_HEIGHT: int
SceneManager()
initialize(Stage):void
gotoWelcomeScreen():void
gotoSceneOf(Pane):void
getFadeIn():Thread
getBgMediaPlayer():MediaPlayer
<<Java Class>>
Main
application
Main()
start(Stage):void
main(String[]):void
stop():void
<<Java Class>>
ResultScreen2
scene
TITLE_FONT: Font
BTN_FONT: Font
STATUS_FONT: Font
SCORE_FONT: Font
title: Canvas
before: Canvas
after: Canvas
nextBtn: Canvas
backBtn: Canvas
opacityBefore: Canvas
opacityAfter: Canvas
width: int
height: int
bg: Image
ivBg: ImageView
ResultScreen2()
draw(String,double,double,double,double):Canvas
drawHoverIndicator(Canvas,String):void
undrawHoverIndicator(Canvas,String):void
addCanvasEvents(Canvas,String):void


<<Java Class>>
MusicControl
gameLogic
BPM_LIST: ArrayList<Double>
NANO: double
firstTypeNotes: ArrayList<Note>
judgeResult: ArrayList<Integer>
guaranteePerfect: boolean
currentCombo: int
toRenderIdx: int
startTime: double
musicChart: MusicChart
file: Media
mediaPlayer: MediaPlayer
toRender: ArrayList<Note>
currentNote: Note
currentNoteIdx: int
offsetCheck: int
judges: int
damageUpdater: DamageUpdater
isComboBreak: boolean
skillUpdater: SkillUpdater
random: Random
musicNumber: int
duration: double
MusicControl()
handle(long):void
run():void
judge(KeyEvent):void
end():void
resetJudgeResult():void
getJudgeResult():ArrayList<Integer>
getMusicChart():MusicChart
getMediaPlayer():MediaPlayer
getCurrentCombo():int
getfirstTypeNotes():ArrayList<Note>
gettoRenderIdx():int
setIsGuarantee(boolean):void
setDuration(double):void
setMusicChart():void
randomMusic():void
generateBPMList():void
<<Java Class>>
CharacterSelectScreen
scene
TITLE_FONT: Font
BTN_FONT: Font
MENU_FONT_BOLD: Font
NULL: Hero
title: Canvas
knightCanvas: Canvas
spellCasterCanvas: Canvas
clownCanvas: Canvas
priestCanvas: Canvas
backBtn: Canvas
startBtn: Canvas
currentCharName: String
width: int
height: int
opacityKnightCanvas: Canvas
opacitySpellCasterCanvas: Canvas
opacityClownCanvas: Canvas
opacitypriestCanvas: Canvas
bg: Image
img: Image
ivBg: ImageView
CharacterSelectScreen()
draw(String,double,double,double,double):Canvas
drawHoverIndicator(Canvas,String):void
undrawHoverIndicator(Canvas,String):void
addCanvasEvents(Canvas,String):void
<<Java Class>>
DamageUpdater
gameLogic
judgeResult: ArrayList<Integer>
DamageUpdater()
run():void
<<Java Class>>
WelcomeScreen
scene
TITLE_FONT: Font
MENU_FONT: Font
bg: Image
gc: GraphicsContext
WelcomeScreen()
addKeyEventHandler():void
keyPressed(KeyEvent):void
pleasePressEnter(String):void
<<Java Class>>
Entity
model
name: String
level: int
isVisible: boolean
Entity(String,int)
getName():String
setName(String):void
getLevel():int
setLevel(int):void
levelUp():void
isVisible():boolean
setVisible(boolean):void
<<Java Class>>
Field
model
isVisible: boolean
bg: Image
Field()
setBg():void
draw(GraphicsContext,double,double):void
isVisible():boolean
setVisible(boolean):void
<<Java Interface>>
IRenderable
sharedObject
draw(GraphicsContext,double,double):void
isVisible():boolean
<<Java Class>>
Monster
model
currentMaxHp: double
currentHp: double
currentExp: double
maxHp: List<Double>
Exp: List<Integer>
MULTIPLE_HP: List<Double>
MULTIPLE_EXP: int
GROWTH_RATE_HP: double
GROWTH_RATE_EXP: double
NUMMONSTER: int
monImg1: Image
monImg2: Image
monImg3: Image
bossImg: Image
hurtedmonImg1: Image
hurtedmonImg2: Image
hurtedmonImg3: Image
hurtedbossImg: Image
alreadyDead: boolean
Monster(String,int)
setmaxHp():void
setMultipleHp():void
setExp():void
getMultipleHp():List<Double>
getMultipleExp():int
getCurrentMaxHp():double
setCurrentMaxHp(int):void
getCurrentHp():Double
setCurrentHp(Double):void
getMaxHp():List<Double>
getExp():List<Integer>
isDead():boolean
isUpgrade():boolean
update(double,GamePlayScreen):void
newMonster():void
upgradeMonster():void
levelUp():void
decreaseHp(double):void
draw(GraphicsContext,double,double):void
getName():String
setAlreadyDead(Boolean):void
<<Java Class>>
Hero
model
skillName: String
currentExp: int
currentMaxExp: int
MAX_EXP: List<Integer>
atk: double
originalAtk: double
GROWTH_RATE_ATK: double
isSkillActivated: boolean
random: Random
Hero(String,int,String)
setMaxExp():void
getSkillName():String
setSkillName(String):void
getCurrentExp():int
setCurrentExp(int):void
getCurrentMaxExp():int
setCurrentMaxExp(int):void
getMaxExp():List<Integer>
getAtk():double
setAtk(double):void
levelUp():void
activate():void
deactivate():void
increaseExp(double):void
isLevelUp():boolean
update(double):void
getIsSkillActivated():boolean
setIsSkillActive(boolean):void
<<Java Interface>>
Skillable
model
activate():void
deactivate():void
<<Java Class>>
Boss
model
skillName: String
maxBossHp: List<Integer>
BossExp: List<Integer>
lastAtk: double
lastHero: Hero
isSkillActivated: boolean
isSilence: boolean
random: Random
Boss(String,int,String)
setmaxBossHp():void
setBossExp():void
getSkillName():String
setSkillName(String):void
draw(GraphicsContext,double,double):void
getName():String
activate():void
deactivate():void
update(double,GamePlayScreen):void
setSilence(boolean):void
getIsSilence():boolean
getisActivated():boolean
<<Java Class>>
Priest
model
heroImg: Image
bg: Image
skill: Thread
Priest(String,int,String)
draw(GraphicsContext,double,double):void
activate():void
deactivate():void
<<Java Class>>
Clown
model
heroImg: Image
bg: Image
direction: ArrayList<KeyCode>
skill: Thread
Clown(String,int,String)
draw(GraphicsContext,double,double):void
activate():void
deactivate():void
<<Java Class>>
SpellCaster
model
heroImg: Image
bg: Image
skill: Thread
SpellCaster(String,int,String)
draw(GraphicsContext,double,double):void
activate():void
deactivate():void
<<Java Class>>
Knight
model
heroImg: Image
skill: Thread
Knight(String,int,String)
draw(GraphicsContext,double,double):void
activate():void
deactivate():void

Implementation Detail

Package application

1. Class Main extends Application

1.1 Method

+ void start(Stage primaryStage)	Initialize ThreadHolder, primaryStage, set title
+ void main(String[] args)	An entry of an application
+ void stop()	This method are going to execute before JavaFX application terminates. In this program, we use it to stop the remaining threads before exit the program.

Package exception

1. Class EnterToGameException extends Exception

1.1 Method

+ String getMessage()	Override method that return "Please Press [Enter] to enter a game."

Package scene

1. Class SceneManager

1.1 Field

- Stage primaryStage	JavaFX container
- Canvas welcomeCanvas	A welcome canvas
- Pane currentPane	A current Pane
- Scene welcomeSceen	A welcome screen
- Thread fadein	A thread for FadeIn effect
- Thread fadeout	A thread for FadeOut effect

- MediaPlayer bgMediaPlayer	A mediaplayer for background music in each scene
- Media mainBGM	A media (music) for MainMenuScreen
- Media resultBGM	A media (music) for ResultScreen
- int SCENE_WIDTH	Width of the application screen
- int SCENE_HEIGHT	Height of the application screen

1.2 Method

+ void initialize(Stage stage) throws Exception	Initialize mainBGM, resultBGM with selected songs and Initialize bgMediaPlayer with mainBGM. Initialize primaryStage.
+ void gotoWelcomeScreen()	Set the primaryStage's scene to welcomeScreen
+ void gotoSceneOf(Pane pane)	Set the pane to scene , stop bgMediaPlayer and create new bgMediaPlayer by the BGM that related to current scene.
+ Getter method for fadein ,bgMediaPlayer	

2. Class WelcomeScreen extends Canvas

2.1 Field

- Font TITLE_FONT	Font style and size of the title text in the WelcomeScreen
- Font MENU_FONT	Font style and size of the enter text in the WelcomeScreen
- Image bg	Image of the background
- GraphicsContext gc	GraphicsContext of WelcomeScreen canvas

2.2 Constructor

+ WelcomeScreen()	- Draw background as shown in figure 2 - Fill game name as shown in figure 2 - Fill game enter text as shown in figure 2

2.3 Method

- void addKeyEventHandler()	Add KeyEvent handler to this canvas by passing KeyEvent to keyPressed if catch EnterToGameException call pleasePressEnter method
- void keyPressed(KeyEvent e) throws EnterToGameException	- When the player press ENTER, use Gamemanager to start the game - When the player press ESC, close the windows - If the player press other than ENTER or ESC throw EnterToGameException
- void pleasePressEnter(String s)	Draw text with given string s

3. Class MainMenuScreen extends Pane

3.1 Field

- Font MODE_FONT	Font style and size of the mode text in the MainMenuScreen
- Font CHA_INFO_FONT	Font style and size of the character information in the MainMenuScreen
- Font CHA_NAME_FONT	Font style and size of the character name in the MainMenuScreen
- Canvas knightCanvas	A knight information canvas
- Canvas spellCasterCanvas	A spellCaster information canvas
- Canvas clownCanvas	A clown information canvas
- Canvas priestCanvas	A priest information canvas

- Canvas farm	A farm mode canvas
- Canvas boss	A boss mode canvas
- int width	Width of the application screen
- int height	Height of the application screen
- Image bg	Image of the background
- ImageView img	Image for the hero
- ImageView ivBg	Imageview of the background
- Canvas OpacityChaCanvas	An opacity canvas for character canvas
- Canvas OpacityFarmCanvas	An opacity canvas for farm mode canvas
- Canvas OpacityBossCanvas	An opacity canvas for boss mode canvas

3.2 Constructor

+ MainMenuScreen()	- Use ImageView to set the background of the pane - Draw all canvas in the position as shown in figure 3 - Add canvas event for farm mode canvas and boss mode canvas - Add all canvas and imageview to the pane

3.3 Method

- Canvas draw(String name, double width, double height, int posX, int posY)	● Create a canvas and put it in the posX and posY translation ● If it's opacityFarmCanvas or opacityBossCanvas, draw a canvas as shown in figure 4 ○ Color is desaturated CORAL ○ Round rectangle with 50 pixels arc and given width and height ○ The opacity is 0.5 ● If it's opacityChaCanvas, draw a canvas as shown

	in figure 5 ○ Color is BISQUE ○ Round rectangle with the given width and height ● If it's farm canvas or boss canvas, draw a canvas as shown in figure 4 ○ Stroke's color is black ○ Stroke's round rectangle with 50 pixels arc and given width and height ○ Stroke line width is 5 pixels ○ Set font to MODE_FONT and fill text with the given name in the center of the canvas ● Else, it's character canvas ○ set image to be the correct character (compared with the given name) and draw it as shown in figure 5 ○ Stroke's color is black ○ Stroke line width is 7 pixels ○ set character name font to be CHA_NAME_FONT ○ fill the given name with given width and height and place it in the right position ○ set character information font to CHA_INFO_FONT ○ fill the character information (level, attack and exp) (Get data from Class GamaManager)
+ public void drawHoverIndicator(Canvas canvas, String name)	Draw hover the given canvas ● Color is CORAL ● Fill a round rectangle with 50 pixels arc and given width and height
+ public void undrawHoverIndicator(Canvas canvas, String name)	Undraw hover the given canvas ● Color is desaturated CORAL ● Fill a round rectangle with 50 pixels arc and given width and height

- private void addCanvasEvents(Canvas canvas, String canvasName)	Add MouseEvent handler to give canvas ● When the player click the canvas ○ If it's farm canvas, set current mode to Farm and go to CharacterSelectScreen ○ If it's boss canvas, set current mode to boss, go and start GamePlayScreen ● When the player move mouse enter the canvas, draw hover that canvas ● When the player move mouse exit the canvas, undraw hover that canvas

4. Class CharacterSelectScreen extends Pane

4.1 Field

- Font TITLE_FONT	Font style and size of the title
- Font BTN_FONT	Font style and size of the button
- Font CHA_NAME_FONT	Font style and size of the character name
- Hero NULL	A dummy Hero
- Canvas title	A canvas for "Select Character" text
- Canvas knightCanvas	A knight canvas
- Canvas spellCasterCanvas	A spellCaster canvas
- Canvas clownCanvas	A clown canvas
- Canvas priestCanvas	A priest canvas
- Canvas backBtn	A back canvas
- Canvas startBtn	A start canvas
- String currentCharName	A string for current character
- int width	Width of the application screen
- int height	Height of the application screen

- Canvas opacityKnightCanvas	An opacity canvas for knight canvas
- Canvas opacitySpellCasterCanvas	An opacity canvas for spell caster canvas
- Canvas opacityClownCanvas	An opacity canvas for clown canvas
- Canvas opacityPriestCanvas	An opacity canvas for priest canvas
- Image bg	Image of the background
- Image img	Image of the hero
- ImageView ivBg	Imageview of the background

4.2 Constructor

+ CharacterSelectScreen()	● Initialize Image and ImageView ● Use ImageView to set the background of the pane with the width and height ● Draw all canvas in the position as shown in figure 6 ● Add canvas event for all character canvas, back canvas and start canvas ● Add all canvas and imageview to the pane

4.3 Method

- Canvas draw(String name, double width, double height, int posX, int posY)	● Create a canvas and put it in the posX and posY translation ● If it's title canvas, draw a canvas as shown in figure 7 ○ Font's color is BLACK ○ Set font to TITLE_FONT ○ Fill the text in the center of the canvas with given width and height ● If it's character canvas, draw a canvas as shown in figure 8

	○ Set image to be the correct character (compared with the given name) and draw it as shown in figure 8 ○ Stroke's color is darker TAN ○ Stroke line width is 10 pixels ○ Fill the round rectangle with arc 50 pixels and the given width and height ○ Text's color is BLACK ○ Set font to CHA_NAME_FONT ○ Set character name font to be CHA_NAME_FONT ○ Fill the given name in the center of the canvas ● Else, it's opacity character canvas, draw a canvas as shown in figure 8 ○ Color is WHITE ○ Fill the round rectangle with arc 50 pixels and the given width and height ○ Opacity is 0.8 ● Else, it's start canvas or exit canvas, draw a canvas as shown in figure 9 ○ Color of the round rectangle is LIGHTBLUE ○ Fill the round rectangle with arc 50 pixels and the given width and height ○ Color of the font is BLACK ○ Set font to be BTN_FONT ○ Fill the given name in the center of the canvas with given width and height
+ void drawHoverIndicator(Canvas canvas, String name)	Draw hover the given canvas ● If it's start canvas or back canvas ○ Color is desaturated LIGHTBLUE ○ Fill a round rectangle with 50 pixels arc and given width and height ○ Font's color is BLACK ○ Set font to BTN_FONT ○ Fill the given name in the center of the canvas with given width and height

	● Else, it's opacity character canvas ○ Color is desaturated LIGHTCORAL ○ Fill the round rectangle with arc 50 pixels and the given width and height ○ Opacity is 0.9
+ void undrawHoverIndicator(Canvas canvas, String name)	Undraw hover the given canvas ● If it's start canvas or back canvas ○ Color is LIGHTBLUE ○ Fill a round rectangle with 50 pixels arc and given width and height ○ Font's color is BLACK ○ Set font to BTN_FONT ○ Fill the given name in the center of the canvas with given width and height ● Else, it's opacity character canvas ○ Color is WHITE ○ Fill the round rectangle with arc 50 pixels and the given width and height ○ Opacity is 0.9
- void addCanvasEvents(Canvas canvas, String canvasName)	Add MouseEvent handler to give canvas ● When the player click the canvas ○ If it's character canvas, set currentCha (Compared by given canvasName), set currentChaname and undraw other character canvas ○ If it's start canvas, create a gamePlayScreen and go to gamePlayscreen by using FadeIn effect from SceneManager then start gamePlayScreen ○ If it's back canvas, create a mainMenuScreen and go to mainMenuScreen ● When the player move mouse enter the canvas, draw hover that canvas ● When the player move mouse exit the canvas and

	the given name is not equal to currentChaname, undraw hover that canvas

5. Class GamePlayScreen Extends Pane

5.1 Field

- Font TITLE_FONT	Font style and size of the title
- Font BTN_FONT	Font style and size of the yes/no button
- Font MENU_FONT	Font style and size of the monster' information, hero's information and exit menu
- Font COMBO_FONT	Font style and size for Combo Canvas.
- Image SKILL_ACTIVATED	Image to show when Hero's skill is activated.
- Image BOSS_SKILL_ACTIVATED	Image to show when Boss's skill is activated.
- Image FAIL	Image to show when you cannot defeat the Boss.
- Image LEVEL_UP	Image to show when Hero leveled up.
- int width	Width of the application screen
- int height	Height of the application screen
- boolean singlepulse	A boolean variable to check if it's singlepulse press or not
- boolean isCreate	A boolean variable to check if GamePlayScreen is already created
- Canvas bg	A background canvas
- Canvas monsInfo	A monster information canvas
- Canvas monsCanvas	A monster canvas
- Canvas heroInfo	A hero information canvas
- Canvas heroCanvas	A hero canvas
- Canvas eventCanvas	A canvas for events those are levelup and defeat

- Canvas skillCanvas	A canvas for SKILL_ACTIVATED image
- Canvas bossCanvas	A canvas for BOSS_SKILL_ACTIVATED image
- Canvas comboCanvas	A canvas for combo count
- Canvas gamePlay	A game play canvas
- Canvas exitMenu	An exit menu
- Canvas yesBtn	A yes button canvas
- Canvas noBtn	A no button canvas
- Image playzoneImg	An Image of the play zone
- Image tapZone	An Image of tap zone
- ImageView ivPlayzone	An ImageView of the play zone
- ImageView ivTapZone	An ImageView of the tap zone
- MusicControl musicControl	A MusicControl object for each game
- GamePlayScreen instance	An singleton instance of this class
- GraphicsContext gcBg	GraphicsContext for background
- GraphicsContext gcHero	GraphicsContext for hero
- GraphicsContext gcMon	GraphicsContext for monster

5.2 Constructor

- GamePlayScreen()	● Initialize event and set their translation to be like as shown in figure 10 ● Initialize gcBg, gcHero, gcMon, instance, isCreate and bg, heroCanvas, monCanvas, eventCanvas, skillCanvas, bossSkillCanvas ● Set isCreate to be true ● Paint heroCanvas and monCanvas ● Draw monsInfo, heroInfoand gamePlay ● AddCanvasEvents for gamePlay

	● Set FocusTraversable of gamePlay to be true ● Set all Images and ImageViews ● Clear whatever that has been in the pane ● Add bg, all canvas (except combo canvas) and all ImageView to the pane ● Translate skillCanvas to (90,40) ● Translate bossSkillCanvas to (500,60) ● Translate eventCanvas to center of screen (above playzone)

5.3 Method

+ void paint()	● Clear gcHero ● For each Irenderable, if it's visible ○ If it's a field, set background and draw background using gcBg ○ If it's a hero, draw a hero using gcHero in a position shown in figure 12 ○ If it's a monster, draw a monster using gcMon in a position shown in figure 13 ○ If it's a boss, draw a boss using gcMon in a position shown in figure 16
+ synchronized void changeHero(Hero back)	Play the effect that turn current character to the other that its skill is activated (Using ScaleTransition to swap between current heroCanvas and newly created one) ● Create a canvas for back character (activated one) ● Initialize two ScalTrasition for hide Front and show Back ● Set duration of each to 300 milliseconds ● Set FromX and ToX of ScaleTransition that hide front to 1 and 0 respectively ● Set back canvas ScaleX to 0 ● Set FromX and ToX of ScaleTransition that show back to 0 and 1 respectively ● Add onFinishedEvent to hide front to play show back transition

	● Add onFinishedEvent to show back transition to change the currently heroCanvas to a newly created one ● Add back canvas to GamePlayScreen Pane ● Play hide front transition
+ void setImage()	Set all Images
+ void setIv()	Set all ImageViews
+ Canvas draw(String name, double width, double height, double posX, double posY)	Draw a canvas using given width, height, posX, posY to be as shown in figure 10 ● Initialize a canvas, set the translation to be given posX, posY ● If the given name is “MonsInfo” ○ Draw a rectangle with the given width and height ○ Rectangle’s color is PALEVIOLETRED ○ Text’s font is MENU_FONT ○ Text-Baseline is center ○ Text-Alignment is left ○ Text’s color is WHITE ○ Fill the monster’s information (name, level, hp) ● Else if the given name is “HeroInfo” ○ Draw a rectangle with the given width and height ○ Rectangle’s color is BURLYWOOD ○ Text’s font is MENU_FONT ○ Text-Baseline is center ○ Text-Alignment is left ○ Text’s color is WHITE ○ Fill the monster’s information (name, level, atk, hp) ● Return the canvas
+ void start()	Initialize musicControl and make it start when it’s ready
- void addCanvasEvents(Canvas	Add KeyEvent handler to give canvas

canvas, String name)	● When the player pressed an ArrowKey and singlepulse is false, judge the timing of pressied time ● When the player pressed an ESCAPE key, show Exit Menu ● When the player released an ArrowKey, set singlepulse to be false ● When the player click the canvas ○ Create and initialize a main menu pane ○ If the given name is “No”, set visible of exitMenu, yesBtn, noBtn to be false ○ Else, the given name is “Yes”, end the MusicControl and go to mainMenuScreen
+ void setMonsInfo()	● Remove monInfo from the pane ● Draw new monInfo ● Add monInfo to the pane
+ void setHeroInfo()	● Remove heroInfo from the pane ● Draw new heroInfo ● Add heroInfo to the pane
+ void updateCombo()	redraw the comboCanvas if current combo is more than 5 ● Clear canvas ● Draw “ X Combo!” where “X” is current combo to position X=200, Y=200 ● Set fill to WHITE ● Set TextAlignment to CENTER ● Set TextBaseLine to VPos.CENTER ● Set Font to COMBO_FONT
+ void showSkillActivated()	Use anonymous Thread to draw SKILL_ACTIVATED image to skillCanvas , sleep for 600 milliseconds then clear canvas
+ void showBossSkillActivated()	Use anonymous Thread to draw BOSS_SKILL_ACTIVATED image to bossSkillCanvas, sleep for 600 milliseconds then clear canvas
+ void showLevelUP()	Use anonymous Thread to draw LEVEL_UP image to

	eventCanvas , sleep for 1000 milliseconds then clear canvas
+ void showFail()	Use anonymous Thread to draw FAIL image to eventCanvas , sleep for 1000 milliseconds then clear canvas
+ void IsCreated(boolean x)	Set isCreate to be the given boolean
+ Getter method for gcBg, gcHero, gcMon, heroCanvas, combo and instance	
+ Getter method for monCanvas, isCreate and musicControl	

6. Class ResultScreen extends Pane

6.1 Field

- Font TITLE_FONT	Font style and size of the title
- Font BTN_FONT	Font style and size of the next button
- Font STATUS_FONT	Font style and size of the sign and how many times the player got the sign
- Canvas title	A title canvas
- Canvas criticalPerfect	A criticalPerfect sign canvas
- Canvas perfect	A perfect sign canvas
- Canvas great	A great sign canvas
- Canvas good	A good sign canvas
- Canvas miss	A miss sign canvas
- Canvas nextBtn	A next button canvas
- int width	Width of the application screen

- int height	Height of the application screen
- Image bg	Image of the background
- ImageView ivBg	Imageview of the background

6.2 Constructor

+ ResultScreen()	● Initialize Image and ImageView ● Use ImageView to set the background of the pane with the width and height ● Draw all canvas ● AddCanvasEvents for all canvas ● Add ivBg and all canvas to the pane

6.3 Method

+ Canvas draw(String name, double width, double height, double posX, double posY)	Draw a canvas using given width, height, posX, posY to be as shown in figure 11 ● Initialize a canvas, set the translation to be given posX, posY ● If the given name is “Result” ○ Text’s font is TITLE_FONT ○ Text-Baseline is center ○ Text-Alignment is center ○ Text’s color is WHITE ○ Fill the given name on the center of the canvas ● If the given name is a sign ○ Draw a round rectangle with the stroke and given width and height ○ The opacity is 0.9 ○ Stroke’s color isr TAN ○ Stroke line width is 5 pixels ○ Fill the stroke round rectangle with arc 50 pixels and the given width and height ○ Rectangle’s color is ALICEBLUE

	○ Fill the round rectangle with arc 50 pixels and the given width and height ○ Text's color is BLACK ○ Set font to STATUS_FONT ○ Text-Baseline is center ○ Fiill the given name and how many times you got that sign (Get data from gameResult in Class GamaManager) ● Else, it's next button canvas ○ Draw a round rectangle with the given width and height ○ Round rectangle's color is TAN ○ Text's font is BTN_FONT ○ Text-Baseline is center ○ Text-Alignment is center ○ Text's color is BLACK ○ Fill the given name on the center of the canvas ● Return the canvas
+ void drawHoverIndicator(Canvas canvas, String name)	Draw hover the given canvas if the given name is "Next" ● Draw a round rectangle with 50 pixels arc and the given width and height ● Round rectangle's color is brighter TAN ● Text's font is BTN_FONT ● Text-Baseline is center ● Text-Alignment is center ● Text's color is BLACK ● Fill the given name on the center of the canvas
+ void undrawHoverIndicator(Canvas canvas, String name)	Undraw hover the given canvas if the given name is "Next" ● Draw a round rectangle with 50 pixels arc and the given width and height ● Round rectangle's color is TAN ● Text's font is BTN_FONT ● Text-Baseline is center ● Text-Alignment is center ● Text's color is BLACK

	● Dill the given name on the center of the canvas
- void addCanvasEvents(Canvas canvas, String name)	Add MouseEvent handler to give canvas ● When the player click the canvas ○ Create and initialize the result2 pane and mainMenuScreen pane ○ If the given name is “Next” and the current mode is “Boss”, go to mainMenuScreen ○ If the given name is “Next” and the current mode is not “Boss”, go to resultScreen2 ● When the player move mouse enter the canvas, draw hover that canvas ● When the player move mouse exit the canvas, undraw hover that canvas

7. Class ResultScreen2 extends Pane

7.1 Field

- Font TITLE_FONT	Font style and size of the title
- Font BTN_FONT	Font style and size of the back button and next button
- Font STATUS_FONT	Font style and size of the hero’s old status and current status
- Font SCORE_FONT	Font style and size of scores
- Canvas title	A title canvas
- Canvas before	An old status canvas
- Canvas after	A current status canvas
- Canvas nextBtn	A next button canvas
- Canvas backBtn	A back button canvas
- Canvas boardBefore	An opacity canvas for old status canvas
- Canvas boardAfter	An opacity canvas for current status canvas

- int width	Width of the application screen
- int height	Height of the application screen
- Image bg	Image of the background
- ImageView ivBg	Imageview of the background

7.2 Constructor

+ ResultScreen2()	● Initialize Image and ImageView ● Use ImageView to set the background of the pane with the width and height ● Draw all canvas in the position as shown in figure 12 ● Add canvas event for all canvas ● Add all canvas and imageview to the pane

7.3 Method

- Canvas drawButton(String name, double width, double height, int posX, int posY)	● Create a canvas and put it in the posX and posY translation ● If the given name is “Well Done” ○ Text’s font is TITLE_FONT ○ Text-Baseline is center ○ Text-Alignment is left ○ Text’s color is WHITE ○ Fill the given name on the center of the canvas ● If the given name is “Before” ○ Draw a round rectangle with the stroke and given width and height ○ Stroke’s color isr TAN ○ Stroke line width is 10 pixels ○ Fill the stroke round rectangle with arc 50 pixels and the given width and height ○ Text’s color is WHITE ○ Set status font to STATUS_FONT

	○ Text-Baseline is center ○ Fill the old status of currentHero (name, level, attack, exp) (Get data from statusBefore in Class GameManager) ○ Draw an exp bar ■ Fill the stroke rectangle in white color ■ Rectangle's color is SANDYBROWN ■ Rectangle's width is exp before divide max exp before ■ Text's font is SCORE_FONT ■ Text-Baseline is center ■ Text-Alignment is center ■ Text's color is BLACK ■ Fill the exp before and max exp before seperated by "/" ● Else If the given name is "After" ○ Draw a round rectangle with the stroke and given width and height ○ Stroke's color isr TAN ○ Stroke line width is 10 pixels ○ Fill the stroke round rectangle with arc 50 pixels and the given width and height ○ Text's color is WHITE ○ Set status font to STATUS_FONT ○ Text-Baseline is center ○ Fill the current status of currentHero (name, level, attack, exp) (Get data from statusAfter in Class GameManager) ○ Draw an exp bar ■ Fill the stroke rectangle in white color ■ Rectangle's color is SANDYBROWN ■ Rectangle's width is exp after devide max exp after ■ Text's font is SCORE_FONT ■ Text-Baseline is center ■ Text-Alignment is center

	■ Text's color is BLACK ■ Fill the exp after and max exp after separated by "/" ● Else If the given name is "opacityBefore" or "opacityAfter" ○ The opacity is 0.5 ○ Draw a round rectangle with 5 pixels arc and the given width and height ○ Round rectangle's color is BLACK ● Else, it's button canvas ○ Draw a round rectangle with the given width and height ○ Round rectangle's color is TAN ○ Text's font is BTN_FONT ○ Text-Baseline is center ○ Text-Alignment is center ○ Text's color is BLACK ○ Fill the given name on the center of the canvas ● Return the canvas
+ void drawHoverIndicator(Canvas canvas, String name)	Draw hover the given canvas if the given name is "Next" or "Back" ● Draw a round rectangle with 50 pixels arc and the given width and height ● Round rectangle's color is brighter TAN ● Text's font is BTN_FONT ● Text-Baseline is center ● Text-Alignment is center ● Text's color is BLACK ● Fill the given name on the center of the canvas
+ void undrawHoverIndicator(Canvas canvas, String name)	Undraw hover the given canvas if the given name is "Next" or "Back" ● Draw a round rectangle with 50 pixels arc and the given width and height ● Round rectangle's color is TAN ● Text's font is BTN_FONT

	● Text-Baseline is center ● Text-Alignment is center ● Text's color is BLACK ● Fill the given name on the center of the canvas
- void addCanvasEvents(Canvas canvas, String canvasName)	Add MouseEvent handler to give canvas ● When the player click the canvas ○ Create and initialize the resultScreen pane and mainMenuScreen pane ○ If the given name is "Next", go to mainMenuScreen and UpdateStatusBefore (use Class GameManager) ○ Else if the given name is "Back" go to resultScreen ● When the player move mouse enter the canvas, draw hover that canvas ● When the player move mouse exit the canvas, undraw hover that canvas

Package model

1. Class *Entity* implements IRenderable

1.1 Field

# String name	Entity's name
# int level	Entity's level
# boolean isVisible	Entity's visibility

1.2 Constructor

+ Entity(String name, int level)	Set name and level to be given name and level

1.3 Method

+ void *levelUp()*	
+ Getter and setter method for name and level	

2. Class *Hero* extends Entity implements skillable

2.1 Field

# String skillName	Name of skill
# int currentExp	Current exp
# int currentMaxExp	Current max exp
# List<Integer> MAX_EXP	List of max exp
# double atk	Attack
# double originalAtk	Original attack
# double GROWTH_RATE_ATK	Growth rate attack
# boolean isSkillActivated	The flag indicate that skill is activate or not
# Random random	A random object

2.2 Constructor

+ Hero (String name, int level, String skillName)	• Set name, level and skill name • Set maxExp • Initialize currentExp, currentMaxExp, originalAtk and isSkillActivated

2.3 Method

+ void setMaxExp()	Set maxExp
+ void levelUp()	● Increase level by one ● Increase atk by GROWTHRATEATK ● Set new currentMaxExp ● Set new currentExp ● Set new originalAtk ● Update hero by remaining exp
+ void increaseExp(double exp)	Increase exp by given exp
+ boolean isLevelUp()	● Return true if currentExp is more than currentMaxExp ● Otherwise, return false
+ void update(double exp)	● Increase exp ● If level up ○ Level up ○ showLevelUP (Use Class GamePlayScreen)
+ void *activate()*	
+ void *deactivate()*	
+ Getter and Setter method for skillName, currentExp, currentMaxExp, atk and isSkillActive	
+ Getter method for maxExp	

3. Class Knight extends Hero

3.1 Field

- Image heroImg	Knight's image
- Thread skill	A skill thread

3.2 Constructor

+ Knight(String name, int level, String skillName)	● Set name, level and skillname ● Initialize knight's image

3.3 Method

+ void draw(GraphicsContext gc, double x, double y)	● Clear rectangle ● Draw image in the position x and y
+ void activate()	Create a skill thread that do according to Knight's skill and add it to ThreadHolder and SkillUpdater
+ void deactivate()	Create a skill thread that do according to Knight's skill and add it to ThreadHolder and SkillUpdater

4. Class SpellCaster extends Hero

4.1 Field

- Image heroImg	SpellCaster's image
- Image bg	Background image of SpellCaster
Thread skil	A skill thread

4.2 Constructor

+ SpellCaster(String name, int level, String skillName)	● Set name, level and skillname ● Initialize spellCaster's image ● Initialize background image

4.3 Method

+ void draw(GraphicsContext gc, double x, double y)	● Clear rectangle ● Draw image in the position x and y
+ void activate()	Create a skill thread that do according to SpellCaster's skill

	and add it to ThreadHolder and SkillUpdater
+ void deactivate()	Create a skill thread that do according to SpellCaster's skill and add it to ThreadHolder and SkillUpdater

5. Class Clown extends Hero

5.1 Field

- Image heroImg	Clown's image
- Image bg	Background image of Clown
- ArrayList<KeyCode> direction	An ArrayList of KeyCode that is in range of Clown's skill
- Thread skill	A skill thread

5.2 Constructor

+ Clown(String name, int level, String skillName)	● Set name, level and skillname ● Initialize clown's image ● Initialize background image

5.3 Method

+ void draw(GraphicsContext gc, double x, double y)	● Clear rectangle ● Draw image in the position x and y
+ void activate()	Create a skill thread that do according to Clownr's skill and add it to ThreadHolder and SkillUpdater
+ void deactivate()	Create a skill thread that do according to Clown's skill and add it to ThreadHolder and SkillUpdater

6. Class Priest extends Hero

6.1 Field

- Image heroImg	Priest's image
- Image bg	Background image of Priest
- Thread skill	A skill thread

6.2 Constructor

+ Priest(String name, int level, String skillName)	● Set name, level and skillname ● Initialize priest's image ● Initialize background image

6.3 Method

+ void draw(GraphicsContext gc, double x, double y)	● Clear rectangle ● Draw image in the position x and y
+ void activate()	Create a skill thread that do according to Priest's skill and add it to ThreadHolder and SkillUpdater
+ void deactivate()	Create a skill thread that do according to Priest's skill and add it to ThreadHolder and SkillUpdater

7. Class Monster extends Entity

7.1 Field

# double currentMaxHp	Current max exp
# double currentHp	Current hp
# double currentExp	Current exp
- List<Double> maxHp	List of max hp
- List<Integer> Exp	List of exp

- List<Double> MULTIPLE_HP	List of multiple hp
- int MULTIPLE_EXP	Multiple exp
- double GROWTH_RATE_HP	Growth rate of hp
- double GROWTH_RATE_EXP	Growth rate of exp
- int NUMMONSTER	Number of all monsters
- Image monImg1	Image of monster1
- Image monImg2	Image of monster2
- Image monImg3	Image of monster3
# Image bossImg	Image of boss
- Image hurtedmonImg1	Image of hurted monster1
- Image hurtedmonImg2	Image of hurted monster2
- Image hurtedmonImg3	Image of hurted monster3
# Image hurtedbossImg	Image of hurted boss
# boolean alreadyDead	

7.2 Constructor

+ Monster(String name, int level)	● Set name and level ● Set maxHp ● Set multipleHp ● Set exp ● Initialize currentMaxHp ● Initialize currentHp ● Initialize currentExp ● Set isVisible to false ● Set alreadyDead to false

7.3 Method

+ void setmaxHp()	Set max hp
+ void setMultipleHp()	Set multiple hp
+ void setExp()	Set exp
+ boolean isDead()	If current hp is less than 0, return true. Otherwise, return false
+ boolean isUpgrade()	Return true if level of current character is 5, 9, 12, 15, 17, 19. Otherwise, return false
+ void update(double atk, GamePlayScreen gamePlayScreen)	● Decrease hp by given atk ● If the monster is dead ○ Update current character by its exp ○ If monster is upgrade, upgrade monster ○ Else ■ If currentNumMon is 3, level up ■ New monster ○ Increase level by one ○ Draw a monster
+ void newMonster()	● If currentNumMon is 3, set it to 0 ● Increase currentNumMon by one ● Set new currentMaxHp ● Set new currentHp ● Set new currentExp
+ void upgradeMonster()	● Increase each value in maxHp list and Exp list ● Set new currentMaxHp ● Set new currentHp ● Set new currentExp ● Set currentNumMon to 1
+ void levelUp()	Increase each value in maxHp list and Exp list by GROWTHRATEHP and GROWTHRATEEXP
+ void decreaseHp(double atk)	Decrease currentHp by given atk

+ void draw(GraphicsContext gc, double x, double y)	● Clear rectangle ● Draw current hurted monster and new monster
+ String getName()	Return monster's name
+ void setAlreadyDead(Boolean x)	Set alreadyDead to be given x

8. Class Boss extends Monster implements Skillable

8.1 Field

- String skillName	Skill name
- List<Integer> MAXBOSSHP	List of max boss hp
- List<Integer> BOSSEXP	List of boss exp
- double lastAtk	Last attack
- Hero lastHero	Last hero
- boolean isSkillActivated	The flag indicate that skill is activated or not
# boolean isSilence	The flag indicate that silence or not
# Random random	Random object of this class

8.2 Constructor

+ Boss (String name, int level, String skillName)	● Set name and level ● Initialize skill name ● Set MAXBOSSHP ● Set BOSSEXP ● Initialize currentMaxHp ● Initialize currentHp ● Set isVisible to true ● Set isSkillActivated to false ● Set isSilence to false ● Set alreadyDead to false

8.3 Method

+ void setmaxBossHp()	Set max boss hp
+ void setBossExp()	Set boss exp
+ void draw(GraphicsContext gc, double x, double y)	Draw boss's image at the position given x and y
+ String getName()	Return "Boss"
+ void activate()	Create a skill thread that do according to Boss's skill and add it to ThreadHolder and SkillUpdater
+ void deactivate()	Create a skill thread that do according to Bosst's skill and add it to ThreadHolder and SkillUpdater
+ void update(double atk, GamePlayScreen gamePlayScreen)	● Decrease hp by given atk ● If boss is dead and alreadyDead is false ○ Update current character by currentExp ○ Set alreadyDead to true ○ Increase level by one ○ Set new currentMaxHp ○ Set new currentHp ○ Set new currentExp ○ End musicControl
+ Getter & Setter method for skillName and isSilence	
+ Getter method for isActivated	

9. Class Field implements IRenderable

9.1 Field

- boolean isVisible	The flag indicate visibility
- Image bg	Background image

9.2 Constructor

+ Field()	Set visible to be true

9.3 Method

+ void setBg()	Initialize image that belongs to current character
+ void draw (GraphicsContext gc, double x, double y)	Draw field's image at the position given x and y with scene's width and scene's height
+ Getter and Setter method for isVisible	

10. Interface Skillable

10.1 Method

+ void activate()	
+ void deactivate()	

11. Class JudgeStyle extends ImageView

11.1 Field

- Image image	An image of judgestyle

11.2 Constructor

+ JudgeStyle(int direction)	Initialize image according to direction

11.3 Method

+ void show()	Create a thread to show an image for 150 milliseconds

Package gameLogic

1. Class DamageUpdater extends Thread

1.1 Field

- ArrayList<Integer> judgeResult	An array that store judge results in each judge

1.2 Method

+ void run()	Update GameManager by current judge results and set new judgeResult in MusicControl in every 1 second in game

2. Class SkillUpdater extends Thread

2.1 Field

- boolean stillActivate	A boolean variable for check if the skillUpdater is curently using
- ArrayList<Thread> skills	An arrayList for skill thread of each skill

2.2 Method

+ void run()	Initialize skills arrayList, call activateSkills and start to add a skill of a hero that currently available to activate every 150 milliseconds
+ void activateSkills()	Create and start Thread that start currently available thread in skills ArrayList in every 1 second

3. Class Note

3.1 Field

- Integer type	An interger varible that indicate Note type those are 0 or 1
- KeyCode direction	A keycode that indicate direction

- Double startTime	A variable that indicate the start time in game that this Note should be rendered
- Canvas canvas	A canvas for Note image
- Image image	A note image

3.2 Constructor

+ Note(Integer type,KeyCode direction)	Initialize note image and canvas according to direction, ● set translate of a canvas to (-100 , 472)

3.3 Method

+ void setDirection(KeyCode e)	Set direction and image of this Note according to KeyCode
+ Getters and Setters method for all fields except image field	

4. Class MusicChart

4.1 Field

- ArrayList<Note> chart	An ArrayList that store every Notes in each game
- Double bpm	A BPM of current song
- Integer notesPerBar	A number of notes in a bar
- Double hitPerSecond	A number of hit in a second
- Double delayPerHit	A delay per hit
- Double duration	A duration of current song
- Integer currentNoteIdx	A integer that locate current note index in an ArrayList
+ Double NANO	A nano value 10^9

4.2 Constructor

+ MusicChart(Double bpm, int notesperbar, double duration)	Initialize field according to its properties and generate chart.

4.3 Method

+ void generateChart()	Generate ArrayList of Note by random. ● Random type of note ● If it's type 1 then random the direction of note ● The number of total Note generated is a number of duration divided by delayPerHit
+ Getters and Setter for all Fields	

5. Class MusicControl extends AnimationTimer

5.1 Field

- ArrayList<Double> BPM_LIST	An ArrayList of all songs BPM
+ double NANO	A nano value 10^9
- ArrayList<Note> firstTypeNotes	An ArrayList that store type 1 Notes of current song
- ArrayList<Integer> judgeResult	An ArrayList that store current game judge results
- Integer currentCombo	A variable that store current combos
- Integer toRenderIdx	A variable that indicate next render Note in an ArrayList
- boolean guaranteePerfect	A variable that indicate if it's currently guarantee perfect judge from SpellCaster skill
- ArrayList<Note> toRender	An ArrayList of currently render Notes
- Double startTime	A varible that store start time of a game in nano second
- MusicChart musicChart	A MusicChart of current game
- Media file	A media file of current song

- MediaPlayer mediaPlayer	A mediaplayer of current game
- Note currentNote	A current note
- Integer currentNoteIdx	A current Note index in music chart
- Integer offSetCheck	A variable that indicate if it's first time to render
- Integer judges	A variable that indicate judge result type
- DamageUpdater damageUpdater	A DamageUpdater for this game
- boolean isComboBreak	A variable that indicate if current combo is already break so far
- SkillUpdater skilUpdater	A SkillUpdater for this game
- Random random	A Random object for this game
- Integer musicNumber	A randomed music number
- Double duration	A current song durration in seconds

5.2 Constructor

+ MusicControl()	Initialize every field according to their propoties. Random a music for this game

5.3 Method

+ void run()	Set startTime and start this AnimationTimer
+ void setMusicChart()	Initialize musicChart to randomed music file and add type 1 note to firstTypeNotes ArrayList. Set the start time of each note to judge time minus by 2 seconds to make them move to the tap area in 2 seconds
+ void randomMusic()	Initialize file to a randomed music
+ void generateBPMList	Generate ArrayList of BPM of all song in game

+ void handle(long now)	A handle method of AnimationTimer ● If it's a first loop start mediaPlayer, damageUpdater and skillUpdater ● If current time of game is more than song duration plus 4 seconds call end() method ● Translate all Notes according to current time and start time of each Note ● Show Miss JudgeStyle if a note is pass the judge zone and remove it from ArrayList and update combo
+ void judge(KeyEvent e)	This method use for judge the tapped time and judged time of each Note ● Critical Perfect : +- 0.015 sec. ● Perfect : +- 0.03 sec. ● Great : +- 0.1 sec. ● Good : +- 0.2 sec. ● Show JudgeStyle according to performance ,update combo and remove it from toRender ArrayList
+ void end()	Stop all threads and mediaplayer then go to ResultScreen
+ void resetJudgeResult()	Initialize new judgeResult
+ Getters for judgeResult, musicChart, mediaPlayer, firstTypeNotes and toRenderIdx	
+ Setters for duration and guaranteePerfect	

6. Class GameManager

6.1 Field

- Hero currentCha	Current Hero in each game
- Monster currentMonKnight	A monster for Knight
- Monster currentMonSpell	A monster for SpellCaster
- Monster currentMonPriest	A monster for Priest
- Monster currentMonClown	A monster for Clown
- Monster currentMon	Current monster in each game
- int currentNumMon	A number of monster that hero have killed so far in each game
- int currentBoss	Current boss
- Hero knight	Hero Knight
- Hero spellCaster	Hero spellCaster
- Hero clown	Hero clown
- Hero priest	Hero priest
- Monster boss	Monster boss
- Field field	Field
- String currentMode	Current mode
- List<Integer> gameResult	List of game result
- List<Double> statusBefore	List of status before
- ArrayList<Image> notesImage	List of notes image
- ArrayList<Hero> heroes	List of heroes
- Random random	Random object of this class
- boolean isGameFinished	The flag indicate that the game is finished or not

6.2 Method

+ void newGame()	● Initialize random, knight, spellCaster, clown, priest, heroes, boss, field, currentMonClown, currentMonKnight, currentMonSpell, currentMonPriest, currentMon, gameResult, statusBefore, and statusBefore ● Add knight, spellCaster, clown, priest to heroes ● Set isGameFinished to false ● Set currentCha as knight ● Add knight, spellCaster, clown, priest and field to RenderableHolder ● Set currentBoss to 1 ● Set currentNumMon to 1 ● Go to MainMenuScreen ● Add arrow images to notesImage
+ void UpdateStatusBefore()	● Add level, atk, currentExp and CurrentMaxExp's current character to statusBefore ● Set currentCha to knight ● Initialize gameResult
+ synchronized void setCurrentCha(String hero, boolean isGameStart)	● Set lastHero to currentCha ● For each iRenderable in renderableHolder ○ If it's hero and it's name is not given name, set visible to be false ○ Else if it's hero and it's name is given name, set visible to be true ● Set current character ● If isCreated is true, change hero to currentcha ● If isGameStart is true ○ Set field background ○ Add level, atk, currentExp and CurrentMaxExp's current character to statusBefore
+ void setCurrentMode(String mode)	● set currentMode to given mode ● If current mode is boss ○ Remove currentMon from

	renderableHolder ○ Set currentMon to boss ○ Set CurrentCha by random character ○ Add currentMon to renderableHolder
+ void setCurrentMon()	● Remove currentMon from renderableHolder ● Set currentMon that belongs to current character ● Add currentMon to renderableHolder
+ void update(List<Integer> list, GamePlayScreen gamePlayScreen)	● If current mode is boss ○ Set atk to be the sum of all character's attack ○ Else, set atk to be currentCha's attack ● Use given list of how many of each signs the player get to update current monster and gameResult ● Call setMonsInfo and setHeroInfo
+ Getter and Setter method for StatusBefore, CurrentBoss, currentNumMon, GameResult, IsGameFinished	
+ Getter method for CurrentCha, CurrentMode, CurrentMon, Knight, SpellCaster, Clown, Priest, notesImages, Heroes	

Package sharedObject

1. Interface IRenderable

1.1 Method

+ void draw(GraphicsContext gc, double x, double y);	A method to draw its image to given GraphicsContext at (x,y)
+ boolean isvisible()	A method to return if its visible or not

2. Class RenderableHolder

2.1 Field

- RenderableHolder instance	A singleton for RenderableHolder
- List<IRenderable> iRenderable	A list of all object that implements IRenderable

2.2 Constructor

+ RenderableHolder()	Initialize iRenderable

2.3 Method

+ void add (IRenderable entity)	Add entity to iRenderable
+ void clear()	Clear iRenderable
+ Getters for instance and iRenderable	

3. Class ThreadHolder

3.1 Field

+ ThreadHolder instance	A singleton for ThreadHolder
- ArrayList<Thread> threads	An ArrayList of all threads in game

3.2 Constructor

+ ThreadHolder()	Initialize threads

3.3 Method

+ ArrayList<Thread> getThreads()	Getter for threads