Project Hero

by

1. Natthanan Ruengchaijatuporn 5930179921
2. Sawanphob Chavana 5931070221

2110215 Programming Methodology

2017/1

Chulalongkorn University

Project : Hero

1. About this game

Hero is a game that players assume that he or she will be a hero ,in order to fight with a monster to SAVE THE WORLD. Players will go into each stage and fight with monsters in that stage. The more monster the players kill, the more points the players will get. But if hero's hp is 0 or time is 0, the game will over, then an alert box displaying the score and players must click OK button, then they will go back to stage selection menu. After finishing all stage, the players complete this game.

The game has four scenes, a main menu, Sign In or Sign Up scene, Sign Up scene and a game scene. The main menu scene is The first one shown up when players start the program.

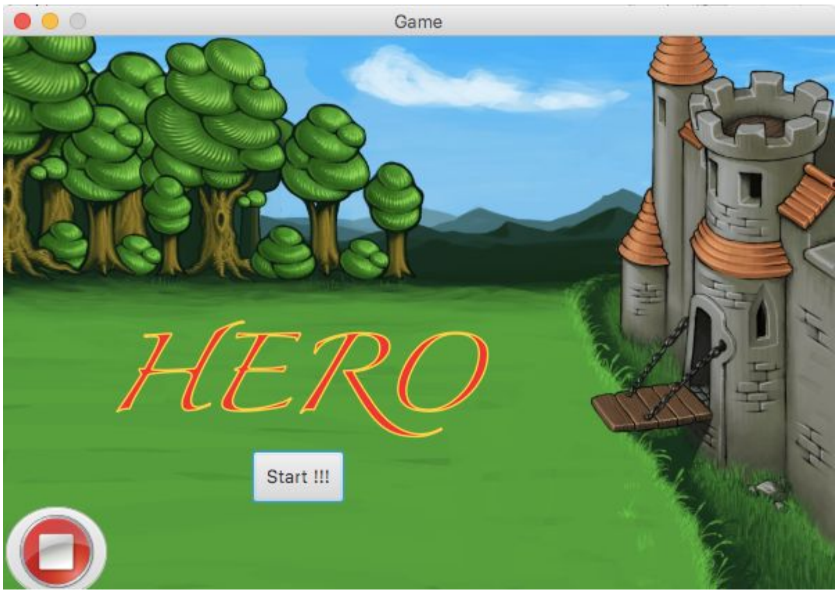


Figure 1 : The main menu scene of the application

At the main menu scene, the player :

- Press Start !!! button to start game.
- Press Stop button to close game.

When a player press Start !!! button to start the game. The program's window switches to the Sign In or Sign Up scene as shown in figure 2.

There are four parts displaying on this scene.

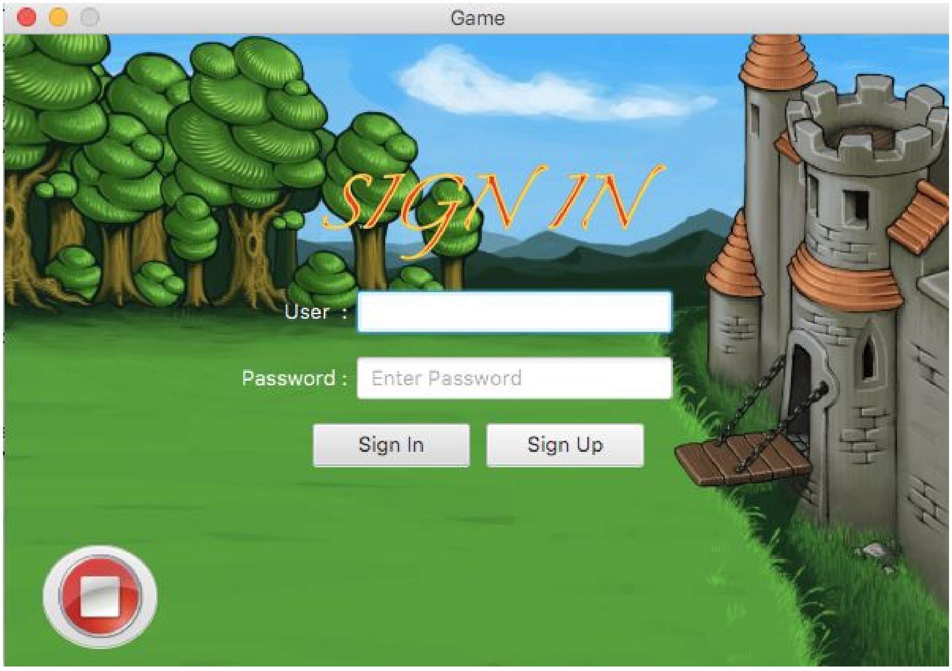


Figure 2 : The Sign In or Sign Up scene of the application

1. User text and user textfield for typing your username of your account.
2. Password text and password textfield for typing your password of your account.
3. Sign In button is pressed after filling user textfield and password textfield.
4. Sign Up button is for the players who don’t have account for this game.
5. Stop button is for exiting game.

If the players don’t have his or her account, they will press Sign Up button for creating their own account of this game. After clicking that button, the program’s window switches to the Sign Up scene as shown in figure 3. There five parts displaying on this scene.

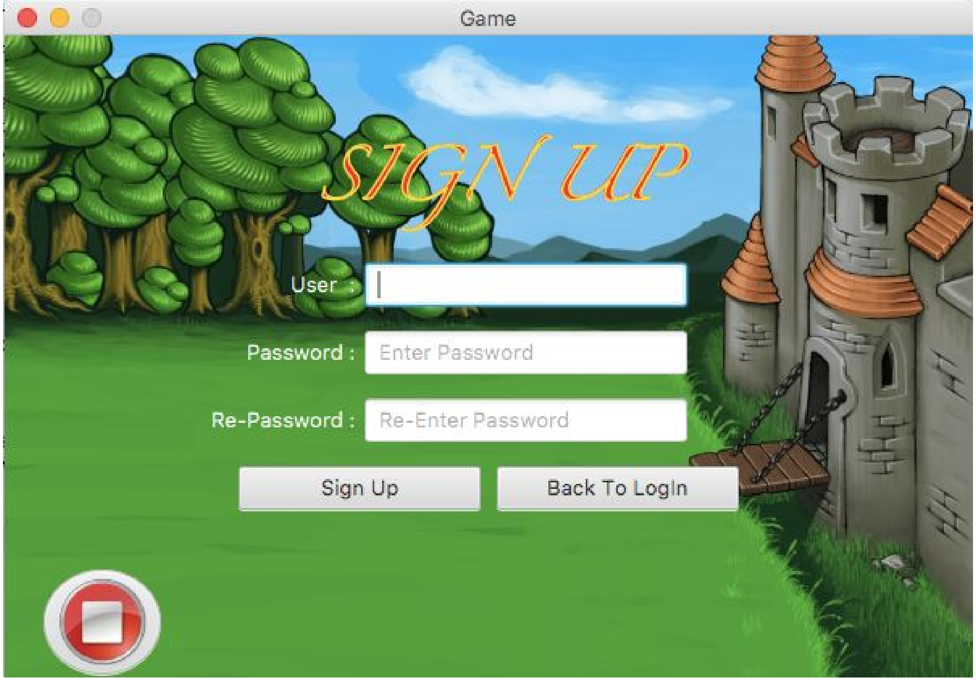


Figure 3 : The Sign Up menu of this application

1. User text and user textfield for typing your desired username of your account.
2. Password text and password textfield for typing your desired password of your account.
3. Re-Password text and re-password textfield for rechecking that your password.
4. Sign Up button is for creating account for this game.
5. Back To Login button is used for switches the scene back to the Sign In or Sign Up scene as shown in figure 2.

After the players already sign up thier account, they will press Back To Login button to switch back to the Sign In or Sign Up scene as shown in figure 2. Then, now, they can sign in with their account. After they press Sign In button, The program's window switches to the selection stage menu. Firstly, if the hero's level dosen't reach level 10 in time, stage 2 will not available as shown in figure 4. So, the players must reach level 10 in 60 seconds, in oreder to unlock stage 2.

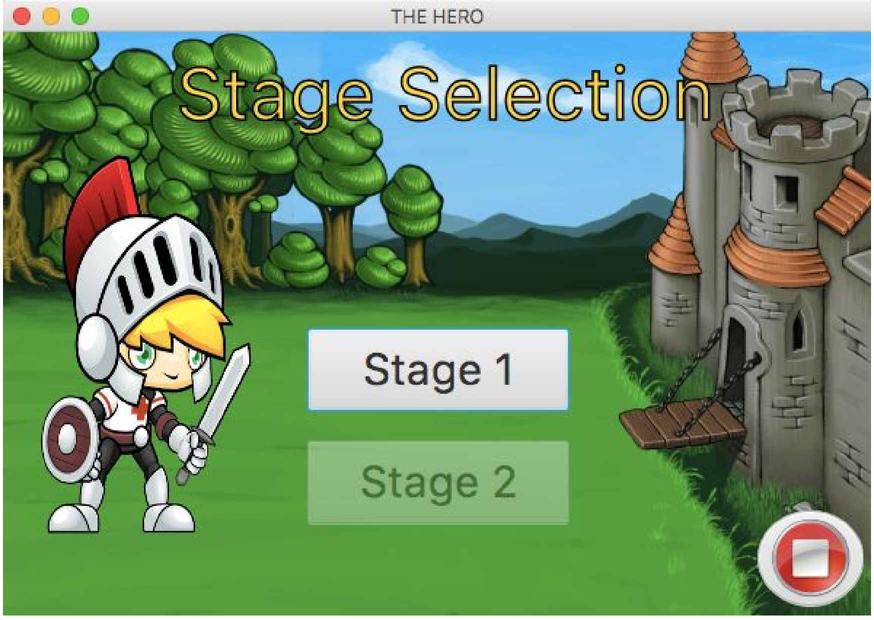


Figure 4 : if the hero's level dosen't reach level 10, stage 2 will not available.

In stage selection scene, there are three buttons.

1. Stage 1 button : After click this, player will go to stage 1.
2. Stage 2 button(Only available when hero's level is 10 in time(60 seconds)) : After click this, player will go to stage2.
3. Stop button : After click this, the program will exit.

After players click Stage 1 button. The program's window switches to game scene of stage 1 as shown in figure 5. There are six parts displaying on this scene.

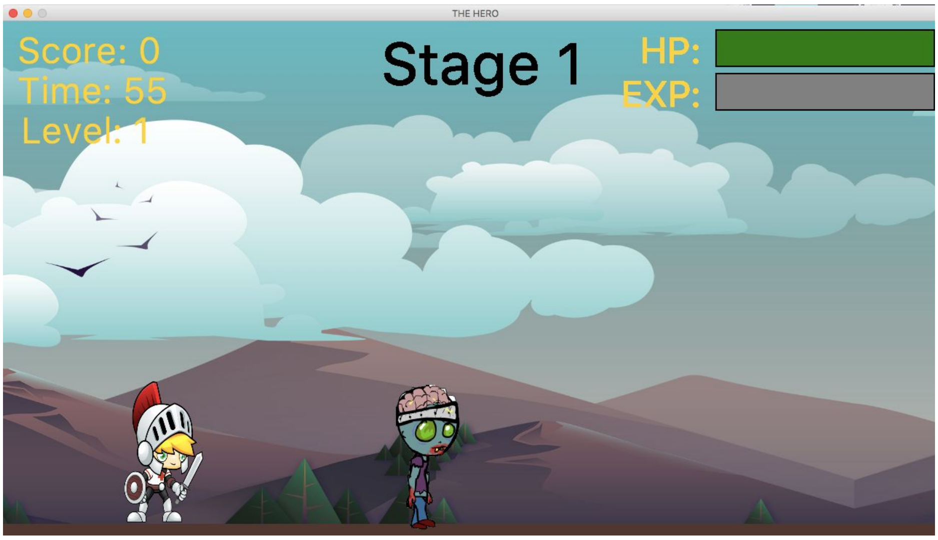


Figure 5 : The game screen of stage 1.

1. Score text at the top-left corner of the screen.

2. Time text at the top-left corner of the screen and below score text.
3. Level text at the top-left corner of the screen and below time text.
4. Stage text at the top-middle of the screen.
5. HP text and bar at the top-right corner of the screen.
6. EXP text and bar at the top-right corner of the screen and below HP text and bar.

First, a hero is start at the left of the stage 1. The players can move thier hero by using their keyboard.

- Press A button for moving to left.
- Press D button for moving to right.
- Press W button for jumping.
- Press SPACEBAR button for attacking

While the game is running, a monster is moving from right of the stage toward left and right direction repeatedly. Its moving direction changes to opposite when it hits the edges of the stage or it is attacked.

When a hero faces up with a monster, if the monster is facing to a hero and it is near enough, it will attack the hero automatically and your hero's HP will decrease. When the hero attacks a monster, the blood of the monster will spray. If the hero attacks a monster until it's hp going to 0, the monster will disappear from this stage and the players will earn one hundred points as shown in figure 6.

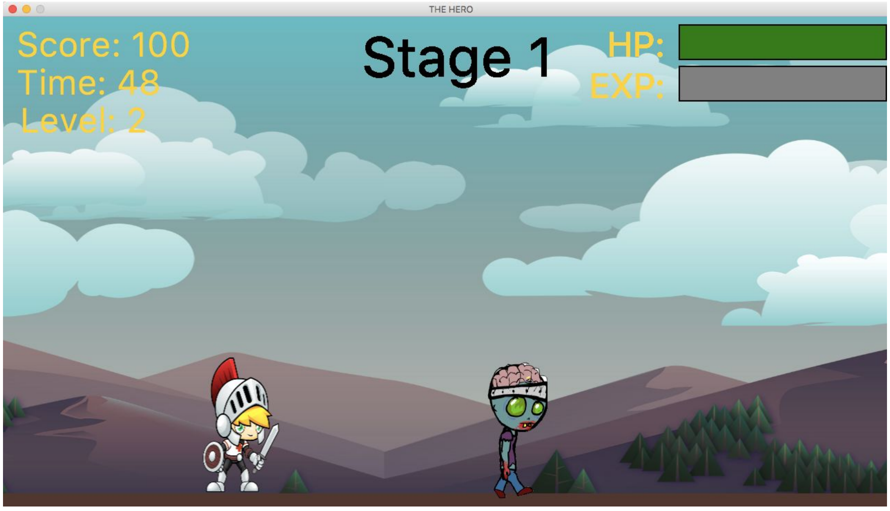


Figure 6 : When a monster is killed, players will earn one hundred points.

After clearing stage 1 with level of hero 10, then stage 2 is now available to select as shown as figure 7. So, now, the players can select stage 2.

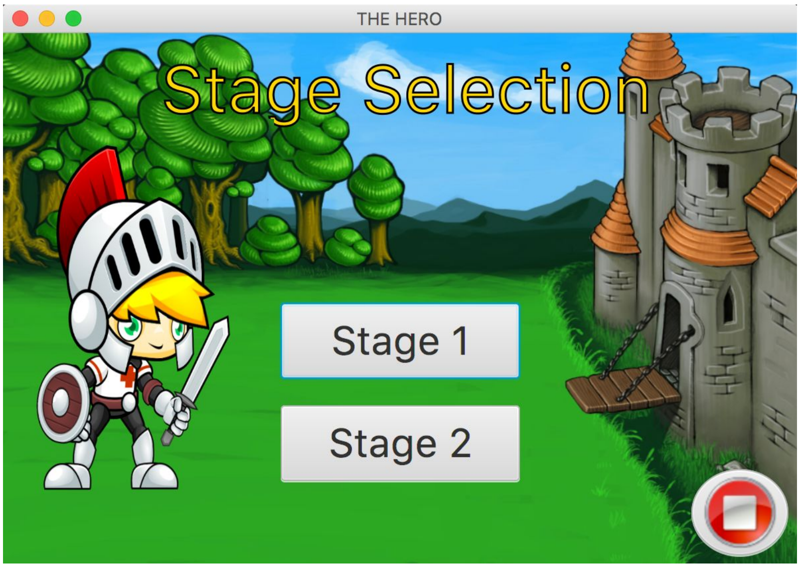


Figure 7 : After clearing stage 1 with level of hero 10, then stage 2 is now available to select.

After players click Stage 2 button. The program's window switches to game scene of stage 2 as shown in figure 8.

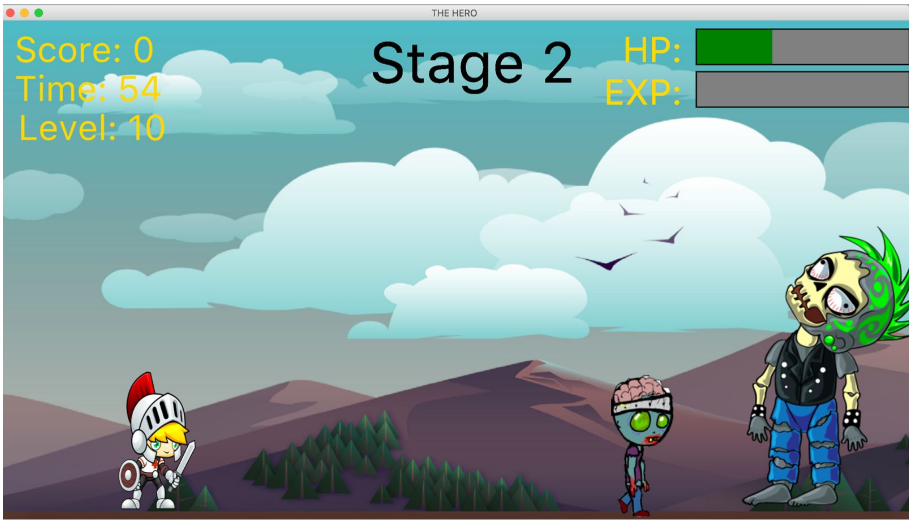
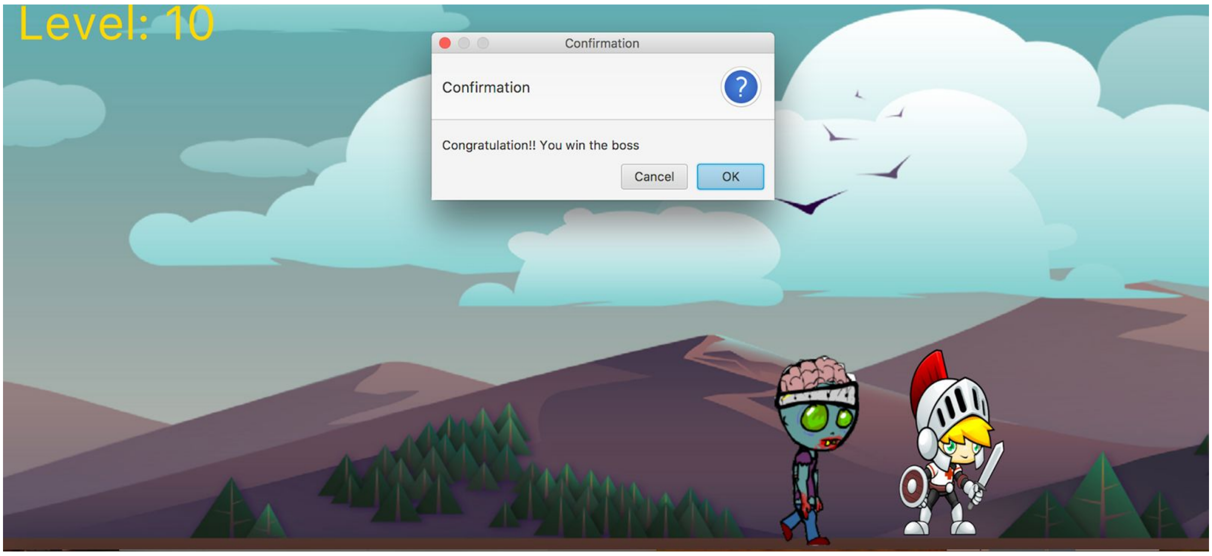


Figure 8 : The game screen of stage 2.

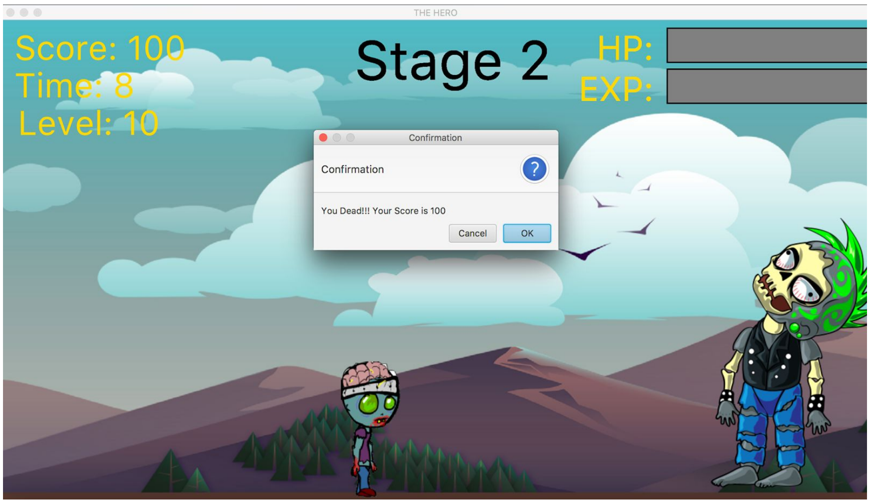
When the players clear stage 2 of this game, the alert box will show your score as shown as figure 9. After the players click OK button,the scene will switch to main menu scene.



Congratulation!! You win the boss

Figure 9 : When the player clear all stages

When your hero’s HP is 0, the alert box will show your score and two button like Cancel button and OK button as shown as figure 10.

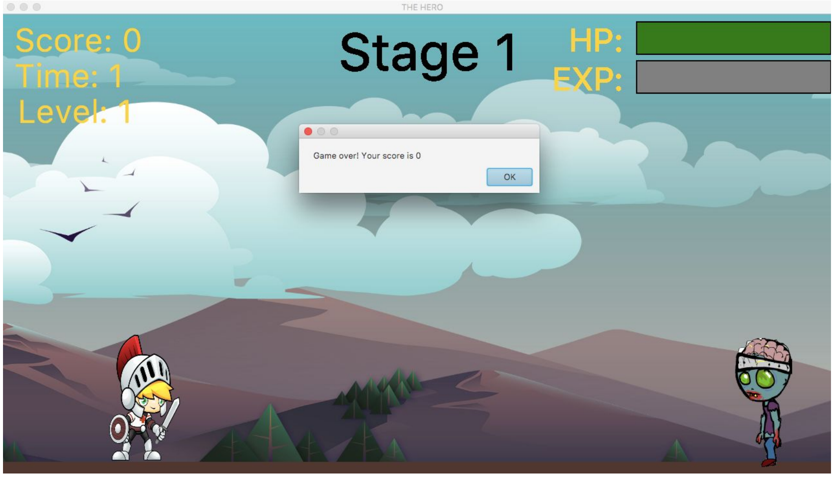


You Dead!!! Your Score is 100

Figure 10 : When a hero is dead

If the players select OK button, the scene will switch to stage selection scene as shown as figure 7. If the players select Cancel button, nothing will happen.

Finally, when the time is over. The application must stop and an alert box will pop up and show the score and two button like END GAME button and CONTINUE button as shown as figure 11.



Game over! Your score is 0

Figure11 : When the time is over

2. Implementation Detail

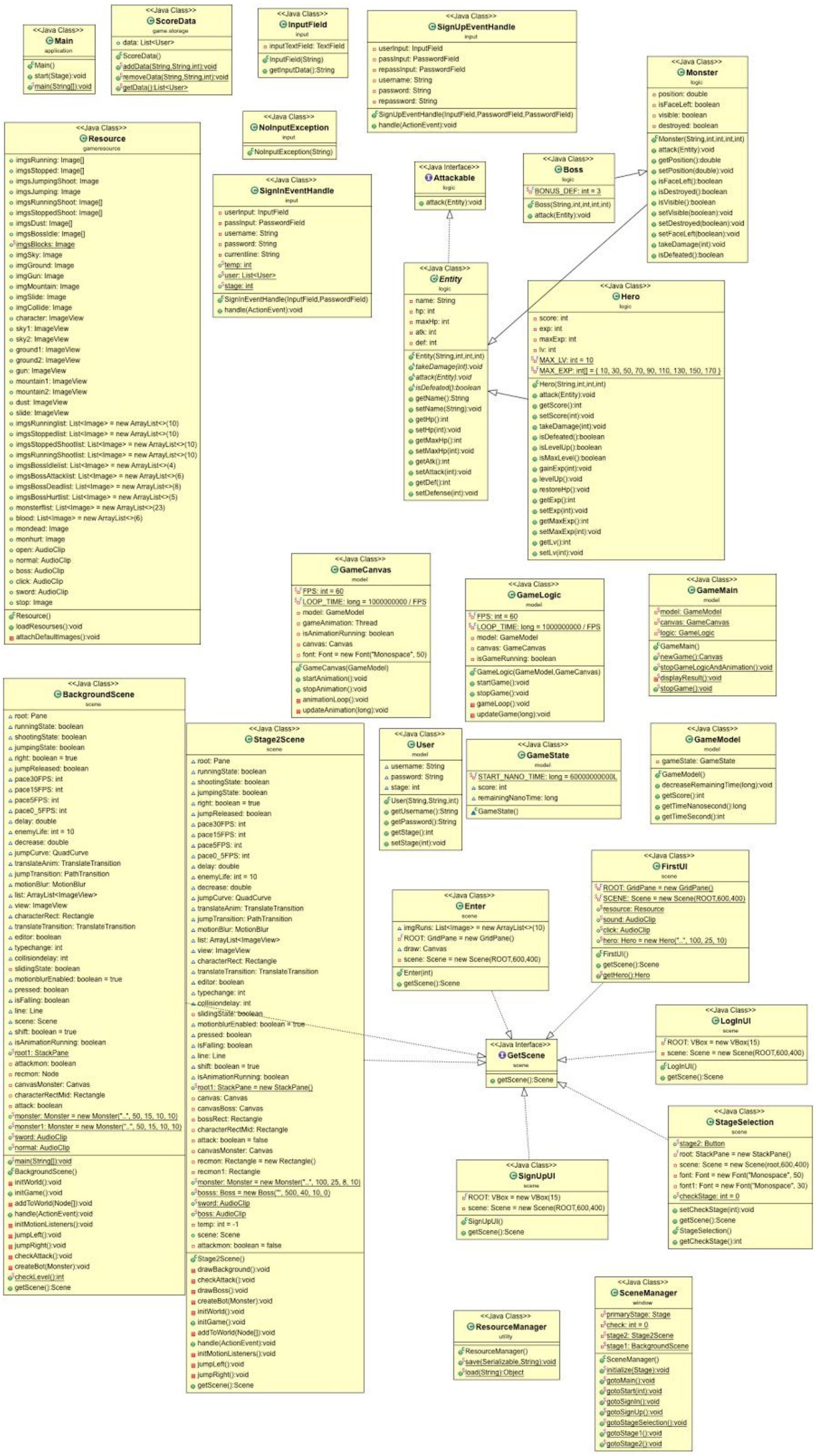


Figure 10 : The UML diagram of the program

The relation of the class is show in figure 10.

2.1 Package logic

2.1.1 Interface Attackable

2.1.1.1 Method

+ void attack(Entity opponent)	

2.1.2 Class Entity

2.1.2.1 Field

- String name	Entity’s name.
- int hp	Entity’s current hp.
- int maxHp	Entity’s max hp.
- int atk	Entity’s attack points.
- int def	Entity’s defense points.

2.1.2.2 Constructor

+ Entity(String name,int maxHp,int attack,int defense)	Initialize name, maxHp, hp, attack, defense. Hp starts at the same as maxHp.

2.1.2.3 Method

+ void takeDamage(int damage)	
+ void attack(Entity opponent)	
+ boolean isDefeated()	
+ Getter & Setter methods for every field	

2.1.3 Class Hero

2.1.3.1 Field

- int exp	Hero’s current exp.
- int maxExp	Hero’s max exp.
- int lv	Hero’s level.
- int MAX_LV	Hero’s max level
- int[] MAX_EXP	Hero’s max exp for each level

2.1.3.2 Constructor

+ Hero(String name,int maxHp,int attack,int defense)	Initialize name, maxHp, hp, attack, defense. Hp starts at the same as maxHp. Exp starts at 0. Lv start at 1. MaxExp start at MAX_EXP[0].

2.1.3.3 Method

+ void attack(Entity opponent)	Random bonus damage then set original attack plus with bonus damage. Call method takeDamage and the damage is equal to this Hero’s attack points decrease by opponent’s defense points. Then, set attack back to original damage.
+ void takeDamage(int damage)	Decrease Hero’s hp by damage and If the remain hp is less than zero, set it to zero.
+ boolean isDefeated()	True if Hero’s hp is less than or equal to zero.
+ boolean isLevelUp()	True if Hero’s exp is greater than or equal to max exp.
+ boolean isMaxLevel()	True if Hero reaches max level.
+ void gainExp(int exp)	Increase Hero’s exp by exp and check if Hero is level up and Hero’s level is

	not max level, then call method levelUP().
+ void levelUp()	increase Hero's level by 1 and set exp to exp minus max exp of the level before.
+ restoreHp()	Set Hero's hp to max hp.
+ Getter & Setter methods for every field except MAX_LV and MAX_EXP	

2.1.4 Class Monster

2.1.4.1 Field

- double position	Monster's position.
- boolean isFaceLeft	True if monster is face left.
- boolean visible	True if monster's alive.
- boolean destroyed	True if monster's hp is 0.

2.1.4.2 Constructor

+ Monster(String name, int maxHp, int attack, int defense, int position)	Initialize name, maxHp, hp, attack, defense, position. Hp starts at the same as maxHp. IsFaceLeft is true. Visible is true. Destroyed is false.

2.1.4.3 Method

+ void attack(Entity opponent)	Call method takeDamage and the damage is equal to this monster's attack points decrease by opponent's defense points.
+ Getter & Setter methods for every field	

2.1.5 Class Boss

2.1.5.1 Field

- int BONUS_DEF = 3	Boss's bonus defense.

2.1.5.2 Constructor

+ Boss(String name, int maxHp, int attack, int defense, int position)	Initialize name, maxHp, hp, attack, defense, position. Hp starts at the same as maxHp. IsFaceLeft is true. Visible is true. Destroyed is false.

2.1.5.3 Method

+ void attack(Entity opponent)	Random bonus damage then set original attack plus with bonus damage. Call method takeDamage and the damage is equal to this Hero's attack points decrease by opponent's defense points. Then, set attack back to original damage.

2.2 Package input

2.2.1 Class InputField extends HBox

2.2.1.1 Field

- TextField inputTextField	Textfield for recieving input.

2.2.1.2 Constructor

+ InputTextField(String text)	Initializes the InputTextField - Set the alignment of the InputTextField to Pos.BASELINE_CENTER. - Set the spacing to 5 - Instantiate a Label and a TextField : - A label creation is already given.

	- Assign the instantiated TextField field to inputTextField - inputTextField has text Enter with appropriate promt text and has a width of 200. - Add a label and inputTextField to the InputField as children.

2.2.1.3 Method

+ String getInputData()	Return text in the field

2.2.2 Class NoInputException extends Exception

2.2.2.1 Constructor

+ NoInputException	Exception for no input in InputTextField and call constructor of exception

2.2.3 Class SigninEventHandle implements EventHandler<ActionEvent>

2.2.3.1 Field

- InputField userinput	InputField for recieving username input.
- PassWordField passInput	PassWordField for recieving password input.
- String username	Username’s input.
- String password	Password’s input
- String currentline	Currentline for checking each line in .txt file
+ int stage	Stage.
+ int temp	For checking sign in state.
+ List<User> user	List of users.

2.2.3.2 Constructor

+SigninEventHandle(InputField userInput ,PasswordField passInput)	Initialize userInput, passInput.

2.2.3.3 Method

+ void handle(ActionEvent argo)	Check that username and password that user input are valid and also check that user input the username or password or not.

2.2.4 Class SignupEventHandle implements EventHandler<ActionEvent>

2.2.4.1 Field

- InputField userInput	InputField for recieving username input.
- PassWordField passInput	PassWordField for recieving password input.
- PassWordField repassInput	PassWordField for recieving repasswordinput.
- String username	Username that user input which get from userinput.
- String password	Password that user input which get from passInput.
- String repassword	Password that user input which get from repassInput.

2.2.4.2 Constructor

+ SignupEventHandle(InputField userInput ,PasswordField passInput ,PasswordField repassInput)	Initialize userInput, passInput, repassword.

2.2.4.3 Method

+ void handle(ActionEvent argo)	Create new account for user with the

	input username and password, check that user input the username or password or not and also has repassword field to ensure that password.

2.3 Package application

2.3.1 Class Main

2.3.1.1 Method

+ void start(Stage primaryStage)	The main entry point for the JavaFx application. This method should : - SceneManager initialize primaryStage. - SceneManager go to main stage. - Set title of window. - Set the window to be center on the sceen. - Set resizable to be false. - Set fullscreen to be false.
+ void main(String[] args)	An entry point of the application.

2.4 Package gameresource

2.4.1 Resource

2.4.1.1 Field

+ Image[] imgRunning	List of character running images.
+ Image[] imgStopped	List of character stopped images.
+ Image imgsJumpingShoot	Image of character jumping shoot.
+ Image imgsJumping	Image of character jumping.
+ Image[] imgsRunningShoot	List of of character running shoot images.
+ Image[] imgsStoppedShoot	List of character stopping shoot image.

+ Image[] imgsDust	List of dust images.
+ Image[] imgBossIdle	List of boss images.
+ Image imgsBlocks	Image of block.
+ Image imgSky	Image of sky.
+ Image imgGround	Image of ground.
+ Image imgSlide	Image of chracter sliding.
+ Image imgCollide	Image of character colliding.
+ ImageView character	Image of character.
+ ImageView sky1	Image of sky1
+ ImageView sky2	Image of sky2
+ ImageView ground1	Image of ground1
+ ImageView ground2	Image of ground2
+ ImageView mountain1	Image of mountain1
+ ImageView mountain2	Image of mountain2
+ ImageView slide	Image of slide
+ List<Image> imgsRunninglist	List of character running images.
+ List<Image> imgsStoppedlist	List of character stopping images.
+ List<Image> imgsStoppedShootlist	List of character stopping shoot images.
+ List<Image> imgsRunningShootlist	List of character running shoot images.
+ List<Image> imgsBossIdlelist	List of boss images.
+ List<Image> imgsBossAttacklist	List of boss attacking image.
+ List<Image> imgsBossDeadlist	List of boss dead images.
+ List<Image> imgsBossHurtlist	List of boss hurting images.

+ List<Image>monsterlist	List of monster moving images.
+ List<Image> blood	List of blood spraying images.
+ Image mondead	Image of dead monster.
+ Image monhurt	Image of hurted monster.
+ AudioClip open	Openning song.
+ AudioClip normal	General song in game.
+ AudioClip boss	Boss stage song.
+ AudioClip click	Clicking sound.
+ AudioClip sword	Sword sound effect.
+ Image stop	Image of stopping.

2.4.2 Method

+ void loadResorces()	Load all resource in field.
+ void attachDefaultImages()	Set size of character, ground1, ground2.

2.5 Package window

2.5.1 Class ScenenManager

2.5.1.1 Field

- Stage primaryStage	The primaryStage of SceneManager

2.5.1.2 Method

+ void initialize(Stage stage)	Initialize primaryStage and set primaryStage to be shown.
+ void gotoMain()	Initiate FirstUI and set primaryStage to that.
+ void gotoStart(int i)	Initiate Enter with parameter i and set primaryStage to that.
+ void gotoSignIn()	Initiate LoginUI and set primaryStage

	to that.
+ void gotoSignUp()	Initiate SignUpUI and set primaryStage to that.
+ void gotoStageSelection()	Initiate StageSelection and set primaryStage to that.

2.6 Package utility

2.6.1 Class ResourceManager

2.6.1.1 Method

+ void save(Serializable data,String fileName)	Save data with this fileName.
+ Object load(String fileName)	Load save game of this fileName.

2.7 Package game.storage

2.7.1 Class ScoreData implements Serializable

2.7.1.1 Field

+ List<User> data	List of all users.

2.7.1.2 Constructor

+ ScoreData()	Initialize data.

2.7.1.3 Method

+ void addData(String username, String password, int stage)	Add User with these username, password and stage to data.
+ List<User> getData()	Get user from data.

2.8 Model

2.8.1 Class GameState

2.8.1.1 Field

- long START_NANO_TIME	The total times of the game is 60 seconds.

int score	Current score of the player.
long remainingNanoTime	Remaining time left in nanoseconds.

2.8.1.2 Constructor

+ GameState	Set score to zero and remainingNanoTime to START_NANO_TIME.

2.8.2 Class User implements Serializable

2.8.2.1 Field

String username	User's username.
String password	User's password.
int stage	User's current stage.

2.8.2.2 Constructor

+ User(String username, String password, int stage)	Initialize all fields with the given parameters.

2.8.2.3 Method

+ Getter method for every fields and Setter method for stage	

2.8.3 Class GameModel

2.8.3.1 Field

- GameState gameState	A game state.

2.8.3.2 Constructor

+ GameModel(GameState gameState)	Initialize gameState.

2.8.3.3 Method

+ void decreaseRemainingTime(long decreasedNanoTime)	Subtract the remaining time by decreasedNanoTime.
+ int getScore()	Return current score.
+ long getTimeNanoSecond()	Return remaining time in nanoseconds.
+ int getTimeSecond	Return remaining time in second.

2.8.4 Class GameCanvas extends Canvas

2.8.4.1 Field

- int FPS	Number of frames rates per second. Default is 60.
- long LOOP_TIME	Time period between each update of a game animation.
- GameModel gameModel	A game model.
- Thread gameAnimation	A thread for game animation.
- boolean isAnimationRunning	The flag indicate that the game is start and not end yet.
- Font font	A font for this canvas.

2.8.4.2 Constructor

+ GameCanvas(GameModel model)	- Initialize canvas with width and height equal to 1300 and 710 respectively. - Set model with given parameter. - Set isAnimationRunning to false.

2.8.4.3 Method

+ void startAnimation()	Start the game animation loop thread and set isGameRunning to true and

	setDaemon of gameAnimation to be true.
+ void stopAnimation()	Set isAnimationRunning to false.
+ void animationloop()	The game animation loop : - The loop will stop when isGameRunning is false. - The animation will update when the time has pass by a LOOP_TIME. - For each loop, call updateAnimation().
+ void updateAnimation()	Draw the screen interface as shown in figure 4. - Set all text font to font. - Score, time, level, hp, exp color is gold. - Stage color is black. - Background of hp bar and exp bar is gray. - The amount of hp and exp represent by amount of green color in each bar. - Score text at the top-left corner of the screen. - Time text at the top-left corner of the screen and below score text. - Level text at the top-left corner of the screen and below time text. - Stage text at the top-middle of the screen. - HP text and bar at the top-right corner of the screen. - EXP text and bar at the top-right corner of the screen

	and below HP text and bar.

2.8.5 Class GameLogic

2.8.5.1 Field

- int FPS	Number of frame rates per second. Default is 60.
- long LOOP_TIME	Time period between each update of a game logic.
- GameModel model	A game model.
- GameCanvas canvas	A game canvas.
- boolean isGameRunning	The flag indicate that the game is start and not end yet.

2.8.5.2 Constructor

+ GameLogic(GameModel model, GameCanvas canvas)	Set model and canvas by the given parameters and set isGameRunning to false.

2.8.5.3 Method

+ void startGame()	Set isGameRunning to true and set thread daemon to true. Then start the game loop.
+ void stopGame()	Set isGameRunning to false.
+ void gameLoop()	The main game loop: - The loop will stop when isGameRunning is false. - The game will only update when the elapse time is equals or greater than the LOOP_TIME. - For each loop, call updateGame().

+ void updateGame(long elapsedTime)	Decrease the remaining time by the given elapsedTime using model. If the remaining time reaches zero, stop the game.

2.8.6. Class GameMain

2.8.6.1 Field

- GameModel model	A game model.
- GameCanvas canvas	A game canvas.
- GameLogic logic	A game logic.

2.8.6.2 Method

+ Canvas newGame()	Initialize model, canvas and logic, switch to the game scene by utilizing SceneManager, then start the game logic and game animation. Return canvas.
+ void stopGameLogicAndAnimation()	Tell logic to stop the game and tell canvas to stop the animation.
+ void displayResult()	Show Alert game over message and score. After the player click OK button swtich to stage selection by SceneManager.
+ void stopGame()	Stop the game logic and the game animation then display the game result as an alert box.

2.9 Package scene

2.9.1 Interface GetScene

+ Scene GetScene()	

2.9.2 Class Enter implements GetScene

2.9.2.1 Field

List<Image> imgRuns	List of running image of character.
- GridPane root	Grid pane is root of this scene.
Canvas draw	A canvas.
- Scene scene	new scene with root and set width and height equal to 600 and 400 respectively.

2.9.2.2 Constructor

+ Enter(int a)	Set root padding with insets(100,90,0,0). Set background to be Sega-Game-Background.jpg. Initailize draw with width and height equal to 300 and 50 respectively. Draw animation of character running into castle and the go to stage a which is parameter.
+ Scene getScene()	Return scene.

2.9.3 Class LogInUI implements GetScene

2.9.3.1 Field

- VBox root	VBox with spacing 15.
- Scene	Scene with root and set width and height equal to 600 and 400 respectively.

2.9.3.2 Constructor

+ LoginUI()	Set alignment to center. Set padding with insets(20). Initialize canvas with width and height equal to 250 and 100 respectively. Set font to Zapfino and size to 30. Set fill to red and set stroke to gold with line width 1.5. Draw text SIGN IN. Set background to Sega-Game-Background.jpg. Set

	up stop button at bottom left. Create HBox with spacing 5. Create InputField for User and PasswordField for password. Below PasswordField, there are Sign In and Sign Up button with eventhandler. Then, add to HBox.

2.9.3.3 Method

+ Scene getScene()	Return scene.

2.9.4 Class SignUpUI implements GetScene

2.9.4.1 Field

- VBox root	VBox with spacing 15.
- Scene	Scene with root and set width and height equal to 600 and 400 respectively.

2.9.4.2 Constructor

+ LoginUI()	Set alignment to center. Set padding with insets(20). Initialize canvas with width and height equal to 250 and 100 respectively. Set font to Zapfino and size to 30. Set fill to red and set stroke to gold with line width 1.5. Draw text SIGN UP. Set background to Sega-Game-Background.jpg. Set up stop button at bottom left. Create HBox with spacing 5. Create InputField for User, PasswordField for password and another PasswordField for Re-Password. Below PasswordField, there are Sign Up and Back To Login button with eventhandler. Then, add to HBox.

2.9.4.3 Method

+ Scene getScene()	Return scene.

2.9.5 Class FirstUI implements GetScene

2.9.5.1 Field

- GridPane root	Grid pane.
- Scene	Scene with root and set width and height equal to 600 and 400 respectively.
+ Resource resource	A resource.
+ AudioClip sound	Background for this scene.
+ AudioClip click	Sound for click in this scene.

2.9.5.2 Constructor

+ FirstUI()	Initialize resource. Initialize sound with resource.open and Initial click with resource.click. Create canvas with width and height equal to 600 and 400 respectively. Set font to Zapfino with size 50. Set fill to red and set stroke to gold then, draw text “HERO” at position (75,270). Set background to Sega-Game-Backgrounds.jpg. Create start button below text “HERO” and add eventhandler. Create stop at buttom left corner with image from resource then add eventhandler to stop button.

2.9.5.3 Method

+ Scene getScene()	Return scene.

2.9.6 Class StageSelection implements GetScene

2.9.6.1 Field

- StackPane ROOT	Stack pane to be a root.
- Scene scene	Scene with ROOT and set width and height equal to 600 and 400 respectively.
- Font font	Font for word “Stage selection”. Initial with font Monospace and size to 50.
- Font font1	Font for word “Stage 1” and “Stage 2”. Initial with font Monospace and size to 30.
+ int checkStage	Number for checkStage initialize with zero.

2.9.6.2 Constrctor

+ StageSelection()	Set alignment of root to center. Set padding with insets(20). Set background with image Sega-Game-Backgrounds.jpg. Create canvas with width and height equal to 600 and 400 respectively. Set font with font, set fill to be gold and set stroke to black. Draw text “Stage Selection” with postion (120,60). Create new thread to show animation of character when stopping. Then, start thread. Create new VBox with spacing 20. Set alignment of VBox to center right. Set translateX with -190 and translateY to 70. Set font to font1. Create two buttons and set PrefSize(180,20). The first button is Stage 1. The other button is Stage 2. Add eventhandler to both buttons. Create stop button at bottom right with image from resource set PrefSize to be 55, translateX with 255,

	translateY with 160. Then, add eventhandler.

2.9.6.3 Method

+ void setCheckStage(int checkStage)	Set stageCheck to stageCheck which is given parameter.
+ int getCheckStage()	Return checkStage.
+ Scene getScene()	Return scene.

2.9.7 Class BackgroundScene implements EventHandler<ActionEvent>, getScene

2.9.7.1 Field

# Resource res	A resource of game.
# Pane root	A pane of scene.
# boolean runningState	True if character running. Otherwise is false.
# boolean shootingState	True if character shooting. Otherwise is false.
# boolean jumpingState	True if character jumping. Otherwise is false.
# boolean right	True if character faces right. Otherwise is false.
# boolean jumpReleased	True if jump button is released. Otherwise is false.
# int pace30FPS	Pace for 30 FPS.
# int pace15FPS	Pace for 15 FPS.
# int pace5FPS	Pace for 5 FPS.
# int pace0_5FPS	Pace for 0.5 FPS.

# double dalay	Amount of delay of that time in game.
# int enemyLife	Life of enemy initialize with 10.
# double decrease	Amount that use for decreasing.
# QuadCurve jumpCurve	Curve for jumping.
# Translate Transition translateAnim	Translation transition for animation.
# Path Tranition jumpTransition	Traslation for jumping.
# MotionBlur motionBlur	Use for blur something like blur background while jumping.
# ArrayList<ImageView> list	list of image.
# ImageView view	view for this stage.
# Rectangle characterRect	Rectangle that uses to represent canvas position.
# Translate Transition translate	Translation.
# boolean editor	True if editable. Otherwise is false.
# int typeChange	Amount of typeChange.
# int collisiondelay	The delay when collision is happen.
- boolean slidingState	True if character is sliding. Other wise is false.
# boolean motionblurEnabled	True if that thing is enable to blur.
# boolean pressed	True if keyboard is pressed. Otherwise is false.
#boolean isFalling	True if character is falling. Otherwise is false.
#Line line	Line in this scene.
#boolean shift	True if anything is shift.
#boolean isAnimationRunning	True if animation is running. Otherwise is false.

+ StackPane root1	Using stackpane for root.
- boolean attackmon	True if character is attacking. Otherwise is false.
- Node recmon	Node of recmon.
- Canvas canvasMonster	Canvas of monster.
- Rectangle characterRectMid	Frame of character that use for detecting when moster
- boolean attack	True if monster is attacked. Otherwise is false.
+ Hero hero	Create hero with hp = 100, attack = 25 and defense = 10.
+ Monster monster	Create monster with hp = 50, attack = 15 and defense = 10.
+ Monster monster1	Create monster with hp = 50, attack = 15 and defense = 10.
+ AudioClip sword	Sound when hero attacks monster.
+ AudioClip normal	Sound of background for this stage.

2.9.7.2 Constructor

+ BackgroundScene()	Firstly, call method initResources to load all resources. Initialize sword with res.sword and normal with res.normal. Set cycle of normal to 100 cycles then play it. Initialize root with pane and root1 with stack pane then add root into root1. Initialize scene with root1 and set width and height equal to 1950 and 1250 respectively. Set scene of primaryStage to be scene. Set resizable of primaryStage to be false the show primaryStage. Initialize line and list. Call method initMotionListeners(), initWorld()

	and initGame() to set up everythin to be already for start game. Add GameMain.newGame() to root1 then, create bot with monster .

2.9.7.3 Method

+ void main(String[] args)	An entry point of the application.
- void initWorld()	Initialize jumpCurve, motionBlur, characterRect and translateTransition. Set interPolartor of translateTransition to Interpolar.EASE_IN. Set fill to transparent color. Set width and height of characterRect then bind both xProperty and yProperty of characterRect. Initialize jumpTransition with duration 1000 millisecond, jumpCurve and res.character. Set start point of curve, control x and y property of curve and end point of curve. Set ground1 translateXproperty to bind with character. Add res.ground1 and res.character to world.
+ void initGame()	Initialize timeline with keyFrame equal to duration 33.33 millisecond. Set cycle of timeline to be indefinite then play timeline and initialize transIteAnim with duration 20000 milliseconds and res.character.
- void initResources()	Initialize res and call method loadResources of res.
- void addToWorld(Node… n)	Add all node n to root.
- void addToWorld	Add node n to root.
+ void handle(ActionEvent event)	Control timeframe and check state.

- void initMotionListerners()	Set event handler to root. If player presses A, character will move to left. If player presses D, character will move to right. If player presses W, character jump to the direction that he faces. If player presses spacebar, character will attack with his sword.
- void jumpLeft()	Set property of left jumping.
- void jumpRight()	Set property of right jumping.
- void unbindWorld()	Unbind x property of all node in the world of this game.
- void rebindWorld(boolean left)	Bind x property of background in the world of this game.
- void checkAttack()	If canvases of monster and hero are intersection. Monster will attack hero automatically.
+ void createBot(double, Monster)	Draw a monster walks repeatedly from left to right and right to left. If the monster is attacked by hero, monster's blood will spray off. If monster's current hp is less than 25%, the monster will change to another status with red arm. If the monster is defeated, it will change to dead status and then disappear.
+ int checkLevel()	Return 1 if level of hero is 10, Stage 2 will be available. Otherwise return 0.
+ Scene getScene()	Return scene.

2.9.8 Class Stage2Scene implements EventHandler<ActionEvent>, getScene

2.9.8.1 Field

# Resource res	A resource of game.
# Pane root	A pane of scene.

# boolean runningState	True if character running. Otherwise is false.
# boolean shootingState	True if character shooting. Otherwise is false.
# boolean jumpingState	True if character jumping. Otherwise is false.
# boolean right	True if character faces right. Otherwise is false.
# boolean jumpReleased	True if jump button is released. Otherwise is false.
# int pace30FPS	Pace for 30 FPS.
# int pace15FPS	Pace for 15 FPS.
# int pace5FPS	Pace for 5 FPS.
# int pace0_5FPS	Pace for 0.5 FPS.
# double dalay	Amount of delay of that time in game.
# int enemyLife	Life of enemy initialize with 10.
# double decrease	Amount that use for decreasing.
# QuadCurve jumpCurve	Curve for jumping.
# Translate Transition translateAnim	Translation transition for animation.
# Path Tranition jumpTransition	Translation for jumping.
# MotionBlur motionBlur	Use for blur something like blur background while jumping.
# ArrayList<ImageView> list	list of image.
# ImageView view	view for this stage.
# Rectangle characterRect	Rectangle that uses to represent canvas position.
# Translate Transition translate	Translation.

# boolean editor	True if editable. Otherwise is false.
# int typeChange	Amount of typeChange.
# int collisiondelay	The delay when collision is happen.
- boolean slidingState	True if character is sliding. Other wise is false.
# boolean motionblurEnabled	True if that thing is enable to blur.
# boolean pressed	True if keyboard is pressed. Otherwise is false.
#boolean isFalling	True if character is falling. Otherwise is false.
#Line line	Line in this scene.
#boolean shift	True if anything is shift.
#boolean isAnimationRunning	True if animation is running. Otherwise is false.
+ <u>StackPane root1</u>	Using stackpane for root.
- Canvas canvas	Canvas for drawing.
- Canvas canvasBoss	CanvasBoss for canvas of boss.
- Rectangle boosRect	Rectangle for boss.
- Rectangle recmon1	Rectangle for monster.
- int temp	Counting value for loop.
- boolean attackmon	True if character is attacking. Otherwise is false.
- Node recmon	Node of recmon.
- Canvas canvasMonster	Canvas of monster.
- Rectangle characterRectMid	Frame of character that use for detecting when moster
- boolean attack	True if hero attacks moster.

	Other wise
+ Hero hero	Create hero with hp = 100, attack = 25 and defense = 10.
+ Monster monster	Create monster with hp = 50, attack = 15 and defense = 10.
+ Monster monster1	Create monster with hp = 50, attack = 15 and defense = 10.
+ AudioClip sword	Sound when hero attacks monster.
+ AudioClip normal	Sound of background for this stage.

2.9.8.2 Constructor

+ Stage2Scene()	Initialize sword with res.sword and normal with res.normal. Set cycle of normal to 100 cycles then play it. Initialize root with pane and root1 with stack pane then add root into root1. Initialize scene with root1 and set width and height equal to 1950 and 1250 respectively. Set scene of primaryStage to be scene. Set resizable of primaryStage to be false the show primaryStage. Initialize line and list. Call method initMotionListeners(), initWorld() and initGame() to set up everythin to be already for start game. Add GameMain.newGame() to root1 then, create bot with monster .

2.9.8.3 Method

+ void main(String[] args)	An entry point of the application.
- void initWorld()	Initialize jumpCurve, motionBlur, characterRect and translateTransition. Set interPolartor of translateTransition to

	Interpolar.EASE_IN. Set fill to transparent color. Set width and height of characterRect then bind both xProperty and yProperty of characterRect. Initialize jumpTransition with duration 1000 millisecond, jumpCurve and res.character. Set start point of curve, control x and y property of curve and end point of curve. Set ground1 translateXproperty to bind with character. Add res.ground1 and res.character to world.
+ void initGame()	Initialize timeline with keyFrame equal to duration 33.33 millisecond. Set cycle of timeline to be indefinite then play timeline and initialize translteAnim with duration 20000 milliseconds and res.character.
- void addToWorld(Node… n)	Add all node n to root.
+ void handle(ActionEvent event)	Control timeframe and check state.
- void initMotionListerners()	Set event handler to root. If player presses A, character will move to left. If player presses D, character will move to right. If player presses W, character jump to the direction that he faces. If player presses spacebar, character will attack with his sword.
- void jumpLeft()	Set property of left jumping.
- void jumpRight()	Set property of right jumping.
- void checkAttack()	If canvases of monster and hero are intersection. Monster will attack hero automatically.
+ void createBot(double, Monster)	Draw a monster walks repeatedly from left to right and right to left. If

	the monster is attacked by hero, monster’s blood will spray off. If monster’s current hp is less than 25%, the monster will change to another status with red arm. If the monster is defeated, it will change to dead status and then disappear.
- void drawBackground()	Draw background in canvas then add to root.
+ Scene getScene()	Return scene.
- void drawBoss()	Create a boss. If the boss is attacked, it will change to hurting state. If the boss is dead, it will jump up and then fall to the ground.