Java Project Report

On

Bullet Hell

Submitted by

Sirawitch Trevibul 6231368521

Thanainan Li 6231328421

Subject

Course Programming Methodology Semester 1

Story behind the game : Bullet Hell

Figure 1 : The game "Bullet Heaven 2 "

The term "bullet hell" refers to a sub-genre of shooters whose main gameplay focus is dodging seemingly endless waves of colored bullets. In these games the player controls a small ship or character which flies around the environment, shooting at waves of enemies until eventually reaching a stage boss. A true bullet hell shooter tests the player's skill in being able to recognize patterns in the stream of bullets and then navigate through the barrage. Bosses will often use complex bullet patterns that restrict the player to very tight safe zones that require maximum precision and focus to survive. It's not uncommon that a Bullet Hell will use bright colors and distracting visual elements in an attempt to break the player's concentration.

1.User Manual

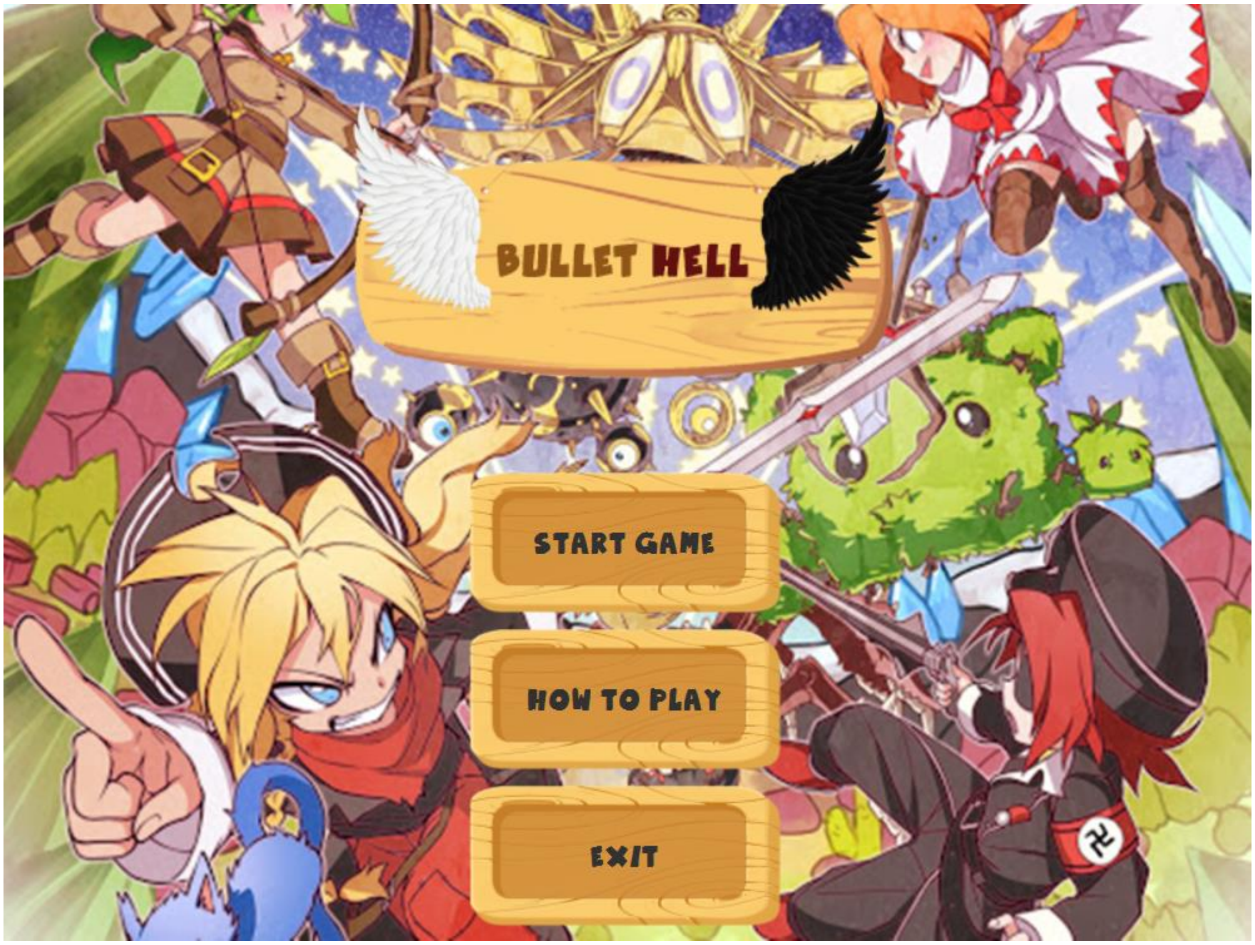


Figure 2 : Welcome screen of the game

When start the program, you will see a welcome screen that have 3 buttons : START GAME , HOW TO PLAY and Exit as show in figure 2.

When you click EXIT:

- Your game will exit.

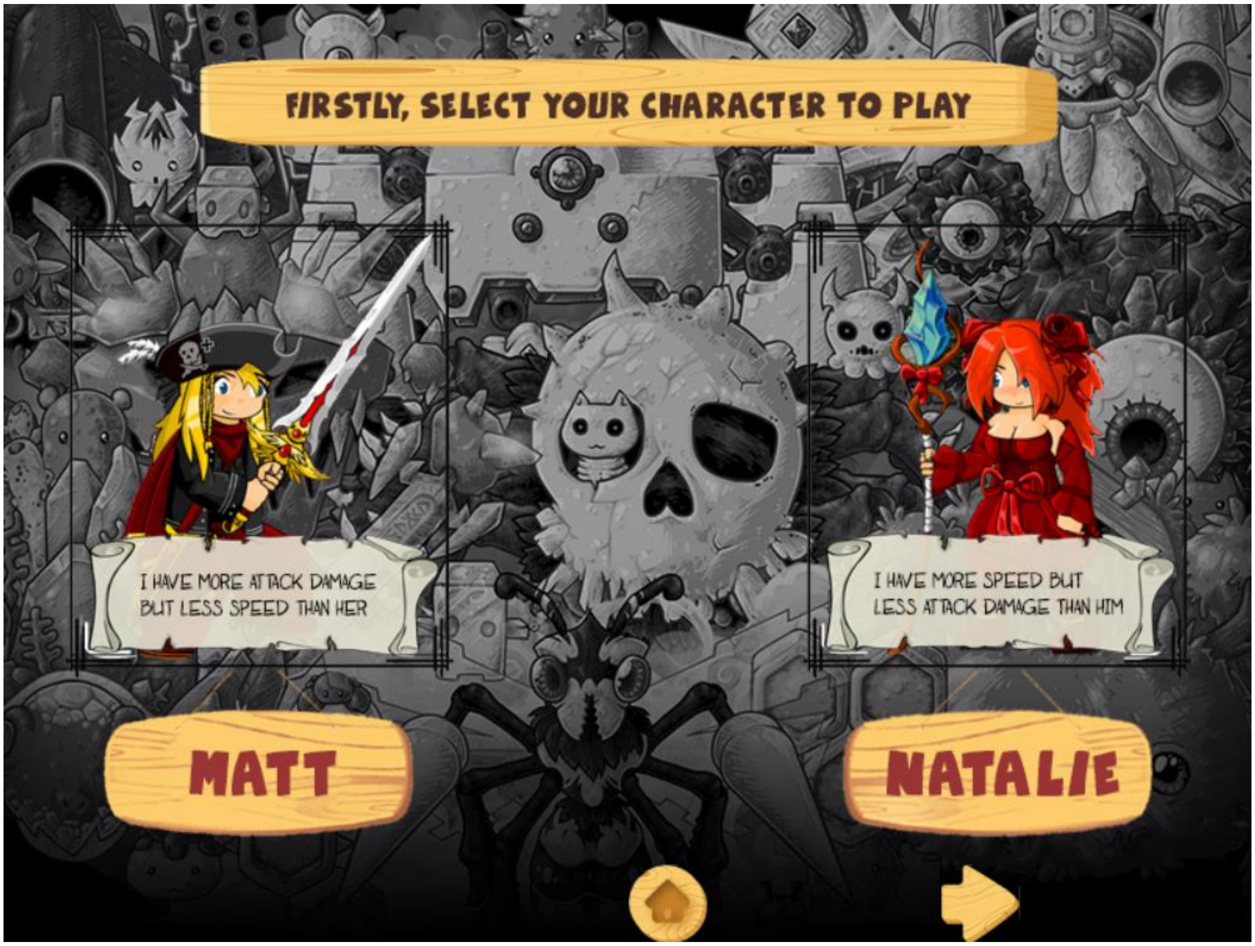
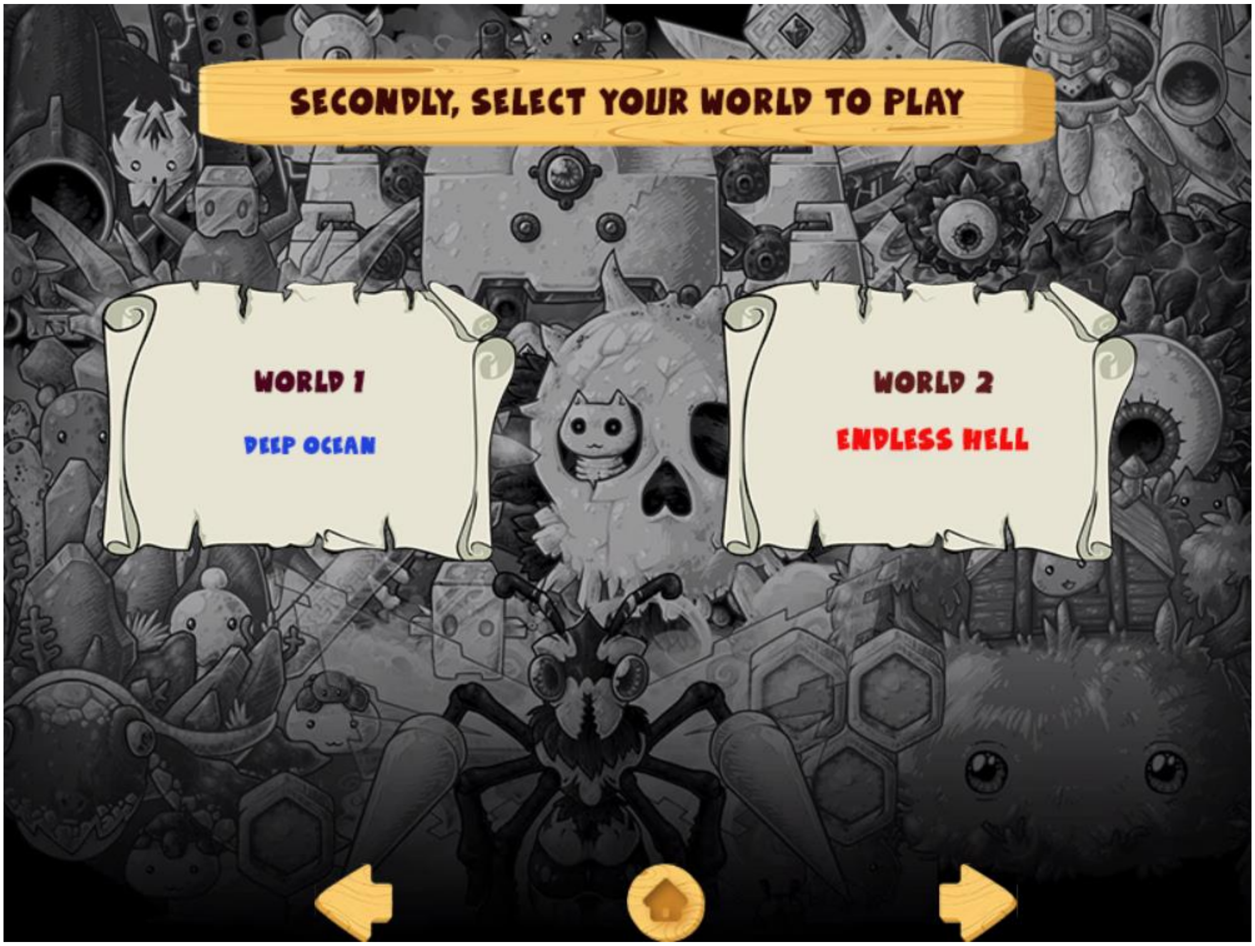

FIRSTLY, SELECT YOUR CHARACTER TO PLAY
I HAVE MORE ATTACK DAMAGE BUT LESS SPEED THAN HER
I HAVE MORE SPEED BUT LESS ATTACK DAMAGE THAN HIM
MATT
NATALIE

Figure 3.1 : How to play screen 1



SECONDLY, SELECT YOUR WORLD TO PLAY

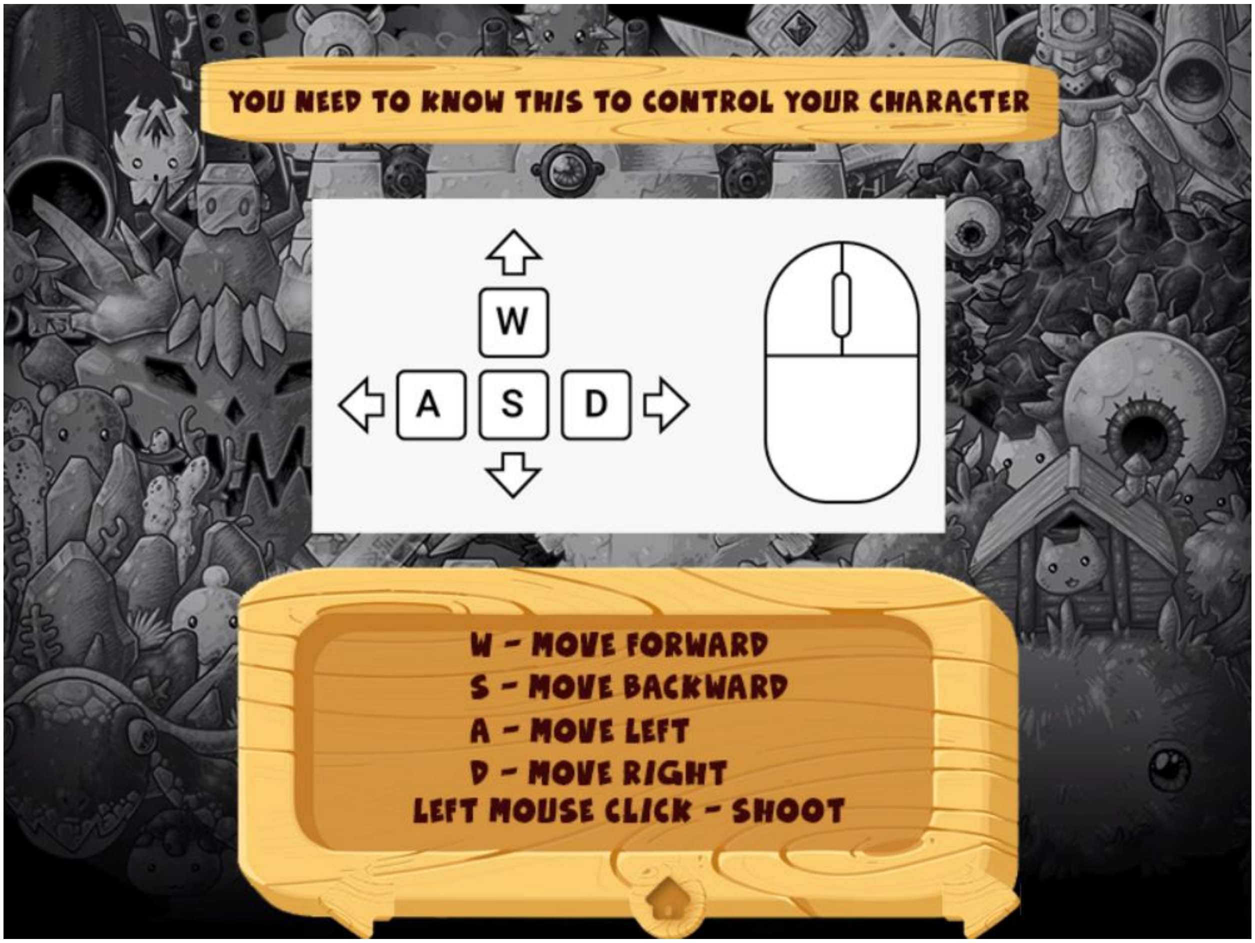
WORLD 1

DEEP OCEAN

WORLD 2

ENDLESS HELL

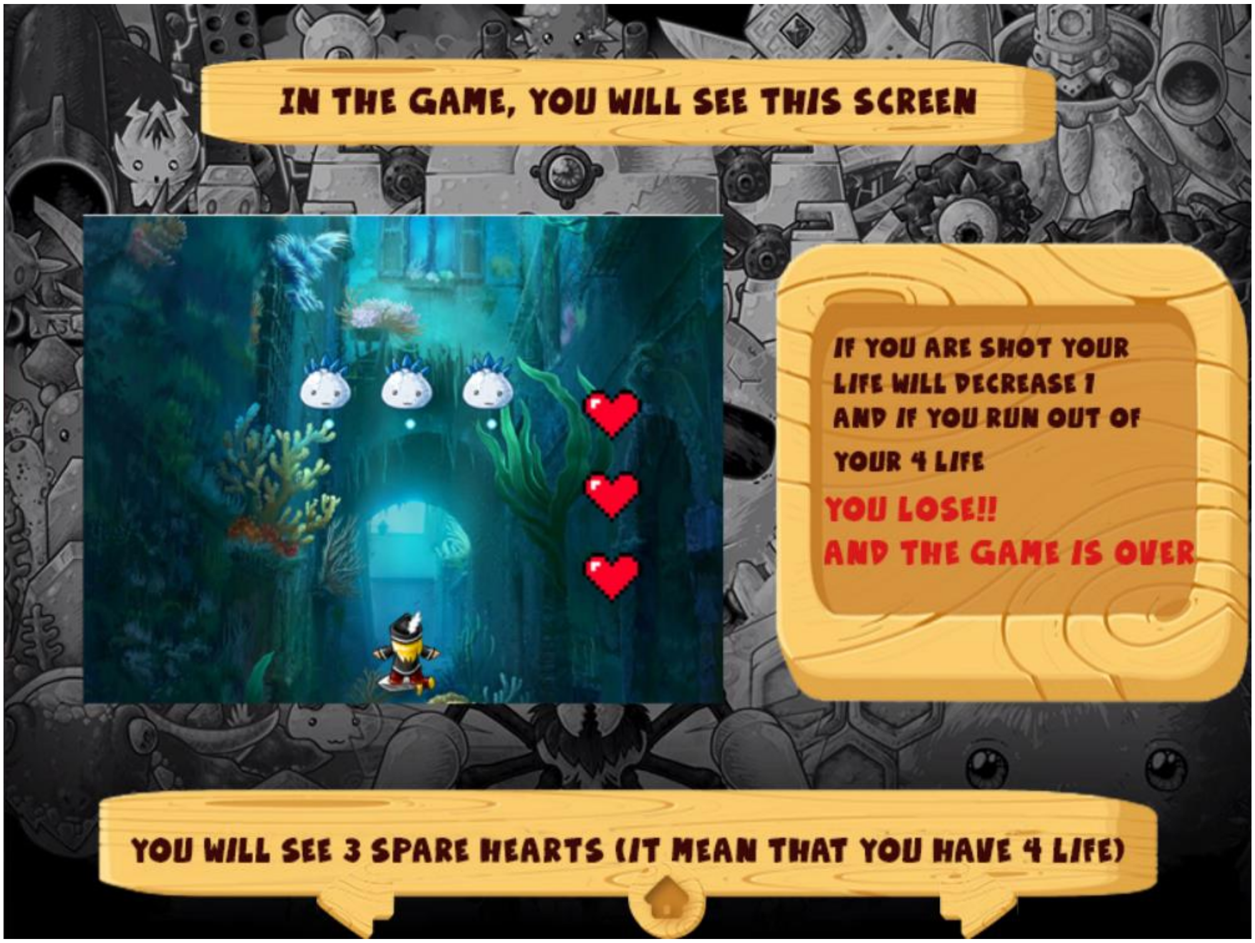
Figure 3.2 : How to play screen 2



YOU NEED TO KNOW THIS TO CONTROL YOUR CHARACTER

W - MOVE FORWARD
S - MOVE BACKWARD
A - MOVE LEFT
D - MOVE RIGHT
LEFT MOUSE CLICK - SHOOT

Figure 3.3 : How to play screen 3



IN THE GAME, YOU WILL SEE THIS SCREEN

IF YOU ARE SHOT YOUR LIFE WILL DECREASE 1 AND IF YOU RUN OUT OF YOUR 4 LIFE

YOU LOSE!!

AND THE GAME IS OVER

YOU WILL SEE 3 SPARE HEARTS (IT MEAN THAT YOU HAVE 4 LIFE)

Figure 3.4 : How to play screen 4

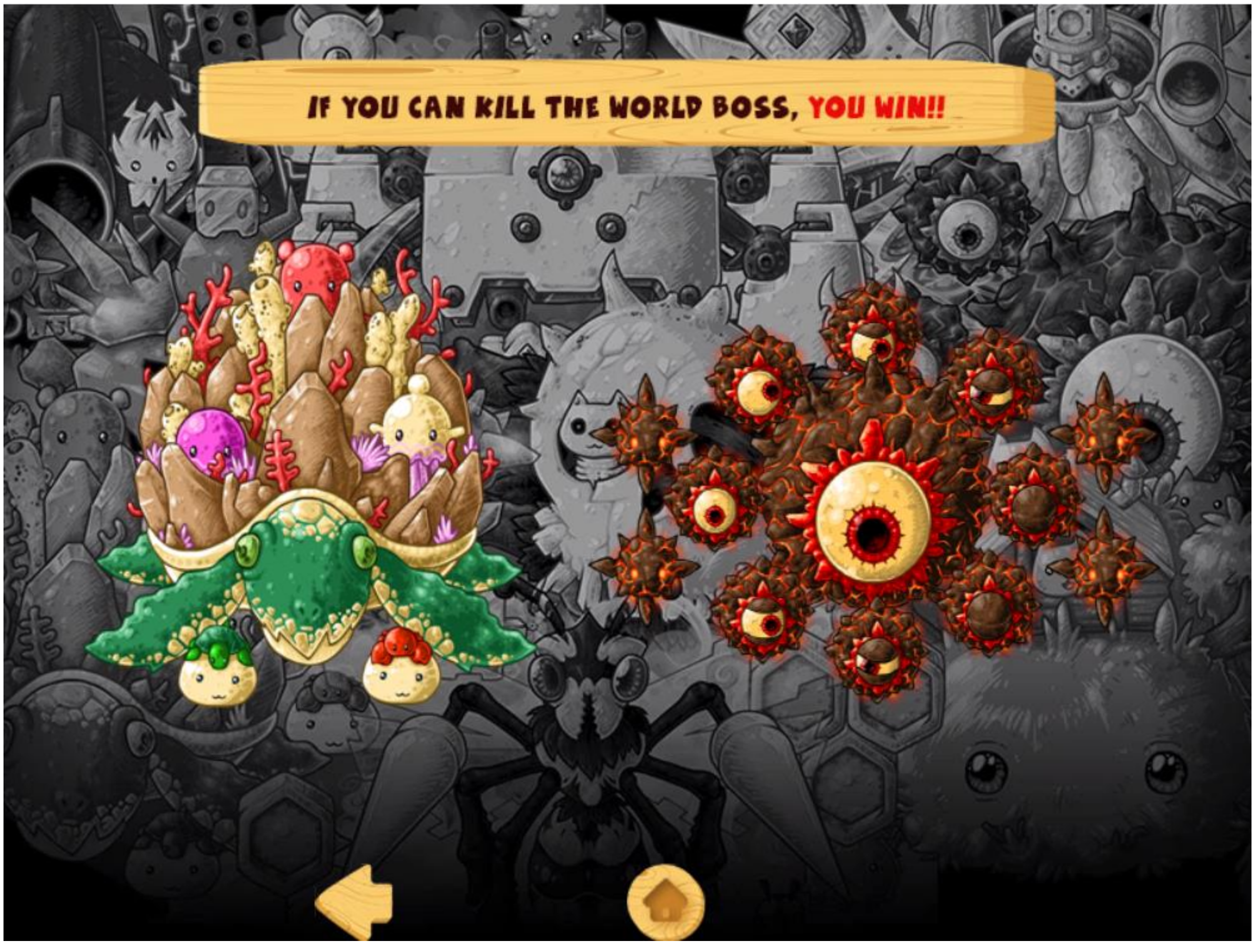


Figure 3.5 : How to play screen 5

When you click HOW TO PLAY:

- You will see an instruction of the game as shown in figure

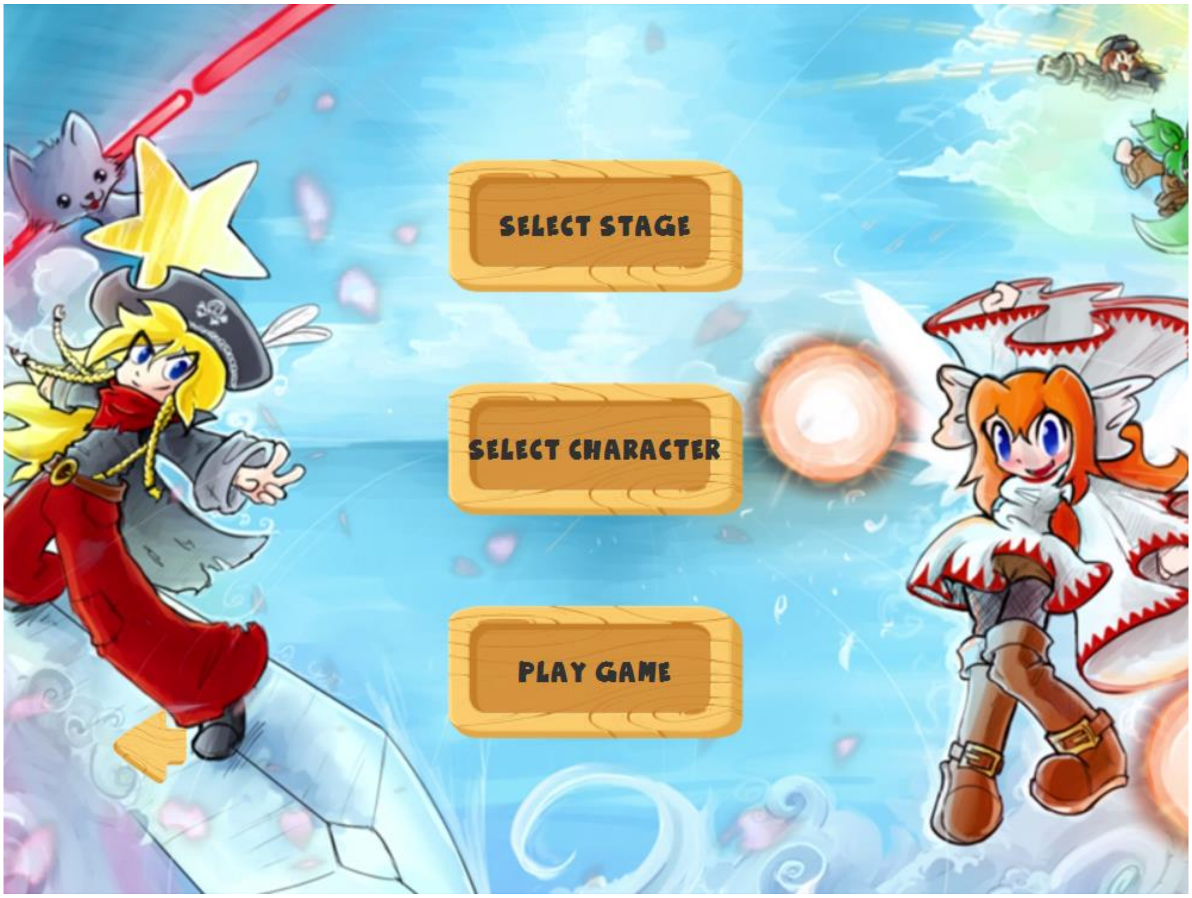


Figure 4 : Main screen

When you click START GAME:

- You will see 3 buttons: SELECT STAGE, SELECT CHARACTER, PLAY GAME in the screen as shown in figure
- You need to select stage and character before click play game.

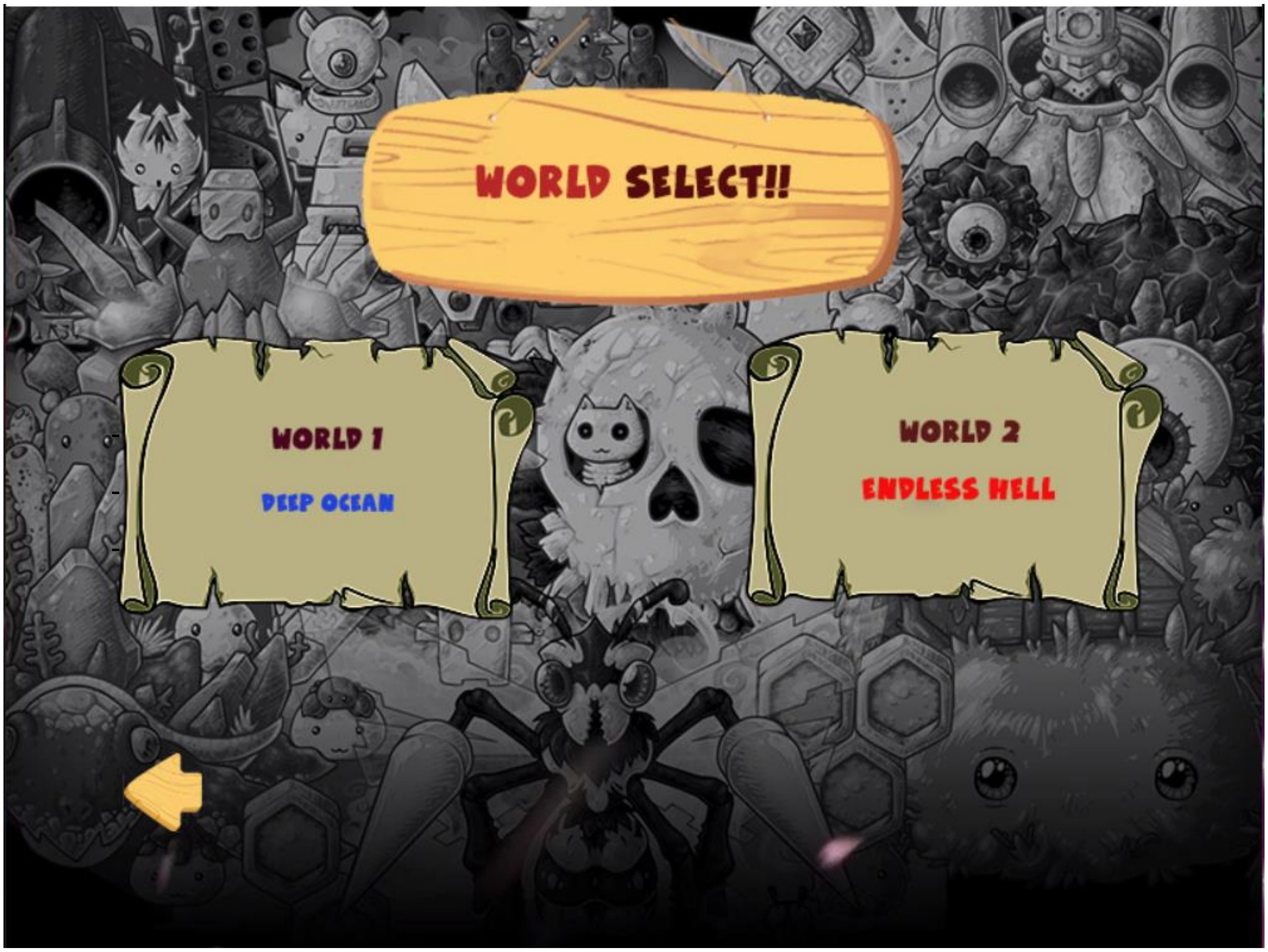


Figure 5 : World Select Screen

- When you click SELECT STAGE:

  There will be 2 stage that you can select: “Deep ocean” and “Endless hell”.

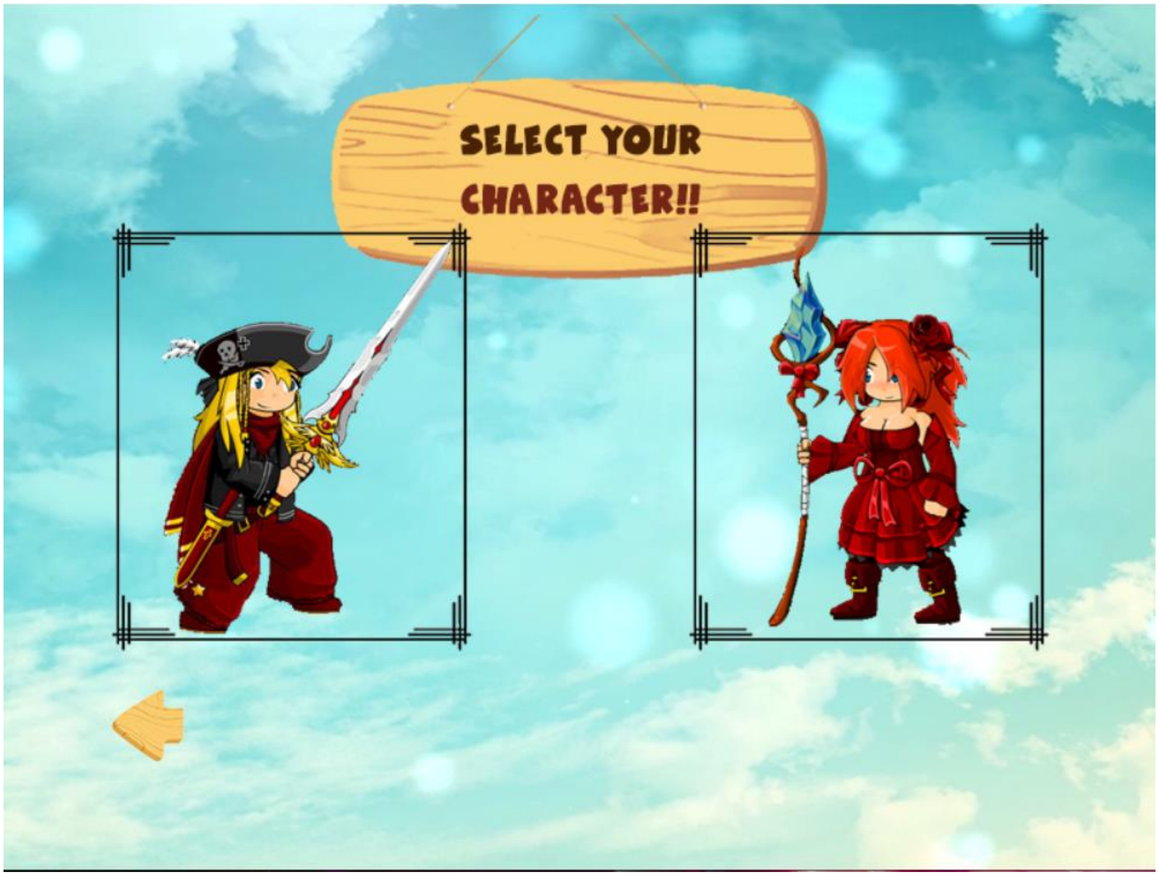


Figure 6 : Character select screen

- When you click SELECT CHARACTER:

  There will be 2 character that you can select: “Matt” and “Natalie”.

- “Matt” have more attack damage but less speed.
- “Natalie” have more speed but less attack damage.
- When you click PLAY GAME:

  game will start

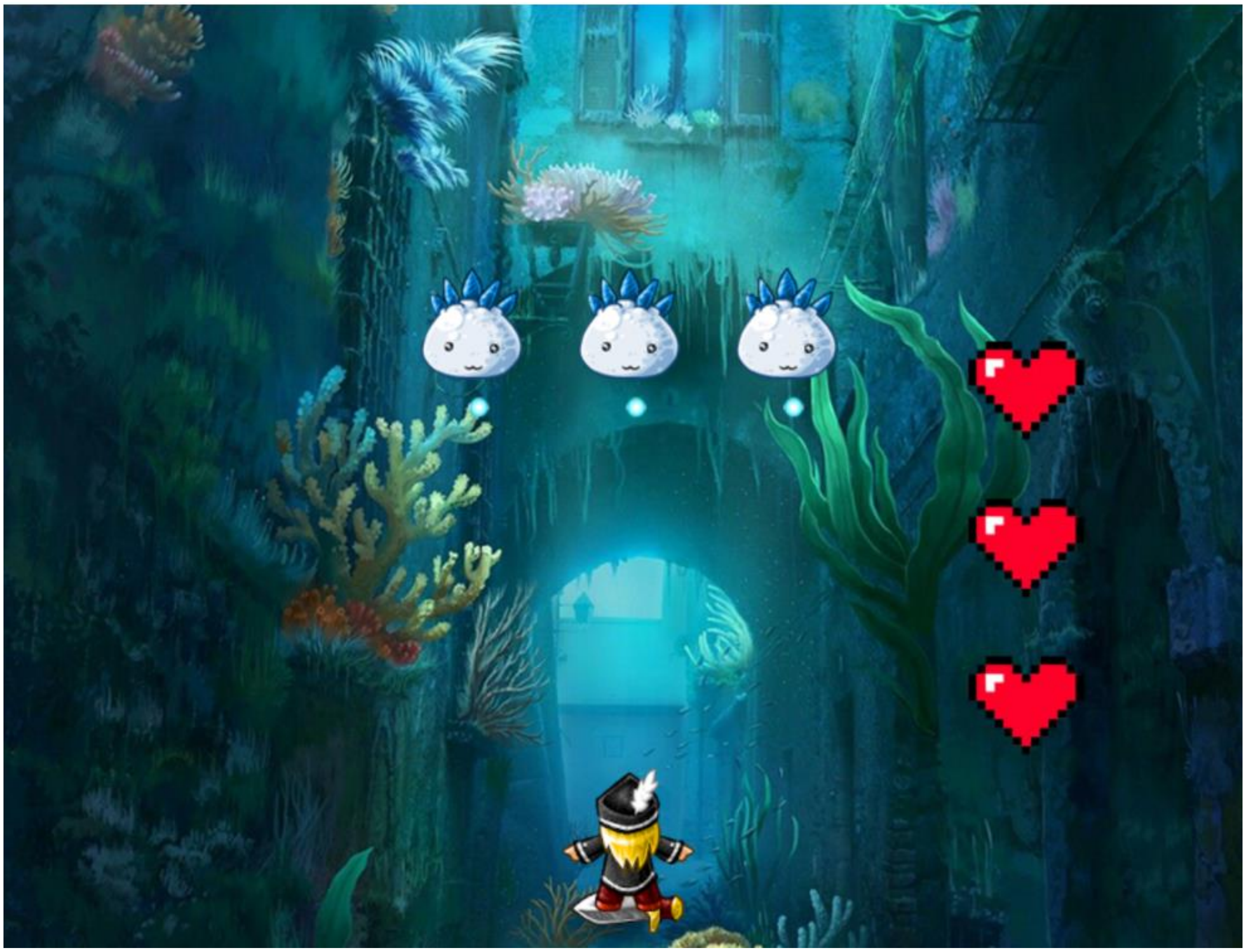


Figure 7 : Game Screen

Game Screen:

- In the game screen, you will see 3 spare heart that mean you have 4 life.
- If you have no more life, you lose and you will see a game over screen.
- Use W - move forward

  S - move backward

  A - move left

  D - move right

  Left mouse – Shoot
- The enemy will spawn and you need to kill them all by shooting.
- If the last boss of the world die, you win and you will see a game over screen.

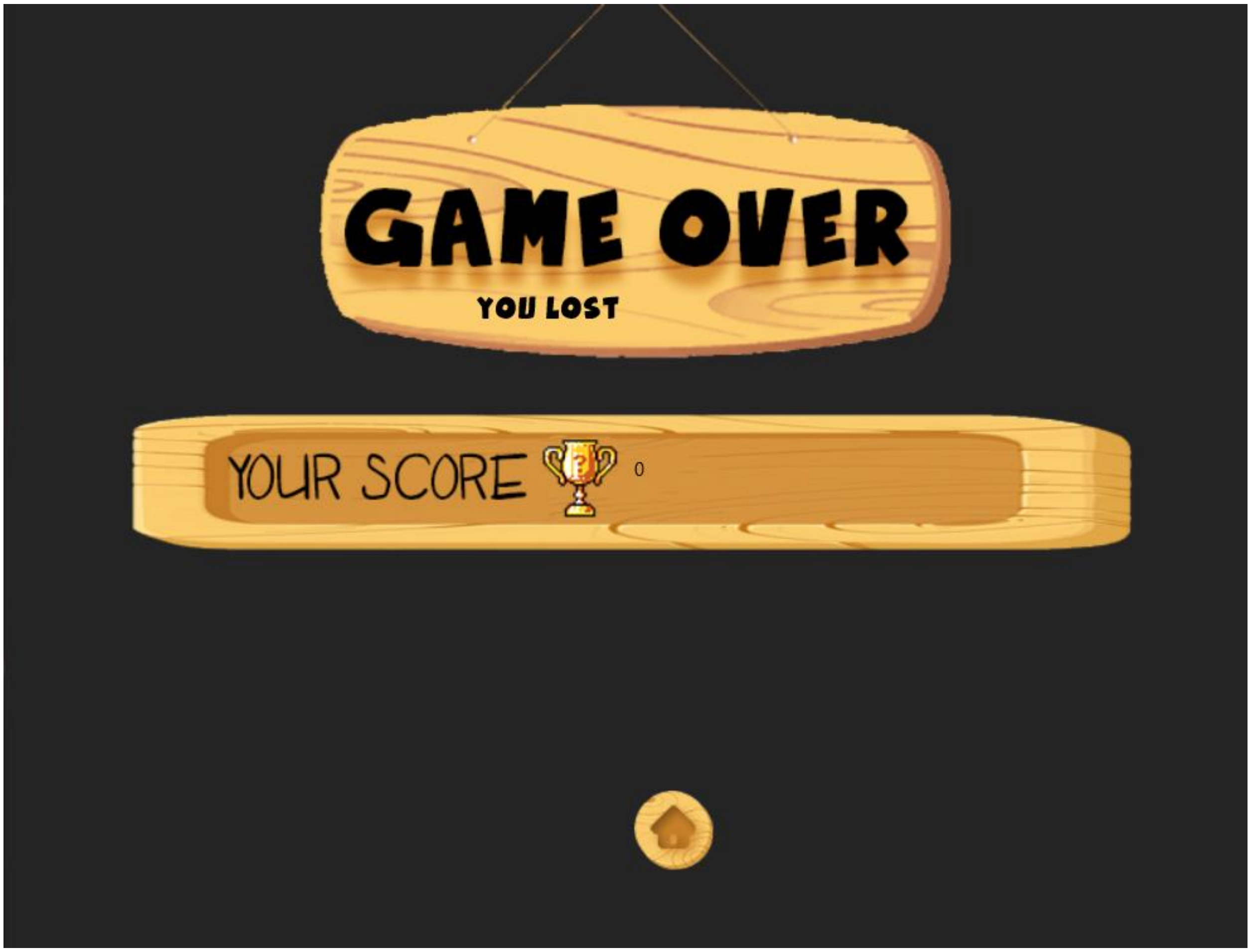


Figure 8 : Game Over Screen

Game Over Screen:

- You will see your score that came from you kill enemy.

2. Implementation Detail

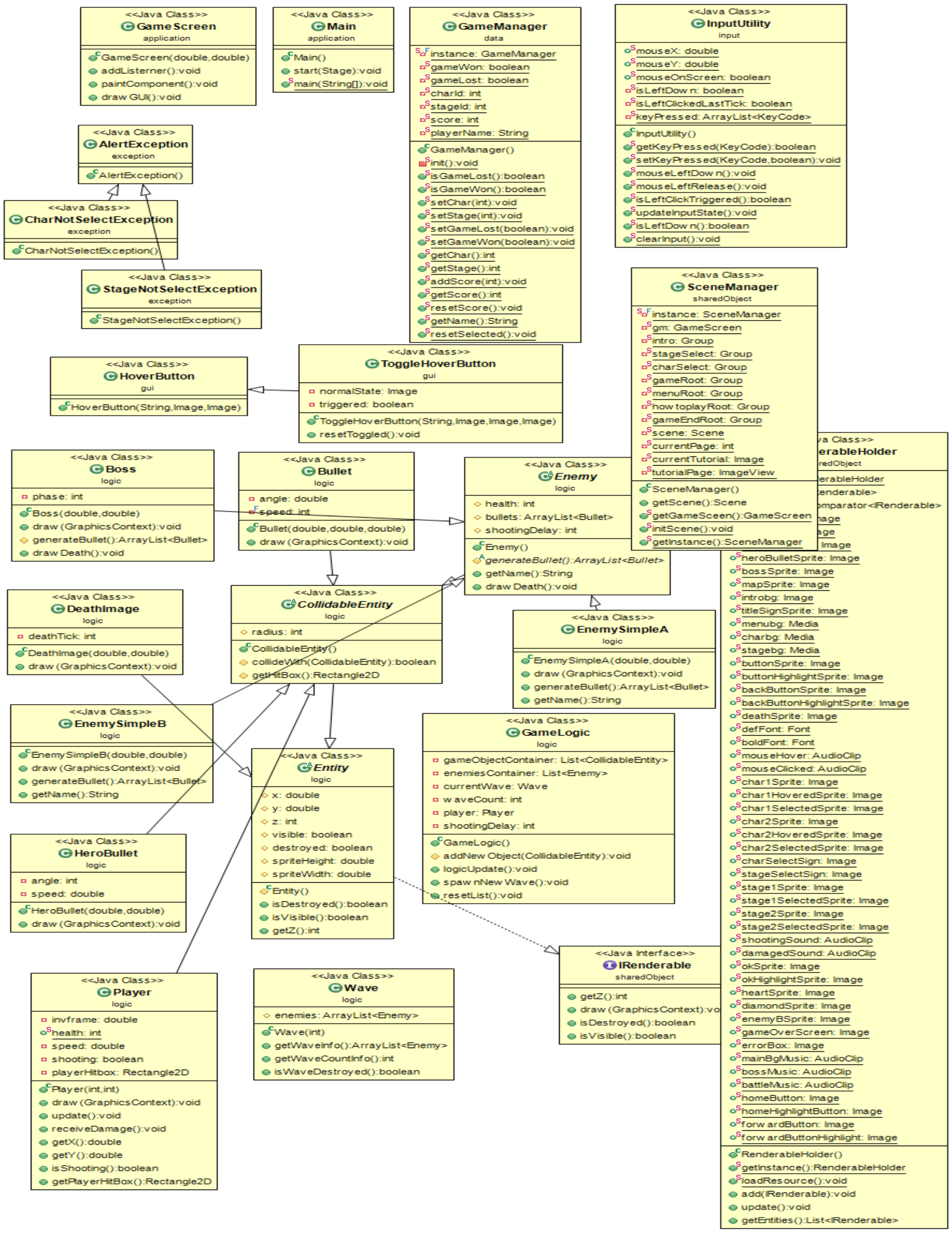


Figure 9 : UML Diagram of the application

Access Modifier Notations can be listed below

+ (public)

# (protected)

- (private)

2.1 Package application

2.1.1 class Main

2.1.1.1 Method

+ void start	The main entry point for the JavaFX applications.
+ void main	An entry point of the application.

2.1.2 class GameScreen extends Canvas

2.1.2.1 Constructor

+ GameScreen(double width, double height)	Initialize width and height of the canvas

2.1.2.2 Method

+ addListerner()	Get input form user
+ paintComponent()	Paint the component in Game screen
+ drawGui()	Draw heart sprite in Game screen

2.2 Package data

2.2.1 class GameManager

2.2.1.1 Field

- static final GameManager instance	Initialize the game manager
- static boolean gameWon	True if game won
- static boolean gameLost	True if game lost
- static int charId	ID of the character (Matt or Natalie)
- static int stageId	ID of the stage (Stage 1 or 2)
- static int score	Score when kill enemy
- static String playerName	Player name

2.2.1.2 Constructor

+ GameManager()	This class is not meant to be call

2.2.1.3 Method

- static void init()	Initialize gameWon , gameLost, charId, stageId, playerName, score
+ static Boolean isGameLost()	Return true if game lost , false otherwise
+ static Boolean isGameWon()	Return true if game win, false otherwise
+ static void setChar(int charid)	Setter for charId
+ static void setStage(int staged)	Setter for stageId
+ static void setGameLost(boolean b)	Setter for gameLost
+ static void setGameWon(boolean b)	Setter for gameWon
+ static int getChar()	Getter for charId
+ static int getStage()	Getter for stageId
+ static void addScore(int scoreAdd)	Add scoreAdd to score
+ static int getScore()	Getter for score
+ static void resetScore()	Reset the score to 0
+ static String getName()	Getter for playerName
+ static void resetSelected()	Reset the stageId and charId to 0

2.3 Package exception

2.3.1 class AlertException extends Exception

2.3.2 class CharNotSelectException extends AlertException

2.3.3 class StageNotSelectException extends AlertException

2.4 Package gui

2.4.1 class HoverButton extends Button

2.4.1.1 Constructor

+ HoverButton(String text, Image normal, Image hovered)	Create button with text that change sprite when mouse is hovered on it

2.4.2 class ToggleHoverButton extends HoverButton

2.4.2.1 Field

- Image normalState	Image button
- boolean triggered	Check if button is toggled

2.4.2.2 Constructor

+ ToggleHoverButton(String text, Image normal, Image hovered, Image toggled)	Create button with text that change sprite when mouse is hovered on it or it is clicked
+ void resetToggled()	Reset triggered

2.5 Package input

2.5.1 class InputUtility

2.5.1.1 Field

- static boolean isLeftDown	True if left mouse is hold down
- static ArrayList<KeyCode> keyPressed	List of keycode

2.5.1.2 Method

+ static boolean getKeyPressed(KeyCode keycode)	Return true if KeyCode is pressed
+ static void setKeyPressed(KeyCode keycode, boolean pressed)	check if keycode is being pressed and remove it from keyPressed if released
+ static void mouseLeftDown()	Set ifLeftDown = true, isLeftClickedLastTick = true
+ static void mouseLeftRelease()	Set ifLeftDown = false
+ static Boolean isLeftDown()	Getter for isLeftDown

2.6 Package logic

2.6.1 abstract class Entity implements IRenderable

2.6.1.1 Field

# double x,y	Position of entity

# int z	Position of entities depth relating to other entities
# boolean visible,destroyed	State of visibility and destruction of sprite
# double spriteHeight	Height of sprite
# double spriteWidth	Width of sprite

2.6.1.2 Constructor

# Entity()	This class is not meant to be called.

2.6.1.3 Method

+ boolean isDestroyed()	Getter for destroyed
+ boolean isVisible()	Getter for visible
+ int getZ()	Getter for z

2.6.2 abstract class CollidableEntity extends Entity

2.6.2.1 Method

# boolean collideWith(collidableEntity other)	Return true if this entity collide with other entity
# Rectangle2D getHitBox()	Return hitbox

2.6.3 abstract class Enemy extends CollidableEntity

2.6.3.1 Field

# int health	Enemy health
# ArrayList<Bullet> bullets	Arraylist of enemy bullet
# abstract ArrayList<Bullet> generateBullet()	Generate the enemy bullet
# int shootingDelay	Delay of enemy shooting

2.6.3.2 Method

+ String getName()	Getter for Enemy name
+ void drawDeath()	Render the death of enemy

2.6.4 class EnemySimpleA extends Enemy

2.6.4.1 Constructor

+ EnemySimpleA(double x, double y)	Initialize the detail of EnemySimpleA

2.6.4.2 Method

+ void draw(GraphicsContext gc)	Render the EnemySimpleA
+ ArrayList<Bullet> generateBullet()	Generate the EnemySimpleA bullet
+ public String getName()	Getter for EnemySimpleA name

2.6.5 class EnemySimpleB extends Enemy

2.6.5.1 Constructor

+ EnemySimpleB(double x, double y)	Set the detail of EnemySimple B

2.6.5.2 Method

+ void draw(GrapicsContext gc)	Render the EnemySimpleB
+ ArrayList<Bullet> generateBullet()	Generate the EnemySimpleB bullet
+ String getName()	Getter for EnemySimpleB name

2.6.6 class Boss extends Enemy

2.6.6.1 Constructor

+ Boss(double x, double y)	Initialize the detail of Boss

2.6.6.2 Method

+ void draw(GraphicsContext gc)	Render the boss
# ArrayList<Bullet> generateBullet()	Generate the boss bullet
+ void drawDeath()	Render the death of boss

2.6.7 class DeathImage extends Entity

2.6.7.1 Field

- int deathTick	Amount of frame deathImage last

2.6.7.2 Constuctor

+ DeathImage(double x, double y)	Initialize position of death image

2.6.7.3 Method

+ void draw(GraphicsContext gc)	Render the death image

2.6.8 class Bullet extends CollidableEntity

2.6.8.1 Field

- double angle	Bullet angle when shoot
- final int speed	Bullet speed

2.6.8.2 Constructor

+ Bullet(double x, double y,double angle)	Initialize Bullet position, radius, angle, width, height

2.6.8.3 Method

+ void draw(GraphicsContext gc)	Render the Bullet

2.6.9 class HeroBullet extends CollidableEntity

2.6.9.1 Field

- int angle	Hero bullet angle when shoot
- double speed	Hero bullet speed

2.6.9.2 Constructor

+ HeroBullet(double x, double y)	Initialize Hero bullet position, sound, radius, width, height

2.6.9.3 Method

+ void draw(GraphicsContext gc)	Render the hero bullet

2.6.10 class Player extends CollidableEntity

2.6.10.1 Field

- double invframe	Invisible frame
+ static int health	Player health
- double speed	Player speed
- int damage	Player damage
- boolean shooting	Is player shooting
- Rectangle2D playerHitbox	Player hit box

2.6.10.2 Constructor

+ Player(int x, int y)	Initialize Player position, radius, width, height, hit box

2.6.10.3 Method

+ void draw(GraphicsContext gc)	Render the player
+ void update()	update player object with input and update invframes
+ void recieveDamage()	Render invisible frame when player is shot
+ double getX()	Getter for X
+ double getY()	Getter for Y
- int getDamage()	Getter for damage
+ boolean isShooting()	Getter for shooting boolean

+ Rectangle2D getPlayerHitBox()	Getter for player hit box

2.6.11 class Wave

2.6.11.1 Field

# ArrayList<Enemy> enemies	List of enemy

2.6.11.2 Constructor

+ Wave(int waveCount)	Create wave according to wave id

2.6.11.3 Method

+ ArrayList<Enemy> getWaveInfo()	Return list of enemies in this wave
+ int getWaveCountInfo()	Return number of enemies in this wave
+ boolean isWaveDestroyed()	True if enemies in this wave is all dead

2.6.12 class GameLogic

2.6.12.1 Field

- List<CollidableEntity> gameObjectContainer	List of gameObject
- List<Enemy> enemiesContainer	List of enemy
- Wave currentWave	Current wave
- int waveCount	Wave id
- Player player	Player object
- int shootingDelay	Shooting delay for player

2.6.12.2 Constructor

+ GameLogic()	Initialize the GameLogic

2.6.12.3 Method

# void addNewObject(CollidableEntity entity)	Add new object to List of gameObject
+ void logicUpdate()	Create bullet for hero and enemies and check and resolve collision of objects
+ void spawnNewWave()	Spawn new wave when previous wave is destroyed
+ void resetList()	Reset gamelogic list when start a new game

2.7 Package sharedObject

2.7.1 interface IRenderable

2.7.1.1 Method

+ int getZ()	Getter for z
+ void draw(GraphicsContext gc)	Render the sprite of object
+ boolean isDestroyed()	Getter for destroyed
+ boolean isVisible()	Getter for visible

2.7.2 class RenderableHolder

2.7.2.1 Field

- List<IRenderable> entities	List of entities to draw
- Comparator<IRenderable> comparator	Comparator for sort z function
+ static Image playerSprite	Image player
+ static Image bulletSprite	Image bullet
+ static Image enemyASprite	Image enemy A
+ static Image heroBulletSpite	Image player bullet
+ static Image bossSprite	Image boss
+ static Image mapSprite	Image map background
+ static Image introbg	Image intro background
+ static Image titleSignSprite	Image title sign
+ static Media menubg	Video menu background
+ static Media charbg	Video character select background
+ static Media stagebg	Video Stage background

+ static Image buttonSprite	Image button
+ static Image buttonHighlightSpite	Image highlight button
+ static Image backButtonSprite	Image back button
+ static Image backButtonHighlightSprite	Image highlight back button
+ static Image deathSprite	Image when enemy death
+ static Font defFont	Font def
+ static Font boldFont	Font bold
+ static AudioClip mouseHover	Sound when mouse on
+ static AudioClip mouseClicked	Sound when mouse click
+ static Image char1Sprite	Image character 1
+ static Image char1HoveredSprite	Image when mouse on character 1
+ static Image char1SelectedSprite	Image when select character 1
+ static Image char2Sprite	Image character 2
+ static Image char2HoveredSprite	Image when mouse on character 2
+ static Image char2SelectedSprite	Image when select character 2
+ static Image charSelectSign	Image character select sign
+ static Image stageSelectSign	Image stage select sign
+ static Image stage1Sprite	Image stage 1
+ static Image stage1SelectedSprite	Image when select stage 1
+ static Image stage2Sprite	Image stage 2
+ static Image stage2SelectedSprite	Image when select stage 2
+ static AudioClip shootingSound	Sound when shoot
+ static AudioClip damageSound	Sound when receive damage
+ static Image okSprite	Image OK button
+ static Image okHighlightSprite	Image highlight OK button
+ static Image heartSprite	Image heart
+ static Image enemyBSprite	Image enemy B
+ static Image gameOverScreen	Image gameOverScreen
+ static Image errorBox	Image error box

+ static AudioClip mainBgMusic	Sound main menu
+ static AudioClip bossMusic	Sound when boss come
+ static AudioClip battleMusic	Sound when battle
+ static Image homeButton	Image home button
+ static Image homeHighlightButton	Image highlight home button

2.7.2.2 Constructor

+ RenderableHolder()	Initialize the entity

2.7.2.3 Method

+ static RenderableHolder getInstance()	Return singleton approach of RenderableHolder class
+ static void loadResource()	Classloader all of the resources
+ void add(IRenerable entity)	Add entity to list of entities
+ void update()	Check if entity in entities list is destroyed and remove it
+ List<IRenderable> getEntities()	Getter for entities

2.7.3 class SceneManager

2.7.3.1 Field

- static final SceneManager instance	Return singleton approach of SceneManager class
- static GameScreen gm	Root of game screen
- static Group intro	Root of intro scene
- static Group stageSelect	Root of stage select scene
- static Group charSelect	Root of character select scene
- static Group gameRoot	Root of game scene
- static Group menuRoot	Root of menu scene
- static Group howtoplayRoot	Root of how to play scene
- static Group gameEndRoot	Root of game over scene

- static Scene scene	Scene

2.7.3.2 Constructor

+ SceneManager()	Initialize the game screen

2.7.3.3 Method

+ Scene getScene()	Getter for scene
+ static GameScreen getGameScreen	Getter for game screen
+ static void initScene()	Initialize every root
+ static SceneManager getInstance()	Return instance