The Legend slayer

1) Introduction

In this game , you are playing as a brave soldier whose duty is to save the world from the monsters that you know it as well from legendary fairytale . you need to kill the monsters as many as you can . Good luck soldier !!!

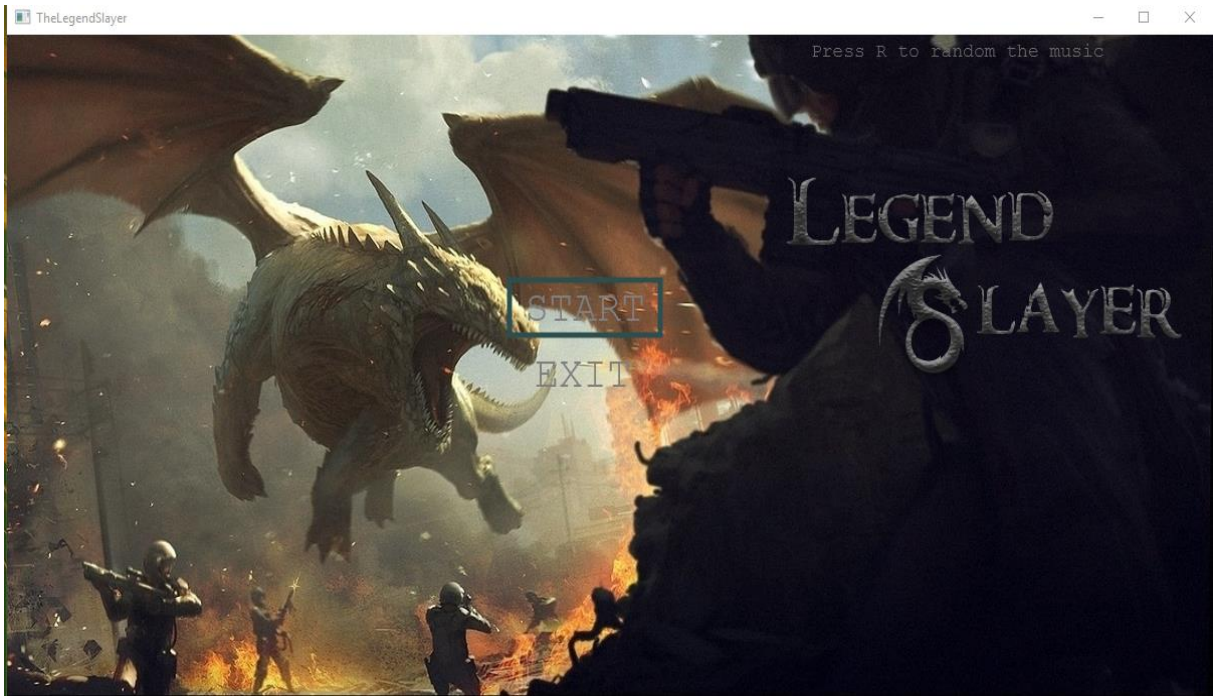
2) Game Control

At the StartWindow

Figure1

Use Spacebar to interact with the window. At Start, press Spacebar to enter the main menu of the game. After that, you can choose to start game or exit by pressing the arrow keys up and down.

The instruction that was choose will be surrounded by a black stroke rectangle. Also you can press R to change music



.

figure 2

On the Game Window :

You were born at the castle(the left bottom of the map) and the monster will be generate at the forest (the right top of the map)

Use the arrow keys (←↑→↓) to control the direction of the character that you play. Press spacebar to fire. Let kill the monster !!!


The Legend Slayer
score:0
life:10


The Legend Slayer
score:60
life:10

figure 3 & 4

you will get score when you kill the monster and the monster has a several species . There are slime , zombie , unicorn , god and Dragon ordering by the level of monster . and the score that you get from killing the monster is also up to the level of the monster that you kill . for example , when you kill the slime , you will get 10 score . and when you kill the Dragon you will get 500 score .

you has 10 life as the initial lifepoint , if you die 10 time ,the game will be over!!! . so , you should kill the monster as long as you are alive to get more and more score .

the monster will be generated itself automatically , and the rate of generating monsters is always increase .

figure 5

When the character is attacked by the monster, the life will decrease by 1. If the life become zero, game over. The“GameOver” word will be shown on the screen and then the game will show the Game Over window with the score result and you can press space to go back to main start window.

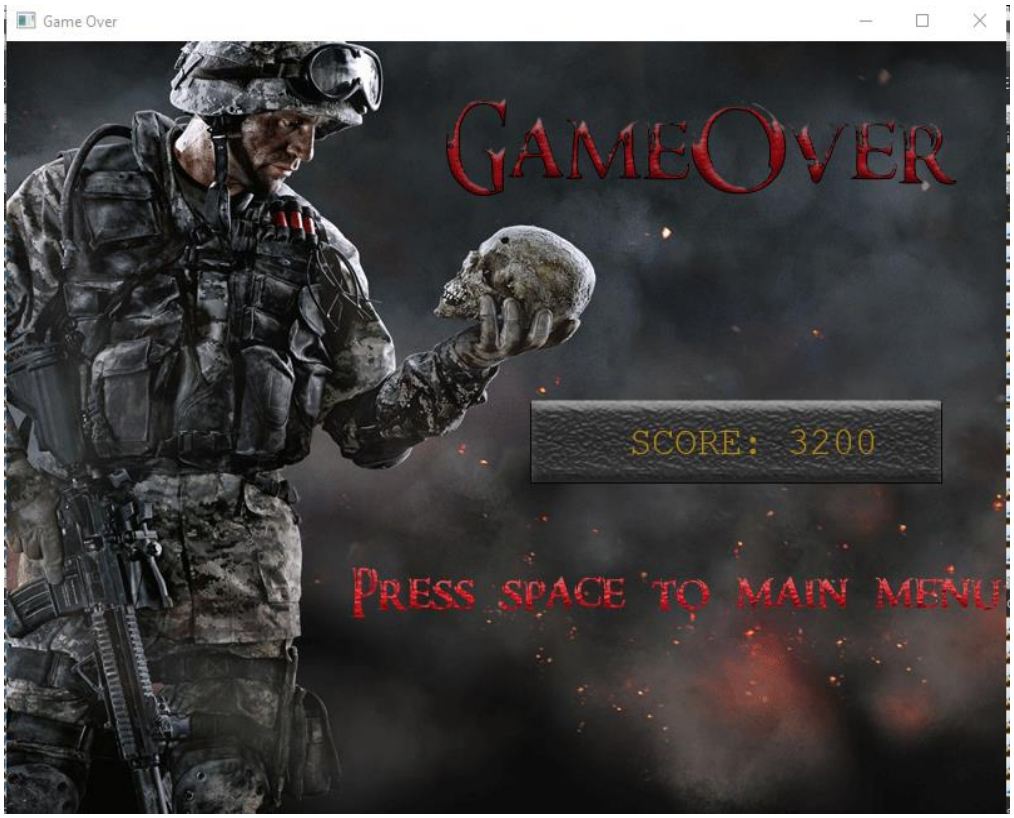


figure 6

3. Game Controls

Press ← to move left

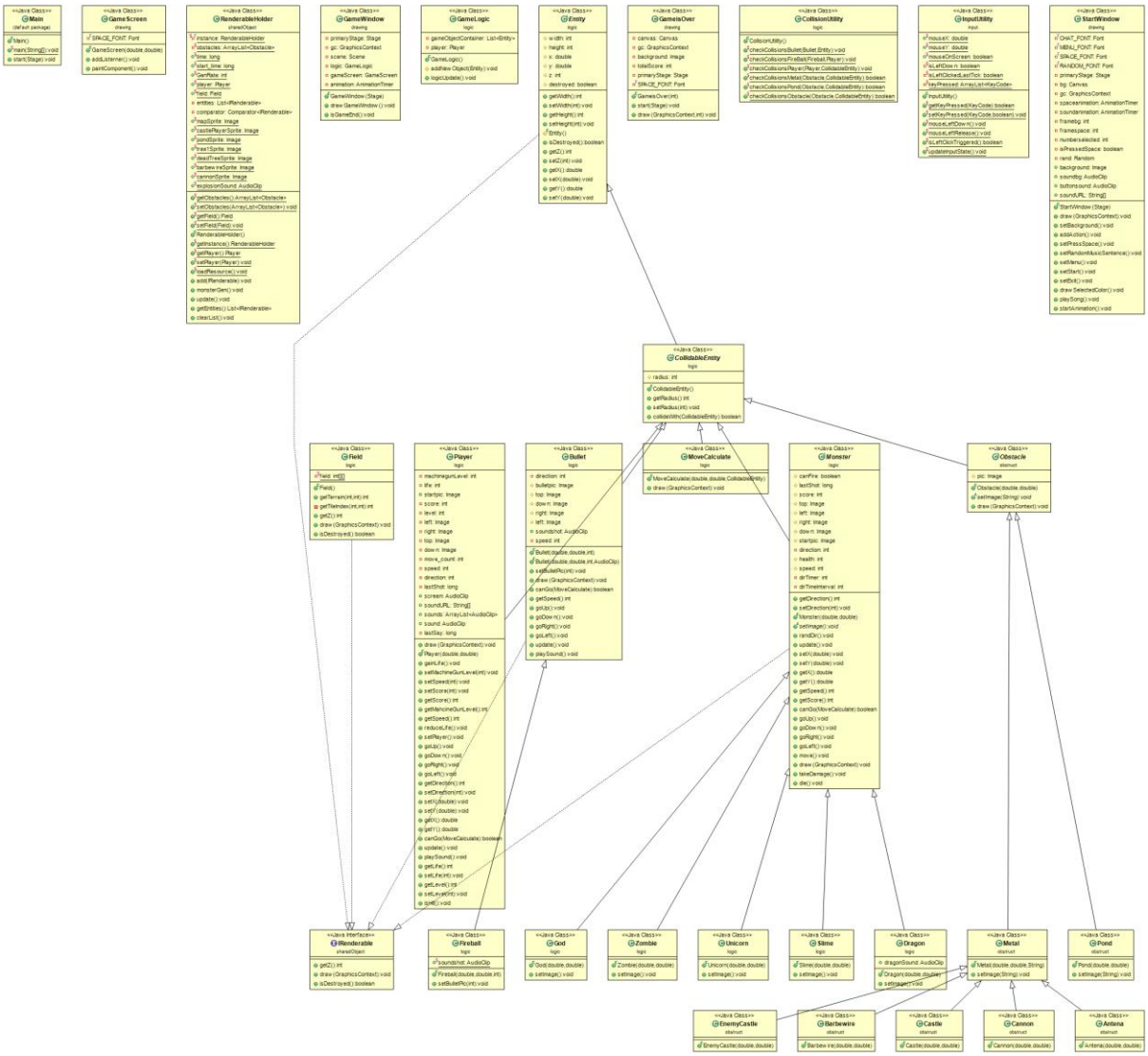
Press ↑ to move up

Press ↓ to move down

Press → to move right

Press spacebar to fire – the bullets will be fired on the current direction of the player

UML Diagram



can read more at UML file

5. Class Details - Methods - Constructor -Fields

5.1 Package application

5.1.1 Class Main extends Application

5.1.1.1 Method

+ void start(Stage primaryStage)	The main entry point for the JavaFX applications
+ void main(String[] args)	An entry point of the application

5.2 Package drawing

5.2.1 Class StartWindow

5.2.1.1 Field

+ Font MENU_FONT	Font for Menu's Name
+ Font SPACE_FONT	Font for the sentence "press Spacebar to Enter the Main Menu" at the first screen that shows when we call the StartWindow
+ Font RANDOM_FONT	Font for the string "Press R to random the music " on the top-right of screen
- Stage primaryStage	Stage from Main got from constructor
- Canvas bg	canvas of this screen
- GraphicsContext gc	GraphicsContext from the bg
- AnimationTimer spaceanimation	animation on the start that will wink the

	sentence “Press spacebar to enter the Main Menu”
- AnimationTimer soundanimation	animation for playing the song in the loop when song stops or finishes from playing, it will call method playSong to continue the song
- int framebg	for the effect wink of the string “Press spacebar to enter the Main Menu”. it is used in the method draw
- int framespace	for the effect wink of the string “Press spacebar to enter the Main Menu”. it is used in the method draw
- int numberselected	for the method drawSelectedColor about which menu should have the grey rectangle around it
- boolean isPressedSpace	true if the player press spacebar so it will setMainmenu and when the player press spacebar again, it will
- Random rand	For randoming for the number to choose which song will play
+ Image background	background of the screen
+ AudioClip soundbg	sound of the bg that will be initialize in the constructor

+ AudioClip buttonsound	sound when you press UP or DOWN key when selected on the main menu (choose between “START” and “EXIT”). it’s initialized here
+ String[] soundURL	URL of all the songs that will play in this screen

5.2.1.2 Constructor

+ StartWindow(Stage primaryStage)	Initialize rand for the number of the song that will play on the screen and Initialize canvas for bg and gc. Also Initialize soundbg with the number from the rand and then play it

5.2.1.3

+ void draw(GraphicsContext gc)	starts the animation for the background and effect including the song to play
+ void setBackground()	set the background image on the canvas and the title of the game
+ addAction()	add the KeyEvent on this screen

+ void setPressSpace()	When this window starts the sentence “ Press spacebar to enter the main menu” will be on the right-side of the screen with the grey color
+ void setRandomMusicSentence()	set the sentence “Press R to random the music” on the canvas at the top-right of the screen
+ void setMenu()	call the method setStart and setExit to set the menu on the right-side of the screen
+ void setStart()	set the text “START” on the center of the screen and it’s under the text “EXIT”
+ void setExit()	Set the text “EXIT” in the screen and it’s under the text “START”
+ void drawSelectedColor()	Draw the grey rectangle on the canvas around the “EXIT” or “START” when it’s selected
+ void playSong()	Random the song to play on this screen
+ void startAnimation()	Call the method draw to start the animation

5.2.2 Class GameWindow

5.2.2.1 Field

- Stage primaryStage	to store the stage for when game is over, we can use the stage change to GameisOver page
- GraphicsContext gc	for initializing to draw every compenents on game screen
- Scene scene	to set the current scene to the current stage
- InitializeGame initialize	to create obstacle
- GameScreen gameScreen	to be the area for drawing every components
- AnimationTimer animation	be the thread for drawing every components

5.2.2.2 Constructor

+ GameWindow(Stage primaryStage)	using StackPane and getting canvas into it including setting width and height at the proper size

5.2.2.3 Method

+ void drawGameWindow()	using AnimationTimer Thread for updating every component in RenderableHolder,any input state , drawing every components and checking whether game is end or it
+ void isGameEnd()	check that if player has life less than 0 , GameWindow will be closed and GameisOver will be opened together with display the total score of the player

5.2.3 Class GameScreen

5.2.3.1 Field

- Font SPACE_FONT	Font for the sentence “Press Spacebar to Enter the Main Menu” at the first screen that shows when we call the StartWindow

5.2.3.2 Constructor

+ GameScreen(double width, double height)	extending from Canvas and set preference width and height

5.2.3.3 Method

- void addListener()	to pass the keycode and mouse action to InputUtility in input package and can change coordinate of the player to the position the mouse clicked
+ void paintComponent()	iterate the instance in RenderableHolder for checking whether the component is destroyed or not ,if it's not destroyed.it will be drawn on the canvas and display the current life and score.

5.2.4 Class GameisOver

5.2.4.1 Field

- Canvas canvas	the area to draw gameover background,it will emerge total score
- GraphicsContext gc	the tool for drawing gameover background and total score of the player
- Image background	the background of GameisOver class
- int totalScore	to store the total score when the player die

- Stage primaryStage	to store the stage of GameWindow to change to stage of GameisOver

- Font SPACE_FONT	Font for the sentence “Press spacebar to Main Menu” at the last screen that shows when we call the StartWindow

5.2.4.2 Method

+ void start(Stage primaryStage)	using StackPane and adding Canvas into it for drawing the total score,setting the backdround game and set action that if spacebar is pressed , GameisOver page is closed and open StartWindow page
+ void draw(GraphicsContext gc,int totalScore)	receive gc to draw the background game at the canvas and get total score for drawing the value at the last screen of the game

5.3 Package input

5.3.1 Class InputUtility

5.3.1.1 Field

+ double mouseX	the value of x-axis at the coordinate of mouse
+ double mouseY	the value of y-axis at the coordinate of mouse
+ boolean isLeftDown	check whether the left-side of mouse is clicked or not
+ boolean isLeftClickedLastTick	check that the left-clicked of mouse is the last tick
+ ArrayList<KeyCode> keyPressed	to store all of the key pressed by order

5.3.1.2 Method

+ boolean getKeyPressed(KeyCode keycode)	check whether parameter keycode is contained in keyPressed or not
+ void setKeyPressed(KeyCode keycode, boolean pressed)	if key is pressed is true and keyPressed list is not contain the keycode,the keyPressed list will add it but if not ,the keyPressed list will remove it and both of the actions will print out the parameter keycode on the console

+ void mouseLeftDown()	adjust the value of isLeftDown and isLeftClickedLastTick is true.(when mouse is left-clicked, this method will be called)
+ void mouseLeftRelease()	change to the value of isLeft to false (when mouse is released from left-clicking,the method will be called)
+ boolean isLeftClickTriggered()	return the value of isLeftClickedLastTick
+ void updateInputState	change the value of isLeftClickedLastTick to false

5.4 Package logic

5.4.1 Class Bullet

5.4.1.1 Field

- int direction	the direction of bullet by 0 is top, 1 is right , 2 is down and 3 is left.
# Image bulletpic	the current picture of bullet
# Image top	the top of the image
# Image down	the down of the image
# Image right	the right of the image
# Image left	the left of the image
+ AudioClip soundshot	the sound when the bullet is fired

+ int speed	the speed of the bullet

5.4.1.2 Constructor

+ Bullet(double x, double y, int direction)	set the coordinate,direction,radius and playing sound of the bullet
+ Bullet(double x, double y, int direction, AudioClip soundshot)	set the coordinate,direction,radius,sound and playing sound of the bullet

5.4.1.3 Method

+ void setBulletPic(int direction)	set all of the images of the bullet
+ void draw(GraphicsContext gc)	draw the current image of the bullet
- boolean canGo(MoveCalculate future)	check whether the monster collide with something or not ,check whether it collide with the thing allow the monster to pass it or not
+ int getSpeed()	get the value of speed of the bullet
+ void goUp()	create MoveCalculate class by setting that it's the next move of the bullet if can go , it will reduce the value of y and set the current image is top
+ void goDown()	create MoveCalculate class by setting that it's the next move of the bullet if can

	go , it will increase the value of y and set the current image is down
+ void goRight()	create MoveCalculate class by setting that it's the next move of the bullet if can go , it will increase the value of x and set the current image is right
+ void goLeft()	create MoveCalculate class by setting that it's the next move of the bullet if can go , it will reduce the value of x and set the current image is left
+ void update()	update moving followed by the direction

+ void playSound()	play sound of releasing the bullet

5.4.2 Class CollidableEntity

5.4.2.1 Field

# int radius	using for calculating collision

5.4.2.2 Method

+ int getRadius()	get the value of radius
+ void setRadius(int radius)	set the value of radius

+ boolean collideWith(CollidableEntity)	check whether two objects collide with together or not

5.4.3 Class CollisionUtility

5.4.3.1 Method

+ void checkCollisionsBullet(Bullet bullet, Entity other)	check the object.if the object is Monster ,its health is reduced and destroy the bullet
+ void checkCollisionsFireBall(Fireball fireball, Player player)	check the object if the object is Player, the player will be hit
+ void checkCollisionsPlayer(Player player, CollidableEntity other)	check the object.If it's a monster,the player will be hit
+ boolean checkCollisionsMetal(Obstacle obstacle, CollidableEntity other)	check the object.If it's a player,monster or bullet will return true otherwise false
+ boolean checkCollisionsPond(Obstacle obstacle,CollidableEntity other)	check the object.If it's a player or monster will return true otherwise false
+ void checkCollisionsObstacle(Obstacle obstacle, CollidableEntity other)	check collision between the object and obstacle by using checkCollisionsMetal and checkCollisionsPond

5.4.4 Entity

5.4.4.1 Field

# int width	the width of the image
# int height	the height of the image
# double x	the value of x-axis of the coordinate
# double y	the value of y-axis of the coordinate
# int z	the status that which draws before
# boolean destroyed	the status told whether the object is destroyed or not and there is the default value being false

5.4.4.2 Method

+ int getWidth()	get the width of the image
+ void setWidth(int width)	set the value of the width
+ int getHeight()	get the height of the image
+ void setHeight()	set the value of the height
+ boolean isDestroyed()	get the value of destroyed
+ int getZ()	get the value of z
+ void setZ(int z)	set the value of z
+ double getX()	get the value of x

+ void setX(double x)	set the value of x
+ double getY()	get the value of y
+ void setY(double y)	set the value of y

5.4.5 Class Field

5.4.5.1 Field

- int[][] field	array of integers.There are 3 different values in it.0 means transparent, 1 means deadtree and 2 means tree

5.4.5.2 Method

+ int getTerrian(int x,int y)	to call the value of field the column yth and row xth
- int getTileIndex(int x,int y)	find the part of image to crop it
+ int getZ()	set it to -9999 for draw it first
+ void draw(GraphicsContext gc)	draw the background and draw dead tree and tree followed by the field variable
+ boolean isDestroyed()	return the status whether the field is dedstroyed or not

5.4.6 Class InitializeGame

5.4.6.1 Constructor

+ InitializeGame()	initialize every obstacles , player,field and some monster.The obstacle have Antena, Pond, Cannon, Castle, EnemyCastle, BarbeWire.

5.4.7 class MoveCalculate

+ MoveCalculate(double x, double y, CollidableEntity entity)	set the position of the CollidableEntity object and calculate the next move (use for checking collision)

5.4.8 abstract class Monster

5.4.8.1 constructor

Monster(double x, double y)	set the position of the monster

5.4.8.2 field

# boolean canFire	is the monster can fire ?
# long lastShot	keep the last time that monster fire
# int score	score of each monsters that player will get if that monster were kill

# Image top	top image of the monster
# Image left	left image of the monster
# Image right	right image of the monster
# Image down	down image of the monster
# Image startpic	start image of the monster
- int direction	direction of the monster 0,1,2,3 stand for top,right,down,left by ordering

# int health	health of the monster
# int speed	speed of the monster

- int dirTimer	generalize timer for the random move
- dirTimeInterval	delay rate of the random move

5.4.8.3 method

+ int getDirection()	get the value of the direction
+ setDirection(int direction)	set the value of the direction
+ void setImage()	set all image direction of the monster
+ void randDir()	random the direction

+ setX(double total)	set position x (inside the map)
+ setY(double total)	set position y (inside the map)
+ double getX()	get the value of x
+ double getY()	get the vale of y
+ getSpeed()	get the value of the speed
+ getScore()	get the value of the score
- boolean canGo	check the collision of the monster if the monster collide with the obstacle monster will not move to that direction anymore
+ void goUp()	move to the up direction if can go
+ void goDown()	move to the down direction if can go
+ void goRight()	move to the right direction if can go
+ void goLeft()	move to the left direction if can go
+ void move()	use randDir() to random new direction the move
+ void draw(GraphicsContext gc)	draw the monster graphic
+ void takeDamage()	decrease health “1” and if health < 0 monster will be die
+ void die()	set” destroyed” of the monster to be true

+ void update()	update the random the move direction of the monster and if the monster can fire (specific Dragon) it will also random to direction to fire

5.4.9 Class Slime

+ Slime(double x, double y)	set the position and then automatically set width , height , health , speed , radius , score and image in unique value for the slime
+ void setImage()	set all of image direction

5.4.10 Class Zombie

+ Zombie(double x, double y)	set the position and then automatically set width , height , health , speed , radius , score and image in unique value for the zombie
+ void setImage()	set all of image direction

5.4.11 Class Unicorn

+ Unicorn(double x, double y)	set the position and then automatically

	set width , height , health , speed , radius , score and image in unique value for the unicorn
+ void setImage()	set all of image direction

5.4.12 Class God

+ God(double x, double y)	set the position and then automatically set width , height , health , speed , radius , score and image in unique value for the god
+ void setImage()	set all of image direction

5.4.13 Class Dragon

+ Dragon(double x, double y)	set the position and then automatically set width , height , health , speed , radius , score and image in unique value for the Dragon set Dragon to be fire able (set canFire) and then play the Dragon Sound
+ void setImage()	set all of image direction

5.4.13 Class Fireball

+ Fireball(double x, double y, int direction)	set the fireball position and direction and then set the fireball sound automatically
+ void setBulletPic()	set all of image direction

5.5 package obstruct

5.5.1 abstract class obstacle

5.5.1.1 field

# Image pic	image of obstruct objects (will be drawn)

5.5.1.2 constructor

+ Obstacle(double x,double y)	the constructor of the obstacle and set x , y (will be drawn at that position)

5.5.1.3 method

+ setImage(String img)	set the image (by classLoader)
+ drawImage(GraphicsContext gc)	draw the obstacle object

5.5.2 Class Metal

+ Metal(double x, double y,String img)	set the position x,y and and setImage(img)

5.5.3 Class Antena

+ Antena(double x, double y)	set the position x,y and set the antena and then set the unique width , height and the radius

5.5.4 Class Barbewire

+ Barbewire(double x, double y)	set the position x,y and set the Barbewire and then set the unique width , height and the radius

5.5.5 Class Cannon

+ Cannon(double x, double y)	set the position x,y and set the Cannon and then set the unique width , height and the radius

5.5.6 Class EnemyCastle

+ EnemyCastle(double x, double y)	set the position x,y and set the Enemy Castle and then set the unique width , height and the radius

5.5.7 Class Castle

+ Castle(double x, double y)	set the position x,y and set the Castle and then set the unique width , height and the radius

5.5.8 Class Pond

+ Pond(double x, double y)	set the position x,y and set the Pond and then set the unique width , height and the radius

5.6 Package shareObject

5.6.1 Interface IRenderable

5.6.1.1 Method

+ int getZ();	get z
+ void draw (GraphicsContext gc);	draw graphic
+boolean isDestroyed()	check destroy?

5.6.2 class RenderableHolder

5.6.2.1 Field

- RenderableHolder Logic instance	instance of the class
- ArrayList<Obstacle> obstacles	list of Obstacle object
- long time	keep the time of monster generator (use in monsterGen() method
- long start time	keep the start game time
- int GenRate	use to increase monsterGen() rate
- List<IRenderable> entities	list of all IRenderables
-Comparator<IRenderable> comparator	the comparator that use for draw ordering
+ Player player	the player variable
+ Field field	the field variable
+ Image mapSprite	map image
+ Image castlePlayerSprite	castlePlayer image
+ Image pondSprite	ponSprite image
+ Image tree1Sprite	tree image
+ Image deadTreeSprite	dead tree image

+ Image barbewireSprite	barbewire image
+ Image cannonSprite	cannon image

5.6.2.2 Constructor

+ RenderableHolder_Logic()	initialize all variable

5.6.2.3 Method

+ long getTime()	get the value of time
+ void setTime(long time)	set the value of time
+ long getStart_time()	get the value of start_time
+ void setStart_time(long start_time)	set the value of start_time
+ ArrayList<Obstacle> getObstacles()	get the obstacles
+ void setObstacles(ArrayList<Obstacle> obstacles)	set the obstacles
+ Field getField()	get the field
+ void setField(Field field)	set the field
+ RenderableHolder_Logic getInstance()	get the instance of the class
+ Player getPlayer()	get the player
+ setPlayer(Player player)	set the player

+ add(IRenderable entity)	add the IRenderable object to the entities
- loadResource()	load all initialize image
- void monsterGen()	generate monster every (10 second - genRate)always genarate slime and zombie if the time pass more than 30 , 60 , 120 second the higher level monster (Unicorn , God , Dragon will be generated) by using start_time variable to compare with the system time and the rate of generating monster will be increase by genRate
+ List<IRenderable> getEntities()	get the Entities (list of all irenderable objects)
+ void clearList()	clear all entities
+ void update()	- update monsterGen() - remove all destroy object from the list - check collision of the bullet and fireball - check collision of the player