

# TOWER HATER 2.0

---

By

Chawan Piansaddhayanon 5931013521

Nattaphon Thammarcheep 5931022121

This is the final project's report for 2110215

Programming Methodology I.

# Table of Contents

<b>1. Introduction</b>	<b>3</b>
<b>2. User Manual</b>	<b>3</b>
<b>3. Implement Details</b>	<b>7</b>
<b>3.1 Package : application</b>	<b>8</b>
<b>3.1.1 Class Main extends application</b>	<b>8</b>
<b>3.2 Package : button</b>	<b>8</b>
<b>3.2.1 Interface IButton</b>	<b>8</b>
<b>3.2.2 abstract class NormalButton extends Canvas implements IButton</b>	<b>8</b>
<b>3.2.3 class PlayButton extends NormalButton</b>	<b>9</b>
<b>3.2.4 class ExitButton extends NormalButton</b>	<b>9</b>
<b>3.2.5 class MenuButton extends NormalButton</b>	<b>10</b>
<b>3.2.6 class ReplayButton extends NormalButton</b>	<b>10</b>
<b>3.2.7 class RetryButton extends ReplayButton</b>	<b>11</b>
<b>3.2.8 class ExitIcon extends Canvas implements IButton</b>	<b>12</b>
<b>3.2.9 class HomeIcon extends Canvas implements IButton</b>	<b>13</b>
<b>3.2.10 class PauseIcon extends Canvas implements IButton</b>	<b>14</b>
<b>3.2.11 class CharacterButton extends NormalButton</b>	<b>16</b>
<b>3.3 Package : exception</b>	<b>16</b>
<b>3.3.1 class PoorException extends Exception</b>	<b>16</b>
<b>3.4 Package :utility</b>	<b>17</b>
<b>3.4.1 class Utility</b>	<b>17</b>
<b>3.4.2 class InputUtility</b>	<b>17</b>
<b>3.5 Package :sceneManager</b>	<b>18</b>
<b>3.5.1 class SceneManager</b>	<b>18</b>
<b>3.6 Package :sharedObject</b>	<b>21</b>
<b>3.6.1 interface IRenderable</b>	<b>21</b>
<b>3.6.2 class RenderableHolder</b>	<b>21</b>
<b>3.7 Package : model_general</b>	<b>24</b>
<b>3.7.1 Class Board implements IRenderable</b>	<b>24</b>
<b>3.7.2 Class Entity implements IRenderable</b>	<b>27</b>
<b>3.7.3 Class Bullet extends Entity implements IMovable</b>	<b>28</b>
<b>3.7.4 Class PenetrateBullet extends Bullet</b>	<b>29</b>
<b>3.8 Package : model_attacker</b>	<b>30</b>
<b>3.8.1 Class Attacker extends Entity implements IMovable</b>	<b>30</b>
<b>3.8.2 Class Peasant extends Attacker</b>	<b>32</b>
<b>3.8.3 Class Footman extends Attacker</b>	<b>32</b>
<b>3.8.4 Class Wardog extends Attacker</b>	<b>33</b>

3.8.5 Class Berserker extends Attacker	33
3.8.6 Class Sapper extends Attacker	34
3.8.7 Class Saboteur extends Attacker	34
3.8.8 Class General extends Attacker	35
3.9 Package : model_defender	35
3.9.1 Class Defender extends Entity	35
3.9.2 Class HQ extends Defender	36
3.9.3 Class Tower extends Defender	37
3.9.4 Class PenetrateTower extends Tower	39
3.9.5 Class Wall extends Defender	39
3.10 Package : logic	40
3.10.3 class interface IMovable	40
3.10.2 class Gamelogic	40
3.10.3 class GameState	41
3.11 Package : drawing	42
3.11.1 Class GameScreen extends Canvas	42
3.11.2 Class LoseScreen extends Pane	43
3.11.3 Class Menubar extends VBox	44
3.11.4 Class PlayScreen extends HBox	49
3.11.5 Class StartScreen extends Pane	52
3.11.6 Class WaitScreen extends Pane	53
3.11.7 Class CharacterScreen extends Pane	56

## 1. Introduction

This project is a tower attacking game created with JAVA programming language. We have five boards that have head quarter, wall, tower, and penetrate tower and seven characters which are peasant, Footman, Wardog, Berserker, Sapper, Saboteur and General. The user plays this game by using money to deploy the characters to the board for destroying the head quarter. The two obstacles are limited time and limited money.

## 2. User Manual



Figure 1) Screen that is shown when you start the game.

The user starts with the menu screen shown in figure 1. He will have three choices which are play the game, go to character description screen or exit the game. The character description screen is shown in figure 2. The seven characters have different ability and different cost, so you should use it wisely to be the victorious in this game.



Figure 2) Screen that shows the detail of each character.

The seven characters are

- Peasant (Cost 50) "The Cheapest - Low Price Low Return"
- Footman (Cost 240) "The Tanker - High Health Low Damage"
- Wardog (Cost 120) "The Fastest - High Speed Unit But Deal Half Damage to HQ";
- Berserker (Cost 300) "The Destroyer - High Damage Low Health"
- Sapper (Cost 200) "Wall Bomber - Suicide Attack to Wall"
- Saboteur (Cost 200) "The HQ Hater - Deal Extra Massive Damage to HQ"
- General (Cost 1000) "The Strongest - High Price High Return"

If the user is ready to challenge this game, he will click the play button to go to play screen shown in Figure 3.



Figure 3) The Play Screen

The user can deploy the seven types of character. It will choose the best path to attack the head quarter. If the user can win this stage, it will show the win screen shown in Figure 4, so he can choose to play next stage, play this stage again or exit the game. On the other hand, if he lose by lack of money when all units die or the time is up, it will show the loose screen shown in Figure 5, so he can choose to play this stage again, go to main menu or exit the game.



Figure 4) Screen that is shown when you clear the stage.



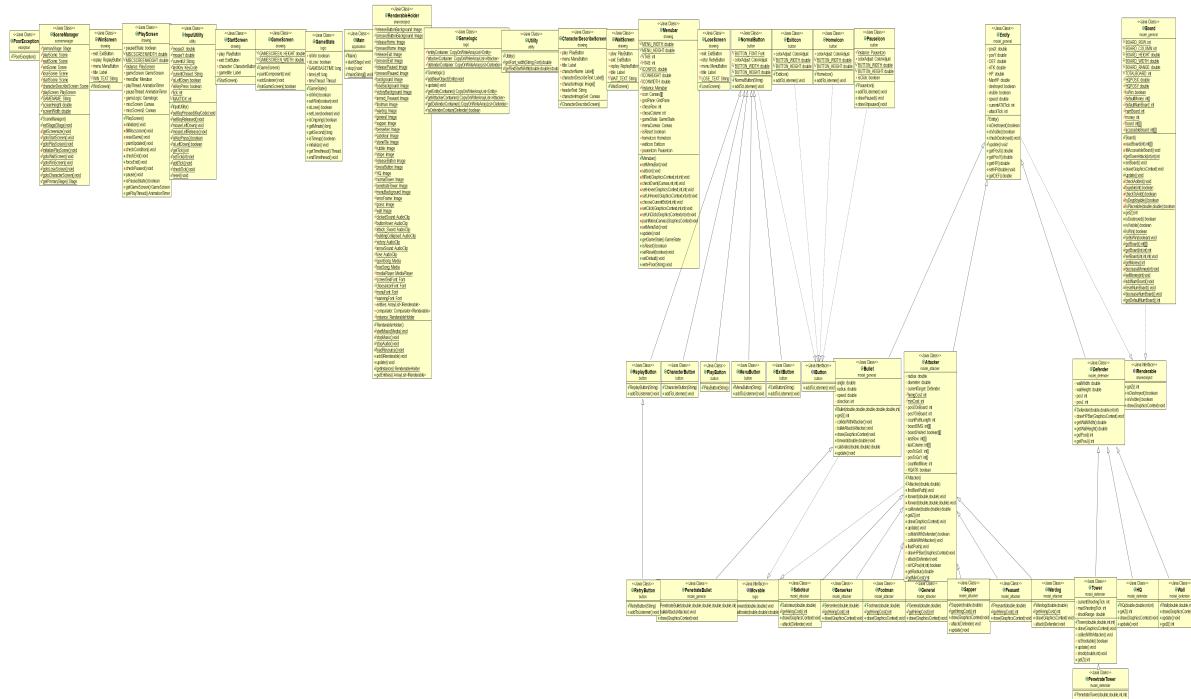
Figure 5) Screen that is shown when you lose the game.



Figure 6) Screen that is shown when you clear the game.

If the user can clear all five stages with his brilliant mind, it will show the clear game screen, so the user can choose to play again, go to main menu or exit the game.

### 3. Implement Details



The UML Diagram of the program

## 3.1 Package : application

### 3.1.1 Class Main extends application

#### 3.1.1.1 Method

+ void start(Stage primaryStage)	The main entry point for the JavaFX applications. It send primaryStage to SceneManager
+ void stop()	Stop all sound and thread then Exit the application.
+ void main(String[] args)	An entry point of the application.

## 3.2 Package : button

This package collect all button and icon.

### 3.2.1 Interface IButton

#### 3.2.1.1 Method

+ void addToListener()	Do nothing
------------------------	------------

### 3.2.2 abstract class NormalButton extends Canvas implements IButton

#### 3.2.2.1 Field

+ double BUTTON_WIDTH	The width of a normal button. Set it to 400.
+ double BUTTON_HEIGHT	The height of a normal button. Set it to 100.
# ColorAdjust colorAdjust	Adjustment for NormalButton's canvas

#### 3.2.2.2 Constructor

+ NormalButton(String text)	Initialize colorAdjust. Initialize canvas. The canvas width and height is equals to BUTTON_WIDTH and BUTTON_HEIGHT respectively. Then draw by this following steps 1) Draw RenderableHolder.releaseButton Background by set x=0,y=0,w=BUTTON_WIDTH ,h=BUTTON_HEIGHT.
-----------------------------	--

	<p>2)Draw Font The current word's font is RenderableHolder.buttonFont. The text is on the center of the NormalButton(set width by using Utility.getTextStartWidth and Utility.getFont_width). Add event handlers for this canvas by calling addListener().</p>
--	--

### 3.2.2.3 Method

+ void addListener()	<p>This is the method for calling eventHandler.</p> <ul style="list-style-type: none"> <li>- When the mouse enter this canvas play RenderableHolder.buttonHover sound and adjust the canvas's brightness to 0.1.</li> <li>- When the mouse exit this canvas padjust the canvas's brightness to 0.</li> <li>- When we left click this canvas do the following steps.</li> </ul> <ol style="list-style-type: none"> <li>1) Stop all Audio</li> <li>2) Play RenderableHolder.clickedSound sound.</li> <li>3) tell the SceneManager to go to PlayScreen.</li> </ol>
----------------------	---

### 3.2.3 class PlayButton extends NormalButton

#### 3.2.3.1 Constructor

+ PlayButton(String text)	Initialize all fields by utilizing its parent constructor.
---------------------------	--

### 3.2.4 class ExitButton extends NormalButton

#### 3.2.4.1 Constructor

+ ExitButton(String text)	Initialize all fields by utilizing its parent constructor.
---------------------------	--

#### 3.2.4.2 Method

+ void addListener()	<p>This is the method for calling eventHandler.</p> <ul style="list-style-type: none"> <li>- When the mouse enter this canvas play RenderableHolder.buttonHover</li> </ul>
----------------------	--

	<p>sound and adjust the canvas's brightness to 0.1.</p> <ul style="list-style-type: none"> <li>- When the mouse exit this canvas padjust the canvas's brightness to 0.</li> <li>- When we left click this canvas do the following steps.</li> </ul> <ol style="list-style-type: none"> <li>1) Play RenderableHolder.clickedSound sound.</li> <li>2) Exit the application</li> </ol>
--	---

### 3.2.5 class MenuButton extends NormalButton

#### 3.2.5.1 Constructor

+ MenuButton(String text)	Initialize all fields by utilizing its parent constructor.
---------------------------	--

#### 3.2.5.2 Method

+ void addToListener()	<p>This is the method for calling eventHandler.</p> <ul style="list-style-type: none"> <li>- When the mouse enter this canvas play RenderableHolder.buttonHover sound and adjust the canvas's brightness to 0.1.</li> <li>- When the mouse exit this canvas padjust the canvas's brightness to 0.</li> <li>- When we left click this canvas do the following steps.</li> </ul> <ol style="list-style-type: none"> <li>1) Stop all Audio.</li> <li>2) Play RenderableHolder.clickedSound sound.</li> <li>3) tell the SceneManager to go to StartScreen.</li> </ol>
------------------------	---

### 3.2.6 class ReplayButton extends NormalButton

#### 3.2.6.1 Constructor

+ ReplayButton(String text)	Initialize all fields by utilizing its parent constructor.
-----------------------------	--

#### 3.2.6.2 Method

+ void addListener()	This is the method for calling eventHandler. <ul style="list-style-type: none"><li>- When the mouse enter this canvas play RenderableHolder.buttonHover sound and adjust the canvas's brightness to 0.1.</li><li>- When the mouse exit this canvas padjust the canvas's brightness to 0.</li><li>- When we left click this canvas do the following steps.<ol style="list-style-type: none"><li>1) Stop all Audio.</li><li>2) Play RenderableHolder.clickedSound sound.</li><li>3) decrease decreaseNumboard in Class Board by 1( Call Board. decreaseNumboard()).</li><li>4) tell the SceneManager to go to PlayScreen.</li></ol></li></ul>
----------------------	---



### 3.2.7 class RetryButton extends ReplayButton

#### 3.2.7.1 Constructor

+ RetryButton(String text)	Initialize all fields by utilizing its parent constructor.
----------------------------	--

#### 3.2.7.2 Method

+ void addListener()	This is the method for calling eventHandler. <ul style="list-style-type: none"><li>- When the mouse enter this canvas play RenderableHolder.buttonHover sound and adjust the canvas's brightness to 0.1.</li><li>- When the mouse exit this canvas padjust the canvas's brightness to 0.</li><li>- When we left click this canvas do the following steps.<ol style="list-style-type: none"><li>1) Stop all Audio.</li><li>2) Play RenderableHolder.clickedSound sound.</li></ol></li></ul>
----------------------	--

	3) tell the SceneManager to go to PlayScreen.
--	---

3.2.8 class ExitIcon extends Canvas implements IButton

This is the Icon that can exit the Application if pressed.

### 3.2.8.1 Field

- double <u>BUTTON_WIDTH</u>	The width of a normal button. Set it to Menubar.MENU_WIDTH*0.2+1.
- double <u>BUTTON_HEIGHT</u>	The height of a normal button. Set it to Menubar.ICONPOS*0.8/3.
- ColorAdjust colorAdjust	Adjustment for ExitIcon's canvas

### 3.2.8.2 Constructor

+ ExitIcon()	<p>Initialize colorAdjust.</p> <p>Initialize canvas. The canvas width and height is equals to BUTTON_WIDTH and BUTTON_HEIGHT respectively.</p> <p>Draw RenderableHolder.releaseExit by set x=0,y=0,w=BUTTON_WIDTH ,h=BUTTON_HEIGHT .</p> <p>Add event handlers for this canvas by calling addToListener().</p>
--------------	--

### 3.2.8.3 Method

+ void addToListener()	<p>This is the method for calling eventHandler.</p> <ul style="list-style-type: none"> <li>- When the mouse enter this canvas play RenderableHolder.buttonHover sound and adjust the canvas's brightness to 0.1.</li> <li>- When the mouse exit this canvas padjust the canvas's brightness to 0.</li> <li>- When we left click this canvas do the following steps.</li> </ul> <ol style="list-style-type: none"> <li>1) Draw RenderableHolder.pressedExit by set x=0,y=0,w=BUTTON_WIDTH ,h= BUTTON_HEIGHT</li> <li>2) Stop all Audio.</li> <li>3) Play RenderableHolder.clickedSound</li> </ol>
------------------------	--

	<p>sound.</p> <p>4) Exit the application</p>
--	--

3.2.9 class Homelcon extends Canvas implements IButton

This is the Icon that can go to Mainmenu(StartScreen) if pressed.

### 3.2.9.1 Field

- double BUTTON_WIDTH	The width of a normal button. Set it to Menubar.MENU_WIDTH*0.2+1.
- double BUTTON_HEIGHT	The height of a normal button. Set it to Menubar.ICONPOS*0.8/3.
- ColorAdjust colorAdjust	Adjustment for Homelcon 's canvas

### 3.2.9.2 Constructor

+ Homelcon()	<p>Initialize colorAdjust.</p> <p>Initialize canvas. The canvas width and height is equals to BUTTON_WIDTH and BUTTON_HEIGHT respectively.</p> <p>Draw RenderableHolder.releaseHome by set x=0,y=0,w=BUTTON_WIDTH ,h=BUTTON_HEIGHT .</p> <p>Add event handlers for this canvas by calling addToListener().</p>
--------------	--

### 3.2.9.3 Method

+ void addToListener()	<p>This is the method for calling eventHandler.</p> <ul style="list-style-type: none"> <li>- When the mouse enter this canvas play RenderableHolder.buttonHover sound and adjust the canvas's brightness to 0.1.</li> <li>- When the mouse exit this canvas padjust the canvas's brightness to 0.</li> <li>- When we left click this canvas do the following steps.</li> </ul> <ol style="list-style-type: none"> <li>1) Draw RenderableHolder.pressedHome by set x=0,y=0,w=BUTTON_WIDTH ,h=BUTTON_HEIGHT</li> <li>2) Set DefaultNumboard in Board to 0</li> <li>3) Stop all Audio.</li> </ol>
------------------------	--

	<p>4) Play RenderableHolder.clickedSound sound.</p> <p>5) Force the current game to end(call forceEnd() in Class PlayScreen by using it's singleton).</p> <p>6) Tell the SceneManager to go to StartScreen.</p> <p>7) Reset InputUtility.</p>
--	---

### 3.2.10 class Pauselcon extends Canvas implements IButton

This is the Icon that can pause/unpause the game.



Figure a)

Figure b)

Figure a) Pauselcon's GUI when the button is unpressed(defaultstate, isClick=false).

Figure b) Pauselcon's GUI when the button is pressed(isClick=true).

#### 3.2.10.1 Field

<u>- double BUTTON_WIDTH</u>	The width of a normal button. Set it to Menubar.MENU_WIDTH*0.2+1.
<u>- double BUTTON_HEIGHT</u>	The height of a normal button. Set it to Menubar.ICONPOS*0.8/3.
<u>- ColorAdjust colorAdjust</u>	Adjustment for Pauselcon's canvas
<u>+ Pauselcon instance</u>	Singleton of this class.
<u>- boolean isClick</u>	Check whether the button is pressed(if the button is pressed twice then it is unpressed).

#### 3.2.9.2 Constructor

<u>+ Pauselcon()</u>	Initialize colorAdjust. Initialize instance=this. set isClick to false. Initialize canvas. The canvas width and height is equals to BUTTON_WIDTH and BUTTON_HEIGHT respectively.
----------------------	---

	<p>Draw RenderableHolder.releasePaused by set x=0,y=0,w=BUTTON_WIDTH ,h=BUTTON_HEIGHT .</p> <p>Add event handlers for this canvas by calling addListener().</p>
--	---

### 3.2.10.3 Method

+ void addListener()	<p>This is the method for calling eventHandler.</p> <ul style="list-style-type: none"> <li>- When the mouse enter this canvas play RenderableHolder.buttonHover sound and adjust the canvas's brightness to 0.1.</li> <li>- When the mouse exit this canvas padjust the canvas's brightness to 0.</li> <li>- When we left click this canvas</li> <li>- If(this canvas is not clicked(isClick==false) call drawPaused() and pasue the game(call pause() in Class PlayScreen by using it's singleton).</li> <li>- Otherwise call drawUnpaused() and unpasue the game(call pause() again in Class PlayScreen by using it's singleton).</li> </ul>
+ void drawPaused()	<p>This method change this Icon's apperance.</p> <p>Set isClick to true.</p> <p>Do the following steps</p> <ol style="list-style-type: none"> <li>1) Draw RenderableHolder.pressedPaused by set x=0,y=0,w=BUTTON_WIDTH ,h=BUTTON_HEIGHT</li> <li>2) Play RenderableHolder.clickedSound sound.</li> </ol>
+ void drawUnpaused()	<p>This method change this Icon's apperance.</p> <p>Set isClick to false.</p> <p>Do the following steps</p> <ol style="list-style-type: none"> <li>1) Draw</li> </ol>

	<p>RenderableHolder.releasePaused by set x=0,y=0,w=BUTTON_WIDTH ,h=BUTTON_HEIGHT</p> <p>2) Play RenderableHolder.clickedSound sound.</p>
--	--

### 3.2.11 class CharacterButton extends NormalButton

#### 3.2.5.1 Constructor

+ CharacterButton(String text)	Initialize all fields by utilizing its parent constructor.
--------------------------------	--

#### 3.2.5.2 Method

+ void addToListener()	<p>This is the method for calling eventHandler.</p> <ul style="list-style-type: none"> <li>- When the mouse enter this canvas play RenderableHolder.buttonHover sound and adjust the canvas's brightness to 0.1.</li> <li>- When the mouse exit this canvas padjust the canvas's brightness to 0.</li> <li>- When we left click this canvas do the following steps.</li> </ul> <p>3) Stop all Audio.          4) Play RenderableHolder.clickedSound sound.          5) tell the SceneManager to go to CharacterScreen.</p>
------------------------	--

## 3.3 Package : exception

### 3.3.1 class PoorException extends Exception

This Exception occur when you are poor(Not enough money to buy the clicked Attacker).

#### 3.3.1.1 Constructor

+ PoorException()	Tell Menubar that we can't buy the clicked Attacker(send String "YOU CANT AFFORD THIS UNIT" to method
-------------------	---

	writepoor() by using Menubar's singleton).
--	--

## 3.4 Package :utility

### 3.4.1 class Utility

This class is used for calculating method that are often used.

#### 3.4.1.1 Method

<code>+ double getFont width(String text,Font font)</code>	This method return received strings' width when the font is applied to the string.
<code>+double getTextStartWidht(double totalwidth,double fontlength)</code>	This method return the position of xAxis by receiving the width of the canvas and the length of the text of which font is applied and return $(totalwidth - fontlength)/2$ (Return the position that would make the text in XAxis aligned at the center).

### 3.4.2 class InputUtility

This class collect and change the state after the EventHandler in GameScreen and Menubar is handled.

#### 3.4.2.1 Field

<code>+ double mouseX</code>	Current mouse location in xAis.
<code>+ double mouseY</code>	Current mouse location in yAis.
<code>+ String currentUI</code>	This String collect whether the current mouse position is in GameScreen or not("x" means that the cursor is not in GameScreen, "GAME" for otherwise).
<code>+KeyCode lastKey;</code>	Last pressed key.
<code>+String currentChosed</code>	The last chose Attacker("x" mean that none of the bot chose, Attackers' classname for otherwise).
<code>-boolean isLeftDown</code>	This Field tell whether the left mouse is being hold. Default is false.
<code>-static boolean isKeyPress</code>	This Field tell whether any Keyboard is being pressed, Default is false.
<code>- int tick</code>	The number of tick that have

	pass.Default is 10.(tick is
<u>+int MAXTICK</u>	Number of maximum tick(tick can't exceed MAXTICK).Default is equal to tick .

### 3.4.2.2 Method

<u>+ setKeyPressed(KeyCode keycode)</u>	Set Lastkey to keycode. Set isKeyPress to true.
<u>+ setKeyReleased()</u>	Set isKeyPress to false.
<u>+ void mouseLeftDown()</u>	Set isLeftDown to true(The left mouse is being hold).
<u>+void mouseLeftRelease()</u>	Set isLeftDown to false. Set tick to Maxtick.
<u>+void checkTick()</u>	If the left mouse is being hold call then add the number of Tick by 1.
<u>+boolean isKeyPress()</u>	Getter for isKeyPress.
<u>+boolean isLeftDown()</u>	Getter for isLeftDown.
<u>+ int getTick()</u>	Getter for tick.
<u>+void setTick(int tick)</u>	Setter for tick.
<u>+void addTick()</u>	Add the number of tick by 1.
<u>+void reset()</u>	set All Field this class to default except mouseX and mouseY.

## 3.5 Package :sceneManager

### 3.5.1 class SceneManager

#### 3.5.1.1 Field

<u>-Stage primaryStage</u>	A primary stage.
<u>-Scene playScene</u>	A play scene(Scene that we play).
<u>-Scene waitScene</u>	A waiting scene(Scene that appear when you clear a stage in the game).
<u>-Scene winScene</u>	A won scene(Scene that appear when you win the game).
<u>-Scene loseScene</u>	A lost scene(Scene that appear when

	you lose the game).
<u>-Scene startScene</u>	A starting scene(Scene that appear when you start the game(Main menu) ).
<u>-Scene characterDescribeScene</u>	A starting scene for <u>characterDescribeScreen</u>
<u>- PlayScreen playScreen</u>	A play screen.
<u>+String GAMENAME</u>	Name of the game.Default is "TOWER HATER 2.0".
<u>+double screenHeight</u>	The height of the screen.
<u>+double screenWidth</u>	The width of the screen.

### 3.5.1.2 Method

<u>+void setStage(Stage primaryStage)</u>	<p>Set this primaryStage to given primaryStage.</p> <p>Set this primaryStage's title to GAMENAME.</p> <p>Set this primaryStage's height to screenHeight.</p> <p>Set this primaryStage's width to screenWidth.</p> <p>Set this primaryStage's FullScreenExitHint to “”;</p> <p>Make primaryStage aligned at the center of the screen.</p> <p>Show this primaryStage.</p>
<u>+void getScreensize()</u>	<p>If OS is window 10 set screenHeight to height of screen resolution*<math>\frac{2}{3}</math> and set screenWidth to width of screen resolution*<math>\frac{2}{3}</math>.</p> <p>Otherwise set screenHeight to height of screen resolution*0.8 and set screenWidth to width of screen resolution*0.8.</p>
<u>+void gotoStartScreen()</u>	<p>Initialize startScene by initializing startScreen then set scene's width to screenWidth and set scene's height to screenHeight.</p> <p>Set primaryStage's scene to StartScene.</p> <p>Set this primaryStage's screen to fullscreen.</p>

<u>+void gotoPlayScreen()</u>	Initialize playScene by calling initializePlayScene(). Set primaryStage's scene to playScene. Set this primaryStage's screen to fullscreen. Start playScreen's AT(AnimationTlmer).
<u>+void initializePlayScene()</u>	Initialize playScreen Initialize playScene by using playScreen then set scene's width to screenWidth and set scene's width to screenHeight. Request playScreen's focus.
<u>+void gotoWaitScreen()</u>	Initialize waitScene by initializing waitScreen then set scene's width to screenWidth and set scene's width to screenHeight. Set primaryStage's scene to waitScene. Set this primaryStage's screen to fullscreen.
<u>+void gotoWinScreen()</u>	Initialize winScene by initializing winScreen then set scene's width to screenWidth and set scene's width to screenHeight. Set primaryStage's scene to winScene. Set this primaryStage's screen to fullscreen.
<u>+void gotoLoseScreen()</u>	Initialize loseScene by initializing loseScreen then set scene's width to screenWidth and set scene's width to screenHeight. Set primaryStage's scene to loseScene. Set this primaryStage's screen to fullscreen.
<u>+void gotoCharacterScreen()</u>	Initialize characterDescribeScene by initializing CharacterDescribeScreen then set scene's width to screenWidth and set scene's width to screenHeight. Set primaryStage's scene to characterDescribeScene. Set this primaryStage's screen to fullscreen.

## 3.6 Package :sharedObject

### 3.6.1 interface IRenderable

#### 3.6.1.1 Method

+ int getZ()	(Priority in RenderableHolder)
+boolean isDestroyed()	Do nothing
+boolean isVisible()	Do nothing
void draw(GraphicsContext gc)	Do nothing

### 3.6.2 class RenderableHolder

#### 3.6.2.1 Field

<u>+Image releaseButtonBackground</u>	BackGround of all button that is not in Playscreen when it is unpressed not it is not pressed.
<u>+Image pressedButtonBackground</u>	BackGround of all button that is not in Playscreen when it is pressed.
<u>+Image releaseHome</u>	Image of Homelcon when it is unpressed not it is not pressed.
<u>+Image pressedHome;</u>	Image of Homelcon when it is pressed.
<u>+Image releaseExit</u>	Image of ExitIcon when it is unpressed not it is not pressed.
<u>+Image pressedExit</u>	Image of ExitIcon when it is pressed.
<u>+Image releasePaused</u>	Image of PausedIcon when it is unpressed not it is not pressed.
<u>+Image pressedPaused</u>	Image of PausedIcon when it is pressed.
<u>+Image background</u>	BackGround of StartScreen.
<u>+Image loseBackground</u>	BackGround of LoseScreen.
<u>+Image victoryBackground</u>	BackGround of WinScreen and WaitScreen.
<u>+Image armed Peasant</u>	Image of a peasant.
<u>+Image footman</u>	Image of a footman.
<u>+Image wardog</u>	Image of a war dog.

<u>+Image general</u>	Image of a general.
<u>+Image sapper</u>	Image of a sapper.
<u>+Image berserker</u>	Image of a berserker.
<u>+Image saboteur</u>	Image of a saboteur.
<u>+Image stoneTile</u>	Image of a wall.
<u>+Image rubble;</u>	Image of destroyed Defender.
<u>+Image stripe</u>	Image of a stripe(Show that attackers cannot deploy in the striped tile).
<u>+Image releaseButton;</u>	BackGround of button that is in Playscreen when it is unpressed not it is not pressed.
<u>+Image pressButton</u>	BackGround of button that is in Playscreen when it is pressed.
<u>+Image HQ</u>	Image of a HQ.
<u>Image normalTower</u>	Image of NormalTower.
<u>Image penetrateTower</u>	Image of PenetrateTower.
<u>+Image menuBackground</u>	Image of topleft of the menubar.
<u>+Image errorFrame</u>	Image of ErrorFrame(It is a background for displaying the text when some problem occur.
<u>+Image grass</u>	Image of a grass.
<u>+Image wall</u>	Image of a wall.
<u>+AudioClip clickedSound</u>	The sound when all button is clicked.
<u>+AudioClip buttonHover</u>	The sound when all button is being hovered.
<u>+AudioClip attack_Sword</u>	The sound when Attacker attacks.
<u>+AudioClip buildingCollapsed</u>	The sound when the Defenders are destroyed.
<u>+AudioClip victory</u>	The sound when we clear the stage.
<u>+AudioClip arrowSound;</u>	The sound when bullet is shot.

<u>+AudioClip lose</u>	The sound when we lose the stage.
<u>+Media openSong</u>	The song when you enter StartScreen.
<u>+Media loseSong:</u>	The song when you enter LoseScreen.
<u>+MediaPlayer mediaPlayer</u>	A media player for Media.
<u>+Font screenTextFont</u>	Font of Label in StartScreen,LoseScreen,WinScreen,WaitScreen.
<u>+Font chooseIconFont</u>	Font of Icon in Menubar.
<u>+Font menuFont</u>	Font of menu information in Menubar.
<u>+Font warningFont</u>	Font of error that is caught by any exception,which will be shown in Menubar.
<u>+Font buttonFont</u>	Font for all button.
<u>-ArrayList&lt;IRenderable&gt; entities</u>	This list collect all entities.
<u>-Comparator&lt;IRenderable&gt; comparator</u>	Comparator of entities.
<u>-RenderableHolder instance</u>	Singleton of RenderableHolder

### 3.6.2.2 Constructor

<u>+RenderableHolder()</u>	Initialize entities and set comparator to sort in ascending order.
----------------------------	--

### 3.6.2.3 Method

<u>+void startMusic(Media media)</u>	Stop the music. Initialize new mediaPlayer with media. Set cycle count to MAX_INTEGER. Play the media.
<u>+void stopMusic()</u>	Stop the Music(catch NullPointerException(the song hasn't start yet))
<u>+void stopAudio()</u>	Stop all sound(Song is included) except RenderableHolder.clickedSound.
<u>+void loadResource()</u>	Load all Image and AudioClip. If Image or AudioClip is not found show an alert in Figure c) and exit the application.

+void add(IRenderable entity)	Add entity to entities,then sort entities by using Comparator<IRenderable> comparator as a comparator.
+void update()	Remove IRenderable Object if it is destroyed,if the destroyed object is a defender play Renderabel.buildingCollapsed sound and if the destroyed object is a HQ set isWin in Board to true.
+ RenderableHolder getInstance	This is instance's getter.
+ ArrayList<IRenderable> getEntities()	This is entities' getter.

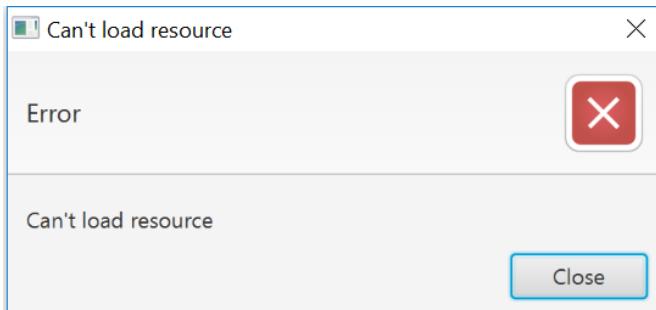


Figure c) This alert is shown when resource can not be found.

### 3.7 Package : model\_general

#### 3.7.1 Class Board implements IRenderable

##### 3.7.1.1 Field

+int BOARD_ROW	The amount of row of board.Default is 30.
+int BOARD_COLUMN	The amount of column of board.Default is 30.
+double BOARD_HEIGHT	The height of each tile.Default is GameScreen.GAMESCREEN_HEIGHT/BOARD_ROW.
+double BOARD_WIDTH	The width of each tile.Default is GameScreen.GAMESCREEN_WIDTH/BOARD_COLUMN.
+double BOARD_RANGE	The min value between BOARD_WIDTH and BOARD_HEIGHT

<u>+int TOTALBOARD</u>	Number of game board
<u>+double HQPOSX</u>	Position of Head Quater in X-axis
<u>+double HQPOSY</u>	Position of Head Quater in Y-axis
<u>+boolean isWin</u>	Tell whether the player win the game. The default is false.
<u>-int[] defaultMoney</u>	The starting money for summon the soldier in each game.
<u>-int defaultNumboard</u>	The starting game board.
<u>-int numBoard</u>	The current game board.
<u>-int money</u>	The current money.
<u>-int[][] board</u>	The values that tell the type of each tile  -1 = Destroyed object 0 = Plain 1 = Wall 2 = Tower 3 = Head Quater 4 = Penetrate Tower
<u>-int[][] accessibleBoard</u>	The values that tell the player can or cannot deploy the soldier in this tile  0 = Cannot deploy the soldier in this tile 1 = Can deploy the soldier in this tile

### 3.7.1.2 Constructor

<u>+ Board()</u>	Set the current board to default board Set the isWin to false Set the current money to the default money of the current board Fill the accessibleBoard (call fillAcessibleBoard() method) Add object of all tile to Gamelogic (call setBoard() method)
------------------	--

### 3.7.1.3 Method

<code>-int[][] readBoard(int numBoard)</code>	Read board[][] from files
<code>-void fillAccessibleBoard()</code>	Fill the accessibleBoard using data from board
<code>+int getTowerAttack(int posX,int posY)</code>	Return the amount of tower that can attack this position plus by three. if it is the wall multiply the amount of tower by four.
<code>+void setBoard()</code>	Set up board from reading data Gamelogic.addNewObject for every defender and find HQ position
<code>+void draw(GraphicsContext gc)</code>	Draw every object in board
<code>+void update</code>	Call checkAdded method
<code>-void checkAdded()</code>	Check the money for deploy chosen attacker
<code>-boolean buyable(int hiringCost)</code>	Check that the money can or cannot deploy the attacker. If cannot, throw PoorException
<code>-boolean checkToAdd()</code>	Check the isDeployable and isPlaceable methods
<code>-boolean isDeployable()</code>	Check the tick count that can or cannot deploy the attacker at this time
<code>-boolean isPlaceable()</code>	Check that the mouse position can or cannot place with data from accessibleBoard
<code>+int getZ()</code>	return -9999 (Priority in RenderableHolder)
<code>+boolean isDestroyed()</code>	return true
<code>+boolean isVisible()</code>	return false
<code>+boolean isWin()</code>	Getter method of isWin
<code>+void setIsWin(boolean isWin)</code>	Setter method of isWin
<code>+int[][] getBoard()</code>	Getter method of board
<code>+int getBoard(int i,int j)</code>	Getter method of board[][]
<code>+void setBoard(int i,int j,int val)</code>	Setter method of board[][]

<u>+int getMoney</u>	Getter method of money
<u>-void decreaseMoney(int decreaseMoney)</u>	Decrease the money by decreaseMoney
<u>+void setMoney(int newMoney)</u>	Setter method of money
<u>+void addNumBoard()</u>	Increase the value of defaultNumBoard by one
<u>+void resetNumBoard()</u>	Set the value of defaultNumBoard to zero
<u>+void decreaseNumBoard()</u>	Decrease the value of numboard by one
<u>+int getDefaultNumBoard()</u>	Getter method of defaultNumBoard

### 3.7.2 Class Entity implements IRenderable

#### 3.7.2.1 Field

#double posX	The position in x-Axis
#double posY	The position in y-Axis
#double DEF	The defence status
#double ATK	The attack status
#double HP	The HP status
#double MaxHP	The Max HP status
#boolean destroyed	Tell whether the entity is destroyed or not. The default value is false.
#boolean visible	Tell whether the entity is visible or not. The default value is true.
#double speed	The speed status
#int currentATKTick	The number of tick that have pass
#int attackTick	The attack speed status

#### 3.7.2.2 Constructor

+ Entity()	Set destroyed to false Set visible to true
------------	---

### 3.7.2.3 Method

+boolean isDestroyed()	Call the checkDestroyed function and then return value of destroyed
+boolean isVisible()	Getter Method of visible
+void checkDestroyed()	Check the entity is destroyed or not Update the value of destroyed
+abstract void update()	Do nothing
+double getPosX()	Getter Method of posX
+double getPosY()	Getter Method of posY
+double getHP()	Getter Method of HP
+void setHP(double hP)	Setter Method of HP
+double getDEF()	Gettter Method of DEF

### 3.7.3 Class Bullet extends Entity implements Imovable

Note: Bullet HP is the total distance they can travel.

#### 3.7.3.1 Field

#double angle	The angle of the bullet (0-180)
#double radius	The radius of bullet
#double speed	The speed of bullet
#int direction	The exact quadrant of bullet <b>See more about angle and direction in method colliedWithAttacker in Class Tower.</b>

#### 3.7.3.2 Constructor

+ Bullet(double shootRange,double posX,double posY,double ATK,double angle,int direction)	Set the value of ATK,DEF, radius, HP, speed, posX, posY, angle and direction
---	--

### 3.7.3.3 Method

+int getZ()	return 7 (Priority in RenderableHolder)
#void collideWithAttacker()	If the bullet collide with attacker, it will deal damage to attacker by calling bulletAttack method.
#void bulletAttack(Attacker attacker)	Deal damage to attacker and destroy itself.
+void draw(GraphicsContext gc)	Draw the bullet.
+void forward(double xAxis,double yAxis)	Move the bullet to the direction and decrease HP.
+double calibrate(double velocity,double speed)	Calculate the velocity/speed use in forward method.
+void update()	Move forward and check collide with attacker and set the value of radius.

### 3.7.4 Class PenetrateBullet extends Bullet

#### 3.7.4.1 Constructor

+ PenetrateBullet(double Shootrange, double posX, double posY, double ATK, double angle, int direction )	Call super constructor, divide ATK by 6 and set the value of radius
--	---

#### 3.7.4.2 Method

#void bulletAttack(Attacker attacker)	Deal damage to attacker but not destroyed itself.
+void draw(GraphicsContext gc)	Draw the penetrate bullet

## 3.8 Package : model\_attacker

3.8.1 Class Attacker extends Entity implements IMovable

### 3.8.1.1 Field

#double radius	The radius value of attacker
#double diameter	The diameter value of attacker
#model_defender.Defender currentTarget	The current target of the attacker
#int hiringCost	The cost use for deploy this attacker
#int minCost	The minimum cost of attacker
#int posXOnBoard	The grid position on x-Axis
#int posYOnBoard	The grid position on y-Axis
#int countPathLength	The length of path to head quarter
#int boardDMG[][]	The damage that attacker take to the path to this position.
#boolean boardVisited[][]	Tell whether we find the path to this position or not. The default is false.
#int lastRow[][]	The last position before go to this position in x-Axis
#int lastColumn[][]	The last position before go to this position in y-Axis
#int posToGoX[]	The position to go in x-Axis
#int posToGoY[]	The position to go in y-Axis
#int countNotMove	Count the time that the attacker stuck in the same position
#boolean HQATK	Tell whether this attacker had ever attack head quarter or not.

### 3.8.1.2 Constructor

+ Attacker(double posX,double posY)	Set the value of posX, posY, posXOnBoard, posYOnBoard and HQATK then call findBestPath method
-------------------------------------	---

### 3.8.1.3 Method

+void findBestPath()	Use priority queue and Dijkstra's algorithm to find the least damage taken path to Head Quarter Position
+void forward(double xAxis,double yAxis)	Move toward xAxis and yAxis direction
+double calibrate(double velocity, double speed)	Calculate the velocity/speed use in forward method
+int getZ()	(Priority in RenderableHolder)
+void draw(GraphicsContext gc)	Draw the attacker image on canvas
+void update()	Call collideWithAttacker method Call collideWithDefender() if it return false then move to the path from findBestPath method
#boolean collideWithDefender()	Check whether the attacker is collide with defender or not. If it collides then deal damage to defender.
#void collideWithAttacker()	Call fluidPush Method for three times
#void fluidPush()	If current Attacker intersects other Attacker push the other Attacker back with the direction it each other faces. If the pushed attacker collide with the defender,then bounce back and bounce the current Attacker with 0.1 times original speed.
#void drawHPBar(GraphicsContext gc)	Draw the HP bar on canvas

#void attack(Defender defender)	Deal damage to the defender
#boolean isHQPos(int nowX,int nowY)	Check whether this position is HQ or not
+double getRadius()	Getter method for radius
<u>+int getMinCost()</u>	Getter method for minCost

### 3.8.2 Class Peasant extends Attacker

#### 3.8.2.1 Constructor

+ Peasant(double posX,double posY)	Set ATK, DEF, radius, diameter, speed, HP, MaxHP, attackTick and currentATKTick of this soldier
------------------------------------	---

#### 3.8.2.2 Method

<u>+ int getHiringCost()</u>	return the cost that use for hire this soldier
+void draw(GraphicsContext gc)	Use gc to draw Draw photo from Renderable at position x=posX-radius, y=posY-radius, w=diameter, h=diameter. Draw a HP bar.

### 3.8.3 Class Footman extends Attacker

#### 3.8.3.1 Constructor

+ Footman(double posX,double posY)	Set ATK, DEF, radius, diameter, speed, HP, MaxHP, attackTick and currentATKTick of this soldier
------------------------------------	---

#### 3.8.3.2 Method

+ int <u>getHiringCost()</u>	return the cost that use for hire this soldier
+void draw(GraphicsContext gc)	Use gc to draw Draw photo from Renderable at position x=posX-radius, y=posY-radius, w=diameter, h=diameter. Draw a HP bar.

### 3.8.4 Class Wardog extends Attacker

#### 3.8.4.1 Constructor

+ Wardog(double posX,double posY)	Set ATK, DEF, radius, diameter, speed, HP, MaxHP, attackTick and currentATKTick of this soldier
-----------------------------------	---

#### 3.8.4.2 Method

+ int <u>getHiringCost()</u>	return the cost that use for hire this soldier
+void draw(GraphicsContext gc)	Use gc to draw Draw photo from Renderable at position x=posX-radius, y=posY-radius, w=diameter, h=diameter. Draw a HP bar.
#void attack(Defender defender)	Deal damage to defender If the defender is Head Quater. It will deal only half damage.

### 3.8.5 Class Berserker extends Attacker

#### 3.8.5.1 Constructor

+ Berserker(double posX,double posY)	Set ATK, DEF, radius, diameter, speed, HP, MaxHP, attackTick and currentATKTick of this soldier
--------------------------------------	---

#### 3.8.5.2 Method

+ int <u>getHiringCost()</u>	return the cost that use for hire this soldier
+void draw(GraphicsContext gc)	Use gc to draw Draw photo from Renderable at position x=posX-radius, y=posY-radius, w=diameter, h=diameter. Draw a HP bar.

### 3.8.6 Class Sapper extends Attacker

#### 3.8.6.1 Constructor

+ Sapper(double posX,double posY)	Set ATK, DEF, radius, diameter, speed, HP, MaxHP, attackTick and currentATKTick of this soldier
-----------------------------------	---

#### 3.8.6.2 Method

+ int <u>getHiringCost()</u>	return the cost that use for hire this soldier
+void draw(GraphicsContext gc)	Use gc to draw Draw photo from Renderable at position x=posX-radius, y=posY-radius, w=diameter, h=diameter. Draw a HP bar.
#void attack(Defender defender)	Deal damage to defender If the defender is Wall, and the Wall HP is not full, It will deal damage equal to the Wall HP.
+void update()	Call collideWithAttacker method Call collideWithDefender() if it return false then walk to the nearest wall.

### 3.8.7 Class Saboteur extends Attacker

#### 3.8.7.1 Constructor

+ Saboteur(double posX,double posY)	Set ATK, DEF, radius, diameter, speed, HP, MaxHP, attackTick and
-------------------------------------	--

	currentATKTick of this soldier
--	--------------------------------

### 3.8.7.2 Method

+ int <u>getHiringCost()</u>	return the cost that use for hire this soldier
+void draw(GraphicsContext gc)	Use gc to draw Draw photo from Renderable at position x=posX-radius, y=posY-radius, w=diameter, h=diameter. Draw a HP bar.
#void attack(Defender defender)	Deal damage to defender If the defender is Head Quarter, It will deal damage 6 times of normal attack

## 3.8.8 Class General extends Attacker

### 3.8.8.1 Constructor

+ General(double posX,double posY)	Set ATK, DEF, radius, diameter, speed, HP, MaxHP, attackTick and currentATKTick of this soldier
------------------------------------	---

### 3.8.8.2 Method

+ int <u>getHiringCost()</u>	return the cost that use for hire this soldier
+void draw(GraphicsContext gc)	Use gc to draw Draw photo from Renderable at position x=posX-radius, y=posY-radius, w=diameter, h=diameter. Draw a HP bar.

## 3.9 Package : model\_defender

### 3.9.1 Class Defender extends Entity

#### 3.9.1.1 Field

# double wallWidth	Width of Defender.
#double wallHeight	Height of Defender.
#int posI	Position of first dimension of the int[][] board in class Board.
#int posJ	Position of second dimension of the array of board in class Board.

### 3.9.1.2 Constructor

+ Defender(double posX,double posY,int posI,int posJ)	Set posX,posY,posI,posJ to given posX,posY,posI,posJ.
---	---

### 3.9.1.3 Method

#void drawHPBar(GraphicsContext gc)	<p>Draw a HP bar if this defender's health is not full.</p> <p>Instruction for drawing HP bar</p> <ol style="list-style-type: none"> <li>1) Draw a DARKGREEN bar at x=posX, y=posY, w=wallWidth, h=4</li> <li>2) Draw a DARKGREEN bar at x=posX+wallWidth*(ratio of hitpoint/maximum hitpoint), y=posY, w=wallWidth*(1-*(ratio of hitpoint/maximum hitpoint)), h=4</li> <li>3) Draw DARKGREEN bar's border with black color with default stroke's width.</li> </ol>
+ double getWallWidth()	Getter of wallWidth.
+double getWallHeight()	Getter of wallHeight.
+int getPosI()	Getter of posI.
+int getPosJ()	Getter of posJ.

## 3.9.2 Class HQ extends Defender

HQ stands for HeadQuarter

### 3.9.2.1 Constructor

+ public HQ(double posX,double posY,int posI,int posJ)	<p>Initialize posX,posY,posI,posJ by utilizing its parent constructor.</p> <p>Set ATK to 0.</p> <p>Set DEF to 0.</p> <p>Set HP to 5000.</p> <p>Set MaxHp to equal to HP.</p>
--	--

	Set wallWidth to Board.BOARD_WIDTH*4. Set wallHeight to Board.BOARD_HEIGHT*4.
--	--

### 3.9.2.2 Method

+ int getZ()	return 3(Priority in RenderableHolder is 3).
+void draw(GraphicsContext gc)	Use gc to draw RenderableHolder.HQ at position x=posX-5, y=posY-5, w=wallWidth+5, h=wallHeight+10. Draw a HP bar.
+void update()	Do nothing.

### 3.9.3 Class Tower extends Defender

#### 3.9.3.1 Field

# int currentShootingTick	Currents shooting tick(If can shoot the bullet if it exceed maxShootingTick).
#int maxShootingTick	Max shooting tick.
#double shootRange	Range that Defender can shoot the bullet.

#### 3.9.3.2 Constructor

+Tower(double posX,double posY,int posI,int posJ)	Initialize posX,posY,posI,posJ by utilizing its parent constructor. Set ATK to 50. Set DEF to 10. Set HP to 500. Set MaxHp to equal to HP. Set shootRange to 5. Set currentShootingTick to 60. Set maxShootingTick to equal to currentShootingTick . Set wallWidth to Board.BOARD_WIDTH. Set wallHeight to Board.BOARD_HEIGHT.
---	---

#### 3.9.3.3 Method

+void draw(GraphicsContext gc)	Use gc to draw
--------------------------------	----------------

	<p>RenderableHolder.normalTower at position x=posX, y=posY, w=wallWidth, h=wallHeight. Draw a HP bar.</p>
# protected void colliedWithAttacker()	<p>Find the closest Attacker that is in shooting range.If the tower can shoot the bullet (call isShootable()), it will shoot the bullet by passing the angle and direction between the tower and the Attacker to method shoot.</p> <p>Angle is calculate by</p> $\theta = (180/\pi) \times \arctan(dy/dx)$ <p>where dy is difference between the Attacker and the tower in y axis, and dx is difference between the Attacker and the tower in x axis.</p> <p>As you can see that range of <math>\arctan(dy/dx)</math> is <math>[-1, 1]</math>, so we can not distinguish between quadrant 1,3 and quadrant 2,4 ,so we send direction to tell the exact quadrant .</p> <p>direction is calculated by:</p> <p>First set direction to 0  if <math>dy &lt; 0</math> direction +=2  if <math>dx &lt; 0</math> direction +=1</p> <p>Note :</p> <p>quadrant 1 direction is 0  quadrant 2 direction is 1  quadrant 3 direction is 3  quadrant 4 direction is 2</p>
#boolean isShootable()	If currentShootingTick exceeds maxShootingTick return true,otherwise return false.
+void update()	Increase currentShootingTick by 1 and check whether any Attacker is in shooting range(call colliedWithAttacker()).
+shoot(double theta,int direction)	<p>Set RenderableHolder.arrowSound's volume to 0.05.</p> <p>Play RenderableHolder.arrowSound sound.</p> <p>Add new Bullet to attackerContainer in clas GameLogic by passing by passing shootRange, posX, posY, ATK, angle,</p>

	direction).
+int getZ()	return 2(Priority in RenderableHolder is 2).

### 3.9.4 Class PenetrateTower extends Tower

#### 3.9.4.1 Constructor

+PenetrateTower(double posX, double posY, int posI, int posJ)	Initialize posX, posY, posI, posJ by utilizing its parent constructor. Set shootRange to 4. Set currentShootingTick to 100. Set maxShootingTick to equal to currentShootingTick .
#void shoot(double theta, int direction)	Same as shoot(double theta, int direction) in class Tower but add object PenetrateBullet instead of Bullet
+void draw(GraphicsContext gc)	Use gc to draw RenderableHolder.penetrateTower at position x=posX, y=posY, w=wallWidth, h=wallHeight. Draw a HP bar.

### 3.9.5 Class Wall extends Defender

#### 3.9.5.1 Constructor

+Wall(double posX, double posY, int posI, int posJ)	Initialize posX, posY, posI, posJ by utilizing its parent constructor. Set ATK to 0. Set DEF to 0. Set HP to 1000. Set MaxHp to equal to HP. Set wallWidth to Board.BOARD_WIDTH. Set wallHeight to Board.BOARD_HEIGHT.
---	--

#### 3.9.5.1 Method

+void draw(GraphicsContext gc)	Use gc to draw RenderableHolder.wall at position x=posX, y=posY, w=wallWidth, h=wallHeight. Draw a HP bar.
+void update()	Do nothing.
+int getZ()	return 1(Priority in RenderableHolder is

	1).
--	-----

### 3.10 Package : logic

#### 3.10.3 class interface IMovable

##### 3.10.3.1 Method

+void forward(double xAxis,double yAxis)	Do nothing
+double calibrate(double velocity,double speed)	Do nothing

#### 3.10.2 class Gamelogic

##### 3.10.2.1 Field

<u>-CopyOnWriteArrayList&lt;Entity&gt;</u> <u>entityContainer</u>	A container for all Entity.
<u>-CopyOnWriteArrayList&lt;Attacker&gt;</u> <u>attackerContainer</u>	A container for all Attacker.
<u>-CopyOnWriteArrayList&lt;Attacker&gt;</u> <u>defenderContainer</u>	A container for all Defender.

##### 3.10.2.2 Constructor

+Gamelogic()	Initialize entityContainer,attackerContainer,defenderContainer. Add new Board to RenderableHolder.
--------------	--

##### 3.10.2.3 Method

<u>+addNewObject(Entity entity)</u>	Add entity to entityContainer. If an entity is an Attacker add entity to attackerContainer. If an entity is a Defender add entity to defenderContainer.
<u>+void update()</u>	Update all entity in entityContainer. Remove destroyed object in entityContainer,defenderContainer,attackerContainer. If entity in defenderContainer is removed let all Attacker expect Sapper find a new path.
<u>+CopyOnWriteArrayList&lt;Entity&gt;</u> <u>getEntityContainer()</u>	Getter of entityContainer.

<code>+CopyOnWriteArrayList&lt;Entity&gt; getAttackerContainer()</code>	Getter of attackerContainer.
<code>+CopyOnWriteArrayList&lt;Defender&gt; getDefenderContainer()</code>	Getter of defenderContainer.
<code>+boolean isDefenderContain(Defender defender)</code>	Return true if defender is found in defenderContainer, otherwise return false.

### 3.10.3 class GameState

#### 3.10.3.1 Field

- boolean isWin	Tell whether the user is win or not. The default is false.
- boolean isLose	Tell whether the user is lose or not. The default is false.
- long GAMEBASETIME	The game's base time. Set to to 151*1000000000L(151 second).
- long timeLeft	The current time left
- Thread timeThread	A thread that count the time for GAMEBASETIME/100000000L second.

#### 3.10.3.2 Constructor

<code>+ GameState()</code>	Call initialize() and start a thread that count the time for GAMEBASETIME/1000000000L second, the time won't be counted if the game is paused or lose or cleared.
----------------------------	---

#### 3.10.3.3 Method

<code>+ boolean isWin()</code>	Getter method of isWin
<code>+ void setWin(boolean isWin)</code>	Setter method of isWin
<code>+ boolean isLose()</code>	Getter method of isLose
<code>+ void setLose(boolean isLose)</code>	Setter method of isLose
<code>+ boolean isOngoing</code>	Return true if isWin and isLose are both false(The game is not win or lose) else return false.

+ long getMinute()	Return timeLeft in minutes.
+ long getSecond()	Return timeLeft in seconds.
+ boolean isTimeUp()	Return true if timeLeft is less than 1000000000L(1 second ) else return false.
+ void initailize()	Set value for isWin, isLose, timeLeft, prevTime
+ Thread getTimethread()	Getter method of timeThread.
+ void endTimethread()	Interrupt timeThread.

### 3.11 Package : drawing

#### 3.11.1 Class GameScreen extends Canvas

##### 3.11.1.1 Field

+ double GAMESCREEN_HEIGHT	This is a GameScreen's height.Set it to SceneManager.screenHeight.
+ double GAMESCREEN_WIDTH	This is a GameScreen's width.It is equal to GAMESCREEN_HEIGHT.

##### 3.11.1.2 Constructor

+ GameScreen()	Initialize this class with GAMESCREEN_WIDTH,GAMESCREEN_HEIGHT and set up an event Handler(call addListener()).
----------------	--

##### 3.11.1.3 Method

+void paintComponent()	Draw all entity in entities in RenderableHolder by using this class(This class extends Canvas,so can use this class's GraphicsContext).
+void addListener()	The method collect all GameScreen's event handler. If the key is pressed send the pressed key to InputUtility. If the key is released tell InputUtility to release the key. If the left mouse is clicked tell InputUtility that left moused is clicked. If the left mouse is released tell

	<p>InputUtility that left moused is released.</p> <p>If the cursor is in GameScreen tell InputUtility that it's current UI is "GAME".</p> <p>If the cursor is moved send X,Y cursor to InputUtility.</p> <p>If the cursor is dragged send X,Y cursor to InputUtility.</p>
<u>+boolean isInGameScreen()</u>	If cursor is in GameScreen return true, otherwise return false.

### 3.11.2 Class LoseScreen extends Pane

#### 3.11.2.1 Field

- ExitButton exit	This is an exit button.
- RetryButton retry	This is an retry button(Play the Stage you have just played again).
- MenuButton menu	This is an menu button(Return to main menu).
- Label title	Title of Lose Screen.
- String LOSE_TEXT	String that will be be in Label title. Default is "YOU LOSE".

#### 3.11.2.2 Constructor

<u>+ LoseScreen()</u>	<p>In short: Stop all audio and draw Screen like in Figure 5).</p> <p>Stop all audio.</p> <p>Play RenderableHolder.loseSong.</p> <p>Set screen size to width= SceneManager.screenWidth, and height = SceneManager.screenHeight.</p> <p>Set background of this Pane to RenderableHolder.loseBackground.</p> <p>Initialize retry with string "Retry".</p> <p>Relocate retry to</p>
-----------------------	--

	<p>x=(SceneManager.screenWidth-NormalButton.BUTTON_WIDTH)*1.5/2,y=SceneManager.screenHeight*0.4.</p> <p>Initilialize menu with string "Main Menu".</p> <p>Relocate menu to x=(SceneManager.screenWidth-NormalButton.BUTTON_WIDTH)*1.5/2,y=SceneManager.screenHeight*0.6.</p> <p>Initilialize exit with string "exit".</p> <p>Relocate exit to x=(SceneManager.screenWidth-NormalButton.BUTTON_WIDTH)*1.5/2,y=SceneManager.screenHeight*0.8.</p> <p>Initilialize title with string LOSE_TEXT.</p> <p>Set title's font to RenderableHolder.screenTextFont.</p> <p>Set title's font's color to red.</p> <p>Relocate title to x=center of screen in x axis ,y=SceneManager.screenHeight*0.2.</p> <p>Add menu,retry,exit,title to the pane.</p>
--	--

### 3.11.3 Class Menubar extends VBox

#### 3.11.3.1 Field

<u>+double MENU_WIDTH</u>	Width of Menubar.
<u>+double MENU_HEIGHT</u>	Height of Menubar.
<u>-int VTAB</u>	Number of Tab in Vertical.Default is 4.
<u>-int HTAB</u>	Number of Tab in Horizontal.Default is 2.
<u>+double ICONPOS</u>	Starting position of icon tab int yaxis.Deault is MENU_HEIGHT*0.25.
<u>+double ICONHEIGHT</u>	Height of each icon.Default is (MENU_HEIGHT-ICONPOS)/VTAB.
<u>+double ICONWIDTH</u>	Width of each icon.Default is MENU_WIDTH/HTAB.
<u>+Menubar instance</u>	Singleton of this class.

-Canvas[][] icon	Canvas of all icon. icon[0][0] stands for Peasant. icon[0][1] stands for Footman. icon[1][0] stands for Wardog. icon[1][1] stands for Berserker. icon[2][0] stands for Sapper. icon[2][1] stands for Saboteur. icon[3][0] stands for General. icon[3][1] stands for Reset(Reset the game).
-GridPane gridPane	Pane that collect all icon.
-int choseRow	Current row of chose icon.Default is -1.
-int choseColumn	Current column of chose icon.Default is -1.
-GameState gameState	A GameState.
-Canvas menuCanvas	Canvas of menu's information tab.
-boolean isReset	Determine whether reset button is clicked. Default is false.
-Homelcon homelcon	A Homelcon.
-ExitIcon exitIcon	A ExitIcon.
-Pauselcon pauselcon	A Pauselcon.

### 3.11.3.2 Constructor

+ Menubar()	Initialize instance,icon,gameState,then set Menubar(call setMenuBar()).
-------------	---

### 3.11.3.3 Method

-void setMenuBar()	set menu information tab and icon(call setMenuTab() and setIcon()).
-void setIcon()	Initialize GridPane,then initialize each canvas and set each canvas like Figure d.2),add an event handler for each icon then add them to gridPane. Add homelcon,exitIcon,pauselcon like Figure d.1) by add these three icon into this Pane then 1) relocate homelcon to

	<p>x=MENU_WIDTH*0.8,y= 0.</p> <p>2) relocate homelcon to x=MENU_WIDTH*0.8,y=ICONPOS*0.8/3.</p> <p>3) relocate homelcon to x=MENU_WIDTH*0.8,y=ICONPOS*0.8*2/3</p>
-void fillText(GraphicsContext gc,int i, int j)	Fill all text and image in each icon like each icon in Figure d.2).
-void checkEvent(Canvas canvas, int row, int column)	<p>This method is event handler for each icon in Figure d.2).</p> <p>If the icon is being hovered increase icon's brightness by 0.2 like in Figure e)and play RenderableHolder.buttonHover sound.</p> <p>If the icon is being unhovered reset icon's brightness to default.</p> <p>if we click the already clicked icon do nothing,otherwise change the choseRow and choseColumn to current clicked icon then set clicked image to clicked icon and set unclicked image to previous icon.</p>
-void setHover(GraphicsContext gc,int row,int column)	Set hoverd icon's brightness to 0.2 and play RenderableHolder.buttonHover sound.
-void setUnHover(GraphicsContext gc,int row,int column)	Revert hoverd icon's brightness default.
-void chooseCurrentBot(int row, int column)	Send clicked icon unit's type to InputUtility.currentChosed, the clicked icon is Reset set isReset to true, otherwise set InputUtility.currentChosed to "x"(Not choosing any icon").
-void setClick(GraphicsContext gc,int row,int column)	Play RenderableHolder.clickedSound sound and change icon's image to be like in Figure f).
-void setUnClick(GraphicsContext gc,int row,int column)	set icon's image to default(Like icon in Figure d.1)).
-void paintMenuCanvas(GraphicsContext gc)	Paint menuCanvas like in Figure d.3) but don't fill money,stage,time text.

-void setMenuTab()	Initialize menuCanvas then paint menuCanvas like in Figure d.3) and Figure d.4) but don't fill money,stage,time text,then add menuCanvas to this class's Pane.
+void update()	If the stage is cleared set isWin in gameState to true.Start a thread that fill money,stage,time text in menuCanvas like in Figure d.3).
+GameState getGameState()	Getter for gameState.
+boolean isReset()	Getter for isReset.
+void setDefault()	Reinitialize gameState. Set isRese,choseRow,choseColumn to default. Call update(). Unclick all icon(Beware for ArrayIndexOutOfBoundsException is noting os chose).
+void writePoor(String text)	This method is called from PoorException. It show text that is sent from PoorException for 0.5 sceond by starting a thread.



Figure d) A menubar contains  
 Figure d.1) Other three icon   Figure d.2) All icon in gridPane  
 Figure d.3) Menu information tab   Figure d.4)errorFrame



Figure e) A hovered icon



Figure f) A clicked icon

### 3.11.4 Class PlayScreen extends HBox

#### 3.11.4.1 Field

- boolean pausedState	It tell whether the game is pause or not.
+ double MISCSCREENWIDTH	Width of miscScreen and miscScreen2. Set it to (SceneManager.screenWidth-(GameScreen.GAMESCREEN_WIDTH+Menubar.MENU_WIDTH))/2.
+ double MISCSCREENHEIGHT	Height of miscScreen and miscScreen2. Set it to (SceneManager.screenHeight).
+ PlayScreen instance	PlayScreen's singleton.
- GameScreen gameScreen	A GameScreen.
- Menubar menuBar	A Menubar.
- AnimationTimer playThread	A thread for playing normal animation.
- AnimationTimer pauseThread	A thread for playing animation when paused.
- Gamelogic gameLogic	A Gamelogic .
- Canvas miscScreen	A miscScreen.
- Canvas miscScreen2	Another miscScreen.

#### 3.11.4.2 Constructor

+ PlayScreen()	Initialize instances. call initialize() Start a playThread that in this thread it will do the following steps
----------------	---

	<ol style="list-style-type: none"> <li>1) Update gameLogic.</li> <li>2) Paint the screen.</li> <li>3) Increase Tick in InputUtility then check whether the game is paused.</li> <li>4) check whether the game is end(The default state is player isn't win or lose).</li> <li>5) If reset icon is clicked reset the game.</li> <li>6) If the game is end go to another screen.</li> </ol> <p>Initialize a pauseThread that in this thread it will do the following steps</p> <ol style="list-style-type: none"> <li>1) update menuBar.</li> <li>2) Check whether the game is paused or won or lose.</li> <li>3) If reset icon is clicked reset the game.</li> </ol>
--	---

### 3.11.4.3 Method

-void playStateCheck()	Check whether the game is paused or won or lose.
- void initialize()	Initialize gameScreen,menuBar,miscScreen,miscScreen2 ,gameLogic and add gameScreen,menuBar,miscScreen,miscScreen2 into this class's pane.
- void fillMiscreen()	Fill a miscScreen and miscScreen2 like Figure g) (Draw a canvas with w=MISCSCREENWIDTH,h=MISCSCREENHEIGHT then fill the canvas with black color).
- void resetGame()	Stop all sound. Clear all entities. Reinitialize a gameLogic; Stop pauseThread,Start playThread. Set pausedState to false,then draw unpausePauselcon. Reset InputUtility then set menuBar to deafult.
- void paintUpdated()	Tell gameScreen to paint all component and update RenderableHolder and menuBar.

- void checkCondition()	If Board tell us that the game is win set gameState's isWin to true. If all Attacker is killed and you don't have money to buy any Attacker or the time is up set gameState's isLose to true.
- void checkEnd()	If the current stage is cleared do the following steps 1) Stop call sound. 2) add defaultNumBoard by 1 3) Stop the time. if current stage is last stage call forceEnd() then go to WinScreen, otherwise reset the game call forceEnd() then go to WaitScreen. If you lose the current stage do the following steps 1) Stop call sound. 2) Stop the time. 3) Reset the game,call forceEnd() then go to WaitScreen.
+ void forceEnd()	Stop playThread and clear all entities in RenderableHolder.
- void checkPaused()	If SPACEBAR is pressed(not hold) pause the game and set InputUtility.lastKey to null.
+ void pause()	If pausedState is false(the game is not paused) call Pauselcon drawPaused(),then set pausedState to true,stop playThread and start pauseThread. Otherwise(the game has already been paused) call Pauselcon drawUnpaused() ,then set pausedState to false,stop pauseThread and start playThread.
+ boolean isPausedstate()	Getter method of pausedState
+ GameScreen getGameScreen()	Getter method of gameScreen

+ AnimationTimer getPlayThread()

Getter method of playThread

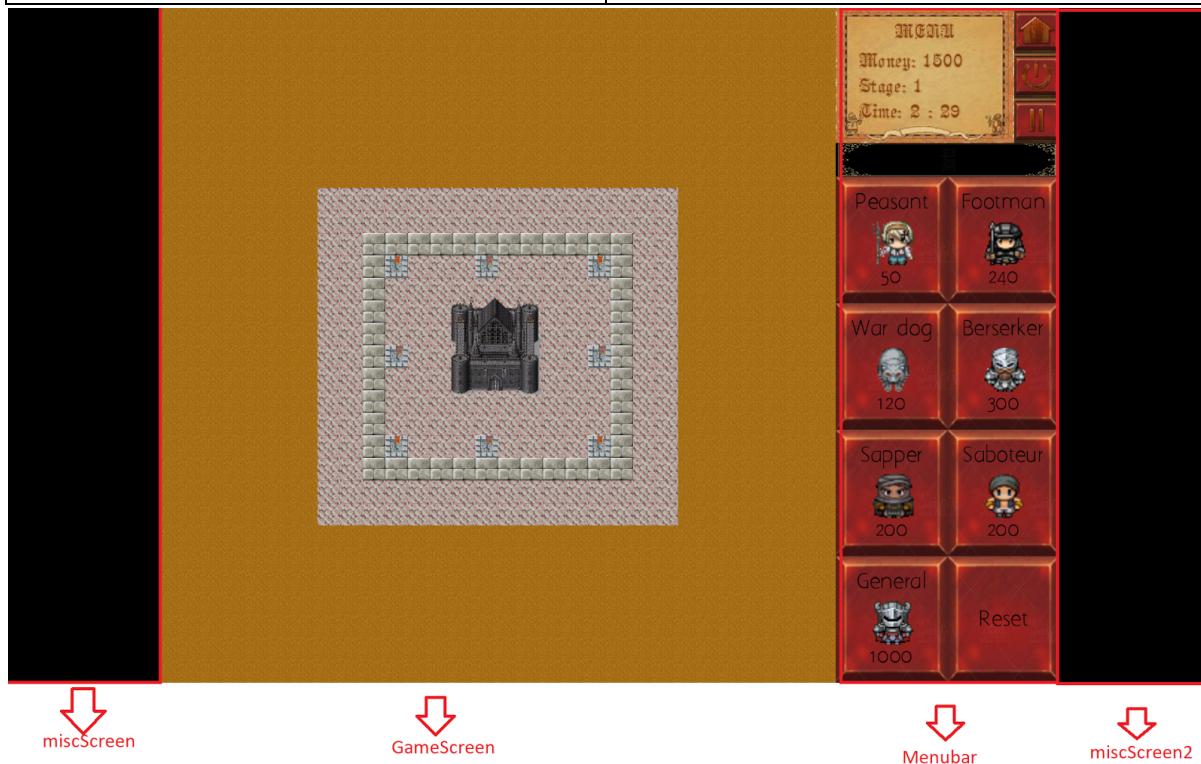


Figure g) A PlayScreen

### 3.11.5 Class StartScreen extends Pane

#### 3.11.5.1 Field

- ExitButton exit	This is an exit button.
- CharacterButton character	This is a character button(Go to the character describe page).
- PlayButton play	This is an play button(Play the first stage).
- Label gameTitle	Title of Start Screen.

#### 3.11.5.2 Constructor

+ StartScreen()

In short:Stop all audio and draw Screen like in Figure 1). Stop all audio.

	<p>Play RenderableHolder.openSong.</p> <p>Set screen size to width= SceneManager.screenWidth, and height = SceneManager.screenHeight.</p> <p>Set background of this Pane to RenderableHolder.background.</p> <p>Initilialize play with string "Play".</p> <p>Relocate play to x=(SceneManager.screenWidth-NormalButton.BUTTON_WIDTH)/2,y=SceneManager.screenHeight*0.4.</p> <p>Initilialize character with string "Character".</p> <p>Relocate character to x=(SceneManager.screenWidth-NormalButton.BUTTON_WIDTH)/2,y=SceneManager.screenHeight*0.6.</p> <p>Initilialize exit with string "Exit".</p> <p>Relocate exit to x=(SceneManager.screenWidth-NormalButton.BUTTON_WIDTH)/2,y=SceneManager.screenHeight*0.8.</p> <p>Initilialize gameTitle with string from SceneManager.GAMENAME.</p> <p>Set title's font to RenderableHolder.screenTextFont.</p> <p>Set title's font's color to red.</p> <p>Relocate title to x=center of screen in x axis ,y=SceneManager.screenHeight*0.2.</p> <p>Add play,character,exit,gameTitle to the pane.</p>
--	--

### 3.11.6 Class WaitScreen extends Pane

#### 3.11.6.1 Field

- ExitButton exit	This is an exit button.
- ReplayButton replay	This is an replay button(Play the Stage you have just played again).
- PlayButton play	This is an play button(Play next Stage).

- Label title	Title of Wait Screen.
- String WAIT_TEXT	String that will be be in Label titie. Default is "YOU WIN".

### 3.11.6.2 Constructor

+ WaitScreen()	<p>In short:Stop all audio and draw Screen like in Figure 4).</p> <p>Stop all audio.</p> <p>Play RenderableHolder.victory sound.</p> <p>Set screen size to width= SceneManager.screenWidth, and height = SceneManager.screenHeight.</p> <p>Set background of this Pane to RenderableHolder.victoryBackground.</p> <p>Initilialize play with string "Next Stage".</p> <p>Relocate play to  <math>x=(SceneManager.screenWidth-NormalButton.BUTTON_WIDTH)*1.5/2, y=SceneManager.screenHeight*0.4</math>.</p> <p>Initilialize replay with string "Play Again".</p> <p>Relocate replay to  <math>x=(SceneManager.screenWidth-NormalButton.BUTTON_WIDTH)*1.5/2, y=SceneManager.screenHeight*0.6</math>.</p> <p>Initilialize exit with string "Exit".</p> <p>Relocate exit to  <math>x=(SceneManager.screenWidth-NormalButton.BUTTON_WIDTH)*1.5/2, y=SceneManager.screenHeight*0.8</math>.</p> <p>Initilialize title with string WAIT_TEXT.</p> <p>Set title's font to RenderableHolder.screenTextFont.</p> <p>Set title's font's color to red.</p> <p>Relocate title to x=center of screen in x axis  <math>,y=SceneManager.screenHeight*0.2</math>.</p> <p>Add play,replay,exit,title to the pane.</p>
----------------	---

### 3.11.7 Class WinScreen extends Pane

#### 3.11.7.1 Field

- ExitButton exit	This is an exit button.
- ReplayButton replay	This is an replay button(Play the Stage you have just played again).
- MenuButton menu	This is an menu button(Return to main menu).
- Label title	Title of Win Screen.
- String WIN_TEXT	String that will be be in Label titie. Default is "YOU CLEAR THE GAME!!!";

### 3.11.7.2 Constructor

+ WinScreen()	<p><b>In short:</b>Stop all audio and draw Screen like in Figure 6).</p> <p>Stop all audio.</p> <p>Play RenderableHolder.victory sound.</p> <p>Set screen size to width= SceneManager.screenWidth, and height = SceneManager.screenHeight.</p> <p>Set background of this Pane to RenderableHolder.victoryBackground.</p> <p>Initilialize replay with string "Play Again".</p> <p>Relocate replay to x=(SceneManager.screenWidth-NormalButton.BUTTON_WIDTH)*1.5/2,y=SceneManager.screenHeight*0.4.</p> <p>Initilialize menu with string "Main Menu".</p> <p>Relocate menu to x=(SceneManager.screenWidth-NormalButton.BUTTON_WIDTH)*1.5/2,y=SceneManager.screenHeight*0.6.</p> <p>Initilialize exit with string "exit".</p> <p>Relocate exit to x=(SceneManager.screenWidth-NormalButton.BUTTON_WIDTH)*1.5/2,y=SceneManager.screenHeight*0.8.</p> <p>Initilialize title with string WIN_TEXT.</p> <p>Set title's font to RenderableHolder.screenTextFont.</p>
---------------	--

	<p>Set title's font's color to red.          Relocate title to x=center of screen in x axis  <math>,y=SceneManager.screenHeight*0.2</math>.          Add menu,replay,exit,title to the pane.</p>
--	--

### 3.11.7 Class CharacterScreen extends Pane

#### 3.11.6.1 Field

- PlayButton play	This is an play button(Play next Stage).
- MenuButton menu	This is an menu button(Go to main menu)
- Label title	Title of Character Screen.
- Label characterName[]	Name of characters
- Label characterDescribeText[]	Description text of characters
- String HEADERTEXT	String that will be be in Label titie.
- Canvas characterImageSet	Canvas for drawing images.

#### 3.11.6.2 Constructor

+ CharacterDescriptionScreen()	<b>In short:</b> Stop all audio and draw Screen like in Figure 2). Stop all audio. Set screen size to width= $SceneManager.screenWidth$ ,and height = $SceneManager.screenHeight$ . Set background of this Pane to RenderableHolder.background. Initialize play with string "Play". Relocate play to $x=(SceneManager.screenWidth-NormalButton.BUTTON_WIDTH)*0.5/2$
--------------------------------	--

```
,y=SceneManager.screenHeight*0.8.  
Initilialize menu with string "Main  
Menu".  
Relocate menu to  
x=(SceneManager.screenWidth-NormalB  
utton.BUTTON_WIDTH)*1.5/2,y=SceneM  
anager.screenHeight*0.8.  
Initialize and relocate  
characterName[]  
Initilialize title with string headerTEXT.  
Set title's font to  
RenderableHolder.screenTextFont.  
Set title's font's color to red.  
Relocate title to x=center of screen in  
x axis  
,y=SceneManager.screenHeight*0.05.  
Initilie characterImage and draw  
characterImage all character right next  
to every characterName[].  
//  
Add characterImageSet,menu,play,title  
to the pane.
```