

# **Rat Defender**

Created by

6631365021 Inthawut Vongjittuttikrai

2110215 Programming Methodology  
Semester 2 Year 2023 Chulalongkorn University

# **Presentation's Video**

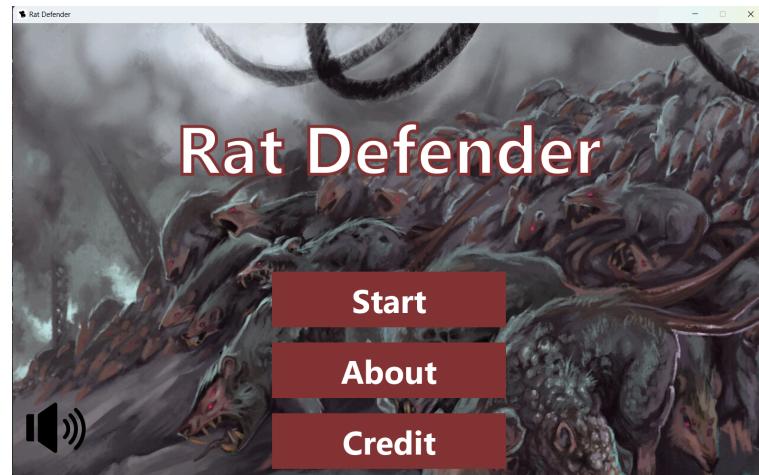
## **Introduction**

Rat Defender is a game that was inspired from the popular tower-defense game, “Plant Vs Zombies”. In Rat Defender you are playing as the witch doctors’ side where you want to protect yourselves from getting infected by the rats.

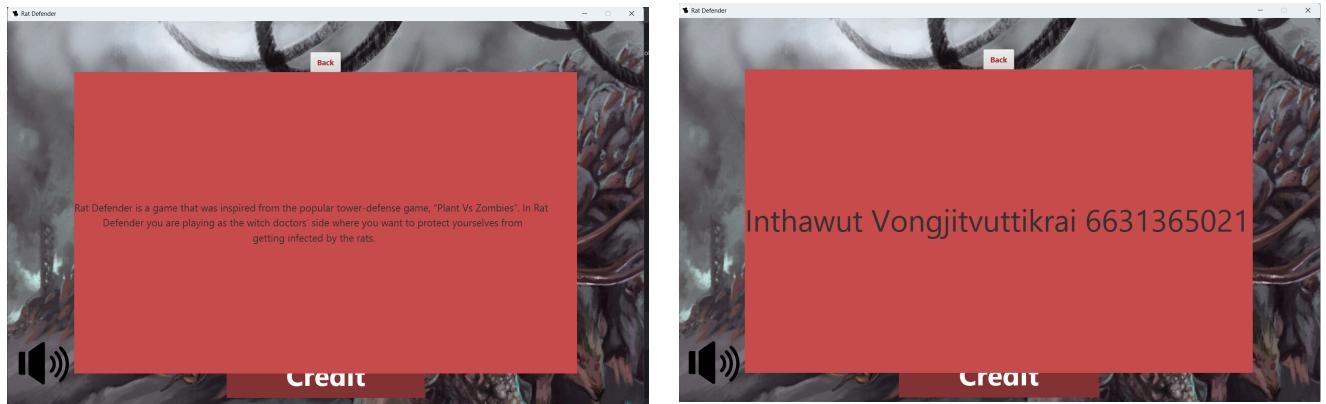
## **Rules**

The game screen is divided into squares where you can place your units, one on each square. There are three different types of units you can place which are producers who make you souls (game’s currency), shooters who attack from afar ,and melee attackers who attack close up enemies. The enemies or rats will come through each horizontal lane. Only the units in that lane can attack or be attacked by the rats. Attacking the rats will reduce their health and eventually kill them. You win by getting rid of all the walking rats and you will lose by letting the rat get past your field.

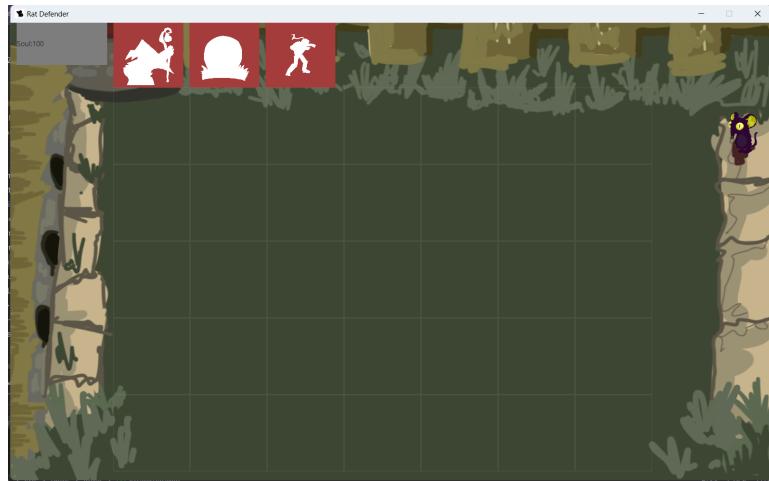
# Example



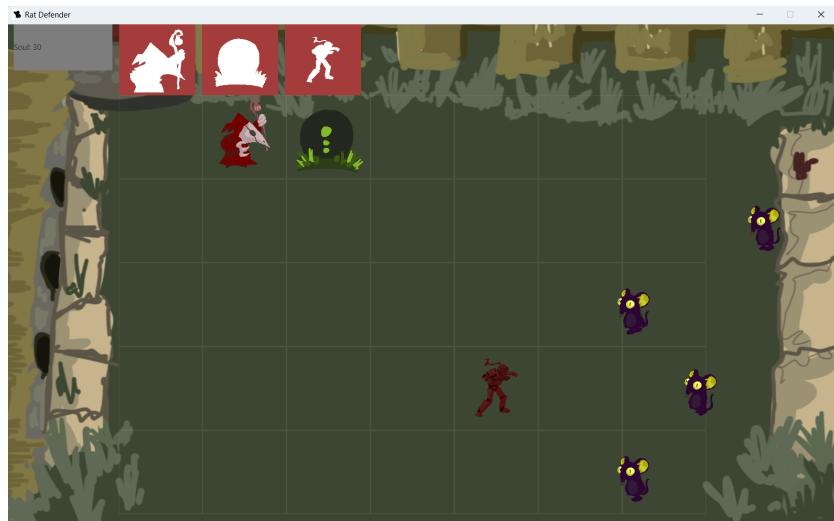
You can press start to start a game, about to show about page and credit to show credit page. You can also stop the background music by clicking the sound icon in the bottom left corner.



When you press the about or credit button a page will pop up with a back button which can be pressed to go back.



When you press the play button you will be greeted by the field. You can select your unit to play at the top while the rats slowly walk toward you.

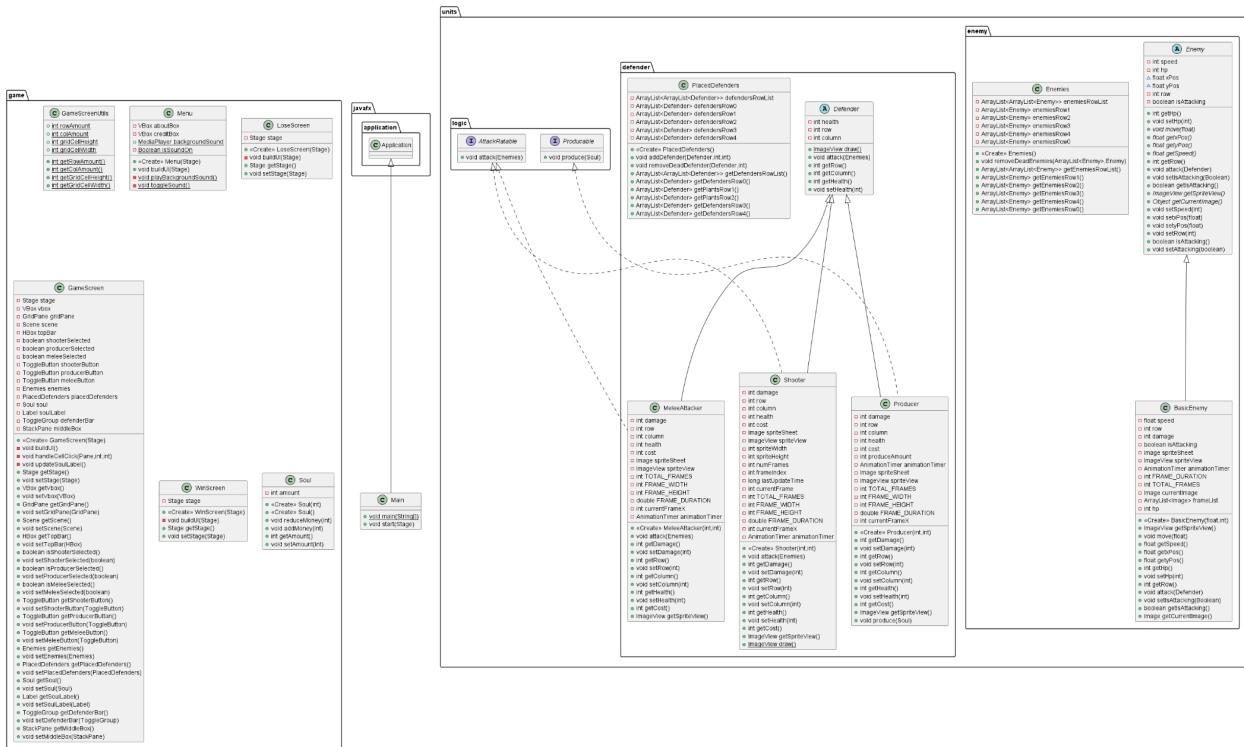


Build your defense however you like on the squares.



However, whenever a rat get passed your defense, you will lose. You can still go back to Main Menu though.

# Class Diagram



## 1.Package game

## 1.1 class GameScreen

### 1.1.1 Fields

- Stage stage	Current stage
- VBox vbox	For holding topbar and middleBox
- GridPane gridPane	Pane with cells where you can

	place units
- Scene scene	Current scene
- HBox topBar	For holding currency and unit buttons
- boolean shooterSelected	Check if shooter is selected
- boolean producerSelected	Check if producer is selected
- boolean meleeSelected	Check if melee attacker is selected
- ToggleButton shooterButton	Button for selecting shooter
- ToggleButton producerButton	Button for selecting producer
- ToggleButton meleeButton	Button for selecting melee attacker
- Enemies enemies	List of lists of enemies in each row
- PlacedDefenders placedDefenders	List of lists of defenders in each row
- Soul soul	Soul
- Label soulLabel	Soul label for showing its amount
- ToggleGroup defenderBar	Togglegroup for each defender's button
- StackPane middleBox	A stackpane for holding gridPane

### 1.1.2 Constructor

+ GameScreen(Stage stage)	Initialize stage Use method buildUI()
---------------------------	--

### 1.1.3 Methods

+ void buidUI	Initialize all fields
---------------	-----------------------

- Initialize the column constraints to GridCellWidth and row constraints to GridCellHeight.
- Add the constraints to all columns and rows.
- Creating Pane called cell in every space created from intersection then set its border to grey with low opacity.
- Set on mouse clicked event to use method handleCellClicked
- Set gridPane alignment to top left
- Set topBar preferred height to 100px
- Create shooter button from shooterLogo.png with fit height of 100px and preserving its ratio
- Set on mouse entered and exited to changes the shooter button's color.
- Create producer button from producerLogo.png with fit height of 100px and preserving its ratio
- Set on mouse entered and exited to changes the producer button's color.
- Create melee button from meleeLogo.png with fit height of 100px and preserving its ratio
- Set on mouse entered and exited to changes the melee button's color.

- Add each of the buttons to defenderBar toggle group.
- Set soulLabel's background to grey, width to 150px and height to 70px
- Add soulLabel to topBar
- Add shooterButton to topBar
- Add producerButton to topBar
- Add meleeButton to topBar
- Set each of their margin in topBar
- Add gridPane to middleBox
- Add topBar and middleBox to vbox
- Create backgroundView with gamescreenbg.png
- Addlistener to selected property for each defender button
- Create multiple new BasicEnemy with random rows from 0 to 4
- Add them to each their respective rows
- Create canvas and set mouse transparent to true
- Initialize StackPane root
- Initialize move thread where it draw enemy in new position on canvas and clear old ones after using method move()
- Initialize prevX and prevY for ratInRow
- In moveThread, if the enemy is at the position of a defender, it will stop moving and start attacking when it's done attacking it will start moving again

- Create a new ArrayList for enemy who dies(health reaches 0) and waiting to be removed
- Create a new ArrayList for defender who dies(health reaches 0) and waiting to be removed
- Remove the dead enemies
- Remove the dead defenders
- if the enemies are all dead, wait 1000 ms and then go to winScreen
- If one enemy reach the far left side of the screen, go to loseScreen
- Sleep the moveThread for 20 ms
- In shootThread, loop for each plant that is an instanceof Attackable call attack(enemies) method
- Sleep the attackThread for 100 ms
- Initialize soulThread
- In soulThread, loop for each plant that is an instanceof producable and call method produce() then call updateSoulLabel()
- Sleep the soulThread for 2300 ms
- Start each thread
- Add backgroundView vbox and canvas to root
- Create new scene with root
- Set stage's min height to 800 and min width to 1280
- Show stage

- void handleCellClick(Pane grid, int row, int column)	- If any of the defender is selected and soul amount is enough, defender's sprite will be added to the cell
- void updateSoulLabel()	- Update soulLabel with current amount of soul amount
+ getter of all fields	
+ setter of all fields	

## 1.2 class GameScreenUtils

### 1.2.1 Fields

+ <u>int rowAmount</u>	Set the value to 5
+ <u>Int colAmount</u>	Set the value to 7
+ <u>int gridCellHeight</u>	Set the value to 128
+ <u>int gridCellWidth</u>	Set the value to 128

### 1.2.3 Methods

+ getter of all fields	
------------------------	--

## 1.3 class LoseScreen

### 1.3.1 Fields

- Stage stage	Current stage
---------------	---------------

### 1.3.2 Constructor

+ LoseScreen(Stage stage)	Set this.stage to stage Call method buildUI
---------------------------	--

### 1.3.3 Methods

+ void buidUI	<ul style="list-style-type: none"> <li>- Initialize StackPane root</li> <li>- Initialize Button menuButton</li> <li>- Initialize Text menuText</li> <li>- Set menuText's font to Inter, BOLD with the size of 55px</li> <li>- Set menuText's color to white</li> <li>- Set menuButton graphic to menuText</li> <li>- Set menuButton background color</li> <li>- Set menuButton handle on click to change the scene</li> <li>- Set menuButton handle mouse entered and exited to change the button's color</li> <li>- Create ImageView for background</li> <li>- Create youLost text</li> <li>- Change the youLost text's properties</li> <li>- Initialize VBox textBox then add youLost text</li> <li>- Add textBox, menuButton to root</li> <li>- Set root to scene</li> <li>- Set stage's scene</li> <li>- Set stage's width and height to 1280px and 800px</li> <li>- Show stage</li> </ul>
+ getter of all fields	
+ setter of all fields	

## 1.4 class WinScreen

### 1.4.1 Fields

- Stage stage	Current stage
---------------	---------------

### 1.4.2 Constructor

+ Lose(Stage stage)	Set this.stage to stage Call method buildUI
---------------------	--

### 1.4.3 Methods

+ void buidUI	<ul style="list-style-type: none"><li>- Initialize StackPane root</li><li>- Initialize Button menuButton</li><li>- Initialize Text menuText</li><li>- Set menuText's font to Inter, BOLD with the size of 55px</li><li>- Set menuText's color to white</li><li>- Set menuButton graphic to menuText</li><li>- Set menuButton background color</li><li>- Set menuButton handle on click to change the scene</li><li>- Set menuButton handle mouse entered and exited to change the button's color</li><li>- Create ImageView for background</li><li>- Create youWon text</li><li>- Change the youWon text's properties</li><li>- Initialize VBox textBox then add youWon text</li><li>- Add textBox, menuButton to root</li><li>- Set root to scene</li></ul>
---------------	--

	<ul style="list-style-type: none"> <li>- Set stage's scene</li> <li>- Set stage's width and height to 1280px and 800px</li> <li>- Show stage</li> </ul>
+ getter of all fields	
+ setter of all fields	

## 1.5 class Menu

### 1.5.1 Fields

- VBox aboutBox	Box for about page
- VBox creditBox	Box for credit page
+ <u>MediaPlayer</u> <u>backgroundSound</u>	Background sound
- <u>boolean isSoundOn</u>	Boolean for background sound(on/off)

### 1.5.2 Constructor

+ Menu(Stage stage)	Call buildUI(stage)
---------------------	---------------------

### 1.5.3 Methods

+ void buildUI(Stage stage)	<ul style="list-style-type: none"> <li>- Initialize StackPane root</li> <li>- Initialize Button playButton</li> <li>- Initialize Button aboutButton with aboutBox</li> <li>- Initialize Button creditButton with creditBox</li> <li>- Set each button to have</li> </ul>
-----------------------------	--

	<ul style="list-style-type: none"> <li>- mouseEntered and mouseExited</li> <li>- Set playButton on pressed to change scene</li> <li>- Initialize Button backButton</li> <li>- Set aboutButton and credit button to show about and credit with backButton</li> <li>- Set backButton to hide aboutBox and creditBox</li> <li>- Add each button to root</li> </ul>
- <u>void playBackgroundSound()</u>	<ul style="list-style-type: none"> <li>- Set the path of MediaPlayer to background music path</li> <li>- Assign MediaPlayer to backgroundSound</li> <li>- Start playing backgroundSound</li> </ul>
- <u>Void toggleSound()</u>	<ul style="list-style-type: none"> <li>- If isSoundOn is true then stop playing backgroundSound</li> <li>- If isSoundOn is false then stop playing backgroundSound</li> </ul>
+ getter of all fields	
+ setter of all fields	

## 1.6 class Soul

### 1.6.1 Fields

- int amount	Soul amount
--------------	-------------

### 1.6.2 Constructor

+ Soul(int startSoul)	amount = startSoul
+ Soul()	amount = 100

### 1.6.3 Methods

+ void reduceMoney(int reduceAmount)	Reduce amount by reduceAmount
+ Void addMoney(int increaseAmount)	Increase amount by increaseAmount
+ getter of all fields	
+ setter of all fields	

## 2.Package units.enemy

### 2.1 abstract class Enemy

#### 2.1.1 Fields

- int speed	enemy's speed
- int hp	enemy's hp
- float xPos	enemy's x position
- float yPos	enemy's y position
- int row	enemy's row
- boolean isAttacking	enemy's attacking status

#### 2.1.2 Methods

+ void abstract move(float speed)	Move the enemy
+ void attack(Plant plant)	Attack the plant
+ getter of all fields	
+ setter of all fields	

### 2.2 class BasicEnemy extends Enemy

#### 2.2.1 Fields

- final float speed	speed = 3
- int row	enemy's row
- Int damage	Set enemy's damage to 1
- boolean is Attacking	enemy's attacking status
- Image spriteSheet	enemy's spritesheet
- ImageView spriteView	enemy's spritesheet as image view
- AnimationTimer animationTimer	enemy's animation timer
- int FRAME_DURATION	Frame duration = 20
- int TOTAL_FRAMES	Total frames = 7
- Image currentImage	enemy's current image
- ArrayList<Image> frameList	ArrayList for images of enemy's frames

## 2.2.2 Constructor

+ BasicEnemy(float x, int row)	<ul style="list-style-type: none"> <li>- Set xPos to x</li> <li>- Set yPos to 150 + row*gridCellHeight</li> <li>- Set isAttacking to false</li> <li>- Initiate frameList</li> <li>- Add frame1 to 7</li> <li>- Set currentImage to frame1</li> <li>- Initiate animationTimer</li> <li>- Set animation to update every FRAME_DURATION *1_000_000 to change the current frame to the next one (if the last frame is used set it to the first)</li> </ul>
--------------------------------	--

### 2.2.3 Methods

+ void move(float speed)	If the enemy is not attacking, reduce the x position by speed and start animationTimer else stop the animationTimer
+ void attack(Plant plant)	Attack the plant by reducing the plant health by the enemy's damage
+ getter of all fields	
+ setter of all fields	

## 2.3 class Enemies

### 2.3.1 Fields

- ArrayList<ArrayList<Enemy>> enemiesRowList	Array list of array list of enemies
- ArrayList<Enemy> enemiesRow1	Array list of enemies in row 1
- ArrayList<Enemy> enemiesRow2	Array list of enemies in row 2
- ArrayList<Enemy> enemiesRow3	Array list of enemies in row 3
- ArrayList<Enemy> enemiesRow4	Array list of enemies in row 4
- ArrayList<Enemy> enemiesRow0	Array list of enemies in row 0

### 2.3.2 Constructor

+ Enemies()	- Initialize enemiesRowList - Initialize enemiesRow1 - Initialize enemiesRow2 - Initialize enemiesRow3 - Initialize enemiesRow4 - Initialize enemiesRow0 - Add enemiesRow0,1,2,3,4 to enemiesRowList
-------------	--

### 2.3.3 Methods

+ void removeDeadEnemies(ArrayList<Enemy> enemyInRow, Enemy enemy)	- Remove dead enemy from the selected row
+ getter of all fields	
+ setter of all fields	

## 3.Package units.logic

### 3.1 Interface AttackRatable

+ void attack(Enemies enemies)	Reduce the enemy in enemy health by damage
--------------------------------	--

### 3.2 Interface Producable

+ void produce(Soul soul)	Increase soul amount by producer's property
---------------------------	---

## 4.Package units.defender

### 4.1 abstract class Defender

#### 4.1.1 Fields

- int health	defender's health
- int row	defender's row
- Int column	defender's column

#### 4.1.2 Methods

+ ImageView draw()	return null
+ void attack(Enemies enemies)	attack an enemy in enemies
+ getter of all fields	
+ setter of all fields	

## 4.2 class MeleeAttacker extends Defender implements AttackRatable

### 4.2.1 Fields

- damage	meleeAttacker's damage
- int row	meleeAttacker's row
- int column	meleeAttacker's column
- int health	meleeAttacker's health
- int cost	cost = 50
- Image spritesheet	meleeAttacker's spriteSheet
- ImageView spriteView	meleeAttacker's spritesheet as image view
- AnimationTimer animationTimer	enemy's animation timer
- final double FRAME_DURATION	Frame duration = 0.1
- int TOTAL_FRAMES	Total frames = 7
- final int FRAME_WIDTH	Frame width = 1400px
- final int FRAME_HEIGHT	Frame height = 1205px
- int currentFrameX	Current x to use in viewport

### 4.2.2 Constructor

+ MeleeAttacker( int row, int column)	- set row's value to row - set column's value to column - set damage to 2 - Set health to 50
---------------------------------------	---

	<ul style="list-style-type: none"> <li>- Initiate spriteSheet to MeleeSpriteSheet.png and set viewport(with frame width and frame height)</li> <li>- Set spriteview fit height to 128 px and preserve its ratio</li> <li>- Initiate animationTimer</li> <li>- Change the viewport to shift to the right by frame width everytime the time has passed by FRAME_DURATION * 1_000_000</li> </ul>
--	---

#### 4.2.3 Methods

+ void attack(Enemies enemies)	If the enemies in the same row is not empty and its x position is right by the meleeAttacker to the right, the enemy's health is reduced by melee attacker's damage
+ getter of all fields	
+ setter of all fields	

## 4.3 class Shooter extends Defender implements AttackRatable

### 4.3.1 Fields

- damage	shooter's damage
- int row	shooter's row
- int column	shooter's column
- int health	shooter's health
- int cost	cost = 50
- Image spritesheet	shooter's spriteSheet
- ImageView spriteView	shooter's spritesheet as image view
- AnimationTimer animationTimer	enemy's animation timer
- final double FRAME_DURATION	Frame duration = 0.1
- int TOTAL_FRAMES	Total frames = 8
- final int FRAME_WIDTH	Frame width = 1200px
- final int FRAME_HEIGHT	Frame height = 1200px
- Int currentFrameX	Current x to use in viewport

### 4.3.2 Constructor

+ Shooter( int row, int column)	<ul style="list-style-type: none"><li>- set row's value to row</li><li>- set column's value to column</li><li>- set damage to 1</li><li>- Set health to 50</li><li>- Initiate spriteSheet to ShooterSpriteSheet.png and</li></ul>
---------------------------------	---

	<p>set viewport(with frame width and frame height)</p> <ul style="list-style-type: none"> <li>- Set spriteview fit height to 128 px and preserve its ratio</li> <li>- Initiate animationTimer</li> <li>- Change the viewport to shift to the right by frame width everytime the time has passed by FRAME_DURATION * 1_000_000_000.0</li> </ul>
--	--

### 4.3.3 Methods

+ void attack(Enemies enemies)	If the enemies in the same row is not empty and its x position is in the screen and to the right of shooter's position, the enemy's health is reduced by shooter's damage
+ getter of all fields	
+ setter of all fields	

## 4.4 class Producer extends Defender implements Producible

### 4.4.1 Fields

- damage	shooter's damage
- int row	shooter's row
- int column	shooter's column
- int health	shooter's health
- int cost	cost = 50
- Image spritesheet	shooter's spriteSheet
- ImageView spriteView	shooter's spritesheet as image view
- AnimationTimer animationTimer	enemy's animation timer
- final double FRAME_DURATION	Frame duration = 0.1
- int TOTAL_FRAMES	Total frames = 8
- final int FRAME_WIDTH	Frame width = 1024px
- final int FRAME_HEIGHT	Frame height = 1640px
- Int currentFrameX	Current x to use in viewport
- Int produceAmount	Set produce amount to 10

### 4.4.2 Constructor

+ Producer( int row, int column)	- set row's value to row - set column's value to column - set damage to 0
----------------------------------	---

	<ul style="list-style-type: none"> <li>- Set health to 50</li> <li>- Initiate spriteSheet to producerSpriteSheet.png and set viewport(with frame width and frame height)</li> <li>- Set spriteview fit height to 128 px and preserve its ratio</li> <li>- Initiate animationTimer</li> <li>- Change the viewport to shift to the right by frame width everytime the time has passed by FRAME_DURATION * 1_000_000_000.0</li> </ul>
--	--

#### 4.4.3 Methods

+ void produce(Soul soul)	Increase the soul amount by produceAmount
+ getter of all fields	
+ setter of all fields	

### 4.5 class Enemies

#### 4.5.1 Fields

- ArrayList<ArrayList<Defender>> defendersRowList	Array list of array list of defenders
- ArrayList<Defender> defendersRow1	Array list of defenders in row 1
- ArrayList<Defender> defendersRow2	Array list of defenders in row 2
- ArrayList<Defender> defendersRow3	Array list of defenders in row 3
- ArrayList<Defender> defendersRow4	Array list of defenders in row 4
- ArrayList<Defender> defendersRow0	Array list of defenders in row 0

#### 4.5.2 Constructor

+ Enemies()	<ul style="list-style-type: none"> <li>- Initialize enemiesRowList</li> <li>- Initialize enemiesRow1</li> <li>- Initialize enemiesRow2</li> <li>- Initialize enemiesRow3</li> <li>- Initialize enemiesRow4</li> <li>- Initialize enemiesRow0</li> <li>- Add enemiesRow0,1,2,3,4 to enemiesRowList</li> </ul>
-------------	--

#### 4.5.3 Methods

+ void removeDeadEnemies(ArrayList<Enemy> enemyInRow, Enemy enemy)	- Remove dead enemy from the selected row
+ getter of all fields	
+ setter of all fields	