Shooting Game

Object Oriented Programming Project Assignment 2018

## Requirement Specification

Project URL: <https://github.com/leonwong24/JavaProject>

Most of the class initialize code are from YouTube tutorial series :  <https://www.youtube.com/watch?v=dEKs-3GhVKQ&list=PLah6faXAgguMnTBs3JnEJY0shAc18XYQZ>

This project is to simulate a survive shooting game. A user has to control the player to keep distance from three type of enemy in the game. Each enemy type has corresponding movement speed, attack rate , and attack strength and will behave differently with others if hit by the bullet.

A user can control the player by using the W,A,S,D key and can shoot out bullet by left clicking the mouse. The bullet has a fixed fire rate and will damage the enemy if it hits the enemy. A player will be given 100 health point and the game will be over if the player died(health point equals or below to 0 ). Each enemy killed will be rewarded with score points, and when the player died (Game over) , the score that the player scored will be shown on the Game Over screen.

Whenever the game is over, the game will load up the high score saved in the highscore.data . If a player broke the high score, the score will be save to the highscore.data thus the previous high score will be replaced.



## Class Diagram

*Due to the complexity of the class project, the following class diagram has been produced by IntelliJ:*

*Each below represent the meaning of the icon in the UML class diagram*

* :Class
* :Class variable/attribute
* :Class defined /declared within the class
* :Static
* :Public
* :Private

*The pdf version of the UML class diagram can be assessed in my GitHub repository JavaProject/classDiagram*

### Overall project class diagram

## 

### Package assets class diagram

## 

### Package entities class diagram

## 

### Package inputs class diagram

## 

### Package main class diagram

## 

### Package states class diagram

## VOPC Diagram

### State Class:

State

+tick()

+render()

+setState()

+getCurrentState()

Graphics

javax.awt

### ImageLoader Class:

ImageIo

javax.imageio

java.awt

BufferedImage

ImageLoader

+loadImage()

IOException

java.io

### Display Class:

Display

+Display()

-createDisplay()

+getCanvas()

+getFrame

JFrame

Canvas

javax.swing.

java.awt

### MouseManager Class:

MouseManager

+MouseManager()

+tick()

+mousePressed()

+mouseReleased()

MouseEvent

MouseListener

java.awt.event

java.awt.event

### KeyManager Class:

KeyManager

+KeyManager()

+tick()

+keyPressed()

+keyReleased()

KeyEvent

KeyListener

java.awt.event

java.awt.event

### Walker Class:

Walker

+Walker()

+tick()

+render()

+getDamage()

-chaseTarget()

+hitByBullet()

Timer

TimerTask

java.util

java.util

### Crawler class:

Crawler

+Walker()

+tick()

+render()

+getDamage()

-chaseTarget()

+hitByBullet()

Timer

TimerTask

java.util

java.util

### SpriteSheet class:

SpriteSheet

+SpriteSheet()

+crop()

BufferedImage

java.awt.image