

2110215 Programing Methodology Project 1/2017

# Last Defense

Developers

Natchanon Arunkeereewat 5931019321

Siwa Boonpunmongkol 5931065121

## 1 Introduction

- This game take place in the very far future, where the Earth is invaded by aliens. You are the last line of defense. If you fail, humanity will fall. You are the last hope!
- There is 1 main character which you can control it up or down and shoot bullet as below.



- There are 3 types of enemy. Red, blue and yellow enemy as below.



- There are 3 types of bullet. Red, blue and yellow bullet.
- To kill enemy, the shot bullet's color must be the same color as enemy.
- If bullet's color and enemy's color are matched, you will get 100 points and that enemy will be eliminated. Else, your score will be subtracted by 50.
- The game's difficulty will increased as your score goes up.
- The game will end if you get hit by an enemy or an enemy reach the left side of screen.

## 2 How to play

There are 4 screens – Main screen / Tutorial screen / Game screen / Score screen

### Main screen

This is the first screen you will see when you launch the game.

There is a HUGE text that tell you the game's title, in case you didn't know.

- ENTER – Start game
- S – Toggle silent mode
- T – View tutorial
- ESCAPE – Exit game

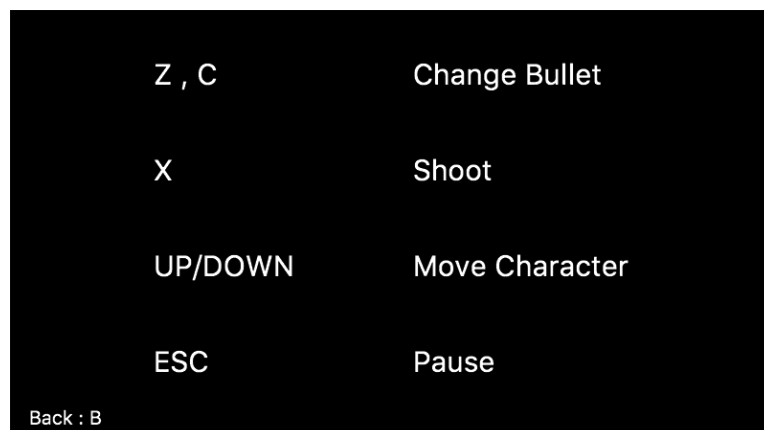


### Tutorial screen

The screen that tell how to play the game. Nothing much to do on this screen.

If you are done reading, just press B to go back to main menu.

- B – Back to main screen
- S – Toggle silent mode



## Game screen

The screen where you play. You will see your character on the left side of the screen.

Enemies will come from the right side of the screen.

The bullet status on the bottom left can tell you what bullet color you are using right now.

On the bottom right, there is your score. Shoot enemy with correct bullet color to get score.

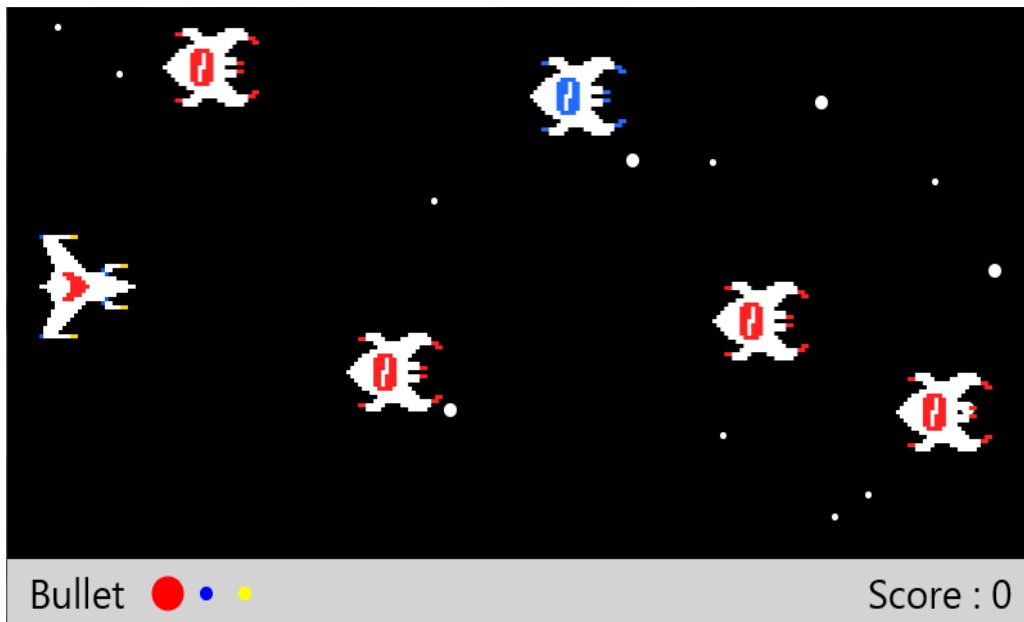
And on the background, there is a DYNAMIC-BREATHTAKING-INEFFABLE-SPACE-BACKGROUND.

The game will end when enemy reach left side or your character hit by enemy.

- UP / DOWN – Move character up and down
- X – Shoot current color bullet
- ESCAPE – Pause game
- Z , C – Change bullet color
- S – Toggle silent mode

When pause

- ENTER – Resume game
- ESCAPE – Exit game

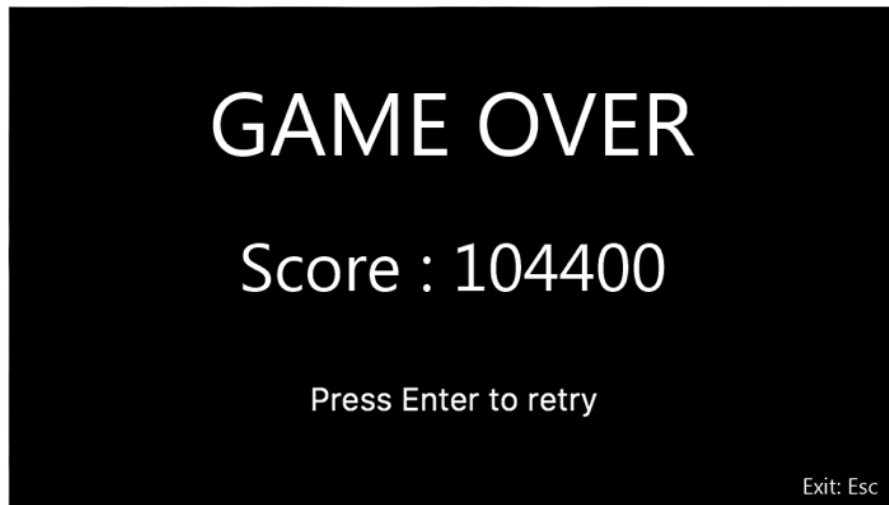


## Score screen

This screen show how much score you got after end game (and, of course, the world is on fire).

There is a very peaceful background music to make you feel calm even when you can't save the world.

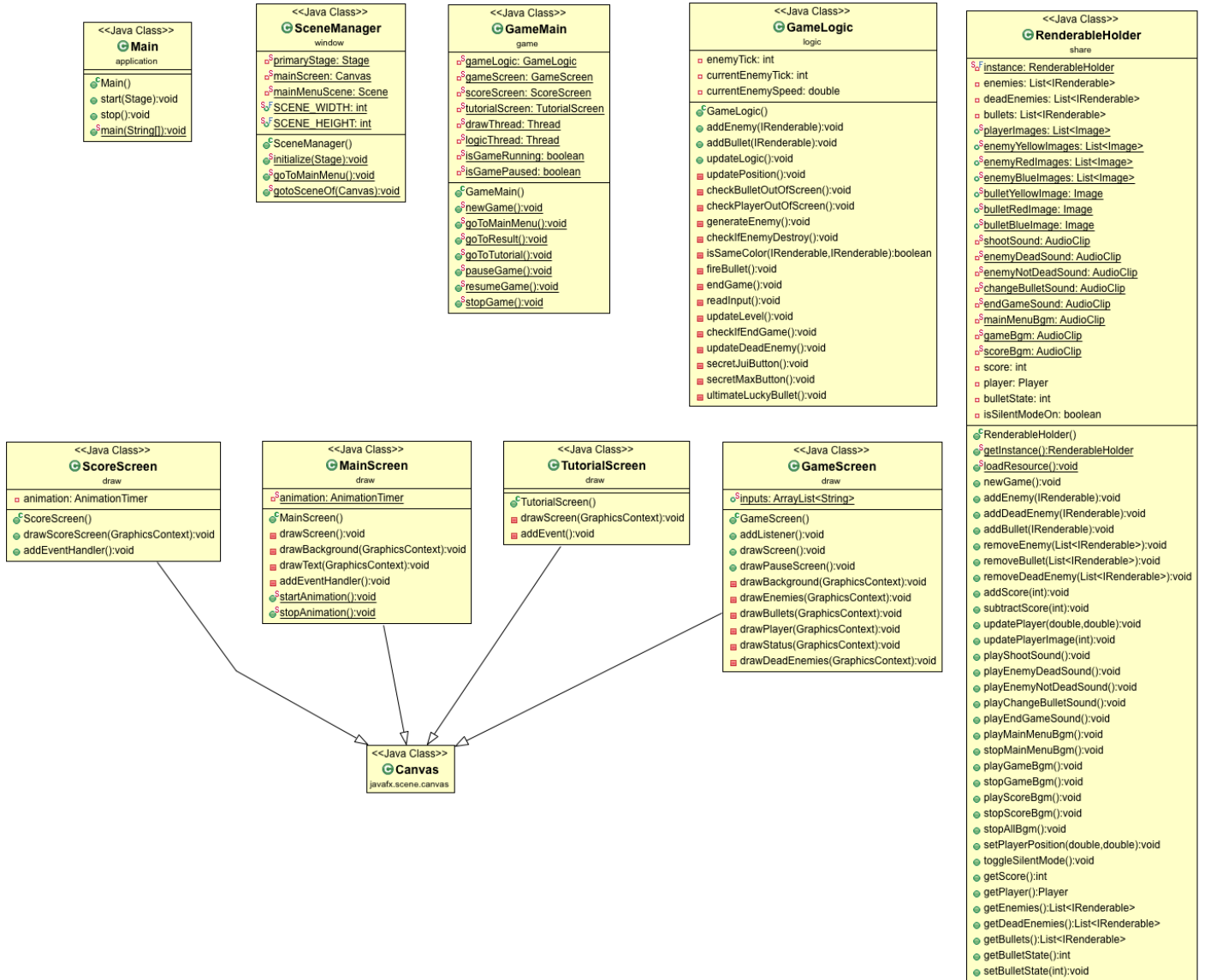
- ENTER – Play again
- S – Toggle silent mode
- ESCAPE – Exit game



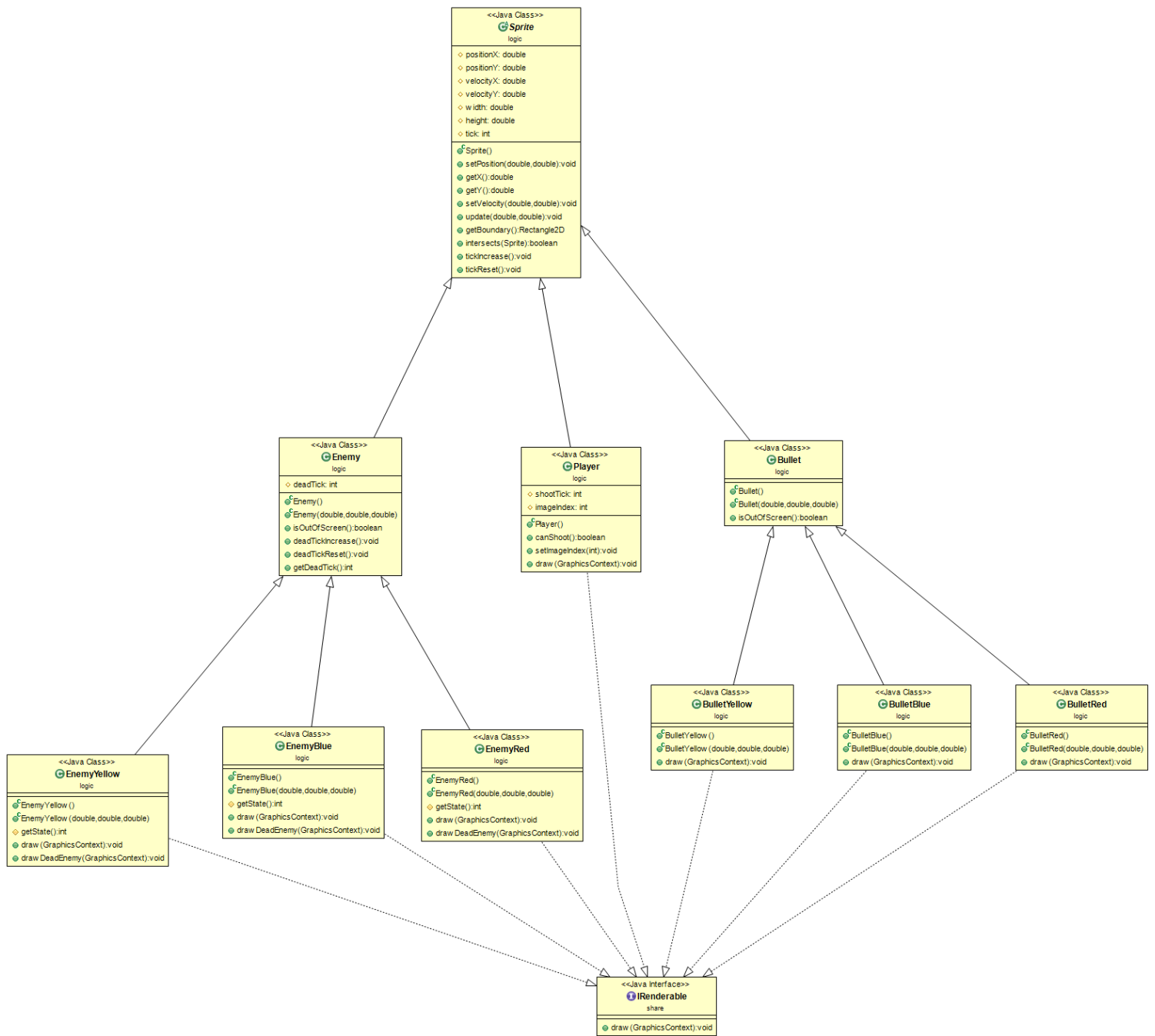
## Tips :

- You can toggle sound effect and BGM on/off by pressing S button anywhere in this game.
  - If you move until you hit the top or bottom edge of screen, your character will appear at another edge.
- This make this game a bit easier. Real man won't use this trick.

### 3 Implementation Detail



UML Diagram (1)



UML Diagram (2)

## 3.1 Package application

### 3.1.1 Class Main extends Application

#### 3.1.1.1 Method

+ void start(Stage stage)	Initialize SceneManager with stage then go to main menu. Set stage's title to "Last Defense" and set center on screen.
+ void stop()	Stop playing all BGM (Call RenderableHolder's method). Stop game (Call GameMain's method). Stop main menu animation (Call MainScreen's method).
+ void main(String[] args)	An entry point of the application.

## 3.2 Package draw

### 3.2.1 Class GameScreen extends Canvas

#### 3.2.1.1 Field

+ ArrayList<String> inputs	List of keys that are pressing and waiting to be read.
----------------------------	--

#### 3.2.1.2 Constructor

+ GameScreen()	Initialize <i>Canvas</i> with width and height from <i>SceneManager</i> . Add listener. (Use method below)
----------------	---

#### 3.2.1.3 Method

+ void addListener()	When any key is pressed, If that key's code is not in inputs, then add to inputs. When any key is released, remove that key's code from inputs.
+ void drawScreen()	Draw game screen. Draw background first, then draw enemies, bullets, player and status bar. (Use methods below)
+ drawPauseScreen()	Draw pause screen. Fill white text "PAUSE" with 35 font size, "Press ENTER to resume" and "Press ESC to exit" with 15 font size.
+ void drawBackground(GraphicsContext gc)	Clear screen with black rectangle. Fill 10 5*5-pixels-ovals and 5 10*10-pixels-ovals. All with random position.
+ void drawEnemies(GraphicsContext gc)	Draw all enemies in <i>enemies</i> list from <i>RenderableHolder</i> .
+ void drawBullets(GraphicsContext gc)	Draw all bullets in <i>bullets</i> list from <i>RenderableHolder</i> .



+ void drawPlayer(GraphicsContext gc)	Draw <i>player</i> from <i>RenderableHolder</i> .
+ void drawStatus(GraphicsContext gc)	Fill (0,400) to (800,450) of screen with light grey color. Fill black text “Bullet” with 30 font size at (17,436). Draw 3 ovals with color red, blue and yellow next to “Bullet” text. Draw oval with current bullet color with 25*25-oval, others with 10*10-oval. Fill black text “Score : ” follows by current score with 30 font size at bottom right of screen.
+ void drawDeadEnemies(GraphicsContext gc)	For each dead enemy in <i>deadEnemies</i> list from <i>RenderableHolder</i> . Increase its <i>deadTick</i> then draw dead enemy by calling <i>drawDeadEnemy</i> method.

### 3.2.2 Class MainScreen extends Canvas

#### 3.2.2.1 Field

- <u>AnimationTimer animation</u>	Animation for screen drawing and BGM playing.
-----------------------------------	---

#### 3.2.2.2 Constructor

+ MainScreen()	Initialize <i>Canvas</i> with width and height from <i>SceneManager</i> . Add event handler. (Use method below) Initialize animation with the following process. <ul style="list-style-type: none"> <li>- Play main menu BGM.</li> <li>- Draw main menu screen. (Use method below)</li> </ul> Start animation.
----------------	---

#### 3.2.2.3 Method

- void drawScreen()	Draw main menu background then draw text. (Use method below)
- void drawBackground(GraphicsContext gc)	Clear screen with black rectangle. Fill 10 5*5-pixels-ovals and 5 10*10-pixels-ovals. All with random position.
- void drawText(GraphicsContext gc)	Fill white text “Last Defense” with 80 font size and “Press enter to start” with 30 font size. Both are center-aligned at (400,120) and (400,350) respectively. Fill white text “Tutorial: T” and “Exit: ESC” with 20 font size at bottom right and bottom left of screen.
- void addEventHandler()	Add event handler for key event When ENTER is pressed, stop animation and BGM, then start a

	<p>new game. (Call GameMain's method)</p> <p>When ESCAPE is pressed, stop animation and BGM, then exit the program.</p> <p>When T is pressed, go to tutorial screen. (Use GameMain's method)</p> <p>When S is pressed, toggle silent mode. (Use RenderableHolder's method)</p>
+ void startAnimation()	Start animation.
+ void stopAnimation()	Stop animation.

### 3.2.3 Class ScoreScreen extends Canvas

#### 3.2.3.1 Field

- AnimationTimer animation	Animation for screen drawing and BGM playing.
----------------------------	---

#### 3.2.3.2 Constructor

+ ScoreScreen()	<p>Initialize <i>Canvas</i> with width and height from <i>SceneManager</i>.</p> <p>Add event handler. (Use method below)</p> <p>Initialize animation with the following process.</p> <ul style="list-style-type: none"> <li>- Play score BGM.</li> <li>- Draw score screen. (Use method below)</li> </ul> <p>Start animation.</p>
-----------------	---

#### 3.2.3.3 Method

+ drawScoreScreen(GraphicsContext gc)	<p>Fill background with black rectangle.</p> <p>Fill white text "GAME OVER" with 80 font size, "Score : " follow by score with 60 font size and "Press ENTER to retry" with 30 font size. All center-aligned at height 100, 230, 350 respectively. Fill white text "Exit: Esc" with 20 font size at bottom right of screen.</p>
+ addEventHandler()	<p>Add event handler for key event.</p> <p>When ENTER is pressed, stop animation and BGM, then start new game.</p> <p>When ESCAPE is pressed, stop animation and BGM, then exit the program.</p> <p>When S is pressed, toggle silent mode. (Use RenderableHolder's method)</p>

## 3.2.4 Class TutorialScreen extends Canvas

### 3.2.4.1 Constructor

+ TutorialScreen()	Initialize <i>Canvas</i> with width and height from <i>SceneManager</i> . Draw tutorial screen then add event handler. (Use method below)
--------------------	--

### 3.2.4.2 Method

- void drawScreen(GraphicsContext gc)	Fill screen with black rectangle. Fill white text “Z , C Change Bullet”, “X Shoot”, “UP/DOWN Move Character” and “ESC Pause” with 30 font size. All are center-aligned and set base line to top. Fill texts above at height 50, 150, 250 and 350 respectively. Fill white text “Back: B” with 20 font size at bottom right of screen.
- void addEvent()	Add event handler for key event. When B is pressed, start main screen animation and go to main menu. When S is pressed, toggle silent mode. (Use <i>RenderableHolder</i> ’s method)

## 3.4 Package game

### 3.4.1 Class GameMain

#### 3.4.1.1 Field

- <u>GameLogic gameLogic</u>	Logic of this game.
- <u>GameScreen gameScreen</u>	Main Game screen.
- <u>ScoreScreen scoreScreen</u>	Score Screen.
- <u>TutorialScreen tutorialScreen</u>	Tutorial Screen.
- <u>Thread drawThread</u>	Draw game screen thread.
- Thread logicThread	Game logic thread.
- boolean isGameRunning	Boolean show that is game running or not.
- <u>boolean isGamePaused</u>	Boolean show that is game pause or not.

#### 3.4.1.2 Constructor

+ GameMain()	Initialize logic, screen, threads, start all threads then go to gameScreen scene.
--------------	---

#### 3.4.1.3 Method

+ <u>void newGame()</u>	Initialize game logic and game screen then run this main game loop by thread.
+ <u>void goToMainMenu()</u>	Go to main menu page.
+ <u>void goToResult()</u>	Go to score page.
+ <u>void goToTutorial()</u>	Go to tutorial page.
+ <u>void pauseGame()</u>	Change <i>isGamePaused</i> value to true.
+ <u>void resumeGame()</u>	Change <i>isGamePaused</i> value to false.
+ <u>void stopGame()</u>	Change <i>isGameRunning</i> value to false.

### 3.4 Package logic

#### 3.4.1 Class Bullet extends Sprite

##### 3.4.1.1 Constructor

+ Bullet()	Initialize width and height by default value.
+ Bullet(double posX, double posY, double veloX)	Initialize width and height then set value <i>posX posY veloX</i> of bullet.

##### 3.4.1.2 Method

+ boolean isOutOfScreen()	Check that this bullet is out of screen or not.
---------------------------	---

#### 3.4.2 Class BulletBlue extends Bullet implements IRenderable

##### 3.4.2.1 Constructor

+ BulletBlue()	Initialize <i>BulletBlue</i> .
+ BulletBlue(double posX, double posY, double veloX)	Initialize <i>BulletBlue</i> and set value <i>posX posY veloX</i> of bullet.

##### 3.4.2.2 Method

+ void draw(GraphicsContext gc)	Draw bullet image.
---------------------------------	--------------------

### 3.4.3 Class BulletRed extends Bullet implements IRenderable

#### 3.4.3.1 Constructor

+ BulletRed()	Initialize <i>BulletRed</i> .
+ BulletRed(double posX, double posY, double veloX)	Initialize <i>BulletRed</i> and set value <i>posX posY veloX</i> of bullet.

#### 3.4.3.2 Method

+ void draw(GraphicsContext gc)	Draw bullet image.
---------------------------------	--------------------

### 3.4.4 Class BulletYellow extends Bullet implements IRenderable

#### 3.4.4.1 Constructor

+ BulletYellow()	Initialize <i>BulletYellow</i> .
+ BulletYellow(double posX, double posY, double veloX)	Initialize <i>BulletYellow</i> and set value <i>posX posY veloX</i> of bullet.

#### 3.4.4.2 Method

+ void draw(GraphicsContext gc)	Draw bullet image.
---------------------------------	--------------------

### 3.4.5 Class Enemy extends Sprite

#### 3.4.5.1 Field

# int deadTick	Tick count of dead image.
----------------	---------------------------

#### 3.4.5.2 Constructor

+ Enemy()	Initialize width height and <i>deadTick</i> of enemy by default value.
+ Enemy(double posX,double posY,double veloX)	Initialize width height and <i>deadTick</i> and set <i>posX posY veloX</i> of enemy.

#### 3.4.5.3 Method

+ boolean isOutOfScreen()	Check that enemy is out of screen or not.
+ void deadTickIncrease()	Increase <i>deadTick</i> by one.
+ void deadTickReset()	Set <i>deadTick</i> to zero.
+ int getDeadTick ()	Getter method for <i>deadTick</i> .

### 3.4.6 Class EnemyBlue extends Enemy implements IRenderable

#### 3.4.6.1 Constructor

+ EnemyBlue()	Initialize <i>EnemyBlue</i> .
+ EnemyBlue(double x, double y, double vX)	Initialize position and velocityX of <i>EnemyBlue</i> by x, y, vX value.

#### 3.4.6.2 Method

# int getState()	Get state of <i>EnemyBlue</i> images.
+ void draw(GraphicsContext gc)	Draw <i>EnemyBlue</i> image.
+ void drawDeadEnemy(GraphicsContext gc)	Draw dead enemy image when bullet collided with enemy.

### 3.4.7 Class EnemyRed extends Enemy implements IRenderable

#### 3.4.7.1 Constructor

+ EnemyRed ()	Initialize <i>EnemyRed</i> .
+ EnemyRed (double x, double y, double vX)	Initialize position and velocityX of <i>EnemyRed</i> by x, y, vX value.

#### 3.4.7.2 Method

# int getState()	Get state of <i>EnemyRed</i> images.
+ void draw(GraphicsContext gc)	Draw <i>EnemyRed</i> image.
+ void drawDeadEnemy(GraphicsContext gc)	Draw dead enemy image when bullet collided with enemy.

### 3.4.8 Class EnemyYellow extends Enemy implements IRenderable

#### 3.4.8.1 Constructor

+ EnemyYellow ()	Initialize <i>EnemyYellow</i> .
+ EnemyYellow (double x, double y, double vX)	Initialize position and velocityX of <i>EnemyYellow</i> by x, y, vX value.

#### 3.4.8.2 Method

# int getState()	Get state of <i>EnemyYellow</i> images.
+ void draw(GraphicsContext gc)	Draw <i>EnemyYellow</i> image.
+ void drawDeadEnemy(GraphicsContext gc)	Draw dead enemy image when bullet collided with enemy.

### 3.4.10 Class GameLogic

#### 3.4.10.1 Field

- int enemyTick	Bound of tick enemy to generate enemy to <i>enemyList</i> .
- int currentEnemyTick	Current enemy tick of enemy.
- double currentEnemySpeed	Current enemy speed.

#### 3.4.10.2 Constructor

+ GameLogic()	Initialize every field of this class.
---------------	---------------------------------------

#### 3.4.10.3 Method

+ void addEnemy(IRenderable enemy)	Add enemy to <i>enemyList</i> .
+ void addBullet(IRenderable bullet)	Add bullet to <i>bulletList</i> .
+ void updateLogic()	Update logic of game.
- void updatePosition()	Update position of enemy and bullet on screen.
- void checkBulletOutOfScreen()	Check that bullet out of screen or not. If bullet out of screen then remove bullet from <i>bulletList</i> .
- void checkPlayerOutOfScreen()	If player out of screen then move player to other side of screen.
- void generateEnemy()	Random generate enemy to <i>enemyList</i> .
- void checkIfEnemyDestroy()	Check that bullet is collide with enemy or not. If bullet collide with same color enemy then remove this enemy and this bullet out of list.
- boolean isSameColor(IRenderable bullet, IRenderable enemy)	Check that bullet is same color with enemy or not.
- void fireBullet()	Add bullet to <i>bulletList</i> . (position of bullet depend on position of player.)
- void endGame()	When end game go to result page.
- void readInput()	Read input from <i>GameScreen</i> input.
- void updateLevel()	Update velocity of enemy when score past on some point.
- void checkIfEndGame()	Check that game end or not. (Game end when enemy collide with player or enemy go to left side of screen.)
- void updateDeadEnemy()	Update tick dead of enemy. If tick past some point then remove form <i>deadEnemyList</i> .
- void secretJuiButton()	Secret button that will clear all enemies out of list !!! (Don't tell anyone about this secret. This combo button you have to find by yourself.)

- void secretMaxButton()	Secret button that will increase generate enemy speed of game !!! (Don't tell anyone of this secret. This combo button you have to find by yourself.)
- void ultimateLuckyBullet()	Super Ultra Hyper Secret button that will random generate bullet into screen. (You can test your luck with this button.)

### 3.4.10 Class Player extends Sprite implements IRenderable

#### 3.4.10.1 Field

# int shootTick	Tick that determine if player can shoot or not to prevent player from shooting too fast and make game a bit more challanging.
# int imageIndex	Current index of player image. (0 – Not move, 1 – Move up, 2 – Move down)

#### 3.4.10.2 Constructor

+ Player()	Initialize <i>Sprite</i> . Set <i>width</i> and <i>height</i> to 75. Set <i>shootTick</i> and <i>tick</i> to 15. Set <i>position</i> to (25, 165).
------------	---

#### 3.4.10.3 Method

+ boolean canShoot	This method check if player can shoot or not. If <i>tick</i> is more than or equal <i>shootTick</i> , then reset <i>tick</i> and return true. Else return false.
+ void setImageIndex(int imageIndex)	Set <i>imageIndex</i> to given <i>imageIndex</i> .
+ void draw(GraphicsContext gc)	Draw image from <i>playerImages</i> in <i>RenderableHolder</i> at <i>imageIndex</i> . The position is current Player position.

### 3.4.11 Abstract Class Sprite

#### 3.4.11.1 Field

# double positionX	Current position on X axis.
# double positionY	Current position on Y axis.
# double velocityX	Current velocity on X axis.
# double velocityY	Current velocity on Y axis.
# double width	Width of this sprite.
# double height	Height of this sprite.
# int tick	Current tick.



### 3.4.11.2 Constructor

+ Sprite()	Set <i>positionX</i> , <i>positionY</i> , <i>velocityX</i> and <i>velocityY</i> to 0.
------------	---

### 3.4.11.3 Method

+ void setPosition(double x, double y)	Set <i>positionX</i> and <i>positionY</i> to x and y respectively.
+ double getX()	Return <i>positionX</i> .
+ double getY()	Return <i>positionY</i> .
+ void setVelocity(double x, double y)	Set <i>velocityX</i> and <i>velocityY</i> to x and y respectively.
+ void update(double x, double y)	Add <i>positionX</i> and <i>positionY</i> by x and y respectively.
+ Rectangle2D getBoundary()	Return Rectangle2D with same width, height, <i>positionX</i> and <i>positionY</i> .
+ boolean intersects(Sprite s)	Determine if this sprite intersects with given sprite <i>s</i> . If intersects return true, else return false. (Use Rectangle2D's method)
+ void tickIncrease()	Increase tick by 1.
+ void tickReset()	Set tick to 0.

## 3.5 Package share

### 3.5.1 Interface IRenderable

#### 3.5.1.1 Method

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

### 3.5.2 Class RenderableHolder

#### 3.5.2.1 Field

- List<IRenderable> enemies	List of all enemies on screen.
- List<IRenderable> deadEnemies	List of all dead enemies that animation still running.
- List<IRenderable> bullets	List of all bullets on screen.
+ <u>List&lt;Image&gt; playerImages</u>	List of player images.
+ <u>List&lt;Image&gt; enemyYellowImage</u>	List of yellow enemy images.
+ <u>List&lt;Image&gt; enemyRedImage</u>	List of red enemy images.
+ <u>List&lt;Image&gt; enemyBlueImage</u>	List of blue enemy images.
+ <u>Image bulletYellowImage</u>	Image of yellow bullet.

+ Image <u>bulletRedImage</u>	Image of red bullet.
+ Image <u>bulletBlueImage</u>	Image of blue bullet.
- AudioClip <u>shootSound</u>	Sound when shoot bullet.
- AudioClip <u>enemyDeadSound</u>	Sound when enemy dead.
- AudioClip <u>enemyNotDeadSound</u>	Sound when shoot enemy with wrong color bullet.
- AudioClip <u>changeBulletSound</u>	Sound when change bullet color.
- AudioClip <u>endGameSound</u>	Sound when game end before go to score screen.
- AudioClip <u>mainMenuBgm</u>	Main screen background music.
- AudioClip <u>gameBgm</u>	Game screen background music.
- AudioClip <u>scoreBgm</u>	Score screen background music.
- int score	Current score.
- Player player	Current main player character.
- int bulletState	Current bullet color (1 – Red, 2 – Blue, 3 – Yellow).
- boolean silentModelsOn	Current silent mode status.

### 3.5.2.2 Constructor

+ RenderableHolder()	Initialize <i>score</i> , <i>player</i> , <i>bullet</i> and all lists.
----------------------	--

### 3.5.2.3 Method

+ void newGame()	Clear <i>enemies</i> , <i>deadEnemies</i> , <i>bullets</i> . Set <i>score</i> to 0, Initialize <i>player</i> , set <i>bulletState</i> to 0. Clear <i>GameScreen</i> 's inputs.
+ void addEnemy(IRenderable enemy)	Add <i>enemy</i> to <i>enemies</i> .
+ void addDeadEnemy(IRenderable deadEnemy)	Add <i>deadEnemy</i> to <i>deadEnemies</i> .
+ void addBullet(IRenderable bullet)	Add <i>bullet</i> to <i>bullets</i> .
+ void removeEnemy(List<IRenderable> removeEnemyList)	Remove all elements in <i>removeEnemyList</i> from <i>enemies</i> .
+void removeBullet(List<IRenderable> removeBulletList)	Remove all elements in <i>removeBulletList</i> from <i>bullets</i> .
+ void removeDeadEnemy(List<IRenderable> removeDeadEnemyList)	Remove all elements in <i>removeDeadEnemyList</i> from <i>deadEnemies</i> .
+ void addScore(int score)	Add current score with given score.
+ void subtractScore(int score)	Subtract current score with given score.
+ void updatePlayer(double x, double y)	Update player position with distance x and y.

	(Call Sprite's <i>update</i> method)
+ void updatePlayerImage(int index)	Set player's image to given image index. (Call Player's <i>setImageIndex</i> method)
+ void playShootSound()	If Silent mode is off, play <i>shootSound</i> with 0.5 volume level.
+ void playEnemyDeadSound()	If Silent mode is off, play <i>enemyDeadSound</i> with 0.9 volume level.
+ void playEnemyNotDeadSound()	If Silent mode is off, play <i>enemyNotDeadSound</i> .
+ void playChangeBulletSound()	If Silent mode is off, play <i>changeBulletSound</i> with 0.5 volume level.
+ void playEndGameSound()	If Silent mode is off, play <i>endGameSound</i> .
+ void playMainMenuBgm()	If <i>mainMenuBgm</i> is not playing and silent mode is off, then play <i>mainMenuBgm</i> with 1.5 volume level.
+ void stopMainMenuBgm	Stop playing <i>mainMenuBgm</i> .
+ void playGameBgm()	If <i>gameBgm</i> is not playing and silent mode is off, then play <i>gameBgm</i> with 1.2 volume level.
+ void stopGameBgm()	Stop playing <i>gameBgm</i> .
+ void playScoreBgm()	If <i>scoreBgm</i> is not playing and silent mode is off, then play <i>scoreBgm</i> with 0.8 volume level.
+ void stopScoreBgm()	Stop playing <i>scoreBgm</i> .
+ void stopAllSound()	Stop playing <i>mainMenuBgm</i> , <i>gameBgm</i> and <i>scoreBgm</i> .
+ void setPlayerPosition(double x, double y)	Set player position to x and y. (Call Sprite's <i>setPosition</i> method)
+ void toggleSilentMode()	If silent mode is on, turn it off and stop playing all BGM. Else, turn silent mode on by set <i>silentModelsOn</i> to true.
+ Getter for score, player, enemies, deadEnemies, bullets, bulletState	
+ Setter for bulletState	

## 3.6 Package window

### 3.6.1 Final Class SceneManager

#### 3.6.1.1 Field

- <u>Stage primaryStage</u>	This program's primary stage.
- <u>Canvas mainScreen</u>	Initialize <i>MainScreen</i> for this variable.
- <u>Scene mainMenuScene</u>	This program's main menu scene. Initialize this Scene with a Pane of <i>mainScreen</i> .
+ <u>SCENE_WIDTH</u>	This program's scene width. Set to 800.
+ <u>SCENE_HEIGHT</u>	This program's scene height. Set to 450.

#### 3.6.1.2 Method

+ initialize(Stage stage)	Set <i>primaryStage</i> to stage and show <i>primaryStage</i> .
+ goToMainMenu()	Request focus for main screen and set scene of <i>primaryStage</i> to <i>mainMenuScene</i> .
+ goToSceneOf(Canvas canvas)	Create new scene with given <i>canvas</i> then request focus for that <i>canvas</i> . Set scene of <i>primaryStage</i> to that scene.

## Credit

### **Producer**

Siwa B. Hojui

Natchanon A.

### **Music & Sound Effect**

Natchanon A.

Tazy / Quonux / zagi2 / rhodesmas / n\_audioman – freesound.com

### **Game Programmer**

Siwa B. Hojui

Natchanon A.

### **Character Design**

Natchanon A.