

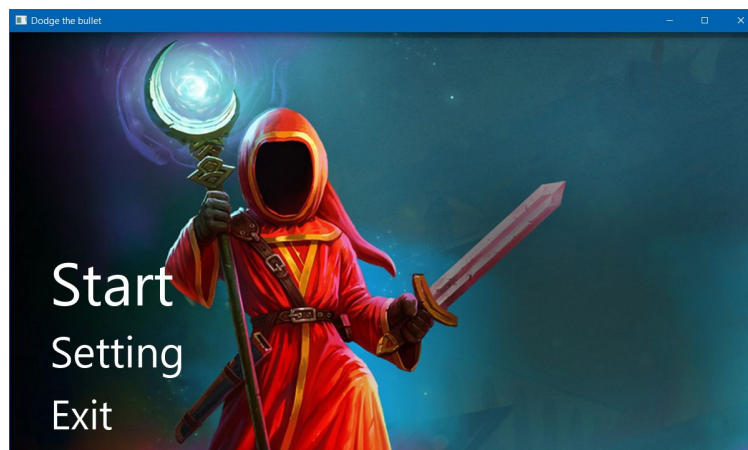
DODGE THE BULLET

by Supavich Asavakanoksilp 5831072421
and Parin Meparinya 5831037521
students of CPCU

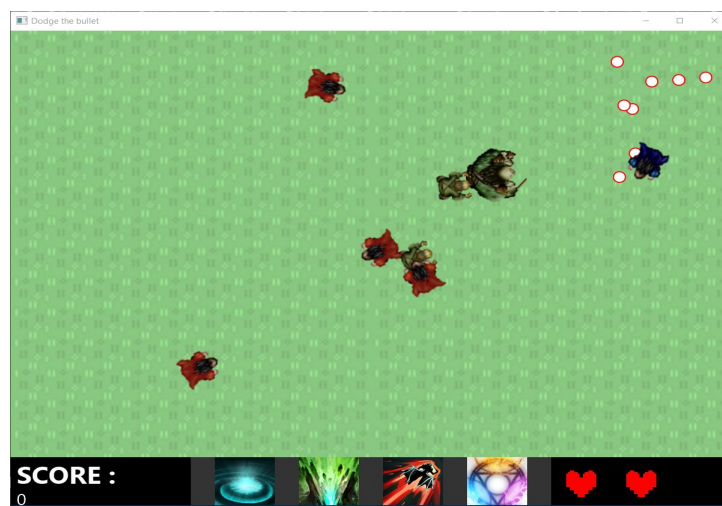
Overview

dodge the bullet เป็นเกมแนว bullet hell,twin-sticks(keyboard) shooter ที่เราควบคุมตัวละครเพื่อสู้กับ troll, witch,goblin,และ wisp ใน arena (เนื่องจากไม่สามารถหารูปที่สื่อได้ถึงตัวตน จึงมีการเปลี่ยนแปลงรูปตัวละครเล็กน้อย) โดยที่ตัวละครที่เราเล่นจะมี 4 สกิลให้ใช้เพื่อช่วยต่อสู้กับศัตรูหลากหลายรูปแบบ เกมจะจบเมื่อผู้เล่นเสียพลังชีวิตทั้งหมด

ตัวอย่าง Menu

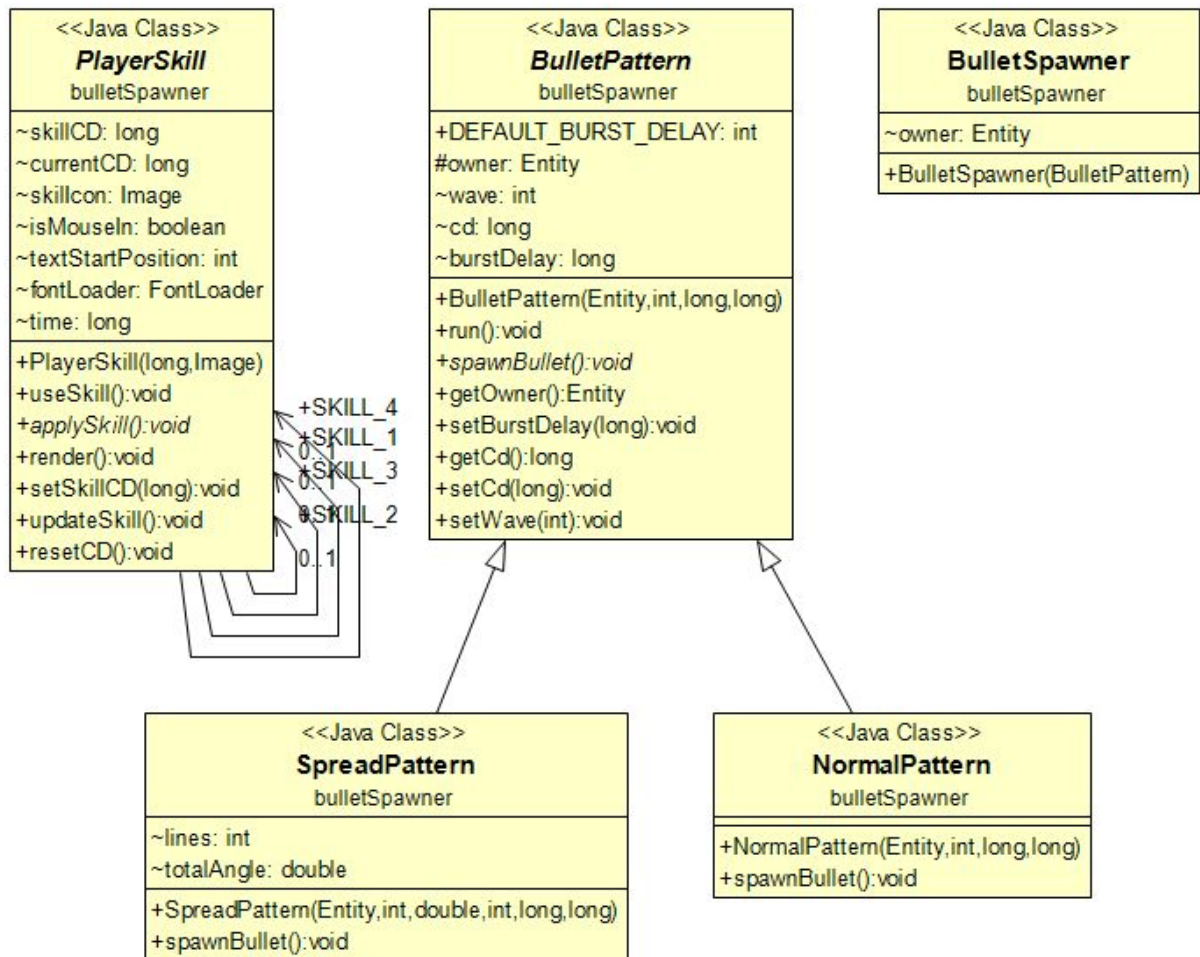


ตัวอย่าง Game Scene



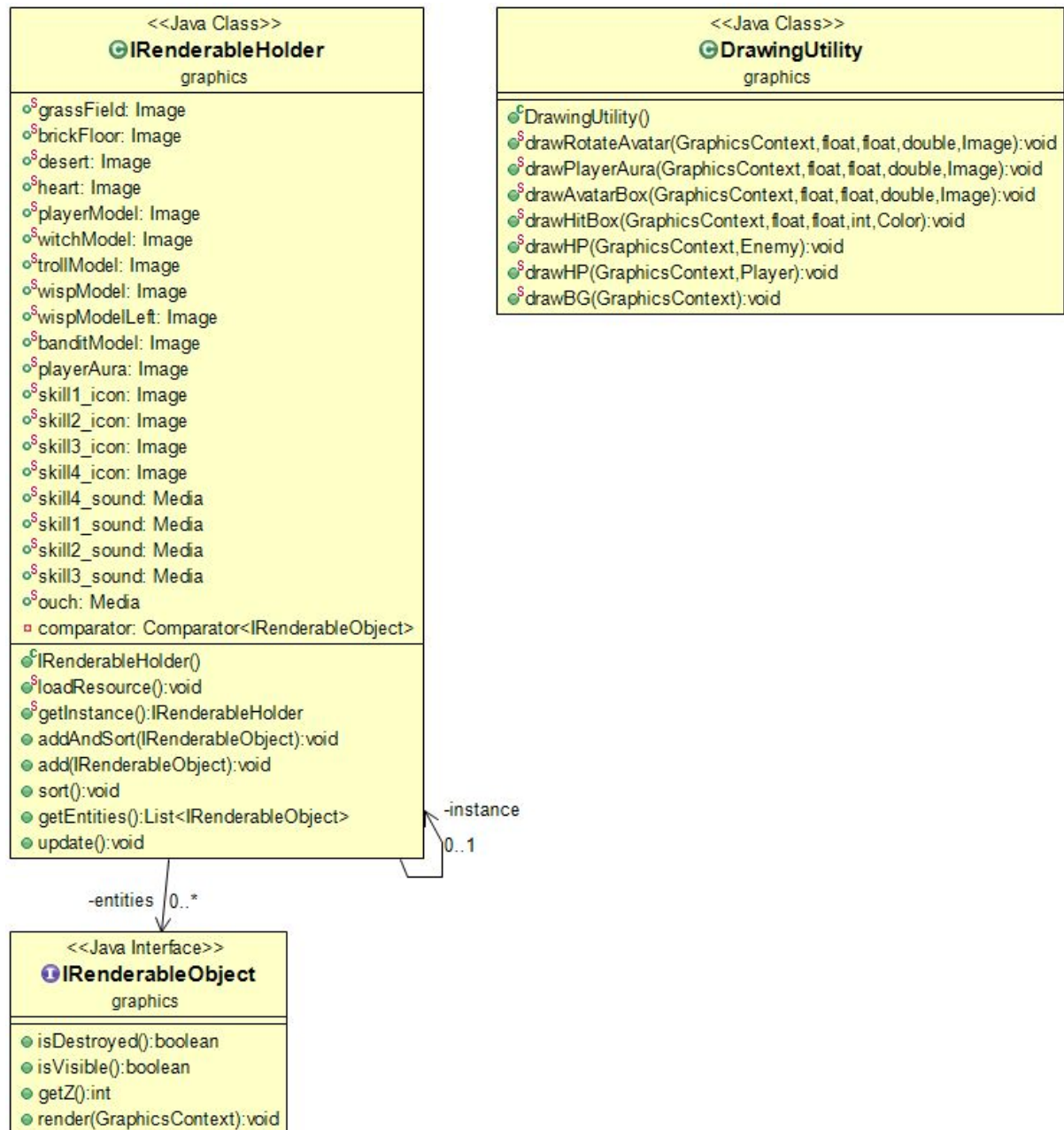
uml

package bulletSpawner



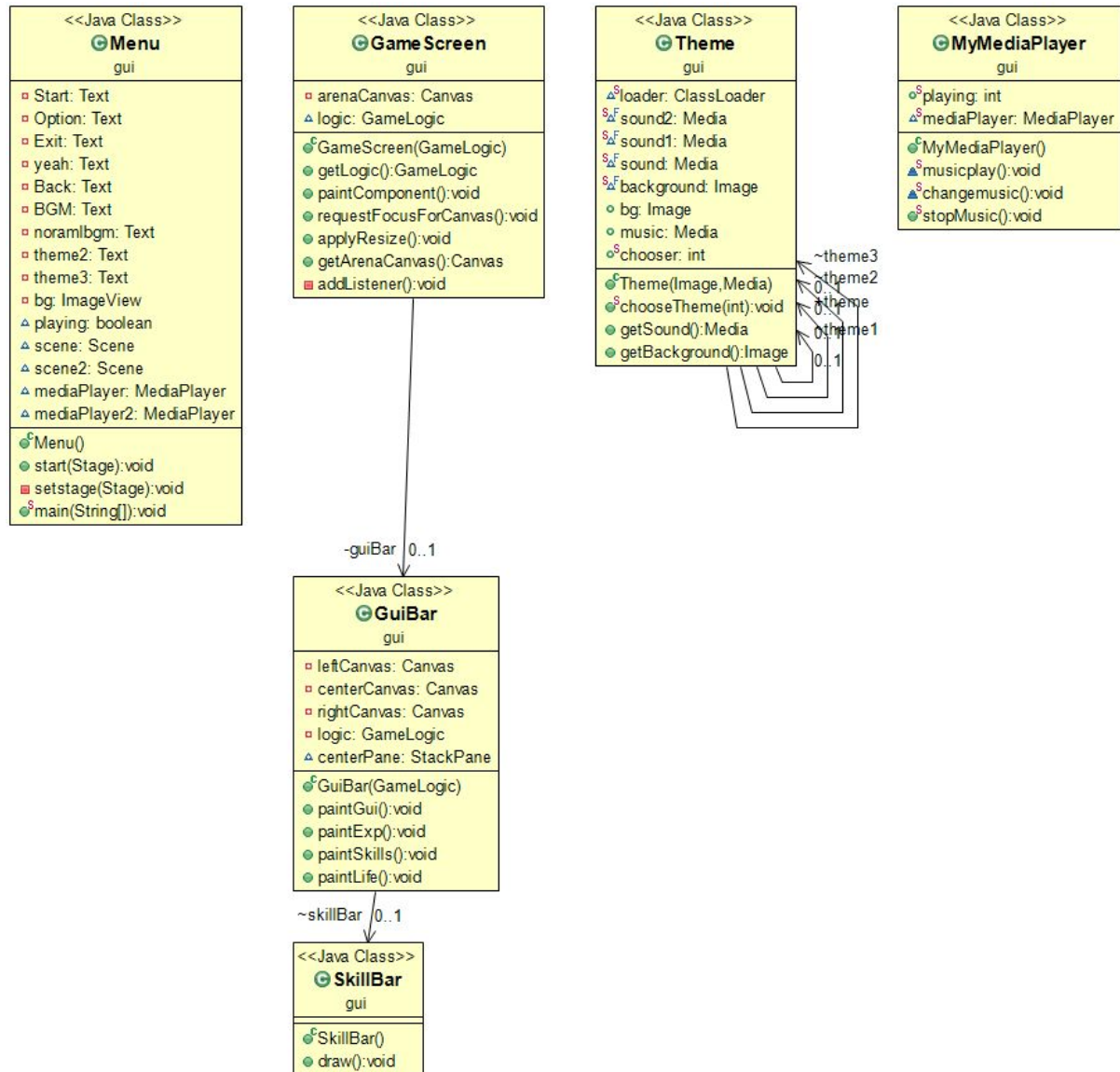
uml

package graphics



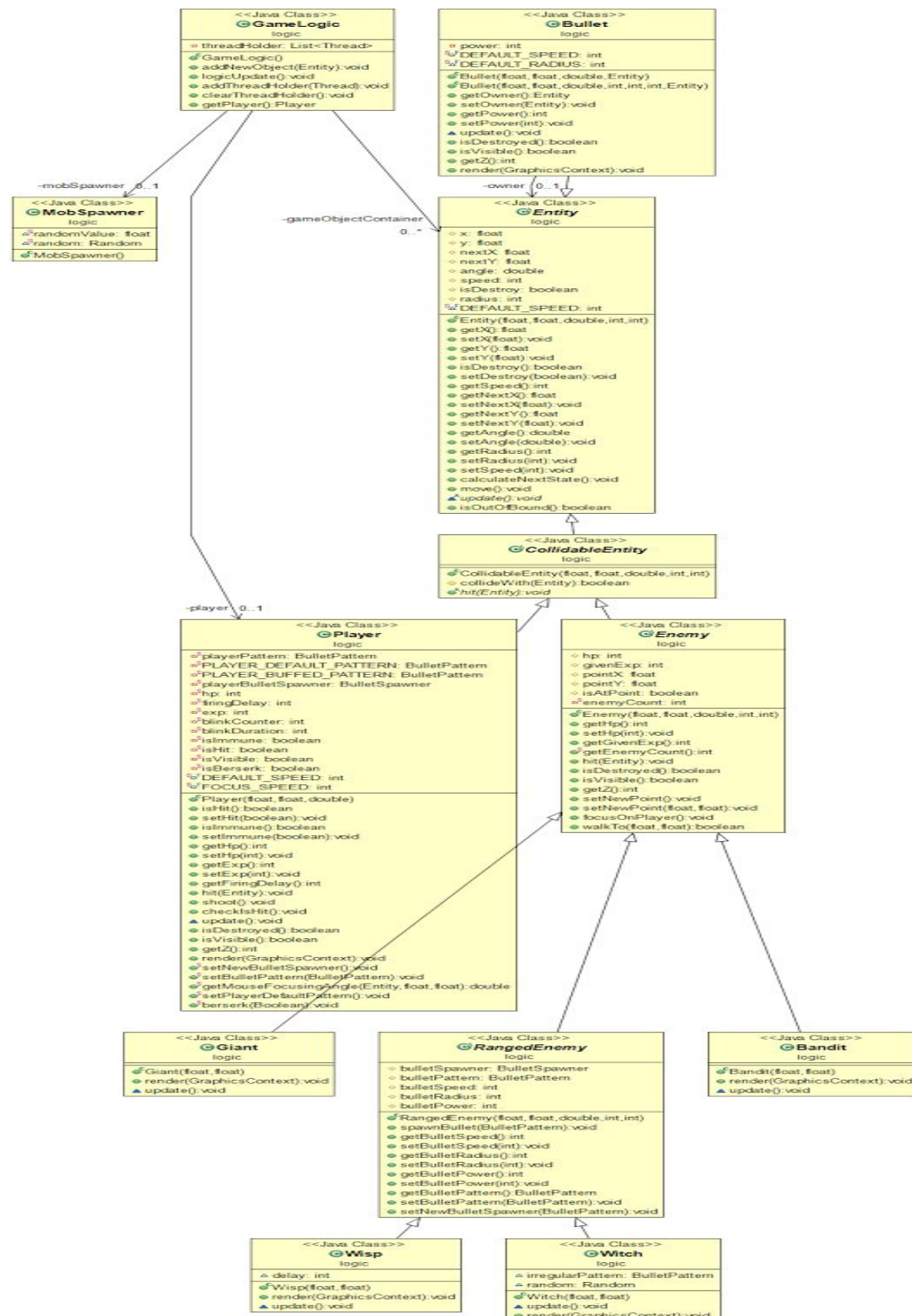
uml

package gui



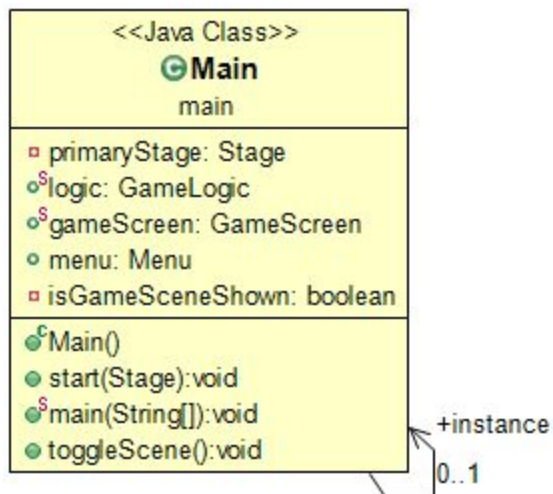
uml

package logic *please extends image before see the component*

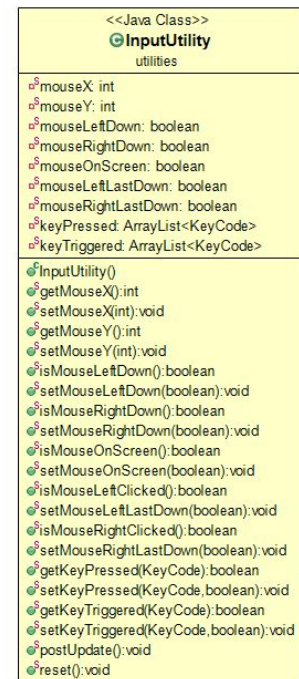
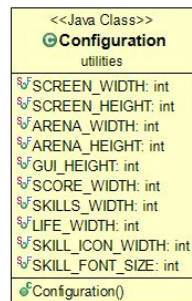


uml

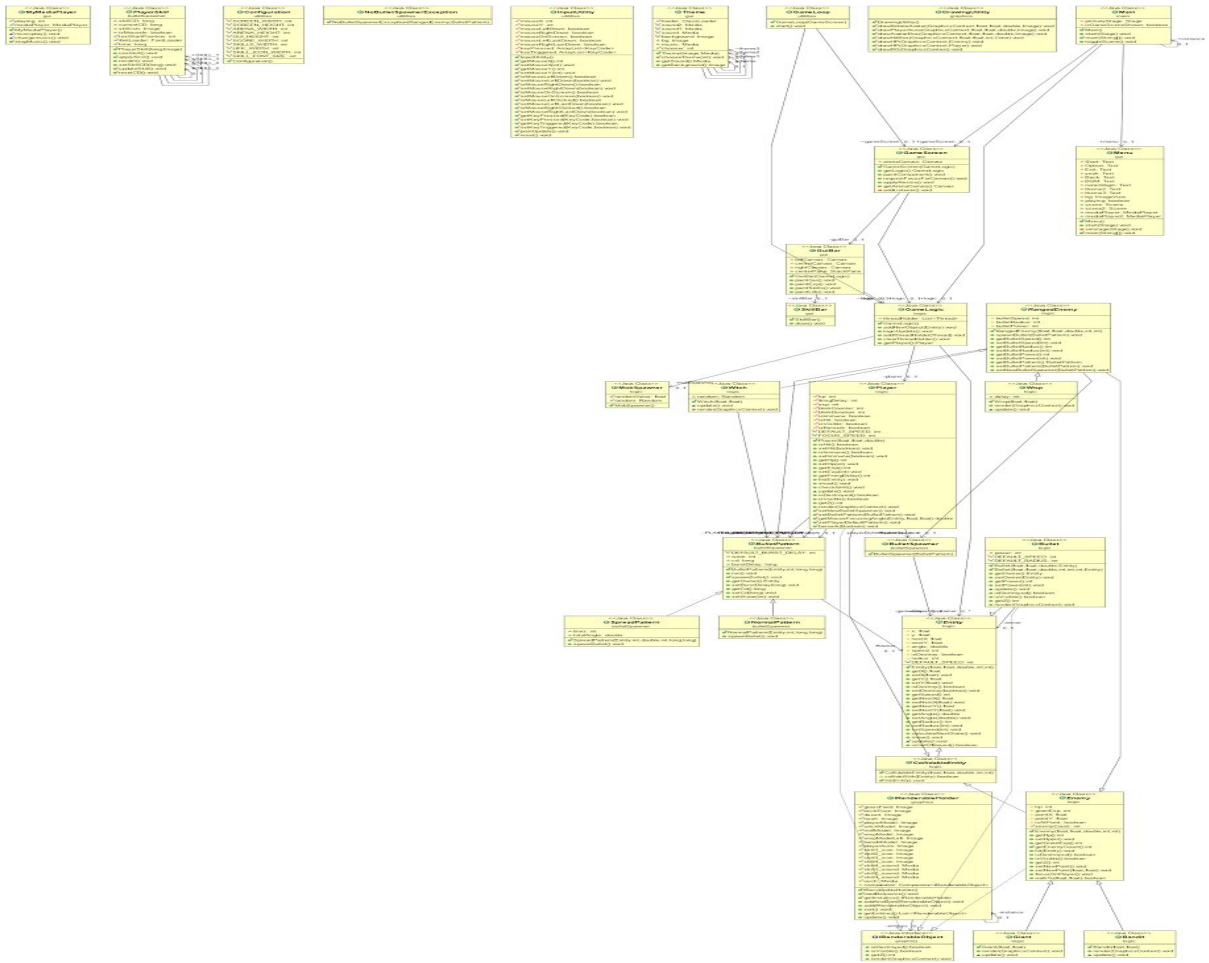
package main



package utilities



summary uml *please extends image before see the component*



Class detail

1. package bulletSpawner :

จะคอยควบคุมเกี่ยวกับการสร้าง Bullet

1.1 class BullePattern

เป็นตัวกำหนดรูปแบบ Bullet ที่จะสร้างขึ้น

Field

int Default_Burst_Dalay : ตั้งค่า burst delay เริ่มต้น ซึ่งมีค่า 150

Entity owner : สร้าง entity

int wave : ตั้งค่า wave

long cd : Zeit cooldown

long burstDelay: เก็บค่า burst delay

Constructor

BullePattern(Entity owner, int wave, long cd, long burstDelay) :

ประกาศค่าให้ field

Method

public void run() : จะเริ่ม spawnBullet() แล้วจะ sleep เป็นเวลา cd;

public abstract void spawnBullet() : เป็น abstract method ทำหน้าที่สร้าง Bullet

public Entity getOwner() : return owner

public void setBurstDelay(long burstDelay) : set burst delay

public long getCd() : return cd

public void setCd(long cd) : set cd

public void setWave(int wave) : set wave

1.2 class BulletSpawner extends Thread

เป็น Thread ที่จะคอยสร้าง Bullet ตาม BullePattern

Field

Entity owner

Constructor

public BulletSpawner(BullePattern pattern) :

initializes field by set pattern as a Runnable and set this owner to be the same with the pattern's.

1.3 class NormalPattern extends BullePattern

เป็นรูปแบบการสร้าง Bullet รูปแบบปกติโดยจะปล่อยมาทีละ wave เท่ากับ wave แต่ละ wave จะเว้นระยะเวลา burstDelay

Constructor

public NormalPattern(Entity owner, int wave, long cd, long burstDelay)

Method

public void spawnBullet() : สร้าง Bullet ตามรูปแบบที่กล่าวไว้ก่อนหน้านี้ที่มี owner เป็นเจ้าของ

1.4 class PlayerSkill

เป็น Canvas ที่เก็บค่าเกี่ยวกับ skill ของผู้เล่นและจะเป็นตัววาดสถานะของ skill ใน GameScreen ด้วย

Field

long skillCD : ตั้งค่า cool down skill
long currentCD : cool down ปัจจุบัน
Image skillIcon : skill icon
boolean isMouseIn : เช็คค่า mouse ที่อยู่มั้ย
static long time : เก็บค่าเวลาที่เช็คล่าสุด

Constructor

public PlayerSkill(long cd, Image icon) : initializes cd and icon Method

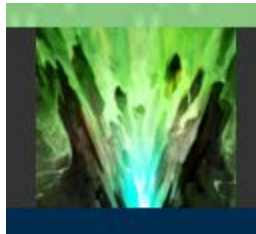
Method

public void useSkill() : เป็น method ที่ถูกเรียกตอนที่ผู้ใช้ หาก currentCD ของ skill นี้น้อยกว่าหรือเท่ากับ 0 ให้เรียก applySkill() เพื่อให้เกิดผลลัพธ์

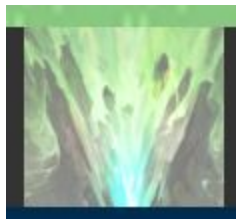
public void applySkill() : แสดงผลลัพธ์ของการใช้ skill

public void render() : วาดภาพ skill icon และสถานะของ skill นั้นๆ

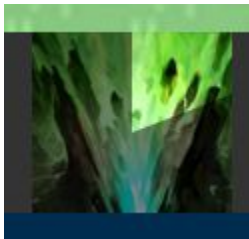
ทำการวาดภาพ skill ของตน ดังรูป



หากมี mouse อยู่ด้านในให้ทำการ Highlight ดังรูป



หาก skill กำลัง cooldown ให้แสดงระยะเวลาที่เหลือก่อนกลับมาใช้ได้อีกครั้งในรูปแบบ ดังรูป



public static void updateSkill() : อัปเดต currentCD ทุก skill โดยอาศัย time เปรียบเทียบกับเวลาปัจจุบันในการอัปเดต ถูกเรียกทุก loop game

public static void resetCD() :

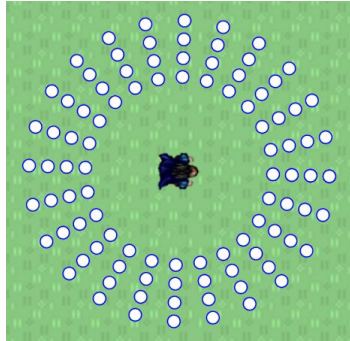
ทำการรีเซ็ต currentCD ทุกสกิลให้พร้อมใช้งาน

Inner object

ข้างล่างนี้เป็น skill ที่ผู้เล่นมีจำนวน 4 skills สร้างขึ้นโดยใช้ class **PlayerSkill**

`public final static PlayerSkill SKILL_1 :`

เป็น **PlayerSkill** ที่พอใช้จะสร้าง Bullet รอบ ตัว จำนวน 4 ชุด ชุดละ 24 ลูก กระจายทุกทิศทาง แต่ละชุดจะห่างกัน 100 millis

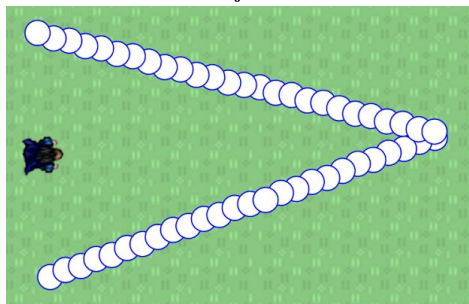


`public final static PlayerSkill SKILL_2 :`

เป็น **PlayerSkill** ที่พอใช้จะปล่อย Bullet

ที่มีขนาดใหญ่พิเศษและทำความเสียหายมากกว่าปกติตรงหน้าผู้เล่นหลายๆ คู่

แต่ละคู่จะอยู่ห่างจากผู้เล่นในระยะที่ต่างกัน Bullet แต่ละคู่จะวิ่งไปทางซ้ายและทางขวาของผู้เล่นฝั่งละลูก



`public final static PlayerSkill SKILL_3 :`

เป็น **PlayerSkill** ที่พอใช้แล้วจะทำการเคลื่อนย้ายผู้เล่นไปที่ตำแหน่ง mouse ของผู้เล่นและทำการสร้าง Bullet รอบตัวจำนวนหนึ่งที่มีขนาดเล็กกว่าปกติ

`public final static PlayerSkill SKILL_4 :`

เป็น **PlayerSkill** ที่พอใช้แล้วจะทำให้ผู้เล่น berserk เป็นระยะเวลา 5000 millis แล้วกลับมาเป็นปกติ

1.5 class SpreadPattern extends BulletPattern

รูปแบบการปล่อย Bullet จำนวน wave ชุด แต่ละชุดจะปล่อยเท่ากับ lines โดยจะกระจายให้ทั่วภายในมุมที่กำหนดไว้ ดังรูป



Field

`int lines :` จำนวน Bullet ที่ปล่อยในแต่ละชุด

`double totalAngle :` ความกว้างในการปล่อย Bullet ในหน่วยองศา

Constructor

public SpreadPattern(Entity owner, int lines, double totalAngle, int wave, long cd, long burstDelay)

Method

public void spawnBullet() : ทำการสร้าง Bullet ตามรูปแบบที่กล่าวไปก่อนหน้านี้

2. package graphics :

เก็บ class เกี่ยวกับการวาดรูป

2.1 class DrawingUtility

เก็บ method ที่จะคอยวาดรูปในรูปแบบต่างๆ

Method

public static void drawRotateAvatar(GraphicsContext gc, float x, float y, double angle, Image image) : วาดภาพ image ที่มีตรงกลางตรงจุด x, y ที่หันหน้าไปทิศเดียวกับ angle
public static void drawPlayerAura(GraphicsContext gc, float x, float y, double angle, Image image) : คล้ายๆ กับ method drawRotateAvatar แต่จะวาดรูปโปร่งใสแทน
public static void drawBG(GraphicsContext gc) : วาดรูป background ของ GameScreen

2.2 class IRenderableHolder

เป็น class ที่คอยเก็บ Entity ที่ต้องการ render ในแต่ละ frame และยังทำการ load ภาพและเสียงมาเก็บไว้ก่อนแล้วให้ class อื่นเรียกใช้จากในนี้

Field

private static final IRenderableHolder instance : Singleton
private List<IRenderableObject> entities : เก็บ entity ทั้งหมด
private Comparator<IRenderableObject> comparator : เอาไว้เรียงและ public
public static Image / Media จำนวนมาก : เป็นตัวเก็บภาพและเสียงต่างๆ

Constructor

สร้าง entities
สร้าง lamda comparator

Method

public static void loadResource() : load resource ทั้งหมด
public static IRenderableHolder getInstance() : return instance
public void addAndSort(IRenderableObject entity) : call add entity + call sort
public synchronized void add(IRenderableObject entity) :
add entity to entites+ call sort
public synchronized void sort() : sort enetites
public List<IRenderableObject> getEntities() : return entites
public void update() : เอาอันที่ isDestroyed == true ออก

2.3 class IRenderableObject.java

Field

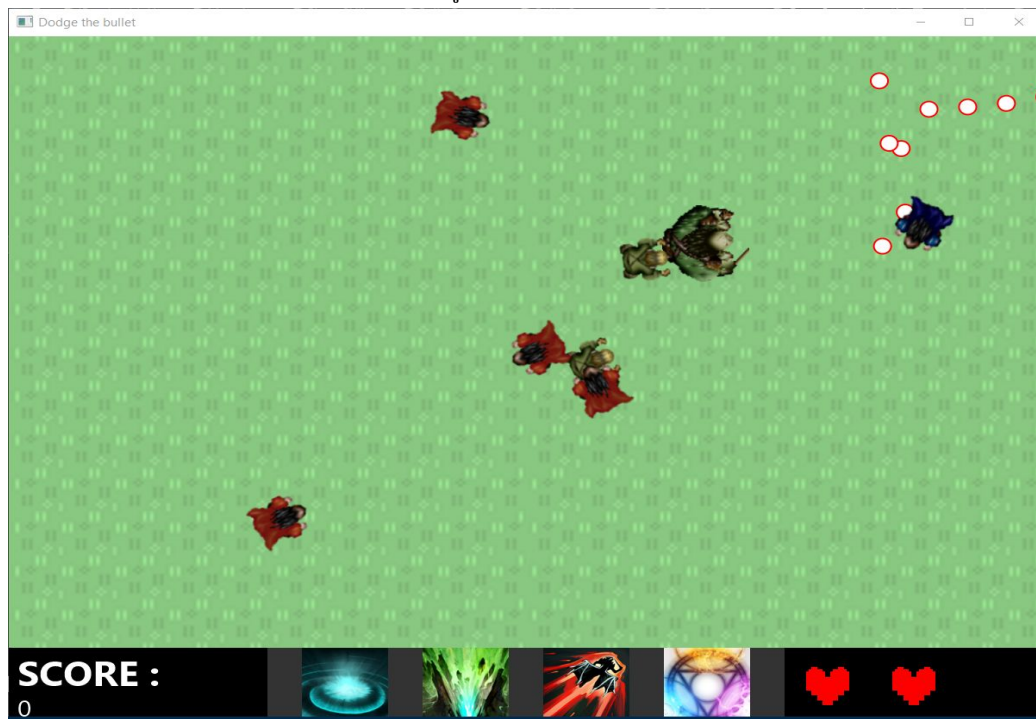
```
public abstract boolean isDestroyed(); //เช็คว่ายัง  
public abstract boolean isVisible(); //เช็คว่ายืน  
public abstract int getZ(); //เช็ค priority  
public abstract void render(GraphicsContext gc); //render ภาพ
```

3. package gui

เกี่ยวกับภาพหน้าต่างในเกม

3.1 class GameScreen extends VBox

หน้าจอขณะเล่นเกมมีลักษณะดังรูป



Field

```
private GuiBar guiBar : สร้าง guibar  
private Canvas arenaCanvas : สร้าง canvas arena หรือส่วนบนของ GameScene  
GameLogic logic : สร้าง gamelogic
```

Constructor

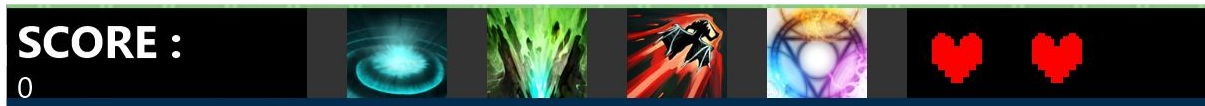
```
public GameScreen(GameLogic logic) : ประกาศค่าให้ field
```

Method

```
public GameLogic getLogic() : return this.logic  
public void paintComponent() : วาดหน้าจอ GameScreen  
public void requestFocusForCanvas() : this.requestFocus  
public void applyResize() : เปลี่ยนขนาดหน้าจอ  
public Canvas getArenaCanvas() : return arenaCanvas;  
private void addListener() : ทำการสร้าง handler คอยตรวจจับ Input
```

3.2 class GuiBar extends BorderPane

ทำการสร้าง GuiBar ที่จะอยู่ด้านล่างของ GameScreen มีลักษณะดังรูป



Field :

private Canvas leftCanvas, centerCanvas, rightCanvas : สร้าง canvas
private GameLogic logic : เก็บค่า gamelogic
StackPane centerPane : สร้าง StackPane สำหรับตรงกลาง
SkillBar skillBar : สร้าง skillbar

Constructor

public GuiBar(GameLogic logic) : เช็หค่า field ทั้งหมด

Method

public void paintGui() : call paintexp, paintskill, paintlife
public void paintExp() : วาด score (อยู่ตรงด้านซ้าย)
public void paintSkills() : วาด skillbar (อยู่ตรงกลาง)
public void paintLife() : วาด life (อยู่ตรงด้านขวา)

3.3 class Menu

สร้างหน้า Menu

Field

private Text Start; สร้าง text
private Text Exit; สร้าง text
private Text yeah; สร้าง text
private Text Back; สร้าง text
private Text BGM; สร้าง text
private Text noramlbgm; สร้าง text
private Text theme2; สร้าง text
private Text theme3; สร้าง text
private ImageView bg; สร้าง image

Method

public void start(Stage stage) throws Exception :
โหลดจาก FXML file มาวาดบน canvas
private void setstage(Stage stage) :
ตั้งให้กดที่ตัว text แล้วเปลี่ยนหน้าจอไปได้(กด Option ให้ขึ้น setting scene
,กดเลือก theme ได้ กดback กลับมาหน้า menu)
public static void main(String[] args) : -> launch(args);

3.4 class MyMediaPlayer

Field

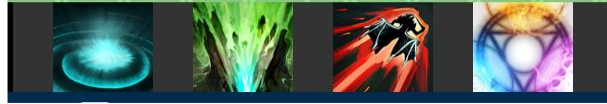
public static int playing; ตัวเลือกว่าจะเล่นเพลงไหน
static MediaPlayer mediaPlayer; สร้างตัวเล่น

Method

static void musicplay() : เลือกเพลงมาเล่นตาม theme
static void changemusic() : เลือกเพลง
public static void stopMusic() หยุดเพลง

3.5 class SkillBar extends HBox

สร้าง SkillBar ที่ใช้ใน GuiBar ด้านในมี icon skill ของผู้เล่น ดังรูป



Constructor

public SkillBar() : วาดข้อมูล skillbar

Method

public void draw() : render each player skill

3.6 class Theme

เก็บค่า BGM และ background ที่จะใช้ใน gameScreen

Field

static ClassLoader loader : class loader

static final Image background : โหลดรูปมาเก็บไว้

public Image bg : เลือก background

public Media music : เลือก เพลง

public static Theme theme : เลือก theme

public static int chooser : ตัวเลือก theme

Constructor

public Theme(Image background, Media sound)

Method

public static void chooseTheme(int i) : เลือก theme

public Media getSound() : return music

public Image getBackground() : return bg

4. package logic

เก็บค่าและดำเนิน logic game

4.1 class Bandit extends Enemy

สร้าง Bandit ที่คอยไล่ตาม Player

Constructor

public Bandit(float x, float y) : กำหนดค่าต่างๆให้ bandit

Method

public void render(GraphicsContext gc) : render Bandit

void update() : call focusonplayer + move

4.2 class Bullet extends Entity implements IRenderableObject

ตัว Bullet ที่สร้างความเสียหายให้กับผู้เล่นหรือศัตรู

Field

private Entity owner : ใช้แสดงความเป็นเจ้าของ

private int power : กำหนด power

public static final int DEFAULT_SPEED = 5 : ตั้งค่า speed ตั้งต้น

static final int DEFAULT_RADIUS = 10 : ตั้งขนาดตั้งต้น

Constructor

public Bullet(float x, float y, double angle, Entity owner) :

ตั้งค่าให้ field แบบใช้ตั้งต้น speed และขนาดตั้งต้น และ power = 1

public Bullet(float x, float y, double angle, int speed, int radius, int power, Entity owner) :

ตั้งค่าให้ field เองทั้งหมด

Method

getter/setter ของ field

void update() : ถ้ายังไม่ถูกทำลายให้เรียก move

public boolean isDestroyed() : return isDestroy

public boolean isVisible() : return true

public synchronized int getZ() : return Integer.MIN_VALUE;

public synchronized void render(GraphicsContext gc) : วาด Bullet

4.3 abstract class CollidableEntity extends Entity

เป็น abstract class ของ Entity ที่สามารถชน Entity อื่นได้

Constructor

public CollidableEntity(float x, float y, double angle, int speed, int radius)

Method

protected boolean collideWith(Entity e) : เช็คว่าจะชนกับ entity อื่นหรือไม่

4.4 abstract class Enemy extends CollidableEntity implements

IRenderableObject

abstract class ที่เป็นตัวแทนของศัตรูในเกม

Field

protected int hp : เก็บค่า hp

protected int givenExp : ตั้ง exp ที่ให้

protected float pointX, pointY : กำหนดจุด x, y สำหรับ walkTo()

protected boolean isAtPoint : เช็ค

private static int enemyCount = 0 : เช็คจำนวน

Constructor

public Enemy(float x, float y, double angle, int speed, int radius) :

กำหนดค่าให้ field

Method

getter/setter field

public void hit(Entity e) :

ถ้าโดน Bullet ของ player ให้ลดเลือด เลือดหมดให้ isDestroy = true ถ้าชน player ให้ลดเลือด player 1 หน่วยและ set isDestroy = true

public void setNewPoint() : random pointX, pointY ภายใน Arena

public void setNewPoint(float x, float y) : set pointX = x, pointY = y

public void focusOnPlayer() เมื่อถูกเรียกจะทำให้ตัวที่ถูกเรียกหันมาหา Player

public boolean walkTo(float x, float y) :

เมื่อถูกเรียกจะทำการเดินไปจุด pointX, pointY ด้วย speed หากถึงจุดแล้วจะให้ค่า true ถ้ายังไม่ถึงให้ค่า false

4.5 abstract class Entity

abstract class เป็นตัวแทนของวัตถุในเกม

Field

protected float x ,y : ตำแหน่ง

protected float nextX : next position x

protected float nextY : next position y

protected double angle : มุมที่หันอยู่

protected int speed : ความเร็ว

protected boolean isDestroy : เช็คว่ายตายรึยัง

protected int radius : ขนาด

static final int DEFAULT_SPEED = 3 : ตั้ง default speed = 3

Constructor

public Entity(float x, float y, double angle, int speed, int radius) : ตั้งค่าให้ field

Method

getter/setter ของ field

public void move() : call calculatenextstate

abstract void update()

public boolean isOutOfBound() : เช็คว่ายเกินขอบมัย

4.6 class GameLogic

Logic หลักของเกม

Field

private Player player : ตัวแทนผู้เล่นหรือ player

private List<Entity> gameObjectContainer : สร้าง entity list

private List<Thread> threadHolder : สร้าง list thread

private MobSpawner mobSpawner : สร้าง mobspawner

Constructor

public GameLogic() ประกาศค่าให้ field เริ่ม mobspawner ให้ทำงาน

Method

public synchronized void addNewObject(Entity entity) : เพิ่ม entity ให้ gameObjectContainer

public synchronized void logicUpdate() : update skill+update all thing in gameObjectContainer + remove all entity that isDestroy = true + clear dead thread

public Player getPlayer() : return player

4.7 class Giant extends Enemy :

สร้าง Enemy ที่มีขนาดใหญ่และมี HP มากกว่าปกติ คอยไล่ตามผู้เล่น

Constructor

public Giant(float x, float y) : เซ็ตค่าให้ field

Method

public void render(GraphicsContext gc) : draw Giant

void update() : call move()+this.focusOnPlayer()

4.8 class MobSpawner extends Thread

Thread ที่คอยสร้าง Enemy เข้ามาใน arena

Field

static float randomValue : สร้างค่ามาเก็บค่า random

static Random random = new Random() : สร้างตัว random

Constructor

public MobSpawner() :

สร้าง runnable มาที่จะคอยสร้าง Enemy ที่จะเกิดตรงขอบด้านใดด้านหนึ่ง มีทั้งหมด 4 ประเภท ได้แก่ Witch, Wisp, Giant และ Bandit แต่ละประเภทมีโอกาสเกิดเท่ากัน มีการจำกัดจำนวนศัตรู และจะสร้างมาเป็นชุด ชุดละจำกัดจำนวน

4.9 class Player extends CollidableEntity implements IRenderableObject

ตัวแทนผู้เล่น

Field

private static BulletPattern playerPattern : สร้างตัวเก็บ bullet pattern

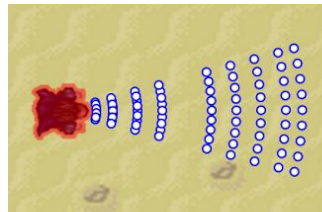
private static BulletPattern PLAYER_DEFAULT_PATTERN :

สร้างรูปแบบ bullet Pattern ตั้งต้นของ Player ดังรูป



private static BulletPattern PLAYER_BUFFED_PATTERN :

สร้างรูปแบบ bullet Pattern ขณะ Player berserk ดังรูป



private static BulletSpawner playerBulletSpawner : create bullet spawner

private static int hp, firingDelay, exp, blinkCounter :

blinkDuration = 0, initial some values

private static boolean isImmune, isHit, isVisible : check status

private static boolean isBerserk : check berserk or not

public static final int DEFAULT_SPEED = 8 : set player default speed

public static final int FOCUS_SPEED = 3 : set player focus speed

Constructor

public Player(float x, float y, double angle) :

ประกาศค่าให้ field, สร้าง bullet pattern เรียกใช้ spawnBullet

Method

getter/setter field + keyboard / mouse input handler
public void hit(Entity e) : เช็คหาชน Enemy's Bullet / Enemy มั้ย
public void shoot() : ยิง Bullet ถ้าไม่ติด delay
public void checkIsHit() :
 ถ้า ishit == true เช็คหาอยู่ในสภาพที่เลือดลด (isImmune) ได้มัย ถ้าไม่ให้ทำการลด HP และทำให้ผู้เล่นเริ่ม immune
void update() : update status player, playerskill, keyboard Input
public void render(GraphicsContext gc) : render ภาพผู้เล่น
public static void berserk(Boolean isBerserk) :
 ถ้า berserk อยู่ให้ ใช้ BulletPattern พิเศษ ในการสร้าง Bullet พร้อมวาด aura
 ทับตัว Playerถ้าไม่ให้ใช้ BulletPattern ปกติ

4.10 abstract class RangedEnemy extends Enemy

abstract class เป็นศัตรูที่สามารถโจมตีระยะไกลได้

Field

protected BulletSpawner bulletSpawner : กำหนด bulletSpawner
protected BulletPattern bulletPattern : กำหนด bulletPattern
protected int bulletSpeed : กำหนด bulletSpeed
protected int bulletRadius : กำหนด bulletRadius
protected int bulletPower : กำหนด bulletPower

Constructor

public RangedEnemy(float x, float y, double angle, int speed, int radius) :
 กำหนดค่าให้ field

Method

getter/setter field
public void spawnBullet(BulletPattern pattern) throws

NoBulletSpawnerException :

 เป็น method ที่เมื่อถูกเรียกจะทำการสร้าง Bullet ตามรูปแบบ pattern
setNewBulletSpawner(BulletPattern pattern) :
 กำหนด bulletSpawner อันใหม่ที่มี pattern ที่ใส่ไป

4.11 class Wisp extends RangedEnemy

เป็น RangedEnemy ที่จะเดินไปทั่ว arena และคอยปล่อย Bullet รอบตัว
โดยเว้นระยะเวลาหลังตอนเกิดแล้วค่อยเริ่มปล่อย Bullet

Field

int delay : กำหนดค่าให้ delay

Constructor

public Wisp(float x, float y) : กำหนดค่าให้กับ field

Method

public void render(GraphicsContext gc) : render wisp
void update() :
 ถ้า wisp ยังไม่ตาย ให้ setnewpoint() และทุกครั้งให้ทำการลดค่า delayเรื่อยๆ ถ้า
Delay หมดให้เริ่มทำการปล่อย Bullet ตามปกติ

4.12 class Witch extends RangedEnemy

เป็น RangedEnemy ที่ปล่อย Bullet 1 ใน 2 รูปแบบซึ่งมาจากการสุ่ม หลังปล่อย Bullet จะทำการพักจนพร้อมแล้วย้ายไปจุดอื่นเพื่อปล่อย Bullet ชุดต่อไป

Field

BulletPattern irregularPattern; กำหนด bulletPattern
Random random = new Random(); สร้างตัว random

Constructor

public Witch(float x, float y) : ประกาศให้ field

Method

void update() :

ถ้ายังไม่ตายให้ focusoplayer และปล่อยกระสุนตาม pattern แล้วย้ายที่ด้วย
setnewpoint หากพร้อมที่จะปล่อย Bullet ชุดใหม่

public synchronized void render(GraphicsContext gc) render

5. package main

5.1 class Main extends Application

Field

public static Main instance = new Main() : สร้าง instance
private Stage primaryStage : สร้าง primarystage
public static GameLogic logic : สร้าง gamelogic
public static GameScreen gameScreen : สร้าง gamescreen
public Menu menu : สร้าง menu
private boolean isGameSceneShown = false : เช็คว่ามี gamescene อยู่มั้ย

Method

public void start(Stage primaryStage) throws Exception :

set primarystage , call toggle scene and call show()

public static void main(String[] args) : call Application.launch(args);

public synchronized void toggleScene() throws Exception :

เรียกใช้เพื่อเปลี่ยนจากหน้าจอ menu กลับไปกลับมาหน้าจอ game

6. package utility

เก็บค่าคงที่และ method ที่มักถูกเรียกใช้

6.1 class Configuration.java

เก็บค่าคงที่เกี่ยวกับหน้าจอต่างๆ

Field

```
public static final int SCREEN_WIDTH = 1200 : ตั้งค่า width ของ screen
public static final int SCREEN_HEIGHT = 900 : ตั้งค่า height ของ screen
public static final int ARENA_WIDTH = 1200 : ตั้งค่า width ของ arena
public static final int ARENA_HEIGHT = 800 : ตั้งค่า height ของ arena
public static final int GUI_HEIGHT = SCREEN_HEIGHT - ARENA_HEIGHT :
    ตั้งค่า GuiHeight มีขนาดเท่ากับส่วนที่เหลือ
public static final int SCORE_WIDTH = 300 :
    ตั้งค่า width ของ scoreBar บน GuiBar
public static final int SKILLS_WIDTH = 600 :
    ตั้งค่า width ของ skillBar บน GuiBar
public static final int LIFE_WIDTH = 300 : ตั้งค่า width ของ lifeBar
public static final int SKILL_ICON_WIDTH = 100 : ตั้งค่า skillicon
public static final int SKILL_FONT_SIZE = 30 : ตั้งค่า skillfont
```

6.2 class GameLoop.java

ดำเนิน loop game

Field

```
GameScreen gameScreen : เก็บ gameScreen
GameLogic logic : เก็บ gameLogic
```

Constructor

```
public GameLoop(GameScreen gameScreen) :
    เก็บค่า gameScreen และ logic ของ gameScreen นั้น
```

Method

```
public void start() :
    ทำการ run loop game ที่จะอัปเดตค่าต่างๆ
```

ในเกมและตรวจสอบสถานะจบเกมพร้อมแสดงผลแพ้ถ้าเกมจบ (เกมจะจบก็ต่อเมื่อ Player มี HP เหลือ 0)

6.3 class Input Utility.java

เป็น class เกี่ยวกับ Input หากมีการตรวจสอบ Input จะเรียก method จากในนี้

6.4 class NoBulletSpawnerException.java

เป็น exception ที่สร้างขึ้นมาให้ทำการเตรียม BulletSpawner อันใหม่ถ้า BulletSpawner ทำงานเสร็จแล้วหรือไม่พบ BulletSpawner ตามปกติทุก Entity ที่มี BulletSpawner ต้องมี BulletSpawner ตลอดเวลา

How to Play

กดปุ่ม start ที่หน้า menu เพื่อเริ่มเล่นเกม คอควบคุมทิศทางด้วยปุ่ม W A S D
ภายในสนามเพื่อหลบศัตรูและลูกกระสุน ตัวละครเราหันหน้าตามจุดที่ mouse ชี้อยู่ กดคลิกซ้ายเพื่อยิงกระสุน 1
ชุดถ้าการยิงไม่ติด cooldown อยู่ เมื่อเราชนศัตรูหรือโดนกระสุนของศัตรูจะเสียเลือด 1
หน่วย(ดูได้จากหัวใจมุมล่างขวา) ตัวผู้เล่นสามารถกดปุ่ม Q E spacebar R เพื่อเรียกใช้สกิล 1, 2, 3, และ 4
ตามลำดับ ซึ่งจะช่วยให้การต่อสู้กับศัตรู แต่ละศัตรูก็จะมีรูปแบบการโจมตีต่างกัน บางตัวมีเลือดเยอะต้อง
บางตัวปล่อยกระสุนรอบตัว เกมจะจบเมื่อหัวใจหมด พร้อมแสดงคะแนนที่ทำได้ตอนจบเกม

Credit

Many of artworks **are not** made by creators of this game.

This is a list of artworks in this game and their creators :

Player and Enemy avatars from <http://opengameart.org/> include :

“Bandit Model” are derived from “Top down woodsman animations”

“Player Model” and “Witch Model” are derived from “Top down mage animations”

“Troll Model” is derived from “Top down troll, animated”

all of these are made by Warlock's Gauntlet team

(<http://musztardasarepska.pl/wgdown/>)

“Wisp Model” is derived from **Dota2 Sticker - lo** by **chroneco**

(<http://chroneco.deviantart.com/art/Dota2-Sticker-lo-463650684>)

“Desert theme” is made by **brainiac256**

(<http://opengameart.org/content/arabesque>)

“Forest theme” is made by **remaxim**

(<http://opengameart.org/content/magical-theme>)

All abilities icons and abilities' sound effect (except “Skill 4”) are from the game **Dota 2**. The **copyright** for it is held by **Valve Corporation**, who created the software.

Sound effect of “Skill 4” is from the game **Overwatch**. The **copyright** for it is held by **Blizzard Entertainment**, who created the software.

All background images of the arena in game are extracted from a Pokémon game from **Nintendo**, developed by **Game Freak**.

“Main Menu Background” is come from **Magicka 2** from **Paradox Interactive**.

“(๐_๐❀) theme” from

“「Secret base～君がくれたもの～」 cover by 【Mes x mokona x 冥月 x 洛】”

from https://www.youtube.com/watch?v=wP1_iGNIXL8

“Heart” (player lifes icon) is from game **Undertale** by **Toby Fox** (<http://undertale.com/>)