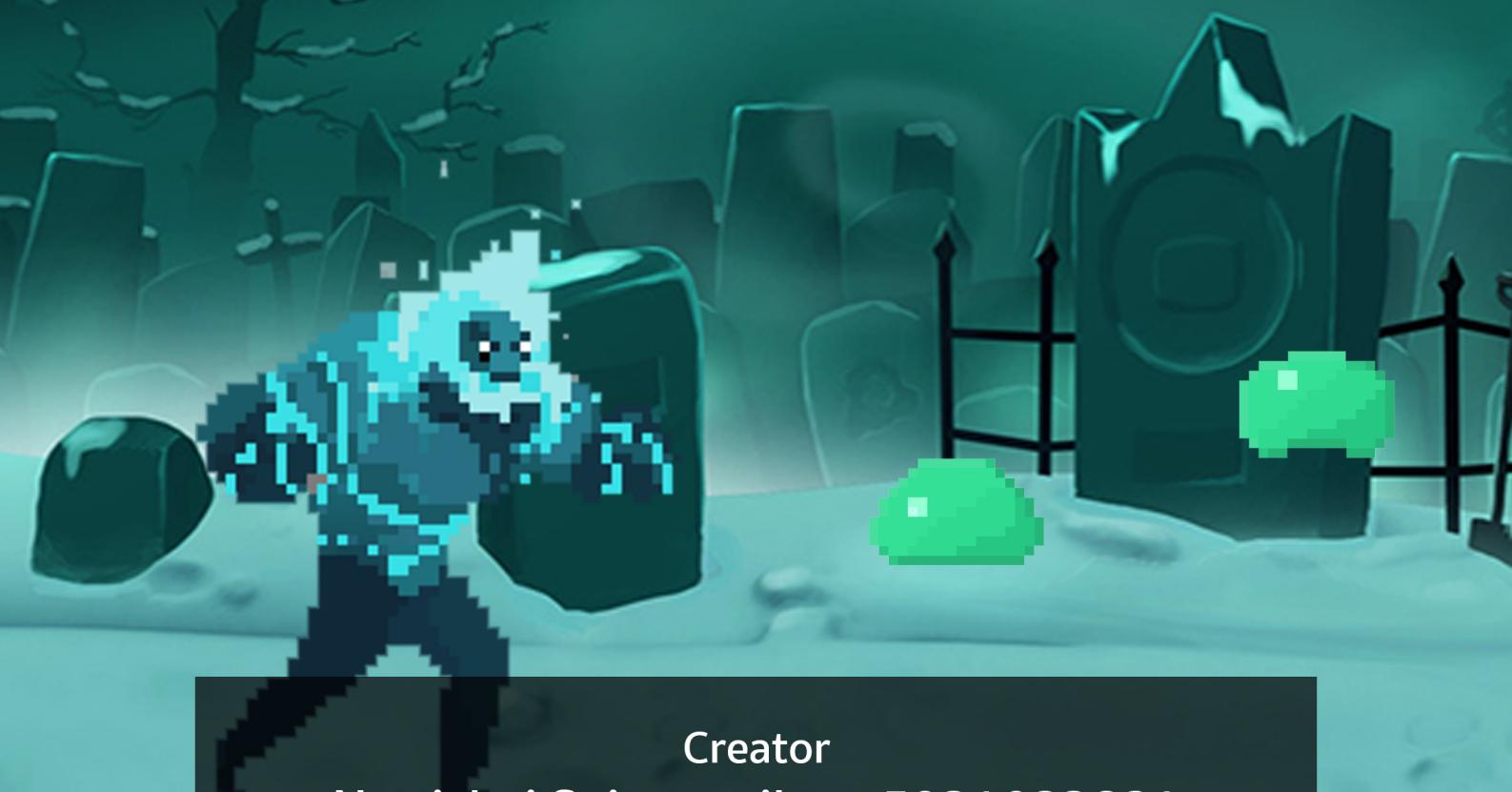


TITAN RUN



Creator

Nattichai Suitpanwihan 5931023821
Danoosit Techapetpaiboon 5931024421

Final project for
2110215 Programming Methodology I

Table of contents

Overview

Introduction	1
Story	1
Gameplay	1

Implementation Detail

1 Package Application

Class Main	14
------------	----

2 Package game.model.character

Class Boss	14
Class Monster	15
Class Player	16

3 Package gamemodel.gui.

Class GUIGradientText	18
Class GUIImage	18
Class GUIProgress	18
Class GUIShape	19
Class GUIText	19

4 Package game.model.item

Class HealthPotion	20
Class Jelly	20

5 Package game.model.obstacle

Class AirObstacle	21
Class GroundObstacle	21
Class HoleObstacle	21

6 Package game.model

Class BackgroundMusic	22
Abstract Class Characters	22
Class Effect	23
Abstract Class Entity	24
Abstract Class GUI	25
Abstract Class Item	25
Class Map	25
Class Model	26

Abstract Class Obstacle	27
Class Score	28
Class Skill	28
7 Package game.property	
Interface Animatable	29
Interface Attackable	29
Enum Direction	29
Class Hitbox	29
Interface Movable	30
Enum PowerState	30
Enum Side	30
Enum State	30
Class UserInterface	30
8 Package game.storages	
Class CharactersData	33
Class EffectsData	33
Class ScoreData	34
Class SkillsData	34
9 Package game.updater	
Class Animations	35
Class Updater	35
10 Package input	
Class GameHandler	37
Class MainMenuHandler	37
Class ScoreViewHandler	37
11 Package Scene	
Class GameMain	38
Class LaodingScreen	38
Class MainMenu	39
Class Rankings	40
Class SceneManager	40
Class ScoreView	41
12 Package utility	
Class DeleteEmptyInputException	41

Class LongInputException	41
Class ResourceManager	41
Class ShortInputException	41
Class WrongInputException	41
UML Diagram	42

Titan Run

Introduction

Titan Run is an endless running game created to be a final project in Programming Methodology I.

The main character is Titan whose aim is to escape the graveyard.

Story

This game is about one character, we call him Titan, tried to run away from the graveyard. However, on the way, there are many obstacles blocking him to go out. Moreover, mysterious monsters constantly appear and try to kill him.

Meanwhile, on the other side of the world, the news arrived at us, Titan Protector Organization. We decided to help him because Titan is nearly extinct nowadays. We have the power to control him from far away place and make him able to use abilities to fight the monsters.

Our company has a lot of works to do, so you were selected to help him. Can you help him out? It depends on your skill. Try to help him back, his destiny is in your hands.

Gameplay

1. Scenes

The game begins with **Loading scene** as shown in Figure 1, which loads all images and game assets. You just wait for a few seconds, the game will automatically transition to **Main Menu scene** (Figure 2).

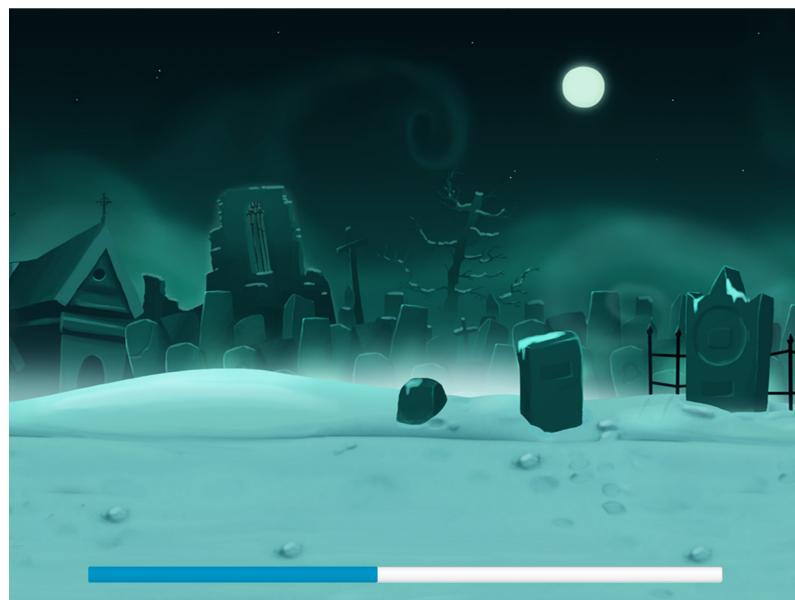


Figure 1: Loading scene

In **Main Menu scene** (Figure 2), There are 3 menus : start, rankings, and exit menus. You can use arrow keys or mouse to select each menu.



Figure 2: Main Menu scene

- **Start menu** will lead you to **Register scene** (Figure 3). You can type your name in the box which has the rule to validate your name. Your name should:
 - has between 2 - 12 characters and
 - use only a - z and A - Z alphabets.

If your name is invalid, it will tell you what is wrong in your name. On the other hand, if your name is valid, you can press ENTER to continue to **Difficulty selection scene** (as shown in Figure 4).



Figure 3: Register scene

In a **Difficulty selection scene**, you can choose how hard will the game be, by click the button or press E for easy, N for normal, H for hard and I for insane. Then it will continue to the **Game scene** (Figure 6).



Figure 4: Difficulty selection scene

- **Rankings menu** will lead you to **Rankings scene** (Figure 5). It will load score data and show top 10 rankings with their names, scores and modes. You can press ENTER to go back to **Main Menu**.

A screenshot of the game's rankings screen. The background is the same dark, atmospheric landscape as the previous screen. The title "Rankings" is centered at the top in a large, bold font. Below the title is a table showing the top 10 players. The table has columns for Rank, Name, Score, and Mode. The data is as follows:

Rank	Name	Score	Mode
1	asd	2049235	I
2	wwwwwwwwwwww	1458305	I
3	wwwwwwwwwwww	975392	H
4	wwwwwwwwwwww	948567	N
5	Po	673647	I
6	asd	443069	E
7	wwwwwwwwwwww	427894	E
8	asd	419020	I
9	asd	220331	I
10	wwwwwwwwwwww	218731	I

Enter to go back

Figure 5: Rankings scene

- **Exit menu** will let you exit the game.

The **Game scene** will start with **normal stage** (Figure 6). In normal stage, your character will run automatically. Then when you run through the whole normal stage, **boss stage** (Figure 7) will come up and you have to fight the boss. If you can defeat it, the game will continue to the next normal stage. If you die wherever, normal or boss stage, the **Death scene** will show up (Figure 8.)



Figure 6: Game scene (normal stage)



Figure 7: Game scene (boss stage)

The **Death scene** (Figure 8) is shown when you are dead, you can press ENTER to go to **Score View scene** (Figure 9 and 10).



Figure 8: Death scene

Score View scene is the summation of your score and tells what rank you are in. If your rank is more than 5, it will call unranked (Figure 9), but if your rank is in Top 5, your name will be shown in Top Runner zone (Figure 10). Press ENTER to go back to **Main Menu**.

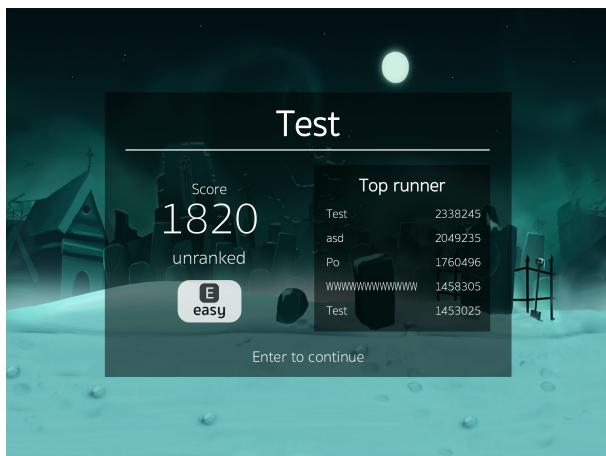


Figure 9: Score View scene (unranked)

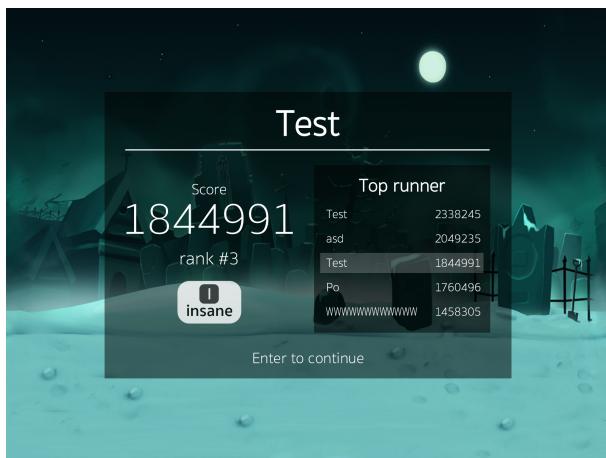


Figure 10: Score View scene (rank #3)

2. Game User Interface

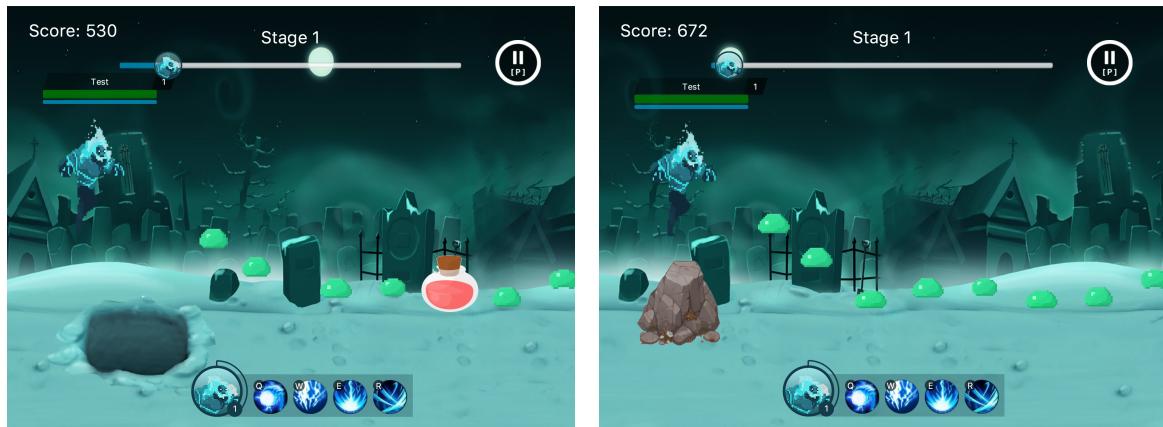


- **Top screen part :**
 - **Score** : your current score, it is the main goal of the game (*see more in **Goals** section*).
 - **Stage** : your current stage, it tells your progression in the game. The stage starts at 1. The higher number, the more challenge you will face.
 - **Distance bar** : how far you run in this stage, the destination is the boss stage.
 - **Pause button** : click it or press P to pause the game (*see more in **Controls** section*).
- **Character part :**
 - **Name** : valid name which you type in **Register scene**.
 - **Level** : how powerful you are, you can compare your level with the monsters to know who has more stat.
 - **Hit Points bar (HP bar)** : your current health, it will be green when it has more than 75%, yellow when 50%, orange when 25% and red when it has less than 25%.
 - **Mana bar** : it is a shield gage. It increases over time until it full, and decrease when you use a shield. (*see more in **Controls** section*).
 - **Your character** : he is a Titan, whom you can control.
- **Map part :**
 - Map will auto-generate items, obstacles, and monsters (*see more in **Items, Obstacles and Monsters** section*).
- **Bottom screen part :**
 - **Character icon zone** : it shows your levels and experiences (*see more in **Levels & Experiences** section*).
 - **Skills** : you have 4 skills and can use it by press QWER. Each skill has its own cooldown, this zone will show you how many cooldowns left in each skill. (*see more in **Controls** section*)

3. Controls

In normal stage

- Press **UP ARROW** to **jump**, press twice to perform a **double jump**.



jump to avoid holes and ground obstacles



double jump is needed to avoid high obstacles

- Press **DOWN ARROW** to **slide** or **fast fall down** if you are jumping



slide to penetrate through air obstacles

- Press **Q** to use **Fireball**. When you use it, the Fireball will be fired straight to your direction. It is your normal attack, so it has a very short cooldown. You can hold down the key Q to attack continuously.



Fireball has 2 colors, orange and blue. It will animate between these colors.

- Press **W** to use **Lightning skill**. Lightning will strike down from the sky, do damage to one nearest monster.



It will strike at the end of screen if there is no monster in the map

- Press **E** to use **Thunderbolt skill**. Thunderbolt will start to strike at your position, then go forward in your direction. It can deal damage to a group of monsters.



- Press **R** to use **Slashy, ultimate skill**. It will deal damage to all enemy on the screen for a moment. Since it has a long cooldown, try to use it wisely.



use it when having a swarm of monsters in the map or when fighting a boss

- Hold down **SPACEBAR** to activate **Shield**. You will be immortal as long as the shield is activated, but while using, it will consume your mana.



use shield before monster's skill comes to you, you will take no damage from it.

- Press **P** or click **Pause button** to pause/resume the game.



In boss stage

- Control is same as normal stage, but in this one, you can walk left and right by press **LEFT ARROW** and **RIGHT ARROW**.



you can move around the map to dodge boss's attacks

4. Game systems

Items

- There are 2 items in the game.
 - Green jelly**, collect it to increase your scores.



it will make more scores per piece when it is in higher stage

- Red potion**, collect it to increase your hit points, and it can make scores too.



same as jelly, in a higher stage, it gives more both health and scores

Obstacles

- There are 3 types of obstacles in the game.
 - **Hole.** You have to jump across the hole as you can see in **Controls - UP ARROW** section, else you will fall down in a hole, and dead finally.



although you have a shield, you still fall down if you do not press jump across it

- **Ground obstacle.** They have 2 heights. One can pass by single jump and the other one by a double jump (*you can check it again in **Controls - UP ARROW** section*), but if you hit it you will get damages. A shield can use to pass through it with no damage.
- **Air obstacles.** You have to slide, otherwise, you will get some damages. Anyway, you can use a shield to penetrate through it same as ground obstacles.



Monsters

- The monsters will jump in from right of the screen, then it moves back to its position. After a moment, it will use a skill to attack you. Each monster has a different unique skill, different hit points, and different attack value.



when you defeat a monster, it will give you some experience and scores.

Boss

- Boss will come when you reach the end of stage. Before it comes, it will have a warning warn you in advance.



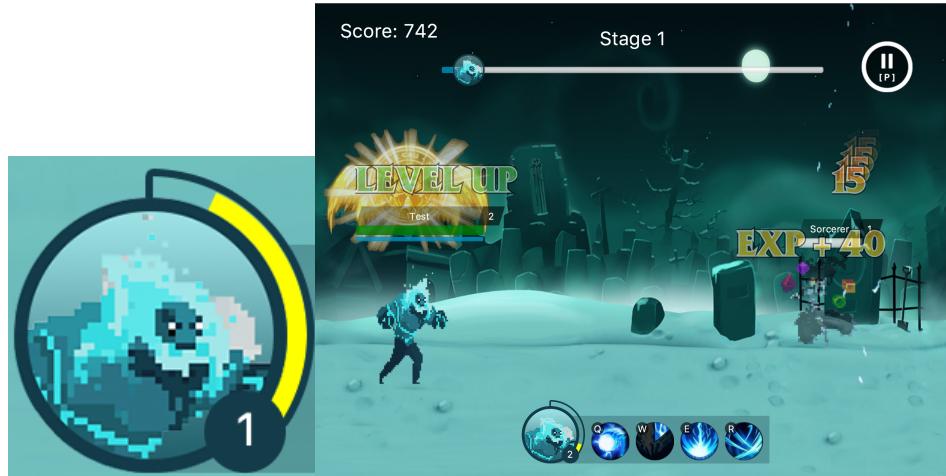
- Boss has many skills and movements, but all are in a cycle. Try to figure it out when you fight the boss.



- If you can kill a boss, your character will auto-walk back to running position, and continually run to next stage.

Levels & Experiences

- **Yellow gage** is your experience bar. You can get experiences from monster and boss kill. When it fills full, you will level up.
- **Level up** will increase your hit points, attack damage, and skill damage.



5. Goals

Only one goal you should aim is **making a high score**. No matter which difficulty mode you choose because this game has only one scoreboard. The game was balanced to weight score in each mode equally. If you play the game with same skill play, you will get score in each mode approximately close.

You can get scores by collect items, and kill monsters or bosses. The higher stage will give scores increased significantly, so try to go as far as you can.

Implementation Detail

1 Package application

1.1 Class Main extends Application

1.1.1 Method

+ void start(Stage primaryStage)	The main entry point for the JavaFX applications.
+ void main(String[] args)	An entry point of the application.

2 Package game.model.character

2.1 Class Boss extends Monster

2.1.1 Field

- Timeline moveTo	A Timeline for boss moving to one position.
- Timeline goAroundTimeline	A Timeline for boss going around the screen.
- Timeline attackTimeline	A Timeline for boss attacking.
- Timeline moveTimeline	A Timeline for boss moving to many positions.
- Timeline beamTimeline	A Timeline for boss using beam.
- boolean isReady	The flag indicate that the boss is ready.
- int step	The current step of boss action.
- boolean isActioning	The flag indicate that the boss is actioning.
- double bossSpeed	Boss speed.

2.1.2 Constructor

+ Boss(double x, double y, int idx)	Initialize boss by using the idx character data from CharactersData at position (x, y).
-------------------------------------	---

2.1.3 Method

+ void move()	Change name position's UI, level position's UI and current position by speed and accel.
+ void moveTo(double posX, double posY, double duration)	Move the boss to position (posX, posY) in duration millisecs by using moveTo timeline.
+ void update()	If boss is not actioning and game is not pausing, then do current step action. The step cycle is go around 2 times, attack, go around again, then use beam.
- void goAround()	Move boss to 4 corners of screen by using goAroundTimeline
- void attack()	Move boss up and down at the right of the screen 3 times and shoot Drill skill simultaneously by using attackTimeline and moveTimeline.

- void beam()	Move boss to overhead player position, show Charge effect and shoot Beam down to player. Do this 5 times by using beamTimeline.
+ void decreaseHp(double d)	Decrease current hit points by d. Check if it died, then update hp bar's UI and show damage text.
+ void die()	Give player the scores and experiences, stop all timeline, create boss dead transition(shaking and fading away), then continue to next stage.
+ void pauseAllTimeline()	Check all timelines if it is not null and is running, then pause it.
+ void continueAllTimeline()	Check all timelines if it is not null and is pausing, then play it.
+ void stopAllTimeline()	Check all timelines if it is not null, then stop it.
+ Getter & Setter methods	

2.2 Class Monster extends Characters

2.2.1 Field

# boolean canMove	The flag indicate that this monster can move.
# boolean canMoveOut	The flag indicate that this monster can move out of the screen.

2.2.2 Constructor

+ Monster(double x, double y, int idx)	Initialize monster by using the idx character data from CharactersData at position (x, y), then update name's and level's UI.
--	---

2.2.3 Method

+ void draw()	Fix canvas's direction, then draw current animation image.
+ void move()	Fix speed and change position by speed if it can move, but if it in the stay position, it will stay for a moment. After all, update name position's UI, level position's UI, hp bar position's UI and this monster position.
- void stayInMap()	Called when this monster is in stay position. Used timer timeline to do <ul style="list-style-type: none"> - Wait 1 sec, then use monster's skill. - Wait 4 secs Do this cycle 2 times, then set canMoveOut to true.
- void useSkill()	Check if monster is not dead yet, then use skill by create Skill and add it to container.
+ void changeImage()	Add currentAnimation by 1 then draw.
+ boolean isCollision(Entity e)	Check if hp less than 0.00001 or they are same side, then check hitboxes, if it true return true, otherwise return false.
+ void affectTo (Characters player)	Deal damage to player equal monster's atk.

+ void injured()	Make monster blinking when it is attacked.
+ void die()	Give scores and experiences to player, stop all timeline, then fade this monster away.
+ boolean isDead()	Check if canvas opacity is zero or it is out of the screen, then remove all of the monster's UI and return true. Otherwise, return false.
+ Getter & Setter methods	

2.3 Class Player extends Characters

2.3.1 Field

+ double LIMIT_NORMAL_ATTACK	Max number of normal attack player can shoot in the screen.
+ double MOVEDOWN_SPEED	Fast fall down's speed.
+ int MAX_JUMP	Max number of jumping times.
# double[] fullCooldown	Full cooldown of each skill.
# State state	Player state.
# Image imageSlide	Sliding image.
# int jump	Number to count jumping times.
# boolean isInjuring	The flag indicate that player is injuring.
# double mode	Game difficulty mode.
# int stage	Current stage.
# double distance	Current distance.
# double score	Current score.
# double exp	Current experience.
# double maxExp	Required experiences to level up.
# double mana	Current mana.
# double maxMana	Max amount of mana.
# double[] cooldown	Current cooldown of each skill.

2.3.2 Constructor

+ Player(double x, double y, int idx)	Initialize player by using the idx character data from CharactersData at position (x, y), and update UI.
---------------------------------------	--

2.3.3 Method

+ void draw()	Fix canvas's direction, then draw current animation image.
+ void move()	Change speed, direction, position, state and update UI.
+ void changeImage()	Add currentAnimation by 1 then draw.
+ void affectTo(Characters e)	Attack character e by player's attack.

+ void jump()	Jump if jump not reach MAX_JUMP and play jump sound.
+ void goDown()	If player is not jumping, then slide. Otherwise, increase speed by MOVEDOWN_SPEED
+ void slide()	Play slide sound. Set state to JUMPING, rotate canvas then draw.
+ void makeShield()	If have enough mana, then create shield.
+ void useShield()	Decrease current mana and set state.
+ void useFireball()	Create fireball at player position if player is not sliding and current fireball cooldown is below zero.
+ void useLightning()	Create lightning skill at nearest master position if player is not sliding and current lightning cooldown is below zero.
+ double nearestMonsterPosition()	Search for nearest monster, if there are no monster then return the end of screen position.
+ void useThunderbolt()	Create thunderbolt skill at player position if player is not sliding and current thunder cooldown is below zero.
+ void useSlashy()	Create slashy skill to cover screen if player is not sliding and current slashy cooldown is below zero.
+ void backToRunningPosition()	Called after passing boss stage, set player state, position, speed and direction to running position.
+ void injured()	Set power state to be immortal, make player blinks and play injured sound.
+ void die()	Prepare score view by send player's name, score and game mode to ScoreView. Play player's dead transition, then show dead text.
+ boolean isDead()	Player is dead when hp is zero or canvas's opacity is zero.
+ void setName(String s)	Set player's name to be s and update UI.
+ void addExp(double e)	Add experience by e, check if it level up, then update UI.
+ void addLevel(int lvl)	Add level by lvl, show level up effect, text and sound, then increase player's stat. After all. update UI.
+ void addMana(double mana)	Add current mana by mana, then update UI.
+ void addJump(int jump)	Add current jump times by jump.
+ void addScore(double s)	Add score by s, then update UI.
+ void resetCooldown(int index)	Reset the idx skill cooldown to be fullCooldown of that skill.
+ void decreaseCooldown(double cd)	Decrease all skill's cooldown by cd, then update UI.
+ void addDistance(double d)	Add current distance by d, then update UI.
+ Getter & Setter methods	

3 Package game.model.gui

3.1 Class GUIGradientText extends GUI

3.1.1 Field

- Font FONT	Friz Quadrata Bold font. Created for damage, heal, exp plus and level up text.
- Stop[] stops	Gradient color stops for all needed color.
- LinearGradient[] FONT_COLOR	Gradient color fonts.

3.1.2 Constructor

+ GUIGradientText(double x, double y, String s, int i)	Initialize gradient text with s string and using FONT_COLOR[i], create at position (x, y). Make and play translating and fading transition.
--	---

3.2 Class GUIImage extends GUI

3.2.1 Field

- Image image	Image of this GUI.
---------------	--------------------

3.2.2 Constructor

+ GUIImage(double x, double y, double w, double h, Image i)	Initialize image by using image i, and create its canvas at position (x, y) with size (w, h).
+ GUIImage(double x, double y, double w, double h, Image i, double width, double height)	Initialize image by using image i, and create its canvas at position (x, y) with size (w, h). Set size of image which will draw later to (width, height).

3.2.3 Method

+ void draw()	Draw image at the center of its canvas.
---------------	---

3.3 Class GUIProgress extends GUI

3.3.1 Field

- Paint paint	Color of this progress.
- double startAngle	Start angle of this progress.
- double progress	Current progress.
- double range	Range of progress angle.

3.3.2 Constructor

+ GUIProgress(double x, double y, double w, double h, double d)	Default progress, initialize black color progress with d progress, size (w, h) and at position (x, y).
+ GUIProgress(double x, double y, double w, double h, Paint p, double s, double d, double r)	Initialize progress with d progress with color p, size (w, h), start at s angle, in range r, and at position (x, y).

3.3.3 Method

+ void draw()	Draw progress following the initial condition.
---------------	--

+ void setProgress(double progress)	Setter of progress.
-------------------------------------	---------------------

3.4 Class GUIShape extends GUI

3.4.1 Field

- Paint paint	Color of this shape.
- double opacity	Opacity of this shape.

3.4.2 Constructor

+ GUIShape(double x, double y, double w, double h, Paint p, double o)	Initialize defualt shape, rectangle, with color p, opacity o, size (w, h) at position (x, y).
+ GUIShape(double x, double y, double w, double h, Paint p, double o, String s)	Initialize s shape with color p, opacity o, size (w, h) at position (x, y).

3.4.3 Method

+ void draw()	Draw default shape, rectangle, following the initial condition.
+ void drawOval()	Draw oval following the initial condition.

3.5 Class GUIText extends GUI

3.5.1 Field

- String text	Text of this GUI.
- Paint paint	Color of this text.
- Font font	Font of this text.

3.5.2 Constructor

+ GUIText(double x, double y, double w, double h, String s, Paint p, double fs)	Initialize text with string s with color p, font size fs, size (w, h), and at position (x, y).
+ GUIText(double x, double y, double w, double h, String s, Paint p, Font f)	Initialize text with string s with color p, font f, size (w, h), and at position (x, y).

3.5.3 Method

+ void draw()	Draw text following the initial condition.
+ void drawDamage()	Draw damage text following the initial condition.
+ Getter & Setter methods	

4 Package game.model.item

4.1 Class HealthPotion extends Item

4.1.1 Field

- double POTION_WIDTH	Default potion width.
- double POTION_HEIGHT	Default potion height.
- Image[] images	Animation images of the potion.

4.1.2 Constructor

+ HealthPotion(double x, double y)	Initialize health potion at position (x, y).
+ HealthPotion()	Created for loading images before game starting only, so do nothing.

4.1.3 Method

+ void draw()	Draw current animation image.
+ void changeImage()	Add currentAnimation by 1, then draw.
+ void affectTo(Characters player)	Give hit points and scores to player.

4.2 Class Jelly extends Item

4.2.1 Field

- double JELLY_WIDTH	Default jelly width.
- double JELLY_HEIGHT	Default jelly height.
- Image[] images	Animation images of the jelly.

4.2.2 Constructor

+ Jelly(double x, double y)	Initialize jelly at position (x, y).
+ Jelly()	Created for loading images before game starting only, so do nothing.

4.2.3 Method

+ void draw()	Draw current animation image.
+ void changeImage()	Add currentAnimation by 1, then draw.
+ void affectTo(Characters player)	Give hit points and scores to player.

5 Package game.model.obstacle

5.1 Class AirObstacle extends Obstacle

5.1.1 Field

- int AIR_HEIGHT	Default air obstacle height.
- Image[] images	Images of air obstacle.

5.1.2 Constructor

+ AirObstacle()	Initialize air obstacle by random image from database. Create jellies around it, then draw.
-----------------	--

5.1.3 Method

+ void draw()	Draw air obstacle image.
+ boolean isCollision(Player e)	Return true if player is not sliding, return false otherwise.

5.2 Class GroundObstacle extends Obstacle

5.2.1 Field

- Image[] images	Images of ground obstacle.
------------------	----------------------------

5.2.2 Constructor

+ GroundObstacle()	Initialize ground obstacle by random image from database. Create jellies around it, then draw.
--------------------	---

5.2.3 Method

+ void draw()	Draw ground obstacle image.
---------------	-----------------------------

5.3 Class HoleObstacle extends Obstacle

5.3.1 Field

- int HOLE_WIDTH	Default hole width.
- Image[] images	Images of hole.

5.3.2 Constructor

+ HoleObstacle()	Initialize hole by random image from database. Create jellies around it, then draw.
------------------	---

5.3.3 Method

+ void draw()	Draw hold image.
+ boolean isCollision(Player e)	Check if player is not jumping, player will die.

6 Package game.model

6.1 class BackgroundMusic

6.1.1 Field

- Media mainMenuBGM	Backgroundmusic in main menu
- Media normalStageBGM	Backgroundmusic in normal stage
- Media bossStageBGM	Backgroundmusic in boss stage
- AudioClip warningSiren	Warning sound
- MediaPlayer mediaPlayer	MediaPlayer of normalStageBGM

6.1.2 Method

+ void playMainMenuBGM()	Play main menu BGM.
+ void playNormalStageBGM()	Play normal stage BGM.
+ void playBossStageBGM()	Play boss stage BGM.
+ void playWarningSiren()	Play warning sound.
+ void startMusic(Media media)	Play music.
+ void pauseMusic()	Pause the music.
+ void continueMusic()	Continue the music.
+ boolean isMainMenuBGMPlaying()	Return true if main menu BGM is playing.

6.2 Abstract Class Characters extends Entity implements Movable, Animatable, Attackable

6.2.1 Field

# int nImage	Number of character images.
# Image[] images	Images of the character.
# int currentAnimation	The index of current animation image.
# double speedX	Character speed in X-axis.
# double speedY	Character speed in Y-axis.
# double accelX	Character acceleration in X-axis.
# double accelY	Character acceleration in Y-axis.
# String name	Name of this character.
# int level	Level of this character.
# double hp	Current hit points of this character.
# double maxHp	Max hit points of this character.
# double atk	Attack damage of this character.
# int skillIndex	The index of character's skill.

# Direction direction	Current character's direction.
# Direction imageDirection	Defualt character's image direction.
# PowerState powerState	Current power state of this character.
# UserInterface userInterface	This character's UI.
# Timeline timer	Character timer's timeline.
# Timeline skillTime	Character skill's timeline.
# Timeline blink	Blink's timeline.

6.2.2 Constructor

+ Characters(double x, double y, int idx)	Initialize Character by using the idx character and creating at (x, y) position.
---	--

6.2.3 Method

+ void draw()	Draw character animation images.
+ void changeSpeed(double accelX, double accelY)	Change current speed by accelerator in both X and Y-axis.
+ void move()	Move character.
+ void updatePosition()	Translate canvas to (positionX, positionY).
+ void changeImage()	Change current animation image.
+ void increaseHp(double d)	Add current hit points by d, then updates hp bar's UI and show heal text.
+ void decreaseHp(double d)	Decrease current hit points by d*game difficulty. Check if it dead, then call die(). Update hp bar's UI show damage text, and call injured().
+ void injured()	Called when character is attacked.
+ void die()	Called when character is dead()
+ boolean isDead()	Check if this character died.
+ void pauseAllTimeline()	Pause all running timeline.
+ continueAllTimeline()	Play all pausing timeline.
+ stopAllTimeline()	Stop all timeline.
+ Getter & Setter methods	

6.3 Class Effect extends Entity implements Animatable

6.3.1 Field

# int nImage	Number of effect images.
# Image[] images	Images of the effect.
# int currentAnimation	The index of current animation image.
# int currentRound	Current round of this animation.

# int round	Total round which this effect will play.
-------------	--

6.3.2 Constructor

+ Effect(double x, double y, int idx, int r)	Initialize effect by using the idx effect from EffectsData at position (x,y).
--	---

6.3.3 Method

+ void draw()	Draw current animation image.
+ void changeImage()	Change current animation image to the next one, then draw.
+ boolean isCollision(Entity e)	Since effect can not collision, so always return false.
+ void affectTo(Characters e)	Effect do not have effect now, so do nothing.
+ boolean isDead()	Effect will dead when currentRound >= round.

6.4 Abstract Class Entity

6.4.1 Field

# Canvas canvas	Entity's canvas.
# double positionX	The position in X-axis. Current position in X-axis.
# double positionY	The position in Y-axis. Current position in Y-axis.
# double width	Width of canvas. Canvas's width.
# double height	Height of canvas. Canvas's height.
# Hitbox hb	Hitbox of this entity. Entity's hit box.
# boolean isCollided	The flag indicate that this entity is collided or not.
# Side side	Entity's side.

6.4.2 Constructor

+ Entity(double x, double y, double w, double h,)	Initialize canvas with size (w, h) at position (x, y). Set hit box to cover canvas.
+ Entity()	Created for letting subclasses can load their resources when program start.

6.4.3 Method

+ void draw()	Draw image in a canvas.
+ void drawHb()	Draw hit box for debugging.
+ boolean isCollision(Entity e)	Check if this entity collision with entity e.
+ void affectTo(Characters e)	Called when this entity has effect to character e.
+ Getter & Setter methods	

6.5 Abstract Class GUI extends Entity

6.5.1 Constructor

+ public GUI(double x, double y, double w, double h)	Initialize this gui in canvas with size (w, h) at position (x, y).
--	--

6.5.2 Method

+ void draw()	Draw GUI.
+ void affectTo(Characters e)	GUI has no effect now, so do nothing.
+ boolean isDead()	GUI dies when it has zero opacity.

6.6 Abstract Class Item extends Entity implements Movable, Animatable

6.6.1 Field

# double speedX	Item speed in X-axis.
# double speedY	Item speed in Y-axis.
# int currentAnimation	The index of current animation image.

6.6.2 Constructor

+ Item(double x, double y, double w, double h)	Initialize item in canvas with size (w, h) at position (x, y).
+ Item()	Created for loading images before game starting only, so do nothing.

6.6.3 Method

+ void draw()	Draw item image.
+ void changeSpeed(double accelX, double accelY)	Item hasn't acceleration now, so do nothing.
+ void move()	Change current position by speed, then updatePosition.
+ void updatePosition()	Translate canvas to current position.
+ void changeImage()	Change current animation image.
+ boolean isDead()	Item dies when it collided already or it is out of the screen.
+ Getter & Setters methods	

6.7 Class Map extends Entity implements Movable

6.7.1 Field

+ double FLOOR_HEIGHT	Default game's floor height.
+ double GRAVITY	Default gravity's acceleration.
+ double PASSIVE_DAMAGE	Default passive damage per frame.
+ double PASSIVE_SCORE	Default passive score per frame.
+ double PASSIVE_MANA_REGEN	Default passive mana regeneration speed.
+ Image[] images	Images of map.

# double speedX	Map speed in X-axis.
# double speedY	Map speed in Y-axis.

6.7.2 Constructor

+ Map(double x, double y, double w, double h)	Initialize map in canvas with size (w, h) at position (x, y), set speed, then draw.
+ Map()	Created for loading images before game starting only, so do nothing.

6.7.3 Method

+ void draw()	Draw map image with 2 times width because it is used to be loop background.
+ void changeSpeed(double accelX, double accelY)	Map hasn't accelerator now, so do nothing.
+ void move()	Change current position by speed, then updatePosition.
+ void updatePosition()	Translate canvas to current position.
+ void affectTo(Characters e)	Map has no effect now, so do nothing.
+ boolean isDead()	Map is use to be loop background, so it's never die.
+ Getter and Setter methods	

6.8 Class Model

6.8.1 Field

- Model container	Singleton of the model, it collects every panes, so it's called container.
- Pane mapPane	Map pane, collects all map's canvas.
- Pane playerPane	Player pane, collects all player's canvas.
- Pane obstaclePane	Obstacle pane, collects all obstacle's canvas.
- Pane skillPane	Skill pane, collects all skill's canvas.
- Pane guiPane	GUI pane, collects all GUI's canvas and other canvas.
- Pane itemPane	Item pane, collects all item's canvas.
- Pane monsterPane	Monster pane, collects all monster's canvas.
- Pane effectPane	Effect pane, collects all effect's canvas.
- Map map	The map which use every scene.
- Player player	A player.
- ArrayList<Obstacle> obstacleList	List of obstacles.
- ArrayList<Skill> skillList	List of skills.
- ArrayList<GUI> guiList	List of GUIs.
- ArrayList<Item> itemList	List of items.

- ArrayList<Monster> monsterList	List of monsters.
- ArrayList<Effect> effectList	List of effects.

6.8.2 Constructor

+ Model()	Initialize all panes and lists.
-----------	---------------------------------

6.8.3 Method

+ void initialize()	Initialize container, add map and player, then set gameHandler's player.
+ void add(Object object)	Classify which object's class is, then add object's canvas to that class pane and add this object to that class list.
+ void remove(Object object)	Classify which object's class is, then remove object's canvas from that class pane and remove this object from that class list.
+ void clearAllData()	Called when change scene and the things which can still remain are monsters and skills, so clear only monster and skill.
+ Getter and Setter methods	

6.9 Abstract Class Obstacle extends Entity implements Movable

6.9.1 Field

# int OBSTACLE_WIDTH	Default obstacle width.
# double OBSTACLE_DAMAGE	Default obstacle damage.
# Image obstacle	Obstacle image.
# double speedX	Obstacle speed in X-axis.
# double speedY	Obstacle speed in Y-axis.

6.9.2 Constructor

+ Obstacle(double x, double y, double w, double h)	Initialize obstacle in canvas with size (w, h) at position (x, y), and set speed.
+ Obstacle()	Created for loading images before game starting only, so do nothing.

6.9.3 Method

+ void draw()	Draw obstacle image.
+ void changeSpeed(double accelX, double accelY)	Map hasn't accelerator now, so do nothing.
+ void move()	Change current position by speed, then updatePosition.
+ void updatePosition()	Translate canvas to current position.
+ void affectTo(Characters player)	If it is a hole, play will die. Otherwise, player will get damage.
+ boolean isDead()	Return true if it is out of the screen.

6.10 Class Score implements Serializable

6.10.1 Field

- long serialVersionUID	File encryption number.
+ String name	Score's player name.
+ double score	Score's total score.
+ double mode	Score's game difficulty mode.

6.10.2 Constructor

+ Score(String name, double score, double mode)	Initialize score by name, score and mode.
---	---

6.11 Class Skill extends Entity implements Movable, Animatable

6.11.1 Field

# int index	The index of skill which this skill will load from SkillsData.
# int nImage	Number of skill images.
# Image[] images	Images of the skill.
# double speedX	Skill speed in X-axis.
# double speedY	Skill speed in Y-axis.
# double accelX	Skill acceleration in X-axis.
# double accelY	Skill acceleration in Y-axis.
# double damage	Skill damage.
# double cooldown	Skill cooldown.
# double collisionDelay	Skill's damage will delay in milliseconds unit.
# int currentAnimation	The index of current animation image.
# int lastAnimation	The last index of animation image which this skill will die.
# Characters owner	Owner of this skill.
# boolean isStickToOwner	The flag indicate that this skill is stick to the owner position.
# boolean isOnceCollision	The flag indicate that this skill can collide only once.
# AudioClip audioClip	Skill sound effect.

6.11.2 Constructor

+ Skill(double x, double y, int idx, Characters owner)	Initialize skill by using the idx skill data from SkillsData at position (x, y), and set owner.
--	---

6.11.3 Method

+ void draw()	Draw current animation image.
+ void changeSpeed(double accelX, double accelY)	Change current speed by acceleration in both X and Y axis.

+ void move()	Change current position by speed or by owner position if it is stick to owner, then updatePosition.
+ void updatePosition()	Translate canvas to current position.
+ void changeImage()	Add currentAnimation by 1, then draw.
+ void affectTo(Characters character)	Decrease character's hp by owner's attack times this skill's damage.
+ boolean isCollision(Entity e)	Check if this skill collide with entity e by following this skill initial condition.
+ boolean isDead()	Skill dies when owner died, it is out of the screen, it can hit only once and hit already, canvas opacity is zero, or it's boss skill and boss is not ready.
+ Getter & Setter methods	

7 Package game.property

7.1 Interface Animatable

7.1.1 Method

+ void changeImage()	Change current animation image to next animation image, then draw.
----------------------	--

7.2 Interface Attackable

7.2.1 Method

+ void increaseHp(double d)	Add current hit points by d.
+ void decreaseHp(double d)	Decrease current hit points by d.
+ void injured()	Show injured effect (blinking).
+ void die()	Called when it died.
+ boolean isDead()	Check if it is dead.

7.3 Enum Direction

7.3.1 Field

+ Direction RIGHT	Right face direction.
+ Direction LEFT	Left face direction.

7.4 Class Hitbox

7.4.1 Field

+ double x	Position of the top-left point of hitbox in X-axis.
+ double y	Position of the top-left point of hitbox in Y-axis.
+ double w	Hitbox width.
+ double h	Hitbox height.

7.4.2 Constructor

+ Hitbox(double x, double y, double w, double h)	Initialize hitbox by x, y, w and h.
--	-------------------------------------

7.5 Interface Movable

7.5.1 Method

+ void changeSpeed(double accelX, double accelY)	Change current speed by acceleration
+ void move()	Move object.
+ void updatePosition()	Translate canvas to current position.

7.6 Enum PowerState

7.6.1 Field

+ PowerState IMMORTAL	Make this object immortal, don't get any damage.
+ PowerState NORMAL	Normal state, which can be attacked normally.

7.7 Enum Side

7.7.1 Field

+ Side PLAYER	Player side.
+ Side MONSTER	Monster side.
+ Side NEUTRAL	Neutral side.

7.8 Enum State

7.8.1 Field

+ State STILL	Standing still state.
+ State RUNNING	Running state.
+ State JUMPING	Jumping state.
+ State SLIDING	Slideing state.

7.9 Class UserInterface

7.9.1 Field

- Model container	Copy container for easier usage.
- Canvas nameArea	Name background/area.
- GUIText nameText	Name text UI.
- Canvas levelArea	Level background/area.
- GUIText levelText	Level text UI.
- ProgressBar hpBar	Hit points bar UI.
- ProgressBar manaBar	Mana bar UI.

- GUIShape cooldownArea	Cooldown oval background/area.
- GUImage[] cooldownIcon	Cooldown icon of each skill.
- GUIProgress[] cooldownProgress	Progress UI of the current cooldown compare with full cooldown of each skill.
- GUIShape[] cooldownTextArea	Skill label background/area.
- GUIText[] cooldownText	Skill label text UI.
- GUImage expArea	Experience background image.
- GUIProgress expProgress	Progress UI of the current experience compare with required experience to level up.
- GUIText expLevel	Level label tet UI in expArea.
- GUIText stageText	Number of the current stage UI.
- ProgressBar distanceBar	Distance bar UI.
- GUImage playerIcon	Player icon.
- GUIText scoreText	Score text UI.
- Canvas pauseArea	Pause background/area.
- GUImage pauseButton	Pause image.
- GUImage youAreDead	Death scene image.

7.9.2 Constructor

+ UserInterface(Characters character)	Initialize contianer, and add appropiate UI to the character.
---------------------------------------	---

7.9.3 Method

- void addNameUI(double length)	Create name UI with length width, and add it to container.
- void addLevelUI()	Create level UI, and add to container.
- void addHpBarUI(double length)	Create hit points bar UI with length width, and add it to container.
- void addManaBarUI()	Create mana bar UI, and add it to container.
- void addCooldownUI()	Create cooldown UI, and add it to container.
- void addExpUI()	Create experience UI, and add it to container.
- void addStageUI()	Create stage UI, and add it to container.
- void addDistanceUI()	Create distance UI, and add it to container.
- void addScoreUI()	Create score UI, and add it to container.
- void addCutSceneUI()	Create all cut scenes UI, and add them to container.
+ void updateNamePos(double x, double y)	Relocate name UI position.

+ void updateLevelPos(double x, double y)	Relocate level UI position.
+ void updateHpPos(double posX, double posY)	Relocate hit points bar UI position.
+ void updateManaPos(double posX, double posY)	Relocate man bar UI position.
+ void updateName(String name)	Set name text to be new name.
+ void updateLevel(int level)	Set both level label text to be new level.
+ void updateHp(double progress)	Set progress of the hit points bar, and fix its color.
+ void updateMana(double progress)	Set progress of the mana bar.
+ void updateCooldown(int index, double progress)	Set progress of the index skill cooldown, then redraw.
+ void updateExp(double progress)	Set progress of the experience, then redraw.
+ void updateDistance(double progress)	If progress is below 1, it's normal stage, so set progress of the distance bar and translate player icon position. Otherwise, it's boss stage, set opacity of distance UI to be zero.
+ void updateScore(double score)	Set score text to be new score.
- void initializePauseArea()	Initialize pause scene UI.
+ void showPauseArea()	If pause scene UI isn't initialize yet, then do it. After that, show it on screen by add it to container.
+ void closePauseArea()	Remove pause scene UI from the container.
+ void dead(Characters character)	If this character is monster, then remove all monster UI, but if it is player, clear guiPane and guiList, then add death scene UI to the container.

8 Package game.storage

8.1 Class CharactersData

8.1.1 Field

- String name	Character name.
- int nImage	Number of character animation images.
- Image[] images	Images of the character.
- double width	Character width.
- double height	Character height.
- Hitbox hb	Character hitbox.
- double speedX	Character speed in X-axis.
- double speedY	Character speed in Y-axis.
- double accelX	Character acceleration in X-axis.
- double accelY	Character acceleration in Y-axis.
- double hp	Character hit points.
- double maxHp	Max character hit points.
- double atk	Character attack damage.
- int skillIndex	The index of the skill of this character.
- Direction imageDirection	Character's image direction.
- PowerState powerState	Character power state.
+ CharactersData[] data	Characters data base.

8.1.2 Constructor

+ CharactersData()	Default constructor.
--------------------	----------------------

8.1.3 Method

+ Getter & Setter methods	
---------------------------	--

8.2 Class EffectsData

8.2.1 Field

- int nImage	Number of effect animation images.
- Image[] images	Images of the effect.
- double width	Effect width.
- double height	Effect height.
+ EffectsData[] data	Effects data base.

8.2.2 Constructor

+ EffectsData()	Default constructor.
-----------------	----------------------

8.2.3 Method

Getter & Setter methods	
-------------------------	--

8.3 Class ScoreData implements Serializable

8.3.1 Field

- long serialVersionUID	File encryption number.
+ List<Score> data	List of score's data.

8.3.2 Constructor

+ ScoreData()	Initialize score's data.
---------------	--------------------------

8.3.3 Method

+ void addData(String name, double score, double mode)	Load score's data save or initialize it if there hasn't save data yet. Add new score with name, score and mode to score's data, then save it in a file.
+ List<Score> getData()	Load and return score's data.

8.4 Class SkillsData

8.4.1 Field

- int nImage	Number of skill animation images.
- Image[] images	Images of skills.
- double width	Skill width.
- double height	Skill height.
- Hitbox hb	Skill hitbox.
- double speedX	Skill speed in X-axis.
- double speedY	Skill speed in Y-axis.
- double accelX	Skill acceleration in X-axis.
- double accelY	Skill acceleration in Y-axis.
- double damage	Skill damage.
- double cooldown	Skill cooldown.
- double collisionDelay	Skill's damage will delay in milliseconds unit.
- int currentAnimation	The index of current animation image.
- int lastAnimation	The last index of animation image which this skill will die.
- boolean isStickToOwner	The flag indicate that this skill is stick to the owner position.
- boolean isOnceCollision	The flag indicate that this skill can collide only once.
- AudioClip skillfx	Skill sound effect.
+ SkillsData[] data	Skills data base.

8.4.2 Constructor

+ SkillsData()	Initialize some skill's field.
----------------	--------------------------------

8.4.3 Method

+ Getter & Setter methods	
---------------------------	--

9 Package game.updater

9.1 Class Animations

9.1.1 Field

+ double ANIMATATION RATE	Animation's frame rate in frames per second unit.
+ double LOOP TIME	Animation's loop time in milliseconds.
- Timeline timerAnimation	Animation timeline.
- Player player	A player.

9.1.2 Constructor

+ Animations()	Initialize a player.
----------------	----------------------

9.1.3 Method

+ void startAnimation()	Start animation's timeline.
+ void pauseAnimation()	Pause animation;s timeline.
+ void continueAnimation()	Continue animation's timeline.
+ void stopAnimation()	Stop animation's timeline.
- void animateAll()	Change image of player, skills, items, monsters and effects.
+ Timeline getTimerAnimation()	Getter of timerAnimation.

9.2 Class Updater

9.2.1 Field

+ double FPS	Game's frame rate in frames per second unit.
+ double LOOP TIME	Game's loop time in milliseconds.
- Timeline timerUpdate	Game's loop timeline.
- Map map	A map.
- Player player	A player.
- Boss roboteck	A boss.
- double spaceObstacle	Space between each obstacle.
- double spaceItem	Space between each item.
- double spaceMonster	Space between each monster.
- double distance	Current distance in this stage.
- boolean isBossStage	The flag indicate that current stage is the boss stage.

9.2.2 Constructor

+ Updater()	Initialize map, spaceObstacle, spaceItem, spaceMonster, distance and isBossStage.
-------------	---

9.2.3 Method

+ void startGame()	Start game loop.
+ void pauseGame()	Pause game loop.
+ void continueGame()	Continue game loop.
+ void stopGame()	Stop game loop.
- void generateMap()	Generate current stage's map.
- void normalStage()	Generate obstacles, items and monsters.
- boolean isEmpty()	Check if this space empty for placing new items.
- void showWarning()	Move all monsters out. Show warning before enter to boss stage and play warning siren sound.
- void bossStage()	Initialize boss and create boss stage transition.
+ void victory()	Called when you passed the boss stage. Then go to next stage.
- void moveAll()	Move all movable entities.
- void checkAllCollision()	Check collision between player and other entities which has hitbox, and between monsters and skills.
- void checkPairCollision(ArrayList<? extends Entity> first, ArrayList<? extends Entity> second)	Pair all members between 2 lists, and check if it collision.
- void removeAllDead()	Remove if any entities die.
- void updatePlayer()	Decrease player's cooldown, add player's mana, and add distance.
- void drawHitbox()	Draw all entities which has hitbox, for debugging only.
+ boolean isBossReady()	Check if the boss is ready.
+ Getter & Setter methods	

10 Package input

10.1 Class GameHandler

10.1.1 Field

- HashSet<KeyCode> keys = new HashSet<>()	Set of active keys.
- State saveState	Player's state save.
- Player player	A player.

10.1.2 Method

+ void keyPressed(KeyEvent event)	Recieve input from any pressed key in game scene. See more in Control section.
+ void keyReleased(KeyEvent event)	Recieve input from any released key in game scene. See more in Control section.
+ void keyHeld()	Recieve input from any held key in game scene. See more in Control section.
+ Getter methods	

10.2 Class MainMenuHandler

10.2.1 Method

+ void keyPressed(KeyEvent event)	Recieve input any key is pressed in main menu. See more in Gameplay section.
+ void mouseMoved(MouseEvent event, int idx)	Recieve mouse cursor event in main menu. See more in Gameplay section.
+ void mouseClicked(MouseEvent event)	Recieve input from clicked mouse in main menu. See more in Gameplay section.
+ registerKeyPressed(KeyEvent event)	Recieve input from any pressed key in register scene. See more in Gameplay section.
+ void selectMode(int idx, Image image)	Select the idx mode and change that mode's image to reference image.
+ void modeKeyPressed(KeyEvent event)	Recieve input from pressed key in mode selection scene. See more in Gameplay section.
+ void chooseMode(double difficulty)	Set game's difficulty, then go to game.

10.3 Class ScoreViewHandler

10.3.1 Method

void keyPressed(KeyEvent event)	Recieve input any key is pressed in scoreview scene. See more in Gameplay section.
---------------------------------	---

11 Package scene

11.1 Class GameMain

11.1.1 Field

- Pane gamePane	Game's pane.
- Updater updater	Game's logic loop.
- Animations animations	Game's animation loop.
- double difficulty	Game's difficulty.
- double speed	Game's speed.
- double obstacleSpace	Game's obstacle space.
- double itemSpace	Game's item space.
- double monsterSpace	Game's monster space.
- double stageDistance	Current distance in this stage.

11.1.2 Method

+ void initialize()	Initialize game's pane and model, then set player.
+ Pane getGamePane()	Getter of gamePane.
+ void startGame()	Start game loop and animation loop.
+ void pauseOrResumeGame()	Pause or resume game.
+ void pauseGame()	Pause game loop and animation loop.
+ void continueGame()	Continue game loop, animation loop.
+ void stopGame()	Stop game loop, animation loop.
+ Getter & Setter methods	

11.2 Class LoadingScreen

11.2.1 Field

- Thread progressThread	Progress thread.
- Thread loadingThread	Loading thread.
- Stage window	Game window.
- Scene scene	Game scene.
- Pane pane	Loading screen's pane.

11.2.2 Method

+ void load(Stage stage)	Initialize pane, scene and show window. Then runProgress and loadResource.
+ void runProgress()	Initialize loading progress bar and progress thread. Then start progress thread.
+ void loadResource()	Initialize loading thread, then start loading thread.

11.3 Class MainMenu

11.3.1 Field

<u>- GUI[] menus</u>	Initialize all menus.
<u>- AudioClip selectfx</u>	Sound effect when select menu.
<u>- AudioClip movefx</u>	Sound effect when change selected menu.
<u>- Pane mainMenuPane</u>	Main menu pane.
<u>- int selectedMenu</u>	Current selected menu.
<u>- boolean isTransitioning</u>	The flag indicate that main menu is transitioning.

11.3.2 Method

<u>+ void initialize()</u>	Initialize main menu pane, and play main menu BGM.
<u>+ Pane getMainMenu()</u>	Getter of mainMenuPane.
<u>+ void moveDown()</u>	Move current selected menu down.
<u>+ void moveUp()</u>	Move current selected menu up.
<u>+ void selectMenu()</u>	Play SFX, and draw selected menu's background.
<u>+ void chooseEffect()</u>	Create and play menu choosing effect.
<u>+ void choose()</u>	Go to current selected menu scene.
<u>- void register()</u>	Called when choosing start menu, initialize register scene and show transitions.
<u>+ void addRegisterName(KeyEvent event)</u>	Check exception of the register input. Throw exception if it is invalid. If it is valid, then add character input to the name's text.
<u>+ void deleteRegisterName()</u>	Called when user press BACKSPACE. Check exception, then remove character from name's text if it is valid.
<u>+ void confirmName()</u>	Confirm register name if it valid, throw exception otherwise. Play choosed sound effect.
<u>+ void selectMode()</u>	Initialize mode selection scene. Create and show transitions.
<u>+ void chooseModeEffect(int idx, double difficulty)</u>	Create choosed mode effect at the idx mode menu. Then choose game difficulty.
<u>- void chooseMode(double difficulty)</u>	Play choosed SFX, then pass difficulty to chooseMode method in MainMenuHandler.
<u>+ void playMovefx()</u>	Play move SFX.
<u>+ void playSelectfx()</u>	Play select SFX.
<u>+ Getter & Setter methods</u>	

11.4 Class Rankings

11.4.1 Field

- Font FONT	Load “SukhumvitSet” font.
- Canvas canvas	Ranking’s canvas.
- Pane rankingsPane	Ranking’s pane.
- List<Score> data	Score data’s list.

11.4.2 Method

+ void initialize()	Initialize rankings scene.
+ Pane getRankingsPane()	Getter of rankingsPane.

11.5 Class SceneManager

11.5.1 Field

+ int SCREEN_WIDTH	Default screen’s width.
+ int SCREEN_HEIGHT	Default screen’s height.
- Map map	Game’s background.
- Stage primaryStage	Game’s window.
- Pane root	Root pane.
- Scene scene	Game scene.
- Pane oldPane	Old pane in the scene.
- Pane newPane	New pane to replace the old pane.
- boolean isTransitioning	The flag indicate that game is transitioning.

11.5.2 Method

+ void initialize(Stage stage)	Initialize oldPane, root, scene and set scene to primaryStage.
+ void gotoMainMenu()	Initialize main menu pane, change handler to the MainMenuHandler, then go to main menu.
+ void gotoGame()	Initialize game pane, change handler to the GameHandler, then go to game.
+ void gotoRankings()	Initialize rankings pane, change handler to the ScoreViewHandler, then go to rankings scene.
+ void gotoScoreView()	Initialize score view pane, change handler to the ScoreViewHandler, then go to score view scene.
+ void gotoSceneOf(Pane pane, double speed)	Create transition from old pane to new pane. Set speed of the new pane. Then play the transition.
+ void setTransitioning(boolean isTransitioning)	Force all keys releasing, change player’s power state. Then set isTransitioning.
+ Getter & Setter methods	

11.6 Class ScoreView

11.6.1 Field

- Font FONT	Load "SukhumvitSet" font.
- Canvas canvas	Score view's canvas.
- Pane scoreViewPane	Score view's pane.
- List<Score> data	Score data's list.
- String playerName	Player's name.
- double playerScore	Player's score.
- double playerMode	Player's mode.

11.6.2 Method

+ void initialize()	Initialize score view scene.
+ Getter & Setter methods	

12 Package utility¹

12.1 Class DeleteEmptyInputException extends Exception

12.1.1 Method

+ String getMessage()	Return "No characters to be deleted".
-----------------------	---------------------------------------

12.2 Class LongInputException extends Exception

12.2.1 Method

+ String getMessage()	Return "At most 12 characters".
-----------------------	---------------------------------

12.3 Class ResourceManager

12.3.1 Method

+ void save(Serializable data, String fileName)	Save data to file name.
+ Object load(String fileName)	Load data from file name.

12.4 Class ShortInputException extends Exception

12.4.1 Method

+ String getMessage()	Return "At least 2 characters".
-----------------------	---------------------------------

12.5 Class WrongInputException extends Exception

12.5.1 Method

+ String getMessage()	Return "Allowed characters: a - z, A - Z".
-----------------------	--

12.6 style.css

UML Diagram

