

Permpani Documentation

Created by

Permrawee Tahtniyom 6331333621

Phurin Taengsriwan 6331337121

2110215 Programming Methodology

Semester 2 Year 2020

Chulalongkorn University

Dungeon Survival

1.Overview

Introduction

Dungeon Survival is a 2D shooting game. The objective is to survive as long as you can. Many monsters will try to kill you!

How to play

- Use WASD to move.
- Click to shoot.
- The more time passed the more dangerous monsters are.
- Defeat the boss to increase your stats.
- You can click Esc to pause the game.

Instruction



The stage, time and player health are shown at the bottom of the screen.



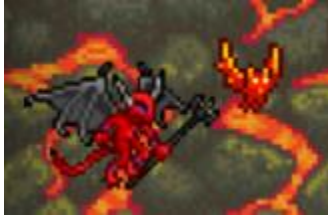
When monster is spawning, it will be a white fade on their body.



A player will be immune for a while when taken damage



Some monster can shoot.

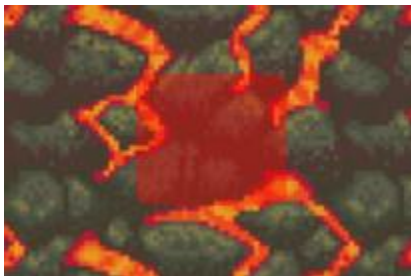


Some monster can summon underlings.



Some monster can dash.

When the monster is going to dash, the dash path will appear.



There will be a red zone to tell that the boss is going to appear there.

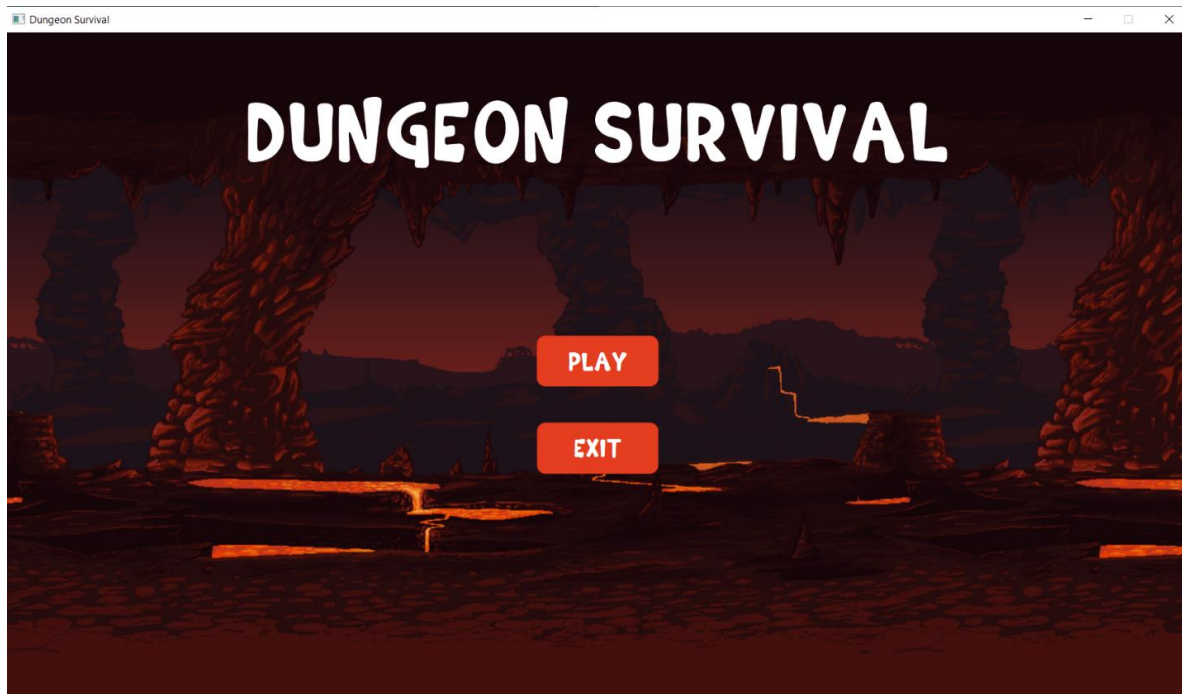


There will be a health bar of the boss at the top of the screen.



You can choose 3 buff after killing a boss.

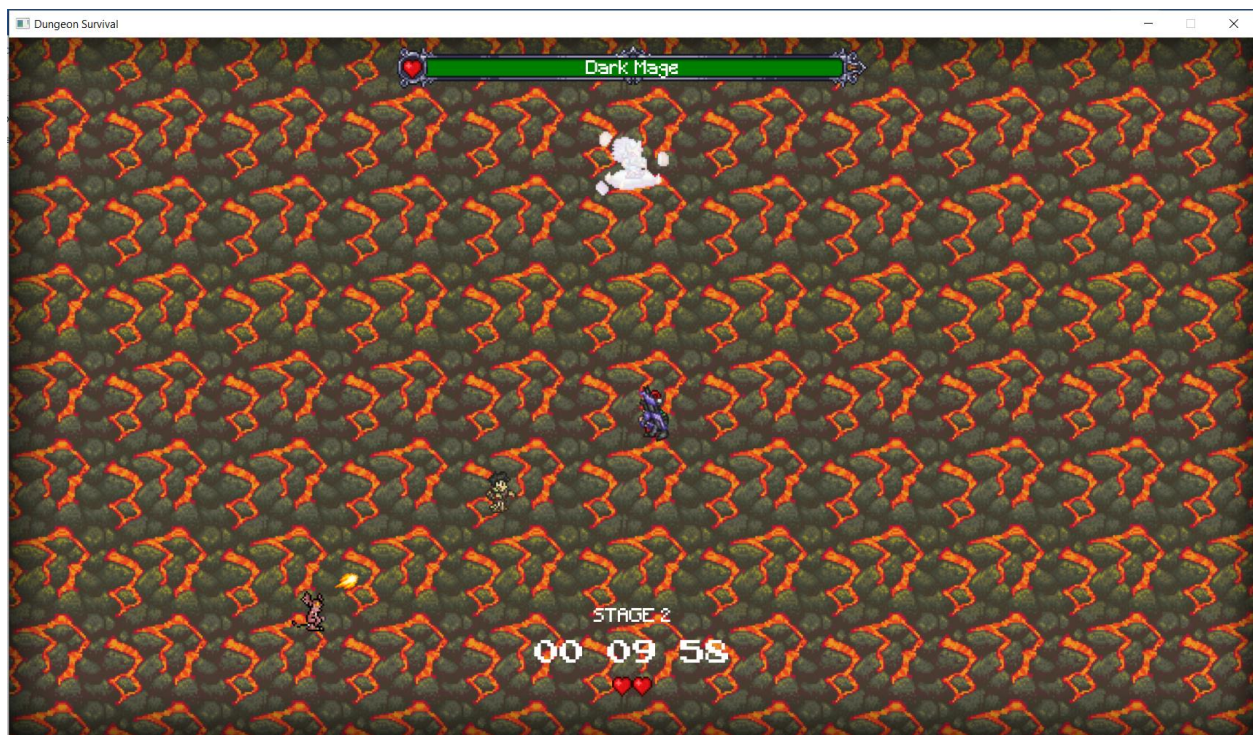
Main Menu Scene



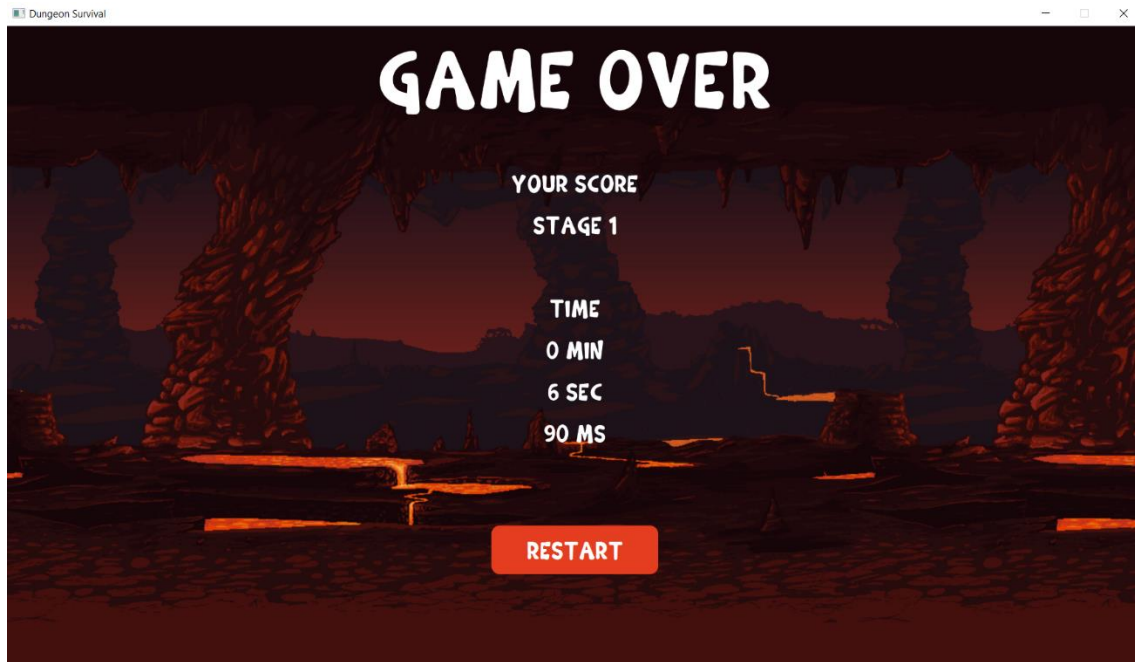
Click Play to start the game or click exit to exit the game.



Select the weapon you want to use for fighting.

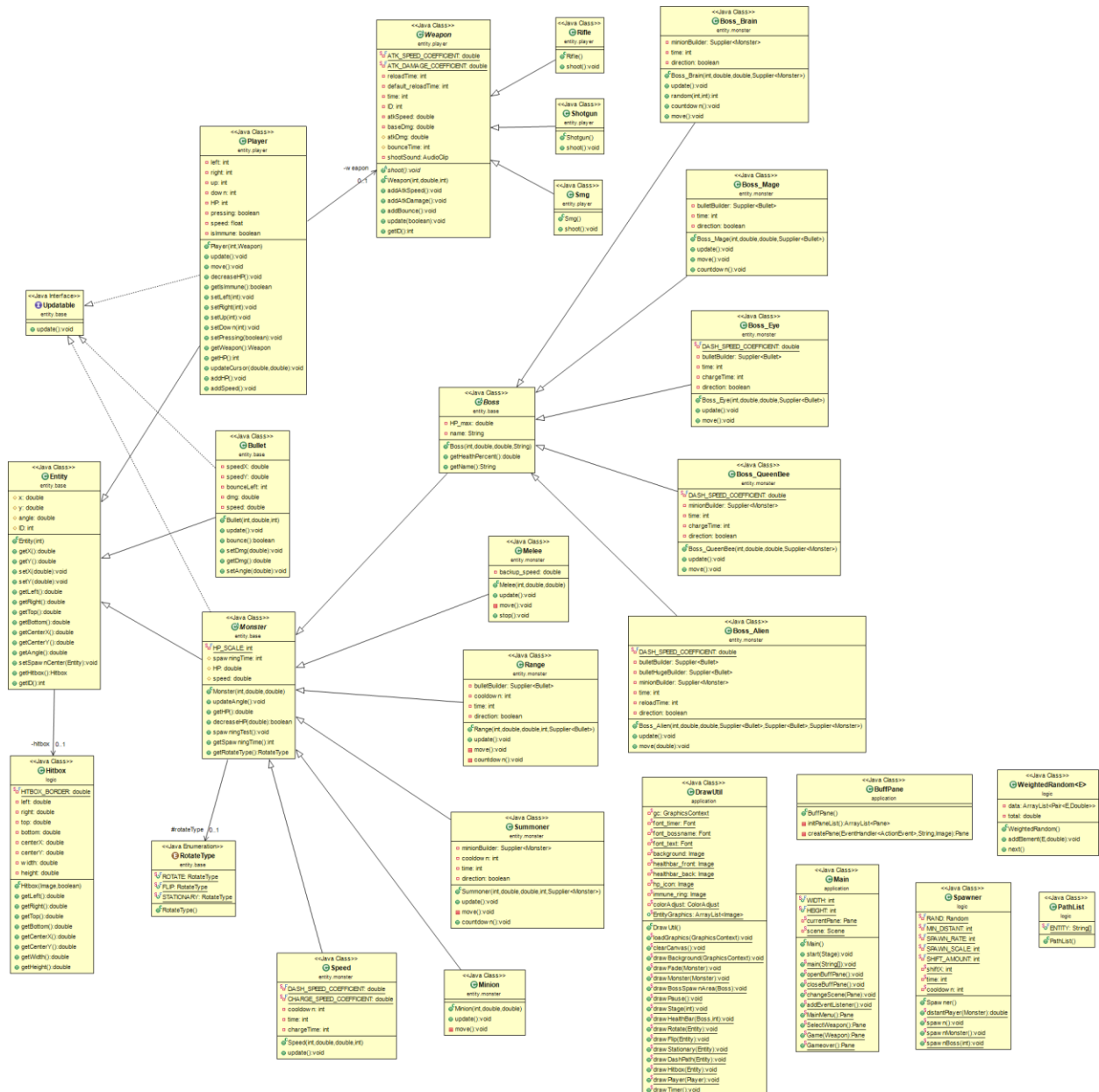


Have fun!



The game is over when you died.

Class Diagram



2. Implementation Details:

2.1 package logic

2.1.1 Class EntityData

2.1.1.1 Fields

<ul style="list-style-type: none">+ <u>final Supplier<Bullet> BULLET 1</u>+ <u>final Supplier<Bullet> BULLET 2</u>+ <u>final Supplier<Bullet> BULLET 3</u>+ <u>final Supplier<Bullet> BULLET LASER</u>+ <u>final Supplier<Bullet> BULLET MAGE</u>+ <u>final Supplier<Bullet> BULLET ALIEN</u>+ <u>final Supplier<Bullet> BULLET ALIEN-HUGE</u>+ <u>final Supplier<Monster> MELEE LV1</u>+ <u>final Supplier<Monster> MELEE LV2</u>+ <u>final Supplier<Monster> MELEE LV3</u>+ <u>final Supplier<Monster> SPEED LV1</u>+ <u>final Supplier<Monster> SPEED LV2</u>+ <u>final Supplier<Monster> SPEED LV3</u>+ <u>final Supplier<Monster> RANGE LV1</u>+ <u>final Supplier<Monster> RANGE LV2</u>+ <u>final Supplier<Monster> RANGE LV3</u>+ <u>final Supplier<Monster> RANGE ALIEN</u>+ <u>final Supplier<Monster> MINION LV1</u>+ <u>final Supplier<Monster> MINION LV2</u>+ <u>final Supplier<Monster> MINION LV3</u>+ <u>final Supplier<Monster> MINION BRAIN</u>+ <u>final Supplier<Monster> MINION QUEENBEE</u>+ <u>final Supplier<Monster> SUMMONER LV1</u>+ <u>final Supplier<Monster> SUMMONER LV2</u>+ <u>final Supplier<Monster> SUMMONER LV3</u>+ <u>final Supplier<Boss> BOSS BRAIN</u>+ <u>final Supplier<Boss> BOSS MAGE</u>+ <u>final Supplier<Boss> BOSS EYE</u>+ <u>final Supplier<Boss> BOSS QUEENBEE</u>+ <u>final Supplier<Boss> BOSS ALIEN</u>	Store constructor for each type of monsters and bullets for easy to use later.
--	--

+ <u>final ArrayList<Supplier<Boss>> BOSS</u> + <u>final ArrayList<Supplier<Monster>> LV1</u> + <u>final ArrayList<Supplier<Monster>> LV2</u> + <u>final ArrayList<Supplier<Monster>> LV3</u> + <u>final ArrayList<ArrayList<Supplier<Monster>>> MONSTER</u>	ArrayList of monsters and bosses for use to randomly spawn later.
--	---

2.1.2 Class GameController

This class is the game system. Most of the game's global variable are kept here.

2.1.2.1 Fields

+ <u>ArrayList<Hitbox> hitboxes</u>	To store hitboxes of each entity ID.
- <u>Player player</u>	To store player of the game.
- <u>ArrayList<Monster> monsters</u>	To store monsters in the game.
- <u>ArrayList<Boss> bosses</u>	To store bosses in the game.
- <u>ArrayList<Bullet> bullets_enemy</u>	To store enemy's bullet in the game.
- <u>ArrayList<Bullet> bullets_player</u>	To store player's bullet in the game.
- <u>Boss spawningBoss</u>	To store boss that is spawning.
- <u>int stage</u>	To count the stage that player is in.
- <u>long elapsedTime</u>	To count the time from start game (1 in this field mean 10 ms).
- <u>final int STAGE_LENGTH</u>	To tell how long one stage is.
- <u>AnimationTimer MainProcess</u>	A thread that runs every frame. Use to draw and update logic of the game.
- <u>Timer timer</u>	A timer.

- <u>boolean is_pause</u>	To check if the game is pausing or not.
- <u>AudioClip bossDeath</u>	The effect sound when boss dead.
- <u>AudioClip backgroundMusic</u>	The music of the game.

2.1.2.2 Methods

+ <u>void initGame(GraphicContext gc)</u>	To initialize fields. Start Main Process, background music and timer.
- <u>boolean isCollide(Entity e1, Entity e2)</u>	To check if 2 entities is collide (hitboxes are intercepted) each other or not.
+ <u>boolean HitBorder(Entity e)</u>	To check if part of entity hitbox is out of scene or not.
+ <u>boolean OutOfScene(Entity e)</u>	To check if the entity center is out of scene or not.
- <u>void drawScene()</u>	To draw every entity on to canvas.
+ <u>void update()</u>	To update value of every entity and spawn monsters and boss.
+ <u>void updateMonsters()</u>	Update value of monsters in ArrayList monsters. If it collide with player, run player.decreaseHP, if it is a melee type, it will stop for a while and if it is a minion type, destroy it.
+ <u>void updateBosses()</u>	Update value of bosses in ArrayList bosses. If its collide with player, run player.decreaseHP.
+ <u>void updateBullet_enemy()</u>	Update value of bullets in ArrayList bullets_enemy. If it hits border and can't bounce, remove it or else if it collide with player, run

	player.decreaseHP and remove it.
+ <u>void updateBullet_player()</u>	Update value of bullets in ArrayList bullets_player. If it hits border and can't bounce, remove it. If it collides with bosses or monsters decrease its HP equal to bullet damage and remove this bullet.
- <u>void initTimer()</u>	To start counting time.
+ <u>void setSpawningBoss(Boss boss)</u>	Setter for spawningBoss field.
+ <u>Player SetPlayer(Player player)</u> + <u>Player getPlayer()</u>	Getter and Setter for player field.
+ <u>int getStage()</u>	Getter for stage field.
+ <u>long getElapsedTime()</u>	Getter for elapsedTime field.
+ <u>void addMonster(Monster mon)</u>	Add a monster to the ArrayList monsters.
+ <u>void addBosses(Boss boss)</u>	Add a boss to the ArrayList bosses.
+ <u>void addBullet_enemy(Bullett bullet)</u>	Add a bullet to the ArrayList bullets_enemy.
+ <u>void addBullet_player(Bullett bullet)</u>	Add a bullet to the ArrayList bullets_player.
+ <u>void switchState()</u>	Check if game is pausing, continue the game or else pause the game.
+ <u>void pauseGame()</u>	Pause game process.
+ <u>void continueGame()</u>	Continue game process.
+ <u>void gameEnd()</u>	Stop game process and change screen to Game over.

2.1.3 Class Hitbox

2.1.3.1 Fields

- <u>final double HITBOX_BORDER</u>	Determine how much hit box will smaller than picture.
- double left	Get the x-coordinate of the left side of the hitbox compare to left side of image.
- double right	Get the x-coordinate of the right side of the hitbox compare to left side of image.
- double top	Get the y-coordinate of the upper side of the hitbox compare to upper side of image.
- double bottom	Get the y-coordinate of the lower side of the hitbox compare to upper side of image.
- double centerX	Get the x-coordinate of the center of the hitbox compare to left side of image.
- double centerY	Get the y-coordinate of the center of the hitbox compare to upper side of image.
- double width	Width of the image.
- double height	Height of the image.

2.1.3.2 Constructor

+ Hitbox(Image img, boolean is_bullet)	Initialize fields. If is_bullet is true, set width and height to minimum value of those or else initialize normally.
--	---

2.1.3.3 Method

+ double getLeft() + double getRight() + double getTop() + double getBottom() + double getCenterX() + double getCenterY() + double getWidth() + double getHeight ()	Getter for each field.
--	------------------------

2.1.4 Class PathList

2.1.4.1 Field

+ <u>final String[] ENTITY</u>	An Array of string that contain all of entity image path.
--------------------------------	---

2.1.5 Class Spawner

2.1.5.1 Fields

- <u>final Random RAND</u>	Randomer.
- <u>final int MIN DISTANT</u>	Minimum distant that monster must spawn far from player.
- <u>int SPAWN RATE</u>	Spawn rate of monsters.
- <u>final int SPAWN SCALE</u>	How fast spawn rate will increase.
- <u>final int SHIFT AMOUNT</u>	Amount of spawn rate that will decrease when boss spawn.
- <u>int shiftX</u>	Accumulate SHIFT_AMOUNT.

- <u>int time</u>	Counter for loop spawn monsters.
- <u>int cooldown</u>	Cooldown time for spawn monsters.

2.1.5.2 Methods

+ <u>double distantPlayer(Monster mon)</u>	Calculate distant between player and [mon] power by 2.
+ <u>void spawn()</u>	Loop for spawn monsters.
+ <u>void spawnMonster()</u>	Randomly spawn a monster from EntityData. The more time passed, the more high level monster will spawn.
+ <u>void spawnBoss(int stage)</u>	Create a boss and put it in GameController.spawningBoss. After a while, add it in ArrayList bosses. Shift spawn rate.

2.1.6 Class WeightedRandom<E>

2.1.6.1 Fields

- ArrayList<Pair<E, Double>> data	An ArrayList contains elements.
- double total	Total weight of elements.

2.1.6.2 Constructor

+ WeightedRandom()	Initialize field.
--------------------	-------------------

2.1.6.3 Methods

+ void addElement(E e, double weight)	Add an element (Pair of E and weight) to data. Add weight to total.
+ E next()	Weighted random element from data.

2.2 package `entity.base`

2.2.1 Class Boss extends Monster

2.2.1.1 Fields

- double HP_max	Max health point of the boss
- String name	Name of the boss

2.2.1.2 Constructor

+ Boss(int ID, double HP, double speed, String name)	- Initialize fields. - Set x-coordinate of the boss to center of the screen.
--	---

2.2.1.3 Methods

+ double getHealthPercent()	Return boss health point in term of percent of its max health point.
+ String getName	Get the name of the boss.

2.2.2 Class Bullet extends Entity implements Updatable

2.2.2.1 Fields

- double speedX	Speed of the bullet in x-axis.
- double speedY	Speed of the bullet in y-axis.
- int bounceLeft	How many bounce time left.
- double dmg	Damage of the bullet.
- double speed	Total speed of the bullet.

2.2.2.2 Constructor

+ Bullet(int ID, double speed, int bounceLeft)	- Initialize fields
--	---------------------

2.2.2.3 Methods

+ void update()	Update the position of the bullet(add position by speed).
+ boolean bounce()	Check the bounce time left and return boolean that tell if this bullet can bounce or not.
+ double getDmg() + void setDmg(double dmg)	- Get the damage of the bullet. - Set the damage of the bullet.
+ void setAngle(double angle)	Setter for the field angle and reassign speedX and speedY .

2.2.3 Class Entity

2.2.3.1 Fields

# double x	x-coordinate of this entity
# double y	y-coordinate of this entity
# double angle	Angle of entity which tell how much is this entity rotate and use to calculate speed and draw.
# int ID	The number that defines this entity appearance
- Hitbox hitbox	Rectangle that uses to calculate if two entity collided each other.

2.2.3.2 Constuctor

+ Entity(int ID)	Initialize fields.
------------------	--------------------

2.2.3.3 Methods

+ double getX() + void setX()	- Get the x-coordinate of the entity. - Set the x-coordinate of the entity.
+ double getY() + void setY()	- Get the y-coordinate of the entity. - Set the y-coordinate of the entity.
+ double getLeft()	Get the x-coordinate of the left side of the hitbox.
+ double getRight()	Get the x-coordinate of the right side of the hitbox.
+ double getTop()	Get the y-coordinate of the top of the hitbox.
+ double getBottom()	Get the y-coordinate of the bottom of the hitbox.

+ double getCenterX()	Get the x-coordinate of the center of the hitbox.
+ double getCenterY()	Get the y-coordinate of the center of the hitbox.
+ double getAngle()	Get angle.
+ void setSpawnCenter(Entity e)	Calculate and set position that this entity will spawn in the middle of another entity[Entity e].
+ Hitbox getHitBox()	Get the hitbox of this entity.
+ int getID()	Get the monster's ID.

2.2.4 Class Monster extends Entity implements Updatable

2.2.4.1 Fields

- <u>final int HP_SCALE</u>	Determine how much [HP] will increase when time passed.
# int spawningTime	How long this monster delay to update after spawn.
# double HP	Health point of this monster.
# double speed	Speed of this monster.
# RotateType rotateType	Type of the monster (ROTATE, FLIP, STATIONARY)

2.2.4.2 Constuctor

+ Monster(int ID, double HP, double speed)	Initialize fields. Set [spawningTime] = 60 . Set [HP] .
--	---

2.2.4.3 Methods

+ void updateAngle()	This method will update angle between player and this monster and store in angle field.
+ double getHP()	Get health point of this monster.
+ boolean decreaseHP(double dmg)	Monster's health point decrease for the specific damage.
+ void spawningTest()	Check that this monster is in spawning state or not.
+ int getSpawningTime()	Get spawningTime field.
+ RotateType getRotateType()	Get the monster's rotateType.

2.2.5 Enum RotateType

2.2.5.1 enum

ROTATE, FLIP, STATIONARY

2.2.6 Interface Updatable

2.2.6.1 Method

+ void update()	Ability to update when game looped.
-----------------	-------------------------------------

2.3 package `entity.monster`

2.3.1 Class `Boss_Alien` extends `Boss`

2.3.1.1 Fields

- <u>final double DASH_SPEED_COEFFICIENT</u>	Coefficient of speed when this boss dash.
--	---

- Supplier<Bullet> bulletBuilder	Contain boss' Bullet.
- Supplier<Monster> minionBuilder	Contain boss' underling.
- int time	Counter for loop of this boss activity.
- int reloadTime	Counter for loop of this boss activity.
- boolean direction	Direction of movement (left, right)

2.3.1.2 Construcuter

+ Boss_Alien(int ID, double HP, double speed, Supplier<Bullet> bulletBuilder)	Initialize fields. Set boss name to Martian. Set rotate type to STATIONARY.
---	---

2.3.1.3 Methods

+ void update()	Loop 2 activity. 1. Shoot and move rapidly. 2. Summon underlings. 3. Shoot a huge bullet.
+ void move()	Move forth and back in the x-axis.

2.3.2 Class Boss_Brain extends Boss

2.3.2.1 Fields

- Supplier<Monster> minionBuilder	Contain the boss' underlings.
- int time	Counter for loop of this boss activity.
- boolean direction	Direction of movement (clockwise, anticlockwise)

2.3.2.2 Constructor

+ Boss_Brain(int ID, double HP, double speed, Supplier<Monster> minionBuilder)	Initialize fields. Set boss name to Brain of Cthulhu. Set rotate type to STATIONARY.
--	--

2.3.2.3 Methods

+ void update()	Update angle. Move and summon underlings.
+ int random(int min, int max)	Return random int between the 2 specific value.
+ void countdown()	Loop to randomly summon underlings.
+ void move()	Move circular around player.

2.3.3 Class Boss_Eye extends Boss

2.3.3.1 Fields

- <u>final double DASH_SPEED_COEFFICIENT</u>	Coefficient of speed when this boss dash.
- Supplier<Bullet> bulletBuilder	Contain the boss' bullet.
- int time	Counter for loop of this boss activity.
- int chargeTime	Counter for loop of this boss activity.
- boolean direction	Direction of movement (clockwise, anticlockwise)

2.3.3.2 Constuctor

+ Boss_Eye (int ID, double HP, double speed, Supplier<Bullet> bulletBuilder)	Initialize fields. Set boss name to Eye of Cthulhu. Set rotate type to ROTATE.
--	--

2.3.3.3 Methods

+ void update()	Loop 2 activities. 1.Dash toward player until GameController.HitBorder is true. 2.Shoot bullets rapidly.
+ void move()	- Move toward player when GameController.OutOfScene is true. - Move circular around player when GameController.OutOfScene is false.

2.3.4 Class Boss_Mage extends Boss

2.3.4.1 Fields

- Supplier<Bullet> bulletBuilder	Contain the boss' bullet .
- int time	Counter for loop of this boss activity.
- boolean direction	Direction of movement (clockwise, anticlockwise).

2.3.4.2 Constuctor

+ Boss_Mage (int ID, double HP, double speed, Supplier<Bullet> bulletBuilder)	Initialize fields. Set boss name to Dark Mage. Set rotate type to FLIP.
---	---

2.3.4.3 Methods

+ void update()	Update angle. Move and shoot bullets in 4 and 6 directions.
+ void move()	Move circular around player.

+ void countdown()	Loop for shoot bullets in 4 and 6 directions.
--------------------	---

2.3.5 Class Boss_QueenBee extends Boss

2.3.5.1 Fields

- <u>final double DASH_SPEED_COEFFICIENT</u>	Coefficient of speed when this boss dash.
- Supplier<Monster> minionBuilder	Contain this boss' underlings.
- int time	Counter for loop of this boss activity.
- int chargeTime	Counter for loop of this boss activity.
- boolean direction	Direction of movement (clockwise, anticlockwise).

2.3.5.1 Constuctor

+ Boss_QueenBee(int ID, double HP, double speed, Supplier<Monster> minionBuilder)	Initialize fields. Set boss name to Queen Bee. Set rotate type to FLIP.
---	---

2.3.5.2 Methods

+ void update()	Loop 2 activities. 1.Dash toward player until GameController.HitBorder is true. 2. Summon underlings.
+ void move()	- Move toward player when GameController.OutOfScene is true. - Move circular around player when GameController.OutOfScene is false.

2.3.6 Class Melee extends Monster

2.3.6.1 Fields

- double backup_speed	Use to back up speed.
-----------------------	-----------------------

2.3.6.2 Constructor

+ Melee(int ID, double HP, double speed)	Initialize fields. Set rotate type to FLIP.
--	--

2.3.6.3 Methods

+ void update()	Update angle. Move.
+ void move()	Move toward player.
+ void stop()	- Monster stop for a while when collide with the player.

2.3.7 Class Minion extends Monster

2.3.7.1 Constructor

+ Minion(int ID, double HP, double speed)	Initialize fields. Set rotate type to FLIP.
---	--

2.3.7.2 Methods

+ void update()	Update angle. Move.
+ void move()	Move toward player.

2.3.8 Class Range extends Monster

2.3.8.1 Fields

- Supplier<Bullet> bulletBuilder	Contain this monster's bullet.
- int cooldown	Cooldown time for this monster to shoot a bullet.
- int time	Counter for loop of this monster activity.
- boolean direction	Direction of movement (clockwise, anticlockwise).

2.3.8.2 Constuctor

+ Range(int ID, double HP, double speed, int cooldown, Supplier<Bullet> bulletBuilder)	Initialize fields. Set rotate type to FLIP.
--	--

2.3.8.3 Methods

+ void update()	Update angle. Move and shoot a bullet.
+ void move()	Move circular around player.
+ void countdown()	Loop for shoot a bullet.

2.3.9 Class Speed extends Monster

2.3.9.1 Fields

- <u>final double DASH_SPEED_COEFFICIENT</u>	Coefficient of speed when this monster dash.
- <u>final double CHARGE_SPEED_COEFFICIENT</u>	Coefficient of charge speed.
- int cooldown	Cooldown time for the monster to do activity.

- int time	Counter for loop of this monster activity.
- int chargeTime	Counter for loop of this monster activity.

2.3.9.2 Constuctor

+ Speed(int ID, double HP, double speed, int cooldown)	Initialize fields. Set rotate type to FLIP.
--	--

2.3.9.3 Method

+ void update()	Loop for charge and dash.
-----------------	---------------------------

2.3.10 Class Summoner extends Monster

2.3.10.1 Fields

- Supplier<Monster> minionBuilder	Contain the monster's underling.
- int cooldown	Cooldown time for the monster to summon a minion
- int time	Counter for loop of this monster activity.
- boolean direction	Direction of movement (clockwise, anticlockwise).

2.3.10.2 Constuctor

+ Summoner(int ID, double HP, double speed, int cooldown, Supplier<Monster> minionBuilder)	Initialize fields. Set rotate type to FLIP.
--	--

2.3.10.3 Methods

+ void update()	Update angle. Move and summon an underling.
+ void move()	Move circular around player.
+ void countdown()	Loop for summon an underling.

2.4 package `entity.player`

2.4.1 Class Player extends Entity implements Updatable

2.4.1.1 Fields

- int left	Store left button pressing state.
- int right	Store right button pressing state.
- int up	Store up button pressing state.
- int down	Store down button pressing state.
- int HP	Health point of this player.
- Weapon weapon	Weapon of this player.
- boolean pressing	Store mouse pressing state.
- float speed	Speed of this player.
- boolean isImmune	Store immune state.

2.4.1.2 Constructor

+ Player(int ID, Weapon weapon)	Initialize fields. Set player position to middle of the screen.
---------------------------------	--

2.4.1.3 Method

+ void update()	Move and update weapon.
+ void move()	Move in 8 directions up to player.
+ void decreaseHP()	If not immune, decrease player HP and immune for a while then if HP<=0 run GameController.gameEnd().
+ boolean getIsImmune() + Weapon getWeapon() + int getHP()	Getter for isImmune, weapon, HP field.
+ void setLeft(int left) + void setRight(int right) + void setUp(int up) + void setDown(int down) + void setPressing(boolean pressing)	Setters for left, right, up, down, pressing fields.
+ void updateCursor(double cursorX, double cursorY)	Update angle between cursor and player.
+ void addHP()	Increase HP.
+ void addSpeed()	Increase speed.

2.4.2 Class Rifle extends Weapon

2.4.2.1 Constructor

+ Rifle()	Initialize fields.
-----------	--------------------

2.4.2.2 Method

+ void shoot()	Shoot bullet in 1 direction.
----------------	------------------------------

2.4.3 Class Shotgun extends Weapon

2.4.3.1 Constuctor

+ Shotgun()	Initialize fields
-------------	-------------------

2.4.3.2 Method

+ void shoot()	Shoot bullet in 3 direction.
----------------	------------------------------

2.4.4 Class Smg extends Weapon

2.4.4.1 Constructor

+ Smg()	Initialize fields
---------	-------------------

2.4.4.2 Method

+ void shoot()	Shoot bullet in 1 direction, but inaccurate (randomly +- pi/18).
----------------	--

2.4.5 Abstract Class Weapon

2.4.5.1 Fields

- <u>final double ATK SPEED COEFFICIENT</u>	Coefficient when addAtkSpeed().
---	---------------------------------

- <u>final double ATK_DAMAGE_COEFFICIENT</u>	Coefficient when addAtkDamage().
- int reloadTime	Time between each bullet shoot.
- int default_reloadTime	Default time between each bullet shoot.
- int time	Counter for loop to shoot.
- int ID	The number that defines this weapon appearance
- double atkSpeed	Store attack speed of this weapon.
# double atkDmg	Store attack damage of this weapon.
- double baseDmg	Store base damage of this weapon.
# int bounceTime	Store bounce time of this weapon.
- AudioClip shootSound	Sound when bullet shot.

2.4.5.2 Constuctor

+ Weapon(int id, double baseDmg, int reloadTime)	Initialize fields.
--	--------------------

2.4.5.3 Method

+ void shoot()	How each gun shoot.
+ void addAtkSpeed()	Increase atkSpeed and reduce reloadTime.
+ void addAtkDamage()	Increase atkDmg.

+ void addBounce()	Increase bounceTime.
+ void update(boolean pressing)	Loop for shoot when pressing is true.
+ int getID()	Getter for ID.

2.5 package application

2.5.1 Class DrawUtil

2.5.1.1 Fields

- <u>GraphicsContext gc</u>	Graphic context of the canvas.
- <u>Font font_timer</u>	Font for timer.
- <u>Font font_bossname</u>	Font for boss name and stage.
- <u>Font font_text</u>	Font for text.
- <u>Image background</u>	Image for background.
- <u>Image healthbar_front</u> - <u>Image healthbar_back</u>	Image for boss health bar.
- <u>Image hp_icon</u>	Image for HP icon.
- <u>Image immune_ring</u>	Image for immune ring.
- <u>ColorAdjust colorAdjust</u>	ColorAdjust for spawning monster.
+ <u>ArrayList<Image> EntityGraphics</u>	ArrayList contain Image for each Entity.

2.5.1.2 Method

<u>+ void loadGraphics(GraphicContext gc temp) throws CantLoadGraphicException</u>	Load and initialize graphic of the game.
<u>+ void clearCanvas()</u>	Clear canvas.
<u>+ void drawBackground(GraphicContext bg)</u>	Draw background.
<u>+ void drawFade(Monster mon)</u>	Draw a spawning monster.
<u>+ void drawMonster(Monster mon)</u>	Draw a monster.
<u>+ void drawBossSpawnArea(Boss boss)</u>	Draw boss spawning area.
<u>+ void drawPause()</u>	Draw pausing screen.
<u>+ void drawStage(int stage)</u>	Draw text to show what stage player in.
<u>+ void drawHealthBar(Boss boss, int count)</u>	Draw health bar of the boss.
<u>+ void drawRotate(Entity e)</u>	Draw an Entity that rotate when change angle.
<u>+ void drawFlip(Entity e)</u>	Draw an Entity that flip when change angle.
<u>+ void drawStationary(Entity e)</u>	Draw an Entity that stationary when change angle.
<u>+ void drawDashPath(Entity e)</u>	Draw dash path of monster that can dash.
<u>+ void drawHitbox(Entity e)</u>	Draw hit box of an Entity for testing.
<u>+ void drawPlayer(Player player)</u>	Draw a player and a weapon.
<u>+ void drawTimer()</u>	Draw a timer.

2.5.2 Class Main extends Application

2.5.2.1 Fields

<u>+ final int WIDTH = 1366</u> <u>+ final int HEIGHT = 768</u>	Size of the screen.
<u>- Pane currentPane</u>	Current pane that is a root node of the scene.
<u>- Scene scene</u>	Scene that show on screen.

2.5.2.2 Method

<u>+ void start(Stage stage)</u>	For JavaFX. Set current root to MainMenu.
<u>+ void main(String[] args)</u>	For JavaFX.
<u>+ void openBuffPane()</u>	Open the buff pane.
<u>+ void closeBuffPane()</u>	Close the buff pane.
<u>+ void changeScene(Pane pane)</u>	Set currentPane to [pane]. Change root of the scene to [pane].
<u>- void addEventListener(Scene scene)</u>	Set buttons action.
<u>+ Pane MainMenu()</u>	Create main menu pane.
<u>+ Pane SelectWeapon()</u>	Create select weapon pane.
<u>+ Pane Game(Weapon weapon)</u>	Create main game pane.
<u>+ Pane Gameover()</u>	Create game over pane.

2.5.3 Class BuffPane extends Pane

2.5.3.1 Constructor

+ BuffPane()	Create buff pane. Random 3 buff to show on screen. Initialize fields.
--------------	---

2.5.3.2 Method

- ArrayList<Pane> initPaneList()	Create an ArrayList contains buff.
- Pane createPane(EventHandler<ActionEvent> event, String text, Image img)	Create buff component.