DEFENSE OF THE BULLETS

Chawit Sakulyuenyong 593 10141 21

Methus Kiatchaiwat 593 10536 21

1. Introduction

Defense of the Bullets: DotB is a strategic, arena game for one player, which will give a subtle feeling of MMO, RPG and MOBA at the same time.

You will be in charge of a powerful, miniature destroyer with a pistol and unlimited ammunitions. As you grow stronger and stronger, towers are not easy to be destroyed. Enemies are waiting at the gate, in the landscape of battles! You and your team are tasked to bring peace by capturing all the towers and cease the madness of chaos. Are you powerful enough to fight this battle? Will you be able to dominate the fate of the world? Let's find out together!

2. How to play

When you first start the game, you will be on menu screen. Before you can play the game, you need to name the character with a string of allowed characters of length 1 to 10. If the name is not valid, error messages will be shown.

Figure 1: Menu screen

Figure 2: Menu screen with error messages

If you want to mute the sound of the game, you can click at the icon at the bottom-left corner to mute the music. Clicking again will unmute the sound.

When you are desired with the name, press [Enter] to go to the side screen, which allows you to select the side you will be playing on. You can choose whatever side you want to play simply by clicking on the icon. If you want to change the name, you can click the arrow icon the go back to edit your name again.

Figure 3: Menu screen after muted

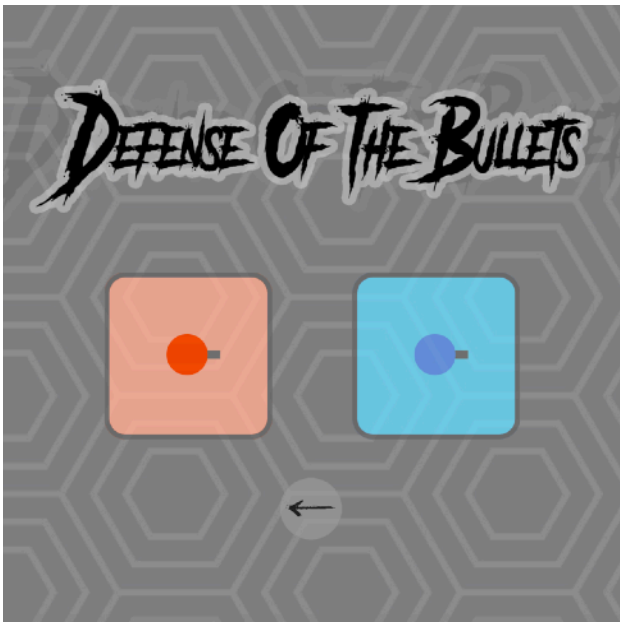


Figure 4: Side screen

On game screen, each side will have 5 players. You will be with 4 bots on your side, playing against 5 bots from another side. You will start with the job “Novice” with level 1.

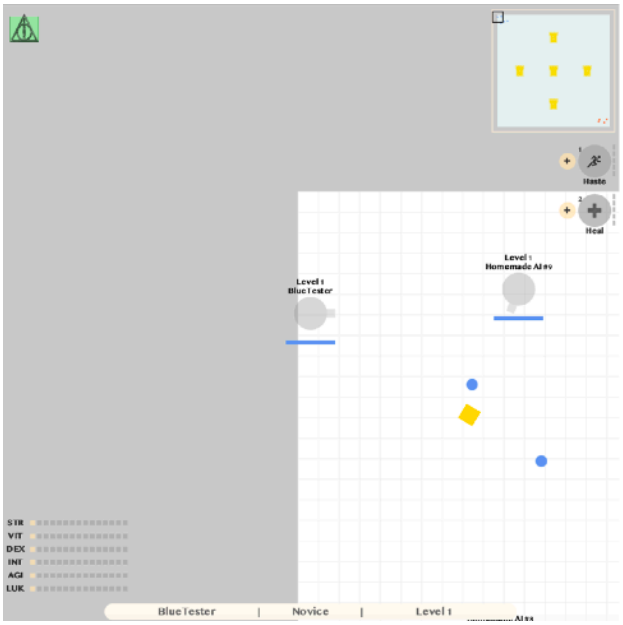


Figure 5: Game screen on Blue side

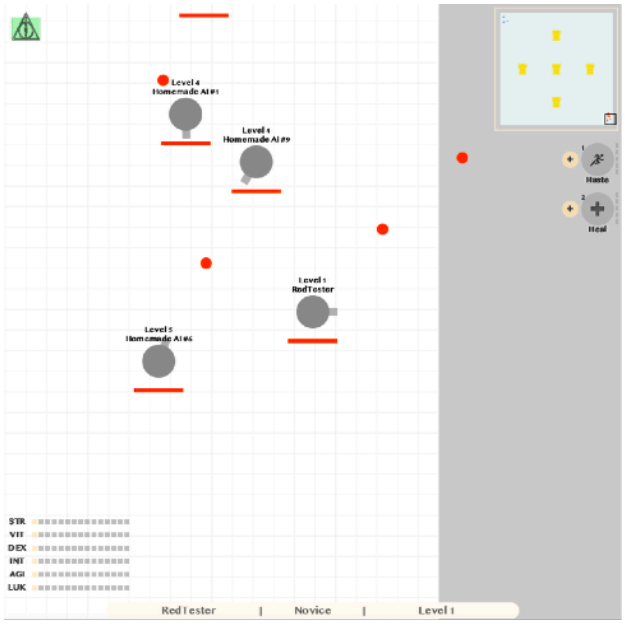


Figure 6: Game screen on Red side

On the top-right corner is mini-map which allows player to see the beyond the screen. Each icon on the mini-map represents other players, towers and the range of view box.

On the right side is skill panel which shows all skills of the player, along with the each skill’s cool-down time and upgrade icon.

On the bottom-left corner is status panel which shows player’s statuses, along with maximum statuses possible and upgrade button.

On the top-left corner is buff and de-buff panel which show the statuses occurring to player, as well as how long the statuses are going to last. Buffs with positive effects are shown in the top row with green background. On the other hand, de-buffs, buffs with negative effects are shown in the second row with red background.

Finally, at the bottom is experience bar which shows current experience point, compared to maximum experience point at that level. The texts above show player’s name, job and level, respectively.

The objective of this game is to conquer all the towers. However, towers are too strong, so you need to upgrade the character by shooting at food first. These are how you can control your character:

- [Mouse Movement] or [Mouse Dragging]: Change direction of the pistol
- [W] or [Up Arrow]: Go up
- [S] or [Down Arrow]: Go down
- [A] or [Left Arrow]: Go left
- [D] or [Right Arrow]: Go right
- [Left Click] or [Spacebar]: Shoot
- [E]: Auto-shoot.

Moreover, to make your character even more powerful, statuses and skills are the key here. We will talk about skills later, but right now, let’s have a look on what these statuses are for:

- STR: Self-attack + Bullet damage
- VIT: Max health point + Lower received damage
- DEX: Bullet health point + Bullet speed
- INT: Bonus damage for special bullets for Mages
- AGI: Move speed + Reload time
- LUK: Critical chance + Critical damage factor.

Statuses and skills can be upgraded directly on their panel, or here are the keyboard shortcuts:

- [Number]: Use skill of position [Number]
- [Shift] + [Number]: Upgrade skill of position [Number]
- [Control] + [Number]: Upgrade status of position [Number].

When you are of level 11, there are 3 more jobs you can choose to change to (if you desire). You can click directly on the top-left icon to change to those jobs. Each and every one of them has unique ability and is designed specifically to be good at different things. Let's take a look on what each job is capable of:

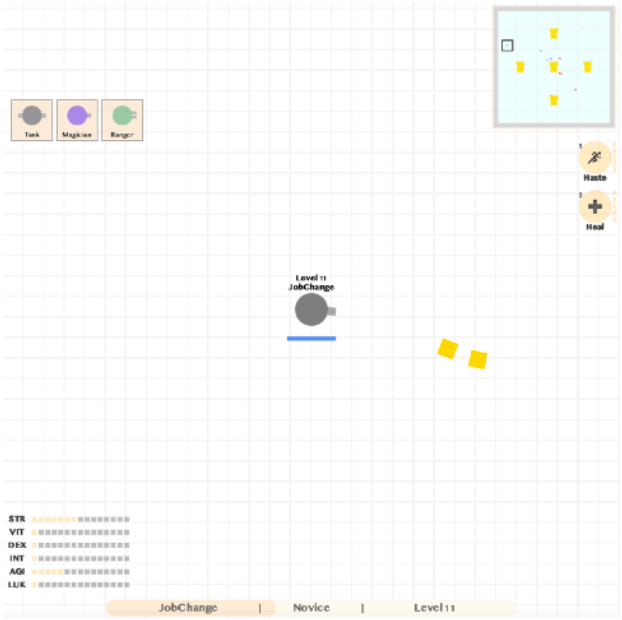


Figure 7: Class change demonstration

Job **Novice**:

Starts of nice and easy as a basic job which is fundamental to every other job.

Skill:

- Haste: Boost move speed for a short amount of time.
- Heal: Increase regeneration rate for some time.

Job **Tank**:

Specializes in being resistance to damage and is particularly difficult to exterminate.

Skill:

- Shield: Add shield to absorb the damage from the enemy.
- Burst: Shoot massive amount of ammunitions around

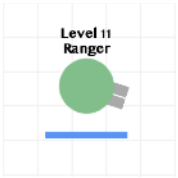
Job **Magician**:

Specialized in magic which can have supernatural effects on target.

Skill:

- Fire Orb: Change normal bullets to fire bullets causing burn on target player.
- Ice Orb: Change normal bullets to ice bullets causing slow on target player.

Job **Ranger**:



Specialized in shooting techniques specifically causing highest damage per second.

Skill:

- Double Attack: Provide with the chance of attacking more than once at a time.
- Frenzy: Increase attack speed dramatically in exchange of lower move speed.

You will need to play and try to conquer all the towers to end the game. However, if you fail to survive in some situations, it is no big deal. Just take a short break for a few seconds and you should be fine! Just know that there is dead penalty which halves the experience points you have gained, so you will end up with lower level every time.

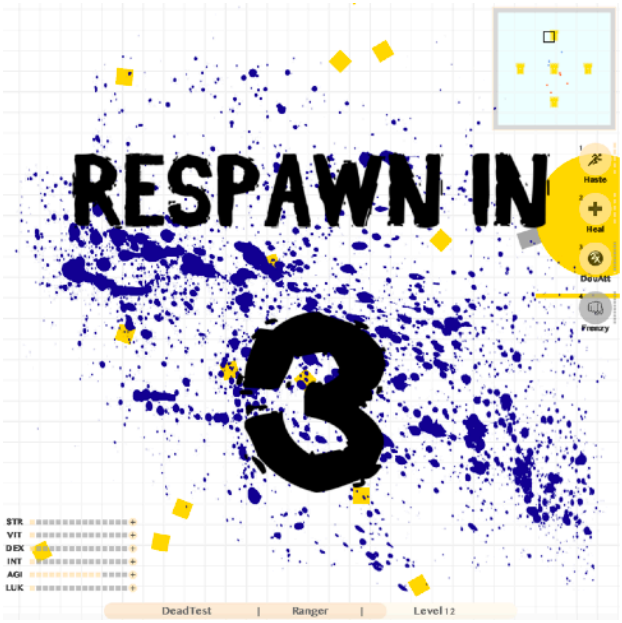


Figure 8: Dead screen

Also note that every time you spawn, you will have invincible buff which will protect you to from any damage for a short amount of time.

Ultimately, if you manage to win or unfortunately lose the game, you will be able to transition to score board screen which show the details of each player at the end of the game.

Figure 9: Victory Screen

Figure 10: Defeat Screen

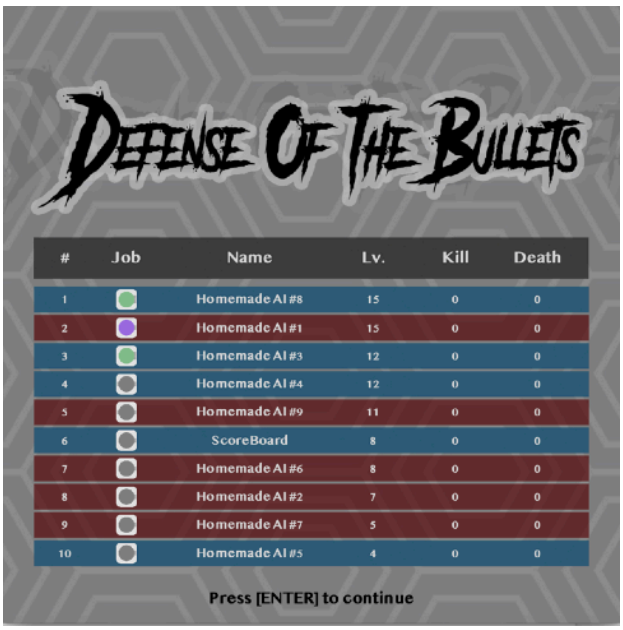


Figure 11: Scoreboard Screen

After that, you will, then, go back to menu screen again. If you wish to play again, you definitely can play again. If you wish to exit the game, just press [ESC] and we will take care everything for you.

3. Implementation

3.1 Package bot

3.1.1 Class *Bot*

3.1.1.1 Field

# double VISION	The Number of pixel bot can view. Default = Main.SCREEN_SIZE/2.0
# int SIZE_OF_GRID	The Number of Grid bot can view in integer. Default = 15
# int NUMBER_OF_CHANGE_POSITION	The Number of opportunity bot change direction. Expected value is 20 seconded to change it. Default = 20 * Main.FRAME_RATE
# int SAFETY_ZONE	An area bot is safe.
# double MOVE_HEURISTIC	Heuristic number for farm in bot Default = 2.6
# ArrayList<Novice> playerFriendList	List of friends bot can view.
# ArrayList<Novice> playerEnemiesList	List of enemies bot can view.
# ArrayList<Bullet> bulletList	List of bullet bot can view.
# ArrayList<Tower> towerList	List of tower bot can view.
# ArrayList<Food> foodList	List of food bot can view.
# int[] change8to4	Change 8 direction to 4 direction 7.0.1 0....1 6...2 to ..... 5.4.3 2....3
# int[] change4to8	Change 4 direction to 8 direction 0....1 7..0.1 ..... to 6....2 2....3 5..4.3
# int[] oppositeDirection	Opposite direction in 8 direction 7.0.1 3.4.5 6...2 to 2....6 5.4.3 1.0.7
# Grid[][] grid	It is instance of Grid that keep first direction to move to destination.
# Novice player	The Novice bot control.
# boolean chkMove	Check bot move.
# Job job	Check Job of bot.

# int status	What bot done in previous frame. 0 is none, 1 is defence tower, 2 is attack tower, 3 is ecaping, 4 is escaping with bullet, 5 is farm
# int prevDirection	Previous direction of bot
# Pair destination	Destination
# boolean chkDestination	Check bot can move to Destination without enemy bullets.
# Entity target	The target bot will shoot.
# Queue<Grid> queue	Find shortest path with out enemy bullets.
# int[][] newPosition	New position of bot. (Use in queue)
# Random rand	Random
# Utility utility	Utility in bot

3.1.1.2 Constructor

+ Bot(Novice player)	Initialize player and set chkMove to false. Create utility to use in bot.

3.1.1.3 Method

+ void move()	Choose direction to move in bot.
# void upgradeSkill()	
# void upgradeStatus()	
# void upgradeJob()	
# void updateGrid()	Update grid and find direction bot can move to with out enemy bullets. (use queue and BFS)
# void findEntityInRange()	Add every Entity bot can view to ArrayList
# void move(int dir)	
# Entity chooseClosestTarget()	Choose target bot will shoot. (Closest target enemy.)
# void heuristicFood(int[] food)	Change some food in bot view. It makes sense that bot does not move to enemy territory.
# void farm()	Farm food for gain EXP and upgrade level.
# int getDirectionWithArea(int area)	The function will return the first direction that bot will move to area. (Bot attempt to escape enemy bullet)
# void escapeWithBullet()	Bot will escape with area that have min of enemy bullet. And attempt to dodge bullet with use grid.

# void escape()	Bot will back to its territory and if it has bullet, then it will call function escapeWithBullet.
# void moveWithDestination()	Bot will move to destination and get first direction in Grid.
# void defenseTower()	Bot will move to defence tower.
# void attackTower()	Bot will move to attack enemy tower.

3.1.2 Class BotNovice extends Bot

3.1.2.1 Field

- int[] upStatus	Sequence of upgrade status.
- int[] upSkill	Sequence of upgrade skill.
- int iteratorSkill	Iterator of Skill upSkill
- int iteratorStatus	Iterator of Skill upStatus

3.1.2.2 Constructor

+ BotNovice(Novice player)	super(player) and set sizeSkill and sizeStatus to 0.

3.1.2.3 Method

+ void move()	super(player) and set sizeSkill and sizeStatus to 0.
# void upgradeSkill()	Upgrade skill with upSkill[sizeSkill] and update 1 to sizeSkill.
# void upgradeStatus()	Upgrade skill with upSkill[sizeStatus] and update 1 to sizeStatus.
# void upgradeJob()	Choose job by random.

3.1.3 Class BotMagician extends BotNovice

3.1.3.1 Field

- int[] upStatus	Sequence of upgrade status.
- int[] upSkill	Sequence of upgrade skill.
- int iteratorSkill	Iterator of Skill upSkill
- int iteratorStatus	Iterator of Skill upStatus

3.1.3.2 Constructor

+ BotMagician(Magician player)	super(player) and set sizeSkill and sizeStatus to 0.

3.1.3.3 Method

+ void move()	super(player) and set sizeSkill and sizeStatus to 0.

# void upgradeSkill()	Upgrade skill with upSkill[sizeSkill] and update 1 to sizeSkill.
# void upgradeStatus()	Upgrade skill with upSkill[sizeStatus] and update 1 to sizeStatus.

3.1.4 Class BotRanger extends BotNovice

3.1.4.1 Field

- int[] upSkill	Sequence of upgrade skill.
- int sizeSkill	

3.1.4.2 Constructor

+ BotRanger(Ranger player)	super(player) and set sizeSkill and sizeStatus to 0.

3.1.4.3 Method

+ void move()	super(player) and set sizeSkill and sizeStatus to 0.
# void upgradeSkill()	Upgrade skill with upSkill[sizeSkill] and update 1 to sizeSkill.
# void upgradeStatus()	Upgrade skill with upSkill[sizeStatus] and update 1 to sizeStatus.

3.1.4 Class BotTank extends BotNovice

3.1.4.1 Field

- int[] upStatus	Sequence of upgrade status.
- int[] upSkill	Sequence of upgrade skill.
- int iteratorSkill	Iterator of Skill upSkill
- int iteratorStatus	Iterator of Skill upStatus

3.1.4.2 Constructor

+ BotTank(Tank player)	super(player) and set sizeSkill and sizeStatus to 0.

3.1.4.3 Method

+ void move()	super(player) and set sizeSkill and sizeStatus to 0.
# void upgradeSkill()	Upgrade skill with upSkill[sizeSkill] and update 1 to sizeSkill.
# void upgradeStatus()	Upgrade skill with upSkill[sizeStatus] and update 1 to sizeStatus.

3.1.5 Class *BotTower*

3.1.5.3 Method

+ void update()	Update direction to shoot each tower.
# void eachUpdate(Tower tower)	Each tower will shoot to closet target.

3.1.6 Class Utility

3.1.6.1 Field

- Novice player	The Player who control this utility.
- Random rand	For random

3.1.6.2 Constructor

+ Utility(Novice player)	Set player.

3.1.6.3 Method

# boolean willCollide(Entity entity, Pair position, double time)	Check entity with position if that time it will collide bullet return true, else false.
# int positionXInGrid(double x)	Get position x in real to Grid (change Component.MAX_SIZE to bot.SIZE_OF_GRID)
# int positionYInGrid(double y)	Get position y in real to Grid (change Component.MAX_SIZE to bot.SIZE_OF_GRID)
# double positionXInReal(int x)	Get position x in Grid to real (change bot.SIZE_OF_GRID to Component.MAX_SIZE)
# double positionYInReal(int y)	Get position y in Grid to real (change bot.SIZE_OF_GRID to Component.MAX_SIZE)
# Pair nextFrame(double direction)	Return Pair of position bot will move in next frame.
# boolean isVisible(Pair tmp)	Return true if bot can view and false otherwise.
# void changeDirectionToTarget(Pair target)	Change direction of muzzle to target.
# Pair getRef(Entity reference, Entity entity)	Get position of entity in view of reference. (red side will rotate 180 degree.)
# Pair getRef(Entity reference, Pair pair)	Get position of Pair in view of reference. (red side will rotate 180 degree.)
# Pair flip(Pair pair)	Rotate pair 180 degree.
# int checkCoordinateForEscape(Entity a, Pair b)	return direction to move to b.
# int checkCoordinate(Entity a, Pair b)	return direction to move to b. if distance between a and b less than 20, it will return value(-2) to tell bot to stop move.

# int canMoveWithDestination(Pair destination, int prevDirection)	Return true if bot can move to destination and false otherwise.
# boolean isTowerInRange()	Return true if tower in range and false otherwise.
# boolean isHitTheWall()	return true if bot will hit the wall and false otherwise.
# int getDirectionAdjacent(int prevDirection)	Check adjacent return direction that bot can move with no bullet attack.
# int getPositionInMap()	return position in map in 4 area.

3.2 Package buff

3.2.1 Class *Buff*

3.2.1.1 Field

# BuffType buffType	Type of Buff
# Novice player	Player who is affected by this Buff
# boolean isActive	Whether or not the Buff is active

3.2.1.2 Constructor

+ Buff(Novice player, BuffType buffType)	Set player and buffType.

3.2.1.3 Method

+ void activateBuff()	Set isActive to true and show Buff's effect.
+ void deactiveBuff()	Set isActive to false and stop Buff's effect.
+ void drawEffect()	
+ void undrawEffect()	
+ Image getImage()	
+ Novice getPlayer()	Getter of player
+ BuffType getBuffType()	Getter of buffType
+ boolean isActive()	Getter of isActive
+ Skill getSkill()	

3.2.2 Enum BuffType

3.2.2.1 Field

+ BuffType BUFF	Represents type of Buff of positive effect.
+ BuffType DEBUFF	Represents type of Buff of negative effect.

3.2.3 Class BurnBuff extends Buff implements Expirable

3.2.3.1 Field

- Image IMAGE	Image of BurnBuff icon
- int MAX_DURATION	Maximum duration of BurnBuff (in frame)
- Magician caster	Caster of this Buff
- int duration	Remaining duration of the Buff
- double burnDamage	Damage dealt per frame

3.2.3.2 Constructor

+ BurnBuff(Novice player, double burnDamage, Magician caster)	- Set corresponding fields. - Set buffType to Buff.DEBUFF. - Set duration to MAX_DURATION. - activateBuff.

3.2.3.3 Method

+ void drawEffect()	No initial effects happen here.
+ void undrawEffect()	No final effects happen here.
+ void update()	Decrease remaining duration, and cause burnDamage to player. - If remaining duration reaches 0, deactivateBuff.
+ Image getImage()	Getter of IMAGE
+ Skill getSkill()	Return null (no skill directly causing this Buff).
+ int getDuration()	Getter of duration
+ int getMaxDuration()	Getter of MAX_DURATION
+ double getBurnDamage	Getter of burnDamage

3.2.4 BurstBuff extends Buff implements Expirable

3.2.4.1 Field

- Skill SKILL	Skill causing BurstBuff
- Image IMAGE	Image of BurstBuff icon
- int maxDuration	Maximum duration of the Buff (in frame)
- int duration	Remaining duration of the Buff (in frame)

3.2.4.2 Constructor

+ BurstBuff(Novice player, int maxDuration)	- Set corresponding fields. - Set buffType to Buff.BUFF. - Set duration to maxDuration. - activateBuff.

3.2.4.3 Method

+ void update()	- Turn player to next angle. - Reset player's reloadCount and shoot. - Decrease duration. - If duration reaches 0, deactiveBuff.
+ void drawEffect()	- Set player's BurstBuff to true. - Increase bullet's damage.

+ void undrawEffect()	- Set player's BurstBuff to false. - Decrease bullet's damage.
+ int getDuration()	Getter of duration
+ Image getImage()	Getter of IMAGE
+ Skill getSkill()	Return null (no skill directly causing this Buff).
+ int getMaxDuration()	Getter of MAX_DURATION

3.2.5 Class DouAttBuff extends Buff

3.2.5.1 Field

- Skill SKILL	Skill causing DouAttBuff
- Image IMAGE	Image of DouAttBuff icon
- int level	Level of DouAtt skill

3.2.5.2 Constructor

+ DouAttBuff(Novice player, int level)	- Set corresponding fields. - Set buffType to Buff.BUFF. - activateBuff.

3.2.5.3 Method

+ void drawEffect()	Set player's ratioDoubleAtt according to level.
+ void undrawEffect()	No final effects happen here.
+ Image getImage()	Getter of IMAGE
+ Skill getSkill()	Return null (no skill directly causing this Buff).

3.2.6 Interface Expirable

3.2.6.3 Method

+ void update()	
+ int getMaxDuration()	
+ int getDuration()	

3.2.7 Class FireOrbBuff extends Buff

3.2.7.1 Field

- Skill SKILL	Skill causing FireOrbBuff
- Image IMAGE	Image of FireOrbBuff icon
- double burnDamage	Damage dealt per frame on other player

3.2.7.2 Constructor

+ FireOrbBuff(Novice player, double burnDamage)	- Set corresponding fields. - Set buffType to Buff.BUFF. - activateBuff.

3.2.7.3 Method

+ void drawEffect()	No initial effects happen here.
+ void undrawEffect()	No final effects happen here.
+ Image getImage()	Getter of IMAGE
+ Skill getSkill()	Getter of SKILL
+ double getBurnDamage()	Getter of burnDamage

3.2.8 Class FrenzyBuff extends Buff implements Expirable

3.2.8.1 Field

- Skill SKILL	Skill causing FrenzyBuff
- Image IMAGE	Image of FrenzyBuff icon
- double reloadFactor	How much player shoots faster during the Buff
- double slow	How much player moves slower during the Buff
- int duration	Remaining duration of the Buff (in frame)
- int maxDuration	Maximum duration of the Buff (in frame)

3.2.8.2 Constructor

+ FrenzyBuff(Novice player, int level, int duration)	- Set corresponding fields. - Set buffType to Buff.BUFF. - Set maxDuration to duration. - activateBuff.

3.2.8.3 Method

+ void drawEffect()	- Decrease player’s reloadDone by reloadFactor. - Decrease player’s speed by slow. - Increase player’s damageFactor.
+ void undrawEffect()	- Increase player’s reloadDone by reloadFactor. - Increase player’s speed by slow. - Decrease player’s damageFactor.
+ void update()	- Decrease duration. - If duration reaches 0, deactiveBuff.
+ int getMaxDuration()	Getter of maxDuration
+ int getDuration()	Getter of duration

+ Image getImage()	Getter of IMAGE
+ Skill getSkill()	Getter of SKILL

3.2.9 Class HasteBuff extends Buff implements Expirable

3.2.9.1 Field

- Skill SKILL	Skill causing HasteBuff
- Image IMAGE	Image of HasteBuff icon
- int duration	Remaining duration of the Buff (in frame)
- int maxDuration	Maximum duration of the Buff (in frame)

3.2.9.2 Constructor

+ HasteBuff(Novice player, int maxDuration)	- Set corresponding fields. - Set buffType to Buff.BUFF. - Set duration to maxDuration. - activateBuff.

3.2.9.3 Method

+ void drawEffect()	Increase player's speed.
+ void undrawEffect()	Decrease player's speed.
+ void update()	- Decrease duration. - If duration reaches 0, deactiveBuff.
+ int getDuration()	Getter of duration
+ int getMaxDuration()	Getter of maxDuration
+ Image getImage()	Getter of IMAGE
+ Skill getSkill()	Getter of SKILL

3.2.10 Class HealBuff extends Buff implements Expirable

3.2.10.1 Field

- Skill SKILL	Skill causing HealBuff
- Image IMAGE	Image of HealBuff icon
- int maxDuration	Maximum duration of the Buff (in frame)
- int duration	Remaining duration of the Buff (in frame)
- int level	Level of Heal skill

3.2.10.2 Constructor

+ HealBuff(Novice player, int level, int maxDuration)	- Set corresponding fields. - Set buffType to Buff.BUFF. - Set duration to maxDuration. - activateBuff.

3.2.10.3 Method

+ void update()	- Decrease duration. - If duration reaches 0, deactiveBuff.
+ void drawEffect()	Increase player's healthRegen.
+ void undrawEffect()	Decrease player's healthRegen.
+ int getDuration()	Getter of duration
+ int getMaxDuration()	Getter of maxDuration
+ Image getImage()	Getter of IMAGE
+ Skill getSkill()	Getter of SKILL

3.2.11 Class IceOrbBuff extends Buff

3.2.11.1 Field

- Skill SKILL	Skill causing IceOrbBuff
- Image IMAGE	Image of IceOrbBuff icon
- double slowFactor	Slow rate on other player

3.2.11.2 Constructor

+ IceOrbBuff(Novice player, double slowRate)	- Set corresponding fields. - Set buffType to Buff.BUFF. - activateBuff.

3.2.11.3 Method

+ void drawEffect()	No initial effects happen here.
+ void undrawEffect()	No final effects happen here.
+ Image getImage()	Getter of IMAGE
+ Skill getSkill()	Getter of SKILL
+ double getSlowFactor()	Getter of slowFactor

3.2.12 Class InvincibleBuff extends Buff implements Expirable

3.2.12.1 Field

- Image IMAGE	Image of InvincibleOrbBuff icon

- int maxDuration	Maximum duration of the Buff (in frame)
- int duration	Remaining duration of the Buff (in frame)
- double opacity	Current opacity of player
- double opacityChange	Change in opacity of player
- Thread blinking	Thread making player blink

3.2.12.2 Constructor

+ InvincibleBuff(Novice player)	- Set corresponding fields.
	- Set buffType to Buff.BUFF.
	- Set maxDuration and duration to 5 seconds.
	- Initiate blinking.
	  - Change opacity by opacityChange.
	  - If opacity < 0.01, opacityChange = +0.01.
	  - If opacity > 0.99, opacityChange = -0.01.
	  - Sleep 5 milliseconds in each loop.
	  - If interrupted, opacity = 1.
	- Start blinking and add blinking to GameComponent's threadList.
	- activateBuff.

3.2.12.3 Method

+ void drawEffect()	Set player's attack and damageFactor to 0.
+ void undrawEffect()	No final effects happen here. Note: When the Buff is deactivated, player's ability is recalculated in other method.
+ int getMaxDuration()	Getter of maxDuration
+ int getDuration()	Getter of duration
+ Image getImage()	Getter of IMAGE
+ Skill getSkill()	Getter of SKILL

3.2.13 Class ShieldBuff extends Buff implements Expirable

3.2.13.1 Field

- Skill SKILL	Skill causing ShieldBuff
- Image IMAGE	Image of ShieldBuff icon
- int duration	Remaining duration of the Buff (in frame)
- int maxDuration	Maximum duration of the Buff (in frame)
- double HPShield	Health point of shield

3.2.13.2 Constructor

+ ShieldBuff(Novice player, double HPShield, int duration)	- Set corresponding fields. - Set buffType to Buff.BUFF. - Set maxDuration to duration. - activateBuff.

3.2.13.3 Method

+ void update()	- Decrease duration. - If duration reaches 0, deactiveBuff.
+ void drawEffect()	Set player's HPShield to HPShield.
+ void undrawEffect()	- Set HPShield to player's HPShield. - Set player's HPShield to 0.
+ Image getImage()	Getter of IMAGE
+ Skill getSkill()	Getter of SKILL
+ int getMaxDuration()	Getter of maxDuration
+ int getDuration()	Getter of duration

3.2.14 Class SlowBuff extends Buff implements Expirable

3.2.14.1 Field

- Image IMAGE	Image of SlowBuff icon
- int MAX_DURATION	Maximum duration of SlowBuff (in frame)
- int duration	Remaining duration of the Buff
- double slowFactor	Slow rate on player

3.2.14.2 Constructor

+ SlowBuff(Novice player, double slowFactor)	- Set corresponding fields. - Set buffType to Buff.DEBUFF. - Set duration to MAX_DURATION. - activateBuff.

3.2.14.3 Method

+ void update()	Decrease remaining duration, and cause burnDamage to player. - If remaining duration reaches 0, deactivateBuff.
+ void drawEffect()	Decrease player's speed by slowFactor.
+ void undrawEffect()	Increase player's speed by slowFactor.
+ int getDuration()	Getter of duration
+ int getMaxDuration()	Getter of MAX_DURATION

+ Image getImage()	Getter of IMAGE
+ Skill getSkill()	Return null (no skill directly causing this Buff).

3.3 Package entity

3.3.1 Class *Entity*

3.3.1.1 Field

# Canvas canvas	Canvas of Entity
# Pair refPoint	Reference point of Entity
# double maxHp	Maximum health point of Entity
# double hp	Current health point of Entity
# double attack	Body damage of Entity
# double direction	Angle Entity is facing (against x-axis)
# Side side	Side of Entity
# boolean isDead	Whether or not Entity is dead

3.3.1.2 Constructor

+ Entity(Pair refPoint, double maxHp, double direction, Side side)	- Set corresponding fields. - Initiate Canvas. - Set isDead to false. - draw.

3.3.1.3 Method

+ void draw()	
+ void changeCenter()	
+ boolean isCollide(Entity entity)	Return whether sum of radii is greater than distance between the two refPoints.
+ void takeDamage(Entity entity)	
+ void die()	Set isDead to true and make canvas invisible.
+ void setRefPoint(Pair refPoint)	Setter of refPoint
+ void setDirection(double direction)	Setter of direction
+ void setAttack(double attack)	Setter of attack
+ Pair getRefPoint()	Getter of refPoint
+ Canvas getCanvas()	Getter of canvas
+ double getMaxHp()	Getter of maxHp
+ double getHp()	Getter of hp
+ double getAttack()	Getter of attack
+ double getDirection()	Getter of direction

+ Side getSide()	Getter of side
+ boolean isDead()	Getter of isDead
+ int getRadius()	
+ int getMaxRadius()	

3.4 Package entity.bullet

3.4.1 Class Bullet extends Entity implements Movable

3.4.1.1 Field

+ int RADIUS	Radius of Bullet
+ int LIFE_DURATION	Life duration of Bullet (in second)
+ int MAX_LIFE_CYCLE	Life duration of Bullet (in frame)
# double speed	Speed of Bullet
# Entity shooter	Shooter of Bullet
# int lifeCycleCount	Current past life duration (in frame)

3.4.1.2 Constructor

+ Bullet(Entity shooter, Pair refPoint, double maxHp, double direction, double attack, double speed, Side side)	- Set corresponding fields. - Set lifeCycleCount to 0.

3.4.1.3 Method

+ void takeDamage(Entity entity)	Decrease hp by entity's attack. - If hp is less than 0, set hp = 0 and die.
+ void draw()	Draw the shape of Bullet and changeCenter to GameComponent's player's refPoint.
+ void move()	- Move Bullet to current direction according to speed. - If Bullet is 50 pixels beyond boundary, die. - Increase lifeCycleCount. - If lifeCycleCount > MAX_LIFE_CYCLE, die.
+ void die()	Die and remove canvas from GameComponent's bulletPane.
+ void changeCenter(Pair center)	Translate canvas to correct position.
+ void setSpeed(double speed)	Setter of speed
+ int getRadius()	Getter of RADIUS
+ int getMaxRadius()	Return maximum radius of all kinds of Bullets which is IceBullet.RADIUS.
+ double getSpeed()	Getter of speed
+ double getLifeCycleCount()	Getter of lifeCycleCount
+ Entity getShooter()	Getter of shooter

3.4.2 Class FireBullet extends Bullet implements Rotatable

3.4.2.1 Field

- Image RED_ORB	Image of FireBullet of side Side.RED
- Image BLUE_ORB	Image of FireBullet of side Side.BLUE
+ int RADIUS	Radius of FireBullet
- double burnDamage	Damage dealt to hit player per frame
- int angle	Current turning angle

3.4.2.2 Constructor

+ FireBullet(Entity shooter, Pair refPoint, double maxHp, double direction, double attack, double speed, Side side, double burnDamage)	- Initiate Bullet with corresponding fields. - Set corresponding fields. - Set angle to 0.

3.4.2.3 Method

+ void rotate()	Change angle and rotate canvas according to angle.
+ void draw()	Draw the shape of FireBullet and changeCenter to GameComponent's player's refPoint.
+ int getRadius()	Getter of RADIUS
+ double getBurnDamage()	Getter of burnDamage

3.4.2 Class IceBullet extends Bullet implements Rotatable

3.4.2.1 Field

- Image RED_ORB	Image of IceBullet of side Side.RED
- Image BLUE_ORB	Image of IceBullet of side Side.BLUE
+ int RADIUS	Radius of IceBullet
- double slowFactor	Slow rate of hit player
- int angle	Current turning angle

3.4.2.2 Constructor

+ IceBullet(Entity shooter, Pair refPoint, double maxHp, double direction, double attack, double speed, Side side, double slowFactor)	- Initiate Bullet with corresponding fields. - Set corresponding fields. - Set angle to 0.

3.4.2.3 Method

+ void rotate()	Change angle and rotate canvas according to angle.
+ void draw()	Draw the shape of IceBullet and changeCenter to GameComponent's player's refPoint.
+ int getRadius()	Getter of RADIUS

+ double getSlowFactor()	Getter of slowFactor

3.5 Package food

3.5.1 Class Food extends Entity implements Rotatable

3.5.1.1 Field

+ int CANVAS_SIZE	Size of canvas of Food
+ int DEFAULT_MAX_HP	Default maximum health point of Food
+ int DEFAULT_ATTACK	Default attack of Food
+ int FOOD_EXP	Experience given by Food
- int rotateDirection	How direction changes

3.5.1.2 Constructor

+ Food(Pair refPoint)	- Set refPoint to given refPoint. - Set maxHp to DEFAULT_MAX_HP. - Set direction to a random positive integer less than 360. - Set attack to DEFAULT_ATTACK. - Set side to Side.NEUTRAL. - Randomly set rotateAngle to -1 or 1.

3.5.1.3 Method

+ void draw()	Draw the shape of Food and changeCenter to GameComponent's player's refPoint.
+ void rotate()	Change direction depending on rotateAngle and rotate canvas according to direction.
+ void takeDamage(Entity entity)	Decrease hp by entity's attack. - If hp is less than 0, set hp = 0 and die.
+ void die(Entity killer)	- Die. - If killer is a bullet and is shot from a player or killer is a player, that player gains experience of FOOD_EXP. - Remove canvas from GameComponent's foodPane.
+ int getRadius()	Return CANVAS_SIZE / 2.
+ int getMaxRadius()	Return getRadius().

3.6 Package entity.job

3.6.1 Class Magician extends Novice

3.6.1.1 Field

+ Job JOB	Job of Magician

3.6.1.2 Constructor

+ Magician(Novice oldPlayer)	- Copy every field from oldPlayer. - Add FireOrb and IceOrb skill.

3.6.1.3 Method

+ void draw()	- Draw the shape of Magician. - ChangeCenter to GameComponent's player's refPoint. - Draw hpBar and add to GameComponent.

+ void shoot()	- If reloadCount < reloadDone, do nothing.
	- Else
	  - Calculate whether this shot is critical or not and set the currentDamage.
	  - If Magician has FireOrbBuff, create FireBullet with burnDamage according to FireOrbBuff.
	  - Else if Magician has IceOrbBuff, create IceBullet with slowFactor according to IceOrbBuff.
	  - Else, create Bullet
	  - Add Bullet to GameComponent's bulletPane.
	  - Set reloadCount to 0.

+ void useSkill(int position)	- If FireOrb is used, remove IceOrbBuff. - Else if IceOrb is used, remove FireOrbBuff. - useSkill
+ Job getJob()	Getter of JOB
+ String toString()	Getter of JOB's name

3.6.2 Class Novice extends Entity implements Shootable, Rotatable, Shootable

3.6.2.1 Field

- double DEFAULT_MAX_HP	Default maximum health point of Novice
- double DEFAULT_ATTACK	Default attack of Novice
- double DEFAULT_SPEED	Default speed of Novice
- double DEFAULT_BULLET_DAMAGE	Default bullet damage of Novice
- double DEFAULT_BULLET_SPEED	Default bullet speed of Novice
- double DEFAULT_BULLET_HP	Default bullet health point of Novice

- double DEFAULT_CRITICAL_CHANCE	Default critical change of Novice
- double DEFAULT_CRITICAL_DAMAGE	Default critical damage of Novice
- int DEFAULT_RELOAD	Default number of frames to reload of Novice
- Job JOB	Job of Novice
+ int CANVAS_SIZE	Size of canvas of Novice
+ int RADIUS	Radius of Novice
# ArrayList<Skill> skillList	List of Skills of Novice
# ArrayList<Buff> buffList	List of Buffs of Novice
# double bulletDamage	Bullet damage of Novice
# double healthRegen	Health point regeneration of Novice
# double bulletHP	Bullet health point of Novice
# double bulletSpeed	Bullet speed of Novice
# double speed	Move speed of Novice
# double damageFactor	Damage factor of Novice
# int reloadDone	Number of frames to reload of Novice
# double criticalDamage	Critical damage of Novice
# double criticalChance	Critical chance of Novice
# Status status	Status of Novice
# String name	Name of Novice
# int kill	Kill count of Novice
# int death	Death count of Novice
# boolean isMoving	Whether or not Novice is moving
# boolean isPlayer	Whether or not Novice is a player
# boolean isChangeJobRequested	Whether or not Novice has requested to change Job
# double moveDirection	Angle which Novice is moving (against x-axis)
# Job newJob	Job which Novice has requested to change to
# HpBar hpBar	HpBar of Novice
# Identity identity	Identity of Novice
# Experience experience	Experience of Novice
# int reloadCount	Past reload frame counter of Novice

# Random random	Object used for random purposes

3.6.2.2 Constructor

+ Novice(Pair refPoint, Side side, String name)	- Set corresponding fields. - Create default buffList, skillList, Status. - Set hp to maxHp. - Set isMoving and isChangeJobRequested to false. - Set isPlayer to true. - Set moveDirection, kill and death to 0. - Add Haste and Heal skills to skillList. - Create Experience with level as 1 and currentExperience as 0. - Set everything else to default values. - Set reloadCount to reloadDone. - upgradeAbility.
+ Novice(Pair refPoint, Experience experience, Side side, int kill, int death, String name)	- Set corresponding fields. - Create default buffList, skillList, Status. - Set hp to maxHp. - Set isMoving and isChangeJobRequested to false. - Set isPlayer to true. - Set moveDirection to 0. - Add Haste and Heal skills to skillList. - Set everything else to default values. - Set reloadCount to reloadDone. - upgradeAbility.
+ Novice(Novice oldPlayer)	- Copy every field from oldPlayer. - Set isChangeJobRequested = false.

3.6.2.3 Method

+ void draw()	- Draw the shape of Novice - ChangeCenter to GameComponent's player's refPoint. - Draw hpBar and add to GameComponent.
+ void drawIdentity()	Create Identity of Novice and add to GameComponent.
+ void changeCenter()	Translate canvas to correct position.
+ void rotate()	Rotate canvas according to direction.
+ void setMoving(double moveDirection)	Set corresponding field and set isMoving to true.
+ void stopMoving()	Set isMoving to false.
+ void move()	If isMoving is true, move refPoint.

+ void heal(double amount)	Increase hp by amount or until hp reaches maxHp and draw hpBar.
+ void takeDamage(Entity entity)	TakeDamage by entity and entity's attack.

+ void takeDamage(Entity entity, double damage)	- Change damage by damageFactor.
	- Decrease hp by damage.
	- If hp < 0, die by entity and end.
	- Else
	  - If entity is IceBullet, addBuff SlowBuff by IceBullet's slowFactor.
	  - Else if entity is FireBullet, addBuff BurnBuff by FireBullet's burnDamage and shooter.

+ void die(Entity killer)	- Die, hpBar dies and identity dies. - Increase death and set reloadCount to 0. - If killer is a bullet and is shot from a player or killer is a player, that player gains one third of gainExperience and also increases kill. - DeactivateBuff of all buffs - Remove canvas from GameComponent's playerPane.

+ void shoot()	- If reloadCount < reloadDone, do nothing.
	- Else
	  - Calculate whether this shot is critical or not and set the currentDamage.
	  - Create and add Bullet to GameComponent's bulletPane.
	  - Set reloadCount to 0.

+ void reload()	If reloadCount > 1, decrease reloadCount.
+ void gainExp(double exp)	Add exp to experience and draw identity.
+ void useSkill(int position)	- If position is incorrect, do nothing and end. - Else if that Skill is not an ActiveSkill or that ActiveSkill is not ready, do nothing and end. - Else if Novice still contains Buff from that Skill and that Skill is Deactivable, deactivateBuff from that Skill and end. - Else, activateSkill.
+ void upgradeSkill(int position)	DecreaseSkillPoint and upgrade skill.
+ void addBuff(Buff buff)	- If that Buff is still on, removeBuff. - Add Buff to buffList. - If Novice is a player, add Buff to buffPane or debuffPane.

+ void removeBuff(Buff buff)	- DeactiveBuff and remove Buff from buffList. - If Novice is a player, remove Buff from buffPane or debuffPane.
+ void upgradeAbility()	- DeactivateBuff of all buffs. - CalculateAbility. - Reactivate all buffs again.
+ void calculateAbility()	Calculate bulletDamage, attack, etc.
+ void upgradeStatus(int status)	- If canUpgradeStatus and experience's pointStatus > 0, upgradeStatus and upgradeAbility. - If that status is VIT, heal hp.
+ void requestChangeJob(Job newJob)	- If newJob = null, set isChangeJobRequest to false and end. - Else, set corresponding field and set isChangeJobRequest to true.
+ Status getStatus()	Getter of status
+ void setSpeed(double speed)	Setter of speed
+ void setDamageFactor(double damageFactor)	Setter of damageFactor
+ void setReloadDone(int reloadDone)	Setter of reloadDone
+ void setPlayer(boolean isPlayer)	Setter of isPlayer
+ void setReloadCount()	Set reloadCount to reloadDone (reset reloadCount).
+ void setBulletDamage(double bulletDamage)	Setter of bulletDamage
+ void setHealthRegen(double healthRegen)	Setter of healthRegen
+ void setKill(int kill)	Setter of kill
+ void setDeath(int death)	Setter of death
+ int getRadius()	Setter of RADIUS
+ int getMaxRadius()	Return getRadius().
+ ArrayList<Skill> getSkillList()	Getter of skillList
+ ArrayList<Buff> getBuffList()	Getter of buffList
+ Experience getExperience()	Getter of experience
+ int getLevel()	Getter of experience's level
+ double getHealthRegen()	Getter of healthRegen
+ HpBar getHpBar()	Getter of hpBar
+ double getSpeed()	Getter of speed

+ double getDamageFactor()	Getter of damageFactor
+ int getReloadDone()	Getter of reloadDone
+ Job getJob()	Getter of JOB
+ Job getNewJob()	Getter of newJob
+ boolean isPlayer()	Getter of isPlayer
+ boolean isMoving()	Getter of isMoving
+ boolean isChangeJobRequested()	Getter of isChangeJobRequested
+ double getBulletDamage()	Getter of bulletDamage
+ int getKill()	Getter of kill
+ int getDeath()	Getter of death
+ String getName()	Getter of name
+ Identity getIdentity()	Getter of identity
+ String toString()	Getter of Job's name

3.6.3 Class Ranger extends Novice

3.6.3.1 Field

+ Job JOB	Job of Ranger
# double ratioDoubleAtt	Chance of double attack
# int shootingState	Which pistol is going to shoot

3.6.3.2 Constructor

+ Ranger(Novice oldPlayer)	- Copy every field from oldPlayer. - Set shootingState and ratioDouAtt to 0. - Add DouAtt and Frenzy skill.

3.6.3.3 Method

+ void draw()	- Draw the shape of Ranger. - ChangeCenter to GameComponent's player's refPoint. - Draw hpBar and add to GameComponent.

+ void shoot()	- If reloadCount < reloadDone, do nothing.
	- Else
	  - Calculate whether this shot is critical or not and set the currentDamage.
	  - create and add Bullet to GameComponent's bulletPane.
	  - Set reloadCount to 0.
	  - Change shootingState to another pistol.
	  - If this shot is a double attack shot, set reloadCount to relatively near to reloadDone.
	Note: In mouse clicking, new reloadCount is so close to reloadDone that player cannot release their mouse before another shot, so this will trigger double attack.

+ void upgradeSkill(int position)	- UpgradeSkill - If the skill is DouAtt, addBuff DouAttBuff with new skill level.
+ double getRatioDoubleAtt()	Getter of ratioDoubleAtt
+ void setRatioDoubleAtt(double ratioDoubleAtt)	Setter of ratioDoubleAtt
+ Job getJob()	Getter of JOB
+ String toString()	Getter of JOB's name

3.6.4 Class Tank extends Novice

3.6.4.1 Field

+ Job JOB	Job of Tank
# double HPShield	Health point of Shield of Tank
# boolean isBurstBuff	Whether or not Tank has BurstBuff

3.6.4.2 Constructor

+ Tank(Novice oldPlayer)	- Copy every field from oldPlayer. - Set isBurstBuff to false and HPShield to 0. - Add Shield and Burst skill.

3.6.4.3 Method

+ void draw()	- Draw the shape of Tank. - ChangeCenter to GameComponent's player's refPoint. - Draw hpBar and add to GameComponent.

+ void shoot()	- If reloadCount < reloadDone, do nothing.
	- Else
	  - Calculate whether this shot is critical or not and set the currentDamage.
	  - create and add Bullet to GameComponent's bulletPane.
	  - Calculate again whether this shot is critical or not and set the currentDamage.
	  - create Bullet on the other side and add it to GameComponent's bulletPane.
	  - Set reloadCount to 0.

+ void takeDamage(Entity entity, double damage)	- Change damage by damageFactor. - If HPShield > entity's damage, decrease HPShield by damage. - Else if HPShield > 0, set HPShield to 0, remove ShieldBuff and take 0 damage. - Else, takeDamage by entity and damage.
+ void calculateAbility()	Calculate bulletDamage, attack, etc. Note: Different formula from Novice.
+ double getHPShield()	Getter of HPShield
+ void setHPShield(double HPShield)	Setter of HPShield
+ boolean isBurstBuff()	Getter of isBurstBuff
+ void setBurstBuff(boolean isBurstBuff)	Setter of isBurstBuff
+ Job getJob()	Getter of JOB
+ String toString()	Getter of JOB's name

3.7 Package entity.property

3.7.1 Class Experience

3.7.1.1 Field

+ int MAX_LEVEL	Maximum level of player
- int level	Level of player
- double currentExp	Current experience point of player
- int skillPoint	Remaining skill point of player
- int spentSkillPoint	Spent skill point of player
- int pointStatus	Remaining status point of player

3.7.1.2 Constructor

+ Experience(int level, double currentExp)	- Set corresponding fields. - Set skillPoint to 1. - Set spentSkillPoint and pointStatus to 0. - updateLevel.

3.7.1.3 Method

+ void updateLevel()	- While level < MAX_LEVEL and currentExp > maxExp,
	  - Decrease currentExp by maxExp,
	  - Increase level and pointStatus,
	  - If new level <= 10 or is odd, increase skillPoint.

+ void addExp(double exp)	- Increase currentExp by exp and updateLevel. - If level >= MAX_LEVEL, set currentExp to 0.
+ void reborn()	Decrease gainedExperience by half and re-updateLevel.
+ double getGainedExperience()	Return gained experience since level 1.
+ int getLevel()	Getter of level
+ double getCurrentExp()	Getter of currentExp
+ double getMaxExp()	Return getMaxExp(level).
+ double getMaxExp(int level)	Return maximum experience at that level, which increases exponentially by level.
+ int getPointStatus()	Getter of pointStatus
+ int getSkillPoint()	Getter of skillPoint
+ int getSpentSkillPoint()	Getter of spentSkillPoint
+ boolean decreasePointStatus()	- If pointStatus > 0, decrease it and return true. - Else, return false.

+ void decreaseSkillPoint()	Decrease skillPoint and increase spendSkillPoint.

3.7.2 Class HpBar

3.7.2.1 Field

- Entity entity	Owner of HpBar
- double maxWidth	Maximum width of HpBar
- double space	Spacing between Entity and HpBar
- double width	Current width of HpBar
- Canvas canvas	Canvas of HpBar

3.7.2.2 Constructor

+ HpBar(Entity entity)	- Set corresponding field. - Set maxWidth and space depending on type of Entity. - Create canvas and draw.

3.7.2.3 Method

+ void draw()	- Draw HpBar according to Entity's hp and maxHp. - If Entity is Tank, also draw HPShield.
+ void changeCenter()	Translate canvas to correct position.
+ Canvas getCanvas()	Getter of canvas
+ void die()	Make canvas invisible and remove canvas from GameComponent's hpBarPane.
+ Entity getEntity()	Getter of entity

3.7.3 Class Identity

3.7.3.1 Field

- Novice player	Owner of Identity
- Canvas canvas	Canvas of HpBar

3.7.3.2 Constructor

+ Identity(Entity entity)	- Set corresponding field. - Create canvas and draw.

3.7.3.3 Method

+ void draw()	Draw Identity according to Entity's name and level.
+ void changeCenter()	Translate canvas to correct position.
+ Canvas getCanvas()	Getter of canvas

+ void die()	Make canvas invisible and remove canvas from GameComponent's identityPane.
+ Novice getPlayer()	Getter of player

3.7.4 Interface Movable

3.7.4.3 Method

+ void move()	
+ void setSpeed(double speed)	
+ double getSpeed()	

3.7.5 Interface Rotatable

3.7.5.3 Method

+ void rotate()	

3.7.6 Interface Shootable

3.7.6.3 Method

+ void shoot()	
+ void reload()	

3.7.7 Enum Side

3.7.7.1 Field

+ Side RED	Represents Entity of RED Side.
+ Side BLUE	Represents Entity of BLUE Side.
+ Side NEUTRAL	Represents Entity of NEUTRAL Side.

3.7.8 Class Status

3.7.8.1 Field

+ int MAX_STATUS	Maximum value of each Status
- int[6] stat	Each status, including STR, VIT, DEX, INT, AGI, LUK

3.7.8.2 Constructor

+ Status()	Set each stat to 1.

3.7.8.3 Method

+ void updateStatus(int status)	If stat[status] < MAX_STATUS, increase it.

+ void getStatus(int status)	- If status is incorrect, return 0. - Else, return stat[status].
+ boolean canUpgradeStatus(int status)	- If status is incorrect or stat[status] >= MAX_STATUS, return false. - Else, return true.
+ String getName(int status)	- If status is incorrect, return empty String. - Else, return String representing status.

3.8 Package tower

3.8.1 Class Tower extends Entity implements Rotatable, Shootable

3.8.1.1 Field

+ int CANVAS_SIZE	Size of canvas of Tower
+ int RADIUS	Radius of Tower
+ int MAX_HP	Maximum health point of Tower
+ int RELOAD_DONE	Number of frame to reload of Tower
+ int BULLET_HP	Bullet Health point of Tower
+ int BULLET_SPEED	Bullet speed of Tower
+ int BULLET_DAMAGE	Bullet damage of Tower
+ double CRITICAL_CHANCE	Critical chance of Tower
+ double CRITICAL_DAMAGE	Critical damage of Tower
# HpBar hpBar	HpBar of Tower
# int reloadCount	Past reload frame counter of Tower

3.8.1.2 Constructor

+ Tower(Pair refPoint, Side side)	- Set corresponding fields. - Set direction to 0. - Set the rest to constant.

3.8.1.3 Method

+ void draw()	- Draw the shape of Tower. - Remove old hpBar from GameComponent's hpBarPane. - Create a new one and add it to GameComponent's hpBarPane.
+ void changeCenter(Pair center)	Translate canvas to correct position.
+ void rotate()	Rotate canvas according to direction.

+ void takeDamage(Entity entity)	- If hp > entity's attack, decrease hp by entity's attack.
	- Else,
	  - Set hp to maxHp.
	  - Change side and redraw the whole tower.

+ void shoot()	- If reloadCount < reloadDone, do nothing.
	- Else
	  - Calculate whether this shot is critical or not and set the currentDamage.
	  - Create and add Bullet to GameComponent's bulletPane.
	  - Set reloadCount to 0.

+ void reload()	If reloadCount > 1, decrease reloadCount.
+ void heal()	Increase hp by amount or until hp reaches maxHp and draw hpBar.
+ void setSide(Side side)	Setter of side
+ HpBar getHpBar()	Getter of hpBar
+ int getRadius()	Getter of RADIUS
+ int getMaxRadius()	Return getRadius().
+ int getBulletSpeed()	Getter of BULLET_SPEED

3.9 Package environment

3.9.1 Class BloodPane extends Pane

3.9.1.1 Field

- Canvas[] bloodSpill	Canvas of blood.
- Canvas[] respawn	Canvas of remaining blood.
- Canvas text	Canvas of name text.
- double textOpacity	Opacity of text canvas.
- int spillStage	Stage of spill.
- int countDown	Countdown timer.

3.9.1.2 Constructor

+ BloodPane(Side side)	Set countDown to 3. Set spillStage to 0. Set Image in bloodSpill and respawn. Set opacity of all canvas to 0. Add all canvas in Pane.

3.9.1.3 Method

+ void drawDeadScene()	Show bloodSpill, text and respawn
+ void undrawDeadScene()	Set countDown to 3. Set spillStage to 0. Set textOpacity to 0. Set opacity of all canvas to 0.

3.9.2 Class BuffIcon extends Pane

3.9.2.1 Field

+ int ICON_SIZE	Default size of icon.
- Buff buff	Now buff in buffIcon.
- Canvas icon	Canvas of icon buff.
- Canvas duration	Canvas for cooldown.
- int position	Position in scene.

3.9.2.2 Constructor

+ BuffIcon(Buff buff, int position)	Set buff and position. Set opacity in all canvas to 0.65 Draw buff.

3.9.2.3 Method

+ void update()	Check buff is not expire. Set new duration cooldown. Draw new duration cooldown.
+ void clear()	Clear all canvas.
+ void setPosition(int position)	Setter of position.
+ Pair getShift()	Return Pair(x, y) of the right of buff canvas position.
+ Buff getBuff()	Getter of buff.

3.9.3 Class BuffPane extends Pane

3.9.3.1 Field

- ArrayList<BuffIcon> iconList	List of BuffIcon

3.9.3.2 Constructor

+ BuffPane()	Initialise iconList.

3.9.3.3 Method

+ void update()	Remove expired buff. Set new position to buff canvas.
+ void addBuff(Buff buff)	Add new buff to iconList and Pane.
+ void removeBuff(Buff buff)	Remove buff in iconList and Pane.
+ void clear()	Clear arrayList and Pane.

3.9.4 Class ClassIcon extends Pane

3.9.4.1 Field

+ double RADIUS	Default radius.
+ int ICON_SIZE	Default size of icon.
- Job job	Job of player
- Canvas icon	Icon canvas
- int position	Positon of each class.

3.9.4.2 Constructor

+ ClassIcon(Job job, int position)	Set job and position. Set opacity of icon to 0.8. Draw icon. Set on mouse clicked on canvas -> requestChangeJob(job). Add icon in pane.

3.9.4.3 Method

+ void clear()	Clear icon canvas.
+ void setPosition(int position)	Setter of Position. Translate icon to getShift().
+ Pair getShift()	Return Pair(x, y) of the right of icon canvas position.

3.9.5 Class ClassPane extends Pane

3.9.5.1 Field

- ArrayList<ClassIcon> classList	List of classList
- boolean isShowing	Check player can upgrade class.

3.9.5.2 Constructor

+ ClassPane()	Initialise classList. Set isShowing to false.

3.9.5.3 Method

+ void showClassPane()	Add class tank, magician and ranger and add all in component. Set isShowing to true.
+ void addClass(Job job)	Create ClassIcon job. Add icon to classList.
+ void addInComponent()	Add all icon in classList to pane.
+ void clear()	Clear all classIcon in classList. Clear all canvas in pane. Set isShowing to false.
+ boolean isShowing()	Getter of isShowing.

3.9.6 Class ExperienceBar extends Pane

3.9.6.1 Field

+ int MAX_WIDTH	Default max width.
+ int MAX_HEIGHT	Default max height.
- Experience experience	Experience of player.
- Canvas expCanvas	Canvas of exp player.
- String name	Name of player.
- Double maxExp	Max of exp player.

- double currentExp	Now exp player.
- Canvas textCanvas	Canvas text player.

3.9.6.2 Constructor

+ ExperienceBar()	Initialise expCanvas by MAX_WIDTH and MAX_HEIGHT. Set translate expCanvas to the right position. Initialise textCanvas by MAX_WIDTH and MAX_HEIGHT. Set translate textCanvas to the right position. Add expCanvas and textCanvas to pane.

3.9.6.3 Method

+ void setExperience(Experience experience)	Setter of experience.
+ void draw()	drawExperience() and drawText().
+ void drawExperience()	Draw now experience to expCanvas.
+ void drawText()	Draw now text to textCanvas.
+ void setName()	Setter of name.

3.9.7 Class Minimap extends Pane

3.9.7.1 Field

+ Image BLUE_TOWER	Image of blue tower.
+ Image RED_TOWER	Image of red tower.
+ Image NEUTRAL_TOWER	Image of newtral tower.
+ Image BLUE_TOWER_BORDER	Image of blue tower border.
+ Image RED_TOWER_BORDER	Image of red tower border.
+ Image NEUTRAL_TOWER_BORDER	Image of neutral tower border.
+ int MAP_SIZE	Default size of map.
+ double RATIO	Default ratio of map.
- Canvas boundary	Boundary canvas.
- Canvas player	Player canvas.
- Canvas tower	Tower canvas.
- Canvas viewBox	View box canvas.

3.9.7.2 Constructor

+ Minimap()	Initialise player and tower canvas to MAP_SIZE. drawBoundary(). Add player canvas and tower canvas to pane.

3.9.7.3 Method

- void drawBoundary()	Initialise boundary to MAP_SIZE. Draw canvas and add to pane.
+ void drawViewBox()	Initialise viewBox. Set width and height to Main.SCREEN_SIZE * RATIO. Change center to the right position. strokeRect and add viewBox to pane.
+ void ChangeCenter()	Change center to the right position.
+ void update()	Update blood of tower. Change any entity to the right position in minimap.
+ Pair onScreen(Pair position)	Return Pair of the right position on screen.

3.9.8 Class SkillIcon extends Pane

3.9.8.1 Field

+ int ICON_SIZE	Default size of icon.
+ int TEXT_SIZE	Default size of text.
+ Image PLUS_IMAGE	Default plus image.
- Novice player	Player
- Skill skill	Now skill in SkillIcon
- Canvas icon	Icon canvas.
- Canvas upgrade	Upgrade canvas.
- boolean isCurrentlyUpgradable	Skill can be upgraded or not.

3.9.8.2 Constructor

+ SkillIcon(Novice player, int position)	Set player and position. Create new canvas icon and upgrade. Change them to the right position. Draw all canvas. Set isCurrentlyUpgradable to true. DrawIcon(). UndrawUpgrade(). Add icon and upgrade canvas to pane.

3.9.8.3 Method

+ void drawIcon()	Draw Icon.

+ void drawUpgrade()	If isCurrentUpgradable return. Set isCurrentlyUpgradeable to true. Add plus image to pane. Set when mouse click, upgrade skill.
+ void undrawUpgrade()	Set isCurrentlyUpgradeable to false Unset on mouse click. Set opacity of upgrade to 0.
+ Pair getIconShift()	Shift icon to the right position.
+ boolean isUpgradable()	Return true if skill is upgradable, otherwise return false.
+ int getPosition()	Getter of skill.getPosition().

3.9.9 Class SkillPane extends Pane

3.9.9.1 Field

- Novice player	Player
- ArrayList<BuffIcon> iconList	List of SkillIcon

3.9.9.2 Constructor

+ SkillPane()	Initialise iconList.

3.9.9.3 Method

+ void addSkill(Skill skill)	Add skill to iconList. Add skill to pane.
+ void clear()	Clear all canvas in pane. Clear iconList.
+ void update()	Update SkillIcon in iconList.
+ void setPlayer(Novice player)	Setter of player.
+ ArrayList<SkillIcon> getIconList()	Getter of iconList.
+ Novice getPlayer()	Getter of player.
+ SkillIcon getIcon(int position)	Getter of icon in position.

3.9.10 Class StatusIcon extends Pane

3.9.10.1 Field

+ int MAX_WIDTH	Default max width.
+ int MAX_HEIGHT	Default max height.
+ int ICON_SIZE	Default icon size.

+ Image PLUS_IMAGE	Default plus image.
- Novice player	Player
- int position	Position of status icon.
- Canvas icon	Icon canvas.
- Canvas upgrade	Upgrade canvas.
- boolean isCurrentlyUpgradable	Status can be upgraded or not.

3.9.10.2 Constructor

+ StatusIcon(Novice player, int position)	Set player and position. Create new canvas icon and upgrade. Change them to the right position. Draw all canvas. Set isCurrentlyUpgradable to true. DrawIcon(). UndrawUpgrade(). Add icon and upgrade canvas to pane.

3.9.10.3 Method

+ void drawIcon()	Draw Icon.
+ void drawUpgrade()	If isCurrentUpgradable return. Set isCurrentlyUpgradeable to true. Add plus image to pane. Set when mouse click, upgrade skill.
+ void undrawUpgrade()	Set isCurrentlyUpgradeable to false Unset on mouse click. Set opacity of upgrade to 0.
+ boolean isUpgradable()	Shift icon to the right position.
+ Pair getIconShift()	Return true if status is upgradable, otherwise return false.
+ int getPosition()	Getter of position.

3.9.11 Class StatusPane extends Pane

3.9.11.1 Field

- Novice player	Player
- ArrayList<BuffIcon> iconList	List of SkillIcon

3.9.11.2 Constructor

+ SkillPane()	Initialise iconList.

3.9.11.3 Method

+ void update()	Draw Icon for each icon in iconList.
+ void setPlayer(Novice player)	Setter of player. Initialise icon in statusIcon and add to iconList and Pane.
+ ArrayList<SkillIcon> getIconList()	Getter of iconList.
+ Novice getPlayer()	Getter of player.
+ SkillIcon getIcon(int position)	Getter of icon in position.

3.10 Package exception

3.10.1 Class DeleteNullException extends Exception

3.10.1.1 Method

+ String getMessage()	Return “No characters to be deleted.”.

3.10.2 Class EmptyNameException extends Exception

3.10.2.1 Method

+ String getMessage()	Return “The name should not be null.”.

3.10.3 Class LongNameException extends Exception

3.10.3.1 Method

+ String getMessage()	Return “The name should not be longer than 10 characters.”.

3.10.4 Class UnsupportedCharacterException extends Exception

3.10.4.1 Field

+ String text	Text

3.10.4.2 Constructor

+ UnsupportedCharacterException(String text)	Set text.

3.10.4.3 Method

+ String getMessage()	Return “Character [” + text + “] is not allowed.”.

3.11 Package main

3.11.1 Class Main

3.11.1.1 Field

+ int SCREEN_SIZE	Size of game screen (square)
+ int FRAME_RATE	Frame rate of gameScene

3.11.1.3 Constructor

+ void main(String[] args)	Main method of the project
+ void start(Stage primaryStage)	- Set Stage of SceneManager to primaryStage. - SetMenuScene and setTitle of the stage. - SetResizable to false, then show.

3.11.2 Class SceneManager

3.11.2.1 Field

- Stage primaryStage	Stage of the game
- Scene gameScene	Game scene
- Scene menuScene	Menu scene
- Scene rankingScene	Ranking Scene
- boolean isMuted	Whether or not the game is muted - Default value is set to false.

3.11.2.3 Method

+ void setGameScene(String name, Side side)	- Initialize GameComponent with name and side. - Add all Panes and Canvases to the gameScene in an appropriate order and set scene size to Main.SCREEN_SIZE. - Add appropriate event handler to gameScene. - StartGame and set Scene to gameScene. - SetOnCloseRequest to stopTimer, stop GameComponent's sound and interrupt all Thread in threadList.
+ void setMenuScene()	- Reset GameComponent. - Add all Panes and Canvases to the menuScene in an appropriate order and set scene size to Main.SCREEN_SIZE. - Add appropriate event handler to gameScene. - Set Scene to menuScene. - SetOnCloseRequest to stopBackground and stopSound.

+ void setRankingScene()	- Add all Panes and Canvases to the rankingScene in an appropriate order and set scene size to Main.SCREEN_SIZE. - Add appropriate event handler to gameScene. - Set Scene to rankingScene. - SetOnCloseRequest to stopBackground and stop MenuComponent's sound.
+ void setStage(Stage primaryStage)	Setter of primaryStage
+ void setMuted(boolean isMuted)	- Setter of isMuted - DrawMute to MenuComponent. - SetMute to MenuComponent and RankingComponent.
+ boolean isMuted()	Getter of isMuted
+ void closeProgram()	Fire WINDOW_CLOSE_REQUEST to primaryStage.

3.12 Package main.game

3.12.1 Class GameComponent

3.12.1.1 Field

+ double MAX_SIZE	Maximum size of the map
+ int GRID_SIZE	Size of background grid (pixel)
+ int GRID_NUMBER	Number of grid in the background
+ int MAX_FOOD_COUNT	Number of Food in the game
+ int MAX_TRACK_NUMBER	Number of background music in the game
+ Media SHAKE_SOUND	Shake sound file
+ Media[MAX_TRACK_NUMBER] GAME_SOUND	Background music files
+ Media[2] RESULT_SOUND	Result sound files
- GameComponent instance	Singleton of GameComponent
- ArrayList<Novice> playerList	List of all players
- ArrayList<Bot> botList	List of Bots
- ArrayList<Tower> towerList	List of Towers
- ArrayList<Bullet> bulletList	List of Bullets
- ArrayList<Food> foodList	List of Food
- ArrayList<Thread> threadList	List of Threads
- Canvas[4] boundaryList	List of boundary canvases
- Novice player	Player in the game
- int trackNumber	Current GAME_SOUND
- MediaPlayer gameMP	GAME_SOUND player
- MediaPlayer shakeMP	SHAKE_SOUND player
- MediaPlayer resultMP	RESULT_SOUND player
- Pane endPane	Ending effect
- BloodPane bloodPane	Blood effect
- ClassPane classPane	Pane for class changing of player
- BuffPane buffPane	Pane for showing Buff of type Buff.BUFF of player
- BuffPane debuffPane	Pane for showing Buff of type Buff.DEBUFF of player
- StatusPane statusPane	Pane for upgrading Status of player
- SkillPane skillPane	Pane for upgrading Skill of player

- ExperienceBar expBar	Pane for showing player's Experience
- Minimap minimap	Pane for showing minimap
- Pane identityPane	Pane for showing Identities
- Pane hpBarPane	Pane for showing HpBars
- Pane playerPane	Pane for showing players
- Pane towerPane	Pane for showing Towers
- Pane bulletPane	Pane for showing Bullets
- Pane foodPane	Pane for showing Food
- Pane boundaryPane	Pane for showing boundaries
- Canvas grid	Background grid

3.12.1.2 Constructor

static	Initialize GAME_SOUND and RESULT_SOUND.
+ GameComponent()	- Initialize all panes (except bloodPane) and lists. - Draw grid and addBoundary. - Start a Thread initiating MediaPlayers.

3.12.1.3 Method

- void addBoundary()	Draw boundaries just wide and height enough to show on screen perfectly, and add to boundaryPane.
+ void shiftBoundary(Pair center)	Translate boundaries to correct positions.
+ Pair spawnPoint(Side side)	Random a spawn position according to side.
+ void initialize(Side side, String name)	- Create a player according to side and name. - SetPlayer skillPane and statusPane. - Set name and experience for expBar. - Add player, bots, and towers. - DrawViewBox on minimap and generateFood. - DrawIdentity and add InvincibleBuff for players. - Initialize bloodPane according to size. - PlayBGM.
+ void generateFood()	While current number of food < MAX_FOOD_COUNT, keep adding Food with random position such that, - It does not collide with anything. - The probability of Food position at the center is twice as high as at the border, for both axes.
- boolean collideWithList(Entity entity, ArrayList<? extends Entity> list)	Return whether or not entity collides with at least a member in list.

+ void addComponent(Object component)	Add component to corresponding list and pane. - If component is a kind of Novice and is not player, also add corresponding Bot.
+ void removeComponent(Object component)	Remove component to corresponding list and pane.
+ void setEnding(String mode)	- Clear old endPane. - Draw the invisible point-sized icon and play RESULT_SOUND according to mode. - Play Timeline to zoom icon in while fading up, then delay. - After delay, reset RankingComponent's instance, allowChangeScene, and show corresponding text.
+ void playBGM()	- Randomize a song, then set gameMP. - Set mute according to SceneManager's isMuted. - Play the song. - When the song ends, keep playingBGM.
+ void playShakeSound()	Set shakeMP and set mute according to SceneManager's isMuted, then play once.
+ void playResultSound(String mode)	Set resultMP according to mode and set mute according to SceneManager's isMuted, then play once.
+ void stopBGM()	Stop gameMP.
+ void setMute(boolean isMuted)	Set mute of all MediaPlayers according to isMuted.
+ void stopSound()	Stop all MediaPlayers.
+ void reset()	- Initialize all panes (except bloodPane) and lists. - Draw grid and addBoundary.
+ GameComponent getInstance()	Getter of instance
+ void setPlayer(Novice player)	Setter of player
+ Pair getShift()	Getter of player's refPoint
+ Novice getPlayer()	Getter of player's refPoint
+ ArrayList<Novice> getPlayerList()	Getter of playerList
+ ArrayList<Bot> getBotList()	Getter of botList
+ ArrayList<Tower> getTowerList()	Getter of towerList
+ ArrayList<Bullet> getBulletList()	Getter of bulletList
+ ArrayList<Food> getFoodList()	Getter of foodList
+ ArrayList<Thread> getThreadList()	Getter of threadList
+ Pane getEndPane()	Getter of endPane

+ BloodPane getBloodPane()	Getter of bloodPane
+ BuffPane getBuffPane()	Getter of buffPane
+ BuffPane getDebuffPane()	Getter of debuffPane
+ StatusPane getStatusPane()	Getter of statusPane
+ SkillPane getSkillPane()	Getter of skillPane
+ ClassPane getClassPane()	Getter of classPane
+ ExperienceBar getExpBar()	Getter of expBar
+ Minimap getMinimap()	Getter of minimap
+ Pane getIdentityPane()	Getter of identityPane
+ Pane getHpBarPane()	Getter of hpBarPane
+ Pane getPlayerPane()	Getter of playerPane
+ Pane getTowerPane()	Getter of towerPane
+ Pane getBulletPane()	Getter of bulletPane
+ Pane getFoodPane()	Getter of foodPane
+ Pane getBoundaryPane()	Getter of boundaryPane
+ Canvas[] getBoundaryList()	Getter of boundaryList
+ Canvas getGrid()	Getter of grid

3.12.2 Class GameHandler

3.12.2.1 Field

+ class PairFx<S, T> extends javafx.util.Pair<S, T> Constructor: + Pair<S, T>(S s, T t)	Equivalence to typedef of javafx.util.Pair<S, T> Constructor: Set corresponding fields
- int SPAWN_TIME	Number of frame to spawn after dead
- HashSet<KeyCode> activeKey	Set of KeyCodes that are being pressed
- HashSet<MouseButton> activeMouse	Set of MouseEvents that are being pressed
- HashSet<PairFx<Entity, Entity>> collisionPair	Set of collided pair of Entity in that frame
- HashMap<Novice, Integer> deadPlayer	Map of dead player and past number of frames
- boolean isAutoshoot	Whether or not auto-shoot is activated
- boolean justDead	Whether or not GameComponent's player has just died in that frame

- boolean isEnd	Whether or not the game has ended
- boolean changeSceneReady	Whether or not scene changing is allowed
- Timeline timer	Timer of the game

3.12.2.2 Constructor

static	- Create empty sets and map. - Set all boolean fields to false.

3.12.2.3 Method

+ void keyPressed(KeyEvent event)	- If the key is Enter and changeSceneReady,
	  - Reset and stop GameComponent's sound.
	  - Start MenuComponent's sound.
	  - RankingComponent starts movingBackground.
	  - SetRankingScene.
	- If the key is game-related key, add to activeKey.
	  - If the key is Command, add as Control.
	- If Number key is just pressed,
	  - If activeKey contains Control, attempt to upgradeStatus.
	  - Else if activeKey contains Shift, attempt to upgradeSkill.
	  - Else, attempt to useSkill.
	- If the key is E, change autoshoot.
	- If the key is M, change SceneManager's isMuted.
	// CHEAT
	- If the key is =, upgrade level, skill and statuses to maximum possible.
	- If the key is [, instantly win.
	- If the key is], instantly lose.

+ void keyReleased(KeyEvent event)	Remove the key from activeKey. - If the key is Command, remove Control instead.
+ void mousePressed(MouseEvent event)	Add corresponding MouseButton to activeMouse.
+ void mouseReleased(MouseEvent event)	Remove corresponding MouseButton to activeMouse.
+ void startGame()	Set timer to run update and BotTower's update in every frame, then start timer.
+ void changeDirection(MouseEvent event)	- If isEnd, do nothing and end. - Else If GameComponent's player is on BurstBuff, do nothing and end. - Else, rotate GameComponent's player's canvas to mouse position.

+ void update()	- If Space is currently pressed or PrimaryMouse is currently clicked or auto-shoot is currently activated, player shoots. - UpdateDeadPlayer - CheckPlayerUpgrade and updateReloadAndSkill - MoveComponent and movePlayer - MoveCenter to GameComponent's player's refPoint - CheckCollision and clearDeadComponent - ChangeClass - CheckEnding - GenerateFood

- void updateDeadPlayer()	- For each member in deadPlayer,
	  - If frame count >= SPAWN_TIME or the game has ended,
	  - Create a new Novice according to member, with old Experience reborn.
	  - DrawIdentity and add InvincibleBuff.
	  - If member is GameComponent's player, undrawDeadScene, clear skillPane, statusPane, classPane and set player for those Panes.
	  - Else, increase frame count.
	- Remove the reborn players.

- void checkPlayerUpgrade()	- Draw or undraw each StatusIcon and SkillIcon depending on whether they are upgradable or not. - If GameComponent's player can change job, ShowClassPane.

- void updateReloadAndSkill()	- For each player,
	  - Heal player according to player's healthRegen.
	  - Reload ammo, reload ActiveSkills, update Buffs,
	  - If some Buffs are deactivated, remove them from player's buffList and upgradeAbility.
	  - If some Skills have not been set with a player, set them.
	- Reload ammo for Towers and heal Towers.
	- Update statusPane, skillPane, buff and debuffPane.

- void moveComponent()	- Bots decide to move or not. - Move Bullets and rotate Rotatable Bullets. - Rotate Food.
- void movePlayer()	Move player according to activeKey and speed.

- void moveCenter(Pair center)	Set center for everything, shiftBoundary, and translate Grid to correct position.

- void checkCollision()	- PairwiseCheckCollision for
	  - bulletList and foodList,
	  - bulletList and towerList,
	  - bulletList and playerList,
	  - playerList and foodList,
	  - playerList and towerList.
	- If GameComponent's player is dead and justDead is false, set justDead to true and drawDeadScene.
	- Draw GameComponent's player's hpBar.

- void pairwiseCheckCollision(ArrayList<? extends Entity list1, ArrayList<? extends Entity> list2)	- If at least 1 list is empty, end and do nothing. - Union 2 lists together. - Sort by refPoint's second. - Clear collisionPair. - Use another PairwiseCheckCollision to the merged list.

- void pairwiseCheckColliosion(ArrayList<? extends Entity> entityList, double shift, double width, double maxDist)	- Check collision within entityList using the algorithm adapted Closest Pair Algorithm, with the band in the middle of size 2 times of maxDist.
	- If two entity collide, are from different Side, and never collide in this frame,
	  - Add them to collisionPair.
	  - Each of them takesDamage from each other.

- void clearDeadComponent()	- For each member in playerList,
	  - If that member is dead, put to deadPlayer and remove from playerList.
	  - If that member is a bot and is dead, remove from botList.
	- Remove dead Bullets and dead Food from lists.
	- Update minimap.

- void changeClass()	For each member in playerList, - If member's isChangeJobRequested, create a new player of that Job and remove old player and its component. - If member is GameComponent's player, also setPlayer, and setPlayer for skillPane, statusPane and expBar.

- void changeEnding()	If all Towers are of the same Side and not of Side Side.NEUTRAL, - If Towers are of same Side as GameComponent's player, endGame of mode Victory. - Else, endGame of mode Defeat.
- void endGame(String mode)	- Set isDead to true. - UpdateDeadPlayer and stopTimer. - StopBGM and playShakeSound. - Shake screen then setEnding of mode mode.
+ void stopTimer()	Stop timer.
+ void allowChangeScene()	Set changeSceneReady to true.
+ void rebornReset()	Set autoshoot and justDead to false.
+ void reset()	RebornReset and set isEnding and changeSceneReady to false.

3.13 Package main.menu

3.13.1 Class MenuComponent

3.13.1.1 Field

+ Image MUTE	Mute icon
+ Image UNMUTE	Unmute icon
+ Image LEFT_ARROW	Left arrow icon
- Media START_SOUND	Initial sound
- Media LOOP_SOUND	Loop sound
- Media TRANSITION_SOUND	Transition sound
- MenuComponent instance	Singleton of the class
- MediaPlayer startMP	START_SOUND player
- MediaPlayer loopMP	LOOP_SOUND player
- MediaPlayer transitionMP	TRANSITION_SOUND player
- Pane backgroundPane	Background Pane
- Pane namePane	Name Pane
- Pane sidePane	Side choosing Pane
- Canvas background	Background Canvas
- Canvas text	Text Canvas
- Canvas red	Red Side Canvas
- Canvas blue	Blue Side Canvas
- Canvas moveBack	Left arrow Canvas
- Canvas mute	Mute icon Canvas
- String name	Name typed
- Timeline mover	Background mover
- int moveCount	Background move counter

3.13.1.2 Constructor

+ MenuComponent(String name)	- Draw components on the screen and sideScreen. - DrawName according to name. - SetMute according to SceneManager's isMuted. - Add appropriate handlers and moveBackground. - Start Thread for loading Media and start after that.

3.13.1.3 Method

+ void addCharacter(KeyEvent event) throws Exception	- DrawName according to event. - Exceptions are thrown if needed.
+ void drawName()	Draw name.
+ void drawMute()	Draw MUTE or UNMUTE icon.
+ void setErrorMessage(String error)	Draw error text.
+ void clearErrorMessage()	Undraw error text.
+ void moveBackground()	Move background.
+ void stopBackground()	Stop mover.
+ void startSound()	SetMute according to SceneManager's isMuted and start sounds.
+ void startTransitionSound()	SetMute according to SceneManager's isMuted and start TRANSITION_SOUND.
+ void setMute(boolean isMuted)	SetMute to all MediaPlayers.
+ void stopSound()	Stop all sounds.
+ void resetName()	- Set name to empty string and drawName. - Set TRANSITION_SOUND to transitionMP.
+ void setMoveCount(int moveCount)	Setter of moveCount
+ MainComponent getInstance()	Getter of instance
+ Pane getBackgroundPane()	Getter of backgroundPane
+ Pane getNamePane()	Getter of namePane
+ Pane getSidePane()	Getter of sidePane
+ Canvas getRed()	Getter of red
+ Canvas getBlue()	Getter of blue
+ Canvas getMoveBack()	Getter of moveBack
+ String getName()	Getter of name

3.13.2 Class GameHandler

3.13.2.1 Field

- int shift	Shift of canvas
- boolean onSideScreen	Whether or not on side screen
- boolean isMPressed	Whether or not M is pressed

3.13.2.2 Constructor

static	- Set shift to 0. - Set all boolean variable to false.

3.13.2.3 Method

+ void keyPressed(KeyEvent event)	- If ESC is pressed, closeProgram and end. - Else if screen is onSideScene and M is justed pressed, set isMPressed to true and change SceneManager's isMuted and end. - Else, try to do what event suggests. - If an error is thrown, showErrorMessage.
+ void keyReleased(KeyEvent event)	If M is released, set isMPressed to false.
+ void moveCanvasForth() throws EmptyNameException	- If name's length = 0, throw EmptyNameException. - Set onSideScreen to true. - Start Timeline to shift canvas to show sideScreen.

+ void moveCanvasBack()	- If screen is onSideScreen,
	  - Set onSideScreen to false.
	  - Start Timeline to shift canvas back to show nameScreen.

+ void shiftCanvas()	Shift canvases according to shift.
+ void moveToGameSceen(String name, Side side)	- StopBackground and stopSound. - SetGameScene. - MoveCanvasBack and resetName.
+ void drawBorder(Canvas canvas, Color color)	Draw border of color for canvas.
+ void undrawBorder(Canvas canvas)	Draw border of gray color for canvas.
+ void drawLightButtonBackground()	Change background of LEFT_ARROW to light gray.
+ void drawDarkButtonBackground()	Change background of LEFT_ARROW to dark gray.

3.14 Package main.ranking

3.14.1 Class RankingComponent

3.14.1.1 Field

- double ICON_SIZE	Size of player's icon
- HashMap<Job, Image> ICON	Icon map for each Job
- RankingComponent instance	Singleton of the class
- Side winnerSide	Side of winner
- Pane backgroundPane	Background Pane
- Pane rankingPane	Ranking Pane
- Canvas background	Background canvas
- ArrayList<Novice> playerList	List of players in ranking
- int moveCount	Background move counter

3.14.1.2 Constructor

static	Add Images to ICON.
+ RankingComponent()	- Set winnerSide. - Initialize and draw components on the screen. - CalculateRanking and drawRanking.

3.14.1.3 Method

+ void calculateRanking()	- Clear and add players to playerList. - Sort them in appropriate order.
+ void drawRanking()	Draw ranking on the table and draw transition text.
+ void moveBackground()	Move background.
+ void stopBackground()	Stop mover.
+ void reset()	- Clear playerList and rankingPane. - CalculateRanking and drawRanking.
+ RankingComponent getInstance()	Getter of instance
+ Pane getBackgroundPane()	Getter of backgroundPane
+ Pane getRankingPane()	Getter of rankingPane
+ int getMoveCount()	Getter of moveCount

3.14.2 Class GameHandler

3.14.2.1 Field

- boolean isMPressed	Whether or not M is pressed

3.14.2.3 Method

+ void keyPressed(KeyEvent event)	- If screen is onSideScene and M is justed pressed, set isMPressed to true and change SceneManager's isMuted and end.
	- Else if Enter is pressed,
	  - StopBackground.
	  - Set MenuComponent's moveCount and moveBackground.
	  - Reset and setMenuScene.

+ void keyReleased(KeyEvent event)	If M is released, set isMPressed to false.

3.15 Package skill

3.15.1 class *ActiveSkill* extends Skill

3.15.1.1 Field

# int maxCooldown	Timing for cooldown of skill.
# int remainingCooldown	Remaining time cooldown.

3.15.1.2 Constructor

+ ActiveSkill(int position, int maxLevel, int maxCooldown)	Set super(position, maxLevel). Set maxCooldown. Set remainingCooldown to 0.

3.15.1.3 Method

+ void update()	Reduce cool down.
+ void activateSkill()	Set remainingCooldown = maxCooldown drawEffect() activateEffect()
+ void reset()	Set remainingCooldown = 0
+ boolean isReady()	Check Skill can be used.
+ int getMaxCooldown()	Getting of MaxCooldown.
+ int getRemainingCooldown()	Getting of RemainingCooldown.
# protected void setMaxCooldown(int maxCooldown)	Setting of maxCooldown.

3.15.2 Class Burst extends ActiveSkill

3.15.2.1 Field

- int DEFAULT_COOLDOWN	Default cool down
- int COOLDOWN_PER_LEVEL	Default max HP.
- int DEFAULT_DURATION	Default duration.
- int POSITION	Default position in array skill.
- int MAX_LEVEL	Default max level.
- Image IMAGE	Default Image Skill.

3.15.2.2 Constructor

+ Burst()	Set super(POSITION, MAX_LEVEL, DEFAULT_COOLDOWN)

3.15.2.3 Method

+ void activateSkill()	Set new MaxCooldown Super.activateSkill()

# void drawEffect()	Nothing happen here.
# void activateEffect()	AddBuff BurstBuff
+ Image getImage()	Getting of IMAGE
+ String toString()	Return “Burst”

3.15.3 Interface Deactivable

3.15.3.1 Method

+ void deactivateEffect()	

3.15.4 Class DouAtt extends Skill

3.15.4.1 Field

- int POSITION	Default position in array skill.
- int MAX_LEVEL	Default max level.
- Image IMAGE	Default Image Skill.

3.15.4.2 Constructor

+ DouAtt()	Set super(POSITION, MAX_LEVEL)

3.15.4.3 Method

# void drawEffect()	Nothing happen here.
# void activateEffect()	AddBuff DouAttBuff
+ Image getImage()	Getting of IMAGE
+ String toString()	Return “DouAtt”

3.15.5 Class FireOrb extends ActiveSkill implements Deactivable

3.15.5.1 Field

- int POSITION	Default position in array skill.
- int MAX_LEVEL	Default max level.
- Image IMAGE	Default Image Skill.

3.15.5.2 Constructor

+ FireOrb()	Set super(POSITION, MAX_LEVEL, 0)

3.15.5.3 Method

# void drawEffect()	Nothing happen here.
# void activateEffect()	AddBuff FireOrbBuff
# void deactivateEffect()	Remove Buff IceOrbBuff

+ Image getImage()	Getting of IMAGE
+ String toString()	Return “Fire Orb”

3.15.6 Class Frenzy extends ActiveSkill

3.15.6.1 Field

- int DEFAULT_COOLDOWN	Default cool down
- int DEFAULT_DURATION	Default duration.
- int POSITION	Default position in array skill.
- int MAX_LEVEL	Default max level.
- Image IMAGE	Default Image Skill.

3.15.6.2 Constructor

+ Frenzy()	Set super(POSITION, MAX_LEVEL, DEFAULT_COOLDOWN)

3.15.6.3 Method

# void drawEffect()	Nothing happen here.
# void activateEffect()	AddBuff FrenzyBuff
+ Image getImage()	Getting of IMAGE
+ String toString()	Return “Frenzy”

3.15.7 Class Haste extends ActiveSkill

3.15.7.1 Field

- int DEFAULT_COOLDOWN	Default cool down
- int COOLDOWN_PER_LEVEL	Default cool down per level.
- int DEFAULT_DURATION	Default duration.
- int POSITION	Default position in array skill.
- int MAX_LEVEL	Default max level.
- Image IMAGE	Default Image Skill.

3.15.7.2 Constructor

+ Haste()	Set super(POSITION, MAX_LEVEL, DEFAULT_COOLDOWN)

3.15.7.3 Method

+ void activateSkill()	Set Max Cooldown Super.activateSkill()

# void drawEffect()	Nothing happen here.
# void activateEffect()	AddBuff HasteBuff
+ Image getImage()	Getting of IMAGE
+ String toString()	Return “Frenzy”

3.15.8 Class Heal extends ActiveSkill

3.15.8.1 Field

- int DEFAULT_COOLDOWN	Default cool down
- int DEFAULT_DURATION	Default duration.
- int POSITION	Default position in array skill.
- int MAX_LEVEL	Default max level.
- Image IMAGE	Default Image Skill.

3.15.8.2 Constructor

+ Heal()	Set super(POSITION, MAX_LEVEL, DEFAULT_COOLDOWN)

3.15.8.3 Method

+ void activateSkill()	Super.activateSkill()
# void drawEffect()	Nothing happen here.
# void activateEffect()	AddBuff HealBuff
+ Image getImage()	Getting of IMAGE
+ String toString()	Return “Heal”

3.15.9 Class FireOrb extends ActiveSkill implements Deactivable

3.15.9.1 Field

- int POSITION	Default position in array skill.
- int MAX_LEVEL	Default max level.
- Image IMAGE	Default Image Skill.

3.15.9.2 Constructor

+ IceOrb()	Set super(POSITION, MAX_LEVEL, 0)

3.15.9.3 Method

# void drawEffect()	Nothing happen here.
# void activateEffect()	AddBuff IceOrbBuff

# void deactivateEffect()	Remove Buff IceOrbBuff
+ Image getImage()	Getting of IMAGE
+ String toString()	Return “Ice Orb”

3.15.10 Class Shield extends ActiveSkill

3.15.10.1 Field

- int DEFAULT_COOLDOWN	Default cool down
- int DEFAULT_DURATION	Default duration.
- int POSITION	Default position in array skill.
- int MAX_LEVEL	Default max level.
- Image IMAGE	Default Image Skill.

3.15.10.2 Constructor

+ Shield()	Set super(POSITION, MAX_LEVEL, DEFAULT_COOLDOWN)

3.15.10.3 Method

# void drawEffect()	Nothing happen here.
# void activateEffect()	AddBuff ShieldBuff
+ Image getImage()	Getting of IMAGE
+ String toString()	Return “Shield”

3.15.11 Class *Skill*

3.15.11.1 Field

# Novice caster	Novice who cast this skill.
# Canvas effect	Canvas of effect.
# int position	Position in array skill.
# int maxLevel	Max Level of skill.
# int level	Now Level.

3.15.11.2 Constructor

+ Skill(int position, int maxLevel)	Set position and maxLevel. Set caster to null and level to 0.

3.15.11.3 Method

# void drawEffect()	
# void activateEffect()	

+ void upgrade()	Upgrade level
+ Boolean isReady()	Return true if level > 0, Otherwise return false.
+ boolean isUpgradable()	Return true if skill can upgrade, Otherwise return false.
+ void setCaster(Novice caster)	Seting of caster.
+ Novice getCaster()	Getting of caster.
+ int getLevel()	Getting of level.
+ int getMaxLevel()	Getting of maxLevel.
+ int getPosition()	Getting of position.
+ Image getImage()	

3.16 Package utility

3.16.1 Class Grid implements Comparable<Grid>

3.16.1.1 Field

- int x, y	Coordinate x and y in Pane
- int gridX, gridY	Coordinate x and y in table grid
- double time	Time that bot will come to (x, y)
- boolean chk	Check bot can move to (x, y)
- int firstDirection	First direction bot moved.

3.16.1.2 Constructor

+ Grid()	Set time to 0 and chk = false
+ Grid(int x, int y, int gridX, int gridY, int time, boolean chk, int firstDirection)	Assign value to each variable.
+ Grid(Grid o)	Copy field of o to this

3.16.1.3 Method

+ int compareTo(Grid other)	Compare function
Generated all getter and setter	

3.16.2 Enum Job

3.16.2.1 Field

+ Job NOVICE	Field enum job.
+ Job TANK	Field enum job.
+ Job MAGICIAN	Field enum job.
+ Job RANGER	Field enum job.
+ Job GUARDIAN	Field enum job.
+ Job PALADIN	Field enum job.
+ Job PYROMANCER	Field enum job.
+ Job CRYOMANCER	Field enum job.
+ Job CANNONNEER	Field enum job.
+ Job SNIPER	Field enum job.
- String name	string name.

3.16.2.2 Constructor

- Job(String name)	Set name

3.16.2.3 Method

+ String toString()	Return name

3.16.3 Class Pair

3.16.3.1 Field

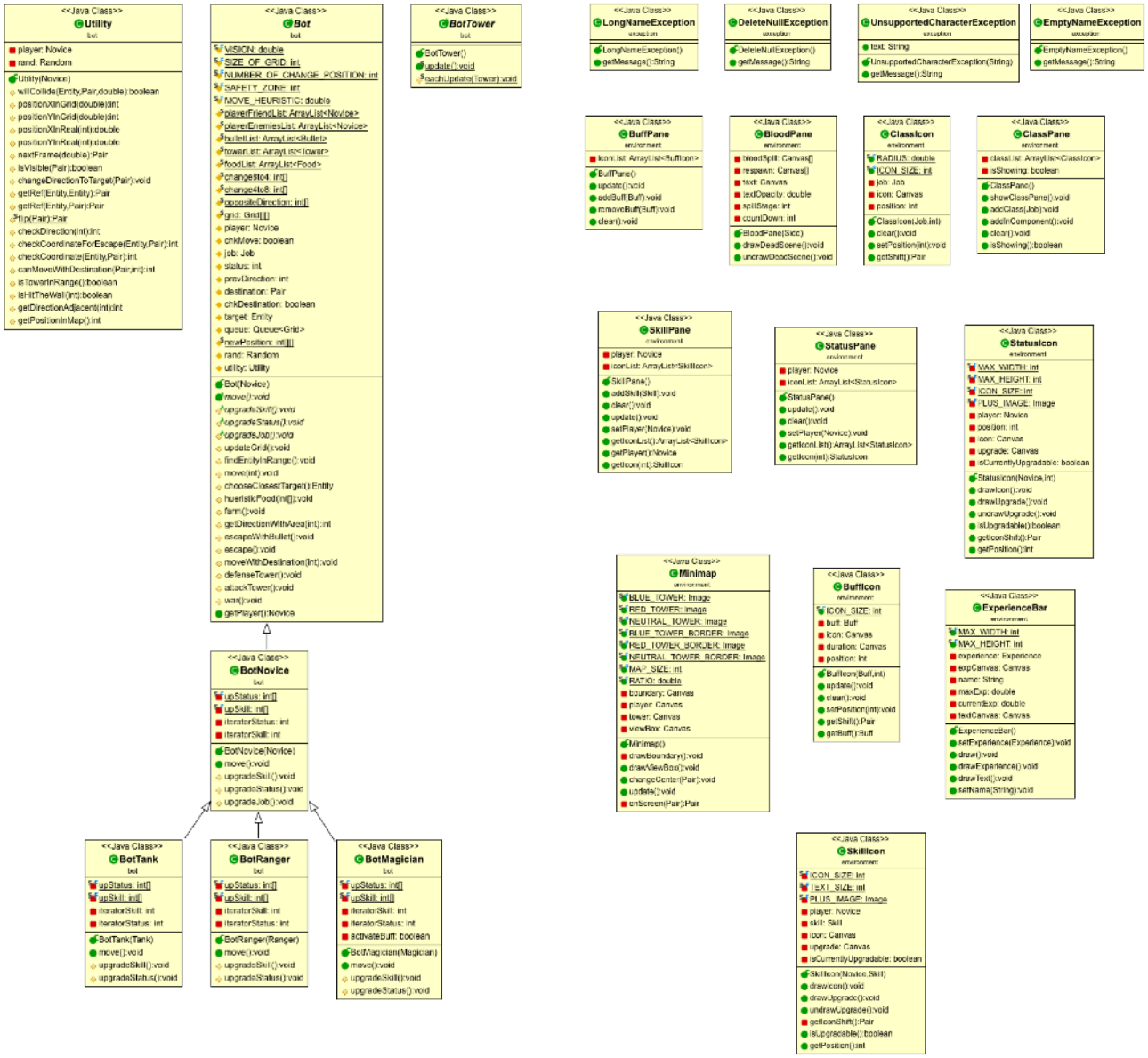
+ double first	Coordinate y in Pane
+ double second	Coordinate x in Pane

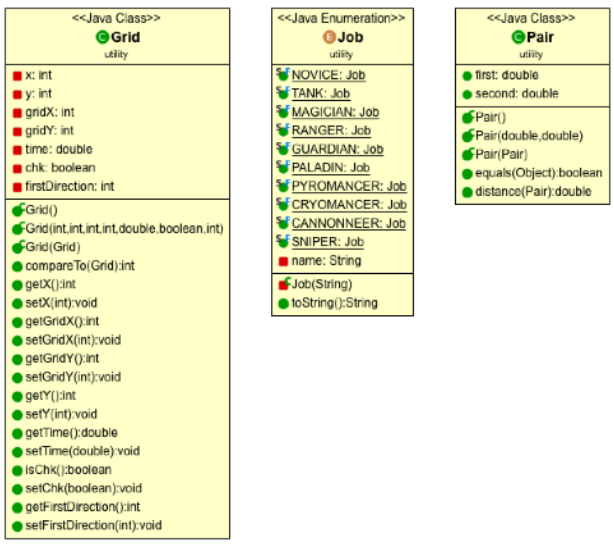
3.16.3.2 Constructor

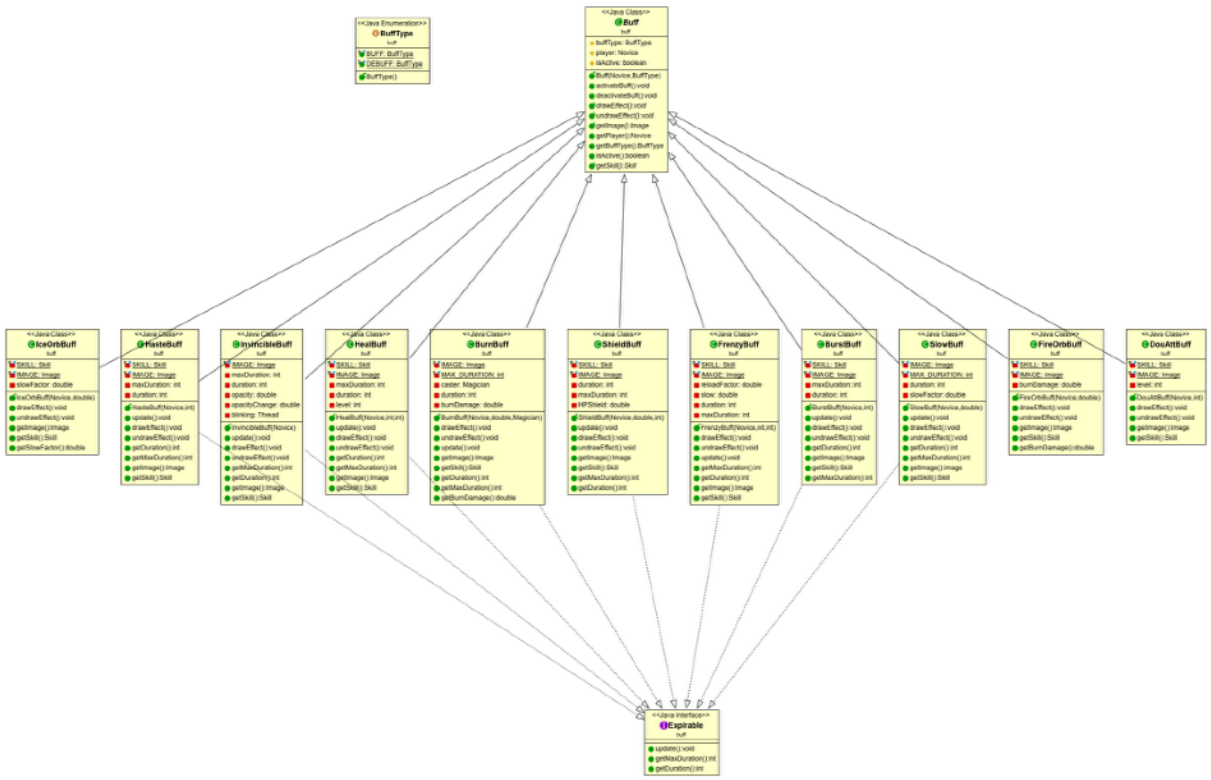
+ Pair(double first, double second)	Set corresponding fields.
+ Pair(Pair tmp)	Copy value from tmp.

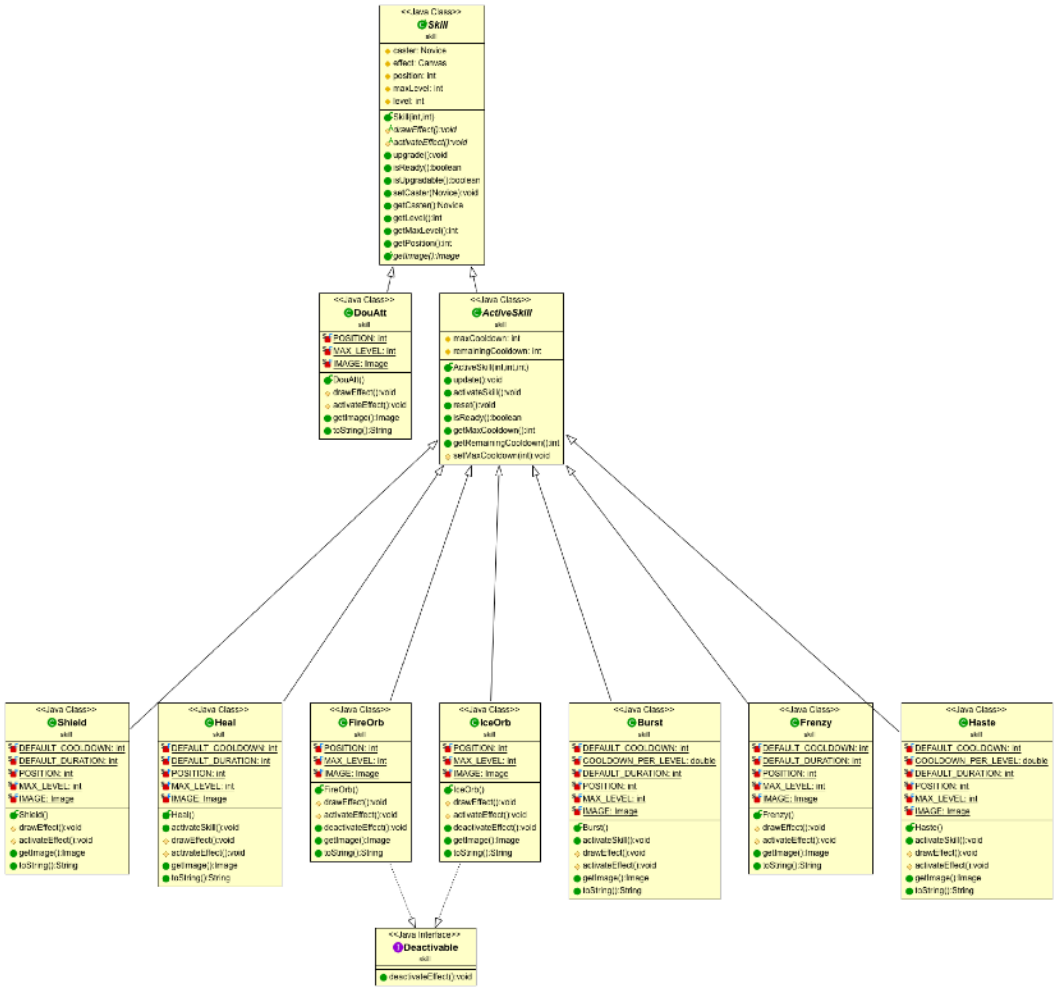
3.16.3.3 Method

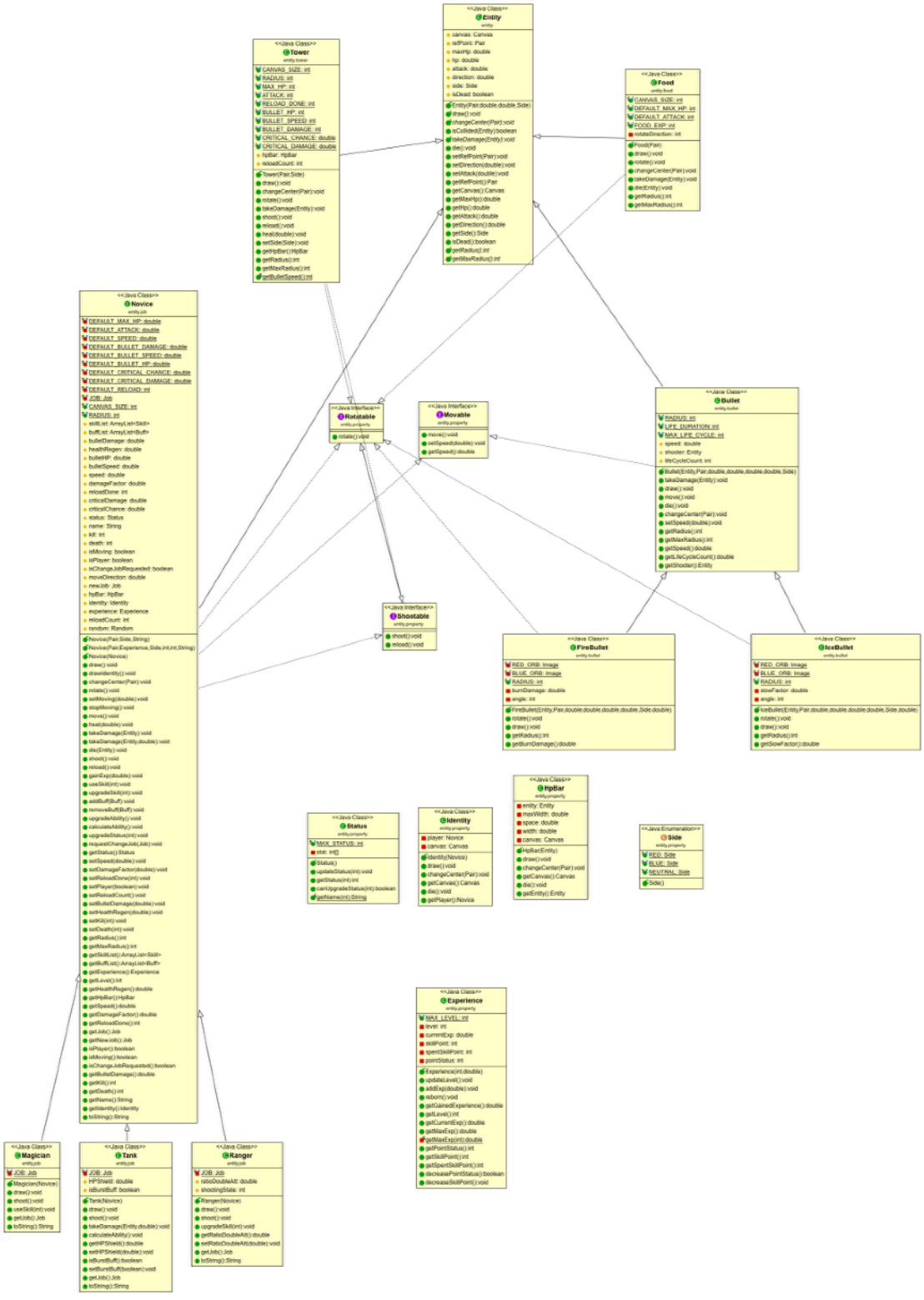
+ boolean equals(Object o)	Return false if o == null. Return false if o is not Pair. Return true if first = o.first and second = o.second. Else Return false.
+ double distance(Pair o)	Return distance between this and o.

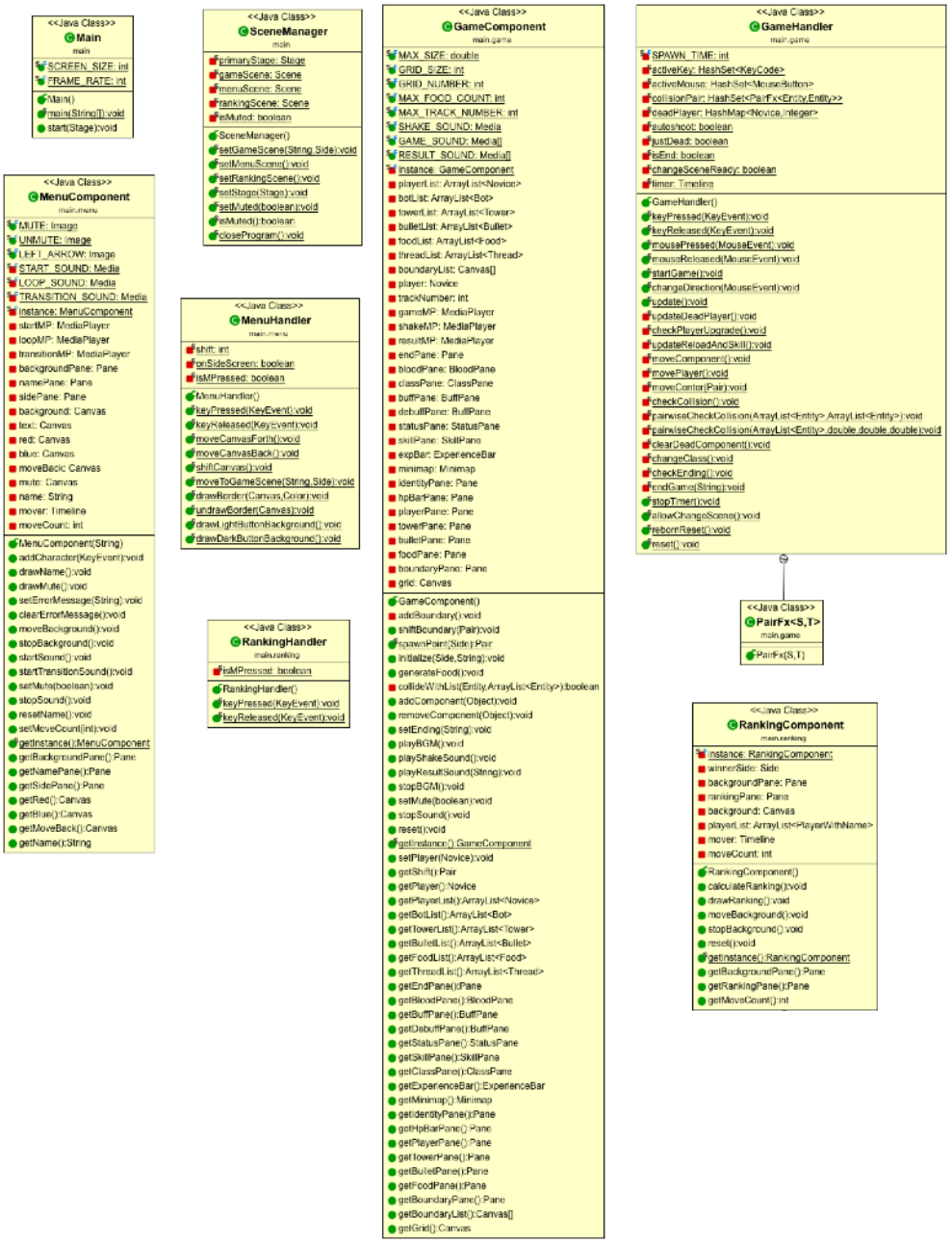

Utility
player: Novice
rand: Random
Bot
VISION: double
SIZE_OF_GRID: int
NUMBER_OF_CHANGE_POSITION: int
SAFETY_ZONE: int
MOVE_HEURISTIC: double
playerFriendList: ArrayList<Novice>
playerEnemiesList: ArrayList<Novice>
bulletList: ArrayList<Bullet>
towerList: ArrayList<Tower>
foodList: ArrayList<Food>
changeBot: int[]
change4to8: int[]
oppositeDirection: int[]
grid: Grid[][]
player: Novice
chkMove: boolean
job: Job
status: int
prevDirection: int
destination: Pair
chkDestination: boolean
target: Entity
queue: Queue<Grid>
nextPosition: int[][]
rand: Random
utility: Utility
Bot(Novice)
move():void
upgradeSkill():void
upgradeStatus():void
upgradeJob():void
updateGrid():void
findEntityInRange():void
move(int):void
chooseClosestTarget():Entity
huristicFood(int[]):void
farm():void
getDirectionWithArea(int):int
escapeWithBullet():void
escape():void
moveWithDestination(Int):void
defenseTower():void
attackTower():void
war():void
getPlayer():Novice
BotTower
BotTower()
update():void
each(Update(Tower):void
LongNameException
LongNameException()
getMessage():String
DeleteNullException
DeleteNullException()
getMessage():String
UnsupportedCharacterException
text: String
UnsupportedCharacterException(String)
getMessage():String
EmptyNameException
EmptyNameException()
getMessage():String
BuffPane
iconList: ArrayList<BuffIcon>
BuffPane()
update():void
addBuff(Buff):void
removeBuff(Buff):void
clear():void
BloodPane
bloodSplit: Canvas[]
respawn: Canvas[]
text: Canvas
textOpacity: double
splitStage: int
countDown: int
BloodPane(Size)
drawDeadScene():void
uncrawDeadScene():void
ClassIcon
RADIUS: double
ICON_SIZE: int
job: Job
icon: Canvas
position: Int
ClassIcon(Job,Int)
clear():void
setPosition(int):void
getShift():Pair
ClassPane
classList: ArrayList<ClassIcon>
isShowing: boolean
ClassPane()
showClassPane():void
addClass(Job):void
addInComponent():void
clear():void
isShowing():boolean
SkillPane
player: Novice
iconList: ArrayList<SkillIcon>
SkillPane()
addSkill(Skill):void
clear():void
update():void
setPlayer(Novice):void
getIconList():ArrayList<SkillIcon>
getPlayer():Novice
getIcon(int):SkillIcon
StatusPane
player: Novice
iconList: ArrayList<StatusIcon>
StatusPane()
update():void
clear():void
set<layer(Novice):void
getIconList() ArrayList<StatusIcon>
getIcon(int):StatusIcon
StatusIcon
MAX_WIDTH: int
MAX_HEIGHT: int
ICON_SIZE: int
PLUS_IMAGE: Image
player: Novice
position: int
icon: Canvas
upgrade: Canvas
isCurrentlyUpgradable: boolean
StatusIcon(Novice,Int)
drawIcon():void
drawUpgrade():void
unCrawUpgrade():void
isUpgradable():boolean
getIconShift():Pair
getPosition():Int
Minimap
BLUE_TOWER: Image
RED_TOWER: Image
NEUTRAL_TOWER: Image
BLUE_TOWER_BORDER: Image
RED_TOWER_BORDER: Image
NEUTRAL_TOWER_BORDER: Image
MAP_SIZE: int
RATIO: double
boundary: Canvas
player: Canvas
tower: Canvas
viewBox: Canvas
Minimap()
drawBoundary():void
drawViewBox():void
changeCenter(Pair):void
update():void
onScreen(Pair):Pair
BuffIcon
ICON_SIZE: int
buff: Buff
icon: Canvas
duration: Canvas
position: Int
BuffIcon(Buff,Int)
update():void
clear():void
setPosition(Int):void
getShift():Pair
getBuff():Buff
ExperienceBar
MAX_WIDTH: int
MAX_HEIGHT: int
experience: Experience
expCanvas: Canvas
name: String
maxExp: double
currentExp: double
textCanvas: Canvas
ExperienceBar()
setExperience(Experience):void
draw():void
drawExperience():void
drawText():void
setName(String):void
BotNovice
upStatus: int[]
upSkill: int[]
iteratorStatus: int
iteratorSkill: int
BotNovice(Novice)
move():void
upgradeSkill():void
upgradeStatus():void
upgradeJob():void
BotTank
upStatus: int[]
upSkill: int[]
iteratorSkill: int
iteratorStatus: int
BotTank(Tank)
move():void
upgradeSkill():void
upgradeStatus():void
BotRanger
upStatus: int[]
upSkill: int[]
iteratorSkill: int
iteratorStatus: int
BotRanger(Ranger)
move():void
upgradeSkill():void
upgradeStatus():void
BotMagician
upStatus: int[]
upSkill: int[]
iteratorSkill: int
iteratorStatus: int
activateBuff: boolean
BotMagician(Magician)
move():void
upgradeSkill():void
upgradeStatus():void
SkillIcon
ICON_SIZE: int
TEXT_SIZE: int
PLUS_IMAGE: Image
player: Novice
skill: Skill
icon: Canvas
upgrade: Canvas
isCurrentlyUpgradable: boolean
SkillIcon(Novice,Skill)
drawIcon():void
drawUpgrade():void
undrawUpgrade():void
getIconShift():Pair
isUpgradable():boolean
getPosition():int


Grid
x: int
y: int
gridX: int
gridY: int
time: double
chk: boolean
firstDirection: int
Grid()
Grid(int,int,int,int,double,boolean,int)
Grid(Grid)
compareTo(Grid):int
getX():int
setX(int):void
getGridX():int
setGridX(int):void
getGridY():int
setGridY(int):void
getY():int
setY(int):void
getTime():double
setTime(double):void
isChk():boolean
setChk(boolean):void
getFirstDirection():int
setFirstDirection(int):void
Job
NOVICE: Job
TANK: Job
MAGICIAN: Job
RANGER: Job
GUARDIAN: Job
PALADIN: Job
PYROMANCER: Job
CRYOMANCER: Job
CANNONNEER: Job
SNIPER: Job
name: String
Job(String)
toString():String
Pair
first: double
second: double
Pair()
Pair(double,double)
Pair(Pair)
equals(Object):boolean
distance(Pair):double


BuffType
Buff
IceOrbBuff
HasteBuff
InvincibleBuff
HealBuff
BurnBuff
ShieldBuff
FrenzyBuff
BurstBuff
SlowBuff
FireOrbBuff
DouAttBuff
Expirable


Skill
DouAtt
ActiveSkill
Shield
Heal
FireOrb
IceOrb
Burst
Frenzy
Haste
Deactivable


Entity
Tower
Food
Novice
Bullet
Rotatable
Movable
Shootable
FireBullet
IceBullet
HpBar
Status
Identity
Experience
Side
Magician
Tank
Ranger


<<Java Class>>
Main
main
SCREEN_SIZE: int
FRAME_RATE: int
Main()
main(String[]):void
start(Stage):void
<<Java Class>>
SceneManager
main
primaryStage: Stage
gameScene: Scene
menuScene: Scene
rankingScene: Scene
isMuted: boolean
SceneManager()
setGameScene(String,Side):void
setMenuScene():void
setRankingScene():void
setStage(Stage):void
setMuted(boolean):void
isMuted():boolean
closeProgram():void
<<Java Class>>
GameComponent
main.game
MAX_SIZE: double
GRID_SIZE: int
GRID_NUMBER: int
MAX_FOOD_COUNT: int
MAX_TRACK_NUMBER: int
SHAKE_SOUND: Media
GAME_SOUND: Media[]
RESULT_SOUND: Media[]
instance: GameComponent
playerList: ArrayList<Novice>
botList: ArrayList<Bot>
towerList: ArrayList<Tower>
bulletList: ArrayList<Bullet>
foodList: ArrayList<Food>
threadList: ArrayList<Thread>
boundaryList: Canvas[]
player: Novice
trackNumber: int
gameMP: MediaPlayer
shakeMP: MediaPlayer
resultMP: MediaPlayer
endPane: Pane
bloodPane: BloodPane
classPane: ClassPane
buffPane: BuffPane
debuffPane: BuffPane
statusPane: StatusPane
skillPane: SkillPane
expBar: ExperienceBar
minimap: Minimap
identityPane: Pane
hpBarPane: Pane
playerPane: Pane
towerPane: Pane
bulletPane: Pane
foodPane: Pane
boundaryPane: Pane
grid: Canvas
GameComponent()
addBoundary():void
shiftBoundary(Pair):void
spawnPoint(Side):Pair
initialize(Side,String):void
generateFood():void
collideWithList(Entity,ArrayList<Entity>):boolean
addComponent(Object):void
removeComponent(Object):void
setEnding(String):void
playBGM():void
playShakeSound():void
playResultSound(String):void
stopBGM():void
setMute(boolean):void
stopSound():void
reset():void
getInstance():GameComponent
setPlayer(Novice):void
getShift():Pair
getPlayer():Novice
getPlayerList():ArrayList<Novice>
getBotList():ArrayList<Bot>
getTowerList():ArrayList<Tower>
getBulletList():ArrayList<Bullet>
getFoodList():ArrayList<Food>
getThreadList():ArrayList<Thread>
getEndPane():Pane
getBloodPane():BloodPane
getBuffPane():BuffPane
getDebuffPane():BuffPane
getStatusPane():StatusPane
getSkillPane():SkillPane
getClassPane():ClassPane
getExperienceBar():ExperienceBar
getMinimap():Minimap
getIdentityPane():Pane
getHpBarPane():Pane
getPlayerPane():Pane
getTowerPane():Pane
getBulletPane():Pane
getFoodPane():Pane
getBoundaryPane():Pane
getBoundaryList():Canvas[]
getGrid():Canvas
<<Java Class>>
GameHandler
main.game
SPAWN_TIME: int
activeKey: HashSet<KeyCode>
activeMouse: HashSet<MouseButton>
collisionPair: HashSet<PairFx<Entity,Entity>>
deadPlayer: HashMap<Novice,Integer>
autoshoot: boolean
justDead: boolean
isEnd: boolean
changeSceneReady: boolean
timer: Timeline
GameHandler()
keyPressed(KeyEvent):void
keyReleased(KeyEvent):void
mousePressed(MouseEvent):void
mouseReleased(MouseEvent):void
startGame():void
changeDirection(MouseEvent):void
update():void
updateDeadPlayer():void
checkPlayerUpgrade():void
updateReloadAndSkill():void
moveComponent():void
movePlayer():void
moveCenter(Pair):void
checkCollision():void
pairwiseCheckCollision(ArrayList<Entity>,ArrayList<Entity>):void
pairwiseCheckCollision(ArrayList<Entity>,double,double,double):void
clearDeadComponent():void
changeClass():void
checkEnding():void
endGame(String):void
stopTimer():void
allowChangeScene():void
rebornReset():void
reset():void
<<Java Class>>
MenuComponent
main.menu
MUTE: Image
UNMUTE: Image
LEFT_ARROW: Image
START_SOUND: Media
LOOP_SOUND: Media
TRANSITION_SOUND: Media
instance: MenuComponent
startMP: MediaPlayer
loopMP: MediaPlayer
transitionMP: MediaPlayer
backgroundPane: Pane
namePane: Pane
sidePane: Pane
background: Canvas
text: Canvas
red: Canvas
blue: Canvas
moveBack: Canvas
mute: Canvas
name: String
mover: Timeline
moveCount: int
MenuComponent(String)
addCharacter(KeyEvent):void
drawName():void
drawMute():void
setErrorMessage(String):void
clearErrorMessage():void
moveBackground():void
stopBackground():void
startSound():void
startTransitionSound():void
setMute(boolean):void
stopSound():void
resetName():void
setMoveCount(int):void
getInstance():MenuComponent
getBackgroundPane():Pane
getNamePane():Pane
getSidePane():Pane
getRed():Canvas
getBlue():Canvas
getMoveBack():Canvas
getName():String
<<Java Class>>
MenuHandler
main.menu
shift: int
isOnSideScreen: boolean
isMPressed: boolean
MenuHandler()
keyPressed(KeyEvent):void
keyReleased(KeyEvent):void
moveCanvasForth():void
moveCanvasBack():void
shiftCanvas():void
moveToGameScene(String,Side):void
drawBorder(Canvas,Color):void
undrawBorder(Canvas):void
drawLightButtonBackground():void
drawDarkButtonBackground():void
<<Java Class>>
RankingHandler
main.ranking
isMPressed: boolean
RankingHandler()
keyPressed(KeyEvent):void
keyReleased(KeyEvent):void
<<Java Class>>
PairFx<S,T>
main.game
PairFx(S,T)
<<Java Class>>
RankingComponent
main.ranking
instance: RankingComponent
winnerSide: Side
backgroundPane: Pane
rankingPane: Pane
background: Canvas
playerList: ArrayList<PlayerWithName>
mover: Timeline
moveCount: int
RankingComponent()
calculateRanking():void
drawRanking():void
moveBackground():void
stopBackground():void
reset():void
getInstance():RankingComponent
getBackgroundPane():Pane
getRankingPane():Pane
getMoveCount():int