Defends of the Crystal

Project A part of 2110215 Programming Methodology Semester 1/2017

Presented by Group JUMP_AND_FRIENDS1

Chalat Limpongsawatt 5931012921
Chayaphol Panjatewakupt 5931011221

Department of Computer Engineering, Faculty of Engineering Chulalongkorn University

Introduction

This game is rpg game that base on game when we young. It has “level system”, “class” and also “skill”

Game’s Lore

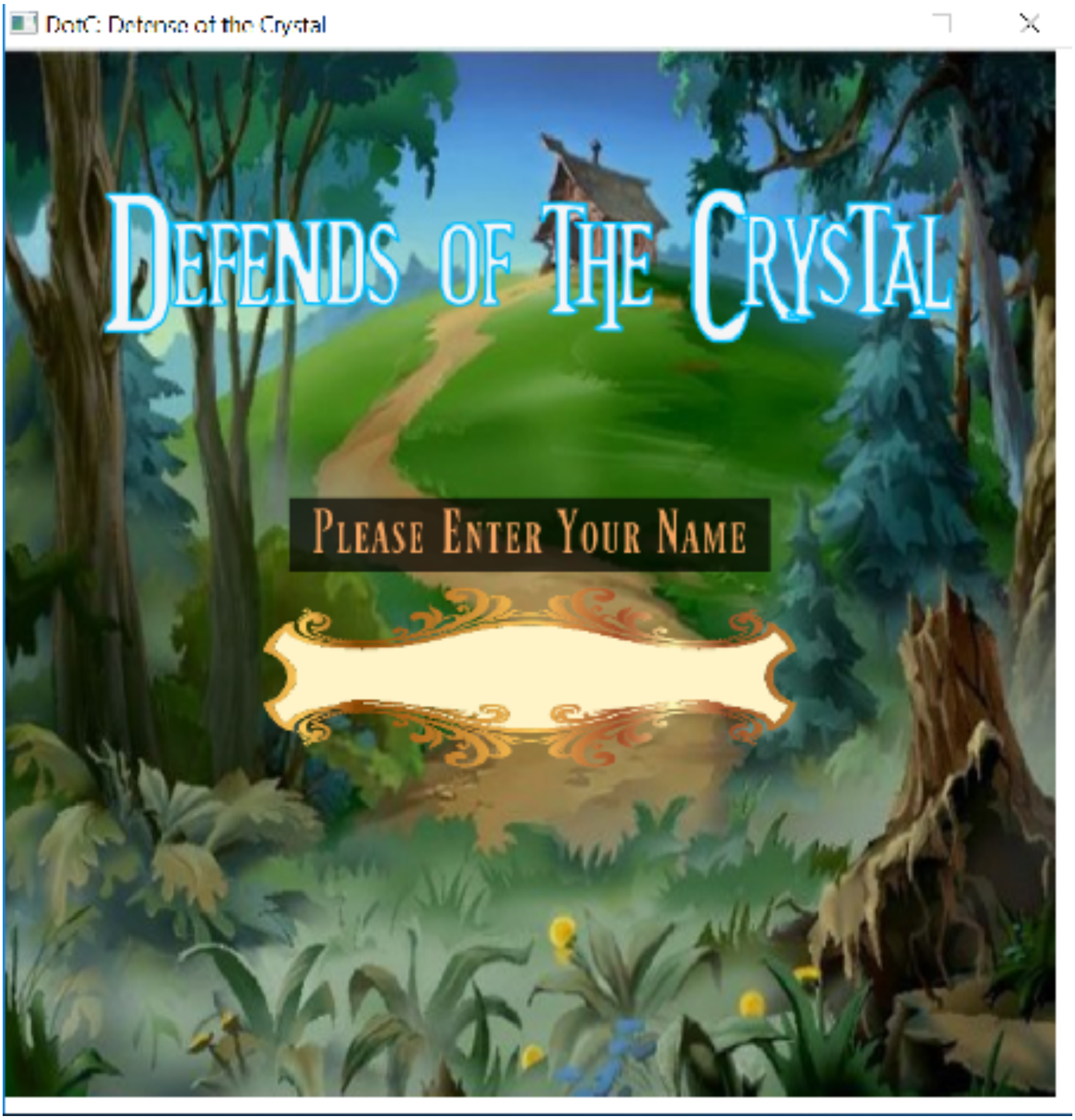
(please read this while listening to Main Menu’s BGM)

Long ago, there were the world where monsters and human live in harmony. The Crystal of the World, giving power of life to beings, lies in the far away mountain that no one could reach. Slimes, Wolves, Goblins, Human, or even Dragons, do not have to hunt for each other’s. Elves protected forests. Wolves protected mountains. Goblins protected caves. Dragons were the ones who conquer all the sky. Human lives peacefully in the village, caring for each other.

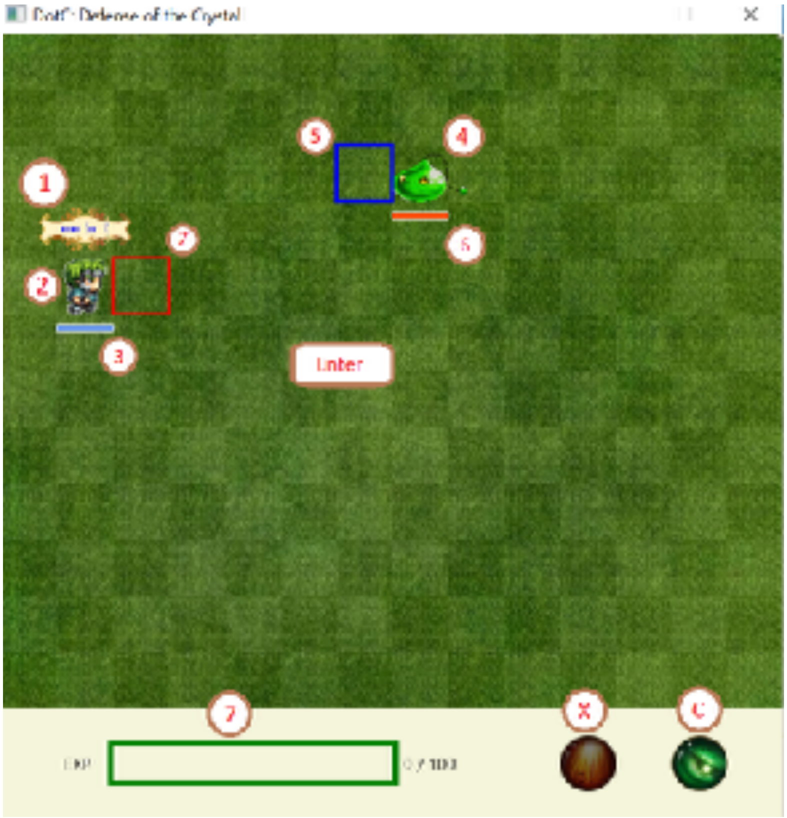
One day. . . The unknown creature, whom race or name was unknown, destroyed the human village. Humans were in fear, knowing that other races were attacked, too. This unknown creature poisons other creatures, making slimes and the others cursed. This curse made them become fiercer, as if it neutralized the power of the Crystal of Life. Most creatures in the land were cursed by the poison.

But there was still one hope. Fragment of the Crystal, held by the Heroes in the Human Village, could repel the poison away. Then the heroes began their adventures to help creatures in the land. Go forward, adventurers, your adventure awaits. Become stronger when fighting monsters and all you have to do is respawn them at their birthplace, so it is not killing.

2. Gameplay



When you open the game, you may face three different background. You may want to listen to the music, so make sure the speaker is turned on. In the main menu, you enter the name of your hero. Character A-Z, a-z, 0-9 are allowed. You must have at least one character in your name. After you name your hero, press ENTER, and you will enter Stage 1.



In the Stage

1. Your hero name and Level
2. Your Hero's Sprite
3. Your HP Bar
4. Monsters' sprites
5. Monsters' attack area
6. Monsters' HP Bar
7. Your EXP Bar

Key Command

↑ or **W** - walk up

← or **A** - walk left

↓ or **S** - walk down

→ or **D** - walk right

Z - Normal Attack

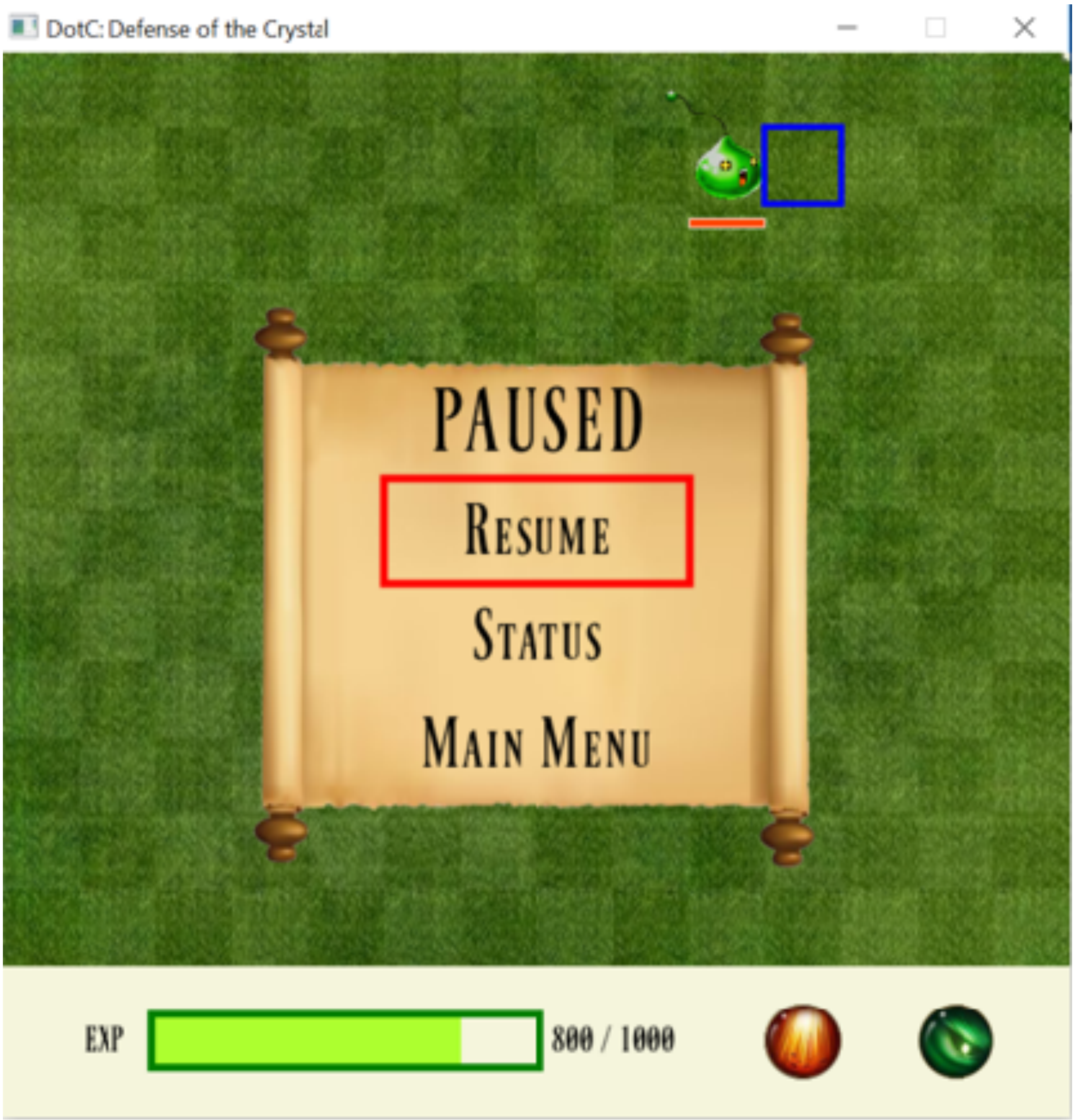
X - special Attack: Ground Smash*

C - Self Recovery

Enter - Open Status Menu
*Ground Smash will be available after **Class Change**

Basic Gameplay:

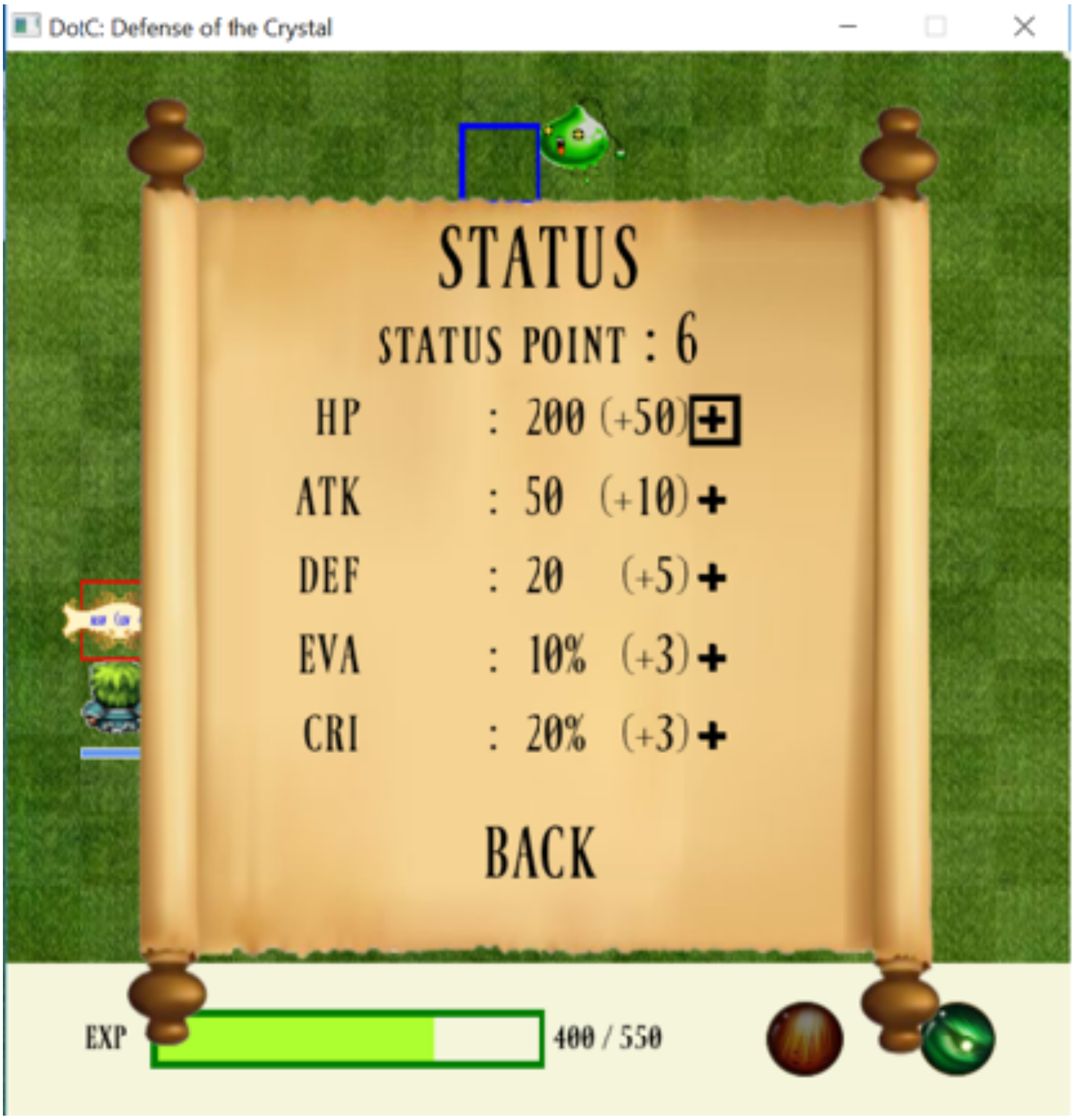
You will start as a Novice first. Press Z to attack monsters until they are exhausted. Proceed to the next Stage. Press ENTER and upgrade your statuses. Press C if your HP is low. If your level is high enough, your class will automatically be changed to Fighter. As a Fighter, you learned a new skill: Ground Smash. Press X to clear mobs of monsters around you quickly.



Pause Menu

You can press **Z** or **ENTER** to choose menu

1. Resume – continue to play the game
2. Status – upgrade your hero statuses
3. Main Menu – return to Main Menu. Noted that the game will be RESET after you go back to Main Menu, the process of your hero will be LOST. Check carefully before you press this button.



STATUS

STATUS POINT : 6

HP	: 200	(+50)
ATK	: 50	(+10)
DEF	: 20	(+5)
EVA	: 10%	(+3)
CRI	: 20%	(+3)

400 / 550

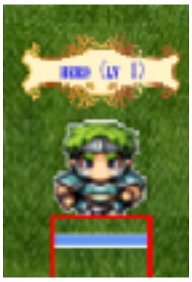
Status Menu

You can press **Z** or **ENTER** to choose which status you will be upgrading.

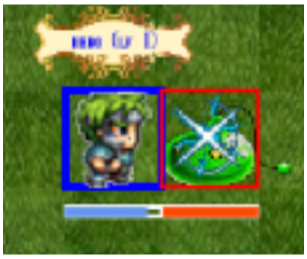
1. STATUS POINT – points that you can pay for upgrading your hero. Obtained when you level up.
2. HP – Max Health Point of your Character. Default: 200. Increased by 10 every 1 status point spent. When your class changes, Max HP is automatically increased by 200.
3. ATK – damage that will be dealt to monster per one time. Default: 50. Increased by 10 every 1 status point spent. When your class changes, ATK is automatically increased by 20.
4. DEF – Defense, the amount of enemy's damage that dealt to you will be cut off. Default: 20. Increased by 5 every 1 status point spent. When your class changes, DEF is automatically increased by 15.

5. EVA – Evasion, the chance that enemy's attack will deal no damage to you. Default: 10%. Increased by 3% every 1 status point spent. When your class changes, EVA is automatically increased by 15%.

6. CRI – Critical, the chance that each of your attack will deal DOUBLE damage. Default: 20%. Increased by 3% every 1 status point spent. When your class changes, CRI is automatically increased by 10%.

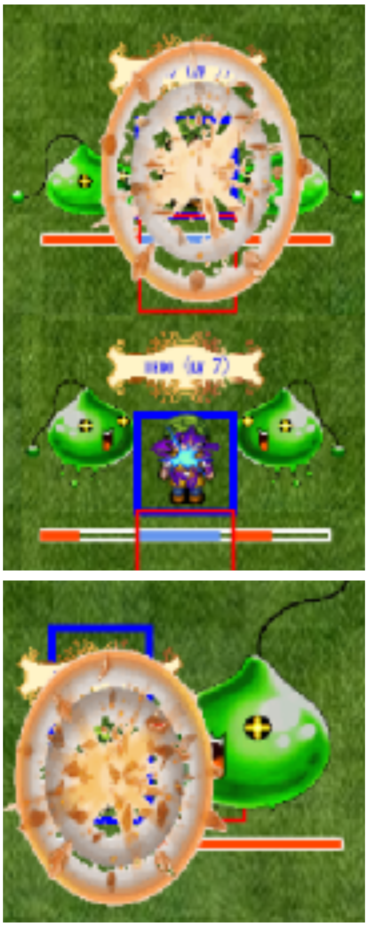


PASSIVE: Auto Recovery – Regenerates own HP every 5 seconds.

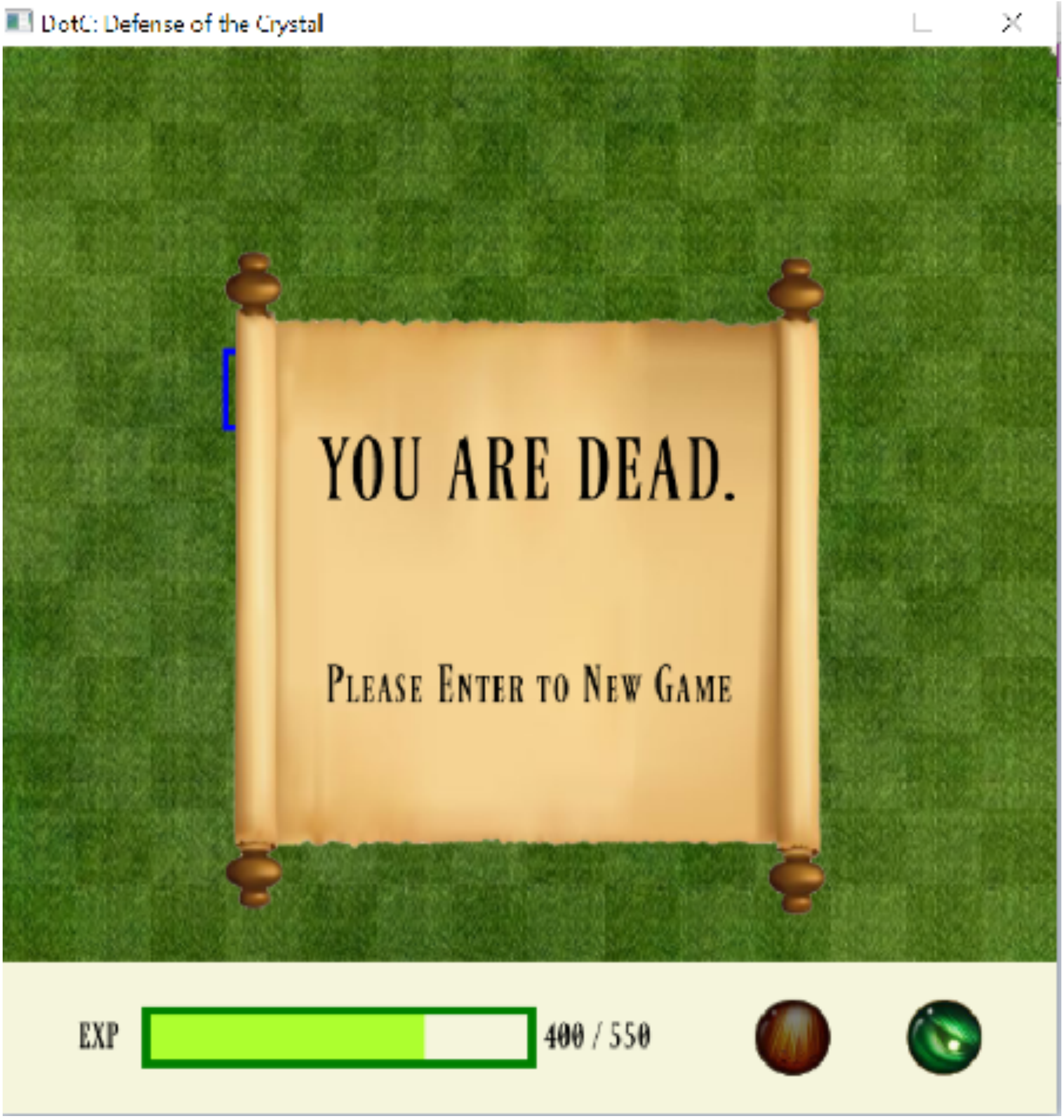


Press Z : Normal Attack - deals damage to enemy. A simple attack that you will definitely use it. This attack can be evaded or critical.

Skill C: Heal – Regenerates HP quickly. After this skill is Triggered, it is on COOLDOWN for a while.



Skill X: Ground Smash:
Available after class change.
Attacks 8 tiles around the character. You can clear a mob of monster very fast this way. Noted that every single attack can be a critical attack. Big monsters like Sublime King will take <u>double</u> damage if attacked closely, and it will take <u>four times</u> damage if both of the attack is critical. A very useful skill to fight mob and big monsters. After this skill is triggered, it is on cooldown for a while.



DotC: Defense of the Crystal

YOU ARE DEAD.

Please Enter to New Game

If you die in this game, you will go back to Main Menu. The process will be RESET.

Class Change:

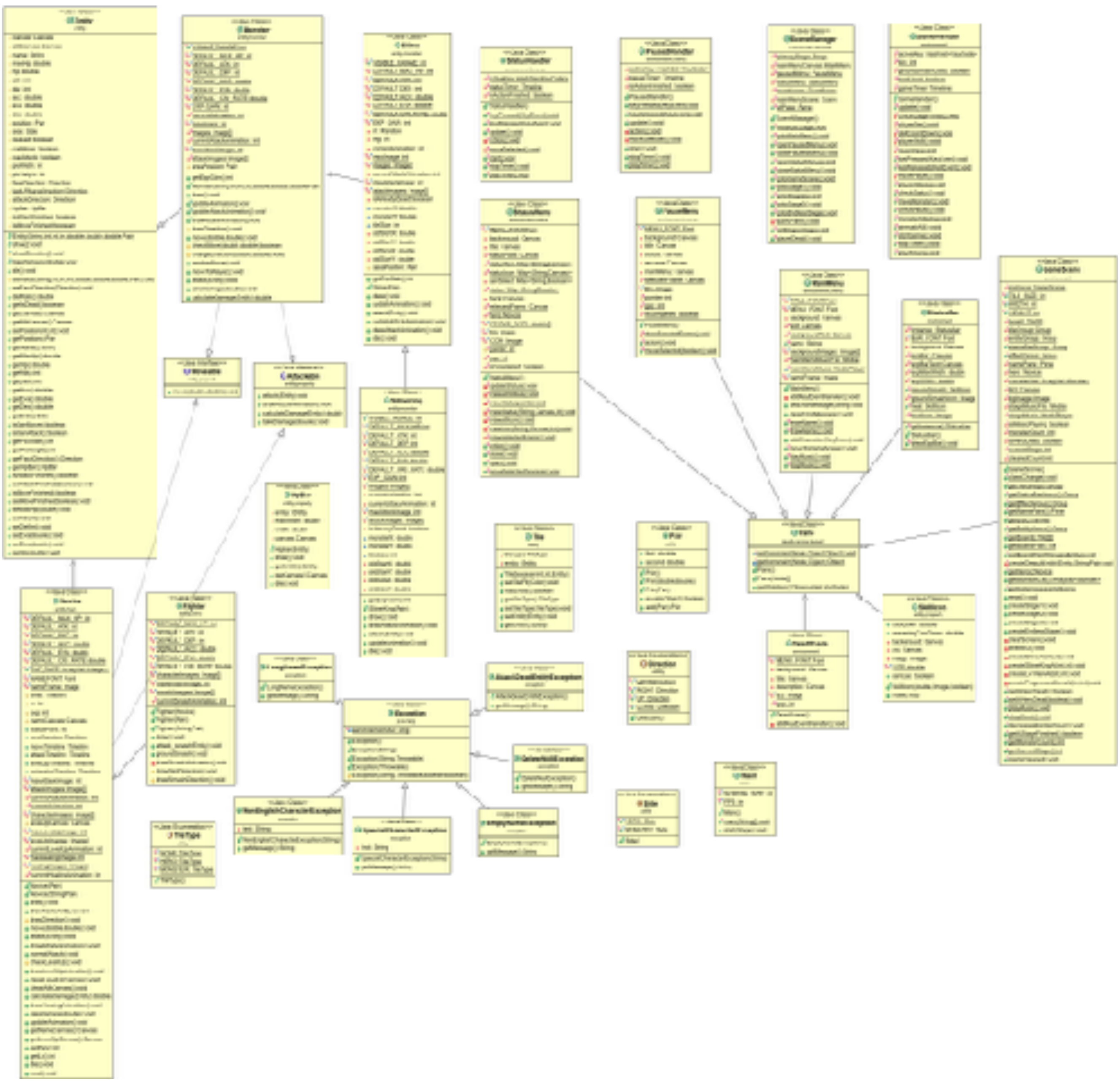
After you reach some level, your class will automatically be changed. Skill X: Ground Smash will be available to use. Your hero sprite is more good-looking, too.

Sublime:

Gulpy Cheerful Little Active Monster that looks like a green drop of water, with a tail stickled on its head. The body is made of water and its skin is slippery. Normally it would be a harmless monster, bouncing in a cute manner. But with the poison, it gets aggressive to human that is nearby.

Sublime King:

Exactly same as previous sublime that is mentioned before, but WAY BIGGER. They are slimes that eats mum mum.. yes, too much. Its size is larger than other sublimes. Mum mum.. Sometimes they can not stop eating. Its skin is very tough and its strength is monsterous, may be because it is a monster by the way? Anyway, defeat it and don’t get mum mummed, okay?



UML Diagram

Implementation Detail

1 Package entity

1.1 Class Entity

1.1.1 Field

# Canvas canvas	used for drawing sprite
# Canvas atkCanvas	used for drawing attack animation
# String name	name of entity
# double maxHp	max hp, hp will not go further than maxHp
# double hp	hp, when hp reaches zero, entity die
# int atk	damage that entity can deal from normal attack
# int def	negative adjustment that applied when taking damage from enemy
# double acc	chance that attack will be succeed.
# double eva	negative adjustment that applied when calculating enemy attack chance
# double dex	chance that attack damage will be doubled(or critical attack)
# Pair position	position of entity
# Side side	Side of entity. Separated to MONSTER, HERO
# boolean isDead	entity is dead or not
# boolean canMove	entity can move or not
# boolean canAttack	entity can attack or not
# int picWidth	number of x tiles that this entity takes
# int picHeight	number of y tiles that this entity takes
# Direction faceDirection	direction that entity is facing to. Up, down, left, or right
# Direction lastLRFaceDirection	last left or right direction that entity faced
# Direction attackDirection	direction that entity is attacking. Up, down, left, or right
# HpBar hpBar	hp bar
# boolean isAttackFinished	entity's attack is finished or not
# boolean isMoveFinished	entity's move is finished or not

1.1.2 Constructor

+ Entity(String name, int maxHp, int attack, int defense, double accuracy, double evasion, double criticalRate,Pair position)	setValue of entity's corresponding field

1.1.3 Methods

+ double calculateDamage(Entity entity)	calculate damage that will be dealt to enemy entity. If attack is miss, deal zero damage. If attack is less than enemy defense, deals 1 damage. If critical succeed, deals double damage
+ void draw()	use canvas to draw sprite
# void drawDirection()	draw possible grid that can be attacked from this entity
+ void move(double moveX, double moveY)	move self position
+ void attack(Entity entity) throws AttackDeadEntityException	attack opponent
+ void takeDamage(double dmg)	decrease self hp
+ void die()	perform effects from dying
# void setValue(String name, int maxHp, int attack, int defense, double accuracy, double evasion, double dexterous, Pair position)	set value of corresponding field input and also initialize canvas, atkCanvas, set canAttack, canMove true, and isDead false
+ void setFaceDirection(Direction Direction)	set self direction
+ double defRate()	defense rate in %, calculated from (def/def+100)
+ boolean getIsDead()	getter of isDead
+ Canvas getCanvas()	getter of canvas
+ Canvas getAtkCanvas()	getter of atkCanvas
+ void setPosition(int x, int y)	set position of entity to tile (x,y)
+ Pair getPosition()	getter of position
+ String getName()	getter of name
+ double getMaxHp()	getter of maxHp
+ double getHp()	getter of hp
+ int getAtk()	getter of atk
+ int getDef()	getter of def
+ double getAcc()	getter of accuracy
+ double getEva()	getter of evasion
+ double getDex()	getter of dexterity
+ Side getSide()	getter of side
+ boolean isCanMove()	getter of canMove
+ boolean isCanAttack()	getter of canAttack
+ int getPicWidth()	getter of picWidth
+ int getPicHeight()	getter of picHeight
+ Direction getFaceDirection()	getter of faceDirection
+ HpBar getHpBar()	getter of hpBar
+ boolean isAttackFinished()	getter of isAttackFinished

+ void setAttackFinished(boolean isAttackFinished)	setter of isAttackFinished
+ boolean isMoveFinished()	getter of isMoveFinished
+ void setMoveFinished(boolean isMoveFinished)	setter of isMoveFinished
+ int getExpGain()	getter of expGain
+ void setMaxHp(double maxhp)	setter of canAttack
+ void setAtk(int atk)	setter of atk
+ void setDef(int def)	setter of def
+ void setEva(double eva)	setter of eva
+ void setDex(double dex)	setter of dex
+ void setHp(double hp)	setter of hp

1.2 package entity.hero

1.2.1 Class Novice extends entity

1.2.1.1 Field

- int DEFAULT_MAX_HP	default hp that will be set to level 1 hero
- int DEFAULT_ATK	default atk that will be set to level 1 hero
- int DEFAULT_DEF	default def that will be set to level 1 hero
- double DEFAULT_ACC	default acc that will be set to level 1 hero
- double DEFAULT_EVA	default eva that will be set to level 1 hero
- double DEFAULT_CRI_RATE	default dex that will be set to level 1 hero
+ ArrayList<Integer> EXP_RATE	exp needed for leveling up for each level, max level is 100.
- Font NAMEFONT	font used to draw name
- Image nameFrame	image of name frame
# Timeline timer	timeline of hero
# int lv	level of hero
# int exp	exp of hero
# Canvas nameCanvas	canvas of hero
+ int statusPoint	status point of hero
+ Timeline healTimeline	heal timeline of hero
+ Timeline moveTimeline	move timeline of hero
+ Timeline attackTimeline	attack timeline of hero
+ Timeline levelUpTimeline	level up timeline of hero
+ Timeline animationTimeline	animation timeline of hero
- int MAXATTACKIMAGE	number of attack image of hero stored in resource folder
- Image[] ATTACKIMAGES	image of attack image of hero
- int currentAttackAnimation	current attack animation of hero
- int currentAnimation	current animation of hero

- Image[] characterImages	character image of hero
# Canvas levelUpCanvas	level up canvas of hero
- int MAXLEVELUPIMAGE	number of level up image of hero stored in resource folder
- Image[] LEVELUPIMAGES	image of level up animation
- int currentLevelUpAnimation	current count of level up animation

1.2.1.2 Constructor

+ Novice(Pair pos)	set fields equal to DEFAULT and place position
+ Novice(String name, Pair pos)	set novice's name, set fields equal to DEFAULT and place position

1.2.1.3 Method

+ void attack(Entity entity)	attacks chosen entity, sets enemy entity can not move or attack until this attack is finished, wait until attack animation finish and deals damage
# void checkLevelUp()	if exp is over exp rate of current level,
+ void cleanAtkCanvas()	clear attack canvas
+ void cleanLevelUpCanvas()	clear level up canvas
+ void die()	delete canvas and nameCanvas from entityGroup, set is dead true, set hp to zero, hp bar die
+ void draw()	draw character image due to faceDirection and also drawDirection and if entity is not dead, redraw hpBar and drawNameAndLv
+ void drawAttackAnimation()	draw attack animation
# void drawDirection()	draw area of attack direction due to faceDirection
+ void drawHealingAnimation()	draw healing animation
+ void drawLevelUpAnimation()	draw level up animation using leveluptimeline
+ void drawNameAndLv()	draw name using nameCanvas and draw image frame
+ int getExp()	getter of exp
+ Canvas getLevelUpCanvas()	getter of levelupCanvas
+ int getLv()	getter of lv
+ Canvas getNameCanvas()	getter of nameCanvas
+ void heal()	increase hp by 5% of maxHp every 0.8 seconds 5 times, draw healing animation
+ void move(double moveX , double moveY)	move x and y tiles, set isMoveFinished to false until moveTimeline is on finished
+ void normalAttack()	checks that area which is attacking has monster, then attack
+ void takeDamage(double)	decrease hp from input, if hp is lower than zero, die
+ void updateAnimation()	increase currentAttackAnimation, drawAttackAnimation

1.2.2 Class Fighter extends Novice

1.2.2.1 Field

- int DEFAULT_ATK	default atk that will be set to default Fighter
- int DEFAULT_DEF	default def that will be set to default Fighter
- double DEFAULT_ACC	default acc that will be set to default Fighter
- double DEFAULT_EVA	default eva that will be set to default Fighter
- double DEFAULT_CRI_RATE	default dex that will be set to default Fighter
- int CHARACTERIMAGES	character image of Fighter
- Image[] MAXSMASHIMAGES	image of smash image of hero
- int currentSmashAnimation	count of current smash animation

1.2.2.2 Constructor

+ Fighter(Novice novice)	create fighter from previous novice, statuses are kept
+ Fighter(Pair pos)	create lv1 fighter
+ Fighter(String name, Pair pos)	create lv1 fighter with name

1.2.2.3 Method

+ void draw()	draw character sprite by using canvas
+ void attack_smash(Entity entity)	attacks monster by smashing the ground
+ void groundSmash()	checks surrounding tiles, attack all monster around the Fighter
- void drawSmashAnimation()	draw smash animation
# void drawSmashDirection	draw a square around the area that ground smash can attack

1.3 package entity.monster

1.3.1 Class Monster extends Entity implements Attackable, Moveable

+ int VISIBLE_RANGE	range that monster can see player
- int DEFAULT_MAX_HP	max hp that will be set to default monster
- int DEFAULT_ATK	atk that will be set to default monster
- int DEFAULT_DEF	def that will be set to default monster
- double DEFAULT_ACC	acc that will be set to default monster
- double DEFAULT_EVA	eva that will be set to default monster
- double DEFAULT_CRI_RATE	dex that will be set to default monster
- int EXP_GAIN	exp gain that will be set to default monster
# Pair areaPosition	area position of monster

1.3.1.1 Field

+ Monster(String name, int maxhp, int attack, int defense, double accuracy, double evasion,double criticalRate, Pair pos)	constructor of monster that sets corresponding field

1.3.1.2 Constructor

# public void draw()	abstract draw
# public void updateAnimation()	abstract updateAnimation
# public void updateAttackAnimation()	abstract updateAttackAnimation
# public void drawAttackAnimation()	abstract drawAttackAnimation
# void drawDirection()	draw possible attack area
+ void move(double moveX, double moveY)	move x tiles and y tiles
# boolean checkMove(double moveX, double moveY)	monster will seek for hero in its visible range, if found return true
# void changeDirection(double x, double y)	change faceDirection due to input x and y
+ void randomMove()	random move 1 tile horizontally or vertically
+ void moveToPlayer()	move in the direction to the player horizontally or vertically
+ void takeDamage(double dmg)	decrease hp equals to the input, if hp<dmg, die
+ int getExpGain()	getter of expGain

1.3.1.3 Method

1.3.2 Class Slime extends Monster

1.3.2.1 Field

+ int VISIBLE_RANGE	range that slime can see player
- int DEFAULT_MAX_HP	max hp that will be set to default slime
- int DEFAULT_ATK	atk that will be set to default slime
- int DEFAULT_DEF	def that will be set to default slime
- double DEFAULT_ACC	acc that will be set to default slime
- double DEFAULT_EVA	eva that will be set to default slime
- double DEFAULT_CRI_RATE	dex that will be set to default slime
- int EXP_GAIN	exp gain that will be set to default slime
# Pair areaPosition	area position of slime
- Random rn	Random for independent slime animation
- int rng	Random Number Generator
- int currentAnimation	current Slime bouncing animation
- int MAXIMAGE	max number of slime images in resource folder
- Image[] IMAGES	slime images
- int currentAttackAnimation	current attack animation count
- int MAXATTACKIMAGE	max number of attack images in resource folder
- Image[] ATTACKIMAGES	attack images
- int tileSize	convention for code reading

1.3.2.2 Constructor

+ Slime(Pair pos)	create slime with default status at the input position

1.3.2.3 Method

+ void draw()	use canvas to draw slime
+ void updateAnimation()	increase current animation and then draw
+ void attack(Entity entity)	attack selected entity, draw attack animation
+ void updateAttackAnimation()	increase current attack animation and then draw
+ void drawAttackAnimation()	draw attack animation
+ void die()	delete canvas and hp bar from gameScene

1.3.3 Class SlimeKing extends Slime

1.3.3.1 Field

+ int VISIBLE_RANGE	range that Slime King can see player
- int DEFAULT_MAX_HP	max hp that will be set to default Slime King
- int DEFAULT_ATK	atk that will be set to default Slime King
- int DEFAULT_DEF	def that will be set to default Slime King
- double DEFAULT_ACC	acc that will be set to default Slime King
- double DEFAULT_EVA	eva that will be set to default Slime King
- double DEFAULT_CRI_RATE	dex that will be set to default Slime King
- int EXP_GAIN	exp gain that will be set to default Slime King
# Pair areaPosition	area position of Slime King
- int currentAnimation	current Slime King bouncing animation
- int MAXIMAGE	max number of Slime King images in resource folder
- Image[] IMAGES	Slime King images
- int currentAttackAnimation	current attack animation count
- int MAXATTACKIMAGE	max number of attack images in resource folder
- Image[] ATTACKIMAGES	attack images
- int tileSize	convention for code reading

1.3.3.2 Constructor

+ SlimeKing(Pair pos)	create slime king with default status at the input position

1.3.3.3 Method

+ void draw()	use canvas to draw slime
+ void attack(Entity entity)	attack selected entity, draw attack animation
+ void drawAttackAnimation()	draw attack animation

1.4 package entity.property

1.4.1 Interface Attackable

1.4.1.1 Method

+ void attack(Entity entity)	attack enemy entity
+ void drawAttackAnimation()	draw attack animation
+ double calculateDamage(Entity entity)	calculate damage
+ void takeDamage(double dmg)	decrease hp

1.4.2 Class HpBar

1.4.2.1 Field

- Entity entity	Hp Bar owner
- double maxWidth	max Width of Hp Bar
- double width	width of hp bar
- Canvas canvas	canvas used to draw hp bar

1.4.2.2 Constructor

+ HpBar(Entity entity)	create hp bar of input entity

1.4.2.3 Method

+ void draw	draw hp bar
+ Entity getEntity()	get hp bar owner entity
+ Canvas getCanvas()	getter of canvas
+ void die()	delete hp bar from statusBarGroup of GameScene

1.4.3 Interface Moveable

1.4.3.1 Method

+ void move(double moveX, double moveY)	move by moveX and moveY tiles

1.4.4 class SkillIcon extends Pane

1.4.4.1 Field

+ double coolDown	duration until the skill can be used again when it is triggered
+ double remainingCoolDown	duration remaining until the skill can be used again
- Canvas background	background canvas
- Canvas arc	arc canvas
- Image image	image of the icon
- static double SIZE	size of icon
+ boolean canUse	skill can be used or not

1.4.4.2 Constructor

+ SkillIcon(double coolDown,Image image,boolean canUse)	sets cooldown, image and can be used

1.4.43 Method

+ void draw()	draw the icon

2 Package environment

2.1 Class GameScene extends Pane

2.1.1 Field

- GameScene instance	for static object
+ int TILE_SIZE	size of tile
+ int WIDTH	width of map (unit)
+ int HEIGHT	height of map (unit)
- Tile[][] board	for contain entities
- Group tileGroup	Group of tile's canvas
- Group entityGroup	Group of entities' canvas
+ Group statusBarGroup	Group of status bar' canvas
+ Group effectGroup	Group of effect' canvas
- Pane namePane	Pane for draw name
- Novice hero	contain entity hero

- ArrayList<Monster> monsterList	list of monster
- Canvas BG	background canvas
- Image bgImage	background image
- Media stageMusicFile	background music
- MediaPlayer stageMusic	background music player
- boolean isMusicPlaying	flag indicates that music is playing
- boolean isStageFinished	flag indicates that stage is finished
- int monsterCount	amount of monsters
- boolean isHeroDead	flag indicates that hero is dead
- int currentStage	contain stage number
- int clearedCount	number of clear stage

2.1.2 Constructor

+ GameScene ()	Initializes fields, create gamestage

2.1.3 Methods

+ void classChange	to change hero from novice to fighter
- createDefaultEntity(Entity entity, String entityType Pair position)	create entity in default setting
- void reset()	reset to first stage
- void createStage1()	create stage 1
- void createStage2()	create stage 2
- void createStage3()	create stage 3
- void createEndlessStage()	create last stage
- void clearScreen()	clear the stage to empty stage
- void drawBG()	draw background
- void createSlimeAt(int x, int y)	create slime at position x,y
- void createSlimeKingAt(int x, int y)	create slime king at position x,y
- void createLv1HeroAt(int x, int y)	create lv 1 hero at position x,y
- void createProgressedHeroAt(int x, int y)	create current hero at position x,y
+ void playMusic()	play music
+ void stopMusic()	stop music
+ void gameCleared()	game is cleared
+ void decreaseMonsterCount()	decrease monster count
+ void getIsStageFinished()	flag indicates that stage is finished

+ Getter method for tileGroup, statusBarGroup, effectGroup, namePane, BG, entityGroup, hero, monsterList, instance, currentStage	
+ Getter & Setter method for board,isHeroDead	

2.2 Class GameHandler

2.2.1 Field

- HashSet<KeyCode> activeKey	contain keyCode from event
- int tick	timer count
- boolean groundSmashUsed	flag indicate that ground smash is used
- boolean healUsed	flag indicate that heal is used
- Timeline gameTimer	Timeline for game running

2.2.2 Methods

+ void keyPressed()	collect key from event
+ void keyReleased()	remove key from event
+ void update()	update game
- void movePlayer()	player move
+ void playerAttack()	player attack
+ void moveMonster()	monstet move
+ void monsterAttack()	monster attack
+ void checkStatus()	check all entities that die or not
+ void animateAll()	animate all
- void regenHp()	regenerate hero hp
- void playerSkill()	player use skill
- void skillCountDown()	skill cooldown
- void checkStageFinish()	check stage if finish or not
- void playerDie()	check player die or not
+ checkPause()	check game is pause or not
+ void startGame()	start all about game
+ void stopTimer()	stop game timer
+ void playTimer()	play game timer

2.3 Class DeadScene extends Pane

2.3.1 Field

- Font MENU_FONT	Font for this menu
- Canvas background	background canvas
- Canvas title	title canvas
- Canvas description	description canvas
- Image BG	background image
- int gap	gap between any text

2.3.2 Constructor

+ DeadScene ()	Initializes fields, create dead scene

2.3.3 Methods

+ void addkeyEventHandler ()	add event handler for dead scene

2.4 Class StatusBar extends Pane

2.4.1 Field

- Font BAR_FONT	Font for this bar
- Canvas background	background canvas
- Canvas expBar	exp bar canvas
- Canvas expBarText	exp bar text canvas
- double expMaxWidth	max width of exp bar
- double expWidth	current width of exp bar
+ SkillIcon groundSmash	ground smash skill icon
- Image groundSmashIcon	ground smash icon image
+ SkillIcon heal	heal skill icon
- Image healIcon	heal icon image

2.4.2 Constructor

+ StatusBar ()	Initializes fields, create status bar

2.4.3 Methods

+ void drawExpBar ()	draw exp bar

2.5 Package environment.menu

2.5.1 Class MainMenu extends Pane

2.5.1.1 Field

- Font TITLE_FONT	Font for this title
- Font MENU_FONT	Font for this menu
- Canvas background	background canvas
- Canvas text	text canvas
- Canvas backgroundText	background text canvas
+ String name	name of hero
- Image[] backgroundImages	list of background images
- Media mainMenuMusicFile	main menu music file
- MediaPlayer mainMenuMusic	main menu music player
- Image nameFrame	image of name frame

2.5.1.2 Constructor

+ MainMenu ()	Initializes fields, create main menu

2.5.1.3 Methods

+ void addkeyEventHandler ()	add event handler for main menu

2.5.2 Class PauseMenu extends Pane

2.5.2.1 Field

- Font MENU_FONT	Font for this menu
- Canvas background	background canvas
- Canvas title	title canvas
- Canvas status	status canvas
- Canvas resume	resume canvas
- Canvas mainMenu	main menu canvas
- Canvas selectedFrame	selected frame canvas
- Image BG	background image
# int pointer	pointer of selected menu
- int gap	gap between menu
- boolean isCompleted	flag indicate that draw selected menu complete

2.5.2.2 Constructor

+ PauseMenu ()	Initializes fields, create pause menu

2.5.2.3 Methods

- void drawSelectedFrame()	draw selected frame
+ void action()	action for any event
+ void moveSelected()	move selected frame

2.5.3 Class PauseHandler

2.5.3.1 Field

- HashSet<KeyCode> activeKey	contain keyCode from event
- Timeline pauseTimer	Timeline for pause menu
- boolean isActionFinished	flag indicates that action is finished

2.5.3.2 Methods

+ void keyPressed()	collect key from event
+ void keyReleased()	remove key from event
+ void update()	update pause menu
- void action()	action for any keyCode
+ void moveSelected()	move selected frame
+ void start()	start pause menu
+ void stopTimer()	stop pause timer
+ void playTimer()	play pause timer

2.5.4 Class PauseMenu extends Pane

2.5.4.1 Field

- Font MENU_FONT	Font for this menu
- Canvas background	background canvas
- Canvas title	title canvas
- Canvas statusPoint	status point canvas
- Map<String, Canvas> statusText	map of status text (status name is key,canvas is value)
- Map<String, Canvas> statusIcon	map of status text (status name is key,canvas is value)
- Map<String, Boolean> canSelect	map of status text (status name is key,boolean (indicate that this menu can select) is value)
- Map<String, Double> status	map of status text (status name is key,status point is value)
- Canvas back	back menu canvas

- Canvas selectedFrame	selected frame canvas
- Novice hero	contain entity hero
- double[] STATUS_RATE	contain rate to up status
- Image BG	background image
- Image ICON	up status icon
# int pointer	pointer of selected menu
- int gap	gap between menu
- boolean isCompleted	flag indicate that draw selected menu complete

2.5.4.2 Constructor

+ StatusMenu ()	Initializes fields, create pause menu

2.5.4.3 Methods

- void drawSelectedFrame()	draw selected frame
+ void action()	action for any event
+ void moveSelected()	move selected frame
- void updateStatus()	update status
- void drawAllStatus()	draw all status
- void drawStatusPoint()	draw status point
- void drawStatus(String s, Canvas statusCanvas, int i)	draw status s in status canvas in row i
- void drawAllIcon()	draw all icon
- void drawIcon(String s, Canvas statusCanvas, int i)	draw icon of s in icon canvas in row i
+ void open()	open status menu
+ void close()	close status menu

2.5.5 Class StatusHandler

2.5.5.1 Field

- HashSet<KeyCode> activeKey	contain keyCode from event
- Timeline statusTimer	Timeline for status menu
- boolean isActionFinished	flag indicates that action is finished

2.5.5.2 Methods

+ void keyPressed()	collect key from event
+ void keyReleased()	remove key from event
+ void update()	update status menu
- void action()	action for any keyCode

+ void moveSelected()	move selected frame
+ void start()	start pause menu
+ void stopTimer()	stop pause timer
+ void playTimer()	play pause timer

2.6 Package environment.window

2.6.1 Class SceneManager

2.6.1.1 Field

- Stage primaryStage	contain stage
- MainMenu mainMenuCanvas	contain main menu canvas
- PauseMenu pausedMenu	contain pause menu canvas
- StatusMenu statusMenu	contain status menu canvas
- DeadScene deadScene	contain dead scene canvas
- Scene mainMenuScene	contain main scene canvas
+ Pane allPane	contain all pane

2.6.1.2 Methods

+ void initialize(Stage stage)	initial primary stage
+ void gotoMainMenu()	go to main menu
+ void gotoGameScene()	go to game scene
+ void openPausedMenu()	open pause menu
+ void closePausedMenu()	close pause menu
+ void openStatusMenu()	open status menu
+ void closeStatusMenu()	close status menu
+ void gotoStage1()	go to stage 1
+ void gotoStage2()	go to stage 2
+ void gotoStage3()	go to stage 3
+ void gotoEndlessStage()	go to last stage
- void putAllPane()	put all pane
+ void setStage(Stage primaryStage)	set stage
+ void playerDead()	show dead scene

3 Package exception

3.1 Class AttackDeadEntityException

3.1.1 Method

+ String getMessage()	return “Cannot Attack Dead Entity.”

3.2 Class DeleteNullException

3.2.1 Method

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

3.3 Class EmptyNameException

3.3.1 Method

+ String getMessage()	return "The name should not be null."

3.4 Class LongNameException

3.4.1 Method

+ String getMessage()	return "The name should not be longer than 10 characters."

3.5 Class NonEnglishCharacterException

3.5.1 Field

+ String text	contain input text

3.5.2 Constructor

+ NonEnglishCharacterException(String text)	Initializes text

3.5.3 Method

+ String getMessage()	return "Character [" + text + "] is not allowed."

3.6 Class SpecialCharacterException

3.6.1 Field

+ String text	contain input text

3.6.2 Constructor

+ SpecialCharacterException(String text)	Initializes text

3.6.3 Method

+ String getMessage()	return "Character [" + text + "] is not allowed."

4 Package utility

4.1 Enum Direction

4.1.1 Field

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

+ Direction Down	Down face direction.

4.2 Enum Side

4.2.1 Field

+ Side HERO	Hero side
+ Side MONSTER	Monster side

4.3 Enum TileType

4.1.1 Field

+ TileType NONE	tile type none
+ TileType HERO	tile type Hero
+ TileType MONSTER	tile type Monster

4.4 Class Pair

4.4.1 Field

+ double first	first of pair
+ double second	second of pair

4.4.2 Constructor

+ Pair ()	Initializes first=0, second=0
+ Pair(double first, double second)	Initializes all fields
+ Pair(Pair tmp)	Initializes all fields

4.4.3 Methods

+ Pair add(Pair o)	return this pair add other pair
+ boolean equals(Object o)	return this pair equal other pair

4.5 Class Tile

4.5.1 Field

- TileType tileType	contain tile type
- Entity entity	contain entitiy in this tile

4.5.2 Constructor

+ Tile (boolean lightColor, int x, int y, Entity entity)	initializes all fields

4.5.3 Methods

+ Pair add(Pair o)	return this pair add other pair
+ boolean equals(Object o)	return this pair equal other pair
+ hasEntity()	return tile has entity
+ Getter & Setter Methods all fields	

5 Package application

5.1 Class Main extends Application

5.1.1 Method

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