Star Hunter
DOCUMENTATION

Create by

Werapat Wangrungroj 6431343021

2110215 Programming Methodology

Semester 2 Academic year 2021

Department of Computer Engineering,

Faculty of Engineering, Chulalongkorn University

Star Hunter

Introduction

Star Hunter is an endless survival game and bullet hell game which requires player's skill to dodge bullet in game. This game tiles and sprite was from Soul Knight and Background music in game was from Loop Hero.

Player Character

 You are the hunter who come to hunt monster away from your planet. In this game you can choose one of 3 types of this character which have different status in each type.

1. Balancer, this type of character is balance in every stat.
2. Tank, this type of character is good at Vitality but cost with Agility and Power.
3. Speedy, this type of character is good at Agility and Power but cost with Vitality.

Status of Player

In this game there 6 different kind of status which can help player to survive in this game.

1. Health – This status indicating your life point, when you take damage from your Health you will gain a little Immunity frame, when your health is reaching 0 the game will end.
2. Attack – This status indicating how much damage you can do per 1 bullet that player will fire out.
3. Speed – This status indicating how fast you can move in the game, The more Speed mean the faster player can move.
4. Shot Speed – This status indicating how fast bullet can move in the game, The more Shot Speed mean the faster bullet can move.
5. Attack Speed – This status indicating how frequency you can shoot bullet, the lesser attack speed mean you can shoot bullet more frequency. This stat count as cooldown of your gun, once player finished firing will start time to count the cooldown. Regular shot have attack speed multiplier 1000 this mean attack speed 1.0 point equal to

cooldown 1.0 seconds in game, if attack speed multiplier increase to 1500 this mean attack speed 1.0 point equal to cooldown of gun 1.5 seconds in game.

6. Armour – This status indicating your armor, this armour going to be used to shield enemy attack before use health (any damage that exceed armour will not be calculate to health), the armour is going to regenerate overtime by about 10% of your maximum armour every 3 seconds if you haven't taken damage for 3 seconds. However, this status doesn't give you immunity frame to player character that mean your armour is going to be wipe out to 0 after enemy do contact damage to you.

Skill Table / Skill Tree

During in game, player can gain experience point by killing monster once player gain enough experience point player will level up and can gain 1 skill point per 1 level up, there's alternative way to get skill point other than level up that is get skill point potion which can drop from all monster. Player can use these skill point by hold [E] button to open Skill Table / Skill Tree to upgrading your stats/ get new ability(during you open Skill Table / Skill Tree the game will be pause and game screen will not update anything), these skill upgrading have different kind of requirement to upgrading your character and different result of upgrading which is

1. Double shot – grant player ability to firing bullet two time in row (have delay for 0.1 second) during this you can't change fire direction until double shot is finished firing, however your attack speed multiplier will increase by 1.5 times of normal one this means you going to shoot less

frequency than normal. Requirement – Need at least level 3 to get this ability and use 1 skill point

2. Tripple shot – grant player ability to firing bullet three time in row (have delay for 0.1 second) during this you can't change fire direction until tripple shot is finished firing , however your attack speed multiplier will increase by 2.0 times of normal one this means you going to shoot less frequency than normal. Requirement – Need at least level 6 to get this ability, need to get Double shot ability beforehand and use 2 skill point

3. Quadra shot – grant player ability to firing bullet four time in row (have delay for 0.1 second) during this you can't change fire direction until quadra shot is finished firing, however your attack speed multiplier will increase by 2.5 times of normal one this means you going to shoot less frequency than normal. Requirement – Need at least level 9 to get this ability, need to get Tripple shot ability beforehand and use 3 skill point

4. Attack Speed – upgrading player status by decreasing your attack speed by 0.1 point this means you can shoot more frequency, with regular shot have attack speed multiplier 1000 which mean attack speed 1.0 point equal to cooldown 1.0 seconds. However, your attack speed can not go below 0.1 point and there're cooldown gun cap which is 0.2 seconds that mean even you have attack speed 0.1 point your regular shot cooldown will be 0.2 seconds since it can not go below the cooldown cap but if you change regular shot to quadra shot if you have attack speed 0.1 point your quadra shot cooldown will be 0.25 seconds instead since quadra shot have attack speed multiplier 2.5 times of normal one. Requirement – use 2 skill point to upgrade.

5. Armour – upgrading player status by increasing your maximum armour and heal your current armour by 3 points. Requirement – use 1 skill point to upgrade.

6. Health – upgrading player status by increasing your maximum health and heal your current health by 5 points. Requirement – use 1 skill point to upgrade.

7. Attack - upgrading player status by increasing your attack by 1 point. Requirement – use 1 skill point.

8. Around Shot – grant player ability to fire 12 bullets around you have cooldown 5 seconds. Requirement – Need at least level 5 to get this ability and use 3 skill point.

* Player can not get duplicate ability if player try to get same ability it will play sound effect error instead.

Monster

There are 2 different kinds of monster in this game which is Normal Monster  and Champion Monster  . both monsters can do contact damage which can wipe out all of your armour and range damage by firing bullet to player. And all of monster attack speed stat will be <their base attack speed> + <randomly number between 0.0-1.0>

 Normal monster - have base stats and stats grow per wave lower than Champion Monster one. Normal monster will spawn every wave in game. Normal Monster will fire projectile like this 

 Champion monster – have base stats and stats grow per wave higher than Normal monster. In addition, Champion monster will grant randomly ability which is

1. Double shot – fire two bullets in spread to player

2. Tripple shot – fire three bullets in spread to player and the middle one is aimed at player

3. Quadra shot – fire four bullets in spread to player

4. Around Shot – fire 12 bullets around this monster in every 5 seconds

5. Reflect – fire 12 bullets around this monster if only player do damage to this monster. However, every second this monster health will lower by about 5% of its maximum health but if that number has decimal will only calculate for the integer part. For example, if 5% of its maximum health is 4.8 will decrease health by 4 points instead.

(* when monster obtain double shot ability, triple shot ability or quadra shot ability, monster will receive attack speed penalty same as player)

Furthermore, all of bullet that come from ability of Champion Monster (this no) have chance to be different speed function other than constant speed and will show different color of bullet which have

1.  this one move with constant speed same as Normal Monster bullet will have function as const function.
2. this one increasing move speed overtime will have function as linear
3.  this one will move with speed between 0 - <its amplitude> will have function as cosine function

Champion monster will spawn every 5 waves in game.

Potion

There're two kinds of potion in this game, both of them can only get by drop from monster randomly (Potion drop after monster corpse is removed). Health potion have chance to drop first then drop Skill Point potion next, but if Health potion drop it will not drop Skill Point Potion next. These potions will automatically disappear after its drop for 10 seconds these potions will warn player by flashing for 3 seconds before it disappeared.

 **Health Potion :** Increase player health by 10 point on contact with player however player health will not exceed maximum health.

 **Skill Point Potion :** Increase player skill point by 1 point on contact with player.

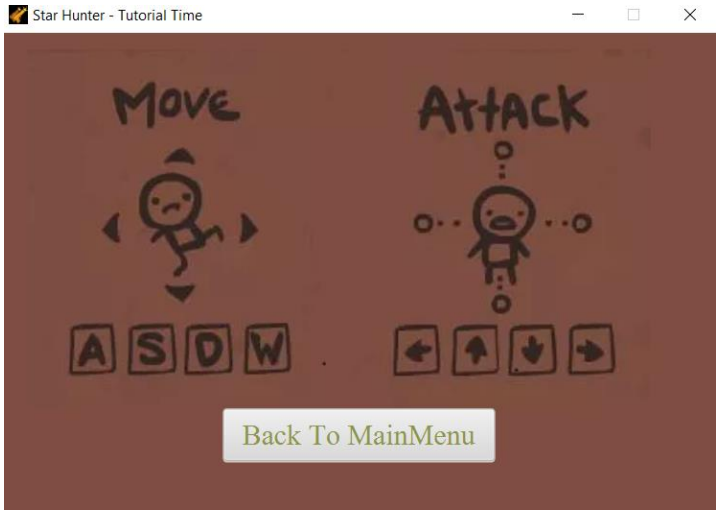
Gameplay

Main objective of this game is to survive from monster as long as possible. In every wave monster will spawn. The game system will start at wave 1 and wave will increase by 1 if player kill all of monster in that wave. The more wave in game the more difficult in game be. Once player's health reach 0 the game will end.

Example

Once the program start, you will be on main menu, there will be 3 button to click first one is start game once player click you will have to choose one of 3 types of character. Second one is tutorial menu will teach you basic key in this game. Third one is exit button once player click this program will end.

Tutorial menu



If player want to back to main menu you can click at Back To MainMenu button

Choosing type menu

Player have to choose one of three type of character in this game which is balancer,tank,speedy.

In-Game

Once player choose type of character, player will enter the game there're health bar and exp bar in the top-left corner on screen in game (there're example of health bar how it shows both health and armour below this). There're also information of character such as level, status, Skill Point and wave number in the top corner on screen.

To control character, player can press keyboard button, [W] for up direction, [A] for left direction, [D] for right direction, [S] for down direction. In this system you can combine key. For example, if you press [W] then press [A] you will go top – left direction.

To fire the bullet, player can press keyboard button, [<-] for fire left direction, [->] for fire right direction, [UP] for fire up direction, [DOWN] for down direction. However, this firing system doesn't design same way as control character it will only choose lasted key that you press. For example, if you press [<-] and not release this button then press [->] player will fire right direction

Health bar example

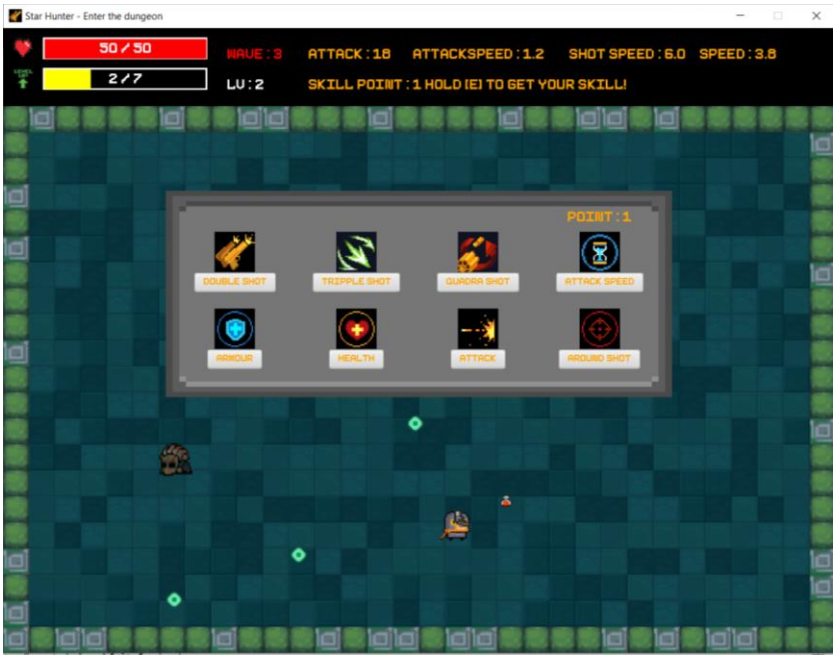
Player has full armour, health bar will show armour instead of your health if you still have armour

Player lost some armour, health bar still shows information of your armour.

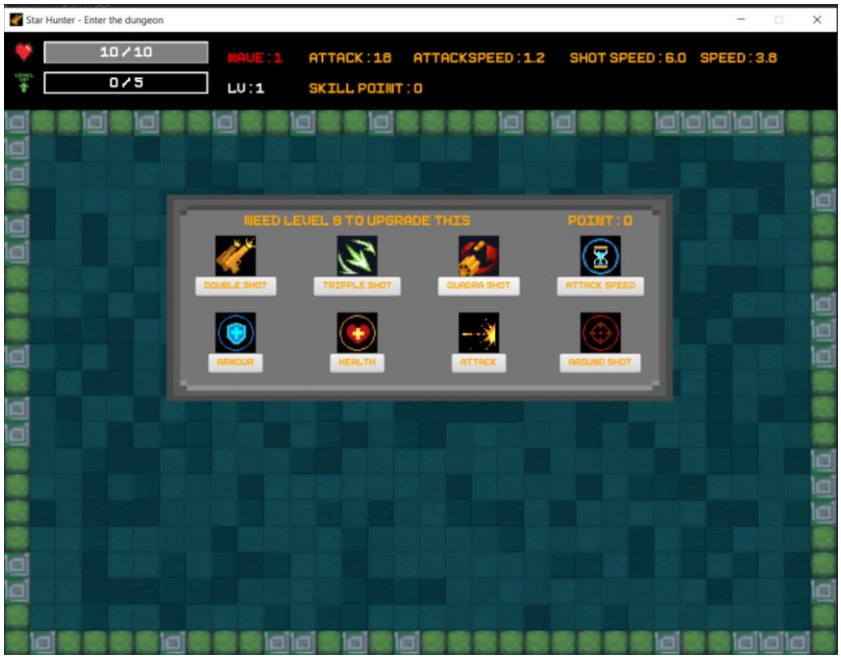
Player lost all armour, now health bar will show amount of your health.

In-Game(2)

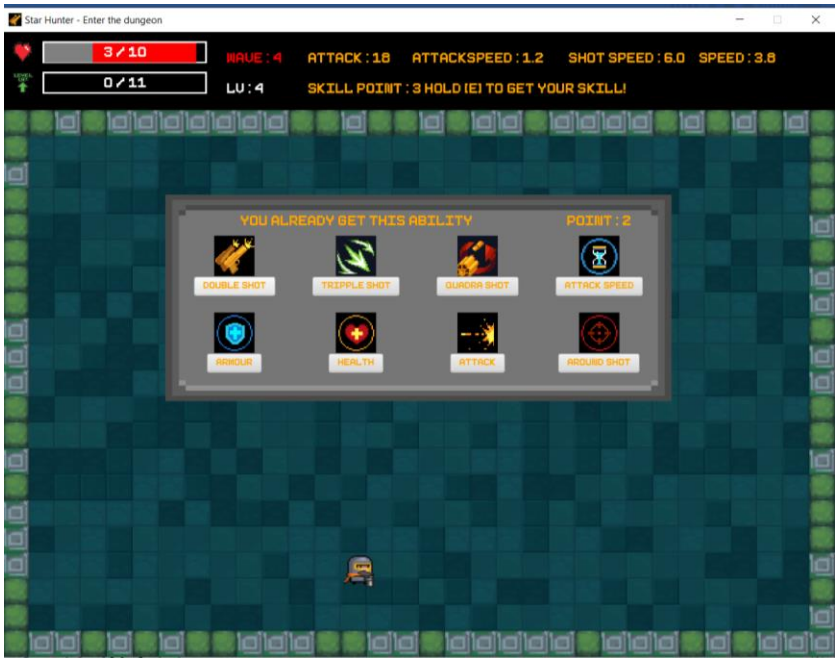
Once player have at least 1 skill point the game will tell that you can hold [E] to upgrading your character.



Once player open this skill table / skill tree the game will be pause and screen of game will not update anything. If player click to get ability or upgrade stats but you didn't meet upgrading's requirement, button will play error sound effect and tell you what doesn't meet requirement instead.

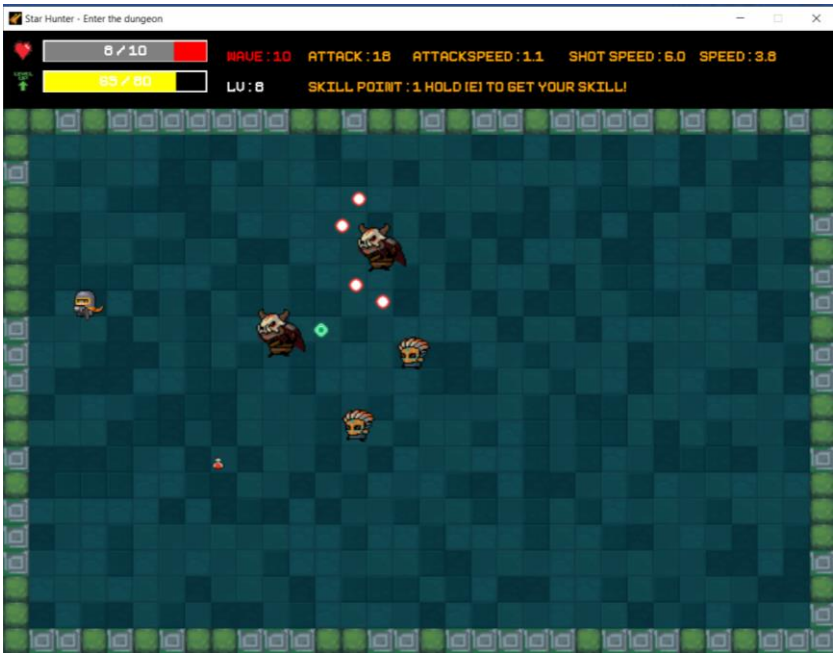


An Example for player try to get duplicate ability



In-Game(3)

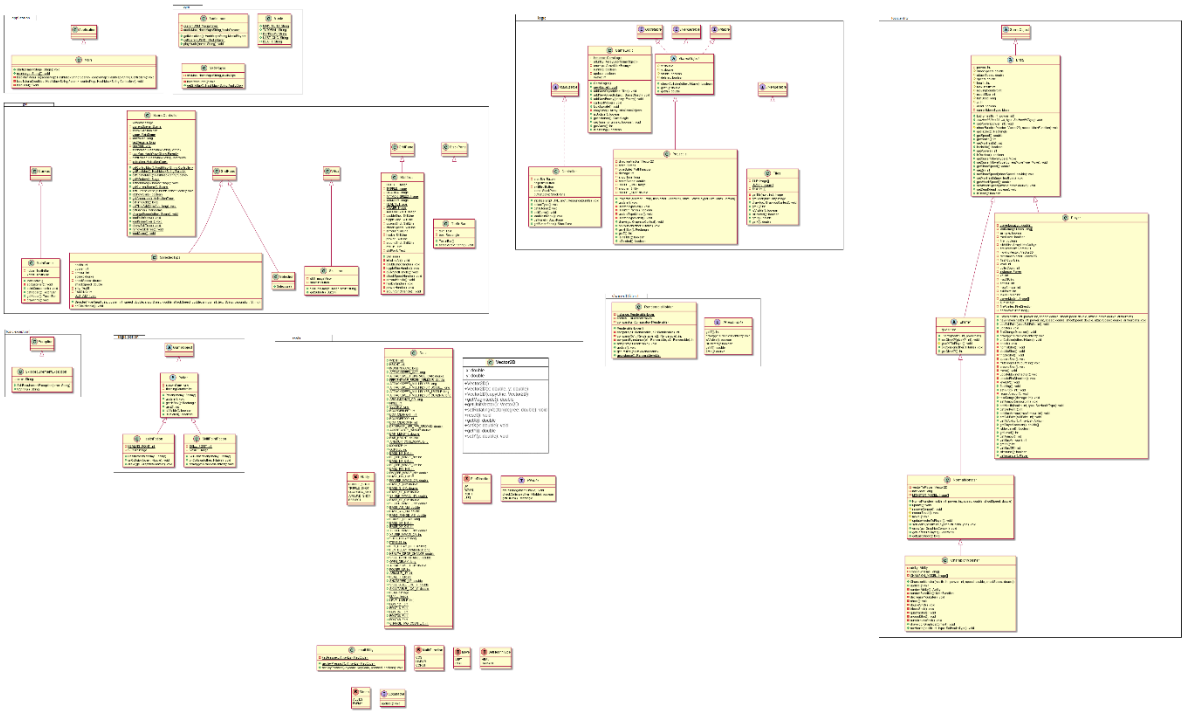
In Every 5 wave, Champion Monster will be spawned in game



After wave 10, music in game will change.

Once player's health reach 0, the game will over. Player can go back to main menu by press "to main menu" button.

Class Diagram



1. Package application
   (1.1) Class Main extends Application
   (1.1.1) Methods

+ void start(Stage primaryStage) throws Exception	-Set window in SceneController as primaryStage -Set primaryStage unresizable -Add image to icons -Get Maps from SceneController and to use in load Methods from Main Class -Call load Methods from Main -Call Static Methods loadMainMenu from SceneController
+ void main(String[] args)	Launch application
- void loadMainMenuFile(HashMap<String, Scene> sceneMap, HashMap<String, Controller> controlMap) throws IOException	Create Scene from MainMenu2.fxml and put in sceneMap Get Controller from MainMenu2.fxml and put in controlMap

-void loadTutorial(HashMap<String, Parent> rootMap, HashMap<String, Controller> controlMap) throws IOException	Get Root and Controller from Tutorial.fxml and put in rootMap and controlMap
-void loadBGM()	Load music as MediaPlayer and put in mediaMap in AudioLoad.

2. Package bgm

(2.1) Class AudioLoad

(2.1.1) Fields

-MediaPlayer currentBGM	Background Music that current playing
-HashMap<String,MediaPlayer> mediaMap	A map that contain all MediaPlayer that can play

(2.1.2) Methods

+ HashMap<String, MediaPlayer> getMediaMap()	Get mediaMap
+ MediaPlayer getCurrentBGM()	Get currentBGM
+ void playMusic(String name)	If currentBGM is not null stop currentBGM Set currentBGM as value of mediaMap with key as name Set currentBGM cycleCount as INDEFINITE And play currentBGM

(2.2) Class Music

(2.2.1) Fields

+ final String MAIN_MENU	Contain location of main menu music
+ final String TUTORIAL	Contain location of tutorial menu music
+ final String DUNGEON	Contain location of first part game music

+ final String LASTLONG	Contain location of mid-late game music
+ final String DEATH	Contain location of Gameover music

(2.3) Class SFXPlayer

(2.3.1) Fields

- HashMap<String,AudioClip> sfxMap	A map that contain all of sound effect in this game

(2.3.2) Static initializer block

Static	Call loadResource method to initialize static fields

(2.3.3) Methods

- void loadResource()	Create instance of AudioClip to use as sound effect and put them into sfxMap
+ HashMap<String, AudioClip> getSfxMap()	Get sfxMap

3. Package gui

(3.1) Class MainGame extends Canvas

(3.1.1) Fields

- TextInBar hpbar	Rectangle with text in it which have same size and location with Health bar Container
- TextInBar xpbar	Rectangle with text in it which have same size and location with Experience bar Container

(3.1.2) Constructor

+ MainGame()	Set width and height equal to constant from Class Data Initialize hpbar and xpbar Set visible as true Initialize Listener with addListener method

(3.1.3) Methods

- void addListener()	Set on key pressed on this Node with static method setKeyPressed from class InputUtility
+ void paintComponent()	-Fill screen with black color -call drawInfo method -draw all element in List from RenderableHolder if that element is visible and not deleted with draw method
+ TextInBar getHpbar()	Get hpbar
+ TextInBar getXpbar()	Get xpbar
- void drawInfo()	-fill text with the game information such as player information current wave (to get Player information use static method get Instance from Class Player) -draw health in health bar with red color with filled rect according to player information -draw armour in health bar with gray color with filled rect according to player information -draw EXP in xpbar with yellow color with filled rect according to player information Set text in health bar as player armour if player have armour more than 0 if not set as health instead Set text in xpbar as exp of player if player is max level show that player max level instead if player have skill point more than 0 it will show tip about how to get ability

	Draw Image of heart and level with image from Data Class constant

(3.2) Class SceneController

(3.2.1) Fields

-Stage window	Stage of this game
-Scene currentScene	Current Scene that showing
-boolean showSkill	Check if current show skill tree
-MainGame game	Canvas of this game
-long lastPause	A time of last pause in this game
-long lastResume	A time of last resume in this game
-long timeAdd	Time difference of last pause and last resume
-HashMap<String,Scene> sceneMap	A map that contained scene from fxml
-HashMap<String,Parent> rootMap	A map that contained root Node from fxml
-HashMap<String,Controller> controlMap	A map that contained Controller from fxml
-AnimationTimer animation	A Java FX thread that using to update over time in this game

(3.2.2) Methods

+ void changeScene(Scene other)	Set Scene to window field (Stage that current showing) as other argument and set currentScene as this other scene Show window stage Set x and set y position stage to the middle of your screen
+ void loadMainMenu()	Set title of this game to “Star Hunter – Main Menu” Change scene to Main menu by get scene from sceneMap

+ void loadGameOver()	Show Game Over Screen and set width and height with 600 x 400 In Game Over Screen will show how many wave that you survive also playing Death music by call static method from AudioLoad Class and there's button that will change screen to Main menu Set title to "Star Hunter – You Die"
+ void showSkillTree()	Showing skill table by create SkillTree Instance and add to AnchorPane root to upgrade your ability also set showSkill as true and set lastPause equal to System.currentTimeMillis()
+ void removeSkillTree()	Remove skill table by remove last element in children list also set showSkill to false set lastResume equal to System.currentTimeMillis() Add timeAdd equal to difference of lastResume and lastPause
+ void loadGame()	-Use root Node in this game as AnchorPane -Call static Method newGame from Class GameLogic -Initialize static MainGame game -Get GameLogic instance for use in animation timer -Set position hpbar and xpbar in Anchorpane -Add hpbar and xpbar to AnchorPane node - Initialize AnimationTimer which will paint component,update logic,update

	List of RenderableHolder if only Game is not pause -Start AnimationTimer
+ getters / setters for other fields	

(3.3) Class SelectedType extends GridPane

(3.3.1) Fields

- int health	Selected player's health
- int power	Selected player's attack
- int armour	Selected player's armour
- double speed	Selected player's speed
- double shootSpeed	Selected player's shootSpeed
- double attackSpeed	Selected player's attackSpeed
- Text[] info	Initialize with Text array have 7 length Which should be use
- final int PADDING	Constant of Padding pixel
- final int DES_WIDTH	Constant of Width in description line

(3.3.2) Constructor

+ SelectedType(int health, int power, double speed, double shootSpeed, double attackSpeed, int armour, String text, String description)	-Set select stats according argument -Initialize instance in Text array according select stats, and description and create instance of type as text argument Set font all of text as FONT16 static constant from Data Class as well as color of text which will be orange except for description line will set color be lime -Set Vgap, Hgap, Padding equal PADDING constant in this class -Set Text type to center -Add Text to grid by row which have health, attack, speed, shoot speed,

	attack speed, armour, description in order -Set on mouse click with onClickHandle -Set background color to blue -Set on mouse enter as -Set background color to red -Play sfxMusic with key as “enter” -Set on mouse leave as -Set background color to blue

(3.3.3) Methods

- void onClickHandle()	-Create instance of player with static method newPlayer with status according to this SelectedType stats -clear StackPane in MainMenu -Set Disable to false in buttonZone -Load Game from SceneController -Play dungeon music in AudioLoad -Play sfxMusic “btn”

(3.4) Class Selection extends GridPane

(3.4.1) Constructor

+ Selection()	-Create 3 instance of SelectedType have 3 type which is Balancer, Tank, and Speedy -Add these instance to grid by column which is Balancer Tank and Speedy in row 1 -Add header text in row 0 ,span 3 by column, and in center with FONT16 from Data class and orange color -Set grid padding with value 10 -Set color background with blue color

(3.5) Class SkillLine extends VBox

(3.5.1) Fields

- ImageView skill	Image of skill
- Button submit	Submit button

(3.5.2) Constructor

+ SkillLine(Image image,String text)	-Set padding to 10 -Initialize ImageView skill with image argument -Set fit width and height to 50 -Set VBox alignment to center -Initialize Button submit with text argument -Set on mouse enter playing “enter” sfx -Set submit button with FONT16 from Data class and color orange -Add skill image and submit button to VBox

(3.5.3) Method

+ Button getSubmit()	Get submit button

(3.6) Class SkillTree extends GridPane

(3.6.1) Fields

- final Image DOUBLE	Image of Double Shot ability
- final Image TRIPPLE	Image of Tripple Shot ability
- final Image QUADRA	Image of Quadra Shot ability
- final Image ATTACK_SPEED	Image of Upgrading your attack speed stat
- final Image ARMOUR	Image of Upgrading your armour stat
- final Image HEALTH	Image of Upgrading your health stat
- final Image POWER	Image of Upgrading your attack stat
- final Image AROUNDSHOT	Image of Around Shot ability
- SkillLine doubleShot	GUI of Double Shot ability
- SkillLine trippleShot	GUI of Tripple Shot ability
- SkillLine quadraShot	GUI of Quadra Shot ability
- SkillLine attackSpeed	GUI of Upgrading attack speed

- SkillLine armour	GUI of Upgrading armour
- SkillLine health	GUI of Upgrading health
- SkillLine power	GUI of Upgrading attack
- SkillLine aroundShot	GUI of Around Shot ability
- Text result	result of using skill point in Text
- Text skillPoint	Text that indicate your current Skill Point

(3.6.2) Constructor

+ SkillTree()	-Set Hgap equal to IconSize from Data constant - Initialize all SkillLine with Image from constant and name of its Ability/Upgrading - Initialize result with null -Initialize skillPoint with “Point : ” + skillPoint -Set Font and Color to FONT24 and Orange in all Text -Set Padding to constant from Data.ICONSIZE -Add doubleShot, tripleShot, quadraShot, attackSpeed in row 1. Add armour, health, power, aroundShot in row 2. Add result and SkillPoint in row 0 and set result to span 3 column -Set result and skillPoint to Center -Set style background to image skilltable.png with width 100% height 100% -Call InitializeAll Method

(3.6.3) Method

- void initializeAll()	Set on action all SkillTree button with its handle if there is no SkillRequirementException it will set

	result that you used skill point if there is result will show you error message why it failed to get ability also play Error sfx music
- void doubleShotHandle() throws SkillRequirementException - void trippleShotHandle() throws SkillRequirementException - void quadShotHandle() throws SkillRequirementException - void attackSpeedHandle() throws SkillRequirementException - void armourHandle() throws SkillRequirementException - void healthHandle() throws SkillRequirementException - void powerHandle() throws SkillRequirementException - void aroundShotHandle() throws SkillRequirementException	If player doesn't meet the requirement of skill it will throws SkillRequirementException with error message If player meet the requirement player will gain that stat or ability. In addition, sfx music named "Successful" will play

(3.7) Class TextInBar extends StackPane

(3.7.1) Fields

- Text text	Text inside Box
- Rectangle box	A box that represents to size of actual Text box

(3.7.2) Constructor

+ TextInBar()	Initialize Text and set font with FONT24 and Color White Initialize box with rectangle same size with healthbar and xpbar set color transparent Add text and box to stackpane to order

(3.7.3) Methods

+ void setText(String text)	Set Content in text

4. Package Logic

(4.1) Class Controller implements Initializable

(4.1.1) Fields

- Button startBtn	A button that will start the game
- Button helpBtn	A button that will show basic movement of this game
- Button exitBtn	A button that will close the game
- StackPane pane	A pane that will show tutorial of game or show selectable type of character
- StackPane buttonZone	A pane that contains 3 buttons

(4.1.2) Methods

+ void initialize(URL arg0, ResourceBundle arg1)	Auto implements method from fxml files Do nothing
+ void selectType()	Create and Add Instance of Selection to pane set disable to buttonzone with true request focus on pane Play sound effect button
+ void getTutorial()	Load Tutorial menu on pane and play tutorial music
+ void exitGame()	Close Stage
+ void loadMainMenu()	Remove all children in pane to return to Main Menu also play music main menu
+ getter other fields	

(4.2) Class GameLogic

(4.2.1) Field

- GameLogic instance	Instance logic of this game
- ArrayList<GameObject> allEntity	List of all GameObject in this game
- ArrayList<Enemy> enemys	List of all enemy in this game
- boolean running	Check if game current in running state
- Boolean update	Check if game current update

- int wave	A number of current wave

(4.2.2) Constructor

+ GameLogic()	Set timeAdd in SceneController to 0 Clear ArrayList<IRenderable> in RenderableHolder Initialize ArrayList allEntity and enemys Create instance of Tile Set running to true Set wave to 0 Get instance of Player Add Tile and player to GameLogic and RenderableHolder List by addNewObject Methods

(4.2.3) Methods

+ void newGame()	Initialize instance of GameLogic again
+ void addNewObject(Tiles tiles)	Add Tiles to RenderableHolder List (Since tiles cannot update so don't add to GameLogic List)
+ void addNewObject(GameObject object)	Add object in game to allEntity list and RenderableHolder list
+ void addNewEnemy(Enemy enemy)	Add enemy to List of enemys and use addNewObject method to add in list of allEntity and Renderable Holder
- void callNextWave()	If size of enemys list is equal or less than 0 will call monster next wave and increase number of waves If wave mod 5 equal to 0 will call Champion Monster wave If not will call Normal Monster wave
+ void logicUpdate()	Set update to true and use copy of allEntity list (to prevent ConcurrentModificationException) to update all object in game Then set update to false

	And check collision between allEntity list by use 2 nested loops If they collision call method oncollision if that object is not deleted Then delete GameObject element from allEntity list if that element is deleted Then check enemys list if there element that deleted remove them from list
- ArrayList<GameObject> copyOne()	Return a copy of allEntity list to prevent ConcurrentModificationException (since new entity will add to a original one not copy one so ConcurrentModificationException will not occur)
+ other getters/setters fields	

(4.3) Abstract Class GameObject implements Updatable,IRenderable,Hitable

(4.3.1) Fields

# double x	Current position in x axis
# double y	Current position in y axis
# boolean visible	To Check if gameobject is visible
# boolean deleted	To Check if gameobject is deleted

(4.3.2) Methods

+ boolean checkCollision(Hitable other)	Check if this Hitable object and other one is collision by getHitBox and get boundsInParent from two fields and check with intersects method
+ double getY()	Return y position
+ double getX()	Return x position

(4.4) Class Projectile extends GameObject

(4.4.1) Fields

- Vector2D directionVector	A direction of projectile that will be update
- States state	State of this projectile is come from player or enemy
- MathFunction typeBullet	A type of speed that projectile will update
- int damage	Value of damage will deal to hp of other if it hit player or enemy
- long shootTime	A time that this projectile is created
- double shootSpeed	A speed that this projectile will update
- final Image[] BULLET_IMG	All Array of Image in projectile class
- Entity shooter	A person who shoots this projectile
- final double BULLET_SIZE	A constant size of projectile

(4.4.2) Static Initialize Block

Static	Initialize all of element Image in BULLET_IMG with Image from file in game

(4.4.3) Constructor

+ Projectile(Entity shooter, Vector2D fireVector, States state, MathFunction, typeBullet)	Set state,shooter and typeBullet equal to argument Set directionVector by create new instance of Vector2D of fireVector Set x,y,damage,shootSpeed with information from shooter Set visible to true and set deleted to false Set shootTime with currentTime - TimeAdd

(4.4.4) Methods

+ void update()	Check typeBullet first then update by its updateType

	Then check if projectile is out of frame if so set visible to false and set deleted to true
- void updateTypeCos()	This methods will update projectile by math cos fuction have argument as different of time now and shootTime and to prevent projectile move back will have absolute math function
- boolean isOutOfFrame()	Check if projectile is out of frame that can play in game
- void updateTypeLinear()	Update projectile by increase speed of projectile over time
- void updateTypeConst()	Update projectile by constant speed
+ void draw(GraphicsContext gc)	If state is ALLIES will draw projectile with image of blue color If state is ENEMY check next condition -if shooter instance of Champion Monster check next condition -if typeBullet is linear draw projectile with image of yellow color -if typeBullet is cos draw projectile with image of purple color -if not either linear or cos draw projectile with image of red color -if shooter instance of Normal Monster draw projectile with image of green color
+ void onCollision(Hitable other)	Deal damage on player with takeDamage if projectile from enemy and other is player and then set deleted as true, set visible as false in this bullet Deal damage on enemy and set deleted as true set visible as false if projectile from player and other is

	enemy will do nothing if enemy is already dead
+ Rectangle getHitBox()	Get HitBox of this projectile
+ int getZ()	Return 0
+ other getters fields	

(4.5) Class Tiles implements IRenderable

(4.5.1) Fields

- final Image[] TILE	A constant that contains image of tile that can pass
- final Image[] BLOCK	A constant that contains image of tile that cannot pass
- int[][] field	A field that contain numbers what that tile should draw

(4.5.2) Static Initialize Block

Static	Initialize number randomly in array of field if they are border will random number 0-1 and it will be block if not it will be 0-4 and it will be tile Initialize Image of TILE and BLOCK

(4.5.3) Methods

- Image getTile(int num)	Get image of Tile from TILE array
- Image getBlock(int num)	Get image of Block from BLOCK array
+ void draw(GraphicsContext gc)	Draw tile from field variable if they are border they will call getBlock if not they will call getTile and draw image that received from return value
+ int getZ()	Return -10
+ boolean isVisible()	Return true
+ boolean isDeleted()	Return false
+ double getX()	Return 0
+ double getY()	Return 0

5. Package logic.entity

(5.1) Abstract Class Entity extends GameObject

(5.1.1) Fields

- int power	Entity attack power
# double shootSpeed	A speed of projectile from this entity
# double attackSpeed	A value to decide how frequency this entity can attack
# double speed	A speed that this entity can move
# int health	Current health of this entity
# int maxHealth	Maximum health of this entity
# int movingCounter	A value that increasing when entity moving (for doing animation)
# int modelSize	A size of this entity
# long lastDead	A time of this entity is dead.
# int z	The value that decides whether to draw first or draw after.
- boolean dead	To check if entity is already dead.
- Move currentMoveType	Current move direction whether left or right.

(5.1.2) Constructor

+ Entity(int health, int power)	Set visible to true Set deleted to false Set this health and maxHealth equal to heath Set this power to power Set z equal to 5 by default Set currentMoveType to Right

(5.1.3) Methods

+ void setHealth(int health,SetHealthType type)	To be implement in other class.
+ void setPower(power)	Set this power if power less than 1 then set power to 1
# void shootBullet(Vector2D refVector,MathFunction speed)	Create instance of bullet according to this is instance of Enemy or Player if Enemy's bullet will have state as

	ENEMY if it player's bullet will have state as AILLES Then addNewObject to GameLogic
+ Rectangle getHitBox()	Return this hitbox with rectangle
+ other getters/setters Fields	

(5.2) Abstract Class Enemy extends Entity

(5.2.1) Fields

- int giveXP	A amount of exp that will give to player if this enemy dead

(5.2.2) Constructor

+ Enemy(int health, int power)	Call super constructor Set movingCounter to 0

(5.2.3) Methods

+ void setGiveXp(int giveXP)	If giveXP less than 0 set to 0 Then set this giveXP equal to giveXP
# void giveXPToPlayer()	If this already dead do nothing Add exp to player equal to this giveXP
+ void onCollision(Hitable other)	If this already dead then do nothing If other is player deal damage to player equal to this power
+ int getGiveXP()	Get giveXP

(5.3) Class NormalMonster extends Enemy

(5.3.1) Fields

- Vector2D vectorToPlayer	A vector that pointing to player
- long lastShoot	A time of last shoot this monster
- final Image[] MONSTER_MODEL	A Image of this monster which have total 6 Image

(5.3.2) Static Initializer Block

Static	Initialize array of Image by using for loop

(5.3.3) Constructor

+ NormalMonster(int health, int power, double speed, double shootSpeed)	Call Super constructor Randomly number of x and y Set position to x and y if they are with in border of this game and distance of this monster to player is more than or equal to Acceptable range(112) Set this shootSpeed and speed according to argument Set randomly attack speed of this monster by <base attack speed > + <random double 0 – 1> (using constant from Class Data) Set lastShoot to time of now – time add Set Vector to player to (0,0) Set dead to false Set giveXP to <base xp> + (int) <xp per wave * wave> (using constant from Class Data) Set model Size to 40

(5.3.4) Methods

+ void update()	If this monster is dead try to call remove corpse and end this method by return method If not updateVector to player If magnitude of vectorToPlayer not equal to 0 do normal shoot Then move
# void removeCorpse()	Check if difference of now time and lastdead more than or equal corpse delay constant If so tries to drop health potion first if not tries to drop skill potion next

	Then make this monster not visible and deleted this monster
# void normalShoot()	Shoot the bullet with math function constant and use reference vector as vector to player if only difference of time now and lastShoot is more or equal than attackSpeed * attackSpeedMultiplyer (from Data class constant) if this is ChampionMonster will have sound effect
# void move()	Try to set currentMoveType accord to x axis in vector to player Try to update position accord to vectorToPlayer * this speed and set x and y to not out of frame that game can play If moving Vector is more or equal 99 set to 0 Add movingCounter by 1
# void updateVectorToPlayer()	Update vectorToPlayer in order to point to player then set it by getUnitVector if player and monster close enough it will be Vector (0,0) instead
+ void setHealth(int health,SetHealthType type)	Deal damage to this monster if Damge type if health below or equal 0 giveXP to player before it die then set time dead
+ void draw(GraphicsContext gc)	Draw this monster according to states of this monster by using image from MONSTER_MODEL
+ other getters fields	

(5.4) Class ChampionMonster extends NormalMonster

(5.4.1) Fields

- Ability ability	Ability of this Champion monster
- long[] coolDownTime	A array of cooldown time index 0 for cooldown around shot index 1 for cooldown decrease champion monster life index 2 for shooting bullet
- final Image[] CHAMPION_MODEL	A array of Image of Champion Monster

(5.4.2) Static Initialize Block

Static	Initialize array of Image by using for loop

(5.4.3) Constructor

+ ChampionMonster(int health, int power, double speed, double shootSpeed)	Call Super constructor Set ability as random ability Initialize coolDownTime with 3 length And set value now time in game(Real time – time add) Randomly set attack speed with <Base Attack Speed> + <Random double 0-1> Randomly set giveXP with <Base Xp for CM> + <Xp per wave for CM * wave> Set model size as 60

(5.4.4) Methods

+ void update()	If this already dead try to call removeCorpse and return If not update vector to player If ability is reflect try to reduce its health Try to shoot every update and move If ability is around will do Around Shot
- Ability randomAbility()	Return randomly ability
- MathFunction randomFunction()	Return randomly mathFunction
- void decreaseYourLife()	Reduce Champion Monster's health at most 5% of Max Health every second by use super setHealth

- void shoot()	If Champion monster already near enough will not shoot If not it will shoot according to its ability
- void doubleShot()	Shoot 2 projectile with random MathFunction diagonal to you if cooldown is over also play sound effect when shooting
- void trippleShot()	Shoot 3 projectile 1 projectile to player 2 other projectile rotate from center one by 60 degree in this 3 projectile will shoot with randomly MathFunction, projectiles will shoot if cooldown is over. Also play sound effect when shooting
- void quadraShot()	Shoot 4 projectile 2 projectile rotate 30 degree and 60 degree from also this also same as other 2 projectile but -30 degree and -60 degree instead all projectile will shoot with randomly MathFuction,projectile will shoot if cooldown is over.Also play sound effect when shooting
- void aroundShot()	Shoot 12 projectile around this Champion Monster by rotate 30 degree every projectile with random MathFunction
- void autoAroundShot()	If cooldown is over will do aroundShot and play sound effect
+ void draw(GraphicsContext gc)	Draw this monster according to states of this monster by using image from CHAMPION_MODEL
+ void setHealth(int health, SetHealthType type)	If it is damage will check next condition

	If ability is reflect will release around shot every time that call this method and call super setHealth Otherwise, just call super setHealth

(5.5) Class Player extends Entity

(5.5.1) Fields

- double[] playerLocation	Location of player in x and y
- long[] lastDamageTaken	Index 0 for time that take damage with health Index 1 for time that take damage with armour
- boolean immune	To check current is immune
- boolean maxLevel	To check current is max Level
- boolean fire	To check current is firing projectile
- ArrayList<Ability> allAbility	List of ability of player
- Vector2D fireDirection	A direction that player will fire projectile to.
- Vector2D movingVector	A vector direction that player will move
- Vector2D rememberVector	A vector that remember vector that player shoot (will not remember new vector if still shooting)
- int flashCount	A value to do flashing immune effect
- int level	Current level of player
- int bulletCount	A number that bullet already shoot
- Player instance	Instance of player
- int xp	Current exp in that time
- int maxXP	Maximum exp in that level
- int armour	Current armour in that time
- int maxArmour	Current maximum armour in that time
- int skillPoint	Current skillPoint at that time
- final int MAXLEVEL	Maxmimum level that player can have
- Image[] playerModel1	Array image of player
- Image gun	Image of gun

- FireDirection firePointer	A value to decide which direction player is firing
- long[] coolDownTime	Array for cooldown time. Index 0 for firing, index 1 for skill around shot, index 2 for regeneration armour, slot 3 for delay in multishot

(5.5.2) Static Initialize Block

Static	Initialize each element in image array by for loop with picture in file game

(5.5.3) Constructor

+ Player(int health, int power, double speed, double shootSpeed, double attackSpeed, int armour)	Use super constructor Set dead false Set immune false Set speed shootSpeed attackSpeed armour equal to argument Set max armour equal to armour Initialize lastDamageTaken and set each element as current time in game Initialize allAbility with empty list Initialize fireDirection and movingVector with Vector2D() Set x and y position to center of screen Set flashCount,movingCounter,bulletCount,xp and skillPoint to 0 Set z to 3 Set level to 1 Set maxXP equal to constant from Data class in array with index level – 1 Set maxLevel equal false Initialize coolDownTime with 4 length and set each element to current time in game Set model size to 40

(5.5.4) Methods

+ void newPlayer(int health, int power, double speed, double shootSpeed, double attackSpeed, int armour)	Recreate instance of player with argument
+ void useSkillPoint(int usedSkillPoint) throws SkillRequirementException	If skillPoint – usedSkillPoint is less than 0 throw SkillRequirementException with string of “You need atleast <usedSkillPoint> to get this ability” If not will set skillPoint with skillPoint – usedSkillPoint
+ void update()	If player is dead will set movingCounter to 0 if die long enough will stop animation and load game over screen If not will check if xp equal or exceed maxXP if so call levelUP If ability is contain around shot will try to call around shot Try to regain armour If InputUtility list of key size more than 0 will try to update fire direction and moving vector If remember vector is not null will replace fireDirection with this method If movingVector magnitude is not 0 will try to move and increasing movingCounter by 1 If not set movingCounter to 0 If movingCounter is more or equal than 99 set it back to 0 If fireDirection magnitude not equal 0 or still firing will do shoot Trie to do flashing Reset fireDirection Vector to (0,0)
- void fireChangeDirection()	Change firePointer according to remeberVector

+ void draw(GraphicsContext gc)	Draw character player according to state of player Draw gun according to firePointer value
+ void onCollision(Hitable other)	Auto add implement, do nothing
- void shoot()	Shoot according to ability highest priority for example if player have double shot and triple shot this method will do triple shot
- void normalShot()	Shoot projectile with constant function and refVector as fireDirection if not on cooldown
- void doubleShot()	Will do this method if not on cooldown or still firing and call multiFire(2) to fire projectile 2 time in row
- void trippleShot()	Will do this method if not on cooldown or still firing and call multiFire(3) to fire projectile 3 time in row
- void quadraShot()	Will do this method if not on cooldown or still firing and call multiFire(4) to fire projectile 4 time in row
- void multiFire(int bulletCount)	If not start fire will remember Vector and use this vector as reference instead of normal one and will set fire if delay cooldown is off will fire projectile and increasing this bullet count by 1 untill this bulletCount reach bulletCount argument will stop firing and remove remember vector then set this bulletCount back to 0
- void aroundShot()	Will do this method if not on cooldown Will fire 12 projectile around you use Vector (1,0) as reference then rotate each one by 30 degree then shoot bullet

- void move()	Will move player according to moving vector * speed this method will not move player if player trying to go out of border of this game and set playerLocation index of 0 and index of 1 according to player's x and y then reset movingVector every time
- void updateMovingVector()	Update moving vector according to key that player press on keyboard
- void updateFireDirection()	Update fireDirection according to key that player press on at the last and that key must be only [left],[right],[up] or [down]
- void levelUP()	Increasing level of player by 1 Increasing skill point of player by 1 Set xp to 0 If level reach Max level Set max level true Set xp to 999999998 Set maxXP to 999999999 if not set maxXP according to data in game
+ void flashing()	Do flashing animation if player is not dead and still immune to damage
+ void setXP(int xp)	If level reach max level do nothing If xp more than maxXP set it equal to this maxXP Set this xp as xp
- void regainArmour()	Try to regain armour if player doesn't take damage for 3 second will increase by Math.ceil(maxArmour * 0.1)
+ void takeDamage()	Will let damage done to armour first if player doesn't have any armour left damage will deal to player's health

	instead (damage that exceed armour will not calculate to player's health)
+ void setArmour()	If armour is more than maxArmour then set it equal to maxArmour If less or equal than 0 will set armour to 0 then play sound effect "shield break" set this armour equal to armour
+ void setHealth(int health,SetHealthType type)	If player is dead then do nothing If not check if it is damage type if so Player will grant immunity frame for 0.5 second play sound effect "hurt" and set this health equal to health unless player's health reach below or equal to 0 will set player is dead and set time dead set health to 0 play sound effect "die" If it is heal type player will increase health and not gain immunity frame like above one and player's health will not exceed player's max health
+ other getters/setters fields	

6. Package logic.exception
    (6.1) Class SkillRequirementException extends Exception
        (6.1.1) Fields

- String error	Error message that you want to print out

(6.1.2) Constructor

+ SkillRequirementException(String error)	Set this error as error

(6.1.3) Methods

+ String toString()	Override method return this string error instead

7. Package logic.potion

(7.1) Abstract Class Potion extends GameObject

(7.1.1) Fields

- long spawnTime	A time that this potion has spawned
- int flashingCounter	A value to doing flashing animation

(7.1.2) Constructor

+ Potion(Enemy enemy)	Set visible to true Set deleted to false Set this x and y position equal to enemy position Set spawn time equal to time in game (now time – time add) Set flashing counter to 0

(7.1.3) Methods

+ void update()	If potion last for 7 second will do flashing animation to warning that this potion going to be deleted If potion last for 10 second will delete this potion and make it not visible
+ Rectangle getHitBox()	Return this potion hit box
+ int getZ()	Return 1
+ other getters fields	

(7.2) Class HealthPotion extends Potion

(7.2.1) Fields

- final int HEALING_POINT	A amount of value healing to player
- final Image SPRITE	A image of this potion

(7.2.2) Constructor

+ HealthPotion(Enemy enemy)	Call Constructor from super

(7.2.3) Methods

+ void onCollision(Hitable other)	Healing to player if the one who hit is player and this potion is not deleted

	next deleted this potion including make this invisible
+ void draw(GraphicsContext gc)	Draw this potion

(7.3) Class SkillPointPotion extends Potion

(7.3.1) Fields

- final int SKILL_POINT	A amount of value that add skill point to player
- final Image SPRITE	Image of this potion

(7.3.2) Constructor

+ SkillPointPotion(Enemy enemy)	Call Constructor from super

(7.3.3) Methods

+ void onCollision(Hitable other)	Add Skill Point to player by SKILL_POINT if the one who hit is player and this potion is not deleted next deleted this potion including make this invisible
+ void draw(GraphicsContext gc)	Draw this potion

8. Package sharedObject

(8.1) Interface IRenderable

(8.1.1) Methods

+ int getZ()	Get Z position (depth) this object
+ void draw(GraphicsContext gc)	Draw this object on canvas, depends on each object
+ boolean isVisible()	To check if an object is visible
+ boolean isDeleted()	To check if an object is deleted
+ double getY()	Get y position this object
+ double getX()	Get x position this object

(8.2) Class RenderableHolder

(8.2.1) Fields

- RenderableHolder instance	Instance of this class
- List<IRenderable> entities	List of all entity that can be draw on canvas

- Comparator<IRenderable> comparator	A comparator List of IRenderable to use in sort method

(8.2.2) Constructor

+ RenderableHolder()	Initialize list of enities Initialize Comparator by using compare method

(8.2.3) Methods

- int compare(IRenderable o1, IRenderable o2)	This method compare by Z value if two object is equal in Z value will be compare by y value next
- int compareY(IRenderable o1, IRenderable o2)	This method compare by Y value if two object is equal in Y value will be compare by instance next
- int compareByInstance(IRenderable o1, IRenderable o2)	If o1 is champion monster and o2 is normal monster will return 1 if not return -1 instead
+ void add(IRenderable entity)	Add entity to list of entities and sort list of entities by this class comparator
+ void update()	Remove element that already deleted in entities list
+ other getters fields	

9. Package tools

(9.1) Enum Ability

(9.1.1) Enum

DOUBLE_SHOT	Represent to ability that can shoot 2 projectiles
TRIPPLE_SHOT	Represent to ability that can shoot 3 projectiles
QUADRA_SHOT	Represent to ability that can shoot 4 projectiles
AROUND_SHOT	Represent to ability that can shoot 12 projectiles around you

REFLECT	Represent to ability that will do something back if it take damage from other entity

(9.2) Enum FireDirection

(9.2.1) Enum

UP	Represent that you currently fire projectile to top.
DOWN	Represent that you currently fire projectile to bottom.
RIGHT	Represent that you currently fire projectile to right.
LEFT	Represent that you currently fire projectile to left.

(9.3) Enum MathFunction

(9.3.1) Enum

COS	Represent that speed of projectile will move like cos function
LINEAR	Represent that speed of projectile will increase over time
CONST	Represent that speed of projectile will be constant at any time

(9.4) Enum Move

(9.4.1) Enum

LEFT	Represent that this entity facing left
RIGHT	Represent that this entity facing right

(9.5) Enum SetHealthType

(9.5.1) Enum

HEAL	Represent that current set health is healing type
DAMAGE	Represent that current set health is damaging type

(9.6) Enum States

(9.6.1) Enum

ALLIES	Represent that this object from player
ENEMY	Represent that this object from enemy

(9.7) Interface Hitable

(9.7.1) Methods

+ void onCollision(Hitable other)	Do something if this object is collision with another one.
+ boolean checkCollision(Hitable other)	Check if this object is collision with another one.
+ Rectangle getHitBox()	Get hitbox this object.

(9.8) Interface Updatable

(9.8.1) Methods

+ void update()	Update logic this object

(9.9) Class Data

(9.9.1) Fields : this class contain a bunch of data constant in game such as width,height, information in game (hp monster per wave etc.),pixel in game, font,gunsize,iconsize,image,exp table and a lot of constant

(9.10) Class InputUtility

(9.10.1) Fields

- ArrayList<KeyCode> keyPressed	A list of key that you press on keyboard

(9.10.2) Methods

+ ArrayList<KeyCode> getKeyPressed	Getter of keyPressed
+ void SetKeyPressed(KeyCode keycode, boolean pressed)	If that key is pressed and not in list keyPressed will add it to keyPressed If that key is not pressed will remove that key of list of keyPressed Check if player still not died and game logic didn't update

	If that key is [E] and pressed and still not show skill will pause the game and open skill table If that key is [E] and not pressed and still show skill will unpause the game and remove skill table to continue play the game

(9.11) Class Vector2D

(9.11.1) Fields

- double x	Current vector position in x axis
- double y	Current vector position in y axis

(9.11.2) Constructor

+ Vector2D()	Create vector (0,0)
+ Vector2D(double x, double y)	Create vector (x,y)
+ Vector2D(Vector2D copyOne)	Create vector that x and y have same value as copyOne

(9.11.3) Methods

+ double getMagnitude()	Get magnitude of this vector
+ Vector2D getUnitVector()	Get unit Vector of this vector if magnitude equal to 0 will return vector (0,0) instead
+ void setRotatingVector(double degree)	Rotate this vector according to the argument
+ void reset()	Reset this vector back to vector (0,0)
+ other getters/setters	