

Void Seekers Implementation Document

Created by

Yanapat Patcharawiwatpong (6430082021)
Vivatsathorn Thitasirivit (6432158421)

**2110215 Programming Methodology
Semester 1 Year 2022
Chulalongkorn University**

Table of Contents

Table of Contents	2
Introduction	4
Key Binding	4
Main Menu Screen	5
Game Screen	5
End Screen (player died)	6
UML Class Diagram	7
Java Implementations (com.game.void_seekers)	8
1. interface	8
(l) AffectPlayer	8
(l) Attack.....	8
(l) Draw	8
(l) Usable.....	8
2. character.base	9
(A) CharacterHealth	9
(A) GameCharacter extends CharacterHealth implements Draw	10
(A) PlayableCharacter extends GameCharacter implements Attack	11
(A) EnemyCharacter extends GameCharacter	13
3. character.derived	14
(C) PlayerIsaac extends PlayableCharacter implements Attack	14
(C) PlayerSoul	14
(C) PlayerJared	14
(C) EnemyGaper	15
4. item.base	16
(A) Item implements Pickable, Draw	16
(A) PocketItem extends Item implements Pickable	16
(A) EffectItem	16
(A) Active extends EffectItem implements Usable	16
(A) Passive extends EffectItem implements Pickable	17
(A) Trinket extends EffectItem.....	17
5. item.derived	18
(C) Battery extends PocketItem	18
(C) Bomb extends PocketItem	18
(C) Coin extends PocketItem	18
(C) Heart extends PocketItem.....	19
(C) Exploding extends PocketItem.....	19
6. item.passive	20
(C) LonelyEye extends EffectItem implements AffectPlayer	20
(C) EternalBlessing extends Passive implements AffectPlayer	20
7. item.active	21
(C) BookOfRage.....	21
8. item.trinket	21
(C) LingeringFeather	21
9. obstacle.base	22
(A) Obstacle implement Draw	22

10. obstacle.derived	22
(C) Bush extends Obstacle implements Draw	22
(C) Crate extends Obstacle implements Draw	22
(C) Mud extends Obstacle	22
(C) Spike extends Obstacle	23
11. projectile.base.....	24
(A) Projectile implements Draw	24
12. projectile.derived	25
(C) NomalProjectile extends Projectile implements Draw, AffectPlayer	25
(C) RedProjectile extends Projectile implements Draw, AffectPlayer	25
(C) WhiteProjectile extends Projectile implements Draw, AffectPlayer	25
13. room.base.....	26
(A) Room implements Draw	26
(E) RoomDirection.....	26
14. room.derived	27
(A) NormalRoom extends Room	27
(C) SpawnRoom extends NormalRoom.....	27
(C) EnemyRoom extends NormalRoom.....	27
(C) BossRoom extends NormalRoom.....	27
15. tools	28
(C) Coordinates implements Clonnable	28
(C) RandomIntRange	28
16. logic	29
(C) final GameLogic	29
(C) GameEvent implements Runnable	30
(C) GameUtils	31
(C) GameAssets.....	32
(C) EnemyEvent implements Runnable	32
(E) GameState	32
17. render.....	32
(A) AbstractScene extends Scene	32
(C) ActiveBar extends AbstractScene.....	33
(C) HealthBar extends AbstractScene	33
(C) InventoryBar extends AbstractScene.....	33
(C) TrinketBar extends AbstractScene.....	34
(C) MenuScene extends AbstractScene	34
(C) GameScene extends AbstractScene	34
(C) EndGaneScene extends AbstractScene.....	34
18. Application	35
(C) Main extends Application	35

Introduction

Void Seekers is inspired by an action RPG shooter (rouge-like) game: “*The Binding of Isaac*.” The main objective of the game is getting highest score possible. The game ends when main character (player) died. The score is based on how much damage can each enemy player encountered deal.

The player spawns in the spawn room, like other rooms, the player can go to next rooms in 4 directions: top, bottom, left, and right. Once player went to another room, the player will encounter enemies and/or items in each room. The difficulty of each room increases proportionally to how far the room is from the spawn room.

The player has 3 options to deal damage and kill enemies:

- Melee attack – if the player is in the proximity range (collides) with enemy, the player can hit enemy with their damage.
- Ranged attack – the player can attack enemy using projectiles. If the projectiles hit and enemy, the enemy will be dealt with projectiles’ damage.
- Effect items – e.g., Bomb. The player can use effect items for area effects which may or may not affect player as well.

Game’s health system is more complex than other games. In this version of game (with respect to original authors of “*The Binding of Isaac*”), the player has 2 health bars:

- Red health – when player is dealt an amount of damage, the maximum capacity of health remains, but the health is reduced. (Like ordinary health system)
- Blue health – this is like an additional layer of health which can’t be regenerated but can increase the whole heart. When player is dealt an amount of damage, the damage will reduce blue health first, then red health. If the whole blue heart (2 hp) is lost, the heart will disappear.

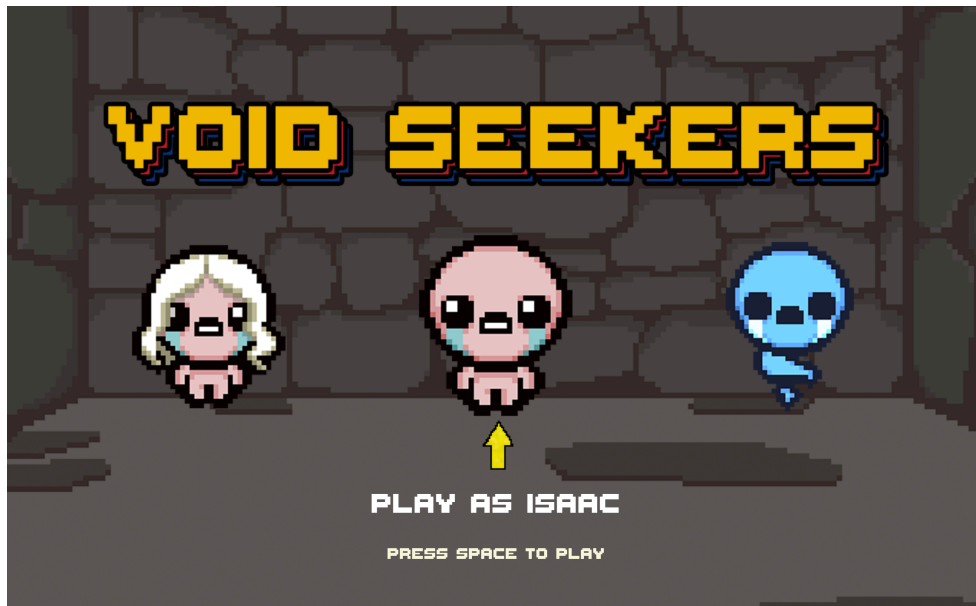
The player has 3 main characters to play:

- Isaac (6 red hp, 0 blue hp, 2 base damage, 5 speed)
- Jared (random hp up to 3 hp, 3 base damage, 5 speed)
- Soul (0 red hp, 2 blue hp, 3 base damage, 7 speed)

Key Binding

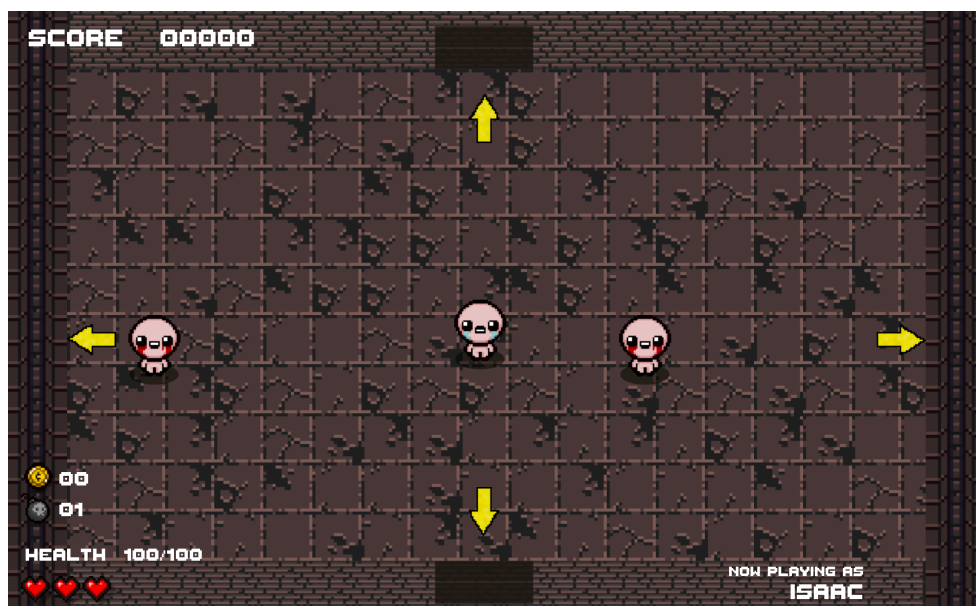
W:	Up
A:	Left / Left Select
S:	Down
D:	Right / Right Select
E:	Drop item
Space:	Attack / Select / Restart
Esc:	Exit Game
M:	Mute Background Music

Main Menu Screen



In Menu Screen, you can press A and D to select main character and press SPACE to play

Game Screen



Top left: Score,
Bottom left: inventory and health status
Bottom right: Playing character

End Screen (player died)



Press SPACE to restart or ESC to exit game

UML Class Diagram



Java Implementations (com.game.void_seekers)

This project contains 9 packages

*The access modifier notations are listed as:

+ (public)
- (private)
(protected)
Underline (static)
Italic (abstract)

1. interface

(I) AffectPlayer

Methods	Functions
+ void affectPlayer(PlayableCharacter player)	States that an object can affect player's statuses.

(I) Attack

Methods	Functions
+ void attack(GameCharacter character)	Attack target with object's damage.

(I) Draw

Methods	Functions
+ void draw()	Render (draw) on canvas.

(I) Usable

Methods	Functions
+ void use()	Use item to applies some effect on the game

2. character.base

(A) CharacterHealth

This Method is universally uses as life points across all characters in this game, including player and enemy. There are 2 types of heaths, red health where taking damage can regenerate by taken collectible red hearts. And blue health where taking damage loses the heart immediately.

Fields and Constructors

- int maxRedHealth	Maximum amount of redHealth character can have
- int redHealth	Amount of current red health
- int maxBlueHealth	Maximum amount of blueHealth character can have
- int blueHealth	Amount of current blue Health
- int maxHealth	Maximum amount of character health (blue and red combine) can have. Note: it is set to default to 20. As this is the maximum amount of health player can have
+ CharaterHealth(int value, int type)	Calls initializeHealth method and addFullHealth
+ CharacterHealth(int value1, int type1, int value2, int type2)	Check if type1 and type2 are equals. If yes, calls initializeHealth method and addFullHealth with parameter passes as value1+value2 and type (as it is the same type). If not, calls initializeFullHealth() method and addFullHealth for both value and type.
+ CharacterHealth(int value)	This constructor is commonly used for setting health for the enemy, calls setMaxHealth method, initializeHealth and setBlueHealth to the value given.

Methods

- void initializeHealth()	Calls setter maxRedHealth and set it to 0 and maxBlueHealth and set it to maxHealth
+ void decreaseRedHealth(int value)	Calls setter redHealth and set current red health deducted by value. Red health value cannot be negative
+ void decreaseBlueHealth(int value)	Calls setter blueHealth and set current blue health deducted by value. Blue health value cannot be negative.
+ void addRedHeartContainers (int value)	Calls setter maxRedHealth and set maxRedHealth added by 2 times the value of the parameter, note that maxRedHealth shall never exceed maxHealth. Set maxBlueHealth by maxHealth deducted by the current maxRedHealth. Also consider if red health and blue health exceed maxHealth, set blue health by difference of maxHealth and maxRedHealth
+ void removeRedHeartContainers(int value)	Set maxRedHealth by curent maxRedHealth deducted by 2 times of intake value. Also, maxRedHealth cannot be odd value and negative value. Set redHealth by the lesser value between redHealth and maxRedHealth. Set maxBlueHealth by maxHealth deducted by the current maxRedHealth.

+ void addFullyRedHeartContainers(int value)	Calls addRedHeartContainers method as passing parameter given. Also calls addHealth method with 2 times the value given and the type of 0.
+ int getRedHeartContainers()	Return the half value of maxRedHealth.
+ void fullyHeal()	Set the current redHealth to current maxRedHealth.
+ void addHealth(int value, int type)	This method uses commonly as the setting the health of any character in the game. Consider type given, if type is 0, set redHealth to current redHealth add by given value. If type is 1, set blueHealth to current blueHealth add by given value.
- void addFullHealth(int value, int type)	This method should be only called when initializing character health. Consider type given, if type is 0, set maxRedHealth to value. Note: maxRedHealth cannot be an odd number. Set redHealth to the given value. And set maxBlueHealth to deduction of maxHealth and maxRedHealth value. If type is 1, just set blueHealth to given value (it is that simple, right?)
- void reduceHealth(int damage)	Decrement damage until it is 0, during that if blueHealth is currently greater than 0, use decreaseBlueHealth method to decrease blueHealth with value of 1. else if redHealth is currently greater than 0, use decreaseRedHealth method to decrease health with the value of 1. If the remaining damage is not 0, end this decrementation.
+ int getAbsoluteTotalHealth()	Value of both redHealth and blueHealth.
+ boolean isDead()	Dies when having no red or blue health left.
Remaining getters & setters	Note: all given fields should never be negative value. RedHealth cannot exceed maxRedHealth and blueHealth cannot exceed maxBlueHealth. Both blue and red health cannot exceed maxHealth. And most importantly, maxRedHealth should not be odd value, as it will cause trouble in healthBar visualization in the game.

(A) GameCharacter extends CharacterHealth implements Draw

This class is base of all character (both enemy and player in the game), this class implements Draw which made object render in UI process.

Fields and Constructors

# Image assetDefaultImage	The base image asset of characters
# Image assetImage	The images that are used for creating animation or triggering event following default image.
# Image assetHurtAnimation	Image when the enemy or character are hurt. Is set default to transparent image. See GameAssets for further information.
# Image assetDeadAnimation	Image indicates that the character has died
# int speed	Speed of character
# int damage	Damage that character will be dealing

# String name	Name of character
# Coordinate coordinate	Location of character in the room
# int width	Width of Character is set to character default size. Gamellogic for further information.
# int height	Height of Character is set to character default size. Gamellogic for further information.
+ GameCharacter()	Calls constructor with respect parameter of value and type that is 1, 1. Set coordinates by using new premodifier. Named this character "Untitled Character"
+ GameCharacter(String name, int health, int x, int y)	Calls constructor with respect parameter of name, health, coordinate
+ GameCharacter(String name, int health, Coordinate coordinate)	Use extended constructor setting health with with type of 0 and given value. Set coordinate and name by using setter methods. And set damage to 0.
+ GameCharacter(String name, int health, int type, Coordinate coordinate)	Use extended constructor setting health and type. Set coordinate and name by its given parameter. And set damage to 0.
+ GameCharacter(String name, int health1, int type1, int health2, int type2, int x, int y)	Use constructor with respective parameter by create new object Coordinate using x, y.
+ GameCharacter(String name, int health1, int type1, int health2, int type2, Coordinate coordinate)	Use extended constructor to setting health with the type of respect to given value and type parameter. Set coordinate and name by its given parameter. And set damage to 0.

Methods

+ drawShadow(GraphicsContext gc)	Store temporary value of GlobalAlpha and Fill using getter method. Then set parameter both GlobalAlpha and Fill using setter, set GlobalAlpha by 0.5 and set Fill by 0.5. Use FillOval with respect parameter x: coordinate.x + double casted height * 0.02f y: coordinate.y + double casted height * 0.8f w: height * 0.8f h: double casted height * 0.333333f Then use temporary store setting globalAlpha and Fill. Note 1: see GraphicsContext for further information Note 2: use floating decimal instead of dividing to increase speed
+ void setCoordinate(int x, int y)	Set coordinate value of x to x given, and value of y to y given.
+ void setRedHealth(int health, int maxHealth)	Set maxRedHealth and redHealth consecutively using method from extended class.
+ void setBlueHealth(int health, int maxHealth)	Set maxBlueHealth and blueHealth consecutively using method from extended class.
The remaining getters & setters	

(A) PlayableCharacter extends GameCharacter implements Attack

This is the class controlling the player and player stats.

Fields and Constructors

- int fireRate	How many shot fired in defend thread time
- int luck	Luck of the player, just told how lucky the player is

- int bombs	Number of bombs player collected
- int coins	Number of coins player collected
- Active active	Active item player holds
- Trinket trinket	Trinket that player holds
- ArrayList<Passive> passives	Passives item that player collected, with default is none
- ArrayList<EffectItem> pickedEffectItem	All previous picked up EffectItem, with default is none
+ PlayableCharacter(String name, int health, Coordinate coordinate, int damage, int speed, int fireRate, int luck)	Use extended constructor with name, health, coordinate. Use setCharacterStat method. then set player number of bomb and con using setter to both 0.
+ PlayableCharacter(String name, int health, int type, int x, int y)	Set all parameter using extended constructor
+ PlayableCharacter(String name, int health1, int type1, int health2, int type2, int x, int y)	Set all parameter using extended constructor
+ PlayableCharacter()	setName to "Untitled Player" and use setCharacterStats().

Methods

+ void attack(GameCharacter gameCharacter)	Reduce character health by player damage
+ void setCharacterStats()	Set damage, speed, fireRate , luck to 1, 1, 1, 0 respectively
+ void setCharacterStats(int damage, int speed, int fireRate, iny luck)	Set damage, speed, fireRate, luck by its given parameters.
+ Item addEffectItem(EffectItem item)	Use addEffectToPickedItem method Consider item type, If item is Passive, use addPassive method to add item to the passive arrayList If item is Active, consider active slot. If active slot is empty (using isActiveSlotEmpty() method) set active by its given item. If not, return what has been returned in swapActiveItem method. If item is Trinket, consider trinket slot. If trinket slot is empty (using isTrinketSlotEmpty() method), if it is set trinket by given item. If not, return what has been returned in swapTrinket method.
+ Item add(Item item)	Consider item. If item is pockelItem. Then if item is Coin. Add coin to player inventory. If item is Bomb, add bomb to player inventory. If item is battery and active slot is empty (use isActiveSlotEmpty() method) set active by increasing active charge by battery power. If item is Heart. If heart is red and redHeartContainers is empty, use addHealth by item value, check for blue heart with same as redHealth condition and add blueHealth as redHealth (use isExistEmpty__HeartContainers() method) If item is EffectItem, calls addEffectItem method and return what have been returned in addEffectItem method. Otherwise, return null if the condition is not as follows form addEffectItem method.
+ dropActiveItem()	Return held active item, then set active slot to null.

+ swapActiveItem(Active newItem)	Swap active and newer active item, return held Active item.
+ dropTrinket()	Return held trinket item, then set trinket slot to null.
+ swapTrinket(Trinket newItem)	Swap trinket and newer trinket, return held trinket.
+ boolean isActiveSlotEmpty()	Check if active item slot is empty.
+ boolean isTrinketSlotEmpty()	Check if trinket slot is empty.
+ boolean isAlreadyPicked(EffectItem item)	Check if EffectItem has been previously collected.
+ void addEffectToPickedItem(EffectItem item)	If pickedUpEffectItem do not contain given item, add effect item to pickedUpEffectItem.
+ int getNumberOfPickedEffectItem()	Self explanatory
+ void addPassive(Passive item)	If given item do not contain in passives item. Add passive to passives ArrayList.
+ int getNumberOfPassiveItem()	Self explanatory.
+ boolean hasBomb()	Self explanatory.
+ boolean hasCoin()	Self explanatory
The remaining getters & setters	

(A) EnemyCharacter extends GameCharacter

All enemies character of the game.

Fields and Constructor

- boolean isAttacking	Self explanatory
+ EnemyCharacter(String name, int health, Coordinates coordinate)	Use extended constructor by its given parameters.
+ EnemyCharacter(String name, int health, int x, int y)	Use extended constructor by its given parameters.
+ EnemyCharacter()	Set name of this enemy to "Untitled Enemy"

Methods

Fields' getters & setters	
---------------------------	--

3. character.derived

Class of all enemy and player in this game.

(C) PlayerIsaac extends PlayableCharacter implements Attack

This is the default character, having 6 red hearts and base stats across all characters.

Fields and Constructors

+ PlayerIsaac()	Use extends constructor setting Name: "Isaac" Health: 6 red hearts And set x, y to 0,0 as starting location Set AssetDefaultImage by using GameAssets.loadImage of GameAssets of isaacURL and height extended field. Set assetImage using super getAssetDefaultImage method. Set damage and speed by 2, 5 respectively.
-----------------	--

Methods

+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate oif x and y.
+ void attack(GameCharacter character)	Reduce character health by this player damage

(C) PlayerSoul

This character is harder to play Having 6 blue non-regenerative hearts, but with better stat than the base character.

Fields and Constructors

+ PlayerSoul()	Use extends constructor setting Name: "Soul" Health: 6 blue hearts And set x, y to 0,0 as starting location Set AssetDefaultImage by using GameAssets.loadImage of GameAssets of isaacURL and height extended field. Set assetImage using super getAssetDefaultImage method. Set damage and speed by 3, 7 respectively.
----------------	--

Methods

+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate oif x and y.
+ void attack(GameCharacter character)	Reduce character health by this player damage

(C) PlayerJared

This character is challenging. Having base stats and randomized heart that equls to 2 to 6 health.

Fields and Constructors

- final int beginHealth	This value is set to 2 times RandomIntRange(1, 3).getResult() from GameUtils
- final int hp	This value is set to default as RandomIntRange from 0 to beginHealth
+ PlayerJared()	Use extends constructor setting Name: "Jared" Health: 2 - 6 (randomized blue and red combines) use super constructor that allows setting 2 health

	<p>method. By first type as hp and second type by beginHealth – hp. And set x, y to 0,0 as starting location Set AssetDefaultImage by using GameAssets.loadImage of GameAssets of isaacURL and height extended field. Set assetImage using super getAssetDefaultImage method. Set damage and speed by 2, 5 respectively.</p>
--	--

Methods

+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate of x and y.
+ void attack(GameCharacter character)	Reduce character health by this player damage

(C) EnemyGaper

This character is enemy version of Isaac. Having base stat, can be deadly if having too many amount.

Fields and Constructors

+ EnemyGaper()	<p>Use extends constructor setting Name: "Gaper" Health: 10 (not given type but blue health as default) And set x, y to 0,0 as starting location Set AssetDefaultImage by using GameAssets.loadImage of GameAssets of isaacURL and height extended field. Set assetImage using super getAssetDefaultImage method. Set assetHurtAnimation by using loadImage gaperHurtURL and height from GameAssets Set assetDeadAnimation by the same method. Set damage and speed by 1, 1 respectively</p>
----------------	--

Methods

+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate of x and y.
+ void attack(GameCharacter character)	Reduce character health by this player damage

4. item.base

All types of item exist within the game.

(A) Item implements Pickable, Draw

Item should have name and Image as its components. All items should nbe pickable by player.

Fields and Constructors

# String name	Name of the item
# Image assetImage	Image of the item
# Coordinates coordinate	Location of item in the room
# int size	Each item may have various sizes
+ Item(String name)	Set name of the item

Methods

Fields all getters and setters	
--------------------------------	--

(A) PocketItem extends Item implements Pickable

Item that can be store directly or collect as in player inventory

Constructors

+ PocketItem(String name)	Use extends constructor settin field of name. And use setSize method to set size of PocketItem as 60
---------------------------	--

(A) EffectItem

These are type oif item that aree store in each special place in Player inventory. These item provide stats which are permanent or temporary to the player.

Fields and Constructors

- String description	Self explanatory
+ EffectItem(String name, String description)	Use extended construcor setting name to the item. Set description by its given fields. And set size of this item to 75
+ EffectItem(String name)	Use extended construcor setting name to the item. And set size of this item to 75

Methods

+ String toString()	Return String in format of name with new line followed by description.
The field getter & setter	

(A) Active extends EffectItem implements Usable

Active item is type of item having charge and maxCharge. This item is store within active slot. When the charge is full, player can press the spacebar button to activate its effect. Charge can be gather by using battery found randomly as player progress the level.

Fields and Constructors

# int charge	Current charge the item has
# int maxChage	The amount of charge in order to activate the item effects.
+ Active(String name, String description, int maxCharge)	Use extended constructor setting name and description. Set charge to 0, and set maxCharge equals to given parameter.
+ Active(Stirng name, int maxCharge)	Use extended constructor setting name and description (with blank string). Set charge to 0, and set maxCharge equals to given parameter.

Methods

+ boolean isAvaliable()	Return true if charge is equals or more than maxCharge.
-------------------------	---

+ void use()	If Active item is available, decrease charge by maxCharge.
The fields getters & setters	Note: charge should not exceed maxCharge in this current game state.

(A) Passive extends EffectItem implements Pickable

Passive item is the type of item that change player stats or appearance, usually involved with affect player and game play. Passive item once pick-up, it can not remove from player inventory by normal means.

Constructors

+ Passive(String name, String description)	Use extended constructor setting name and description.
+ Passive(String name)	Use extended constructor setting the name (description is set with "" blank string)

(A) Trinket extends EffectItem

Trinkets item provide temporary stats while holding the item, this item is stored in Trinket slot and can be remove.

Constructors

+ Trinket(String name, String description)	Use extended constructor setting name and description.
+ Trinket(String name)	Use extended constructor setting the name (description is set with "" blank string)

5. item.derived

The items that are obtains and uses within a game.

(C) Battery extends PocketItem

This is type of item that can be collect when player have active item and the charge of active item is not full. Player can collect this item to increase current charge of the active item.

Fields and Constructors

- int power	Amount of power that increase charge of active item
+ Battery(int type)	If type given is 0, name this battery "normalBattery" else if it is 1, name this battery "largeBattery" Set power, if it is normal battery, the power is 2, and if it is large battery power should be large value (in this case power of large battery is set to 999). And add assets by its type usingt addAssets method.
+ Battery()	Set battery name to "battery" With its power to 2. And add assets to type of 0

Methods

- void addAssets(int type)	If type given is 0 set assetImage by loadImage and get normalBatteryAssets in GameAssets with its size. Else set assets the same method as before, but use assets of largeBattery.
+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate of x and y.
The fields getter & setter	

(C) Bomb extends PocketItem

This is the type of item, where it can be use and explode after given some amount of time, this both damage player and enemy.

Fields and Constructors

- int value	Value of bomb drops
+ Bomb()	Set the name as "bomb" SetValue of the bomb to 1 Set assetImage by using loadImage of bombURL and size by using GameAssets.

Methods

+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate of x and y.
The field getter & setter	

(C) Coin extends PocketItem

In game currency that can be collected form the room and the enemy drop.

Fields and Constructors

- int value	Value of currencty each coin holds
+ Coin(int value)	Set name as "coin" Set value by its given value Set assets by its given assets

Methods

- void addAssets(int value)	Set assets depends on the given value, If value is 5, set assets to nickelURL If value is 10, set asstes to dimeURI Else, set assets to pennyURL Set assetImage by using loadImage of ___URL and size by using GameAssets.
+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate of x and y.
The field getter and setter	Note: set value respectively to the given value 5 -> nickel 10 -> dime Else -> penny Coins only exist with these 3 possibly value only.

(C) Heart extends PocketItem

Health that can be pick up to increase hp and survivability

Fields and Constructors

- int type	Type of hearts (now should be only 0, 1. 0 being, redHeart And 1, being blueHeart)
- int value	Value of heart (now should only be 1, 2 1, being half a heart And 2, full heart.
+ Heart(String name, int value, int type)	Set name by its given name Set value and type Set the assets of hearts. See both rows above for condition of setting the assets. Set assetImage by using loadImage of ___URL and size by using GameAssets.

Methods

+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate of x and y.
The fields getters &s setters	

(C) Exploding extends PocketItem

This is not an Item, but considering it as an exploding bomb when use and drop by the player.

Constructor

+ Exploding()	Set name as "exploding" SetSize of this object as 75 Set assetImage by using loadImage of bombURL and size by using GameAssets.
---------------	--

Methods

+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate of x and y.
---------------	--

6. item.passive

This package hold collections of all Passive item, its affect are as follows:

(C) LonelyEye extends EffectItem implements AffectPlayer

Just Lonely eye where it does wading alone after previous owner left it. Just take it with you for it to experience the outer world.

This item increase player speed and damage by 1.

“I don't mean to be rude

There's things in myself that I see in you

Lonely eyes

She had those lonely eyes

I only know 'cause I have them too

Lonely eyes

No, you don't have to hide

The things you feel inside, I feel too” - Lauv, Lonely Eyes

Constructors

+ LonelyEye()	Set name and description as “Lonely eye”, “Become one with him” respectively. Set assetImage by using loadImage of lonelyEyeURL and size by using GameAssets.
---------------	--

Methods

+ void affectPlayer(PlayableCharacter player)	Increase damage and speed of the player by 1
+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate of x and y.

(C) EternalBlessing extends Passive implements AffectPlayer

Gift of the devout, your strong belief may guide you through hardships.

This item double damage, speed, fireRate, and add luck by 10.

“a faithful man shall abound with blessings” - Proverbs 28:20

Constructors

+ LonelyEye()	Set name and description as “Eternal Blessing”, and “a faithful man shall abound with blessings” respectively. Set assetImage by using loadImage of eternalBlessingURL and size by using GameAssets.
---------------	---

Methods

+ void affectPlayer(PlayableCharacter player)	Increase damage and speed of the player by 1
+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate of x and y.

7. item.active

This package contains the following Active Items as follows:

(C) BookOfRage

"A soft answer turns away wrath, but a harsh word stirs up anger.". This item multiplies damage of the user when uses by 2.

Constructors

+ BookOfRage()	Set name and description as "Book of Rage", and "the gift of wrath" respectively. Set assetImage by using loadImage of bookOfRageURL and size by using GameAssets.
----------------	---

Methods

+ void affectPlayer(PlayableCharacter player)	Double the damage of the player by 2
+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate of x and y.

8. item.trinket

This package contains the following Trinket Items as follows:

(C) LingeringFeather

This item increase speed of the player by 10.

Constructor

+ LingerngFeather()	Set name and description as "Linding Feather", and "Lighter weight" respectively. Set assetImage by using loadImage of lingeringFeatherURL and size by using GameAssets.
---------------------	---

Methods

+ void affectPlayer(PlayableCharacter player)	Increase player speed by 10.
+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate of x and y.

9. obstacle.base

This package contains object that obstruct player and enemy.

(A) Obstacle implement Draw

This class serve as base template for all obstacle

Fields and Constructors

# int height	Height value the higher, it obstruct the character walkable path
# String name	Name of the obstacle
# Image image	Texture of the obstacle
# int size	Size of the obstacle
# Coordinates coordinate	Location of the obstacle in a room
+ Obstacle(int height)	Set the given height

Methods

The fields getters & setters	
------------------------------	--

10. obstacle.derived

This package contains fully usable object that obstruct the player path.

(C) Bush extends Obstacle implements Draw

A greenish bush, obstruct player vision of the stage. Can not be destroy

Constructor

+ Bush(int type)	Set height by 1 Set the assets of bush. Set assetImage by using loadImage of ___URL and size by using GameAssets. Assets depends on the type given. 0 -> bush1URL Else -> bush2URL
------------------	--

Method

+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate of x and y.
---------------	---

(C) Crate extends Obstacle implements Draw

A obstacle that are destory by projectile.

Constructor

+ Crate()	Set height by 1 Set the size of size character default / 2 Set assetImage by using loadImage of crateURL and size by using GameAssets.
-----------	--

Method

+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate of x and y.
---------------	---

(C) Mud extends Obstacle

A obstacle that when player walk over it, it slow down the speed of player movement by half

Constructor

+ Mud()	Set height by 0 Set the size of size character
---------	---

	Set assetImage by using loadImage of mudURL and size by using GameAssets.
--	---

Method

+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate of x and y.
---------------	---

(C) Spike extends Obstacle

A onstacle that damage player when walk over it, damage are equivalent to 2 damage or 1 heart damage.

Constructor

+ Spike(int type)	Set height by 0 Set assetImage by using loadImage of ___URL and size by using GameAssets. Assets depends on the type given. 0 -> spike1URL Else -> spike2URL
-------------------	---

Method

+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate of x and y.
---------------	---

11. projectile.base

This package contains the base template for all projectile in the game.

(A) Projectile implements Draw

*The details of this class are already explained in package

Fields and Constructor

# Image image	A texture for the projectile
# int size	Size of projectile
# int speed	Speed of projectile, default is 1
# boolean[] directions	Projectile direction stored default as array as {false, false, false, false} and set to true as the projectile direction
# int damage	Amount of damage projectile deal to character
# String name	Name of the projectile
# Coordinates coordinate	Location of the projectile
+ Projectile(String name, int damage, int speed, Coordinates coordinate)	Set all given to fields using getter methods
+ Projectile(String name, Coordinates coordinate)	Use suitable constructor, set name and coordinate by given, else set damage and speed to 1.
+ Projectile(String name, int damage, Coordinates coordinate)	Use suitable constructor, set name, damage, coordinate as given, else set speed to 1.
+ Projectile(int damage, Coordinates coordinate)	Use suitable constructor, set damage and coordinate as given, else set name as "Default" and speed to 1.

Methods

The fields getters and setters	
--------------------------------	--

12. projectile.derived

This package contains various type of projectile. In which, each projectile have unique ability or effect applies to the character.

(C) NormalProjectile extends Projectile implements Draw, AffectPlayer

Normal projectile, death 1 damage on contact or half a heart as damage.

Fields and Constructors

+ public NormalProjectile(Coordinates coordinate)	Set name of the projectile as "Normal" and set coordinate as given. Set speed to 5, damage to 1 Set size to the character size default (which can be obtained from GamLogic) by 75 % Set image by using loadImage method and texture from iGameAssets. Set assets as normalProjectileURL
---	--

Methods

+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate of x and y.
+ void affectPlayer(PlayableCharacter player)	Reduce player health by incoming projectile

(C) RedProjectile extends Projectile implements Draw, AffectPlayer

Fields and Constructors

Red projectile, death 2 damage on contact or full heart damage.

+ public NormalProjectile(Coordinates coordinate)	Set name of the projectile as "Red" and set coordinate as given. Set speed to 7, damage to 2 Set size to the character size default (which can be obtained from GamLogic) by 80 % Set image by using loadImage method and texture from iGameAssets. Set assets as redProjectileURL
---	--

Methods

+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with parameter of getAssetImage () and coordinate of x and y.
+ void affectPlayer(PlayableCharacter player)	Reduce player health by incoming projectile

(C) WhiteProjectile extends Projectile implements Draw, AffectPlayer

Fields and Constructors

White projectile, death 1 damage and slow down character on contact.

+ public NormalProjectile(Coordinates coordinate)	Set name of the projectile as "White" and set coordinate as given. Set speed to 4, to 1 Set size to the character size default (which can be obtained from GamLogic) by 50 % Set image by using loadImage method and texture from iGameAssets. Set assets as whiteProjectileURL
---	---

16.1.2 Methods

+ void draw()	Get graphic context by GameLogic and draw graphic context by using drawImage method with
---------------	--

	parameter of getAssetImage () and coordinate of x and y.
+ void affectPlayer(PlayableCharacter player)	Reduce player speed by incoming projectile

13. room.base

This package contains foundation of all rooms in the game.

(A) Room implements Draw

*details of this class is the same as in the package

Fields and Constructors

# int difficulty	Difficulty for each room, increase when player traverse more room
# Image[][] floorTiles	Tile of floor where obstacle spawns
# URL topWallURL	Assets of top wall of the room
# URL bottomWallURL	Assets of bottom wall of the room
# URL leftWallURL	Assets of left wall of the room
# URL rightWallURL	Assets of right wall of the room
# ArrayList<EnemyCharacter> enemyCharacters	Contains all enemy in the room
# ArrayList<Obstacle> obstacles	Contains all obstacle in the room
# ArrayList<Item> items	Contains all items in the room
# ArrayList<Projectile> projectiles	Contains all projectile in the room
# Room rightRoom	Store right room of this current room
# Room leftRoom	Store left room of this current room
# Room topRoom	Store top room of this current room
# Room bottomRoom	Store bottom room of this current room
# Image leftImage	Assets Left wall of the room
# Image rightImage	Assets Right wall of the room
# Image topImage	Assets Top wall of the room
# Image bottomImage	Assets Bottom wall of the room
+ Room(int difficulty)	Set difficulty and all left, right, top, bottom assets of the room. Assets are obtained from GameAssets. See game Assets for further information.
+ Room(int difficulty, URL topWallURL, URL bottomWallURL, URL leftWallURL, URL rightWallURL)	Set difficulty and every wall asset to arguments' assets. (For custom walls)

Methods

+ void generateRoom()	This method has no body and will be implement later. 😊
+ void draw()	Render top, bottom, left, and right wall and floor tiles.
The fields getter & setter for all fields	

(E) RoomDirection

This class provided direction of the room:

TOP	Top direction of the room
BOTTOM	Bottom direction of the room
LEFT	Left direction of the room
RIGHT	Right direction of the room

14. room.derived

This package contains all room types within the game, the room types are as follows

(A) NormalRoom extends Room

This class representm the difficulty and the assets for the room

Constructors

+ NormalRoom(int difficulty)	Set difficulty from extended class of constructor
------------------------------	---

1.1.1. Methods

+ NormalRoom(int difficulty, URL topWallURL, URL bottomWallURL, URL leftWallURL, URL rightWallURL)	Set all given parameters from extended class of constructor
--	---

(C) SpawnRoom extends NormalRoom

This room is type of the room that enemy will spawn in

1.1.2. Constructors

+ SpawnRoom(int diiculty)	Set difficulty from extended class of constructor
---------------------------	---

1.1.3. Methods

+ generateRoom()	Calls generateObstacleMethod Generate 3 bombs and with random coordinate by using method coordinateRandomizer from GameUtil Add bomb to items ArrayList Add enemy Bobo amd Necromacer through getEnemyCharacters method
------------------	--

(C) TreasureRoom extends NormalRoom

1.1.4. Constructors

+ treasureRoom(int diiculty)	Set difficulty from extended class of constructor
------------------------------	---

1.1.5. Methods

+ generateRoom()	Create pedastal from Crate object Set crate at the center of the room, use GameLogic middle_center for this method Add new pedestal to obstacle ArrayList Create new random effectItem from getRandomEffectItem function from GameUtils Set item coordionate directly above pedastal size Add items to items ArrayList
------------------	---

(C) EnemyRoom extends NormalRoom

1.1.6. Constructors

+ treasureRoom(int diiculty)	Set difficulty from extended class of constructor
------------------------------	---

1.1.7. Methods

+ generateRoom()	Calls generateEnemies method Generate bomb, set random coordinate, then add it to items ArrayList Get randomEffectItem, set Random coordinate, then add it to items ArrayList
------------------	---

(C) BossRoom extends NormalRoom

1.1.8. Constructors

+ treasureRoom(int diiculty)	Set difficulty from extended class of constructor
------------------------------	---

1.1.9. Methods

+ generateRoom()	Get boss from enemyCharacter with get difficulty as paramter form GameUtils Set boss name by add boss to the beginning of the name Set height increase by 2 + (difficulty / 10) times the orifginal charater amount
------------------	---

	Set damage output of the boss to 2 + difficulty Set speed increase its original by difficulty Add health boss by bossMaxHealth with u2 + (difficulty / 10) times the original character health amount Set coordinate of boss as the center of the room using GameLogic Middle_Center and add x: 0, y: 0 Method
--	--

15. tools

This package contains useful tools that being use across the game

(C) Coordinates implements Clonnable

This coordinates class store position x, y in which uses to set position of all object in the game

1.1.10. Constructors

+ int x	x location of the plane
+ int y	y location of the plane
+ Coordinates()	Set x: 0 and y: 0
+ Coordinates(int z)	Set x: z and y: z
+ Coordinates(Coordinates coordinates)	Use appropriate constructor as passes coordinates.x and coordinates.y
+ Coordinates(int x, int y)	Set x as field and y as field

1.1.11. Methods

+ double getDistance(Coordinates other)	Get distance using Pythagoras theorem
+ Coordinates add(Coordinates other)	Add x by other.x and y by other.y and return as new Coordinates object
+ Coordinates add(int x, int y)	Add x and y by the given parameter respectively
+ Coordinates minus(Coordinates other)	Minus x by other.x and y by other.y and return as new Coordinates object
+ Coordinates minus(int x, int y)	Minuis x and y by the given parameter respectively
+ String toString()	Return String asd format of (x, y)
+ Coordinates clone()	Return new Coordinate object as a clone of previous object

(C) RandomIntRange

This function generate random interger in the given range

1.1.12. Constructors

+ int result	Result of random generated number
+ int rMin	Lower bound of random generated number
+ int rMax	Upper bound of random generated number
+ RandomIntRange()	*Left Blank
+ RandomIntRange(int min, int max)	Set rMin and rMax by min and max from given input. Then set the result as min + (int) (Math.random * (max - min) + 1)

1.1.13. Methods

+ int getResult()	Return the result
+ void createNewResult(int max, int min)	Set rMin and rMax by min and max from given input. Then set the result as min + (int) (Math.random * (max - min) + 1)
+ int next()	Calls createNewResult(rMin, rMax) then return the result

16. logic

Contains the logic, event loops, polling, timer, background media, and current statuses of the game. Singleton Design Pattern is applied.

(C) final GameLogic

*This package is already given by the package information

1.1.14. Constructors and Fields

*The constants, please see the reference	-
- <u>final GameLogic instance</u>	Use to get the instance across all classes, set to default new GameLogic()
- GameState state	Set constant gameState menu instance
- <u>final Set<EffectItem> usedItem</u>	Set default value as HashSet
- Stage stage	Stage of JavaFX, using for linking assets
- AbstractScene currentScene	Current scene for JavaFX to render
- GameScene gameScene	Game JavaFX scene
- MenuScene menuScene	Menu JavFX scene
- EndGameScene endGameScene	Player Died JavaFX scene
- HealthBar healthBar	Health bar (AbstractScene)
- InventoryBar inventoryBar	Inventory bar (AbstractScene)
- Pane rootPane	JavaFX Root Pane
- ActiveBar activeBar	Active item hot bar (AbstractScene)
- TrinketBar trinketBar	Trinket item hot bar (AbstractScene)
- <u>int score</u>	Default as 0, score of the player during the game session.
- PlayableCharacter character	Main character of the game (player)
- volatile Room currentRoom	Current room that the player is currently in
- <u>final BooleanProperty wPressed</u>	Pressed: detect when player presses/holds the button Flag: trigger once in thread on button release. BooleanProperty – a wrapped Boolean type with getter and setter.
- <u>final BooleanProperty aPressed</u>	
- <u>final BooleanProperty sPressed</u>	
- <u>final BooleanProperty dPressed</u>	
- <u>final BooleanProperty ePressed</u>	
- <u>final BooleanProperty spacePressed</u>	
- <u>final BooleanProperty escPressed</u>	
- <u>final BooleanProperty aFlag</u>	
- <u>final BooleanProperty eFlag</u>	
- <u>final BooleanProperty dFlag</u>	
- <u>final BooleanProperty spaceFlag</u>	
+ GameEvent gameEvent	Game event, called in threaded loop
+ EnemyEvent enemyEvent	Enemy event, called in threaded loop
+ AnimationTimer pollingLoop	Main loop which refreshes every game frames. Useful for polling inputs and rendering movements with speed attribute. (AnimationTimer is more accurate than Thread's sleep)
+ Thread gameLoop	Thread which gameEvent is called in a loop.
+ Thread enemyLoop	Thread which enemyLoop is called in a loop.

+ GameLogic()	Constructor. Start media and setup polling loop.
---------------	--

1.1.15. Methods

+ GameLogic getInstance()	Return the singleton instance of the game.
+ void init(PlayableCharacter playableCharacter)	Initialize game state with score of 0, initialize game events and loops, initialize spawn room and player's new character.
+ void endGame()	End game state when player died. Stop playing background music, transfer state to end game state, stop every game thread except the polling thread which runs forever.
+ void exit()	Stop threads and exit application.
+ void pollInputsInMenu()	Poll key inputs in menu, available keys are ESC, A, D, and SPACE.
+ void pollInputsInEnd()	Poll key inputs in end game state, available keys are ESC and SPACE.
+ void pollInputsInGame()	Poll key inputs in game state, available keys are ESC, W, A, S, D, E, and SPACE.
+ void shootProjectile(Projectile projectile, boolean[] directions)	Spawn projectiles and let it traverse across the frame in any of 8 directions until it hits an enemy or any of 4 walls.
+ void pollObstacles()	Scanning collision with player. If the collision happens, the collided obstacle is removed from the room and the frame.
+ void pollProjectiles()	Scanning collisions of projectiles with enemies or walls. (Enemy can take damage from projectiles)
+ void useBomb()	Spawn bomb in the room, counting down until explosion.
+ void explode(Exploding exp)	Explodes the bomb, damaging any game characters in range.
+ void removeDeadEnemies(ArrayList<EnemyCharacter > enemies)	Remove marked-dead enemies from the room and the frame.
+ void enemiesTargetPlayer()	Enemies targeting (walking towards) player. Called in polling loop.
+ void attack(Attack attackableCharacter, GameCharacter characterToAttack)	Check collision between every pair of enemies and player, then start attacking and invincible frame of the player. Play hurting animation.
+ void transitionToNextRoom(RoomDirection direction)	Create new room node according of the direction of next room. Point next room's opposite direction room to this room. (Link room nodes), then play fade transition.
+ GraphicsContext getGraphicsContext()	Return global JavaFX graphics context.
+ void keyHandler(KeyEvent e, boolean property)	Register keys when pressed or released accordingly.
+ void keyPressedHandler(KeyEvent e)	Register pressed keys.
+ void keyReleasedHandler(KeyEvent e)	Register released (latched) keys.
+ void switchScene(AbstractScene nextScene)	Switch scene from current scene to next scene, playing fade transition. Set current scene to next scene.

(C) GameEvent implements Runnable
Fields and Constructors

- volatile boolean isRunning	Default value as true
------------------------------	-----------------------

Methods

+ void run()	Construct and start low health warning watcher thread, for event loop: check for item collision, remove dead enemies, and check for door collision every constant interval (check slower than AnimationTimer).
- void doorCollisionEvent()	Check if player collides with any doors. If player collides with a door, transition player to the next room with appropriate direction.
- void itemCollisionEvent()	Check if player collides with any item entities. If player collides with an item, the player will pick up that item and the item will be removed from the room and the frame.
- ArrayList<EnemyCharacter> getDeadEnemies()	Get all enemies in the room with zero or less health (should not be less than zero in any scenarios).
+ void kill()	Stop event loop, use with thread interrupting

(C) GameUtils

Fields and Constructors

- final Image[] FLOOR_TILES	List of tile image assets for randomizing floor tiles later.
- final ArrayList<Class<? extends Item>> itemList	List of active items (Class types, not an object). Initialize these classes later when randomizing.

Methods

- Obstacle[] newObstacles()	List of obstacles template for copying when randomizing later.
+ boolean inBound(Coordinates coordinates, int width, int height)	Check if the coordinate is within bounding box of floor (where entities can freely walk and spawn).
+ boolean outOfBound(Coordinates coordinates, int width, int height)	Not in bound.
+ boolean isCollided(GameCharacter gc1, GameCharacter gc2)	Check if two characters are collided (overlapped).
+ boolean isCollided(GameCharacter gc, Coordinates hitBoxCoord, Coordinates hitBoxSize)	Check if character is collided with custom hitbox.
+ boolean isCollided(GameCharacter gc, Item item)	Check if character is collided with item.
+ Room randomizeNextRoom()	Generate next room
+ ArrayList<Pair<Coordinates, Obstacle>> obstacleRandomizer(int maxRange)	Generate list of obstacles and coordinates.
+ Coordinates coordinatesRandomizer()	Generate random coordinate.
+ Image[][] tilesRandomizer(int sizeX, int sizeY)	Generate random tiles for rendering floor. Generate once when generating new room.
+ EffectItem getRandomEffectItem()	Generate random effect items with no duplicates across the whole game.
+ boolean isWithinRange(Coordinates c1, Coordinates c2, int range)	Check if two coordinates are within range.
+ boolean isWithinRange(GameCharacter gc1, GameCharacter gc2, int range)	Check if two characters are within range.
+ boolean isWithinRange(Item item1, GameCharacter gc2, int range)	Check if item and character are within range.

+ void addToUsed(EffectItem Item)	Add the effect item to used items to prevent item spawn duplication.
+ void resetUsedItem()	Reset used items for new game.
+ boolean hasBeenUsed(EffectItem Item)	Check if the effect item has been used.

(C) GameAssets

This class contains a bunch of assets and linkages.

Fields and Constructors

* Most fields are constant (URL and asset linkages to resources and original assets path), please see reference to original code for more information.	-
--	---

Methods

+ Image loadImage(URL url, double fitLength)	Load image from URL with fit height.
+ Image loadImage(URL url, double fitLength, boolean mode)	Load image from URL with fit height (mode=true) or width (mode=false).
+ Image loadImage(URL url, double fitWidth, double fitHeight)	Load image from URL with fit height and width: scale preservation is disabled.
+ URL getURL(String relativePath)	Get URL from relative path from resources folder.
+ Image fromPath(String relativePath, double fitLength;	Load image from relative path with fit height.
+ Font loadGameFont(double size)	Load default game font to font object with font size in px.

(C) EnemyEvent implements Runnable

Fields and Constructors

- volatile boolean isRunning	Self-explanatory, default is true
------------------------------	-----------------------------------

Methods

+ void run()	Check for enemy attacking and set attacking cooldown time for enemy to not let enemy attack player instantly, allowing player to back off before enemy attacks the player.
+ void kill()	Stop event loop, use with thread interrupting

(E) GameState

This class provided the scene of the state game currently in:

MENU	State where user can select the character
ONGOING	State where user play the game
END	State where player got game over

17. render

This package is associated with rendering the scene of all user interfaces in all the state of the game.

(A) AbstractScene extends Scene

Just an ordinary JavaFX scene with automatic canvas initialization and redraw abstract method.

1.1.16. Fields and Constructors

# final Canvas canvas	
-----------------------	--

+ AbstractScene(Pane parent, int width, int height)	Use constructor of the extended class with the given parent. Add new Canvas from given height and width to parent by adding children.
+ AbstractScene(Pane parent, double width, double height)	As same as above.

Methods

+ void redraw()	Each scene will have this method to draw / update each scene passes. Implement later in inherited classes.
+ Canvas getCanvas()	Return canvas.

(C) ActiveBar extends AbstractScene

Fields and Constructors

- final Image defaultImage	Set default as transparent image, using transparentImage form gameAssets.
- Image image	Assets of this activityBar
+ ActiveBar(Pane parent, double width, double height)	Use extended constructor set all given value to it.

Methods

+ void redraw()	Draw player's current active item with name, charges, and maximum charges on the canvas.
Field getters and setters	

(C) HealthBar extends AbstractScene

Fields and Constructors

- Boolean strobeHealth	This is true when red heart is low, the health bar will blink. Set default to false.
- Color healthColor	Color of the health text. The color is set to red when player is at low health.
- final Image assetsEmptyHealth	Use GameAssets to load asset of the given name URL with fit length of 40.
- final Image assetsFullRedHealth	
- final Image assetsFullBlueHealth	
- final Image assetsHalfRedHealth	
- final Image assetsHalfBlueHealth	
+ HealthBar(Pane parent, double width, double height)	Use extended constructor set parent, width, height as value given. Set health color to white.

Methods

+ void redraw()	Draw health (heart) icon and health status texts according to player's red and blue health. Strobe heart icons and set health status text to red when player's health is critical.
- int playerHealth()	Get maxRed, current red and blue health from GameLogix getCharacter get respectively method. If sum of both current blue and red is zero return 0, else return $100 * \frac{(\text{currentRedHealth} + \text{currentBlueHealth})}{(\text{maxRedHealth} + \text{currentBlueHealth})}$
The fields getters and setters	

(C) InventoryBar extends AbstractScene

Fields and Constructors

- final Image assetBomb	Use GameAssets to load asset of the given name URL with fit length of 40.
-------------------------	---

- final Image assetCoin	Use GameAssets to load asset of the given name URL with fit length of 40.
+ InventoryBar(Pane parent, double width, double height)	Use extended constructor set all given value to it.

Methods

+ void redraw()	Draw coin and bomb icons on canvas. Draw text indicating coin and bomb value accordingly.
+ Canvas getCanvas()	Return canvas

(C) TrinketBar extends AbstractScene

Fields and Constructors

- final Image defaultImage	Use GameAssets to load asset of the given name URL with fit length of 40.
- Image image	Image of trinket
+ TrinketBar(Pane parent, double width, double height)	Use extended constructor set all given value to it. Set Image with default Image

Methods

+ void redraw()	Draw player's current trinket item with name.
The field getters and setters	

(C) MenuScene extends AbstractScene

Fields and Constructors

- final String[] selections	Contains all playable character name with default value is {"Isaac", "Soul", "Jared"}
- int currentSelection	Store value of current character selected
+ MenuScene(Pane parent, double width, double height)	Use extended constructor set all given value to it. Set currentSelection to 1

Methods

+ void redraw()	Draw game menu background, arrow selection key, and indicator texts.
+ void moveSelection(int direction)	Move character selection arrow according to A or D keys (called in input polling loop).
+ String getSelection()	Return selections array index of currentSelection

(C) GameScene extends AbstractScene

Fields and Constructors

+ GameScene(Pane parent, double width, double height)	Use extended constructor set all given value to it.
---	---

Methods

+ void redraw()	Clear frame, draw room background, draw enemies' and player's shadows, draw obstacles, draw enemies, draw dropped items, draw player's character, draw projectiles, and draw on-screen text.
- void drawText(GraphicsContext gc)	Draw on-screen texts: character name, "Now playing as", and score.
+ Canvas getCanvas()	Return canvas

(C) EndGameScene extends AbstractScene

Fields and Constructors

+ EndGameScene(Pane parent, double width, double height)	Use extended constructor set all given value to it.
--	---

Methods

+ void redraw()	Clear frame and draw original "YOU DIED" screen from Elden Ring lol.
-----------------	--

18. Application

(C) Main extends Application

Field

- final String GAME_TITLE	Set default as "Void seekers".
---------------------------	--------------------------------

Constructors

+ void main(String[] args)	Run JavaFX application.
+ void start(Stage PrimaryStage)	Set primary window, Health bar, Inventory Bar, Active Item Bar, Trinket Bar, Passing GUI to game logic, Show gui, set key press and release event, Start polling and rendering thread, and Clear everything on close.