Hollow Knight Documentation

Created by
Terash Inkong 6130268621
Nasri Islam 6130287521

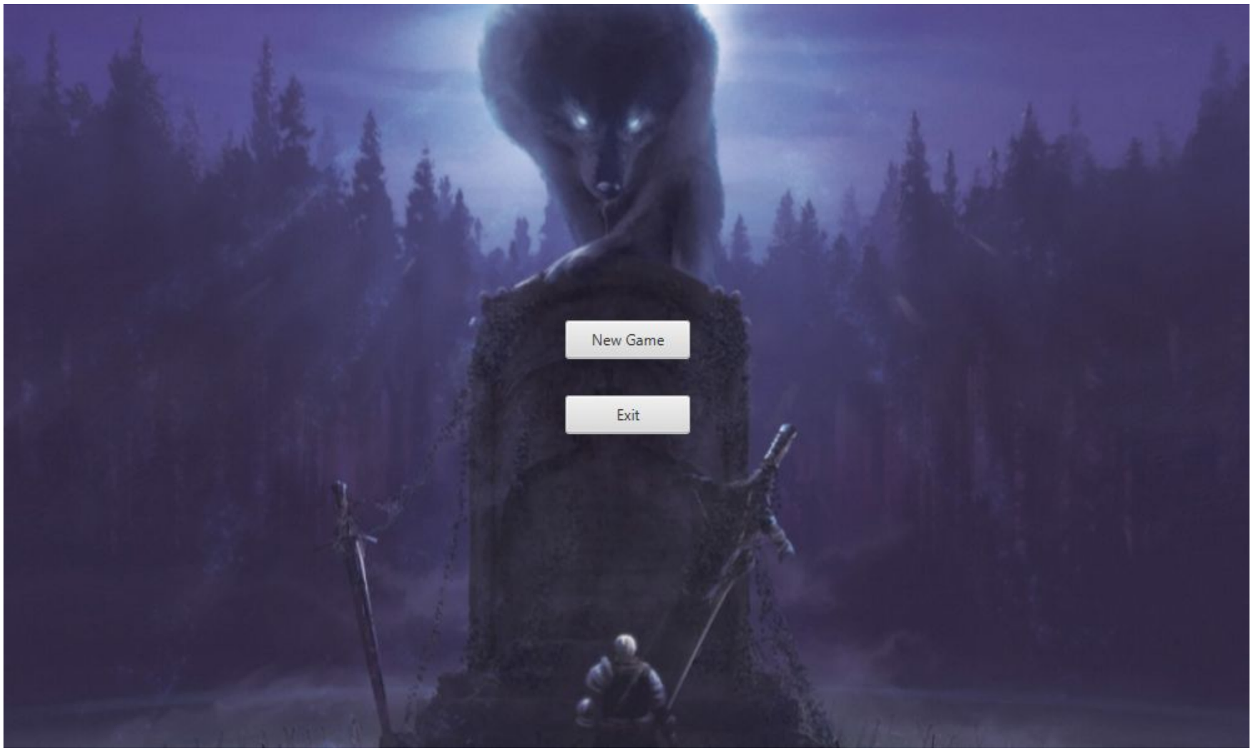
2110215 Programming Methodology
Semester 1 Year 2019
Chulalongkorn University

Hollow Knight Documentation

Introduction

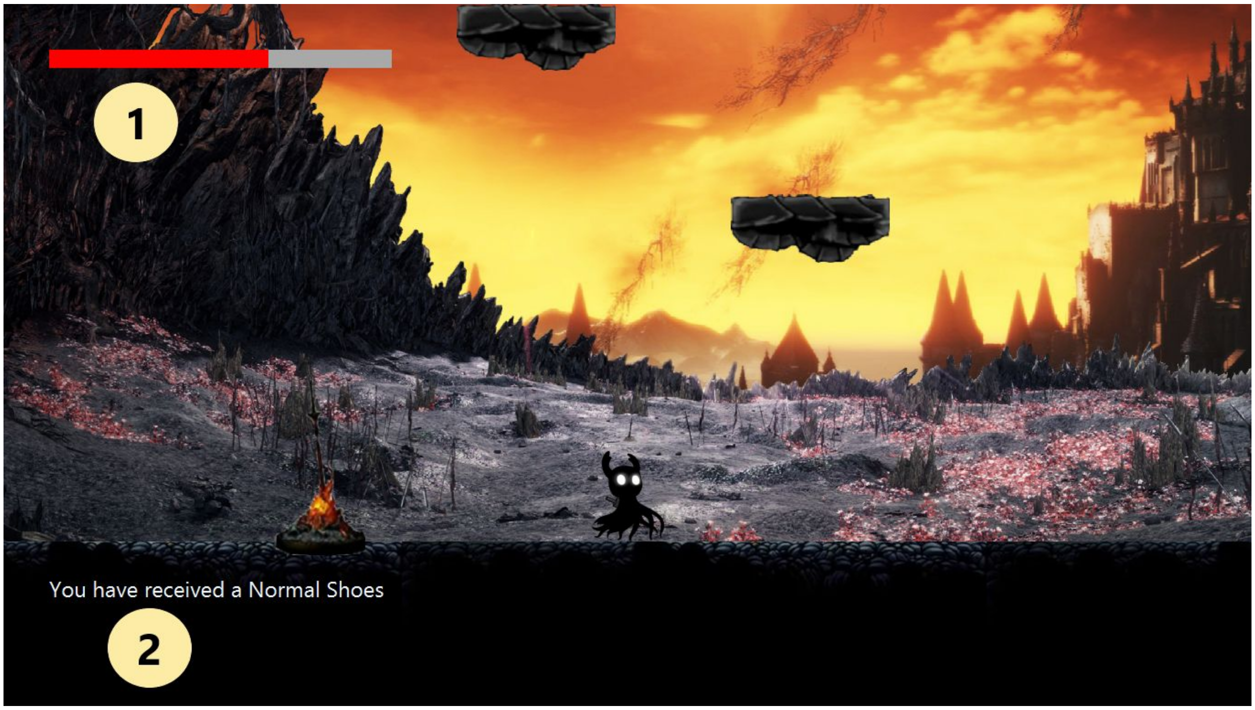
Hollow Knight is a 2D action-adventure where a courageous hero explores the world and battle enemies. Dodge, dash and slash your way through even the most deadly enemy. Equip Item! Ancient weapon that offer power. Choose your favourites and make your journey unique. The objective of this game is to survive and kill every enemy. Players engage in intense acrobatic combat as well as challenging platforming to reach the ending of the game.

Main menu scene



- Press “New Game” to start new game
- Press “Exit” to exit from game

Gameplay scene



Once you press “New Game” Button on Main Menu Scene. After the game is finished loading, It will bring you to gameplay scene.

1) **Hero Hp Bar** : This part shows the remaining health of your hero. Your hero start the game with 100 max Hp. Health is lost by taking damage from enemies. Health can be regained by resting at a checkpoint. If the health are depleted to 0, the hero will be back to the last checkpoint.
2) **Event Log** : This part shows the event occurred, such as receiving an Item or defeating a boss.

Controls

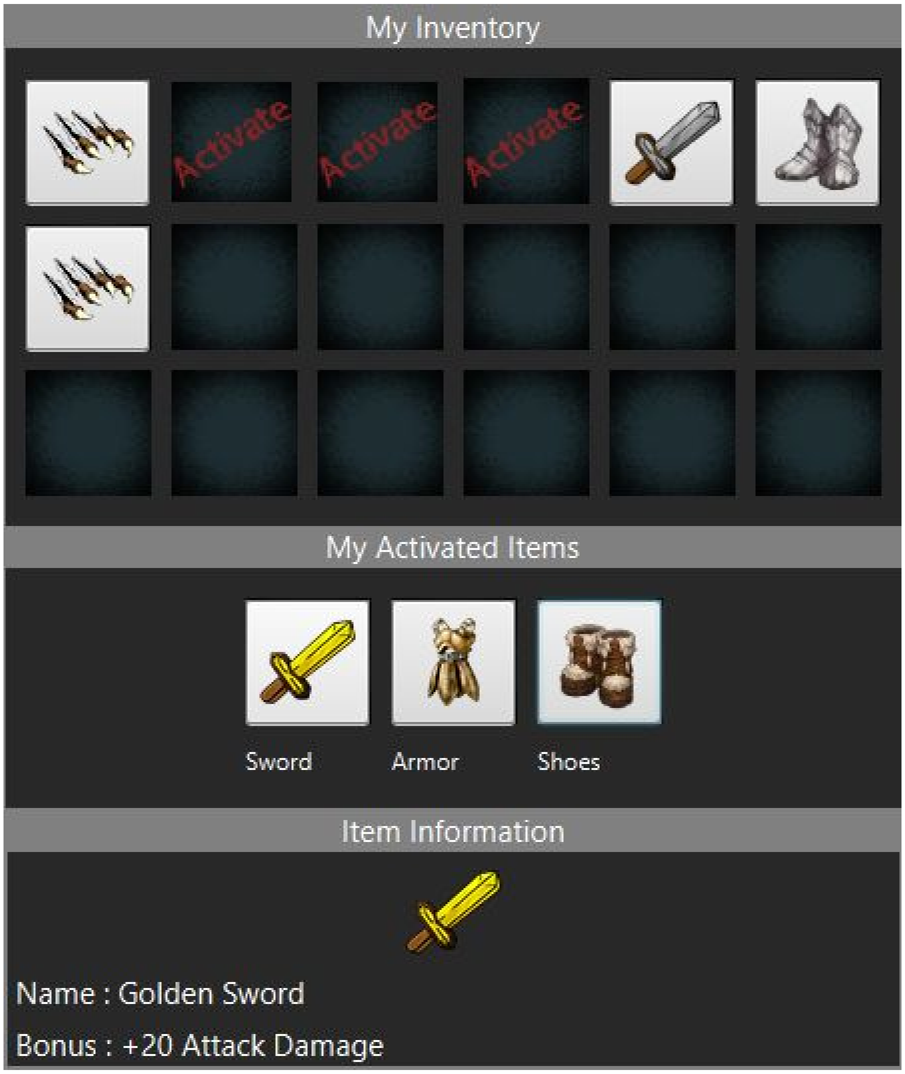
- Left - LEFT Arrow Key
- Right - RIGHT Arrow Key
- Jump - A or SPACE (hold to jump higher)
  - Double Jump - A or SPACE (while in the air)
- Attack - S
  - Upper Slash - UP + S
  - Downward Slash - DOWN + S (while in the air)
- Dash - D
- Inventory - I
- Exit Full Screen - ESC

GamePlay

The hero can attack in four directions, and performing a down attack while in the air allows the hero to pogo on top of the enemy, allowing you to bounce off the enemy and repeat the attack. Press D to dash a short distance in the direction the hero is facing. Can be used in mid-air. Refreshes when the hero lands on the ground. Allows the hero to cross gaps too far to reach by jumping, and makes it easier to dodge enemies and their attacks.

Item is also an important part of Hollow Knight, as its bonus will determine how effective you are in combat. There 3 types of Item, Sword, Armor and Shoes. Sword increase the Attack, Armor increase Health and Shoes for the Speed.

Inventory



Name : Golden Sword

Bonus : +20 Attack Damage

Inventory consists of :

My Inventory) This part contains every Item you have. Left click on the Item to equip. Right click to remove an Item from Inventory.

My Activated Items) This part contains the Item you are equipped. You can equip only one Item for each type. Left click on the Item to unequip.

Item Information) This part shows the information of the Item your mouse are pointing.

Monster

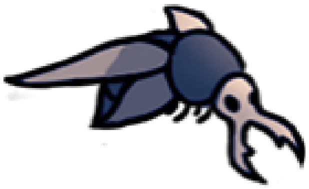
Monster are hostile enemies in Hollow Knight that the hero can confront to obtain Items. Bosses are unique enemies in Hollow Knight which constitute some of the most challenging experiences in the game. Boss do not respawn once killed. When the hero encounters a Boss, their name will appear on-screen, and the music will change into a battle theme. In total, There are 4 different enemies in the whole game, with 2 of them being Boss.

1) Glimback

Glimback is an enemy in Hollow Knight. Large and well-protected, but slow and docile. Whenever Glimback hit the wall, it turns to the opposite direction.

Health	100
Attack	20
Drops	Rare Sword 30% Rare Armor 30% Rare Shoes 30% Legend Armor 5%

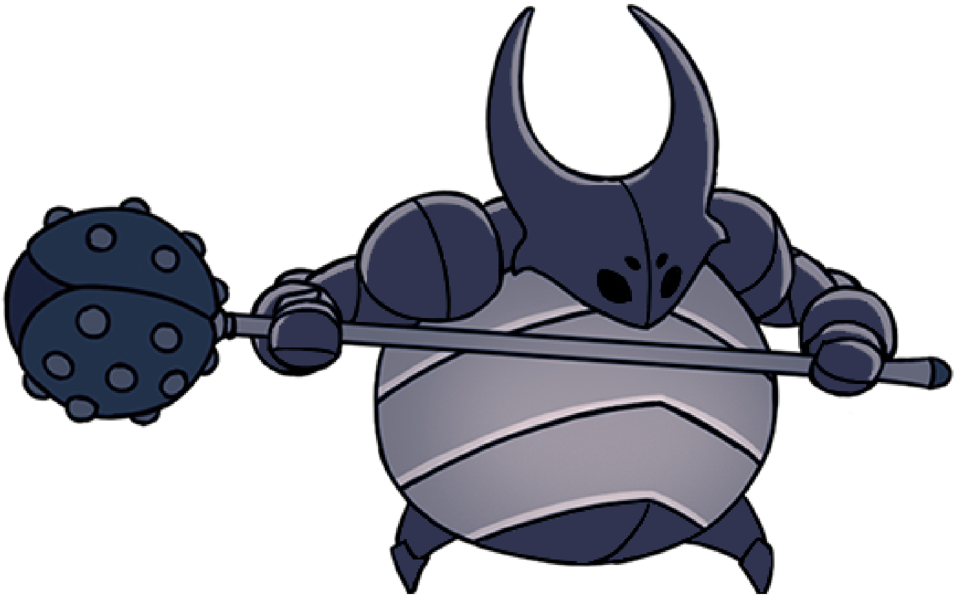
2) Vengefly



Vengefly is an enemy in Hollow Knight. It fly in the air and pursues its prey relentlessly when you get too close.

Health	40
Attack	20
Drops	Normal Sword 20% Normal Armor 20% Normal Shoes 20%

3) False Knight



False Knight is a Boss in Hollow Knight. He is the first Boss that hero will encounter. The False Knight falls from the ceiling when the hero enters the room. False Knight possesses two attacks :

- **Shockwave Slam :** False Knight will rear back priming the mace for a moment before swing it forward, slamming it into the ground. The impact of the mace creates a shockwave that travels forward across the whole arena.

- **Leaping Attack :** False Knight will leap into the air and slam his mace down in front of him when he lands. False Knight targets the hero when he leaps so he will strike where the hero was when he started his leap. While False Knight is in the air, he lifts his mace above his head and then swings it in an overhead arc before he lands.

Health	600
Attack	25
Drops	Golden Sword 100%

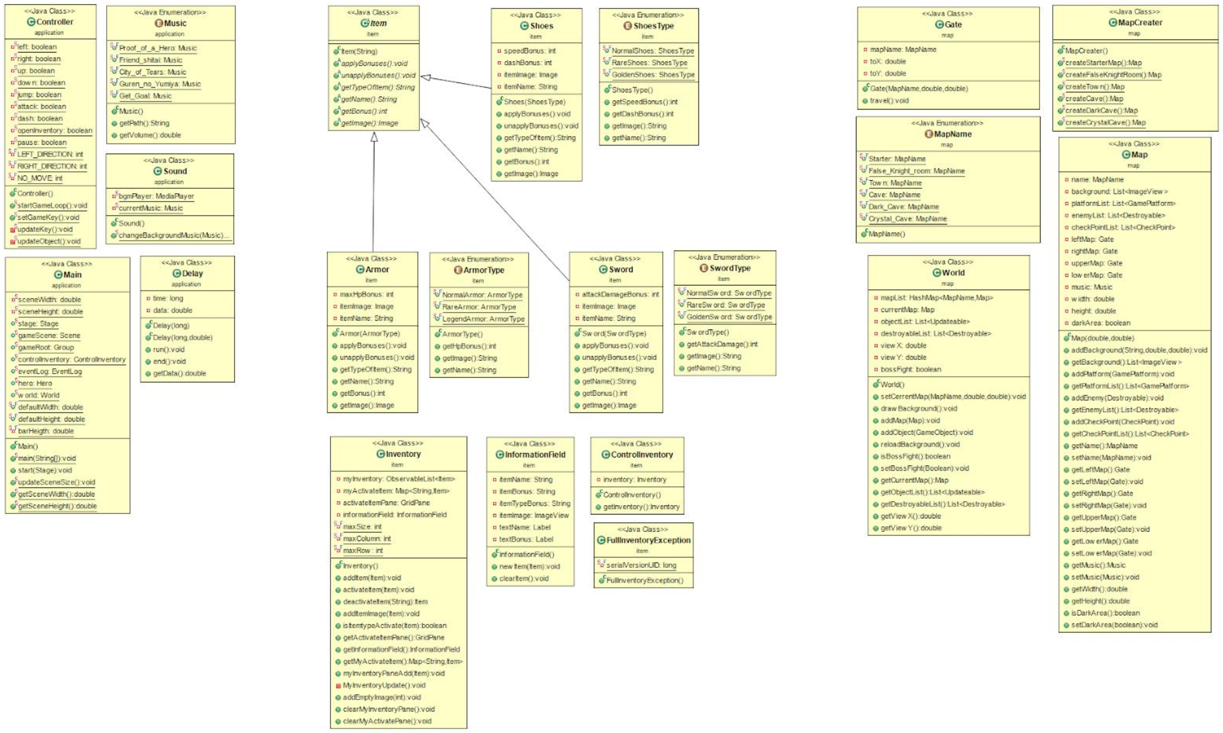
4) Illyasviel von Einzbern

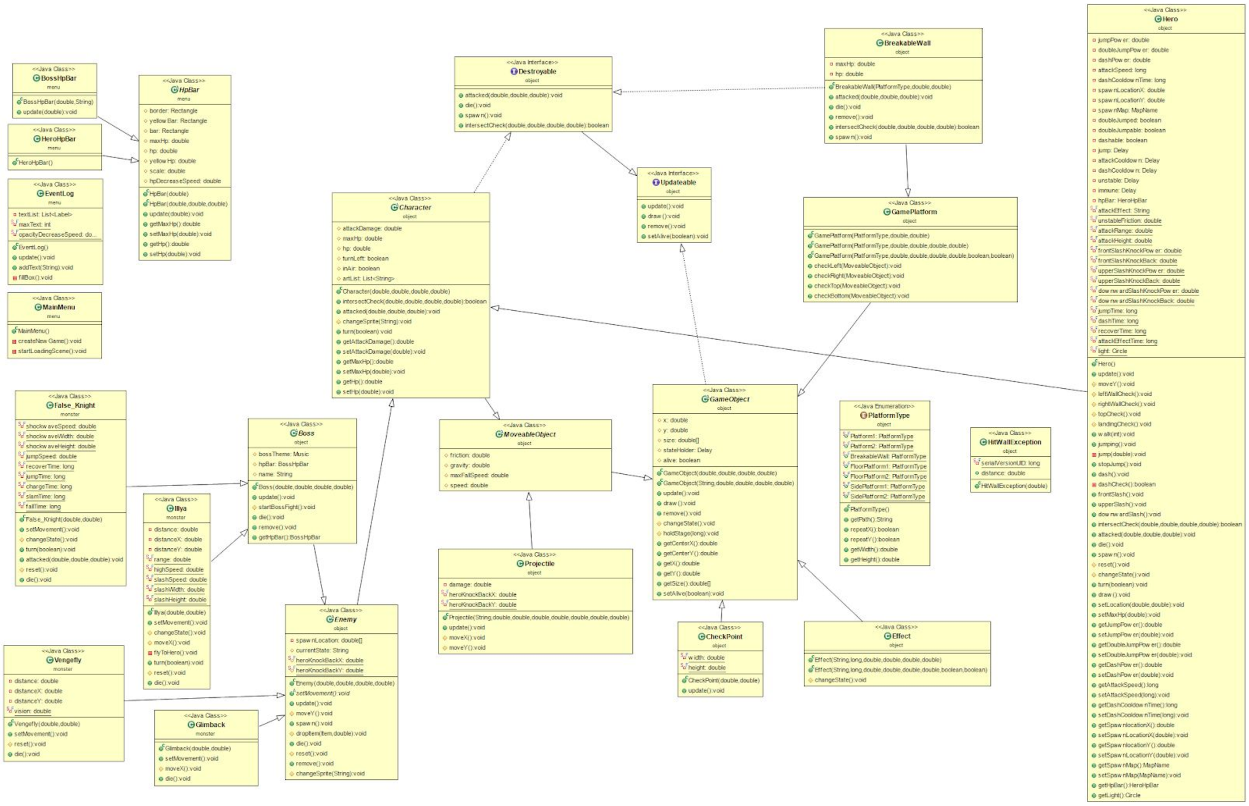
Illya is a Boss in Hollow Knight. She can be found behind the breakable wall in the cave area. When the hero enters the room, Illya teleport to the middle of the arena. Illya fly up in the air and keep distance from the hero.

Illya possesses only one attack with Illya will fire beams of white-hot energy across the arena to your hero. Whenever Illya uses this attack, she flies down and leaves an open opportunity to attack.

Health	500
Attack	20
Drops	Golden Shoes 100%

Implementation Details





1. Package application

1.1 Class Main

1.1.1 Field

private static double sceneWidth	current width of the scene
private static double sceneHeight	current height of the scene
public static Stage stage	primary stage
public static Scene gameScene	game’s scene
public static Group gameRoot	game’s root
public static ControlInventory controlInventory	your inventory
public static EventLog eventLog	event log in the game
public static Hero hero	your hero
public static World world	contain every object in the game

1.1.2 Method

public static void main(String[] args)	main application
public void start(Stage primaryStage)	set stage component and open main menu scene
public static void updateSceneSize()	update current scene’s width and height
Generate getter	

1.2 Class Controller

1.2.1 Field

private static boolean left	press LEFT to move left
private static boolean right	press RIGHT to move right
private static boolean up	press UP to look up
private static boolean down	press DOWN to look down
private static boolean jump	press A or SPACE to jump
private static boolean attack	press S to attack
private static boolean dash	press D to dash

private static boolean openInventory	press I to open inventory
private static boolean pause	pause the game

1.2.2 Method

public static void startGameLoop()	initialize an AnimationTimer that update everything
public static void setGameKey()	set the keyboard input
private static void updateKey()	command your hero
private static void updateObject()	update every object in the world

1.3 Class Sound

1.3.1 Field

private static MediaPlayer bgmPlayer	the media player that play background music
private static Music currentMusic	current background music

1.3.2 Method

public static void changeBackgroundMusic (Music music)	change the background music

1.4 Class Delay extends Thread

1.4.1 Field

private long time	time to sleep
private double data	collects a number (if necessary)

1.4.2 Constructor

public Delay(long time)	initialize a Delay with the following time
public Delay(long time, double data)	initialize a Delay with the following time and data

1.4.3 Method

public void run()	sleep for the time
public void end()	do this function after sleep (if override)
public double getData()	return data

2. Package menu

2.1 Class MainMenu extends VBox

2.1.1 Constructor

public MainMenu()	initializes the MainMenu with a new game button and an exit button

2.1.2 Method

private void createNewGame()	go to the loading scene and change to the game scene when finish loading
private void startLoadingScene()	create the loading scene

2.2 Class EventLog extends VBox

2.2.1 Field

private List<Label> textList	collects all alive text

2.2.2 Constructor

public EventLog()	initializes the EventLog

2.2.3 Method

public void update()	decrease every alive text opacity
public void addText(String text)	add a text to the EventLog
private void fillBox()	fill the EventLog with the empty label

2.3 Abstract Class HpBar extends StackPane

2.3.1 Field

protected Rectangle border	a grey rectangle that show max health point
protected Rectangle yellowBar	a yellow rectangle that show yellow hp
protected Rectangle bar	a red rectangle that show remaining health point
protected double maxHp	your hero’s max health point
protected double hp	your hero’s remaining health point
protected double yellowHp	health point that decrease over time when you take damage

protected double scale	scale between your hp and bar’s width
protected double hpDecreaseSpeed	yellow hp’s decrease speed

2.3.2 Constructor

public HpBar(double size)	initializes the HpBar with that height
public HpBar(double size, double maxHp, double scale)	initializes the HpBar with that height and set maxHp and scale

2.3.3 Method

public void update(double hp)	update the remaining health point
generate getter and setter	

2.4 Class HeroHpBar extends HpBar

2.4.1 Constructor

public HeroHpBar()	initializes the HeroHpBar

2.5 Class BossHpBar extends HpBar

2.5.1 Constructor

public BossHpBar(double maxHp, String name)	initializes the BossHpBar

2.5.2 Method

public void update(double hp)	update the remaining hp and fix the bar’s size when the scene’s size change

3. Package object

3.1 Interface Updateable

3.1.1 Method

public void update()	update this object
public void draw()	draw this object
public void remove()	remove this object from the game
public void setAlive(boolean alive)	set if this object is in the game or not

3.2 Interface Destroyable

3.2.1 Method

public void attacked(double damage, double knockbackX, double knockbackY)	attack this object
public void die()	object die when it was killed
public void spawn()	reset and add this object to the game
public boolean intersectCheck (double x, double y, double width, double height)	check if this object is intersected with the given rectangle

3.3 Abstract Class GameObject extends Group implements Updateable

3.3.1 Field

protected double x	current location on X-axis in the Map
protected double y	current location on Y-axis in the Map
protected double[] size	size of this object (index 0 is width, index 1 is height)
protected Delay stateHolder	a thread that change this object’s state when die
protected boolean alive	this object is in the game or not

3.3.2 Constructor

public GameObject(double x, double y, double width, double height)	initializes the GameObject’s location and size
public GameObject(String imagePath, double x, double y, double width, double height)	initializes the GameObject’s location and size and add an image

3.3.3 Method

public void update()	redraw this object
public void draw()	relocate this object to the proper position
public void remove()	remove this object from the game
protected void changeState()	change this object’s state (empty if this object has no state)
protected void holdStage(long time)	change this object’s state after a time
public double getCenterX()	return the center of this object on X-axis
public double getCenterY()	return the center of this object on Y-axis
generate all getter and setter	

3.4 Class GamePlatform extends GameObject

3.4.1 Constructor

public GamePlatform(PlatformType platformType, double x, double y)	initializes the GamePlatform
public GamePlatform(PlatformType platformType, double x, double y, double width, double height)	initializes the GamePlatform and expand it
public GamePlatform(PlatformType platformType, double x, double y, double width, double height, boolean flipX, boolean flipY)	initializes the GamePlatform, expand it and flip it

3.4.2 Method

public void checkLeft(MoveableObject object) throws HitWallException	throws a HitWallException with the distance between object and GamePlatform if the object is going to hit the left wall of the GamePlatform
public void checkRight(MoveableObject object) throws HitWallException	throws a HitWallException with the distance between object and GamePlatform if the object is going to hit the right wall of the GamePlatform
public void checkTop(MoveableObject object) throws HitWallException	throws a HitWallException with the distance between object and GamePlatform if the object is going to hit the top of the GamePlatform
public void checkBottom (MoveableObject object) throws HitWallException	throws a HitWallException with the distance between object and GamePlatform if the object is going to hit the bottom of the GamePlatform

3.5 Class BreakableWall extends GamePlatform implements Destroyable

3.5.2 Field

protected double maxHp	max health point
protected double hp	remaining health point

3.5.2 Constructor

BreakableWall(PlatformType platformType, double x, double y)	initializes the BreakableWall

3.5.3 Method

public void attacked(double damage, double knockBackX, double knockBackY)	decrease this object's remaining health point
public void die()	permanently remove this object
public void remove()	remove this object from the game and also remove it from destroyableList
public boolean intersectCheck(double x, double y, double width, double height)	return true if this object is intersected with the given rectangle

3.6 Class Effect extends GameObject

3.6.1 Constructor

public Effect(String imagePath, long time, double x, double y, double width, double height)	initializes the Effect that vanish after a time
public Effect(String imagePath, long time, double x, double y, double width, double height, boolean flipX, boolean flipY)	initializes the Effect that vanish after a time and flip it

3.6.2 Method

protected void changeState()	remove this object from the game

3.7 Class CheckPoint extends GameObject

3.7.1 Constructor

public CheckPoint(double x, double y)	initializes the CheckPoint

3.7.2 Method

public void update()	if the CheckPoint is intersected with hero, heal your hero and make the hero spawn at this CheckPoint when die

3.8 Abstract Class MoveableObject extends GameObject

3.8.1 Field

protected double friction	delay this object movement speed
protected double gravity	pull this object down
protected double maxFallSpeed	you cannot fall faster than this speed
protected double speed	this object’s quickness
protected double dx	current velocity in X-axis
protected double dy	current velocity in Y-axis

3.8.2 Constructor

public MoveableObject(double x, double y, double width, double height)	initializes the MoveableObject
public MoveableObject(String imagePath, double x, double y, double width, double height)	initializes the MoveableObject and add an image

3.8.3 Method

public void update()	move this object
protected void moveX()	move this object on X-axis
protected void moveY()	move this object on Y-axis
protected void leftWallCheck() throws HitWallException	throws a HitWallException with the distance between object and GamePlatform if this object is going to hit the right side of any GamePlatform or hit the map’s edge
protected void rightWallCheck() throws HitWallException	throws a HitWallException with the distance between object and GamePlatform if this object is going to hit the left side of any GamePlatform or hit the map’s edge
protected void topCheck() throws HitWallException	throws a HitWallException with the distance between object and GamePlatform if this object is going to hit the bottom of any GamePlatform or hit the map’s edge

protected void landingCheck() throws HitWallException	throws a HitWallException with the distance between object and GamePlatform if this object is going to hit the top of any GamePlatform or hit the map’s edge
protected void reset()	reset the velocity
generate all getter and setter	

3.9 Abstract Class Character extend MoveableObject implements Destroyable

3.9.1 Field

protected double attackDamage	attack damage
protected double maxHp	max health point
protected double hp	remaining health point
protected boolean turnLeft	turn the sprite left or not
protected boolean inAir	this object is in the air or not
protected List<String> artList	collects all sprite name

3.9.2 Constructor

public Character(double x, double y, double width, double height)	initializes the Character

3.9.3 Method

public boolean intersectCheck(double x, double y, double width, double height)	return true if this object is intersected with the given rectangle
public void attacked(double damage, double knockBackX, double knockBackY)	decrease this object’s remaining health point and knock this object
protected void changeSprite(String art)	change the sprite
public void turn(boolean turnLeft)	turn the sprite
generate all getter and setter	

3.10 Class Hero extends Character

3.10.1 Field

private double jumpPower	speed when jump
private double doubleJumpPower	speed when double jump
private double dashPower	speed when dash
private long attackSpeed	time between each attack
private long dashCooldownTime	time between each dash
private double spawnLocationX	the location on X-axis to spawn when you die
private double spawnLocationY	the location on Y-axis to spawn when you die
private MapName spawnMap	the Map to spawn when you die
private boolean doubleJumped	you have double jumped or not
private boolean doubleJumpable	you can double jump or not
private boolean dashable	you can dash or not
private Delay jump	the thread when you are holding jump button
private Delay attackCooldown	the thread when you cannot attack
private Delay dashCooldown	the thread when you cannot dash
private Delay unstable	the thread when you stagger
private Delay immune	the thread when you are unable to be attacked
private HeroHpBar hpBar	your hp bar on the top-left of the scene

3.10.2 Constructor

public Hero()	initializes the Hero

3.10.3 Method

public void update()	update the HeroHpBar and move
protected void moveY()	move the hero on Y-axis and also check if you are landing
protected void leftWallCheck() throws HitWallException	also check the left Gate when you hit the map's edge

protected void rightWallCheck() throws HitWallException	also check the right Gate when you hit the map’s edge
protected void topCheck() throws HitWallException	also check the top Gate when you hit the map’s edge
protected void landingCheck() throws HitWallException	also check the bottom Gate when you hit the map’s edge
public void walk(int direction)	set your velocity to that direction
public void jumping()	jump or double jump if you can
public void jump(double power)	create a jump thread
public void stopJump()	stop jumping and Interrupt the jump thread
public void dash()	dash and change your sprite if you can
private boolean dashCheck()	set the velocity if you are in the dashing state
public void frontSlash()	attack every destroyable object in front of you
public void upperSlash()	attack every destroyable object above you
public void downwardSlash()	attack every destroyable object under you
public boolean intersectCheck(double x, double y, double width, double height)	return true if your is intersected with the given rectangle and not immune
public void attacked(double damage, double knockbackX, double knockbackY)	Decrease your hero’s remaining health point and knock this object and also make it stagger and immune
public void die()	respawn your hero
public void spawn()	move to the spawn location
protected void reset()	reset all cooldowns
protected void changeState()	change to the normal state
public void turn(boolean turnLeft)	turn and move the dash sprite to the proper location
public void draw()	Move the hero and background
public void setLocation(double x, double y)	move to that location in the Map
generate all getter and setter	

3.11 Class Projectile extends MoveableObject

3.11.1 Field

private double damage	attack damage when hit the hero

3.11.2 Constructor

public Projectile(String imagePath, double x, double y, double width, double height, double dx, double dy, double damage)	initializes the Projectile and add the image

3.11.3 Method

public void update()	move and attack the hero when hit the hero
protected void moveX()	Projectile is broken when hit the wall
protected void moveY()	Projectile is broken when hit the wall

3.12 Abstract Class Enemy extends Character

3.12.1 Field

private double[] spawnLocation	the location when spawn (index 0 is X-axis, index 1 is Y-axis)
protected String currentState	The current state name

3.12.2 Constructor

public Enemy(double x, double y, double width, double height)	initializes the Enemy

3.12.3 Method

public abstract void setMovement()	set the movement pattern
public void update()	move and attack the hero when hit the hero
protected void moveY()	move this object on Y-axis and also check if it is landing
public void spawn()	reset and add this object to the game
protected void dropItem(Item item, double dropRate)	add the item to your inventory for the percent of the drop rate
public void die()	enemy dies when they was killed

protected void reset()	reset the sprite and velocity
public void remove()	remove this object from the game (also remove it from destroyableList)
protected void changeSprite(String art)	change the sprite and also set the currentState

3.13 Abstract Class Boss extends Enemy

3.13.1 Field

protected Music bossTheme	the background music when fighting the Boss
protected BossHpBar hpBar	Boss's hp bar on the bottom of the scene
protected String name	name of the Boss

3.13.2 Constructor

public Boss(double x, double y, double width, double height)	initializes the Boss

3.13.3 Method

public void update()	move and update the hp bar
protected void startBossFight()	play the boss theme and start boss fight
public void die()	permanently remove the Boss
public void remove()	Remove the Boss and boss's hp bar
public BossHpBar getHpBar()	return boss's hp bar

4. Package monster

4.1 Class Glimback extends MoveableEnemy

4.1.1 Constructor

public Glimback(double x, double y)	initializes the Glimback

4.1.2 Method

public void setMovement()	always move forward
protected void moveX()	turn to the opposite direction when hit the wall
public void die()	can drop a RareSword, RareArmor and RareShoes at the rate 30% and drop a LegendArmor at the rate 5% when die

4.2 Class Vengefly extends MoveableEnemy

4.2.1 Field

private double distance	distance between this and hero
private double distance	distance between this and hero on X-axis
private double distance	distance between this and hero on Y-axis

4.2.2 Constructor

public Vengefly(double x, double y)	initializes the Vengefly

4.2.3 Method

public void setMovement()	do nothing in the idle state and fly straight to the hero when detect
protected void reset()	change the state to idle
public void die()	can drop a NormalSword, NormalArmor and NormalShoes at the rate 20% when die

4.3 Class False_Knight extends Boss

4.3.1 Constructor

public False_Knight(double x, double y)	initializes the False Knight

4.3.2 Method

public void setMovement()	only move forward on jump and leap state
protected void changeState()	go to the next stage and set the delay, attack the hero on the end of charge and leap state
public void turn(boolean turnLeft)	turn and move the all sprites to the proper location
public void attacked(double damage, double knockbackX, double knockbackY)	False_Knight is harder to stagger
protected void reset()	change the state to idle
public void die()	drop a GoldenSword when die

4.4 Class Illya extends Boss

4.4.1 Field

private double distance	distance between this and hero
private double distance	distance between this and hero on X-axis
private double distance	distance between this and hero on Y-axis

4.4.2 Constructor

public Illya(double x, double y)	initializes the Illya

4.4.3 Method

public void setMovement()	fly to the hero if too far, far away if too close and when attacking, fly to the same plane with the hero
protected void changeState()	go to the next stage and set the delay, create a Projectile on the end of attacking state
protected void reset()	change the state to idle and make it invisible
protected void moveX()	move to the opposite side of the hero when hit the wall
private void flyToHero()	fly straight to the hero
public void turn(boolean turnLeft)	turn and move the all sprites to the proper location
public void die()	can drop a NormalSword, NormalArmor and NormalShoes at the rate 20% when die

5. Package map

5.1 Class Map

5.1.1 Field

private MapName name	name of the map
private List<ImageView> background	collects all backgrounds in this map
private List<GamePlatform> platformList	collects all platforms in this map
private List<Destroyable> enemyList	collects all enemies in this map
private List<CheckPoint> checkPointList	collects all checkpoints in this map
private Gate leftMap	gate on the left side of this map
private Gate rightMap	gate on the right side of this map
private Gate upperMap	gate above this map
private Gate lowerMap	gate under this map
private Music music	background music of this map
private double width	width of this map
private double height	height of this map
private boolean darkArea	this map is dark or not

5.1.2 Constructor

public Map(double width, double height)	initializes the Map with width and height

5.1.3 Method

public void addBackground (String backgroundImagePath, double width, double height)	add a background
public void addPlatform(GamePlatform platform)	add a platform to the platformList
public void addEnemy(Destroyable enemy)	add an enemy to the enemyList
public void addCheckPoint (CheckPoint checkPoint)	add a checkpoint to the checkPointList
generate all getter and setter	

5.2 Class Gate

5.2.1 Field

private MapName mapName	the destination Map
private double toX	the location on X-axis in the destination Map
private double toY	the location on Y-axis in the destination Map

5.2.2 Constructor

public Gate(MapName mapName, double toX, double toY)	initializes the Gate’s destination

5.2.3 Method

public void travel()	travel to the Gate’s destination

5.3 Class MapCreater

5.3.1 Method

public static Map createStarterMap()	create a starter map
public static Map createFalseKnightRoom()	create a False Knight’s boss room
public static Map createTown()	create a town
public static Map createCave()	create a cave (also make it dark)
public static Map createDarkCave()	create a dark cave (also make it dark)
public static Map createCrystalCave()	create a crystal cave

5.4 Class World extends Group

5.4.1 Field

private HashMap<MapName, Map> mapList	collects all map’s name as keys and Map as values
private Map currentMap	the map where your hero is now
private List<Updateable> objectList = new ArrayList<Updateable>()	collects all objects
private List<Destroyable> destroyableList	collects all destroyable objects
private double viewX	the location on X-axis the scene should be on the Map

private double viewY	the location on Y-axis the scene should be on the Map
private boolean bossFight	you are fighting a boss or not

5.4.2 Constructor

public World()	initializes the World (using MapCreater to create Map)

5.4.3 Method

public void setCerrentMap(MapName name, double x, double y)	set everything in the game at that location
public void drawBackground()	relocate all backgrounds to the proper position
public void addMap(Map map)	add a Map to the mapList
public void addObject(GameObject object)	add an object to the game
public void reloadBackground()	refresh the background
generate all getter and setter	

6. Package Item

6.1 Abstract class Item extends Button

6.1.1 Constructor

public Item(String imagePath)	initializes an Item.

6.1.2 Abstract Method

public abstract void applyBonuses()	Increase the hero power when using an item.
public abstract void unApplyBonuses()	Decrease the hero power when stop using an item.
public abstract String getTypeOfItem()	Return type of item.
public abstract String getName()	Return item’s name.
public abstract int getBonus()	Return item’s bonus.
public abstract Image getImage()	Return item’s image.

6.2 Class Sword extends Item

6.2.1 Field

private int attackDamageBonus	Sword’s attack damage.
private Image itemImage	Sword’s image.
private String itemName	Sword’s name.

6.2.2 Constructor

public Sword(SwordType sword)	initializes attackDamageBonus, itemImage and itemName of sword.

6.2.3 Method

public void applyBonuses()	Increase the hero’s attack when using a sword.
public void unapplyBonuses()	Decrease the hero’s attack when stop using a sword.
public String getTypeOfItem()	Return string of type of item (“Sword”).
public String getName()	Return string of item’s name.
public int getBonus()	Return sword’s attackDamageBonus.
public Image getImage()	Return image of this sword.

6.3 enum SwordType

6.3.1 Field

NormalSword	Normal sword.
RareSword	Rare sword.
GoldenSword	Golden sword.

6.3.2 Method

public int getAttackDamage()	Return bonus attack damage of each sword.
public String getImage()	Return Image of each sword.
public String getName()	Return name of each sword.

6.4 Class Armor extends Item

6.4.1 Field

private int maxHpBonus	Increase hero’s max HP.
private Image itemImage	Sword’s image.
private String itemName	Sword’s name.

6.4.2 Constructor

public Sword(SwordType sword)	initializes attackDamageBonus, itemImage and itemName of sword.

6.4.3 Method

public void applyBonuses()	Increase the hero’s max HP when using an armor.
public void unapplyBonuses()	Decrease the hero’s max HP when stop using an armor.
public String getTypeOfItem()	Return string of type of item (“Armor”).
public String getName()	Return string of item’s name.
public int getBonus()	Return armor’s maxHpBonus.
public Image getImage()	Return image of this armor.

6.5 enum ArmorType

6.5.1 Field

NormalArmor	Normal armor.
RareArmor	Rare armor.
LegendArmor	Legend armor.

6.5.2 Method

public int getHpBonus()	Return bonus Hp of each armor.
public String getImage()	Return Image of each armor.
public String getName()	Return name of each armor.

6.6 Class Shoes extends Item

6.6.1 Field

private int speedBonus	Shoes’s speed bonus.
private int dashBonus	Shoes’s dashing speed bonus.
private Image itemImage	Shoes’s image.
private String itemName	Shoes’s name.

6.6.2 Constructor

public Shoes(ShoesType shoes)	initializes speedBonus, dashBonus, itemImage and itemName of Shoes.

6.6.3 Method

public void applyBonuses()	Increase the hero’s speed and dashing speed when wearing shoes.
public void unapplyBonuses()	Decrease the hero’s speed and dashing speed when removing shoes.
public String getTypeOfItem()	Return string of type of item (“Shoes”).
public String getName()	Return string of item’s name.
public int getBonus()	Return shoes’s speedBonus.
public Image getImage()	Return shoes’s image..

6.7 Class Inventory extends GridPane

6.7.1 Field

private ObservableList<Item> myInventory.	collects all hero’s items.
private Map<String, Item> myActivateItem	storing active items. (Key is type of item, value is an item).
private GridPane activateItemPane	A slot for storing active items.
private InformationField informationField	Variable for storing item’s Information.
private static final int maxSize = 18	Maximum size of inventory’s GridPane.
private static final int maxColumn = 6	Maximum column of inventory’s GridPane.
private static final int maxRow = 3	Maximum row of inventory’s GridPane.

6.7.2 Constructor

public Inventory()	initializes Inventory.

6.7.3 Method

public void addItem(Item newItem)	Add an item to myInventory.
public void activateItem(Item newItem)	Activate an item.
public Item deactivateItem(String typeOfItem)	Item disabled.
public void addItemImage(Item newItem)	Add item's image(image of newItem) to GridPane My Activate Item.
public boolean isItemTypeActivate(Item item)	Check if this item type has been activated.
public GridPane getActivateItemPane()	Getter of activateItemPane
public InformationField getInformationField()	Getter of informationField (item's information).
public Map<String, Item> getMyActivateItem()	Getter of myActivateItem.
public void myInventoryPaneAdd(Item newItem)	Add item's image to GridPane Inventory.
public void myInventoryUpdate()	Update GridPane My inventory.
public void addEmptyImage(int index)	Add empty image to GridPane Inventory.
public void clearMyInventoryPane()	Clear GridPane Inventory.
public void clearMyActivatePane()	Clear GridPane My activate Item.

6.8 Class InformationField extends VBox

6.8.1 Field

private String itemName	Item's name.
private String itemBonus	Item's special bonus.
private String itemTypeBonus	Type of special bonus.
private Image itemImage	Item's image
private Label textName	Item's name (Label)
private Label textBonus	Item's bonus (Label)

6.8.2 Constructor

public InformationField()	Initializes InformationField.

6.8.3 Method

public void newItem(Item item)	Add item information to the label.
public void clearItem()	Clear label.

6.8 Class ControlInventory extends VBox

6.8.1 Field

private Inventory inventory	Inventory.

6.8.2 Constructor

public ControlInventory()	Initializes ControlInventory.

6.8.3 Method

public Inventory getInventory	Getter of Inventory.