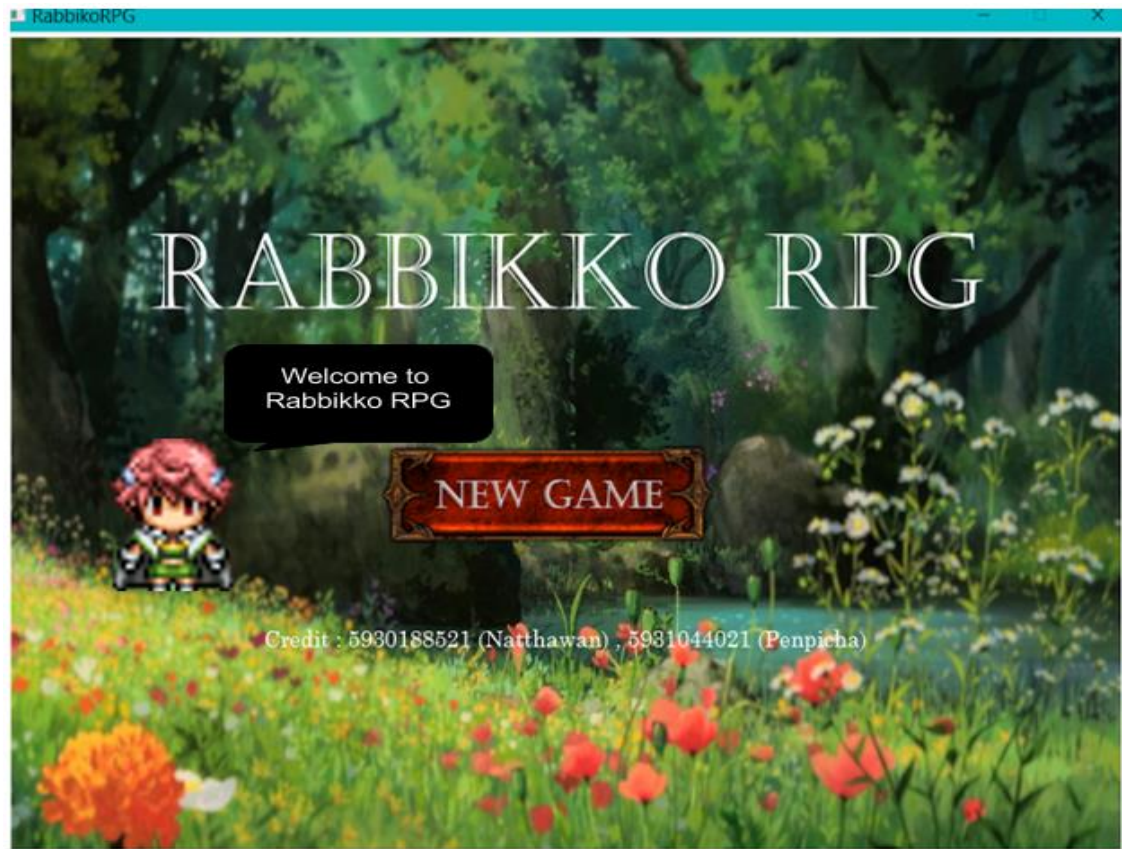


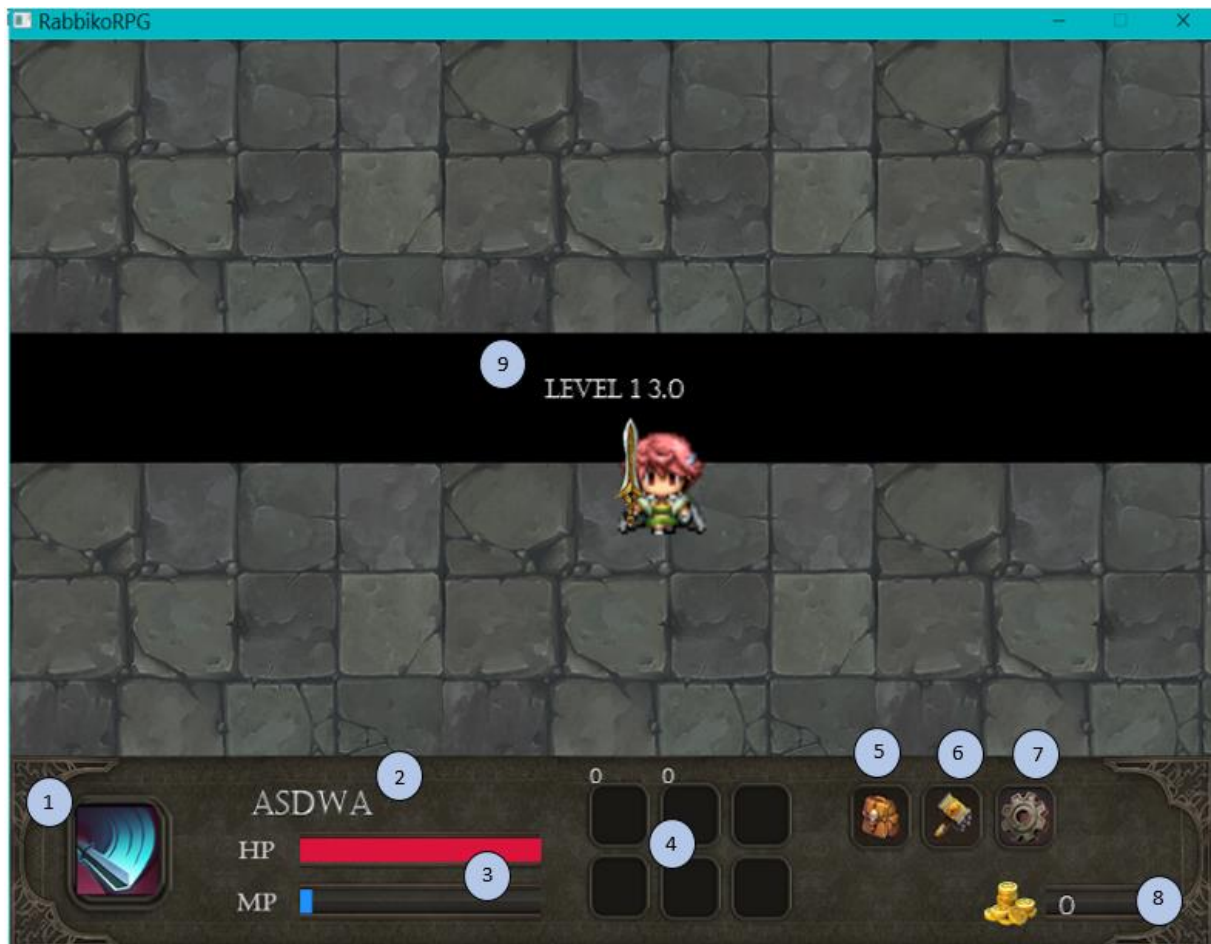
RABBIKKO RPG



Rabikko RPG is a Role-playing game which player plays as Hero role. Player can choose one of three classes , Strength ,Agility or Intelligence ,which based on hero's weapon. Each weapons gives different abilities to hero. To receiving each weapons, player needs to kills the monsters in dungeon to obtain gold coins. Gold coins are used to buy items in the shop menu and those items will be stored in hero's inventory bag which has limited capacity. The goal of this game is to finish each level of the Dungeon.

How to play

- Title screen : Press new game to start the game. This will lead you to next page of the game.
- Opening screen : Enter your name and press "OK" to continue. If you're not entering the name, the game will ask you to enter the name.
- Welcome screen : You will be asked to choose one of three classes , Strength , Agility and Intelligence for your initial ability. You will receive one weapon of your classes.
- Dungeon screen :



- 1 : This is the picture of your current Attribute.
- 2 : Your hero's name.
- 3 : Your hero's Hp and Mp.
- 4 : Inventory shortcut for Hp and Mp potion.
- 5 : Inventory.
- 6 : Shop.
- 7 : Pause the game.
- 8 : Money.
- 9 : Current dungeon level.

Control :

- "A" or "Left arrow" to go left.
- "W" or "Up arrow" to go up.
- "S" or "Down arrow" to go down.
- "D" or "Right arrow" to go right.
- "Space bar" or "Left click" for attack.

Attribute :

1. Strength : This class has the greatest recovery ability among all attribute. The attacking of this class doesn't need Mp.
2. Agility : This class has the greatest attack speed but the least damages. This class also consume Mp for attacking just like Intelligence class so if you're running out of Mp, you can't attack the monster.
3. Itelligence : This class can cause the greatest damage but also consume more Mp than other classes. If you run out of Mp, you won't be able to attack. Make sure that you're supplying enough Mp potions.

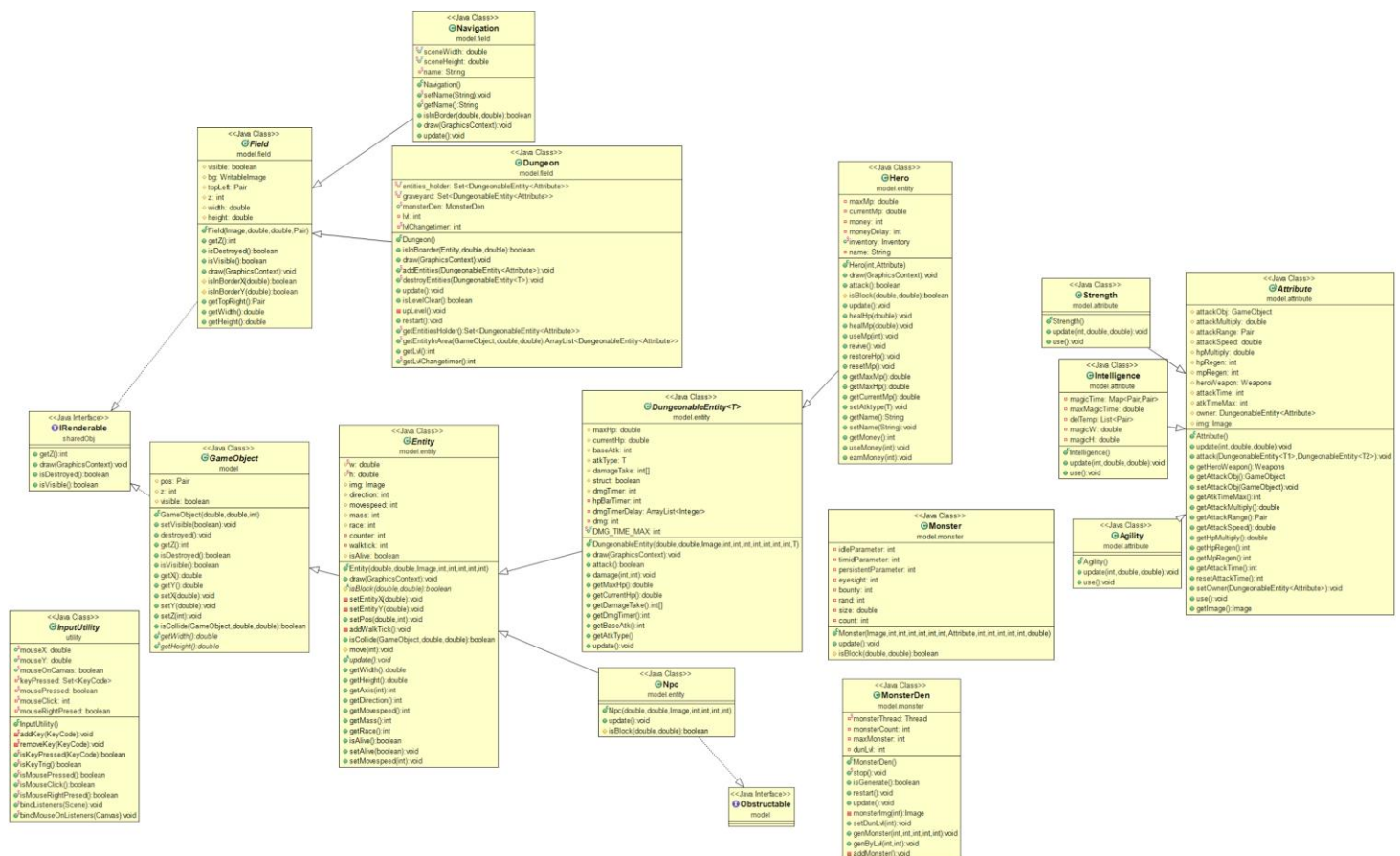
For inventory system, you can store items up to 50 items. You can buy weapons only one for each types but you can buy Hp and Mp potion until your inventory is full.

You can buy items here. If you don't have enough money, your inventory is full or you already bought this weapon before , there will be an alert dialog appears.

You can open this menu to pause your game and click resume to go back to your game.

There will be 3 types of monster just like hero's attribute and have the same effect images. They can attack you if you bump into them. Be careful for your Health point.

1. Revive : Reviving cost you money if you can't afford to revive , your game will be restart instead.
2. Restart : Restart the game to the beginning state. No cost at all.



Interface IRenderable

Package shareObj

interface of all objects that able to be render on the screen

- Method

public int getZ()	Get z axis value for back or front of object.
public void draw(GraphicsContext gc)	Method to draw object on the canvas.
public boolean isDestroyed()	Method indicate object is destroyed.
public boolean isVisible()	Method indicate object visibility.

Field

Package model.field

abstract class of the ground or base element in the game

- Fields

protected boolean visible	Value indicate visibility of objects
protected WritableImage bg	Cropped background image of this field
protected Pair topLeft	(x,y) of top left position
protected int z	Value of z axis, represent the position in foreground and background of objects. positive represent in front, and negative represent behind.
protected double width	Width of the background image
protected double height	Height of the background image

- Methods

public Field(Image bgImage, double width, double height, Pair topLeft)	Constructor initialize all the input field and some default value. Also crop the image into given width and height Default value: visible = true
public int getZ()	Getter of z
public boolean isDestroyed()	Return if this field is no longer use
public boolean isVisible()	Getter of visible
public void draw(GraphicsContext gc)	Draw the image in position of topleft and dimension of width and height
protected boolean isInBorderX(double x)	Is x in the field
protected boolean isInBorderY(double y)	Is y in the field
public Pair getTopLeft()	Getter of topleft

public double getWidth()	Getter of width
public double getHeight()	Getter of height

Dungeon

Package model.field

class of dungeon ground.

- Fields

private static final Set<DungeonableEntity<Attribute>> entities_holder	container contain alive entities in the dungeonContainer of on the ground entities
private static final Set<DungeonableEntity<Attribute>> graveyard	container contain the dead entities waiting for the ritualContainer of death entities waiting for the funeral.
public static MonsterDen monsterDen	monster den, where monster come outMonster den where monster come out
private int lvl	level of the dungeon, indicates the strength of monster.Level indicate the strength of monster
private static int lvlChangetimer	timer for delay before next monster waveTimer indicate delay before the next monster wave.

- Methods

public Dungeon()	Constructor of dungeon <ul style="list-style-type: none"> - Initialize the background using the parent constructor, using the image of dungeon background and dimension equal to the size of the screen - and set z to -99999 to put the image to the back - Initialize the monsterDen - Start the dungeon by set lvl=0 and timer to it max value
public boolean isInBoarder(Entity e, double x, double y)	is the entity still with in the field, graphically.Is the entity stay with in the screen. (calculate graphically)
public void draw(GraphicsContext gc)	draw dungeon background, with topleft position ,and draw floor effect show level change.Draw the level bar according to the current level
public static void addEntities(DungeonableEntity<Attribute> e)	Add the entity to the dungeon container and to the RenderableHolder, and start skip the lvlChangetimer delay

public static <T extends Attribute> void destroyEntities(DungeonableEntity<T> e)	Put the death into the graveyard and make the death disappear from the living sight.
public boolean isLevelClear()	are all the monster in the dungeon and the monster den all deads this level end (no monster on the dungeon ground or with in the monsterden)
private void upLevel()	Level up by 1, and set the timer it max value
public void restart()	Reset the dungeon.
public void update()	Update the dungeon state <ul style="list-style-type: none"> - Performed the funeral and clear the graveyard - Remove the dead entities from the container and put in graveyard - if the death is hero, if he dead, open the dialog and pause the game - If the level is clear up level by one - If with the delay time of stage change, lower the delay time.
public static Set<DungeonableEntity<Attribute>> getEntitiesHolder()	Get the alive entities container
public static ArrayList<DungeonableEntity<Attribute>> getEntityInArea(GameObject object, double x, double y)	Return the arraylist of the entities within the area
public int getLvl()	Getter of lvl
public static int getLvlChangetimer()	Getter of lvlChangetimer

Navigation

Package model.field

Class of navigation bar on the screen

- Fields

public static final double sceneWidth	Screen width of window
public static final double sceneHeight	Screen height of window
private static String name	Name of hero

- Methods

public Navigation()	Initialize navigation bar
public static void setName(String name)	Setter for hero's name
public static String getName()	Getter for hero's name

public boolean isInBorder(double x, double y)	Check if x and y location is in game screen
public void draw(GraphicsContext gc)	Method for drawing navigation bar
public void update()	Checking if player has clicked inventory, shop , pause or shortcut items.

GameObject

Package model

base class for all objects, that will use in the game.

- Fields

protected Pair pos	Pair contain top left (x,y) of image
protected int z	Value of z axis, represent the position in foreground and background of objects. positive represent in front, and negative represent behind.
protected boolean visible	Value indicate visibility of objects

- Methods

public GameObject(double x, double y, int z)	Constructor, initialize all the fields with given parameters.
public void setVisible(boolean visible)	Setter of visible
public void destroyed()	Method use to set all stage for destroy object.
public int getZ()	Getter of z
public boolean isDestroyed()	Is this object have been destroyed
public boolean isVisible()	Getter of visible
public double getX()	Getter of x
public double getY()	Getter of y
public void setX(double x)	Setter of x
public void setY(double y)	Setter of y
public void setZ(int z)	Setter of z
public boolean isCollide(GameObject other, double x, double y)	Value indicates that two objects have intersection on graphics display, or not.
public abstract double getWidth()	Abstract method foreach difference subclasses, indicate object width which depend on object's image.
public abstract double getHeight()	Abstract method foreach difference

	subclasses, indicate object height which depend on object's image.
--	--

Obstructable

Package model

interface of everything on the field that can't walk pass through.

Entity

Package model.entity

- Fields

protected static double w	Width of the entity image
protected static double h	Height of the entity image
protected Image img	Image of the entity
protected int direction	Direction where the entity is facing (front, back, left, right)
protected int movespeed	Number indicate the speed of the entity (0 - 10, where 10 is fastest)
protected int mass	Mass of the entity
protected int race	Race of the entity (human or monster)
private int counter	Walk delay counter
private int walktick	Number indicate the stage entity leg position
protected boolean isAlive	If entity alive, default is true

- Methods

public Entity(double x, double y, Image img, int row, int column, int direction, int movespeed, int mass)	Initialize the field <ul style="list-style-type: none"> - Initialize by given parameter - Set default value w = 32, h = 32, counter = 0 - Crop the spreadsheet image using the given value
public void draw(GraphicsContext gc)	Draw the entity by the give direction and walktick
protected abstract boolean isBlock(double x, double y)	Is the entity have been blocked from moving
private void setEntityX(double x)	Setter for x, if change still with the border and no other blocking the way
private void setEntityY(double y)	Setter for y, if change still with the border and no other blocking the way

public synchronized void setPos(double dPos, int direction)	Setter for pos by the direction
private void addWalkTick()	Add the walktick with the delay
public boolean isCollide(GameObject other, double x, double y)	Is this entity collide with other entity, calculate graphically (should be able to walk behind and infront of the entity)
protected void move(int direction)	Add the walk tick and move to next position
public abstract void update()	Method to Update the stage
public double getWidth()	Getter of entity real width (on the screen)
public double getHeight()	Getter of entity real height(on the screen)
public int getAxis(int direction)	Getter of the direction axis
public int getDirection()	Getter of direction
public int getMovespeed()	Getter of movement speed
public int getMass()	Getter of mass
public int getRace()	Getter of race
public boolean isAlive()	Getter of isAlive
public void setAlive(boolean isAlive)	Setter of isAlive
public void setMovespeed(int movespeed)	Setter of movement speed

DungeonableEntity

Package model.entity

Class of the entity with in the dungeon, the ground where different race attack other.

- Fields

protected double maxHp	Maximum hp of the entity
protected double currentHp	Current hp of the entity
protected int baseAtk	Pure attack damage of the entity
protected T atkType	Attack type of the entity : strength, intelligent, agility
protected int[] damageTake	Total damage take separate by the direction
protected boolean struct	Is entity cannot move or collapse with other
protected int dmgTimer	Timer delay when entity get attack, with in the time delay the entity will be disabled for period of time

private int hpBarTimer	Timer when hp bar appear
private ArrayList<Integer> dmgTimerDelay	Arraylist contain the damage
private int dmg	Damage entity take

- Methods

public DungeonableEntity(double x, double y, Image img, int row, int column, int direction, int movespeed, int mass, int maxHp, int baseAtk, T atkType) public void draw(GraphicsContext gc)	Initialize the field <ul style="list-style-type: none"> - Initialize basic fields using constructor of superclass and the given parameter - Initialize damageTake, currentHp, dmgTimerDelay - Set the default value dmgTimer=0, hpBarTimer =0
public void draw(GraphicsContext gc)	Draw the entity and effect <ul style="list-style-type: none"> - Draw the entity using method from superclass - Draw hp bar and damage effect
public boolean attack()	Default attack style, hit anyone who has different race (wow racist) and in the AOE
public void damage(int dmg, int direction)	Deduce the hp by the damage, active the damage effect, and apply reaction force on this entity
public double getMaxHp()	Getter of max hp (max hp with attribute buff)
public double getCurrentHp()	Getter of current hp
public int[] getDamageTake()	Getter of damage take
public int getDmgTimer()	Getter of damage timer
public int getBaseAtk()	Getter of base attack
public T getAtkType()	Getter of attack type
public void update()	Update the stage using the current field <ul style="list-style-type: none"> - If hp ==0, the entity is death - Reduce the damage timer if within the delay - Update the attack attribute

Hero

Package model.entity

- Fields

private double maxMp	Maximum Mp points of hero
private double currentMp	Current Mp points of hero

private int money	Money
private int moneyDelay	Money increase delay timer
public static Inventory inventory	Hero's inventory bag
private String name	Hero's name

- Methods

public Hero(int direction, Attribute atkType)	<p>Hero constructor</p> <ul style="list-style-type: none"> - Initialize with default hero character image - Set default value: movespeed = 7, mass =50, maxHp = 1000, baseAtk =60 - maxMP=400, currentMp = 0, z = -1, - race of humanity - Initialize the inventory - setAtktype and weapon - Get hero name
public boolean attack()	<p>Use default attack and if attack success full increase mp by base of 15</p> <p>Note: even fail to attack enemy, hero still able to swing the weapon freely.</p>
protected boolean isBlock(double x, double y)	<p>Hero get damaged if walk into the enemy, and can't pass through obstacle on the dungeon</p> <p>Note: able to walk through if in struct state</p>
public void update()	<ul style="list-style-type: none"> - Update the hero from player input - Check if isblock - Increase hp and mp by base heal amount - Hp increase rate is multiply when go up by 20 level.
public void healHp(double i)	Heal the hp with hp regen amount
public void healMp(double i)	Heal the mp with mp regen amount
public void useMp(int i)	Use mp
public void revive()	Revive from death
public void restoreHp()	Restore hp to maxHP
public void resetMp()	Reset Mp to 0
public double getMaxMp()	Getter
public double getMaxHp()	Getter, max hp multiply by the dungeon level
public double getCurrentMp()	Getter

public <T extends Attribute> void setAtktype(T atkType)	Change attack type - Delete the previous attack type - Set attack type and weapon to hero
public String getName()	Getter
public void setName(String name)	Setter
public int getMoney()	Getter
public void useMoney(int i)	Decrease money
public void earnMoney(int i)	Increase money

Monster

Package model.monster

- Fields

private int idleParameter	Value indicate character idleness
private int timidParameter	Value indicate character timid
private int persistentParameter	Value indicate character persistence
private int eyesight	Value indicate range monster see
private int bounty	Reward money after kill this monster
private int rand	Random value
private double size	Monster size, bigger size then stronger monster will be
private int count	Delay count

- Methods

public Monster(Image img, int row, int column, int movespeed, int mass, int maxHp, int baseAtk, Attribute atkType, int idleParameter, int timidParaneter, int persistentParameter, int eyesight, int bounty, double size)	Monster constructor - Initialize all the default field with the given parameter - Set the race to monster
public double getWidth()	Override method, multiple by size
public double getHeight()	Override method, multiple by size
private int heroDirection()	Method calculate hero direction for the monster to flow, however, only if monster can see hero and not too close to the hero (minimum attack range)
public void update()	- Update if monster is not alive or the

	<p>hp reach 0, destroy the monster from the dungeon ground ,destroy it's attackobj, and pay hero the bounty</p> <ul style="list-style-type: none"> - If alive and not under damage effect, random move(may follow hero, or may not) and attack if hero in attack range - Check if monster under the struct state - If alive, update the atktype, dmgtimer, and count
protected boolean isBlock(double x, double y)	<p>If other entity is different race, do other entity damage, and collide with every entity in the area.</p> <p>If under struct state, will not collide with other entity</p>

Attribute

Package model.attribute

- Fields

protected GameObject attackObj	Visual object, use to stimulate the attack
protected double attackMultiply	Multiplier of attack damage
protected Pair attackRange	Area of attack effect
protected double attackSpeed	Attack speed
protected double hpMultiply	Multiplier of max hp
protected int hpRegen	Multiplier of hp regenerate
protected int mpRegen	Multiplier of mp regenerate
protected Weapons heroWeapon	Hero's weapon
protected int attackTime	Attack timer, for attack animation delay
protected int atkTimeMax	Attack timer max
protected DungeonableEntity<Attribute> owner	Owner of this object
protected Image animationImg	Animation image

- Methods

public Attribute()	Set Default attacktimer = 0
public void update(int direction, double x, double y)	<p>Update attack state</p> <ul style="list-style-type: none"> - Decrease attack time if !=0

	- Destroy attackObj if owner already death
public <T1 extends Attribute, T2 extends Attribute> void attack(DungeonableEntity<T1> attacker, DungeonableEntity<T2> other)	Deal damage and apply reaction force to the other, only if other not in damage state
public Weapons getHeroWeapon()	Getter
public GameObject getAttackObj()	Getter
public void setAttackObj(GameObject attackObj)	Setter
public int getAtkTimeMax()	Getter, base attack timer over attack speed
public double getAttackMultiply()	Getter
public Pair getAttackRange()	Getter
public double getAttackSpeed()	Getter
public double getHpMultiply()	Getter
public int getHpRegen()	Getter
public int getMpRegen()	Getter
public int getAttackTime()	Getter
public int resetAttackTime()	Reset attack timer to max
public void setOwner(DungeonableEntity<Attribute> owner)	Set owner of this object, and active the attackobj
public void use()	Reset attack timer
public Image getImage()	Getter

Strength

Package model.attribute

- Methods

public Strength()	Initialize Strength attribute <ul style="list-style-type: none"> - Default using superclass constructor - Get sword form the inventory - Set the field by default, attack multiply =1, attack speed =1, hpmultiply = 1.4, hp regen =5, mp regen =1 - Set attack range to match the weapon - Get animation image - Initialize attack object to match the
-------------------	---

	effect animation
public void update(int direction, double x, double y)	Update the position of attackobj
public int getManaUsed()	Return the amount of mana used to attack which equals to 0

Intelligence

Package model.attribute

- Fields

private Map<Pair, Pair> magicTime	Map of initial position (where magic used) and pair of direction and duration magic travel
private double maxMagicTime	Constant max magic time,
private List<Pair> delTemp	Container of expire magic
private double magicW	Magic spell width
private double magicH	Magic spell height (floor position)

- Methods

public Intelligence()	Initialize Intelligent attribute <ul style="list-style-type: none"> - Default using superclass constructor - Get staff form the inventory - Set the field by default, attack multiply =1.5, attack speed =0.9, hpmultiply = 1, hp regen =1, mp regen =5 - Set attack range to match the weapon - Initialize magicTime, delTemp , and match magicW and magicH to magic effect - Get animation image - Initialize attack object to match the effect animation and override isCollide for magic attack.
public void setOwner(DungeonableEntity<Attribute> owner)	Human can't handle magic well like monster, his attack speed will decrease by half
public void update(int direction, double x, double y)	Update the position of attackobj Update if any magic hit enemy, make the same action as direct attack Update magic state

public void use()	<ul style="list-style-type: none"> - Add new magic to container if spell active less than max spell active constant - Human can't handle magic well like monster, If owner is hero, consume mp by twice as much as normal mp use
public int getManaUsed()	Return the amount of mana used to attack

Agility

Package model.attribute

- Methods

public Agility()	Initialize Agilityattribute <ul style="list-style-type: none"> - Default using superclass constructor - Get bow form the inventory - Set the field by default, attack multiply =1, attack speed =1, hpmultiply = 1.4, hp regen =5, mp regen =1 - Set attack range to match the weapon - Get animation image - Initialize attack object to match the effect animation
public void update(int direction, double x, double y)	Update the position of attackobj
public void use()	<ul style="list-style-type: none"> - If owner is hero, consume mp by constant value
public int getManaUsed()	Return the amount of mana used to attack

Shop

Package model.item

- Methods

public void buy(int index)	Method for buying items. This method will check if hero has enough money to buy that items or if this weapon has bought before.
----------------------------	---

Inventory

Package model.item

- Fields

private static final Useable[] BAG	Inventory bag for storing items of instance Useable
------------------------------------	---

public int bagCapacity	The current capacity of hero's inventory
------------------------	--

- Methods

public Inventory()	Initialize inventory. You will have one weapon in your inventory and capacity is 1
public void reset()	Reset inventory to the state when initialized
public void add(int i)	Add item to inventory and increase bag capacity
public boolean isFull()	Check if inventory is full or if current capacity is 50
public static Useable[] getBag()	Getter for inventory bag

Useable

Package model.item

- Methods

public void use()	Interface method for using items
public void add()	Interface method for adding items
public void reset()	Interface method for reset items to beginning state

Item

Package model.item

- Fields

protected int price	Price of the item
protected Image imgWeapon	Image of the item
protected int amount	Amount of the item

- Methods

public abstract boolean isBuyable()	Abstract method for checking if that items instance can be bought
public abstract boolean isUsable()	Abstract method for checking if that items instance can be used
public int getAmount(),getPrice()	Getter for amount and price
public void add()	Increase amount of that item

Health

Package model.item

- Fields

private final int POINT	Values of healing point of Hp potion
-------------------------	--------------------------------------

- Methods

public Health()	Initialize Hp potion properties set amount price and image
public void use()	Method for using Hp potion If Hp potion can be used , it will increasing hero' Hp by an exact values
public boolean isBuyable()	Checking if hero's inventory is not full and hero has enough money to buy Hp potion
public boolean isUsable()	If the amount of Hp potion is not 0, it can be used
public void reset()	Reset the amount of Hp potion to 0

Mana

Package model.item

- Fields

private final int POINT	Values of healing point of Mp potion
-------------------------	--------------------------------------

- Methods

public Mana()	Initialize Mp potion properties set amount price and image
public void use()	Method for using Mp potion If Mp potion can be used , it will increasing hero' Mp by an exact values
public boolean isBuyable()	Checking if hero's inventory is not full and hero has enough money to buy Mp potion
public boolean isUsable()	If the amount of Mp potion is not 0, it can be used
public void reset()	Reset the amount of Mp potion to 0

Weapon

Package model.item

- Fields

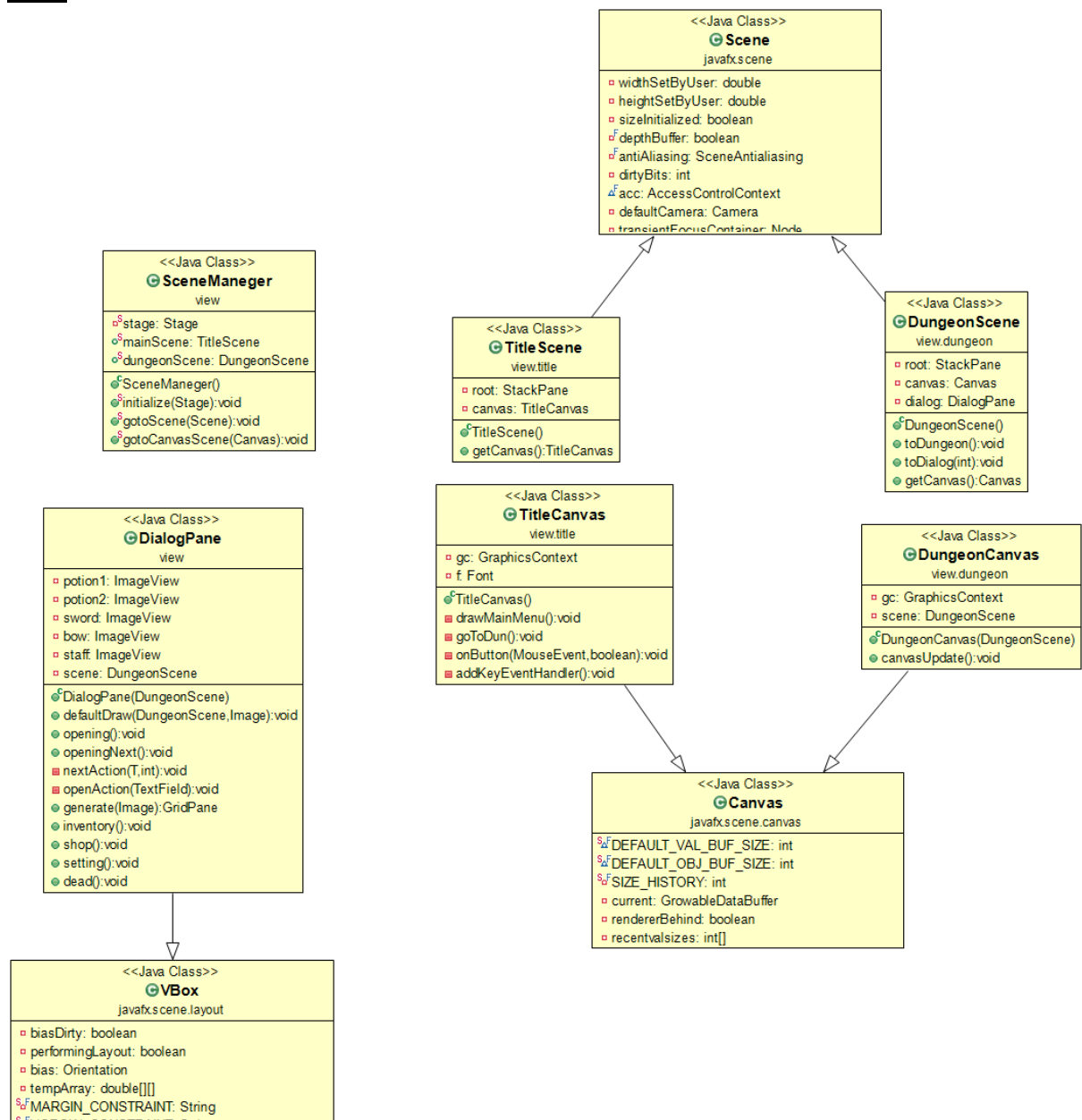
public static final int SWORD	Index of sword in BAG
public static final int BOW	Index of Bow in BAG
public static final int STAFF	Index of Staff in BAG
private int price	Price of the weapon
private Image imgWeapon	Image of that weapon
private int amount	Amount of that weapon
private int type	Type of weapon based on Attributes

- Methods

public Weapons(int price, Image img, Image imgIcon, int type)	Initialize weapon <ul style="list-style-type: none">- Set price of weapon- Set image for weapon- Set icon image for weapon- Set type of weapon
public void held()	Method for equipping the selected weapon. The image of that weapons will show on the hero's hand
public void setWeapon()	Set the weapon to hero <ul style="list-style-type: none">- Take off old weapon- Set new attack type if not the same with the old one- Update screen to show the result of change
public boolean isBuyable()	Checking if the weapons can be bought or not by checking the amount of the weapons and hero's money
public int getPrice()	Getter for weapon price
public int getAmount()	Getter for weapon amount
public void add()	Increase the amount of the selected weapon
public void use()	Use the weapons to attack
public void draw(GraphicsContext gc)	Draw the weapon associate to the current direction of hero
public Image getImage()	Getter for weapon's image
public double getWidth()	Return width of weapon

public double getHeight()	Return height of weapon
public void update(int direction, double x, double y)	Update weapon position to match with hero graphically
public void reset()	Reset the amount of weapon to be 0
public void takeWeaponOff()	Take off the weapon - Destroy hero's weapon and it's attackobj

View



SceneManeger

Package view

- Fields

private static Stage stage	Stage of the game
public static TitleScene mainScene	Instance titlescene
public static DungeonScene dungeonScene	Instance dungeon scene

- Methods

public static void initialize(Stage PrimaryStage)	Initialize primary stage of the game
public static void gotoScene(Scene scene)	Go to scene of the parameter scene
public static void gotoCanvasScene(Canvas canvas)	Go to the scene of the given canvas

DialogPane

Package view

- Fields

private ImageView potion1	Image of Hp potion icon
private ImageView potion2	Image of Mp potion icon
private ImageView sword	Image of sword icon
private ImageView bow	Image of bow icon
private ImageView staff	Image of staff icon
private DungeonScene scene	Scene of dungeon

- Methods

public DialogPane(DungeonScene dunScene)	Initialize dialog in dungeon scene
public void defaultDraw(DungeonScene dunScene, Image image)	Draw default background image for dialog
public void opening()	Create a dialog for getting hero's name from user or welcome screen
public void openingNext()	Create a dialog for selecting initial weapon or attribute
private <T extends Attribute>void	Create new hero with the selected attribute

nextAction(T atkType,int i)	and add selected weapon to inventory bag
private void openAction(TextField textField)	Checking if player has entered the name of hero completely
public GridPane generate(Image img)	Initialize gridPane to store image of items in shop and inventory and close button to close down the dialog
public void inventory()	This method is for generating inventory page <ul style="list-style-type: none"> - You can use items from this screen - You can see items amount in this screen
public void shop()	This method is for generating shop page <ul style="list-style-type: none"> - You can buy items in this page - You can see price and details of items in this page
public void setting()	This is the page when you clicked pause icon. There is a resume button in this page
public void dead()	This is the page that will pop up when your HP reaches 0. There are 2 options to deal with in this screen

TitleScene

Package view.title

- Fields

private StackPane root	Container for tile scene
private TitleCanvas canvas	Canvas of title screen

- Methods

public TitleScene()	Initialize the tile scene add the root container and canvas to the scene
public TitleCanvas getCanvas()	Getter for titleCanvas

TitleCanvas

Package view.title

- Fields

private GraphicsContext gc	Graphic context for title canvas
private Font f	Font using in title canvas

- Methods

public TitleCanvas()	Initialize the canvas of SCENE_WIDTH and SCENE_HEIGHT size. Draw the main menu screen and add event handler to this canvas
private void drawMainMenu()	Draw the main menu screen with contains the game's name, new game button and credit
private void goToDun()	This is the method for changing the current scene to dungeon scene
private void onButton(MouseEvent event, boolean isGoNext)	Add mouse event for new game button
private void addKeyEventHandler()	Add key event for this canvas

DungeonScene

Package view.dungeon

- Fields

private StackPane root	Container for dungeon scene
private Canvas canvas	Canvas for dungeon scene
private DialogPane dialog	Dialog instance for opening a dialog menu

- Methods

public DungeonScene()	Initialize the scene to display the dialog of welcome screen and other dialog
public void toDungeon()	Changing current scene to dungeon scene
public void toDialog(int c)	Go to the dialog of selected number For 0 : go to opening screen For 1 : open the inventory For 2 : open the shop For 3 : open the setting For 4 : open the dead-revive dialog For 5 : open the select-weapon screen
public Canvas getCanvas()	Getter for canvas

DungeonCanvas

Package view.dungeon

- Fields

private GraphicsContext gc	Graphic context for canvas
private DungeonScene scene	Scene of Dungeon scene

- Methods

public DungeonCanvas(DungeonScene scene)	Initialize the canvas filled with black color and bindMouseListeners to this canvas
public void canvasUpdate() throws ConcurrentModificationException	This method is used if the resources has not completely loaded

Controller

Main

Package main

- Fields

public static MediaPlayer m	Media player to control the game's theme song
public static boolean isGameRunning	The value indicates whether the game is running

- Methods

public void start(Stage primaryStage)	The main entry point for the game
public void stop() throws Exception	This method clears the remaining threads and stop the game
public static void main(String[] args)	An entry point of the game

DungeonMain

Pakage main

- Fields

private static DungeonCanvas canvas	Canvas of dungeon scene
private static GameLogic logic	GameLogic instance
private static ForceManeger forceManager	ForceManager instance

- Methods

public DungeonMain()	Initialize logic, canvas and forceManager
----------------------	---

private static AnimationTimer animation = new AnimationTimer()	Initialize the AnimationTimer loop to update the game state
public static void start()	Changing isGameRunning to true and start the animation loop
public static void stop()	Changing isGameRunning to false and stop the animation loop
public static DungeonCanvas getCanvas()	Getter for canvas
public static GameLogic getLogic()	Getter for logic
public static ForceManeger getForceManager()	Getter for forceManager

RenderableHolder

Package shareObj

class of renderable's container. provide with basic method for using this container.

- Fields
-

private static final RenderableHolder instance	Singleton of this class
private List<IRenderable> entities	List contain all object that will be graphically render
private Comparator<IRenderable> comparator	Comparator of two IRenderable objects.
public static Font diaLogFont	Font for using in this game

- Methods

static { ResourceLoader.loadResource(); }	Static setup, use to initiate resources load.
public RenderableHolder()	Constructor
public void add(IRenderable entity)	Method use for adding IRenderable object into the entities list.
public void sort() throws IllegalArgumentException	Method for sorting the entities list by z value.
public void clear()	Remove all member of the entities list.
public void update() throws IllegalArgumentException	Method for update by removing the destroyed objects and sort the entities list by z value.
public static RenderableHolder	Getter for instance

getInstance()	
public List<IRenderable> getEntities()	Getter for entities
public int size()	Method returning size of current entities list.

GameLogic

Package logic

- Fields

public static Dungeon dungeon	Instance dungeon
public static Hero hero	Instance hero
public static Navigation navig	Instance navigation bar

- Methods

public GameLogic()	Initialize dungeon, and navigation, and add them to RenderableHolder
public <T extends Attribute>void newHero(T atkType)	Method to construct new hero and add hero to dungeon ground
public void logicUpdate()	Update dungeon and navigation bar

ForceManeger

Package utility

- Methods

public static <T extends Attribute> void reactionEffect(DungeonableEntity<T> entity, int axis)	Push the entity with the reaction force in the axis force was apply on
private static <T extends Attribute> Pair totalForce(DungeonableEntity<T> entity)	Calculate current total force in each axis
public static <T extends Attribute> int calculateForce(int force, int axis, DungeonableEntity<T> dungeonableEntity)	Method to calculate given force to match with the screen scale
public static <T extends Attribute> int calculateDirection(int direction)	Method to calculate reaction direction

MonsterDen

Package model.monster

- Fields

private static Thread monsterThread	Tread for monster generate
private int monsterCount	Counter for number of monsters
private int maxMonster	/The number of maximum monsters
private int dunLvl	The level of dungeon

- Methods

public MonsterDen()	monsterDen constructor <ul style="list-style-type: none">- Set monstercount = 0, maxMonster = 0- Initialize monsterThread, wait for period of time and generate only if RenderableHolder have less than max number of renderable and hero is alive- If monster less than maximum monster for this level, generate new monster- Level clear, generate new maxMonster, and reset monsterCount to 1- Start the thread
public static void stop()	Stop the monsterThread
public boolean isGenerate()	Return if still in generating process
public void restart()	Reset the value of monsterCount and maxMonster to 0
private Image monsterImg(int i)	Get monster image
public void setDunLvl(int dunLvl)	Setter for dunLvl
public void genMonster(int img, int row, int col, int size, int time)	Generate monster base on the parameter
public void genByLvl(int img, int rand)	Generate monster, monster strength indicate by current level
private void addMonster() throws InstantiationException, IllegalAccessException, IllegalArgumentException, InvocationTargetException, SecurityException	Add monster to dungeon randomly, base on current level. normal monster, boss monster and bonus monster

Utility

Constant

Package Utility

- Fields

public static final double SCENE_WIDTH	Width of the screen
public static final double SCENE_HEIGHT	Height of the screen
public static final int SCENE_X_AXIS	Value represent x axis
public static final int SCENE_Y_AXIS	Value represent y axis
public static final int ENTITY_FRONT	Value represent entity front direction
public static final int ENTITY_LEFT	Value represent entity left direction
public static final int ENTITY_RIGHT	Value represent entity right direction
public static final int ENTITY_BACK	Value represent entity back direction
public static final int ENTITY_HUMANITY	Value represent humanity race
public static final int ENTITY_MONSTER	Value represent monster race
public static final int DUNGEON_CHANGE_TIME_MAX	Maximum dungeon change time, delay before the next wave
public static final double NAVIG_WIDTH	Width of Navigation bar
public static final double NAVIG_HEIGHT	Height of navigation bar
public static final int NAVIG_BAR_HEIGHT	Used as parameter to compute the y location of the hero's name of the navigation bar
public static final int NAVIG_BORDER_WIDTH	The value to indicate the location for drawing text of hero's name
public static final int BOUNTY_MULTIPLYER	Multiplier of monster bounty
public static final int ENTITY_WALK_TICK_DELAY	Value represent delay timer for walk stepanimation
public static final int ENTITY_WALK_STAGE	Value represent number of stage for walk animation
public static final int DMG_TIME_MAX	Value represent maximum cool down time for the entity that take the damage
public static final double BASE_HEAL_AMOUNT	Value represent base amount for healing hp or mp

public static final int BASE_ATTACK_TIMER_MAX	Value represent maximum cooldown time for attack delay
public static final int SWORD	Value represent index of sword in the inventory's bag
public static final int BOW	Value represent index of bow in the inventory's bag
public static final int STAFF	Value represent index of staff in the inventory's bag
public static final int BASE_MP_USE	Value represent amount for mp consume per unit of time
public static final double MAX_MAGIC_TIME	Value represent maximum duration for magic spell , and also indicate distance magic can be travel
public static final int MAX_SPELL_ACTIVE	Value represent maximum number of spell can be active by one mage.
public static final int MAX_NUMBER_RENDERABLE_HOLD	Value represent maximum preferred number of member with in renderable holder.

InputUtility

Package Utility

- Fields

public static double mouseX, mouseY	The position of x and y location of mouse
public static boolean mouseOnCanvas	Checking if mouse is in the canvas
private static Set<KeyCode> keyPressed	Container for storing the keys that have been pressed
private static boolean mousePressed	Value for indicating if mouse is pressed
private static int mouseClicked	If mouse is clicked, this value will be 1. If not clicked the value will be 0
private static boolean mouseRightPresed	Value for indicating if mouse is right clicked

- Methods

private static void addKey(KeyCode code)	Add the pressed KeyCode to the keyPressed
private static void removeKey(KeyCode code)	Remove the KeyCode from the keyPressed
public static boolean isKeyPressed(KeyCode code)	Determine whether the KeyCode is pressed by searching the KeyCode in set keyPressed

public static boolean isMousePressed()	The method returns the value of mousePressed
public static boolean isMouseClicked()	Increasing the value of mouseClicked when mousePressed is true by 1 and set it to 0 when mousePressed is false
public static boolean isMouseRightPresed()	Return the value of mouseRightPressed
public static void bindListeners(Scene scene)	Event handler for MouseEvent and KeyEvent to set the value of fields in this class. If the key is pressed it will be store in keyPressed set and remove from set when that key is released. If mouse is left-clicked the value of mousePressed will change to true same as mouseRightPressed if mouse is right-clicked.
public static void bindMouseOnListeners(Canvas canvas)	Event handler for MouseEvent in that canvas to set the value of fields in this class. If mouse is entered the screen ,this method set the mouseOnCanvas to true and false when mouse is exited the canvas. When mouse is entered the screen mouseX and mouseY will be stored with the values of x and y position of mouse.

Pair

Package Utility

Create fake pair, data structure, with some additional method

- Fields

public double x	first field
public double y	Second field

- Methods

public Pair(double x, double y)	constructor
public Pair(Pair o)	Copy constructor
public void add(double x,double y)	Method adding to the field
public double diffX(double dx)	Return different from dx to x
public double diffY(double dy)	Return different from dy to y
public double diffD(double dx,double dy)	Return distance from (dx,dy) to (x,y)

RandomUtility
Package Utility
- Fields

private static final Random random	Instance of java.util.Random
------------------------------------	------------------------------

- Methods

public static int randomInt(int from, int to)	Return random int in given range
public static int randomByPercent(int rand,int base,int percent)	Return random 1-4 with weight factor
public static int randomByLevel(int lvl)	Return random number base on dungeon current level
public static long randomTime(int sizeFactor)	Return random random time base on given size factor. Size factor indicate the size of container. If too many member in container, return the time to slow down the generating process
public static int random()	Return random using current time

ResourceLoader
Package Utility
- Fields

public static Font diaLogFont	Font using in this game
public static FontLoader fontLoader	Fontloader for computing text width and height
public static Image mainImage	Image of the title screen
public static Image mainBtnImage	Image for New game button
public static Image dialogFrame	Image for dialog
public static Image dungeonBg	Image for dungeon background
public static Image humanImage	Image for Hero
public static Image[] monsterImage	Image for monsters
public static Image navigBar	Image for navigation bar
public static Image inven	Image for inventory button
public static Image shop	Image for shop button
public static Image setting	Image for pause button

public static Image dead	Image for dead-revive dialog
public static Image hpPotion	Image for HP potion
public static Image mpPotion	Image for Mp potion
public static Image sword	Image for sword icon
public static Image sword2	Image of sword for Hero
public static Image bow	Image for bow icon
public static Image bow2	Image of bow for Hero in right direction
public static Image bow3	Image of bow for Hero in left direction
public static Image staff	Image for staff icon
public static Image staff2	Image of staff for Hero
public static Image sEffect	Effect picture for Strength type attribute
public static Image sEffect2	Effect picture for Strength type attribute
public static Image aEffect	Effect picture for Agility type attribute
public static Image aEffect2	Effect picture for Agility type attribute
public static Image aEffect3	Effect picture for Agility type attribute
public static Image aEffect4	Effect picture for Agility type attribute
public static Image mEffect	Effect picture for intelligence type attribute
public static Image monsterEffect	Effect picture for Monster
public static Image strength	Image symbol for Strength attribute
public static Image agility	Image symbol for Agility attribute
public static Image intelligence	Image symbol for Intelligence attribute
public static AudioClip clickSound	Sound for clicking
public static Media titleBgm	Theme song of Rabbikko RPG game
public static AudioClip coin	Sound for purchasing item
public static AudioClip heal	Sound for using the potion
public static Thread loadThread	Thread used to loading the resources in secondLoading

- Methods

static	Call the loadResouce method to load the resources for game
--------	--

public static void loadResource()	This method used to load all the resources that need to be used in the game
public static void initializeSecondLoad()	Initialize thread for resource load in second time
public static void startSecondLoad()	Start second load thread
public static boolean isLoadFinish()	Checking if loadThread finished loading the resources by determine the state of Thread. If this Thread is alive the resources aren't ready to be used