

InfinityBug : Ja Boom Mine

Created by

Chomtana Chanjaraswichai 6131307621

Pongsakorn Pengboon 6131326521

2110215 Programming Methodology

Semester 2 Year 2018

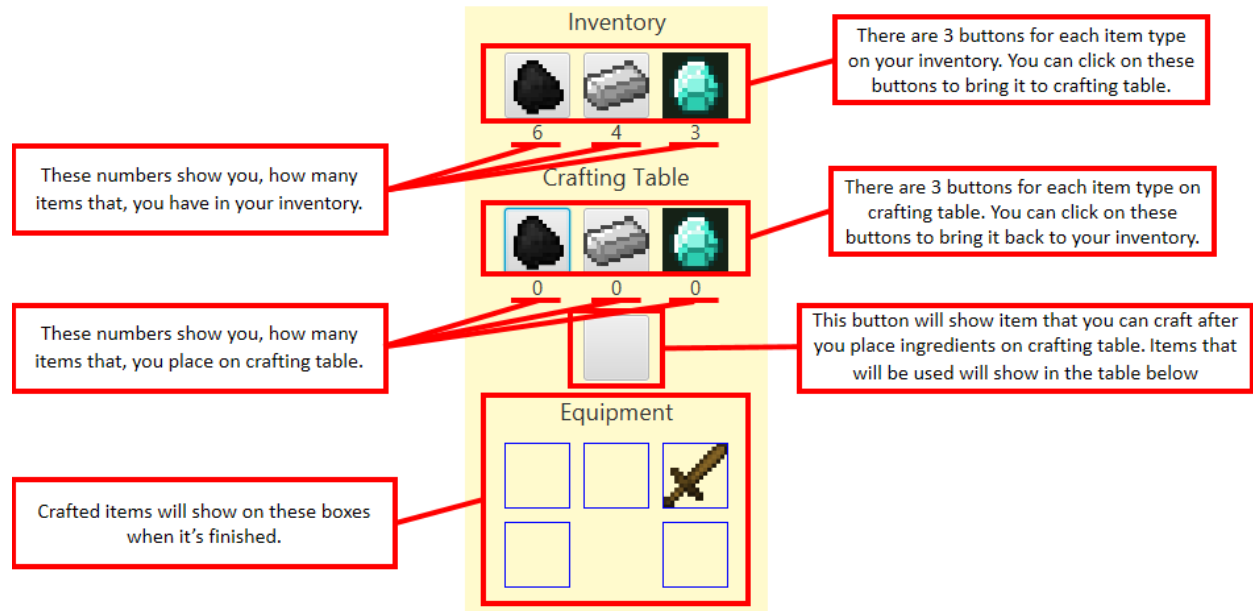
Chulalongkorn University










About this game





Category	Name	Picture	Note
Player's Character	Player		Can attack on range 1 block.
	Player's Bomb		It can be damage only player and ore.
Ore	Coal Ore		Can be destroyed by player's bomb, attacking and boomer.
	Iron Ore		Can be destroyed by player's bomb, attacking and boomer.
	Diamond Ore		Can be destroyed by player's bomb, attacking and boomer.
Item	Coal		Drop by destroying coal ore.
	Iron		Drop by destroying iron ore.
	Diamond		Drop by destroying diamond ore.
Equipment	Wooden Sword		Starting item
	Iron Sword		Can be Crafted. Increase your physical damage.
	Iron Helmet		Can be Crafted. Increase your physical armor.
	Iron Armor		Can be Crafted. Increase your physical armor.
	Iron Pant		Can be Crafted. Increase your physical armor.

Equipment	Iron Boots		Can be Crafted. Increase your physical armor and movement speed.
	Diamond Sword		Can be Crafted. Increase your physical damage.
	Diamond Helmet		Can be Crafted. Increase your physical armor.
	Diamond Armor		Can be Crafted. Increase your physical armor.
	Diamond Pant		Can be Crafted. Increase your physical armor.
	Diamond Boots		Can be Crafted. Increase your physical armor and movement speed.
Monster	Zombie		Can be attack to player on range 1 block.
	Boomer		Charge 1.5 second before blow up when player is in range 2 block

Crafting & Inventory System



Ingredients					
Item	picture	Coal	Iron	Diamond	Note
					
Iron Sword		0	3	0	-
Iron Helmet		2	2	0	-
Iron Armor		1	3	0	-
Iron Pant		3	2	0	-
Iron Boots		4	2	0	-
Diamond Sword		0	2	3	You can craft diamond sword after you have Iron sword.

Diamond Helmet		4	0	2	You can craft diamond helmet after you have Iron helmet.
Diamond Armor		2	0	3	You can craft diamond armor after you have Iron armor.
Diamond Pant		6	0	2	You can craft diamond pant after you have Iron pant.
Diamond Boots		8	0	2	You can craft boots armor after you have Iron boots.

Level System

When you kill monster, then they will drop EXP for you to collect and level up. Your status (max HP, HP regeneration, damage and armor) will be increase by increasing level.

Monster

The more you play the game, the more stronger monster get.

Key for playing

Action	Key
Move left	"A" or "LEFT"
Move right	"D" or "RIGHT"
Move up	"W" or "UP"
Move down	"S" or "DOWN"
Facing to left direction	Scroll mouse to left side of player's character
Facing to right direction	Scroll mouse to right side of player's character
Facing to top direction	Scroll mouse to top side of player's character
Facing to bottom direction	Scroll mouse to bottom side of player's character
Attack to facing direction	LEFT-Click
Placing player's bomb	RIGHT-Click
Unlimited HP command	"p"

Implementation Details

Package application

Class Main Extends Application

Field

+ <u>GameArea gameArea</u>	Center of GUI
+ <u>StackPane eventPane</u>	Transparent Pane in center of GUI for event handing.
+ <u>Random random</u>	Random number generator object.
+ <u>CameraController cameraController</u>	Object for control camera view.
+ <u>ControlPanel controlPanel</u>	Window at left side of GUI.
+ <u>Main instance</u>	Instance of class Main.
+ <u>StackPane root</u>	Root pane in scene.

Method

+ <u>Main getInstance()</u>	Return Instance of class Main.
+ <u>void main(String[] args)</u>	An entry point of the application.
+ <u>void endgame()</u>	This method use to exit program.
+ <u>void start(Stage primaryStage)</u>	Start Javafx.
+ <u>StackPane getEventPane()</u>	Return event pane component.
+ <u>void playBgSound()</u>	Play background sound.

Package controller

Class CameraController

Field

- Entity center	Entity that must be center point of view.
- GameArea area	Center of GUI.
+ <u>CameraController cameraController</u>	Object for control camera.

Constructor

+ CameraController(Entity center,GameArea area)	Generate cameracontroller, there is entity center to be center point of game area.
+ CameraController()	Initial player to be center point of game area.

Method

+ void performSetCenter()	Set view to entity center must be center point.
+ <u>CameraController getInstance()</u>	Get camera controller instance.
+ void <u>registerInstance(CameraController camm)</u>	Register (Set) camera controller instance

Class EventController

Field

- Scene scene	Scene of javafx GUI.
- AnimationTimer timer	Register event from keyboard.
- boolean LEFT	Check if either "Left" or "a" key is being pressed.
- boolean RIGHT	Check if either "Right" or "d" key is being pressed.
- boolean UP	Check if "Up" or "w" key is being pressed.
- boolean DOWN	Check if "Down" or "s" key is being pressed.
- <u>ArrayList<Runnable> onLoadRunnable</u>	Function run when onload.
- <u>Boolean onLoedRunned</u>	Check is onload running?

Constructor

+ EventController(Scene S)	EventController instance with Scene s.
----------------------------	--

Method

- void registerAnimationTimer()	Register animation timer instance.
- void registerOnKey()	Register keyboard event.
- void registerMouse()	Register mouse event.
+ Scene getScene()	Get scene of javafx gui.
+ <u>void onLoad(Runnable r)</u>	Add function from r to onLoadRunnable.
+ <u>void performOnLoad()</u>	Run function that should run on onload.

Class SpawnController

Field

+ <u>double spawnDelayStart</u>	Initialize time delay for spawning a monster.
+ <u>int spawnRadius</u>	Determine the distance between the player and the monster spawn points.

Constructor

+ SpawnController()	Create spawn controller instance.
---------------------	-----------------------------------

Method

- void doMoveAI()	Perform Move AI.
- void spawn()	Spawn monster.

Class TimeController

Field

+ <u>long startingTime</u>	Time at beginning program
----------------------------	---------------------------

Method

+ <u>long getCurrentTime()</u>	Return how long is program running.
+ <u>void resetCurrentTime()</u>	Reset startingTime to current time

Package effect

Class AttackEffect **extends** AutokillEffect

Field

- double atkDamage	Attack damage
- Entity attacker	Attacker

Constructor

+ AttackEffect(Entity target, double atkDamage, Entity attacker)	Create attack effect for attacker attack target with damage atkDamage.
+ AttackEffect(int row, int col, double atkDamage, Entity attacker)	Create attack effect for attacker attack at (row, col) with damage atkDamage

Method

+ <u>synchronized void playSound()</u>	Play attack sound
+ void onCollideWith(Entity target)	Run when effect collide with something at starting of effect, in this case attack target with physical damage atkDamage.
+ String getIcon()	Return effect icon.
Getter & Setter	Getter & Setter for each Field.

Class BombEffect **extends** AutokillEffect

Field

- double atkDamage	Attack damage
- Entity attacker	Attacker
- Thread bombStarter	Bomb start delay thread
- int radius	Bomb radius

Constructor

+ BombEffect(Entity e, double atkDamage, int radius)	Initial attack effect for attacker e place bomb with damage atkDamage and radius.
+ BombEffect(int row, int col, double atkDamage, int radius, Entity attacker)	Initial attack effect for attacker place bomb at (row, col) with damage atkDamage and radius.

Method

+ void onCollideWith(Entity target)	Run if effect collide with something at starting of effect.
+ String getIcon()	Return effect icon.
Getter & Setter	Getter & Setter for every field except bombStarter.

Class FireEffect **extends** AutokillEffect

Field

- double atkDamage	Attack damage
- Entity attacker	Attacker

Constructor

+ FireEffect(Entity e, double atkDamage)	Initial bomb fire effect for attacker e with damage atkDamage.
+ FireEffect(int row, int col, double atkDamage, Entity attacker)	Initial bomb fire effect for attacker attacking at (row, col) with damage atkDamage.

Method

+ <u>synchronized void playSound()</u>	Play bomb fire sound.
+ void onCollideWith(Entity target)	Run if effect collide with something at starting of effect, in this case attack target with bomb damage atkDamage.
+ String getIcon()	Return effect icon.
Getter & Setter	Getter & Setter for each field.

Class AutokillEffect **extends** Effect

Field

- int duration	Duration of this effect.
- Thread autokillTimer	Control autokill delay.

Constructor

+ AutokillEffect(int row, int col, int duration)	Create effect at row col for duration.
--	--

Method

Getter & Setter	Getter & Setter for duration.
-----------------	-------------------------------

Class Effect **extends** Entity

Field

- Block effectBlock	Block containing effect.
- int row	Row effect.
- int col	Column effect.

Constructor

+ Effect(int row, int col)	Create Effect at (row, col)
----------------------------	-----------------------------

Method

+ abstract void onCollideWith(Entity target)	Run when effect collide with something at starting of effect.
Getter & Setter	Getter & Setter for each field.

Package entity

Interface Attackable

Method

+ abstract void attack(CanTakePhysicalDamage target)	Attack to target.
+ abstract Boolean canAttack(CanTakePhysicalDamage target)	Check can attack to target.

Interface CanTakeBombDamage extends HasHP

Method

+ double takeBombDamage(double damage)	Take bomb damage.
--	-------------------

Interface CanTakePhysicalDamage extends HasHP

Method

+ double takePhysicalDamage(double damage)	Take physical damage.
--	-----------------------

Interface GiveEXPOnDead

Method

+ int getExpGived()	The number of EXP that given to killer when this entity has been killed.
---------------------	--

Interface HasArmor

Method

+ double getArmor()	Return current armor of this entity.
+ void setArmor(double armor)	Set armor of this entity.

Interface HasHP

Method

+ double getHP()	Return current HP.
+ void setHP(double hp)	Set current HP.
+ boolean isAlive()	Check entity has HP more than 0 or not.

Interface HasInventory

Method

+ Inventory getInventory()	Return inventory instance of this entity.
----------------------------	---

Interface HasLevel

Method

+ int getLevel()	Return current level.
+ void setLevel(int level)	Set current level.
+ int getCurrentLevelEXPNeeded()	Return the number of EXP to next level
+ void receiveEXP(int exp)	this entity receive EXP
+ int getExp()	Get Current EXP
+ void setExp(int exp)	Set Current EXP

Interface Moveable

Method

+ boolean canMoveTo(int row,int col)	This entity can move to row col ? if cannot move, throw UnmoveableException
+ boolean moveTo(int row,int col)	move this entity to row col if cannot move, thrown UnmoveableException
+ double getRenderRow()	Return row used for rendering.
+ double getRenderCol()	Return col used for rendering.
+ int getRow()	Return row of this entity.
+ int getCol()	Return col of this entity.

Class MoveCollideException **extends** UnmoveableException

MoveCollideException will be thrown, when an entity collide with another entity. It has message and contains action for another time getWith(): get what this exception thrower collide with

Field

- Entity with	The entity, that's collided.
---------------	------------------------------

Constructor

+ MoveCollideException(Entity with)	Initialize the entity, that's collided.
+ MoveCollideException(Entity with, String msg)	Initialize message and entity, that's collided.
+ MoveCollideException(Entity with, String msg, Runnable action)	Initialize message, action and entity, that's collide.
+ MoveCollideException(Entity with, Runnable action)	Initialize action and entity, that's collide.

Method

Getter	Getter for field with
--------	-----------------------

Class MoveOutOfBoundException **extends** UnmoveableException

MoveOutOfBoundException will be thrown, when you move out of bound with message and contains action for another time.

Constructor

+ MoveOutOfBoundException()	Message "Cannot move out of bound" to the exception
+ MoveOutOfBoundException(String msg)	Initial message to the exception.
+ MoveOutOfBoundException(String msg, Runnable action)	Initial message and action to the exception.
+ MoveOutOfBoundException(Runnable action)	Initial action to the exception.

Class NullEntity **extends** Entity

Field

- int row	Row space.
- int col	Column space.

Constructor

+ NullEntity(int row, int col)	Initial row and column space.
--------------------------------	-------------------------------

Method

+ String getIcon()	Return empty string.
+ void render(Block b)	Render this entity to block b and remove background of block b.
Getter & Setter	Getter & Setter for each field.

Enum Direction

LEFT, RIGHT, UP, DOWN

Class Player **extends** Monster **implements** HasHP, HasArmor, CanTakePhysicalDamage, CanTakeBombDamage, HasInventory, HasLevel, Attackable

Field

+ <u>int START_ROW</u>	Starting player's row.
+ <u>int START_COL</u>	Starting player's column.
+ <u>Player mainPlayer</u>	Main player.
- Direction facing	Direction that, player's facing.
- double hp	Number of player's HP.
- double maxhp	Number of player's max HP.
- double regenhp	Number of player's HP regeneration.
- double armor	Number of player's armor.
- double atkDamage	Number of player's damage.
- double bombDamage	Number of damage that, player's taken by bomb.
- int bombRadius	Radius of player's bomb.

- Inventory inventory	Player's inventory.
- Crafter crafter	Player's crafter.
- int level	Player's level.
- int expTable[]	EXP for each level.
- int exp	Number of player's EXP.
- Thread bombThrottle	Thread used for delay bomb.

Method

+ int getCurrentLevelEXPNeeded()	Return number of EXP to next level.
+ void receiveEXP(int exp)	Receive EXP.
+ int getExp()	Return current EXP.
+ void setExp(int exp)	Set current EXP.
+ double getMaxhp()	Return current max HP.
+ void setMaxhp(double maxhp)	Set current max HP.
+ double getRegenhp()	Return current HP regeneration.
+ void setRegenhp(double regenhp)	Set current HP regeneration.
+ void attack(CanTakePhysicalDamage target)	Attack to target with physical damage.
+ void attack()	Attack in direction that, player's facing.
+ void bomb()	Placing player's bomb.
+ boolean canAttack(CanTakePhysicalDamage target)	Check can player attack to target.
+ String getIcon()	Return player's icon.
+ void handleUnmoveableException(UnmoveableException e)	Reaction when UnmoveableException was thrown.
+ void moveLeft()	Move player to the left direction of map.
+ void moveRight()	Move player to the right direction of map.
+ void moveUp()	Move player to the up direction of map.
+ void moveDown()	Move player to the down direction of map.
+ boolean moveTo(int row,int col)	Check can player move to (row, col) location.
+ int getMoveDelay()	Return player's delay for another move.
+ Direction getFacing()	Return direction that, player facing.
+ void setFacing(Direction facing)	Set direction that, player facing.
+ ArrayList<EnemyMonster> getSurroundingEnemies()	Return surrounding monsters.
+ double getArmor()	Return number of current armor.
+ void setArmor(double armor)	Set number or current armor.
+ double getAtkDamage()	Return current attack damage.
+ void setAtkDamage(double atkDamage)	Set current attack damage.
+ double getBombDamage()	Return current bomb damage.
+ void setBombDamage(double bombDamage)	Set current bomb damage.
+ int getBombRadius()	Return radius of bomb effect.
+ void setBombRadius(int bombRadius)	Set radius of bomb effect.
+ double takePhysicalDamage(double damage)	Player take physical damage.
+ double takeBombDamage(double damage)	Player take bomb damage.
+ double getHP()	Return current HP.
+ void setHP(double hp)	Set current HP.

+ Inventory getInventory()	Return player's inventory.
+ Crafter getCrafter()	Return player's crafter

Class UnmoveableException **extends** ExceptionWithAction

UnmoveableException throw, when you cannot move to some place with message and contains action for another time.

Constructor

+ UnmoveableException()	Initial ExceptionWithAction.
+ UnmoveableException(String msg)	Initial message to ExceptionWithAction.
+ UnmoveableException(String msg, Runnable action)	Initial message and action to ExceptionWithAction.
+ UnmoveableException(Runnable action)	Initial action to ExceptionWithAction.

Package entity.monster

Class Boomer **extends** EnemyMonster

Field

- double bombDmg	Number of damage when boomer blew up.
- int bombRadius	Radius of boomer's bomb effect.
- Thread bombThrottle	Thread used to delay bombing

Constructor

+ Boomer(int row,int col)	Spawn at (row, col).
---------------------------	----------------------

Method

+ void handleUnmoveableException(UnmoveableException e)	Handle UnmoveableException. In this case we do nothing if cannot move.
+ boolean moveTo(int row,int col)	Check can boomer move to (row, col) location. And if player in bomb radius It will charge for 1.5 second then bomb
+ void attack(CanTakePhysicalDamage target)	Attack (blew up) to target.
+ boolean canAttack(CanTakePhysicalDamage target)	Check can boomer do physical attack to target.
+ String getIcon()	Return boomer's icon.
+ int getExpGived()	Return EXP drop when boomer was killed.

Class EnemyMonster **extends** Monster **implements** GiveEXPOnDead

Field

- Direction facing	Direction that, monster is facing.
# int moveDelay	Monster's delay for another move.

Constructor

+ EnemyMonster(int row,int col)	Spawn monster at (row, col) location.
---------------------------------	---------------------------------------

Method

+ abstract void handleUnmoveableException(UnmoveableException e)	Handle UnmoveableException.
+ void moveLeft()	Move monster to the left direction of map.
+ void moveRight()	Move monster to the right direction of map.
+ void moveUp()	Move monster to the up direction of map.
+ void moveDown()	Move monster to the down direction of map.
+ boolean moveTo(int row,int col)	Check can monster move to (row, col) location.
+ int getMoveDelay()	Return MoveDelay.
+ Direction getFacing()	Return direction that, monster facing.
+ void setFacing(Direction facing)	Set direction that, monster facing.

Class Zombie **extends** EnemyMonster **implements** Attackable

Field

- double atkDamage	Number of damage when zombie attack.
--------------------	--------------------------------------

Constructor

+ Zombie(int row,int col)	Spawn zombie at (row, col) location.
---------------------------	--------------------------------------

Method

+ double getAtkDamage()	Return current damage from zombie attacking.
+ void setAtkDamage(double atkDamage)	Set base zombie attack damage.
+ boolean canAttack(CanTakePhysicalDamage target)	Check can zombie attack to target.
+ void handleUnmoveableException(UnmoveableException e)	Handle UnmoveableException. In this case we try to attack target if MoveCollideException has been thrown.
+ String getIcon()	Return zombie's icon.
+ int getExpGived()	Return EXP drop when zombie was killed.

Package entity.ore

Class Coal extends Ore

Field

- double hp	Durability of coal ore.
- double armor	Armor of coal ore

Constructor

+ Coal(int row, int col)	Generate coal ore to (row, col) location.
--------------------------	---

Method

+ String getIcon()	Return coal ore's icon.
+ String getItem()	Return drop item when coal ore was destroyed.
Getter & Setter	Getter & Setter for each field.

Class Diamond extends Ore

Field

- double hp	Durability of diamond ore.
- double armor	Armor of diamond ore

Constructor

+ Diamond(int row, int col)	Generate diamond ore to (row, col) location.
-----------------------------	--

Method

+ String getIcon()	Return diamond ore's icon.
+ String getItem()	Return drop item when diamond ore was destroyed.
Getter & Setter	Getter & Setter for each field.

Class Iron extends Ore

Field

- double hp	Durability of iron ore.
- double armor	Armor of iron ore

Constructor

+ Iron(int row, int col)	Generate iron ore to (row, col) location.
--------------------------	---

Method

+ String getIcon()	Return iron ore's icon.
+ String getItem()	Return drop item when iron ore was destroyed.
Getter & Setter	Getter & Setter for each field.

Class Ore **extends** Entity **implements** HasHP, HasArmor, CanTakePhysicalDamage, CanTakeBombDamage, Pickable

Constructor

+ Ore(int row,int col)	Generate ore to (row, col) location.
------------------------	--------------------------------------

Method

+ boolean isAlive()	Check ore's durability more than zero or not.
+ double takePhysicalDamage(double damage)	Ore take effect form physical damage.
+ double takeBombDamage(double damage)	Ore take effect form bomb damage.
+ abstract String getItem()	Return drop item when each ore was destroyed.
+ void pick(Entity target)	Pick up drop item to destroyer.

Interface Pickable

Method

+ void pick(Entity by)	Entity <by> pick this entity
------------------------	------------------------------

Package gui

Class Block **extends** StackPane

Field

+ <u>int WIDTH</u>	Block width
+ <u>int HEIGHT</u>	Block height
- Entity entity	Block's entity
+ Label testLabel	Label used for testing
- int row	Block row
- int col	Block column
- boolean hasentity	This block has entity linked to or not

Constructor

+ Block(int row,int col)	Initialize new block at position row and column
--------------------------	---

Method

+ Entity getEntity()	Get entity of this block
+ void setEntity(Entity entity)	Set entity of this block
+ void removeEntity()	Remove entity of this block
+ boolean hasEntity()	Check whether this block has entity or not
+ int getRow()	Get row of this block
+ int getCol()	Get column of this block
+ <u>Block getBlock(int row,int col)</u>	Get block placed at position row and column
+ void addEffect(Effect e)	Add effect e to this block
+ void removeEffect(Effect e)	Remove effect e from this block

Class BlockView **extends** StackPane

Field

- Block block	Block attached to this block view
- int realRow	Real row of this block view (row of block)
- int realCol	Real column of this block view (column of block)

Constructor

+ BlockView()	Initialize new block view
+ BlockView(int realrow,int realcol)	Initialize new block view with realRow = realrow and realCol = realcol

Method

- void render()	Render this block view
+ Block getBlock()	Get block attached to this block view
+ void setBlock(Block block)	Set block attached to this block view and render again
+ void setBlock(Block block, int realrow,int realcol)	Set block attached to this block view and render again and set realRow = realrow and realCol = realcol
+ <u>BlockView</u> getBlockView(int row,int col)	Get block view at position row and column relative to viewport
Getter & Setter	

Class ControlPanel **extends** VBox

Field

+ <u>int BOX_SIZE</u>	Size of box used in control panel children.
- CrafterBox crafterBox	CrafterBox component in this control panel
- InventoryBox inventoryBox	InventoryBox component in this control panel
- CrafterPerformer craftPerformer	CrafterPerformer component in this control panel
- EquipmentViewer equipmentViewer	EquipmentViewer component in this control panel
- LevelPanel levelPanel	LevelPanel component in this control panel

Constructor

+ ControlPanel()	Initialize new control panel component and then initialize child components and perform update to all child components
------------------	--

Method

+ <u>synchronized void wrongSound()</u>	Play wrong sound
+ void update()	Update all child components

Class CrafterBox **extends** HBox

Field

- Crafter crafter	Crafter linked to this crafter box
- Button coalPic	Coal button
- Label coalAmount	Coal amount label
- Button ironPic	Iron button
- Label ironAmount	Iron amount label
- Button diamondPic	Diamond button
- Label diamondAmount	Diamond amount label

Constructor

+ CrafterBox(Crafter crafter)	Initialize new crafter box with crafter and initialize boxes and then perform update
-------------------------------	--

Method

- VBox genBox(Button pic, Label amount)	Build one box containing item icon and amount
+ void update()	Update inventory box
+ Crafter getCrafter()	Get crafter linked to this crafter box

Class CrafterPerformer **extends** Button

Constructor

+ CrafterPerformer()	Initialize new crafter performer button and perform update
----------------------	--

Method

+ String getIcon()	Get icon of item to craft
+ void update()	Update item to craft

Class EquipmentViewer **extends** GridPane

Field

- StackPane armor	Component containing armor icon
- StackPane helmet	Component containing helmet icon
- StackPane sword	Component containing sword icon
- StackPane pant	Component containing pant icon
- StackPane boot	Component containing boot icon
- String borderCSS	CSS script string for adding border

Constructor

+ EquipmentViewer()	Initialize new equipment viewer component and perform update
---------------------	--

Method

- String getArmorIcon()	Get armor icon
- String getHelmetIcon()	Get helmet icon
- String getSwordIcon()	Get sword icon
- String getPantIcon()	Get pant icon
- String getBootIcon()	Get boot icon
+ void update()	Update equipment viewer

Class GameArea **extends** ScrollPane

Field

- GameAreaInner inner	Game area inner component
+ <u>double SCROLLBAR_WIDTH</u>	Scrollbar width
+ <u>double SCROLLBAR_HEIGHT</u>	Scrollbar height

Constructor

+ GameArea()	Initialize game area component and add game area inner to this game area component
--------------	--

Method

+ GameAreaInner getInner()	Get game area inner component
+ double getWidthReal()	Get width without scrollbar
+ double getHeightReal()	Get height without scrollbar

Class GameAreaInner **extends** GridPane

Field

+ <u>int NUM_COL</u>	Number of column in game
+ <u>int NUM_ROW</u>	Number of row in game
+ <u>int VIEW_COL</u>	Number of column in viewport
+ <u>int VIEW_ROW</u>	Number of row in viewport
- HashMap<Integer, HashMap<Integer, Block>> blocks	Blocks data
- HashMap<Integer, HashMap<Integer, BlockView>> block_views	Block views data
- boolean[][] isDiamondOre	isDiamondOre[i][j] = is row i column j has diamond
- boolean[][] isIronOre	isIronOre [i][j] = is row i column j has iron
- boolean[][] isCoalOre	isCoalOre [i][j] = is row i column j has coal
- int dirI[]	Constant = { -1, -1, -1, 0, 0, 1, 1, 1 } used for reducing repetitive code
- int dirJ[]	Constant = { -1, 0, 1, -1, 1, -1, 0, 1 } used for reducing repetitive code

Constructor

+ GameAreaInner()	Initialize game area inner component and generate blocks and blocks view and place ore
-------------------	--

Method

+ HashMap<Integer, HashMap<Integer, Block>> getBlocks()	Get blocks data
+ HashMap<Integer, HashMap<Integer, BlockView>> getBlockViews()	Get block views data

Class ImageStore

Field

- <u>ImageStore store</u>	ImageStore instance
+ String blockBG	Loading background image.
+ String coalBlock	Loading coal ore image.
+ String ironBlock	Loading iron ore image.
+ String diamondBlock	Loading diamond ore image.
+ String boomerU	Loading boomer facing on up direction image.
+ String boomerR	Loading boomer facing on right direction image.
+ String boomerD	Loading boomer facing on down direction image.
+ String boomerL	Loading boomer facing on left direction image.
+ String zombieU	Loading zombie facing on up direction image.
+ String zombieR	Loading zombie facing on right direction image.
+ String zombieD	Loading zombie facing on down direction image.
+ String zombieL	Loading zombie facing on left direction image.
+ String bomb	Loading bomb image.
+ String fire	Loading fire image.
+ String attack	Loading attacking image.
+ String iron	Loading iron image.
+ String coal	Loading coal image.
+ String diamond	Loading diamond image.
+ String woodenSword	Loading wooden sword (starter item).
+ String ironSword	Loading iron sword.
+ String diamondSword	Loading diamond sword.
+ String ironPant	Loading iron pant.
+ String diamondPant	Loading diamond pant.
+ String ironBoot	Loading iron boots.
+ String diamondBoot	Loading diamond boots.
+ String ironHelmet	Loading iron helmet.
+ String diamondHelmet	Loading diamond helmet.
+ String ironArmor	Loading iron armor.
+ String diamondArmor	Loading diamond armor.

Constructor

+ ImageStore()	Initialize image store
----------------	------------------------

Method

+ <u>ImageStore getInstance()</u>	Get image store instance
-----------------------------------	--------------------------

Class InventoryBox **extends** HBox **implements** HasInventory

Field

- Inventory inventory	Inventory linked to this component
- Button coalPic	Coal button
- Label coalAmount	Coal amount
- Button ironPic	Iron button
- Label ironAmount	Iron amount
- Button diamondPic	Diamond button
- Label diamondAmount	Diamond amount

Constructor

+ InventoryBox(Inventory inventory)	Construct new inventory box component with inventory and initialize child boxes and perform update
-------------------------------------	--

Method

- VBox genBox(Button pic, Label amount)	Build one box containing item icon and amount
+ void update()	Update inventory box
+ Inventory getInventory()	Get inventory linked to this component

Class LevelPanel **extends** VBox

Field

- Label levelLabel	Level label
- Label expLabel	EXP label
- ProgressBar expBar	EXP progress bar
- Label hpLabel	HP Label
- ProgressBar hpBar	HP progress bar

Constructor

+ LevelPanel()	Initialize level panel
----------------	------------------------

Method

+ void update()	Update level panel
-----------------	--------------------

Package item

Class Crafter implements HasInventory

Field

- Inventory inventory	Inventory that linked to this crafter
- HashMap<String,Integer> items	Items in this crafter data

Constructor

+ Crafter(Inventory inv)	Initialize crafter with inventory inv
--------------------------	---------------------------------------

Method

+ Inventory getInventory()	Get inventory that linked to this crafter
+ void add(String name,Integer amount)	Add item <name> with amount <amount> to crafter
+ void add(String name)	Add 1 item <name> to crafter
- void use(String name,Integer amount)	Use item <name> with amount <amount> in crafter
- void use(String name)	Use 1 item <name> in crafter
+ void giveBack(String name,Integer amount)	Give <amount> item <name> back to inventory
+ void giveBack(String name)	Give 1 item <name> back to inventory
+ Integer get(String name)	Get amount of item <name> in crafter
+ boolean has(String name,Integer amount)	Does item <name> in crafter has exactly equal to <amount>
+ String getCraftTarget()	Get what we can craft with given items in crafter
+ void performCraft()	Perform crafting

Class Inventory

Field

- HashMap<String,Integer> items	Items data
- int swordLevel	Sword level
- int armorLevel	Armor level
- int helmetLevel	Helmet level
- int bootLevel	Boot level
- int pantLevel	Pant level

Method

+ void add(String name,Integer amount)	Add item <name> with amount <amount>
+ void add(String name)	Add 1 item <name>
+ void use(String name,Integer amount)	Use item <name> with amount <amount>
+ void use(String name)	Use 1 item <name>
+ boolean has(String name,Integer amount)	Does item <name> has more than <amount>
+ Integer get(String name)	Get amount of item <name>
Getter & Setter	

Class NotEnoughItemException **extends** Exception

Field

- String itemname	Item name
-------------------	-----------

Constructor

+ NotEnoughItemException()	Initialize exception
+ NotEnoughItemException(String itemname)	Initialize exception with item name

Method

+ String getItemname()	Get item name
------------------------	---------------

Package util

Class ExceptionWithAction **extends** Exception

Field

# String message	Exception's message
# Runnable action	Exception's action

Constructor

+ ExceptionWithAction()	Initialize exception
+ ExceptionWithAction(String msg)	Initialize exception with message
+ ExceptionWithAction(String msg, Runnable action)	Initialize exception with message, action
+ ExceptionWithAction(Runnable action)	Initialize exception with action

Method

+ boolean canPerformAction()	Check whether can perform action or not
+ void performAction()	Perform action
Getter & Setter	