

# The escapists

Documentation

Created by

Nuttapol Onton 6330166021

Natchapon Chitturongaporn 6330148721

2110215 Programming Methodology

Semester 1 Year 2021

Chulalongkorn University





# The escapists








## Introduction

The escapists is inspired from series Prison Break. The story is about a prisoner who sentenced to receive the death penalty, and the execution will be happened in next two weeks. So that is his chance to escape from the prison.

## Character

1.  "Scofield" He never trust anyone, he can craft a ladder.
2.  "Beros" He has the knife.
3.  "Mahone" He has key to enter maintenance room.
4.  "Bratt" He is the officer who keeping an eye on you every night.

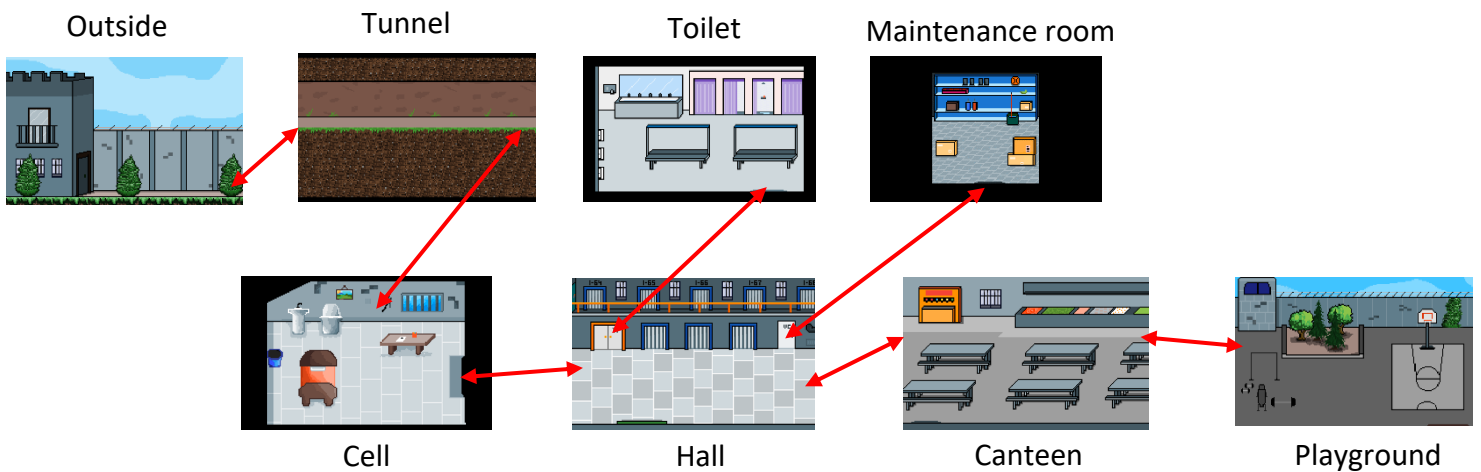
## Item

1.  "Food" eats to receive 50 stamina.
2.  "Key" use to enter maintenance room.
3.  "Knife" use to intimidate Scofield to craft ladder.
4.  "Ladder" use to climb out from cell.
5.  "Shovel" use to mine the cell wall.

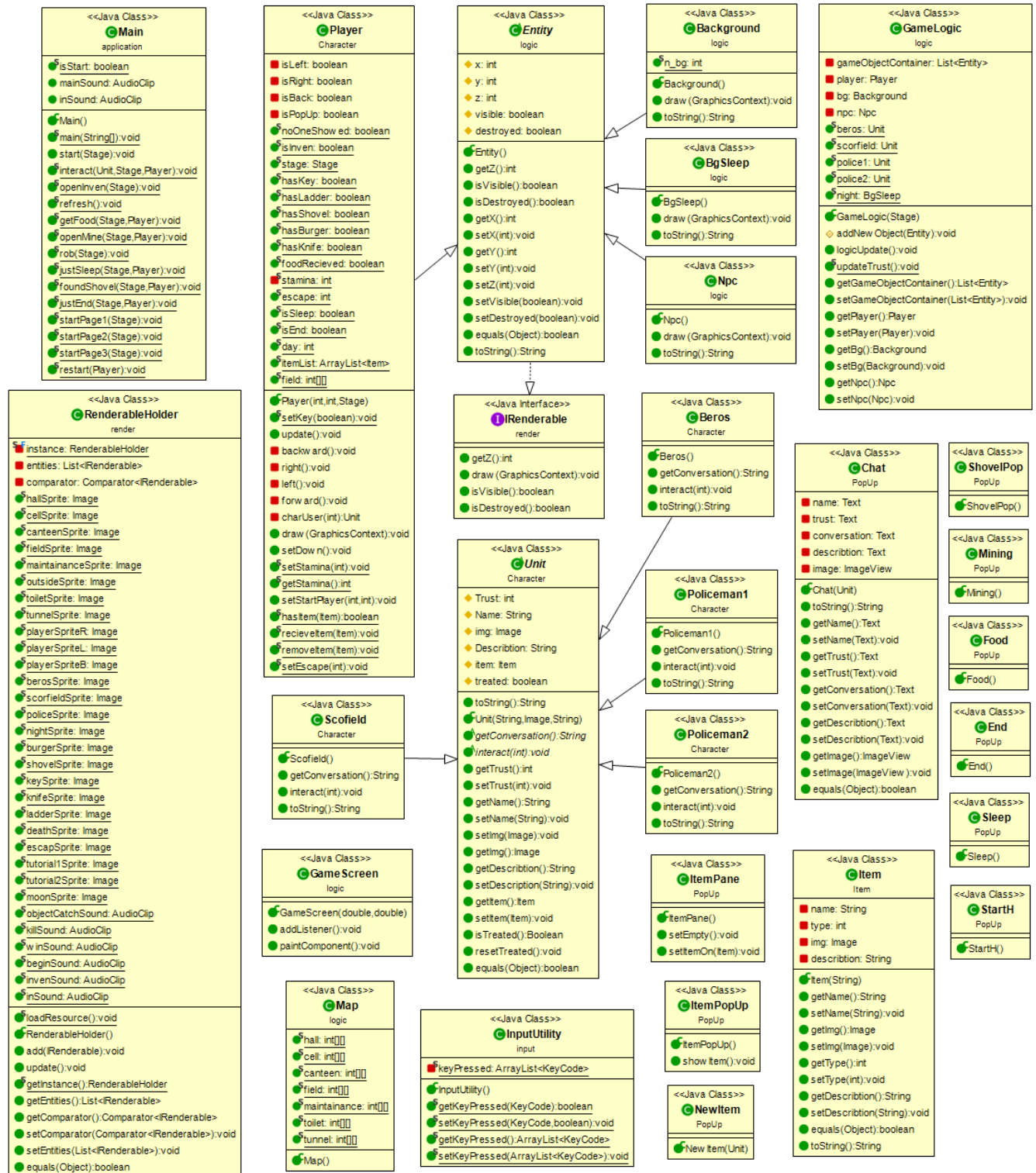
## Rules

1. At the begin of the day, you will have 50 stamina.
2. You can receive food at the canteen only one time per day.
3. If you can make someone trust in you, then you will get item from him.
4. You need to sleep to reset stamina and get new food.
5. You have only 15 days to escape.

## Map



## Class diagram



\*Noted that Access Modifier Notations can be listed below.

+ (public)

# (protected)

- (private)

static will be underlined.

*abstract* will be italic.

## 1. Package application

### 1.1 class Main extends Application

#### 1.1.1 Fields

+ <u>boolean isStart</u>	Game start status.
+ AudioClip mainSound	Sound before the game begin.
+ AudioClip inSound	Sound when the game begin

#### 1.1.2 Methods

+ void main(String[] args)	Application. <i>launch</i> (args);
+ void start(Stage primaryStage) throws Exception	- prepare stage and scene to be ready for launch. - Create animation thread.
+ <u>public void interact(Unit unit, Stage stage, Player player)</u>	- Create Popup. - Popup will change according to unit that Player interact with and Trust. If trust more than or equal to 80, Player will receive new item. But if trust less than 80 Popup will be chat popup. - create OK Button to close the popup window.
+ <u>void openInven(Stage stage)</u>	- Create ItemPopUp to be a popup window. - create OK Button to close the popup window.

	<ul style="list-style-type: none"> <li>- If Player has burger, create empty pane to cover the burger item and set on action to eat and reopen inventory.</li> </ul>
+ <u>void getFood(Stage stage, Player player)</u>	<ul style="list-style-type: none"> <li>- Create Food class to be a Popup</li> <li>- Create OK Button to close the popup window.</li> <li>- update Player.foodReviewed to be true.</li> <li>- update item status in Player.</li> </ul>
+ <u>openMine(Stage stage, Player player)</u>	<ul style="list-style-type: none"> <li>- Create Mining class to be a Popup</li> <li>- Create OK Button to close the popup window.</li> </ul>
+ <u>void rob(Stage stage)</u>	<ul style="list-style-type: none"> <li>- Create NewItem class to be a Popup</li> <li>- Create OK Button to close the popup window.</li> <li>- update ladder status in player.</li> </ul>
+ <u>justSleep(Stage stage, Player player)</u>	<ul style="list-style-type: none"> <li>- Create Sleep class to be a Popup</li> <li>- Create OK Button to back to the cell.</li> <li>- update days left.</li> </ul>
+ <u>void foundShovel(Stage stage, Player player)</u>	<ul style="list-style-type: none"> <li>- Create ShovelPop class to be a Popup</li> <li>- Create OK Button to close the popup window.</li> <li>- update shovel status in player.</li> </ul>
+ <u>void justEnd(Stage stage, Player player)</u>	<ul style="list-style-type: none"> <li>- Create End class to be a Popup.</li> <li>- Create restart Button to restart game.</li> <li>- Create end button to close the game.</li> </ul>
+ <u>void startPage1(Stage stage)</u>	<ul style="list-style-type: none"> <li>- Create startH class to be a Popup (first scene when open the game).</li> <li>- Create start Button to start to call startPage2.</li> </ul>

+ <u>void startPage2(Stage stage)</u>	- Create startH class to be a Popup (tutorial of the game). - Create next Button to start to call startPage3.
+ <u>void startPage2(Stage stage)</u>	- Create startH class to be a Popup (Hints and rules of the game). - Create next Button to start to start the game.
+ <u>void restart(Player player)</u>	- Reset all game variable to default like first time open the game.

## 2. Package Character

### 2.1 abstract Unit

#### 2.1.1 Fields

# int Trust	Trust from each unit.
# String Name	Name.
# Image img	Image of the character.
# String Description	Description for the character.
# Item item	Item that characters has.
# Boolean treated	On this day Player treated this unit or not.

#### 2.1.2 Constructor

+ Unit(String Name, Image img, String des)	- Initialize Fields. - Trust = 50. - treated = false.
--	---

#### 2.1.3 Methods

+ String toString()	- return "Unit [Trust=" + Trust + ", Name=" + Name + ", Description=" + Description + "]".
+ <i>String getConversation()</i>	- Details not here.

+ void interact(int point)	- Details not here.
+ int getTrust	- Return trust.
+ void setTrust(int trust)	- Make sure that trust < 100 and trust > 0.
+ String getName()	- Return name.
+ void setName(String name)	- Set name.
+ image getImg()	- Return img.
+ void setImg(Image img)	- Set img.
+ String getDescription()	- Return description.
+ void setDescription(String description)	- Set description.
+ Item getItem()	- Return item.
+ void setItem(Item item)	- Set item.
+ Boolean isTreated()	- Return treated.
+ void resetTreated()	- Treated = false.
+ boolean equals(Object obj)	- Check two Unit object equals or not.

## 2.2 class Beros extends Unit

### 2.2.1 Constructors

+ Beros()	- Call super class constructor. - This Character has knife Item.
-----------	---

### 2.2.2 Methods

+ String getConversation()	- Conversation will change according to trust.
+ Public interact(int point)	- Set treated = true.
+ String toString()	- Return "Beros [Trust=" + Trust + ", Name=" + Name + ", Description=" + Description + "]".

## 2.3 class Scofield extends Unit

### 2.3.1 Constructors

+ Scofield()	- Call super class constructor. - This Character has ladder Item.
--------------	--



### 2.3.2 Methods

+ String getConversation()	- Conversation will change according to trust.
+ Public interact(int point)	- Set treated = true.
+ String toString()	- return "Scofield [Trust=" + Trust + ", Name=" + Name + ", Description=" + Description + "]."

## 2.4 class Policeman1 extends Unit

### 2.4.1 Constructors

+ Policeman1()	- Call super class constructor. - This Character has moon Item.
----------------	--

### 2.4.2 Methods

+ String getConversation()	- Conversation will change according to trust.
+ Public interact(int point)	- Set treated = true.
+ String toString()	- return "Policeman1 [Trust=" + Trust + ", Name=" + Name + ", Description=" + Description + "]."

## 2.5 class Policeman2 extends Unit

### 2.5.1 Constructors

+ Policeman2()	- Call super class constructor. - This Character has knife Item.
----------------	---

### 2.5.2 Methods

+ String getConversation()	- Conversation will change according to trust.
+ Public interact(int point)	- Set treated = true.
+ String toString()	- return "Policeman2 [Trust=" + Trust + ", Name=" + Name + ", Description=" + Description + "]."

## 2.6 class Player extends Entity

This class contains almost variables of player status.

### 2.6.1 Fields

- boolean isLeft	Current facing left or not.
- boolean isRight	Current facing right or not.
- boolean isBack	Current facing back or not.
- boolean isPopUp	Is there any popup window.
+ <u>boolean noOneShowed</u>	Is "No one here tonight" popup still showed.
+ <u>boolean hasKey</u>	Player has key.
+ <u>boolean hasLadder</u>	Player has ladder.
+ <u>boolean hasShovel</u>	Player has shovel.
+ <u>boolean hasBurger</u>	Player has burger.
+ <u>boolean hasKnife</u>	Player has knife.
+ <u>boolean foodReceived</u>	Player already get the food for today.
+ <u>int stamina</u>	Player stamina.
+ <u>int escape</u>	Percentage of escape complete.
+ <u>boolean isSleep</u>	Sleep.
+ <u>boolean isEnd</u>	Is the game end.
+ <u>int day</u>	The day left.
+ <u>ArrayList&lt;Item&gt; itemList</u>	Player item.
+ <u>Stage stage</u>	Stage that showed in the current time.
+ <u>int[][] field</u>	Map plot.

### 2.6.2 Constructor

+ Player(int x, int y, Stage stage)	- Initial Fields.
-------------------------------------	-------------------

### 2.6.3 Methods

+ <u>void setKey(boolean b)</u>	- Set hasKey.
+ <u>update()</u>	- Listener function. Get input from keyboard and update Player status.

+ void backward()	- If Player in backward position and need to interact with some object or change map.
+ void right()	- If Player in right facing position and need to interact with some object or change map.
+ void left()	- If Player in left facing position and need to interact with some object or change map.
+ void forward()	- If Player in forward position and need to interact with some object or change map.
+ void draw(GraphicsContext gc)	- Change Player Facing when change director.
+ void setStamina(int s)	- Set stamina, make sure $s < 100$ and $s > 0$ .
+ int get stamina()	- Return stamina.
+ int setStartPlayer	- Set Player to be ready for new game.
+ <u>hasItem(Item item)</u>	- Return Player has that item or not.
+ <u>void recieveItem(Item item)</u>	- Add item to itemList and set hasItem.
+ <u>removeItem(Item item)</u>	- Remove item from itemList and set hasItem.
+ <u>void setEscape(int point)</u>	- Make sure percentage of escape complete will not more than 100.

### 3. Package input

#### 3.1 Class InputUtility

This class contains functions that store input value in ArrayList and send input value to another class.

##### 3.1.1 Fields

- <u>ArrayList&lt;KeyCode&gt; keyPressed</u>	List of input.
--	----------------

##### 3.2.2 Methods

+ <u>boolean getKeyPressed(KeyCode keycode)</u>	- Return true if keycode is in keyPressed. otherwise, return false.
+ <u>boolean setKeyPressed(KeyCode keycode, boolean, pressed)</u>	pressed is true.

	<ul style="list-style-type: none"> <li>- If there is no keycode in pressed, add keycode in pressed.</li> <li>- If there is keycode in pressed , do nothing.</li> </ul> <p>Pressed is false.</p> <ul style="list-style-type: none"> <li>- Remove keycode in pressed.</li> </ul>
--	--

## 4. Package Item

### 4.1 Class Item

This class is represent Item, which info is stored for using in another class.

#### 4.1.1 Fields

- String name	Name of Item.
- int type	Type of Item.
- Image img	Image of Item.
- String description	Description of Item.

#### 4.1.2 Constructor

+ Item(String name)	- Initialize name, img, type, description.
---------------------	--

#### 4.1.3 Methods

+ getName() + setName(String name)	- Getter/Setter for name.
+ getImg() + setImg(Image img)	- Getter/Setter for img.
+ getType() + setType(int type)	- Getter/Setter for type.
+ getDescription() + setDescription(String description)	- Getter/Setter for description.
+ boolean equals(Object obj)	- Default generated.

+ String toString()	- Return "Item [ name =" + name + " , type=" + type + " , description=" + description + "]" .
---------------------	---

## 5. Package Logic

This package contains the class that is drawn on Main and map plot.

### 5.1 Abstract Class Entity

Implements: IRenderable

This class is the base class for all Entity in the map.

#### 5.1.1 Fields

# int x	Value of position in X-axis.
# int y	Value of position in Y-axis.
# int z	Value of position in Z-axis.
# boolean visible	A boolean contains this Entity that visible or invisible.
# boolean destroyed	A boolean contains this Entity that destroyed or not.

#### 5.1.2 Constructor

+ Entity()	- Initialize the protected fields.
------------	------------------------------------

#### 5.1.3 Methods

+ int getZ()	- Return the value of position in Z-axis.
+ boolean isVisible()	- Return visible.
+ boolean isDestroyed()	- Return destroyed.
+ boolean equals(Object obj)	- Default generated.
+ String toString()	- Return "Entity [ x =" + x + " , y=" + y + " , z=" + z + "]" .
+ int getX()	- Getter/Setter for x.

+ void setX()	
+ int getY() + void setY()	- Getter/Setter for Y.
+ void setZ()	- Setter for Z.
+ void setVisible()	- Setter for visible.
+ void setDestroyed()	- Setter for destroyed.

## 5.2 Class Background Extends Entity

This class represent Background type Entity, which can be changed by player interact.

### 5.2.1 Fields

+ <u>int n_bg</u>	Serial number that start with 1.
-------------------	----------------------------------

### 5.2.2 Constructor

+ Background()	- Call super class constructure. - Initialize z with 20.
----------------	---

### 5.2.3 Methods

+ void draw(GraphicsContext gc)	<p>If isStart in Main is false.</p> <ul style="list-style-type: none"> <li>- gc will draw image that is beginning image in position (0, 0).</li> </ul> <p>If isStart in Main is true.</p> <ul style="list-style-type: none"> <li>- gc will draw image in position (0, 0).</li> </ul> <p><u>Note</u> : Each image is not the same if n_bg is not equal.</p>
+ String toString()	- Return "Background [ x =" + x + ", y=" + y + ", z=" + z + "]".

## 5.3 Class BgSleep Extends Entity

This class represent BgSleep type Entity, which can be changed by player interact.

### 5.3.1 Constructor

+ BgSleep()	- Call super class constructure. - Initialize z with 0.
-------------	--

### 5.3.2 Methods

+ void draw(GraphicsContext gc)	- gc will draw ending image in position (0, 0).
+ String toString()	- Return "BgSleep [ x =" + x + ", y=" + y + ", z=" + z + "]" .

## 5.4 Class NPC Extends Entity

This class represent NPC type Entity, which can be changed when background changed.

### 5.4.1 Constructor

+ NPC ()	- Call super class constructure. - Initialize z with 15.
----------	---

### 5.4.2 Methods

+ void draw(GraphicsContext gc)	- gc will draw NPC image in any position by n_bg.
+ String toString()	- Return "NPC [ x =" + x + ", y=" + y + ", z=" + z + "]" .

## 5.5 Class Map

This class is map plot in array 2 dimension.

### 5.5.1 Fields

+ <u>int[][] hall</u>	Map plot for hall.
+ <u>int[][] cell</u>	Map plot for cell.
+ <u>int[][] canteen</u>	Map plot for canteen.
+ <u>int[][] field</u>	Map plot for field.
+ <u>int[][] maintainance</u>	Map plot for maintenance.
+ <u>int[][] toilet</u>	Map plot for toilet.
+ <u>int[][] tunnel</u>	Map plot for tunnel.

## 5.6 Class GameScreen extends Canvas

This class represent GameScreen type Canvas, which can store input and draw a component.

### 5.6.1 Constructor

+ GameScreen(double width, double height)	Initialize by using function: <ul style="list-style-type: none"><li>- call super class constructor with width and height.</li><li>- setVisible(true).</li><li>- addListener().</li></ul>
---	--

### 5.6.2 Methods

+ void addListener()	- Use setOnKeyPressed and setOnKeyReleased for receiving input from keyboard when you action with keyboard if you pressed setKeyPressed(Keycode , true) and if you released setKeyReleased(Keycode, false).
+ void paintComponent()	- Use GraphicsContext for Drawing every IRenderable in entities(List in RenderableHolder class) if this IRenderable is visible and not destroyed and isStart in Main is false but if IRenderable instanceof Background ,then this IRenderable draw.

## 5.7 Class GameLogic

This class is center of object that can update and draw GraphicsContext.

### 5.7.1 Fields

- List<Entity> gameObjectContainer	List of Entity.
- Player player	Player in game.



- Background bg	Background in game.
- Npc npc	Npc in game.
+ <u>Unit_beros</u>	Unit in game.
+ <u>Unit_scorfield</u>	Unit in game.
+ <u>Unit_police1</u>	Unit in game.
+ <u>Unit_police2</u>	Unit in game.
+ <u>BgSleep_night</u>	BgSleep in game.

### 5.7.2 Constructor

+ GameLogic(Stage stage)	- Initialize gameObjectContainer, player, police1, police2, beros, scorfield, player, bg, npc, night. - Use addNewObject for player, bg, npc, night.
--------------------------	---

### 5.7.3 Methods

# addNewObject(Entity entity)	- Add Entity in gameObjectContainer and entities in RenderableHolder.
+ void logicUpdate()	- Update player.
+ <u>void updateTrust()</u>	- Update Trust in beros, police1, police2.
+ List<Entity> getGameObjectContainer() + void setGameObjectContainer()	- Getter/Setter for gameObjectContainer.
+ Player getPlayer() + void setPlayer(Player player)	- Getter/Setter for player.
+ Background getBg() + void setBg(Background bg)	- Getter/Setter for bg.
+ Npc getNpc() + void setNpc(Npc npc)	- Getter/Setter for npc.

## 6. Package PopUp

This Package contains all pane that will become to be a Popup window in the future.

### 6.1 class Chat extends GridPane

This class is a Pane for Popup window when Player interacts with any Unit.

#### 6.1.1 Fields

- Text name	Unit.getName().
- Text trust	Unit.getTrust().
- Text conversation	unit.getconversation().
- Text describtion	Unit.getdescribtion().
- ImageView image	Unit.getImg().

#### 6.1.2 Constructor

+ Chat(Unit unit)	<ul style="list-style-type: none"><li>- Initialize fields</li><li>- Create inter button that can push to interact with Unit. The handle will change according to unit that player interacted with.</li><li>- Use add all node to GridPane.</li></ul>
-------------------	--

#### 6.1.3 Methods

+ Getters/Setters for all fields	- No conditions.
+ String toString()	- Return "Chat [name=" + name + ", trust=" + trust + ", conversation=" + conversation + ", description=" + description + "];"
+ boolean equals()	- Default generated.

### 6.2 class End extends GridPane

This class is a pane for Popup window when the game end

#### 6.2.1 Constructor

+ End()	- Create Text that will show win or lose.
---------	---

### 6.3 class Food extends GridPane

This class is a pane for Popup window when Player receives the food.

#### 6.3.1 Constructor

+ Food()	<ul style="list-style-type: none"><li>- Create Text to show Player receiving the food.</li><li>- Create imageView of Burger.</li><li>- Create Text to show description.</li><li>- add all node to GridPane.</li></ul>
----------	---

### 6.4 class ItemPane extends Pane

This Class is a part of ItemPopUp to show Player inventory.

#### 6.4.1 Constructor

+ ItemPane()	- Create Pane and call setEmpty().
--------------	------------------------------------

#### 6.4.2 Methods

+ void setEmpty()	- Set pane back ground to BISQUE color.
+ void setItemOn(Item item)	- Set add item image on pane.

### 6.5 class ItemPopUp extends GridPane

This class id a pane for PopUp window when Player open inventory.

#### 6.5.1 Constructor

+ ItemPopUp()	<ul style="list-style-type: none"><li>- Set GridPane properties.</li><li>- createText to shoe Player stamina.</li></ul>
---------------	---

#### 6.5.2 Method

+ void showItem()	- Create itemPane for all item in Player.ItemList.
-------------------	--

### 6.6 Class Mining extends GridPane

This class is a pane for Popup window when Player mining the cell wall.

#### 6.6.1 Constructor

+ Mining()	<ul style="list-style-type: none"><li>- Create Text to show Mining status.</li><li>- Create Text to show mining condition.</li><li>- Create Button to start mining.</li></ul>
------------	---

### 6.7 class NewItem extends GridPane

This class is a pane for Popup window when Player found new item.

#### 6.7.1 Constructor

+ NewItem(Unit unit)	<ul style="list-style-type: none"><li>- Create Text to show item name.</li><li>- Create ImageView to show item image.</li><li>- Create Text to show item description.</li><li>- Set GridPane properties.</li><li>- Add all node to GridPane.</li></ul>
----------------------	--

### 6.8 class ShovelPop extends GridPane

This class is a pane for Popup window when Player found a shovel.

#### 6.8.1 Constructor

+ ShovelPop()	<ul style="list-style-type: none"><li>- Create Text to show item name.</li><li>- Create ImageView to show shovel image.</li><li>- Create Text to show shovel description.</li><li>- Set GridPane properties.</li><li>- Add all node to GridPane.</li></ul>
---------------	--

### 6.9 class Sleep extends GridPane

This class is a pane for Popup window when Player sleep.

#### 6.9.1 Constructors

+ Sleep()	<ul style="list-style-type: none"><li>- Create Text to show the days left.</li><li>- set GridPane properties.</li></ul>
-----------	---

## 6.10 class StartH() extends GridPane

This class is for first time open game.

### 6.10.1 Constructor

+ StartH()	Set GridPane properties.
------------	--------------------------

## 7. Package render

### 7.1 interface IRenderable

This interface contains base function for any class that

#### 7.1.1 Methods

+ int getZ()	- Details not here.
+ void draw(GraphicsContext gc)	- Details not here.
+ boolean isVisible()	- Details not here.
+ boolean isDestroyed()	- Details not here.

### 7.2 class RenderableHolder

This class

#### 7.2.1 Fields

- <u>final RenderableHolder instance</u>	Initialize instance.
- List<IRenderable> entities	List of entities.
- Comparator<IRenderable> comparator	Comparator for compare IRenderable.
+ <u>Image hallSprite</u>	Image for hall background.
+ <u>Image cellSprite</u>	Image for cell background.
+ <u>Image canteenSprite</u>	Image for canteen background.
+ <u>Image fieldSprite</u>	Image for field background.
+ <u>Image maintainanceSprite</u>	Image for maintenance background.
+ <u>Image outsideSprite</u>	Image for outside background.
+ <u>Image toiletSprite</u>	Image for toilet background.
+ <u>Image tunnelSprite</u>	Image for tunnel background.
+ <u>Image playerSpriteR</u>	Image for player right-side.
+ <u>Image playerSpriteL</u>	Image for player left-side.

+ <u>Image playerSpriteB</u>	Image for player back-side.
+ <u>Image berosSprite</u>	Image for Beros.
+ <u>Image scorfieldSprite</u>	Image for Scorfield.
+ <u>Image policeSprite</u>	Image for police.
+ <u>Image nightSprite</u>	Image for night background.
+ <u>Image burgerSprite</u>	Image for burger.
+ <u>Image shovelSprite</u>	Image for shovel.
+ <u>Image keySprite</u>	Image for key.
+ <u>Image knifeSprite</u>	Image for knife.
+ <u>Image ladderSprite</u>	Image for ladder.
+ <u>Image deathSprite</u>	Image for death background.
+ <u>Image escapSprite</u>	Image for escapist background.
+ <u>Image tutorial1Sprite</u>	Image for tutorial background.
+ <u>Image tutorial2Sprite</u>	Image for tutorial background.
+ <u>Image moonSprite</u>	Image for moon.
+ <u>AudioClip objectCatchSound</u>	Sound when catch object.
+ <u>AudioClip killSound</u>	Sound when you are killed.
+ <u>AudioClip winSound</u>	Sound when you win.
+ <u>AudioClip beginSound</u>	Sound before game begins.
+ <u>AudioClip invenSound</u>	Sound when click button or open inventory.
+ <u>AudioClip inSound</u>	Sound when game begins.

### 7.2.2 Constructor

+ <u>RenderableHolder()</u>	- Initialize entities and comparator.
-----------------------------	---------------------------------------

### 7.2.3 Methods

+ void <u>add</u> (IRenderable entity)	- add entity in entities. - sort entities with comparator.
+ void <u>update</u> ()	- remove Entity when Entity is destroyed.
+ <u>RenderableHolder getInstance</u> ()	- return instance.

+ List<IRenderable> getEntities() + void setEntities(List<IRenderable> entities)	- Getter/Setter for entities.
+ Comparator<IRenderable> getComparator() + void setComparator(Comparator <IRenderable> comparator)	- Getter/Setter for comparator.
+ boolean equals(Object obj)	- Generated default.