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

Outside Tunnel Toilet Maintenance room

Cell Hall Canteen Playground

Class diagram

Main
`<<Java Class>>`
Main
application

- isStart: boolean
- mainSound: AudioClip
- inSound: AudioClip
- winSound: AudioClip

- Main()
- main(String[]):void
- start(Stage):void
- interact(Unit,Stage,Player):void
- openInven(Stage):void
- refresh():void
- getFood(Stage,Player):void
- openMine(Stage,Player):void
- rob(Stage):void
- justSleep(Stage,Player):void
- foundShovel(Stage,Player):void
- justEnd(Stage,Player):void
- startPage1(Stage):void
- startPage2(Stage):void
- startPage3(Stage):void
- restart(Player):void

Player
`<<Java Class>>`
Player
Character

- isLeft: boolean
- isRight: boolean
- isBack: boolean
- isPopUp: boolean
- noOneShowed: boolean
- isInven: boolean
- stage: Stage
- hasKey: boolean
- hasLadder: boolean
- hasShovel: boolean
- hasBurger: boolean
- hasKnife: boolean
- foodRecieved: boolean
- satamina: int
- escape: int
- isSleep: boolean
- isEnd: boolean
- day: int
- itemList: ArrayList<Item>
- field: int[][]

- Player(int,int,Stage)
- setKey(boolean):void
- update():void
- backward():void
- right():void
- left():void
- forward():void
- charUser(int):Unit
- draw (GraphicsContext):void
- setDown():void
- setSatamina(int):void
- getSatamina():int
- setStartPlayer(int,int):void
- hasItem(Item):boolean
- recieveItem(Item):void
- removeItem(Item):void
- setEscape(int):void
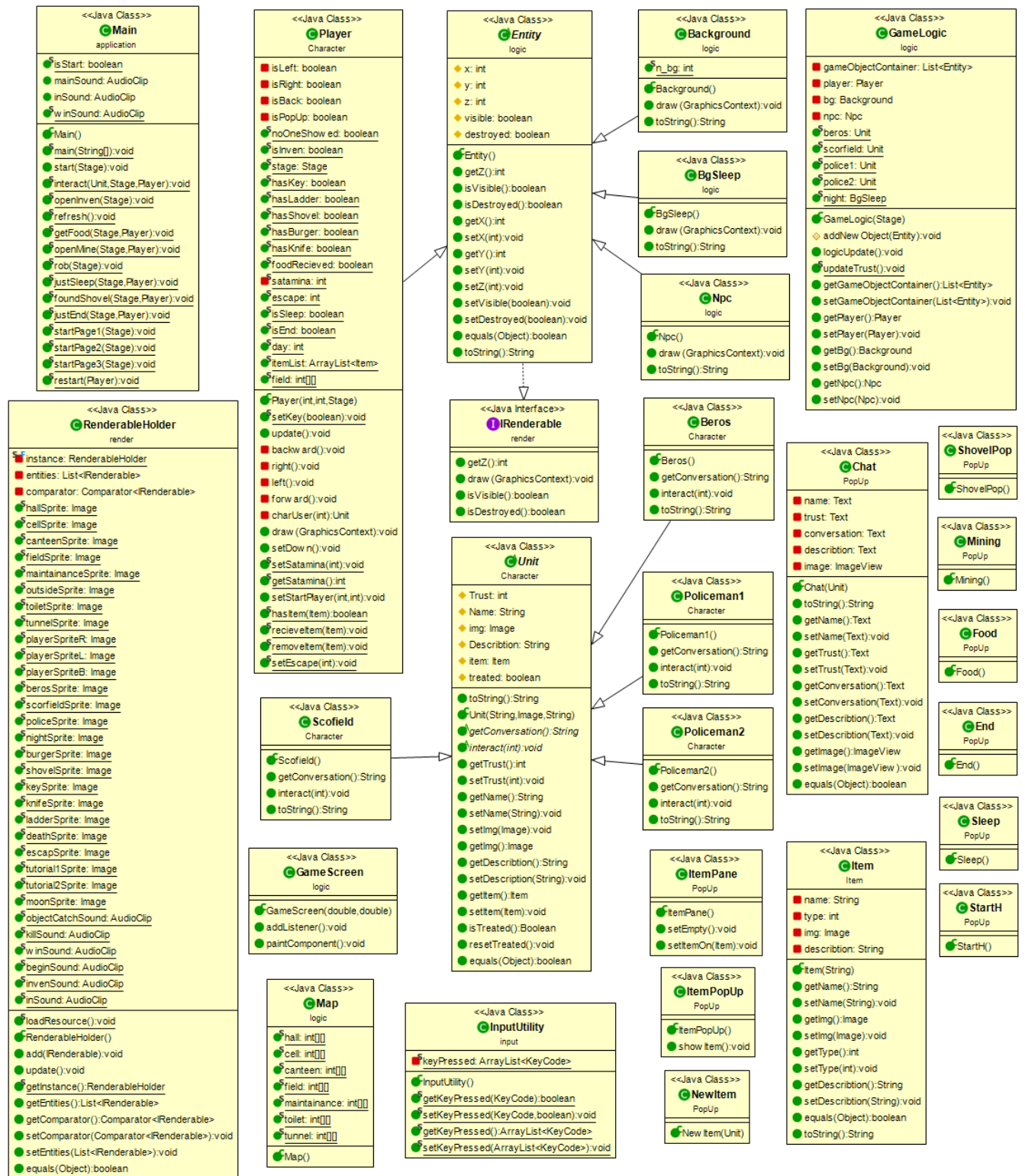
Entity
`<<Java Class>>`
Entity
logic

- x: int
- y: int
- z: int
- visible: boolean
- destroyed: boolean

- Entity()
- getZ():int
- isVisible():boolean
- isDestroyed():boolean
- getX():int
- setX(int):void
- getY():int
- setY(int):void
- setZ(int):void
- setVisible(boolean):void
- setDestroyed(boolean):void
- equals(Object):boolean
- toString():String

Background
`<<Java Class>>`
Background
logic

- n_bg: int

- Background()
- draw (GraphicsContext):void
- toString():String

BgSleep
`<<Java Class>>`
BgSleep
logic

- BgSleep()
- draw (GraphicsContext):void
- toString():String

Npc
`<<Java Class>>`
Npc
logic

- Npc()
- draw (GraphicsContext):void
- toString():String

GameLogic
`<<Java Class>>`
GameLogic
logic

- gameObjectContainer: List<Entity>
- player: Player
- bg: Background
- npc: Npc
- beros: Unit
- scorfield: Unit
- police1: Unit
- police2: Unit
- night: BgSleep

- GameLogic(Stage)
- addNewObject(Entity):void
- logicUpdate():void
- updateTrust():void
- getGameObjectContainer():List<Entity>
- setGameObjectContainer(List<Entity>):void
- getPlayer():Player
- setPlayer(Player):void
- getBg():Background
- setBg(Background):void
- getNpc():Npc
- setNpc(Npc):void

RenderableHolder
`<<Java Class>>`
RenderableHolder
render

- instance: RenderableHolder
- entities: List<IRenderable>
- comparator: Comparator<IRenderable>
- hallSprite: Image
- cellSprite: Image
- canteenSprite: Image
- fieldSprite: Image
- maintainanceSprite: Image
- outsideSprite: Image
- toiletSprite: Image
- tunnelSprite: Image
- playerSpriteR: Image
- playerSpriteL: Image
- playerSpriteB: Image
- berosSprite: Image
- scorfieldSprite: Image
- policeSprite: Image
- nightSprite: Image
- burgerSprite: Image
- shovelSprite: Image
- keySprite: Image
- knifeSprite: Image
- ladderSprite: Image
- deathSprite: Image
- escapSprite: Image
- tutorial1Sprite: Image
- tutorial2Sprite: Image
- moonSprite: Image
- objectCatchSound: AudioClip
- killSound: AudioClip
- winSound: AudioClip
- beginSound: AudioClip
- invenSound: AudioClip
- inSound: AudioClip

- loadResource():void
- RenderableHolder()
- add(IRenderable):void
- update():void
- getInstance():RenderableHolder
- getEntities():List<IRenderable>
- getComparator():Comparator<IRenderable>
- setComparator(Comparator<IRenderable>):void
- setEntities(List<IRenderable>):void
- equals(Object):boolean

IRenderable
`<<Java Interface>>`
IRenderable
render

- getZ():int
- draw (GraphicsContext):void
- isVisible():boolean
- isDestroyed():boolean

Beros
`<<Java Class>>`
Beros
Character

- Beros()
- getConversation():String
- interact(int):void
- toString():String

Policeman1
`<<Java Class>>`
Policeman1
Character

- Policeman1()
- getConversation():String
- interact(int):void
- toString():String

Policeman2
`<<Java Class>>`
Policeman2
Character

- Policeman2()
- getConversation():String
- interact(int):void
- toString():String

Unit
`<<Java Class>>`
Unit
Character

- Trust: int
- Name: String
- img: Image
- Describtion: String
- item: Item
- treated: boolean

- toString():String
- Unit(String,Image,String)
- getConversation():String
- interact(int):void
- getTrust():int
- setTrust(int):void
- getName():String
- setName(String):void
- setImg(Image):void
- getImg():Image
- getDescribtion():String
- setDescription(String):void
- getItem():Item
- setItem(Item):void
- isTreated():Boolean
- resetTreated():void
- equals(Object):boolean

Scofield
`<<Java Class>>`
Scofield
Character

- Scofield()
- getConversation():String
- interact(int):void
- toString():String

GameScreen
`<<Java Class>>`
GameScreen
logic

- GameScreen(double,double)
- addListener():void
- paintComponent():void

Map
`<<Java Class>>`
Map
logic

- hall: int[][]
- cell: int[][]
- canteen: int[][]
- field: int[][]
- maintainance: int[][]
- toilet: int[][]
- tunnel: int[][]

- Map()

InputUtility
`<<Java Class>>`
InputUtility
input

- keyPressed: ArrayList<KeyCode>

- InputUtility()
- getKeyPressed(KeyCode):boolean
- setKeyPressed(KeyCode,boolean):void
- getKeyPressed():ArrayList<KeyCode>
- setKeyPressed(ArrayList<KeyCode>):void

Chat
`<<Java Class>>`
Chat
PopUp

- name: Text
- trust: Text
- conversation: Text
- describtion: Text
- image: ImageView

- Chat(Unit)
- toString():String
- getName():Text
- setName(Text):void
- getTrust():Text
- setTrust(Text):void
- getConversation():Text
- setConversation(Text):void
- getDescribtion():Text
- setDescribtion(Text):void
- getImage():ImageView
- setImage(ImageView):void
- equals(Object):boolean

ItemPane
`<<Java Class>>`
ItemPane
PopUp

- ItemPane()
- setEmpty():void
- setItemOn(Item):void

ItemPopUp
`<<Java Class>>`
ItemPopUp
PopUp

- ItemPopUp()
- showItem():void

NewItem
`<<Java Class>>`
NewItem
PopUp

- NewItem(Unit)

Item
`<<Java Class>>`
Item
Item

- name: String
- type: int
- img: Image
- describtion: String

- Item(String)
- getName():String
- setName(String):void
- getImg():Image
- setImg(Image):void
- getType():int
- setType(int):void
- getDescribtion():String
- setDescription(String):void
- equals(Object):boolean
- toString():String

ShovelPop
`<<Java Class>>`
ShovelPop
PopUp

- ShovelPop()

Mining
`<<Java Class>>`
Mining
PopUp

- Mining()

Food
`<<Java Class>>`
Food
PopUp

- Food()

End
`<<Java Class>>`
End
PopUp

- End()

Sleep
`<<Java Class>>`
Sleep
PopUp

- Sleep()

StartH
`<<Java Class>>`
StartH
PopUp

- StartH()

*Noted that Access Modifier Notations can be listed below.

> + (public)
>
> # (protected)
>
> - (private)
>
> <u>static</u> 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.
+ <u>Audio winSound</u>	Sound when the game end.

1.1.2 Methods

+ void main(String[] args)	Application.*launch*(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. - If Player has burger, create empty pane to cover the burger item and set on action to eat and reopen inventory.
+ <u>void getFood(Stage stage, Player player)</u>	- Create Food class to be a Popup - Create OK Button to close the popup window. - update Player.foodRevieved to be true. - update item status in Player.
+ <u>openMine(Stage stage, Player player)</u>	- Create Mining class to be a Popup - Create OK Button to close the popup window.
+ <u>void rob(Stage stage)</u>	- Create NewItem class to be a Popup - Create OK Button to close the popup window. - update ladder status in player.
+ <u>justSleep(Stage stage, Player player)</u>	- Create Sleep class to be a Popup - Create OK Button to back to the cell. - update days left.
+ <u>void foundShovel(Stage stage, Player player)</u>	- Create ShovelPop class to be a Popup - Create OK Button to close the popup window. - update shovel status in player.
+ <u>void justEnd(Stage stage, Player player)</u>	- Create End class to be a Popup. - Create restart Button to restart game. - Create end button to close the game.
+ <u>void startPage1(Stage stage)</u>	- Create startH class to be a Popup (first scene when open the game).

	- Create start Button to start to call startPage2.
+ <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

2.1.1Fields

# 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 + "]".
+ *String getConversation()*	- 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 getDescribtion()	- 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 + ", Describtion=" + Describtion + "]".

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 + ", Describtion=" + Describtion + "]".

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 isPolpUp	Is there any popup window.
+ boolean noOneShowed	Is "No one here tonight" popup still showed.
+ boolean hasKey	Player has key.
+ boolean hasLadder	Player has ladder.
+ boolean hasShovel	Player has shovel.
+ boolean hasBurger	Player has burger.
+ boolean hasKnife	Player has knife.
+ boolean foodRecieved	Player already get the food for today.
+ int stamina	Player stamina.
+ int escape	Percentage of escape complete.
+ boolean isSleep	Sleep.
+ boolean isEnd	Is the game end.
+ int day	The day left.
+ ArrayList<Item> itemList	Player item.
+ Stage stage	Stage that showed in the current time.
+ int[][] field	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.
+ update()	- 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<KeyCode> keyPressed</u>	List of input.

3.2.2 Methods

+ boolean getKeyPressed(KeyCode keycode)	- Return true if keycode is in keyPressed. otherwise, return false.
+ boolean setKeyPressed(KeyCode keycode , boolean, pressed)	pressed is true. - If there is no keycode in pressed, add keycode in pressed. - If there is keycode in pressed , do nothing. Pressed is false. - Remove keycode in pressed.

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()	- Getter/Setter for description.

+ setDescription(String 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() + void setX()	- Getter/Setter for x.
+ 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

+ int n_bg	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)	If isStart in Main is false.
	   - gc will draw image that is beginning image in position (0, 0).
	If isStart in Main is true.
	   - gc will draw image in position (0, 0).
	Note : Each image is not the same if n_bg is not equal.

+ 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 pot in array 2 dimension.

5.5.1 Fields

+ int[][] hall	Map plot for hall.
+ int[][] cell	Map plot for cell.
+ int[][] canteen	Map plot for canteen.
+ int[][] field	Map plot for field.

+ int□□ maintainance	Map plot for maintenance.
+ int□□ toilet	Map plot for toilet.
+ int□□ tunnel	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: - call super class constructor with width and height. - setVisible(true). - addListener().

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()	- Getter/Setter for npc.

+ void setNpc(Npc 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.
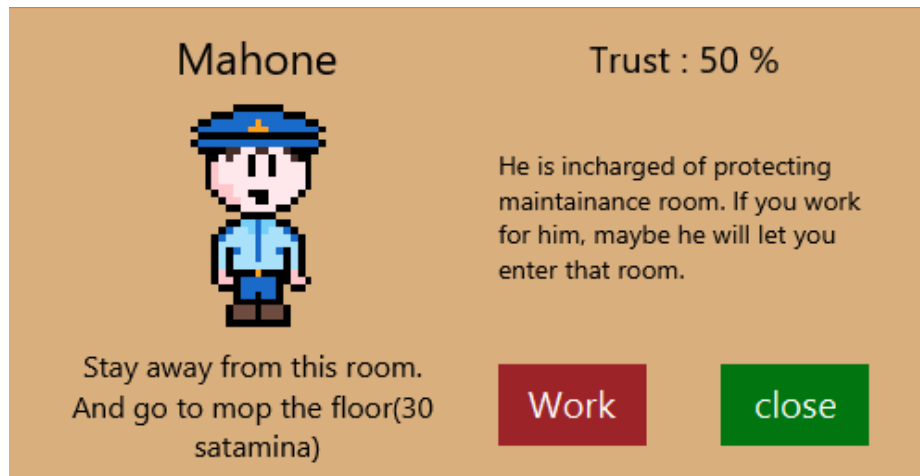
6.1.1 Fields

- 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)	- Initialize fields - Create inter button that can push to interact with Unit. The handle will change according to unit that player interacted with. - Use add all node to GridPane.

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
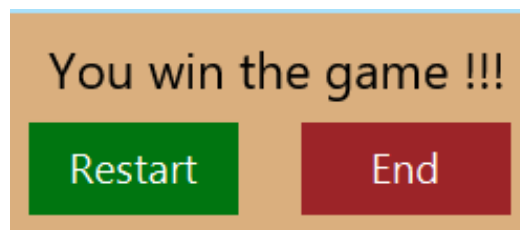
6.2.1 Constructor

+ 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()	- Create Text to show Player receiving the food. - Create imageView of Burger.

| | - Create Text to show description. |
| | - add all node to GridPane. |

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()	- Set GridPane properties.
	- createText to shoe Player stamina.

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.
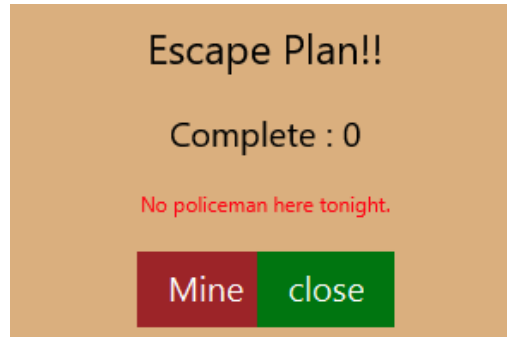
6.6.1 Constructor

+ Mining()	- Create Text to show Mining status.
	- Create Text to show mining condition.
	- Create Button to start mining.

6.7 class newItem extends GridPane

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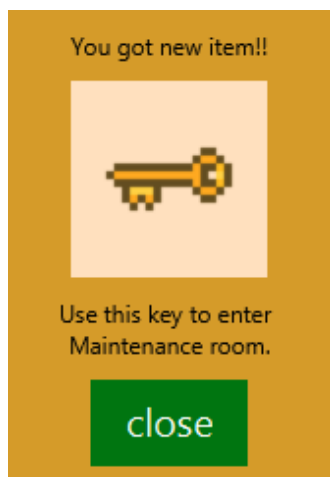
6.7.1 Constructor

+ NewItem(Unit unit)	- Create Text to show item name.
	- Create ImageView to show item image.
	- Create Text to show item description.

	- Set GridPane properties.
	- Add all node to GridPane.

6.8 class ShovelPop extends GridPane

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

6.8.1 Constructor

+ ShovelPop()	- Create Text to show item name.
	- Create ImageView to show shovel image.
	- Create Text to show shovel description.
	- Set GridPane properties.
	- Add all node to GridPane.

6.9 class Sleep extends GridPane

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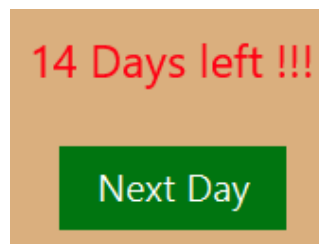
6.9.1 Constructors

+ Sleep()	- Create Text to show the days left.
	- set GridPane properties.

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

- final RenderableHolder instance	Initialize instance.
- List<IRenderable> entities	List of entities.
- Comparator<IRenderable> comparator	Comparator for compare IRenderable.
+ Image hallSprite	Image for hall background.
+ Image cellSprite	Image for cell background.
+ Image canteenSprite	Image for canteen background.

+ Image fieldSprite	Image for field background.
+ Image maintainanceSprite	Image for maintenance background.
+ Image outsideSprite	Image for outside background.
+ Image toiletSprite	Image for toilet background.
+ Image tunnelSprite	Image for tunnel background.
+ Image playerSpriteR	Image for player right-side.
+ Image playerSpriteL	Image for player left-side.
+ Image playerSpriteB	Image for player back-side.
+ Image berosSprite	Image for Beros.
+ Image scorfieldSprite	Image for Scorfield.
+ Image policeSprite	Image for police.
+ Image nightSprite	Image for night background.
+ Image burgerSprite	Image for burger.
+ Image shovelSprite	Image for shovel.
+ Image keySprite	Image for key.
+ Image knifeSprite	Image for knife.
+ Image ladderSprite	Image for ladder.
+ Image deathSprite	Image for death background.
+ Image escapSprite	Image for escapist background.
+ Image tutorial1Sprite	Image for tutorial background.
+ Image tutorial2Sprite	Image for tutorial background.
+ Image moonSprite	Image for moon.
+ AudioClip objectCatchSound	Sound when catch object.
+ AudioClip killSound	Sound when you are killed.
+ AudioClip winSound	Sound when you win.
+ AudioClip beginSound	Sound before game begins.
+ AudioClip invenSound	Sound when click button or open inventory.
+ AudioClip inSound	Sound when game begins.

7.2.2 Constructor

+ RenderableHolder()	- Initialize entities and comparator.

7.2.3 Methods

| + void add(IRenderable entity) | - add entity in entities. |
	- sort entities with comparator.
+ void update()	- remove Entity when Entity is destroyed.
+ <u>RenderableHolder getInstance()</u>	- return instance.
+ List<IRenderable> getEntities()	- Getter/Setter for entities.
+ void setEntities(List<IRenderable> entities)	
+ Comparator<IRenderable> getComparator()	- Getter/Setter for comparator.
+ void setComparator(Comparator <IRenderable> comparator)	
+ boolean equals(Object obj)	- Generated default.