**TongRod99 Documentation**

**Created by**

Tittaya Worawongtad 6431314821

Naphat Wareesawetsuwan 6431315421

**Missing: Hide and Seek**

**Introduction**

Missing: Solitary Hide and Seek is inspired by an RPG named “HideAndSeek[The story of Dorothy]” and other series of Big Fish Games’ casual hidden object adventure games.

The game begins with Anna, the girl who lost her own memory, trying to escape from the room she is in. On the way she finds her lovely doll which hides in each room, she can notice that everything is curiouser and curiouser.

**Rules**

The player uses arrow keys to move the character on the screen and uses the space bar to pick and use an object. The goal of each room is to find a way to open the door.

**Example**

1. Using arrow keys to move the character
2. Using the space bar to pick an object
3. Choosing an object and using it

**Main Menu Scene**

**Class diagram**

+ (public)

# (protected)

- (private)

Underline (static)

Italic (abstract)

**1. package application**

**1.1 class Main**

This class contains the main method. It is an entry point of the application.

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + void start(Stage primaryStage) throws Exception | - Initialize mainMenu with the MainMenu  - Create the scene with the mainMenu  - Set primaryStage scene as the created scene  - Set primaryStage title as “Missing: Hide and Seek”  - Set up primaryStage height as 700  - Set primaryStage width as 1120  - Set primaryStage resizable as false  - Show primaryStage |
| + void main(String[] args) | The entry point of the application |

**2. package entity**

**2.1 entity.base**

**2.1.1 Interface Cuttable**

This interface defines methods for furniture that can be cut.

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + void cut() | This method will be called when the furniture got cut. |
| + Boolean isCut() | State that the furniture got cut or not |

**2.1.2 Interface Lockable**

This interface defines methods for furniture that can be locked.

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + boolean isLocked() | State that the furniture got locked or not |
| + void unlock() | This method will be called when the furniture got unlocked. |

**2.1.3 Interface Openable**

This interface defines methods for furniture that can be opened.

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + void open() | This method will be called when the furniture got opened. |
| + boolean isOpened() | State that the furniture got opened or not |
| + void close() | This method will be called when the furniture got closed. |

**2.1.4 Interface Pickable**

This interface defines methods for furniture that can be picked.

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + void pick() | This method will be called when the furniture got picked. |
| + boolean isPicked() | State that the furniture got picked or not |

**2.1.5 Interface Updateable**

This interface defines methods for furniture that can be updated.

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + void update() | This method will be called when the furniture got updated. |
| + Boolean isUpdated() | It returns **true** if the furniture got updated, returns **false** otherwise. |

**2.1.6 Enum Emotion**

This class represents character’s emotion. It contains the following values: NORMAL, WORRIED and SURPRISED.

**2.1.7 *Abstract* class Container extends Furniture**

This class represents the furniture that can contain something.

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - ArrayList<Item> item | List that contains the item in the furniture. |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + Container(String name, int xPosition, int yPosition) | - Initialize the container fields with respective values.  - Initialize item as an empty ArrayList |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + boolean isContaining() | This method checks if the furniture is containing anything. |
| + getter and setter for each field |  |

**2.1.8 *Abstract* class Furniture implements IRenderable**

This class represents the furniture that is used in the room.

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - String name; | Name of the furniture which will be displayed and use to identify the furniture. |
| - double xPosition | Position of the furniture in X-axis |
| - double yPositon | Position of the furniture in Y-axis |
| - int z | The number which related to the order of rendering image on the screen. |
| - boolean isVisible | State that the furniture is still visible or not |
| - String imageString | The name of the furniture’s image |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + public Furniture(String name, int xPosition, int yPosition) | - Initialize the Furniture fields with respective values.  - Set imageString as name |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + void draw(GraphicsContext gc) | Draw the furniture on its current position. |
| + String toString() | Returns a formatted string in the format of “This is a normal + <name>.” |
| + void observe | This method will represent the furniture’s name which observed by the player |
| + void useItem() | - This method will represent the default text which show that this item can not use with this furniture  - set gameText with “I think it’s better to use it somewhere.” |
| + boolean isVisible() | It returns **true** if the furniture is visible, returns **false** otherwise. |
| + int getZ() | Return z |
| + void draw(GraphicsContext gc) | Draw the furniture on its current position. |
| + boolean equals(Object obj) | This method which is check equality of two Boolean object. |
| + getter and setter for each field |  |

**2.1.9 *Abstract* Class Item implements IRenderable**

This class represents the item that is used in the room

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - String name | Name of the item which will be displayed and use to identify the item |
| - double xPosition | Position of the item in X-axis |
| - double yPosition | Position of the item in Y-axis |
| - int z | The number which related to the order of rendering image on the screen. |
| + boolean isVisible | Keeps track if the item has been visible or not |
| - final double width | Initialize it with 30 |
| - final double height | Initialize it with 30 |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + Item(String name) | - Initialize the item fields with respective values.  - Set to not visible by default. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + boolean isVisible() | It returns **true** if the furniture is visible, returns **false** otherwise. |
| + int getZ() | Return z |
| + void draw(GraphicsContext gc) | Draw the item on its current position. |
| + getter and setter for each field |  |

**2.1.10 *Abstract* class Room**

This class represents room’s map which appears in playing screen.

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - String name | Name of the room which will be displayed and use to identify the item |
| - String startText | The message that appears when the room is start |
| - String endText | The message that appears when the room is end |
| - ArrayList<Furniture> furniture | The list of furniture in the room |
| - boolean isStart | True if the room is started |
| - boolean isEnd | True if the room is ended |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + Room(String name, String startText, String endText) | Initialize the room fields with respective values. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + void start() |  |
| + void end() |  |
| + getter and setter for each field |  |

**2.1.11 Enum RoomName**

This class represents room’s name. It contains the following values:

BEDROOM, LIVING\_ROOM, LIBRARY, GARDEN.

**2.2 entity.furniture**

**2.2.1 Class Bed extends Furniture**

This class represent a bed which is one of the furniture.

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| Bed(String name, int xPosition, int yPosition) | Initialize the bed fields with respective values. |

**2.2.2 Class Bookshelf extends Container**

This class represent a bookshelf which is one of the containers.

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - ArrayList<Item> item | The list of items in the bookkshelf |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| Bookshelf(String name, int xPosition, int yPosition) | - Initialize the bookshelf fields with respective values.  - Initialize item as an empty ArrayList |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + void observe() | - Check all items in bookshelf  - If it is a note, call read() method and remove it  - If it is a key, call pick() method and remove it  - Otherwise, set gameText with “There is a lot of books here, I would love to read if I have time.” |
| + getter and setter for each field |  |

**2.2.3 Class Chair extends Furniture**

This class represent a chair which is one of the furniture.

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| Chair(String name, int xPosition, int yPosition) | Initialize the chair fields with respective values. |

**2.2.4 Class Cupboard extends Container implements Lockable, Openable**

This class represent a bed which is one of the furniture and can be locked/ opened.

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - boolean isLocked | State that the cupboard got locked or not |
| - boolean isOpened | State that the cupboard got opened or not |
| - ArrayList<Item> item | The list of items in the cupboard |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| Cupboard(String name, int xPosition, int yPosition, String text) | - Initialize the cupboard fields with respective values.  - It sets to locked and not been opened by default. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + boolean isOpened() | It returns **true** if the cupboard got opened, returns **false** otherwise. |
| + void open() | - If the cupboard is locked, set gameText as “It’s locked.”  - Otherwise, set isOpened as true and imageString as “OpenedCupboard” |
| + void close() | It set isOpened as false and imageString as cupboard’s name |
| + boolean isLocked() | It returns **true** if the cupboard got locked, returns **false** otherwise. |
| + void unlock() | - If player’s hand has key and this key can use with this cupboard,  - Set isLocked as false  - Call open() method  - Visible all items in cupboard.  - Otherwise, set gameText as “I think it doesn’t fit here.” |
| + void useItem() | - If the cupboard is locked, call unlock() method  - Otherwise, call useItem() method from super class |
| + void observe() | - Check all items in cupboard  - If it is a note, call read() method and remove it  - If it is a key, call pick() method and remove it  - Otherwise, set gameText with “It’s just an empty cupboard.” |
| + getter and setter for each field |  |

**2.2.5 Class Mirror extends Container implements Updateable**

This class represent a mirror which is one of the containers and can be updated.

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| + boolean isUpdated | State that the mirror got updated or not |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| Mirror(String name, int xPosition, int yPosition) | - Initialize the mirror fields with respective values.  - Set to not been updated by default. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + void update() | - Set imageString as “BehindMirror” and visible all items in mirror  - For all items in mirror, set isVisible as true  - Set isUpdated as true |
| + boolean isUpdated() | It returns **true** if the mirror got updated, returns **false** otherwise. |
| + void observe() | - If the mirror is not updated, call update() method  - If the mirror is updated and has items in mirror, check all items in mirror  - If it is a note, call read() method and remove it  - If it is a key, call pick() method and remove it  - Otherwise, set gameText as “It’s broken…” |
| + getter and setter for each field |  |

**2.2.6 Class Picture extends Furniture**

This class represent a picture which is one of the furniture.

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - String description | This represents the picture’s description which observed by the player |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + Picture(String name, int xPosition, int yPosition, String description) | - Initialize the picture fields with respective values. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + void observe() | Set gameText as picture’s description |
| + getter and setter for each field |  |

**2.2.7 Class PictureWithItem extends Picture**

This class represent a picture which has items inside its.

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - ArrayList<Item> itemBehind | The list of items in the safe behind picture |
| - boolean isReveal | State that the safe behind picture got revealed or not. |
| - boolean isSafeBehind | State that has safe behind the picture or not. |
| - boolean isLocked | State that the safe behind picture got locked or not. |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + PictureWithItem(String name, int xPosition, int yPosition, String description) | - Initialize the pictureWithItem fields with respective values.  - Set isSafeBehind, isLocked and isReveal as false |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + void observe() | - If it has safe behind the picture, call observeWithSafe() method  - Otherwise, call observeBehind() method |
| + void observeWithSafe() | - If safe behind the picture is not revealed,  - Set isReveal as true  - Set imageString as “ClosedSafe”  - Set gameText “There is a safe behind.”  - If the safe is locked, set gameText as “It’s locked.”  - Otherwise, call observeBehind() method |
| + void observeBehind() | - If it has items in the safe, Check all the items  - If item is a note, call read() method  - If item is a key, call pick() method  - Otherwise, set gameText as “I never thought items can hide here” |
| + void useItem() | - If it has safe behind the picture and got revealed and locked, call unlock() method  - Otherwise, call useItem() from super class |
| + void unlock() | - If player’s hand has key and this key can use with this safe,  - Set isLocked as false  - Set imageString as “OpenedSafe”  - Visible all items in cupboard.  - Otherwise, set gameText as “I think it doesn’t fit here.” |
| + getter and setter for each field |  |

**2.2.7 Class Sofa extends Container implements Cuttable**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - boolean isCut | State that the sofa got cut or not. |
| - boolean isSomethingBehind | State that it has something behind the sofa or not. |
| - ArrayList<Item> thingBehind | List of items that behind the sofa |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + Sofa(String name, int xPosition, int yPosition, boolean isSomethingBehind) | - Initialize the sofa fields with respective values.  - Set to not been cut by default |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + void cut() | - Set isCut as true  - Set imageString as “CutSofa”  - Set all items in sofa is visible |
| + boolean isCut() | It returns **true** if the sofa got cut, returns **false** otherwise. |
| + void useItem() |  |
| + void observe |  |
| + getter and setter for each field |  |

**2.2.8 Class Statue extends Furniture implements Updateable**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - String letterOnState | The letter on the statue |
| - String answerLetter | The answer of the letter on the statue |
| - boolean isMatch | True if the statue is matched |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + Statue(String name, int xPosition, int yPosition, String letterOnStatue, String answerLetter, boolean isMatch) | Initialize the Furniture fields with respective values.  It set to not been match by default. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + boolean isUpdated() | Return isMatch |
| + void update() |  |
| + getter and setter for each field |  |

**2.2.9 Class TableWithLamp extends Furniture implements Updateble**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
|  |  |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + TableWithLamp(String name, int xPosition, int yPosition) | Initialize the Furniture fields with respective values.  It set to not been updated by default. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + void update() |  |
| + boolean isUpdated() | It returns **true** if the tableWithLamp got updated, returns **false** otherwise. |
| + getter and setter for each field |  |

**2.2.10 Class TableWithNote extends Container**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
|  |  |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + TableWithNote(String name, int xPosition, int yPosition) | Initialize the Furniture fields with respective values. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + getter and setter for each field |  |

**2.2.11 Class Window extends Furniture implements Openable**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - boolean isOpened | True if the window is opened |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + Window(String name, int xPosition, int yPosition) | Initialize the Furniture fields with respective values.  It set to not been opened by default. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + void open() |  |
| + boolean isOpened() | It returns **true** if the widow got opened, returns **false** otherwise. |
| + void close() |  |
| + getter and setter for each field |  |

**2.3 entity.item**

**2.3.1 Class Key extends Item implements Pickable**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - final String String keyURL | URL of key image. The value should be added in constructor as “key.png” |
| - boolean isPicked | True if the key is picked |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + Key(String name) | Initialize the Furniture fields with respective values.  Assign keyURL as “key.png”  It set to not been picked by default. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + boolean isPicked() | It returns **true** if the key got picked, returns **false** otherwise. |
| + void pick() |  |
| + getter and setter for each field |  |

**2.3.2 Class Note extends Item**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - final String noteURL | URL of note image. The value should be added in constructor as “note.png” |
| - String textOnNote | The text which appears on the note |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + Note(String textOnNote) | Initialize the Furniture fields with respective values.  Assign noteURL as “note.png” |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + getter and setter for each field |  |

**2.3.3 Class PocketKnife extends Item implements Pickable**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - final String pocketKnifeURL | URL of pocketknife image. The value should be added in constructor as “pocketKnife.png” |
| - boolean isPicked | True if the pocketknife is picked |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + PocketKnife(String name) | Initialize the Furniture fields with respective values.  Assign pocketKnifeURL as “pocketKnife.png”  It set to not been picked by default. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + void pick() |  |
| + boolean isPicked() | It returns **true** if the pocketknife got picked, returns **false** otherwise. |
| + getter and setter for each field |  |

**3. Package gui**

**3.1 gui.room**

**3.1.1 Class Bedroom extends Room**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
|  |  |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + Bedroom(String name, String startText, String endText) | Initialize the Furniture fields with respective values. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + getter and setter for each field |  |

**3.1.2 Class Garden extends Room**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
|  |  |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| Garden(String name, String startText, String endText) | Initialize the Furniture fields with respective values. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + getter and setter for each field |  |

**3.1.3 Class Library extends Room**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
|  |  |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| Library(String name, String startText, String endText) | Initialize the Furniture fields with respective values. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + getter and setter for each field |  |

**3.1.4 Class LivingRoom extends Room**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
|  |  |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| LivingRoom(String name, String startText, String endText) | Initialize the Furniture fields with respective values. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + getter and setter for each field |  |

**3.2 Class CharacterPane extends Pane**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - final String normalFaceURL | URL of normal face image. The value should be added in constructor as “normalFace.png” |
| - final String worriedFaceURL | URL of worried face image. The value should be added in constructor as “worriedFace.png” |
| - final String surprisedFaceURL | URL of surprised face image. The value should be added in constructor as “surprisedFace.png” |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + CharacterPane() | Assign noramlFaceURL as “normalFace.png, worriedFaceURL as “worriedFace.png”, surprisedFaceURL as “surprisedFace.png” |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + getter and setter for each field |  |

**3.3 Class ControlPane extends VBox**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - Button startButton | The button for starting the game |
| - Button instructionButton | The button for showing the instruction |
| - Button creditButton | The button for showing the credit |
| - Button exitButton | The button for exiting the game |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| ControlPane() | Initialize the Furniture fields with respective values. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + getter and setter for each field |  |

**3.4 Class CreditPane extends VBox**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
|  |  |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| CreditPane() | Initialize the Furniture fields with respective values. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + getter and setter for each field |  |

**3.5 Class DialoguePane extends Pane**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - final String dialoguePane | URL of dialogue background image. The value should be added in constructor as “dialogueBG.png” |
| - Text gameText |  |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + DialoguePane() | Assign dialogueBG\_URL as “dialogueBG.png” |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + getter and setter for each field |  |

**3.6 Class HowToPlayPane extends Pane**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
|  |  |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
|  | Initialize the Furniture fields with respective values. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + getter and setter for each field |  |

**3.7 Class InstructionPane extends VBox**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
|  |  |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + InstructionPane() | Initialize the Furniture fields with respective values. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + getter and setter for each field |  |

**3.8 Class ItemInHandBox extends Pane**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - final String boxBG\_URL | URL of box background image. The value should be added in constructor as “boxBG.png” |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + ItemInHandBox() | Assign boxBG\_URL as “boxBG.png” |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + getter and setter for each field |  |

**3.9 Class MenuButtonPane extends Pane**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
|  |  |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + MenuButtonPane() | Initialize the Furniture fields with respective values. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + getter and setter for each field |  |

**3.10 Class TimerPane extends Pane**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - private Text timer |  |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + TimerPane() | Initialize the Furniture fields with respective values. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + getter and setter for each field |  |

**4. Package input**

**4.1 Class InputUtility**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
|  |  |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
|  | Initialize the Furniture fields with respective values. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + getter and setter for each field |  |

**5. Package logic**

**5.1 Class Player**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - Item itemInHand | Item that the player picks in her/his hand |
| - Emotion playEmotion | Player’s emotion |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + Player() | Initialize the Furniture fields with respective values. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + getter and setter for each field |  |

**5.2 Class Timer**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - int minute |  |
| - int second |  |
| - int ms |  |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + Timer (int m, int s, int ms) | Initialize the Furniture fields with respective values. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + getter and setter for each field |  |

**6. Package screen**

**6.1 Class EndingScreen**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
|  |  |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + EndingScreen() | Initialize the Furniture fields with respective values. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| **+ getter and setter for each field** |  |

**6.2 Class MainMenu extends Scene**

instructionPane/ creditPane

**sidePane**

**Text, letter

Description automatically generated**

**Logo image**

**windowPane**

**controlPane**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - ControlPane controlPane | Includes all the control buttons in the main menu |
| - CreditPane creditPane | The creditPane where the player can see the credit of this game |
| - InstructionPane instructionPane | The instructionPane where to show the player how to play this game |
| - StackPane windowPane | The windowPane where to show the creditPane and instructionPane |
| - VBox sidePane | The sidePane that include the logo image and cotrolPane |
| + final int INDEFINITE | Initialize it to -1 |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + MainMenu() | - Initialize ControlPane with the Control Pane, CreditPane with the Credir Pane, InstructionPane with the Instruction Pane, sidePane with the VBox, windowPane with the Stack Pane  - Add controlPane and logo image to the sidePane  - set up sidePane and windowPane  - Add windowPane and sidePane to the mainMenu  - add background’s music by set cycle count to indefinite |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + void showCreditWindow() | - If windowPane is empty or show something else, clear windowPane and add creditPane to the windowPane  - Otherwise, clear the windowPane |
| + void showInstructionWindow() | - If windowPane is empty or show something else, clear windowPane and add instructionPane to the windowPane  - Otherwise, clear the windowPane |
| + getter and setter for each field |  |

**6.3 Class PlayingScreen**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - Button menuButton | Button which back to main menu |
| - CharacterPane character | characterPane where to show character image and emotion |
| - DialoguePane dialoguePane | dialoguePane where to show the message from what character feels or sees |
| - ItemInHandBox objectInHandBox | ItemInHandBox where show the item which player picks it up |
| - TimerPane timerPane |  |
| - Pane room | room where to show the room’s map |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + PlayingScreen() | - Initialize dialoguePane with the Dialogue Pane, itemInHandBox with the ItemInHandBox, room with the Pane  - Initialize upperPane and lowerPane with the HBox and set it up  - Add sidePane, room to the upperPane and dialoguePane, itemInHandBox to the lowerPane  - Add upperPane, lowerPane to the PlayingScreen  - Set up room and PlayingScreen |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + void initializeMenuButton() | - Initialize menuButton with text “MENU”  - Initialize shadow with the DropShadow  - set up menuButton and shadow  - Add EventHandler on mouse entered to have a shadow effect  - Add EventHandler on mouse exited to set shadow effect as null  - Add EventHandler on mouse click to return to main menu |
| + VBox initializeSidePane() | - Call initializeMenuButton() method  - Initialize character with the CharacterPane, timerPane with the TimerPane, sidePane with the VBox  - Add menuButton, timerPane, characterPane to the sidePane  - Set up the sidePane  - Return the sidePane |
| + getter and setter for each field |  |

**7. Package sharedObject**

**7.1 Interface IRenderable**

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + int getZ() |  |
| + void draw(GraphicsContext gc) |  |

**7.2 Class RenderableHolder**

Field

|  |  |
| --- | --- |
| **Name** | **Description** |
| - final RenderableHodler instance |  |
| - List<IRenderable> entities |  |
| - Comparator<IRenderable> comparator |  |
| + Map<String, Image> furnitureSprite |  |
| + Map<String, Image> itemSprite |  |
| + Map<Emotion, Image> characterFullBody |  |
| + Map<String, Image> characterSprite |  |
| + Map<String, Image> background |  |
| + Map<String, AudioClip> soundFX |  |
| + Map<String, AudioClip> bgMusic |  |

Constructor

|  |  |
| --- | --- |
| **Name** | **Description** |
| + RenderableHolder() | Initialize the Furniture fields with respective values. |

Method

|  |  |
| --- | --- |
| **Name** | **Description** |
| + RenderableHolder getInstance() |  |
| + void loadResource() |  |
| + void add(IRenderable entity) |  |
| + void update() |  |
| + List<IRenderable> getEntities() |  |
| + getter and setter for each field |  |