**The Disappearance of Emily Jane**

David Hatcher

Spring 2020 Object-Oriented Software Design Final Project Report

## PROJECT SUMMARY

The Disappearance of Emily Jane will be a mystery solving game, in the vain of the old Myst style games, but entirely text based. In this game you will be a detective working the case of a young woman who has been reported missing. After weeks of no leads and very little information you decide to search Emily's house one last time in the hopes that you can figure out what happened to her. While there you will be forced to find clues and defeat a sinister entity to determine what happened to Ms. Jane.

## OVERVIEW OF CLASSES

Several classes were created for this project:

1. **Player**: This class contains the information for the player, including flags, and items in their inventory. Each player object represents the person playing the game.
2. **Game:** This class is used to operate the various functions of the game. The game function is loaded only once
3. **Item:** The item class is used to create items for a player's inventory, each item object represents a single item in the game
4. **Interactive:** The interactive class is used to create interactive elements of the rooms in the game
5. **Room:** The room class is the parent class for all the various rooms
6. **Entrance:** The entrance class is used for the entrance room.
7. **Living_Room:** The living_room class is used for the living room
8. **Basement:** The basement class is used for the basement room
9. **Kitchen:** The kitchen class is used for the kitchen room
10. **Library:** The library class is used for the library room
11. **Office:** The office class is used for the office room
12. **Helper:** The helper class is used to store the static functions

## CLASS DESCRIPTIONS

PLAYER CLASS

Abstract class: No

Subclass of: N/A

Composed of: Item

Alterations post proposal:

- One data member, journal_default, was added because I needed to be able to store the default value and add on the values that are added in during the game
- 5 Member functions where added: addJournal to be able to append the journal entries throughout the game,showInventory was added to easily access inventory items, loadData and saveData were added to allow for easier saving for the player values, and getJournalDefault was added to accommodate the addition of journal_default.

| DATA MEMBERS - PLAYER | | | |
|---|---|---|---|
| **Variable Name** | **Data Type** | **Static** | **Description** |
| name | string | no | Holds the players name. |
| inventory | Vector<Item> | No | Holds the items the player has in their inventory |
| Journal | String | No | Holds the players journal |
| Journal default | String | No | Holds the default journal value |
| Flags | Map<string,bool> | No | Holds the players flags for completion aspects of the game |

| MEMBER FUNCTIONS - PLAYER | | | | | |
|---|---|---|---|---|---|
| **Signature** | **Static** | **Virtual** | **Operator** | **Friend** | **Description** |
| Player(string) | No | No | No | No | Constructor for player class |
| Void setFlag(string,bool) | No | No | No | No | Sets a flag on the player |
| Void setJournal(string) | No | No | No | No | Adds a string to the players journal |
| Void setName(string) | No | No | No | No | Sets the players name |
| String getName() | No | No | No | No | Gets the players name |
| String getJournal() | No | No | No | No | Gets the players journal |
| String getJournalDefault() | No | No | No | No | Gets the players default journal string |
| Bool checkFlags(string) | No | No | No | No | Gets the value of a specific flag |
| Bool checkInventory(Item) | No | No | No | No | Checks if a specific item is in the players inventory |
| Void addToInventory(Item) | No | No | No | No | Adds an item to a players inventory |

| | | | | | |
|---|---|---|---|---|---|
| String showInventory() | No | No | No | No | Returns a string of the players inventory |
| Friend bool loadData(string,Player) | No | No | No | Yes | Loads data of player |
| Friend bool saveData(string,Player) | No | No | No | Yes | Saves data of player |

GAME CLASS
Abstract class: No
Subclass of: N/A
Composed of: Room,Item,Player,Interactive,Entrance,Living_Room,Bedroom,Kitchen,Office,Library,
Basement
Alterations post proposal:

- 3 Data members,room_names,rooms,save_loc, removed as I changed the way saving was done in this and the way the rooms we being stored and moved around
- 8 Data members were added because I changed the way the rooms were being stored from an array of all of them to individual data members for each room
- 4 member functions were removed and moved to the Helper class as I was trying to circularly include the Game class in the Room class and the Room class in the Game class and this was causing errors and would not compile.
- 7 member functions were added because I changed the way the rooms were being stored as I wasn't able to do it the way I originally wanted to.

| **DATA MEMBERS - Game** | | | |
|---|---|---|---|
| **Variable Name** | **Data Type** | **Static** | **Description** |
| name | string | no | Holds the name of the player |
| gameRunning | Bool | No | This is used to determine if the game is still running |
| Office | Office | No | Office object container |
| Kitchen | Kitchen | No | Kitchen object container |
| Bedroom | Bedroom | No | Bedroom object container |
| Entrance | Entrance | No | Entrance object container |
| Living_room | Living_Roo | No | Living_Room object container |

|  |  |  |  |
|---|---|---|---|
|  | m |  |  |
| Library | Library | No | Library object container |
| Basement | Basement | No | Basement object container |
| Current_room | Room* | No | This is a pointer to the current room the player is in |
| Player | Player* | No | This is a pointer to the player object |

| **MEMBER FUNCTIONS - Game** |  |  |  |  |  |
|---|---|---|---|---|---|
| **Signature** | **Static** | **Virtual** | **Operator** | **Friend** | **Description** |
| Void buildRooms() | No | No | No | No | Builds all the rooms |
| Kitchen buildKitchen(string) | No | No | No | No | Builds the kitchen object and stores it |
| Entrance buildEntrance(string) | No | No | No | No | Builds the Entrance object and stores it |
| Living_Room buildLiving_Room(string) | No | No | No | No | Builds the Living_room object and stores it |
| Bedroom buildBedroom(string) | No | No | No | No | Builds the bedroom object and stores it |
| Office buildOffice(string) | No | No | No | No | Builds the office object and stores it |
| Library buildLibrary(string) | No | No | No | No | Builds the library object and stores it |
| Basement buildBasement(string) | No | No | No | No | Builds the basement object and stores it |
| Game(string,Player&) | No | No | No | No | Contructor |
| String printJournal | No | No | No | No | Returns the journal contents of the player |
| String interactWith(string,st | No | No | No | No | Allows the player to interact with an object in the a room |

| | | | | | |
|---|---|---|---|---|---|
| ring) | | | | | |
| Bool loadData() | No | No | No | Yes | Loads the players data |
| Bool load() | No | No | No | No | Allows main.cpp to run the loadData function |
| Bool saveData | No | No | No | No | Saves the players data |
| Void printRoomDescription() | No | No | No | No | Prints the rooms description |
| String runGame() | No | No | No | No | This function is the main logic of the game and returns the output to the console |
| Bool checkGameRunning() | No | No | No | No | Returns the data member is game running |

ROOM CLASS
Abstract class: Yes
Subclass of: N/A
Composed of: Items, Helper, Interactive
Alterations post proposal:

- Removed one member function,setExits as I did not end up using it, also changed the paramter of setDescription to a player reference to work with the flags better
- Added getExits to work with exits better, also added getName to be able to display the name from the game class
- Data member exits was changed to a vector of Room pointers rather than a vector of rooms.

| **DATA MEMBERS - Room** | | | |
|---|---|---|---|
| **Variable Name** | **Data Type** | **Static** | **Description** |
| Name | string | No | Holds the name of the room |
| File_path | String | No | Holds the file_path of the room |
| Interactives | Map<string,Interactive> | No | Holds the interactives in the room |
| Exits | Vector<Room*> | No | Holds pointers to rooms that are connected to this one |

| | | | |
|---|---|---|---|
| Description | String | No | Holds the description of the room |
| Items | Vector<Item> | No | Holds the items in this room |

| **MEMBER FUNCTIONS - Room** | | | | | |
|---|---|---|---|---|---|
| **Signature** | **Static** | **Virtual** | **Operator** | **Friend** | **Description** |
| Vector<Room*> get exits() | No | No | No | No | Returns the exits of the room |
| String getName() | No | No | No | No | Returns the name of the string |
| Void setName(string) | No | No | No | No | Sets the name of the room |
| Void setDescription(Player &) | No | Yes | No | No | Sets the description of the room |
| String getDescription() | No | No | No | No | Returns the description of the room |
| Void addInteractive(string, Interactive) | No | No | No | No | Adds and interactive object to the list of interactives in the room |
| Map<string,Interactive> getInteractives() | No | No | No | No | Returns a map of the interactives with strings as the keys |
| Void addExit(Room*) | No | No | No | No | Adds and exit to the |
| String interactWith(string,string,Player&) | No | Yes | No | No | Interacts with something in the room object, this is virtual as all the children over write this. |

ITEM Class
Abstract class: No
Subclass of: N/A
Composed of: N/A
Alterations post proposal:

- None

| DATA MEMBERS - Item | | | |
|---|---|---|---|
| **Variable Name** | **Data Type** | **Static** | **Description** |
| Name | String | No | Holds the name of the item |
| Description | String | No | Holds the description of the item |

| MEMBER FUNCTIONS - Item | | | | | |
|---|---|---|---|---|---|
| **Signature** | **Static** | **Virtual** | **Operator** | **Friend** | **Description** |
| Void setDescription(string) | No | No | No | No | Sets the items description |
| Void setName(string) | No | No | No | No | Sets the name of the item |
| String getName() | No | No | No | No | Gets the name of the item |
| String getDescription() | No | No | No | No | Gets the description of the item |
| Bool operator==(item,item) | No | No | Yes | No | Checks if items are the same |

INTERACTIVE CLASS
Abstract class: No
Subclass of: N/A
Composed of: N/A
Alterations post proposal:

- None

| DATA MEMBERS - Interactive | | | |
|---|---|---|---|
| **Variable Name** | **Data Type** | **Static** | **Description** |
| Name | string | no | Holds the name of the interactive |
| Description | String | No | Holds the description of the interactive |

| **MEMBER FUNCTIONS - Interactive** | | | | | |
|---|---|---|---|---|---|
| **Signature** | **Static** | **Virtual** | **Operator** | **Friend** | **Description** |
| Interactive(string,string) | No | No | No | No | Constructor |
| String getName() | No | No | No | No | Returns the interactives name |
| String getDescription() | No | No | No | No | Returns the interactives description |
| Void setName(string) | No | No | No | No | Sets the interactives name |
| Void setDescription(string) | No | No | No | No | Sets the interactives description |

BASEMENT CLASS
Abstract class: No
Subclass of: Room
Composed of: N/A
Alterations post proposal:

- None

| **DATA MEMBERS - Basement** | | | |
|---|---|---|---|
| **Variable Name** | **Data Type** | **Static** | **Description** |
| Lit_description | string | no | Holds the lit description of the room, when a player has the flashlight |
| Dark_description | String | No | Holds the dark description of the room, when a player does not have the flash light |

| **MEMBER FUNCTIONS - Basement** | | | | | |
|---|---|---|---|---|---|
| **Signature** | **Static** | **Virtual** | **Operator** | **Friend** | **Description** |
| Void readDescription() | No | No | No | No | Uses helper class to get descriptions from file |
| Void | No | No | No | No | Updates the room description |

| | | | | | |
|---|---|---|---|---|---|
| updateDescription(Player&) | | | | | based on player flags |
| Void setDescription(Player&) | No | No | No | No | Sets the room description variable |
| String interactWith(string,string,Player&) | No | No | No | No | Allows for interaction with an interactive in the room |
| String openLockBox(Player&) | No | No | No | No | Interacts with the lockbox interactive object |

BEDROOM CLASS
Abstract class: No
Subclass of: N/A
Composed of: N/A
Alterations post proposal:

- None

| DATA MEMBERS - Bedroom | | | |
|---|---|---|---|
| **Variable Name** | **Data Type** | **Static** | **Description** |
| Room_description | string | no | Holds the room description string |

| MEMBER FUNCTIONS - Bedroom | | | | | |
|---|---|---|---|---|---|
| **Signature** | **Static** | **Virtual** | **Operator** | **Friend** | **Description** |
| Void readDescription() | No | No | No | No | Uses helper class to get descriptions from file |
| Void setDescription() | No | No | No | No | Sets the room description variable |
| String interactWith(string,string,Player&) | No | No | No | No | Allows for interaction with an interactive in the room |
| String checkDiary(Player&) | No | No | No | No | Allows interaction with the diary interactive object in the room |

KITCHEN CLASS
Abstract class: No
Subclass of: Room
Composed of: N/A
Alterations post proposal:

- None

| DATA MEMBERS - Kitchen | | | |
|---|---|---|---|
| **Variable Name** | **Data Type** | **Static** | **Description** |
| Room_description | string | no | Holds the room description string |

| MEMBER FUNCTIONS - Kitchen | | | | | |
|---|---|---|---|---|---|
| **Signature** | **Static** | **Virtual** | **Operator** | **Friend** | **Description** |
| Void readDescription() | No | No | No | No | Uses helper class to get descriptions from file |
| Void updateDescription(Player&) | No | No | No | No | Updates the room description based on player flags |
| Void setDescription(Player&) | No | No | No | No | Sets the room description variable |
| String interactWith(string,string,Player&) | No | No | No | No | Allows for interaction with an interactive in the room |
| String getFlashLight(Player&) | No | No | No | No | Allows for interaction with the diary |

LIBRARY CLASS
Abstract class: No
Subclass of: Room
Composed of: N/A

Alterations post proposal:

- None

| DATA MEMBERS - Library | | | |
|---|---|---|---|
| **Variable Name** | **Data Type** | **Static** | **Description** |
| Door_close_desc | string | no | Holds the description for when the basement door is closed |
| Door_open_desc | String | No | Holds the description for when the basement door is open |
| ... | | | |
| ... | | | |

| MEMBER FUNCTIONS - Library | | | | | |
|---|---|---|---|---|---|
| **Signature** | **Static** | **Virtual** | **Operator** | **Friend** | **Description** |
| Void readDescription() | No | No | No | No | Uses helper class to get descriptions from file |
| Void updateDescription(Player&) | No | No | No | No | Updates the room description based on player flags |
| Void setDescription(Player &) | No | No | No | No | Sets the room description variable |
| String interactWith(string,string,Player&) | No | No | No | No | Allows for interaction with an interactive in the room |
| String checkBookShelf(Player&) | No | No | No | No | Allows for interaction with the bookshelf object |

LIVING_ROOM CLASS
Abstract class: No
Subclass of: Room
Composed of: N/A

Alterations post proposal:

- None

| DATA MEMBERS – Living_Room | | | |
|---|---|---|---|
| **Variable Name** | **Data Type** | **Static** | **Description** |
| name | string | no | Holds the first and last name of a player. |
| … | | | |
| … | | | |
| … | | | |

| MEMBER FUNCTIONS – Living_Room | | | | | |
|---|---|---|---|---|---|
| **Signature** | **Static** | **Virtual** | **Operator** | **Friend** | **Description** |
| Void readDescription() | No | No | No | No | Uses helper class to get descriptions from file |
| Void updateDescription(Player&) | No | No | No | No | Updates the room description based on player flags |
| Void setDescription(Player&) | No | No | No | No | Sets the room description variable |
| String interactWith(string,string,Player&) | No | No | No | No | Allows for interaction with an interactive in the room |

OFFICE CLASS
Abstract class: No
Subclass of: Room
Composed of: N/A
Alterations post proposal:

- None

| DATA MEMBERS - Office | | | |
|---|---|---|---|
| **Variable Name** | **Data Type** | **Static** | **Description** |
| Room_description | string | no | Holds the local room description |

| MEMBER FUNCTIONS - Office | | | | | |
|---|---|---|---|---|---|
| **Signature** | **Static** | **Virtual** | **Operator** | **Friend** | **Description** |
| Void readDescription() | No | No | No | No | Uses helper class to get descriptions from file |
| Void updateDescription(Player&) | No | No | No | No | Updates the room description based on player flags |
| Void setDescription(Player&) | No | No | No | No | Sets the room description variable |
| String interactWith(string,string,Player&) | No | No | No | No | Allows for interaction with an interactive in the room |
| String getHammer(Player&) | No | No | No | No | Allows for interaction with the drawer object in the room |

ENTRANCE CLASS
Abstract class: No
Subclass of: Room
Composed of: N/A
Alterations post proposal:

- None

| DATA MEMBERS - Entrance | | | |
|---|---|---|---|
| **Variable Name** | **Data Type** | **Static** | **Description** |
| beast_description | string | no | Holds the description for when the beast is present in the room |
| No_beast_description | String | No | Holds the description for when the best is not present |

| | | | |
|---|---|---|---|
| | | | in the room |

| MEMBER FUNCTIONS - Entrance | | | | | |
|---|---|---|---|---|---|
| **Signature** | **Static** | **Virtual** | **Operator** | **Friend** | **Description** |
| Void readDescription() | No | No | No | No | Uses helper class to get descriptions from file |
| Void updateDescription(Player&) | No | No | No | No | Updates the room description based on player flags |
| Void setDescription(Player&) | No | No | No | No | Sets the room description variable |
| String interactWith(string,string,Player&) | No | No | No | No | Allows for interaction with an interactive in the room |
| String openDoor(Player&) | No | No | No | No | Allows for interaction with the door interactive object in the room |
| String checkDrawer(Player&) | No | No | No | No | Allows for interaction with the drawer interactive object in the room |

HELPER CLASS
Abstract class: No
Subclass of: N/A
Composed of: N/A
Alterations post proposal:

- All of these functions are either new or moved from the Game class as I was having issues creating my static functions there because of circular inclusions causing compilation issues

| DATA MEMBERS - Helper | | | |
|---|---|---|---|
| **Variable Name** | **Data Type** | **Static** | **Description** |
| n/a | n/a | n/a | n/a |

| **MEMBER FUNCTIONS - Helper** | | | | | |
|---|---|---|---|---|---|
| **Signature** | **Static** | **Virtual** | **Operator** | **Friend** | **Description** |
| Vector<string> explode(string&,char&) | Yes | No | No | No | Turns a string into a vector delimited by the character parameter |
| String readFile(string) | Yes | No | No | No | Reads text from a file |
| Bool writeFile(string,string) | Yes | No | No | No | Writes the second string parameter to a file with the file_path parameter |
| Void writeConsole(string) | Yes | No | No | No | Writes text to the console |
| Item buildItem(string) | Yes | No | No | No | Builds and item from the item_name given |
| String Strtolower(string) | Yes | No | No | No | Converts a string to lowercase |
| String readConsoleNoSani() | Yes | No | No | No | Reads the console but does not sanitize the input |

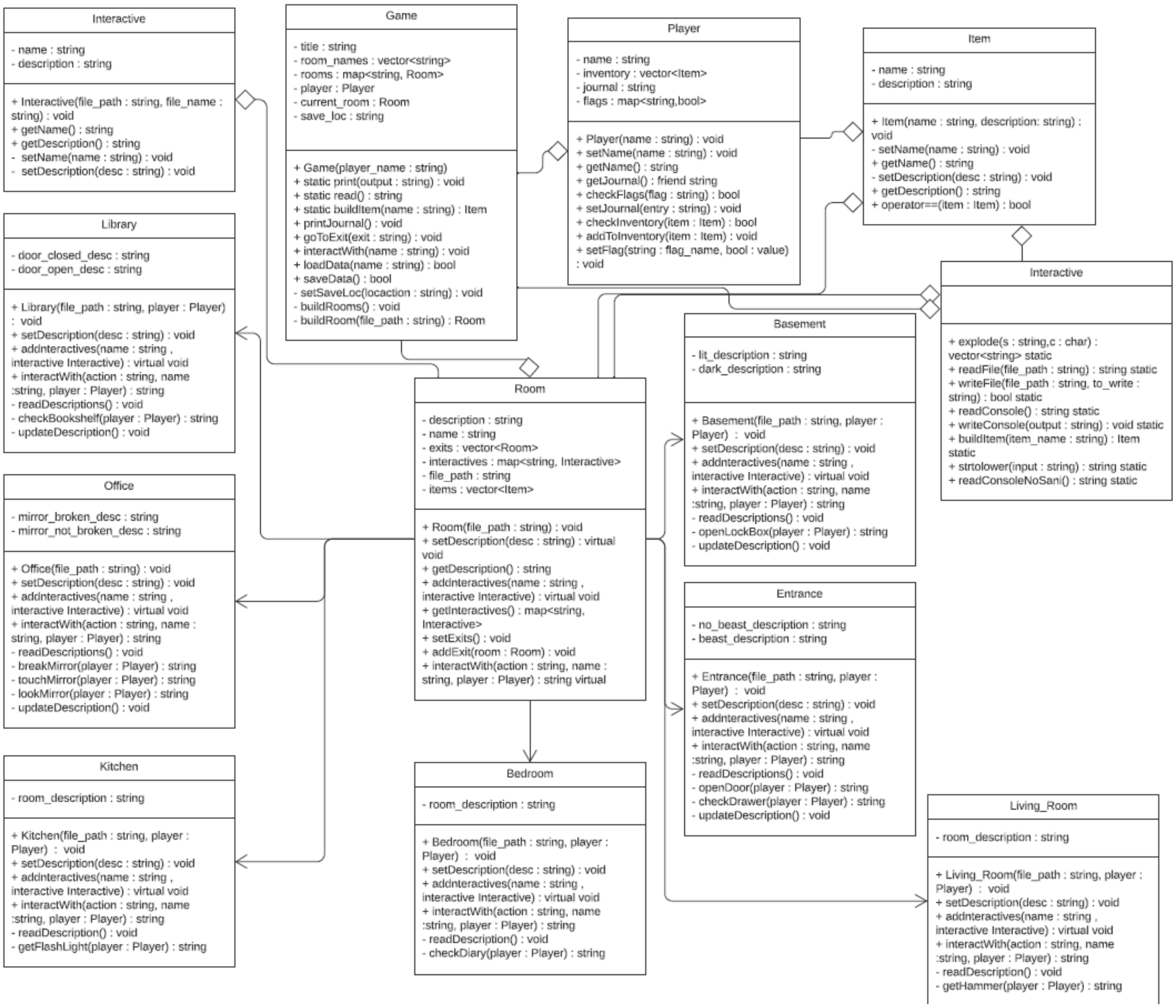
## DEMONSTRATION OF OOP CONCEPTS

| Demonstrated in... | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| File name | Line #s | **Encapsulation** | **Inheritance** | **Polymorphism** | **Static Members** | **Friend Functions** | **Overloaded Operators** | **Text files** |
| Player.h | 11-15 | X | | | | | | |
| Player.h | 34-35 | | | | | X | | |
| Bedroom.h, Entrance.h, Kitchen.h, Living_Room.h, Office.h, Library.h, Basement.h | 7 | | X | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Room.h | 35 | | | X | | | | |
| Helper.h | 15-80 | | | | X | | | |
| Helper.h | 29-55 | | | | | | | X |
| Item.h | 20 | | | | | | X | |

REQUIRED: Click to complete the Teammate Assessment Form.

## UML DIAGRAM


Interactive
- name : string
- description : string
+ Interactive(file_path : string, file_name : string) : void
+ getName() : string
+ getDescription() : string
- setName(name : string) : void
- setDescription(desc : string) : void
Game
- title : string
- room_names : vector<string>
- rooms : map<string, Room>
- player : Player
- current_room : Room
- save_loc : string
+ Game(player_name : string)
+ static print(output : string) : void
+ static read() : string
+ static buildItem(name : string) : Item
+ printJournal() : void
+ goToExit(exit : string) : void
+ interactWith(name : string) : void
+ loadData(name : string) : bool
+ saveData() : bool
- setSaveLoc(locaction : string) : void
- buildRooms() : void
- buildRoom(file_path : string) : Room
Player
- name : string
- inventory : vector<Item>
- journal : string
- flags : map<string,bool>
+ Player(name : string) : void
+ setName(name : string) : void
+ getName() : string
+ getJournal() : friend string
+ checkFlags(flag : string) : bool
+ setJournal(entry : string) : void
+ checkInventory(item : Item) : bool
+ addToInventory(item : Item) : void
+ setFlag(string : flag_name, bool : value) : void
Item
- name : string
- description : string
+ Item(name : string, description: string) : void
- setName(name : string) : void
+ getName() : string
- setDescription(desc : string) : void
+ getDescription() : string
+ operator==(item : Item) : bool
Library
- door_closed_desc : string
- door_open_desc : string
+ Library(file_path : string, player : Player) : void
+ setDescription(desc : string) : void
+ addnteractives(name : string , interactive Interactive) : virtual void
+ interactWith(action : string, name :string, player : Player) : string
- readDescriptions() : void
- checkBookshelf(player : Player) : string
- updateDescription() : void
Interactive
+ explode(s : string,c : char) : vector<string> static
+ readFile(file_path : string) : string static
+ writeFile(file_path : string, to_write : string) : bool static
+ readConsole() : string static
+ writeConsole(output : string) : void static
+ buildItem(item_name : string) : Item static
+ strtolower(input : string) : string static
+ readConsoleNoSani() : string static
Basement
- lit_description : string
- dark_description : string
+ Basement(file_path : string, player : Player) : void
+ setDescription(desc : string) : void
+ addnteractives(name : string , interactive Interactive) : virtual void
+ interactWith(action : string, name :string, player : Player) : string
- readDescriptions() : void
- openLockBox(player : Player) : string
- updateDescription() : void
Room
- description : string
- name : string
- exits : vector<Room>
- interactives : map<string, Interactive>
- file_path : string
- items : vector<Item>
+ Room(file_path : string) : void
+ setDescription(desc : string) : virtual void
+ getDescription() : string
+ addnteractives(name : string , interactive Interactive) : virtual void
+ getInteractives() : map<string, Interactive>
+ setExits() : void
+ addExit(room : Room) : void
+ interactWith(action : string, name : string, player : Player) : string virtual
Office
- mirror_broken_desc : string
- mirror_not_broken_desc : string
+ Office(file_path : string) : void
+ setDescription(desc : string) : void
+ addnteractives(name : string , interactive Interactive) : virtual void
+ interactWith(action : string, name : string, player : Player) : string
- readDescriptions() : void
- breakMirror(player : Player) : string
- touchMirror(player : Player) : string
- lookMirror(player : Player) : string
- updateDescription() : void
Entrance
- no_beast_description : string
- beast_description : string
+ Entrance(file_path : string, player : Player) : void
+ setDescription(desc : string) : void
+ addnteractives(name : string , interactive Interactive) : virtual void
+ interactWith(action : string, name :string, player : Player) : string
- readDescriptions() : void
- openDoor(player : Player) : string
- checkDrawer(player : Player) : string
- updateDescription() : void
Kitchen
- room_description : string
+ Kitchen(file_path : string, player : Player) : void
+ setDescription(desc : string) : void
+ addnteractives(name : string , interactive Interactive) : virtual void
+ interactWith(action : string, name :string, player : Player) : string
- readDescription() : void
- getFlashLight(player : Player) : string
Bedroom
- room_description : string
+ Bedroom(file_path : string, player : Player) : void
+ setDescription(desc : string) : void
+ addnteractives(name : string , interactive Interactive) : virtual void
+ interactWith(action : string, name :string, player : Player) : string
- readDescription() : void
- checkDiary(player : Player) : string
Living_Room
- room_description : string
+ Living_Room(file_path : string, player : Player) : void
+ setDescription(desc : string) : void
+ addnteractives(name : string , interactive Interactive) : virtual void
+ interactWith(action : string, name :string, player : Player) : string
- readDescription() : void
- getHammer(player : Player) : string