

Technical Rundown

Game and Player Relations

The engine will only run whatever room the player is in. However, certain commands will allow for conditions to be changed from a room linked to the current room (i.e. 'lock' will lock the exit linked to the current one). These changed conditions will not set off any triggers within the linked room. Before a player leaves a room, the engine saves the room to the .json file it is associated with.

Building Rooms

All of the information displayed to the user is located in room files (.room files within the Rooms folder of the program). **It is imperative that the all room files use the predetermined template** (see next pages for templates). Should there be an error with loading a room file, there is an error handling program that displays a warning and offers a tutorial. However, after the game will crash because the engine cannot load an invalid room. An alternative to manually filling out the predetermined template would be to use the general functions within the engine's API that allow rooms to be built in python [i.e. *Demo_Adventure.py*].

Notes:

- Every set must have curly brackets surrounding the items within it. Every list must have brackets around it
- Every element in the set, except the last item, needs to have a comma at the end of the line
 - Example:

```
{
    Item_1,      #comma needed because item_2 is also in the set
    Item_2      #no comma needed because item_2 is the last element in the set
}
```
- Every Room file is stored in the Rooms folder, thus the syntax for the directory will begin with `"/Rooms/"`
- Be wary about neglecting to illuminate things, if a room is not illuminated and the player has no items that are illuminated, the player will not be able to see in the room.
- Be wary of which exits are blocked, you could block a player from entering a room on accident
- Room exits can be one sided (i.e. the bedroom exits to the hallway, but the hallway does not exit into the bedroom)
- Spacing is important. Make sure to follow the template's spacing format and use the example files as references
- Sets and lists can be empty,
 - Example:

```
"exits": []      #if there are no exits in the room
"alias": []      #if there are no aliases associated with the object
```

****The engine does use the item inventory yet****

Room Template

<pre>{ "room_name": String, "room_file": "./Rooms/" + String, "description": String, "illuminated": Boolean, "exits": List of Exit, "inventory": List of Item, "triggers": List of Trigger, "user-scripts": List of UserScript }</pre>	<pre>#the room name displayed to the players #current room file's directory, String = the file's saved name #the flavor text to the 'look' command and when first entered #true or false on whether there is enough light to see in the room without items #list of all exits located inside the room. Exit template is below #list of all items in the room that aren't on the player. Item template is on the next page #list of all triggers in the room. Trigger template is on the next page #list of all user scripts in the room. UserScript template is on the next page</pre>
--	--

Exit Template

<pre>String: { "links-to": "./Rooms/" + String, "description": String, "blocked": Boolean, "door": Door Object, "triggers": List of Trigger, "user-scripts": List of UserScript }</pre>	<pre>#exit name that will be used in exit checks #room file's directory for the room the exit leads to, String = the file's saved name #the flavor text to the 'look' command #true or false on if the player can move through the exit #every exit can have at most 1 door. Door Template is below #list of all triggers specific to the exit. Trigger template is on the next page #list of all user scripts specific to the exit. UserScript template is on the next page</pre>
---	--

Door Template

<pre>"open": Boolean, "lock": Lock Object, "triggers": List of Trigger, "user-scripts": List of UserScript</pre>	<pre>#true or false that the door is open #a door can have at most 1 lock. Lock template is on the next page #list of all triggers specific to the door. Trigger template is on the next page #list of all user scripts specific to the door. UserScript template is on the next page</pre>
--	---

"locked": Boolean,
"key": String,
"triggers": List of Triggers,
"user-scripts": List UserScripts

Lock Template

#true or false on whether the lock is locked
#name of the key that opens the door associated to the specific lock
#list of all triggers specific to the lock. It uses the template below
#list of all user scripts specific to the lock. Use the template below

Item Template

```
{  
  "item_name": String,  
  "description": String,  
  "alias": List of Strings,  
  "quantity": Integer,  
  "visible": Boolean,  
  "illuminated": Boolean,  
  "obtainable": Boolean,  
  "inventory": List of Item,  
  "door": Door Object,  
  "triggers": List of Trigger,  
  "User-scripts": List of UserScript  
}
```

#item name that will be used in item checks
#the flavor text to the 'look' command
#list of item's nicknames
#Number of items in the room
#true or false the player can see it
#true or false on if the player can use the object as a light source
#true or false on if the player can obtain the item
#list of all the items within the item
#every item can have at most 1 door. Door Template is on the previous page
#list of all triggers specific to the item. It uses the template below
#list of all user scripts specific to the item. Use the template below

Trigger Template

```
{  
  "trigger_command": String,  
  "connected_to": Object,  
  "type": String,  
  "description": String  
}
```

#command that causes the trigger
#if the trigger is associated with another object (lock, door, etc)
#where the string change is happening, so far "print" is the only command
#flavor text for the triggered command

UserScript Template

{	
"trigger_command": String,	#the primary command that causes the trigger to fire
"before": String,	#python code that will initially execute
"instead": String,	#python code that will execute and override the usual command
"after": String	#python code that will execute after the usual/overridden command
}	

Player Template

{	
"name": String,	#name of the character
"inventory": List of Items,	#list of all the items within the players inventory
"description": String,	#the flavor text to the 'look' command
"character_file": "./Players/" + String,	#the player file location within the 'Players' directory
"current_room_file": "./Rooms/" + String,	#the room file location within the 'Rooms' directory
"alive": Boolean	#whether the player is alive or not
}	

Player Rundown

The engine removes multiple white spaces and uppercase letters.

Examples:

- “go north” = “go north”
- “GO NORth” = “go north”

The engine also allows for fluff in between the command, so long as none of the fluff contains another keyword. This means the sentence cannot have more than one action or object in it. Furthermore, the object receiving the action must be to the right of the action [exception being the command ‘look’]. But there is a special case with regards to the word ‘lock’: the commands ‘look’, ‘lock’, and ‘unlock’ can be used with relation to the word ‘lock’, when in reference to an exit’s lock. Lastly, the words “door” and “lock” can hold meaning depending on the situation.

Valid Commands:

- “Fluff go north” = “go north”
- “go fluff north” = “go north”
- “go north fluff” = “go north”
- “look” = “look”
- “look north lock” = *looks at the north exit’s lock*

Invalid Commands:

- “north go” [There command comes before the object receiving the command]
- “look go north” [There are too many actions in the sentence]
- “go north west” [There are too many objects in the sentence]
- “open lock” [Locks cannot be opened]

****The engine does use the item inventory yet****

Commands	Shortcuts	*Used on* = *shortcuts*	Anticipated Player Error Catches	Condition Expectations	Known Difficulties
go	g travel walk run enter move	north = n, northern south = s, southern east = e, eastern west = w, western *valid exit*	<ul style="list-style-type: none"> When an exit does not exist When an exit is blocked or if a door is locked, closed, or both When the other side of the exit is locked, blocked, or closed 	<p>All exits in the room file should replace spaces with underscores.</p> <p>All room links should lead to a valid room</p>	It allows for players to teleport through a room that doesn't have a door
look	l look around examine	north = n, northern, east south = s, southern, south east = e, eastern, north west = w, western, west *valid exit* *valid exit* door *valid exit* lock *valid item* *valid item* door	<ul style="list-style-type: none"> If there is no exit or lock to look at If there is no description written If a player is trying to look at locks or doors (descriptions don't exist yet) If no light is be projected in the room (from either items or the room itself) When the item attempting to be looked at is not visible 		If a player tries to look at anything in the item inventory

		valid item lock			
unlock		*valid exit* *valid exit* door *valid exit* lock *valid item* *valid item* door *valid item* lock	<ul style="list-style-type: none"> • If there is no lock • If there is no door to unlock • If the exit is invalid • If the key is not in the player inventory 	All room links should lead to a valid room	If a player tries to unlock at anything in the item inventory
lock		*valid exit* *valid exit* door *valid exit* lock *valid item* *valid item* door *valid item* lock	<ul style="list-style-type: none"> • If there is no lock • If there is no door to unlock • If the exit or item is invalid • If the key is not in the player inventory 	All room links should lead to a valid room	If a player tries to lock at anything in the item inventory
open		*valid exit* *valid exit* door	<ul style="list-style-type: none"> • If there is no door to unlock • If the exit or item is invalid • If the door is locked closed 	All room links should lead to a valid room	If a player tries to open anything in the item inventory

		valid item *valid item* door	<ul style="list-style-type: none"> • If the door was already opened 		
close		*valid exit* *valid exit* door *valid item* *valid item* door	<ul style="list-style-type: none"> • If there is no door to unlock • If the exit or item is invalid • If the door is locked open • If the door was already closed 	All room links should lead to a valid room	If a player tries to close anything in the item inventory
block		*valid exit*	<ul style="list-style-type: none"> • If the exit is invalid • If the exit was already blocked 		You can block an exit by simply typing 'block *valid compass direction*'
unblock		*valid exit*	<ul style="list-style-type: none"> • If the exit is invalid • If the exit was already unblocked 		You can unblock an exit by simply typing 'unblock *valid compass direction*'
inventory	i		displays the amount of each item in the inventory		
get	pick up take	*valid item*	Deletes the item in the room if it is the last item Increments item quantity if exists Doesn't pick up unobtainable items		If a player tries to pick up anything in the item inventory
drop		*valid item*	Drops unobtainable items Deletes the item in the room if it is the last item Increments item quantity if exists		If a player tries to drop anything in the item inventory

