Text-Adventure Game Engine

This is the planning of the text adventure game engine. Here will be listed how the classes will work and interact with each other.

Game Objects:

Most components of the game will be extending this class, which will be passed as a parameter for commands.

The class shall contain the basic information about the entity such as:

Unique Identifier, like a name of some sort,

Maybe a description, but that might be delegated to a json file

Types of game objects:

There will be x types of game entities that the user can create in order to make a game, those being:

Player character

This will be the protagonist of the story, the player will act and experience the story through the lens of this character.

Room

Rooms will be... well, rooms. But not just rooms, it can be places outside, landscapes, etc. These will be the places where the player can go, and where things will be and happen. Imagine it as a scenario in a story.

Object

Objects will be inanimate objects, usually part of the scenario, such as a table, a television, a rug, etc. Usually things the player can interact with but not pick up and take it with them.

Item

Items will be things the player can pick up, like a cellphone, a crank, a bottle, etc. Usually things the player will take with them to use later somewhere else.

Non-player character(NPC)

These will be characters that are not controlled by the player, usually used to add to the story or help the player in some way.

Commands

These will be the main form which the player will interact with the world. Each command will be a word, usually an imperative verb such as “look”, or “take”, and possibly a target. These commands will realize functions that will affect the world and/or the story. They represent the actions the player character can do.

How the objects will interact with each other:

The player character will be in a room, which may have objects, NPCs, items and directions for a next room. Items may be picked up and put in the player’s inventory.

The player will have a list of simple preset commands they can do, such as “look”, “take” etc. However, all game objects will be able to enable specific commands that the player can do, for example, there might be a climbable tree Object in a Room the player is in, which will make the “climb” command possible.

How the engine will work:

As a text adventure game, there’s no need for a very complex engine, in fact, most of the gameplay will be conveyed through the descriptions of the game objects and commands the player does. Therefore, the engine will mostly be responsible for managing these objects and the interactions between them. It’s also possible to keep track of successful or attempted commands the player does in order to convey a measure of time if so needed (for example “in 5 actions the player takes from now, it will start raining”.

For ease of writing and editing, the texts shall be stored in a json file, which may follow the template:

{

“player” : {

“onFullHealth”: “You feel pretty well, despite the circumstances”,

“onLowHealth”: “You can feel death’s hand reaching out for you”;

}

}

Class definitions

Room

Rooms will have the following attributes:

connectedRooms (Map<String, Room>)

This will be the map that will link rooms and the directions they are

objectList (List<GameObject>)

This will be a list of the objects that are in the room

itemList(List<GameItem>)

This will be a list of the items that are in the room

description(String)

This will be the description of the room when the player enters it or when he inspects the room again