**Zuul Project**

Final Custom Zuul Game Requirements

Changes to implementation:

* Make changes to the code as described in the book. You must read the chapter and follow the exercises by making changes to the code.

Commands:

* go: Used to go to the another room by appending the exit string. example: go north
* look: Used to get a description of the current room including exits and items.
* take: Used to take an item from the current room. example: take book
* drop: Used to put an item from the players inventory into the current room.
* inventory: Used to list the items the player is carrying.
* use (optional): Used to use an item, depending on the requirements of the game.
* back: Used to go to the previous room.

New Classes:

* **Item**: Items are put in rooms. A player can take and drop items.
* **Player**: A player is used to store the player’s current location, the previous room, and his/her inventory of items. A player should be allowed to hold more than one item.
* **Person**: This optional class can be used to place persons in the room with whom to interact. Persons might give you objects, talk to you, etc. You may need to implement more commands, like “talk” or “give”. Rooms should include a list of perso ns in their long descriptions.

**Final Game Rubric**

|  |  |
| --- | --- |
| 90-100 | Excellent game shows attention to detail and thoughtful design and creative extensions. All requirements were implemented with appropriate coupling and cohesion. Person class has been added and incorporated in a meaningful way. Formatting meets convention. |
| 80-89 | Basic requirements were met. The game has correctly implemented Room, Item, and Player classes. Room/Item objects have full descriptions and appropriate fields. Some features were missing or needed more attention. |
| 70-79 | Some requirements not met or lacking several required features or poorly implemented. |
| below 70 | Lack of effort is apparent. |