Design Process

Made multiple lists of items which implement certain interfaces such as consumables to easily access lists without having to do type checking over and over again at the cost of space, however since there are not many items I thought the tradeoff would be worth it. Made the lists items rather than a list of consumables for example as it allows lists to be more generic and can be easily used in my item search field. Also means we can use functions from items (all things in inventory that implement interfaces are items) and not just the interface. All recipes contain the simple name of the class that they are so when they are crafting it is easier to build. Kept all constructors for subclasses of items the same as it allowed me to use a general get Constructor method when crafting. A field called keyNum was used in the key class to match the corresponding door since it was the easiest way to make sure if Gson initialized a key before a door did not mess up anything this would have been a problem with our previous implementation (Storing the id of the door it is connected to).

Planning

Tests

Character tests [Jeremy]
Moving entity tests [Tommy]
Unit tests- Persistence of games [Bejai]
Unit tests- Inventory [Jeremy]
System Tests -Item [Jeremyl]
Tests for Static entities [Brendan]
Various post submission tests [Brendan]
System Tests - Goals [Raphael]

Code

Implement - Character [Jeremy]
Implement - Inventory [Tommy]
Implement - Colliders [Tommy]
Implement - Persistence Json [Bejai]
Implement - Goals [Raphael]
Implement - Static entities + static entity subclasses [Brendan]
Implement - A way to convert items on the ground to inventory items [Brendan]
Implement - Move Interface [Tommy]
Implement - Item [Jeremy]
Add mercenaries to character [Jeremy]
Research + restructure code for Gson [Bejai]

Documentation

UML diagram [Group]
Post submission UML cleanup [Brendan]
Minutes [Brendan]
Planning [Group]
Assumptions [Group]