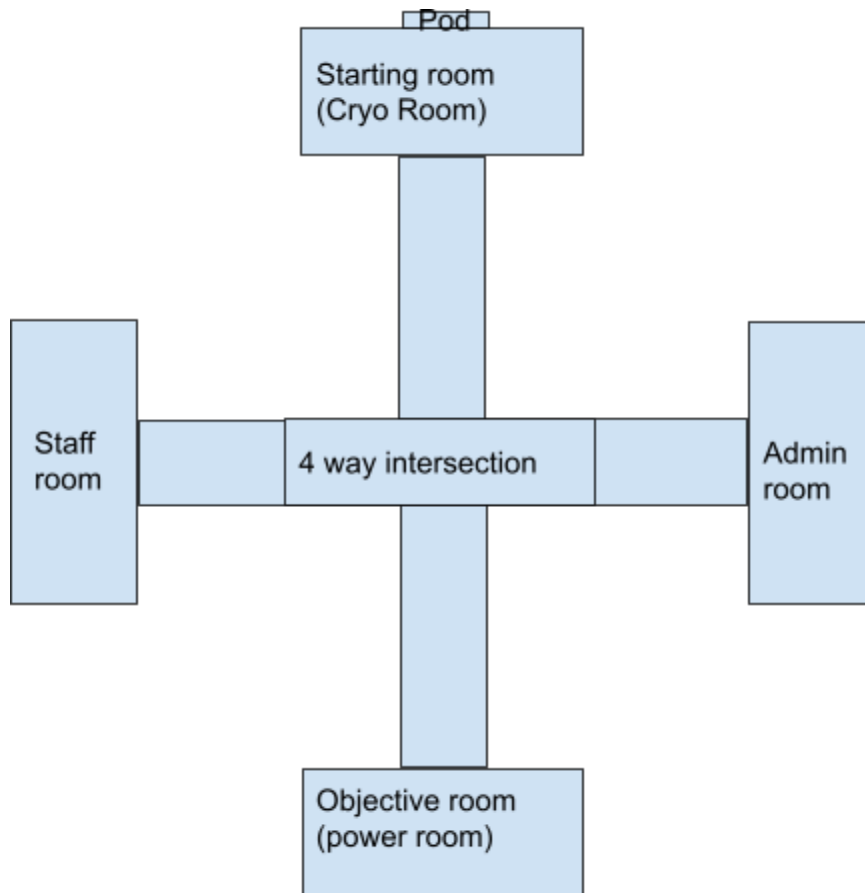


## Criterion B: Design

### Criterion B- Design Overview

This program, as it will be a relatively simple game, will have a simple layout. The game will take place in a spaceship and the player will be a crewmember awoken from cryosleep to see the power of the ship to be off. The objective of the game is to get the power back on. Below is an outline of the map of the game.



The program will be coded so that the player can freely roam from one room to another, certain rooms will be locked until the player finds an item or enters a password, such as the power and admin rooms.

Each room will have these inputs:

Player Inputs	Program Outputs
Moveforward, Movebackward, Moveleft, Moveright	Moves the player around or out of a room. Provides output to where they are going and what they are doing
observe	Gives a description of the player's location

	and what the output will be of the Moveforward,Movebackward,Moveleft, and Moveright inputs
inventory	Outputs a list of the items in the players inventory

Any other inputs will be specified by the program when they can be used. Such as the enter password command to open the door to Admin

These features will be easy and simple in the Java program as they only involve simple devices such as scanners, print lines, and variables to work.

## Rooms:

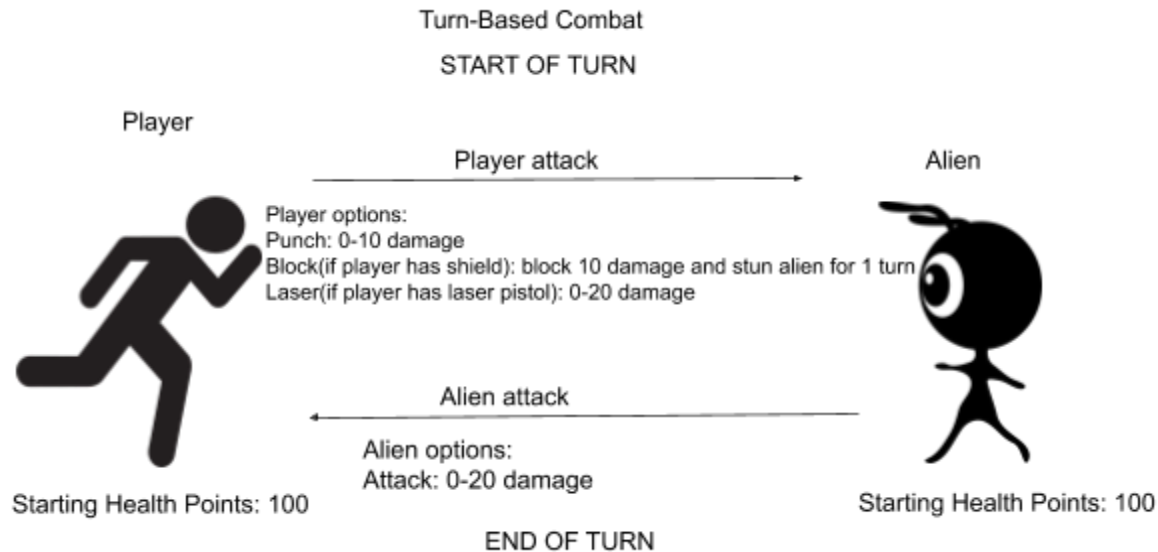
Below is the content of all the rooms

ROOMS	CONTENT IN ROOMS
Cryogenic room (starting and ending room)	Cryo pod(ending), escape pod (ending), exit to 4 way intersection
Staff room	Paper with admin password(Admin is locked unless this password is used to open it), Laser pistol, yellow keycard (for safe in admin)
Admin room	(admin is locked unless the password is used) Shield, Safe (contains a lock hacker for power room)
Power room (objective room)	(Locked unless hack locker is used) Alien(combat), power generator (objective)
4 way intersection	Staff room, cryogenic room, Admin room, Power room

Each room's content will have different inputs depending on the direction and where the player came from. These outcomes will be in the form of methods in Java for better understanding and organization of the code.

## Combat:

The game will contain a combat system with an alien. A diagram of the system is below.



Combat system continues until the alien or player reaches 0 Health Points. This combat system is very simple in Java and will only need a few variables, if statements, and a loop.

### Items:

Password note	Note that contains the password to admin
Yellow keycard	Keycard that will unlock the safe in admin
Lock hacker device	Device found in the safe in admin, is used to unlock the power room lock
Laser pistol	Used in the combat with alien. Does 0-20 damage
Shield	Used in the combat with alien. Blocks 10 damage and stuns

These items, when picked up, will be stored in an array and the player can read what is in that array with the command “inventory”. As each variable is added to the inventory, it disappears from the map, this can be done with if statements that provide two outcomes of the item there and the item gone.

### Endings:

Good ending	Player fixes the generator and goes back to their pod
Good , but strange ending	Player fixes the generator, but escapes

	through the escape pod
Bad ending	Player fails to fix generator and escapes through generator
Bad ending 2	Player is defeated by alien

Each ending depends on certain choices in the game and will only need a few boolean variables and if statements in java for it to work.

## Test Plan:

Action	Way of testing and result
Test if the program starts and runs correctly	Enter 'java ia.java' in command line in the directory of the program. Game launches and gives output.
Check if commands run and work	Enter the commands "Moveforward,MoveBackward,Moveleft,Moveright,inventory,observe" in every room in the game. Game will provide output and interaction to each command.
Check if room methods work	Enter commands that will enter and exit rooms in the game. Output will be different based on what room is entered and exited.
Check if items work	Use a command to collect an item in game, try to collect it in the same place again, and use command 'inventory', test with multiple items. Game provides output to tell the player of the collection of the item. Game says "There is nothing here" when the player attempts to collect it again. Game shows all items in the inventory of the player.
Check if combat and combat items work	Go through combat in game multiple times with different items input "laser", "block", "punch". Game will respond accordingly if the correct item is present or not. Ex. game will output "invalid command" if the player inputs "laser" and they do not have the laser pistol item. Game combat will conclude with either the player or alien defeated.

Test if the endings work	Play through the game 4 times and input the correct inputs for each ending. Game will give different responses and endings for each of those 4 times
--------------------------	--

Word Count: 345