# Mission 17: Advanced Training

Start date: 27 October 2017
**Due: 5 November 2017, 23:59**

Readings:

- SICP: Chapter 3
- Lecture notes on Object-Oriented Programming

**IMPORTANT WARNING:**
Because we provide you with the flexibility in choosing the approach by which you solve the problems in this mission, we require that you submit well commented/annotated programs. Describe your approach to the programming questions, and then comment/annotate the relevant parts that implements your idea. If you fail to do so, you risk having marks deducted by your tutor.

---

## The Training Continues

### Guardians of the Generator: Training Brief by XO Kenneth

One of our Advance Scouts has been able to enter the Death Cube under disguise. She has been able to observe the service bots up close, and it would appear they bear no weapons, only tools for repair and construction. However, when she attempted to disable the service bots, armed security drones started streaming out of the walls of the room. Not only was the supply of drones continuous, they appeared able to spawn more quickly as time passed. Luckily, our Scout managed to escape, and is now recuperating.

Using this hard-won information, you should now train to defeat the potential enemies. Again, I must caution you against aimless wandering while searching for the service bots, and risking separation from your team by moving across too many rooms at once.

Our fleet fast approaches the Cube. I urge all of you to hasten your training. May the Force be with us all.

### Mission Start

This mission consists of **TWO** tasks.

**Unless requirements overlap, do not remove the requirements of past Tasks and Missions.**

## Task 1:

Instead of wandering around aimlessly, you will need to have a better method of finding `ServiceBots` and `SecurityDrones`. Every turn, you should

- remember the room you are in,
- attack any bots or drones in your room, and
- move towards an unvisited room if there is one, or a random one otherwise

Make appropriate changes to your personal Player subclass to simulate this behaviour. Mark your changes using comments (for example, `/* M17 T1 */`).

## Task 2:

The moment a ServiceBot is destroyed, an alarm will sound and a SecurityDrone will spawn at that location equipped with a `zapray`. Each turn, the security drone(s) will zap unauthorised people carrying cards in their rooms and pick up any Keycards lying on the floor. It is thus `very` important that you find the generator room quickly.

Since you are already memorising the rooms you visit, we can modify that to aid in finding the **generator room**. The generator room is a `ProtectedRoom` that contains a `Generator` object. Now, in each turn, you will

1. find out if the generator room is nearby (beside you), and either
   - remember its location if you don't have a `Keycard` or
   - enter the generator room if you have a `Keycard`. Also, you should
2. avoid entering any `ProtectedRooms` if you don't have a `Keycard`, and
3. once you have a `Keycard` and know where the generator room is, you should head to the generator room. You MUST with every move you make, make progress towards the generator room i.e. you should not utilise any scheme which involves moving around randomly - instead the path that you take should be deterministic and progressive in arriving at the generator room.

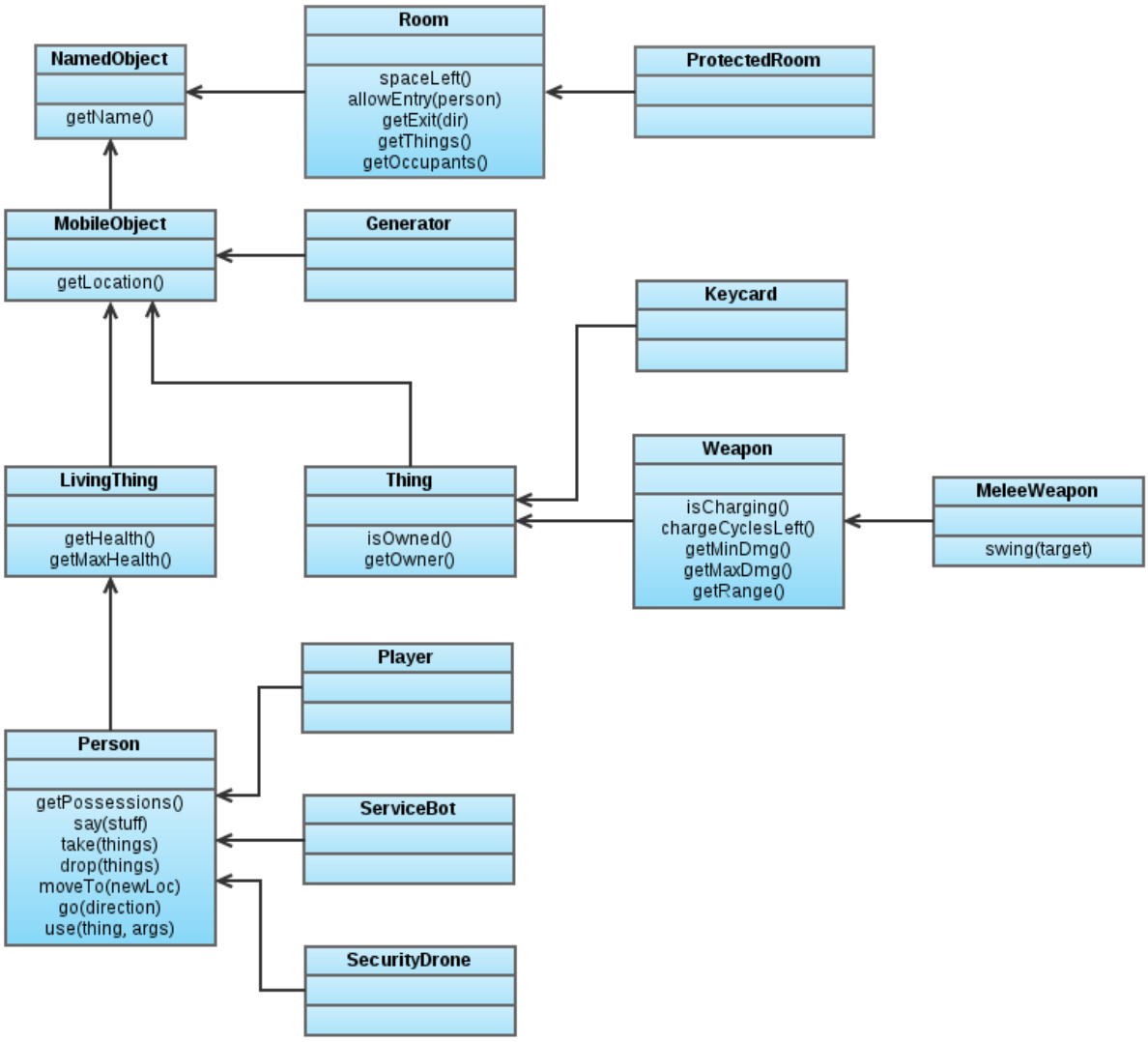
Take into consideration the possibility that you may be seriously injured and evacuated. When you are redeployed, it may be in an area that you have not explored before.

Make appropriate changes to your personal Player subclass to simulate this behaviour. Mark your changes using comments (for example, `/* M17 T2 */`).

## Submission

To submit your work to the Academy, place your solutions into the box that says "Program" on the mission page, click "Save", then click "Finalize Submission". Note that submission is final and that any mistakes in submission requires extra effort from a tutor or the lecturer himself to fix.

## Appendix: A simplified representation of our object hierachy is as follows:




Note that if you choose to explore the the simulation libraries, you will find many private methods and properties not intended for use in your solutions. These private methods and properties are indicated by being prefixed with two underbars (e.g. `LivingThing.prototype.__act`). You will be penalized for using any of these properties.

Certain methods return results of different types. Use `is_object` or `is_instance_of` appropriately for such cases.

## Appendix: Classes and Methods Reference

| **Room's Methods**: | | |
|---|---|---|
| `spaceLeft` | | Returns the remaining space in the room for occupants |
| `getThings` | | Retrieves a list of unowned things inside the room |
| `getOccupants` | | Retrieves a list of all living things inside the room |
| `getExit` | `dir:string` ("north", "south", "east", "west", "up", "down") | Retrieves the room in that direction, or `false` if there is none |
| `getExits` | | Retrieves the list of directions leading out of the room |
| `getNeighbours` | | Retrieves the list of rooms adjacent to the room |
| `allowEntry` | `person:Person` | Checks if `person` can enter |

| **LivingThing's Methods**: | |
|---|---|
| `getHealth` | Retrieves current Health Points |
| `getMaxHealth` | Retrieves maximum Health Points |

| **Person's Methods**: | | |
|---|---|---|
| `getPossessions` | | Retrieves list of it |
| `say` | `stuff:string` | Says stuff |
| `take` | `things:list of things` | Takes all specified |
| `drop` | `things:list of things` | Drops all specified |
| `use` | `thing:Thing, ...` | Uses the item see |
| `moveTo` | `newLoc:Room` | Moves to `newLoc` |
| `go` | `dir:string` ("north", "south", "east", "west", "up", "down") | Moves in that dire |

| **Weapon's Methods** | |
|---|---|
| `isCharging` | Check if a weapon is charging |
| `chargeCyclesLeft` | Check remaining charging time |
| `getMaxDmg` | Checks max damage |
| `getMinDmg` | Checks min damage |
| `getRange` | Checks weapon range |

| **Using Weapons**: | |
|---|---|
| `this.use(melee_weapon, list(target))` | Use `melee_weapon` to attack selected `target` in the same room |