

Mission 18: The Final Showdown

Start date: 30 October 2017

Due: 9 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 Final Preparation

Death to the Cube: Executive Officer Kenneth address the initiates of the Alliance

Tonight, all of us gather here to witness this moment in history. In a few days, we will launch an attack on Darth's mothership, the Death Cube. We have managed to produce enough weapons for all of you, and it is now up to you to learn these weapons within these last days. With all this might at our fingertips, we cannot fail!

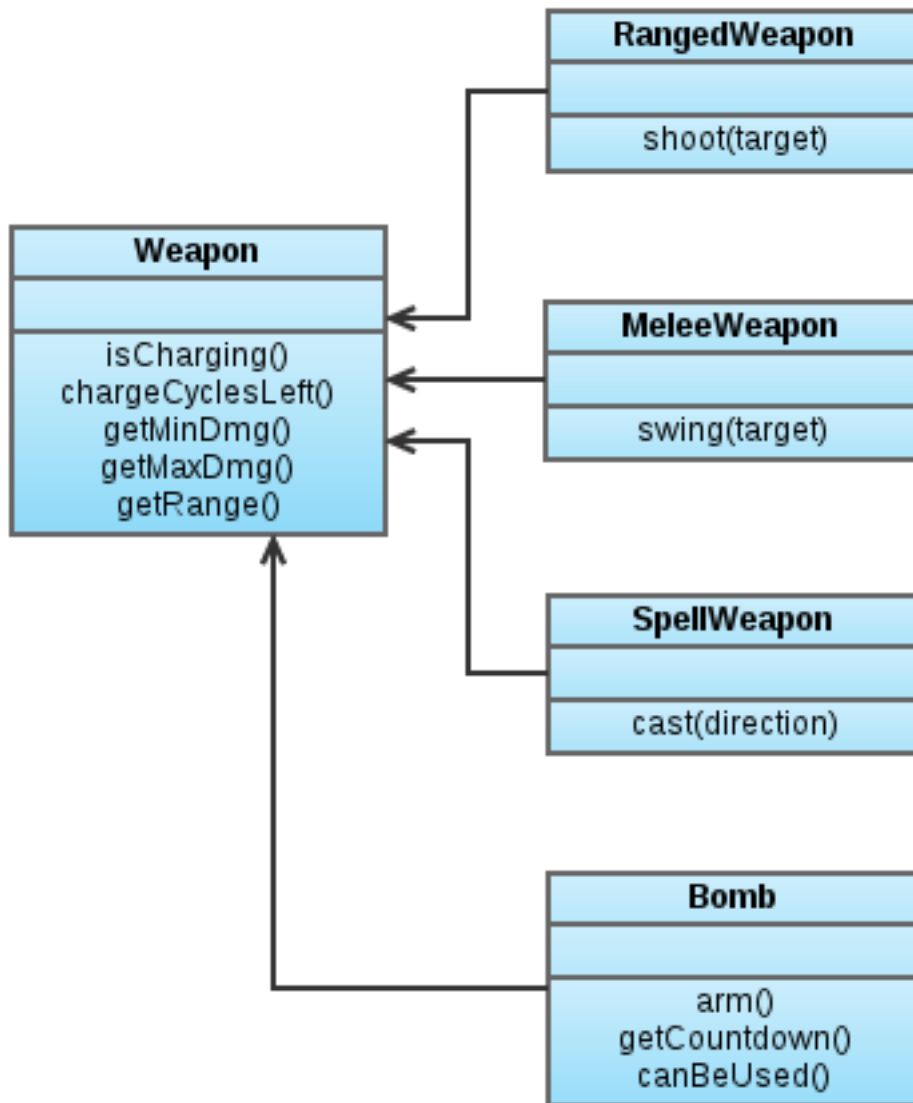
Weapon Master Souvik motions for the weapons to be distributed

You will all receive a set of weapons and source spells. Yes, spells. Bear this in mind: The might of your swing, the accuracy of your shot, the potency of your spells and the strength of your shields all depend on your experience and aptitude in the different aspects of wielding the source. If you do fall in battle, fear not, for I shall be here maintaining a teleportation field that will transport all the wounded back to the infirmary. Once we have ascertained your readiness to rejoin the battle, we shall reinsert you into the fray.

That is all I have for you.

The new weapon types

To properly generate the new conditions in the training simulator, new weapon classes have been created. This diagram should complement the one given in Mission 17.



Mission Start

This mission consists of **TWO** tasks.

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

Task 1:

Time to test out your new weapons! Among the equipment you've received is a laser rifle. This is a long-ranged attack, able to take down robots from afar.

You have also received training to cast force lightning. This is a spell that can be thrown from a distance, and strikes multiple targets. Unlike your other offensive options, spells are more difficult to control, and must be cast in a particular **direction** instead of aimed at a target.

To fully utilise this potential, WM Souvik requires all initiates to undergo ranged combat training. To pass this stage, you must be able to sense enemies from afar in any direction and use your long-ranged attacks to damage them.

Now, instead of **only** attacking bots or drones in your room, you should attack as many bots and drones as possible.

- If there is a bot or drone in your room, and you have a charged **MeleeWeapon**, you should still use it to attack.
- Also, if there is a bot or drone within range of your **RangedWeapon**, and it is charged, you should use that too.
- Finally, if there is a bot or drone within range of your **SpellWeapon**, and it is charged, you should aim that weapon in its direction.

If there are bots and/or drones **only** in your room, and **no other bots or drones in range**, it is **your choice** whether to use your **RangedWeapon** and **SpellWeapon**. The **MeleeWeapon** should still be used.

Make appropriate changes to your custom **Player** class to fulfill the WM's requirement. You should be able to sense enemies in any direction (as opposed to a fixed set of directions), and use your ranged weapon to hit them (unless it is recharging). Submit the changes that you have made (with comments explaining the logic behind your changes). Mark your changes using comments (for example, `/* M18 T1 */`).

Task 2:

The final task is at hand! The XO carefully hands out glowing spheres to everyone, explaining that these are source Bombs. Once set and armed, they will explode in three turns. Keeping in mind the purpose of the mission, the WM has modified them to deal as much damage as possible to the generator, while minimizing casualties to people. Still, it might be a good idea to run.

You are to arm the bomb in the **same room as the Generator**. Make sure that bomb you are using has not already been armed using the `canBeUsed` method.

Make appropriate changes to your custom **Player** class to fulfill the XO's requirement. Submit the changes that you have made (with comments explaining the logic behind your changes). Mark your changes using comments (for example, `/* M18 T2 */`).

Also provide an excerpt of the simulation log showing yourself blowing the generator to bits.

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: Weapon Reference

Range:

The range of a weapon, spell, or bomb all start from 0, with 0 referring to the room the weapon is in. A ranged weapon with range 1, then, can fire at anyone in the current or an adjacent room, a spell with range 1 will hit every enemy in the current and adjacent rooms, and a bomb with range 1 will hit everyone in the current and all adjacent rooms.

MeleeWeapon:

Melee weapons are the weapons you have been using in previous missions. They have the highest damage but also have a maximum range of 0. The parameter to be supplied to a melee weapon is a list containing the intended target, for example, `this.use(melee_weapon, list(target))`. You can also call the shoot function on the weapon, passing one target as a parameter: `lightsaber.swing(target)`.

RangedWeapon:

A ranged weapon often has relatively high attack and range, but can only hit one target at a time. Like melee weapons, a list containing the intended target is passed as an argument when the weapon is used, for example, `this.use(ranged_weapon, list(target))`. You can also call the shoot function on the weapon, which requires passing one target as a parameter: `laser.shoot(target)`.

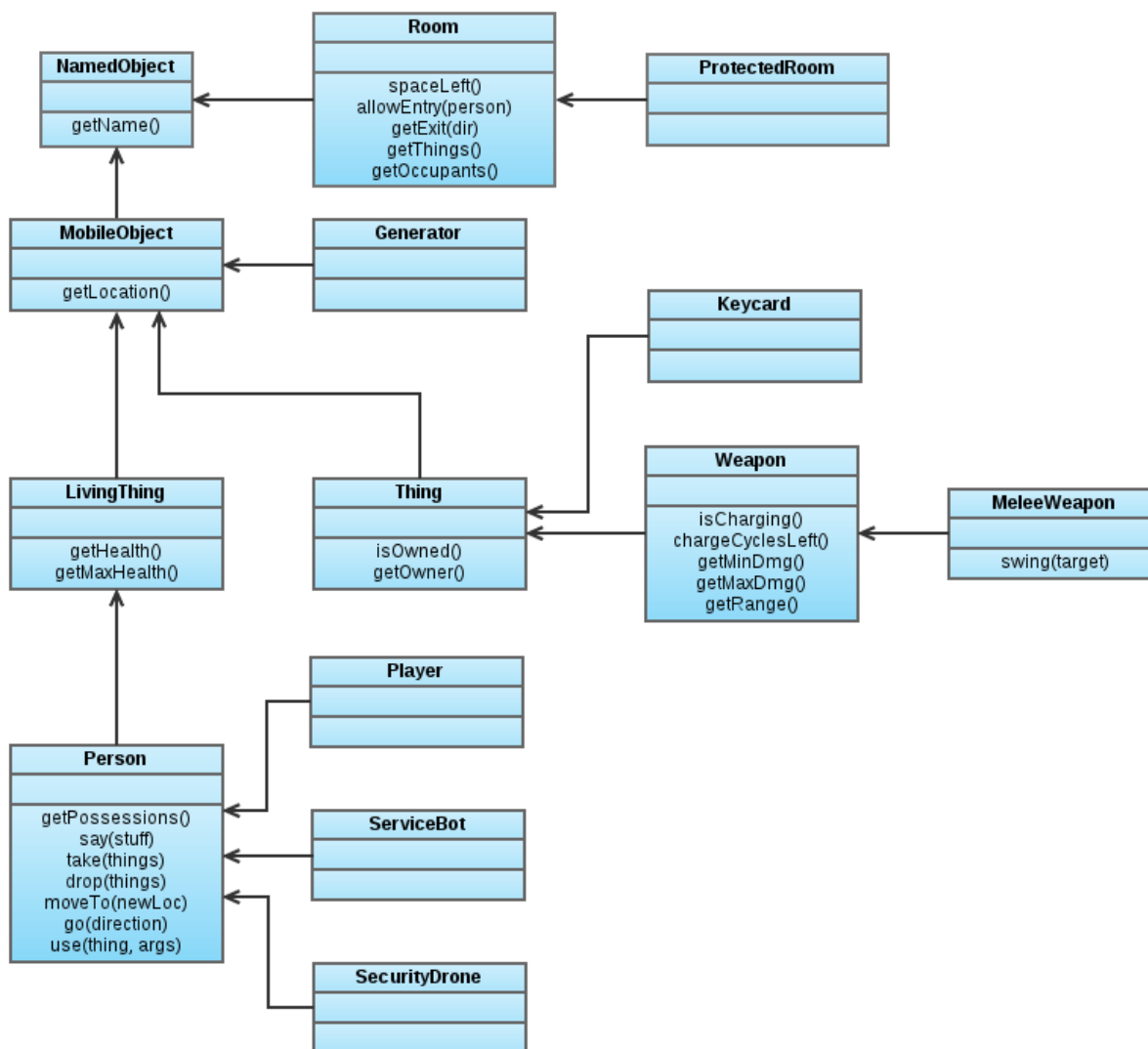
Spell:

A spell deals lesser damage to enemies, but will target every enemy it is able to. The parameter to be supplied when using a spell is the direction, for example, `this.use(spell, "north")`. This will hit every enemy in the current room and `n` rooms to the north, where `n` is the range of the spell (limited to the length of the path). Spells cannot be aimed at rooms in other floors. You can also call the cast function on the spell: `spell.cast("north")`.

Bomb:

A bomb deals damage in a large area. When using a bomb, there is no parameter that needs to be supplied. Do remember, however, that bombs will only explode after `bomb.getCountdown()` clock ticks. To plant a bomb, call `this.use(bomb)`. Alternatively, you can call the arm function on the bomb: `bomb.arm()`.

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 items
say	stuff:string	Says stuff
take	things:list of things	Takes all specified things
drop	things:list of things	Drops all specified things
use	thing:Thing, ...	Uses the item. See use
moveTo	newLoc:Room	Moves to newLoc .
go	dir:string (“north”, “south”, “east”, “west”, “up”, “down”)	Moves in that direction

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:

<code>this.use(melee_weapon, list(target))</code>	Use <code>melee_weapon</code> to attack selected <code>target</code> in the same room
<code>this.use(ranged_weapon, list(target))</code>	Use <code>ranged_weapon</code> to attack selected <code>target</code> in range. Ranged attacks cannot penetrate a <code>ProtectedRoom</code> .
<code>this.use(spell, direction)</code>	Use <code>spell</code> to attack all targets in selected direction , including those in the current room. Ranged attacks cannot penetrate a <code>ProtectedRoom</code> .
<code>this.use(bomb)</code>	Sets the bomb in the current room, which will explode in all directions, except into a <code>ProtectedRoom</code> .
