National University of Singapore
School of Computing
CS1101S: Programming Methodology (Scheme)
Semester I, 2011/2012

Mission 19:
The Final Showdown

Issued: 21 October 2011

Due: 28 October 2011, before 23:59

Death Cube Assault Contest Due: 31 October 2011, before 23:59

Readings:

- SICP: Chapter 3, Section 3.1, 3.2, 3.3
- Concrete Abstractions, Chapter 14
- 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 code. Describe your approach to the programming questions, and then annotate the blocks of relevant code 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 Zi Han address the initiates of the Alliance

JFDI! 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!

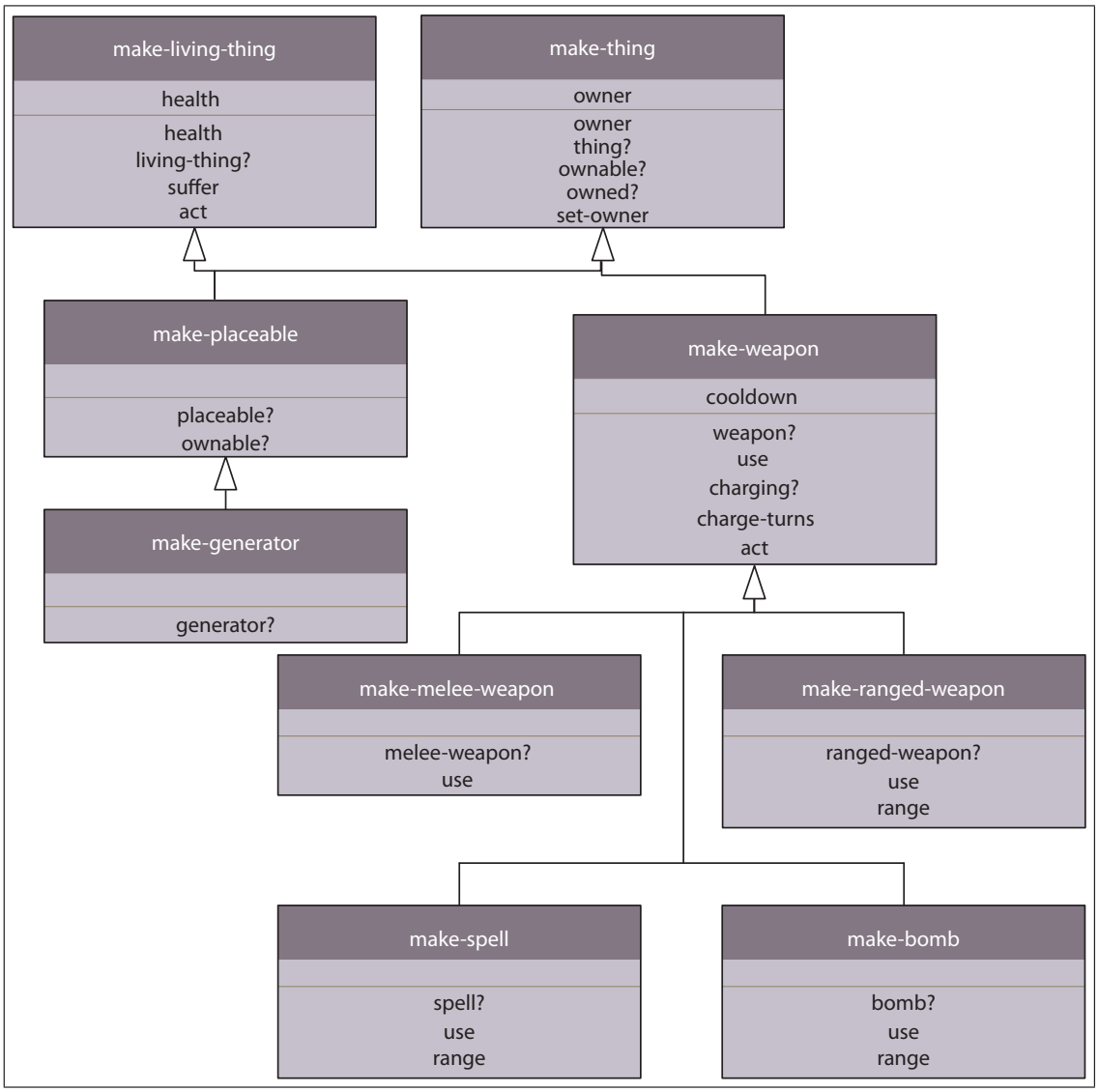
Weapon Master Patrick motions for the weapons to be distributed

You will all receive a set of weapons and Force 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 Force. 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 Force be with you always.

The new weapon types

To properly generate the new conditions in the training simulator, new item 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: (6 marks)

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.

To fully utilise this potential, WM Patrick 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 attack 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 `melee-weapon`, you should still use it to attack.
- Also, if there is a bot or drone within range of your `ranged-weapon`, and it is charged, you should use that too.

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 `ranged-weapon`. The `melee-weapon` should still be used.

Make appropriate changes to `make-player` 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).

Task 2: (4 marks)

The final task is at hand! The XO carefully hands out glowing spheres to everyone, explaining that these are Force Bombs. Once set, 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 place the bomb in the **same room as the generator**.

Make appropriate changes to `make-player` to fulfill the XO's requirement. Submit the changes that you have made (with comments explaining the logic behind your changes).

Please provide clear comments explaining which section is for Task 1, and which is for Task 2.

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

Appendix: Weapon Details

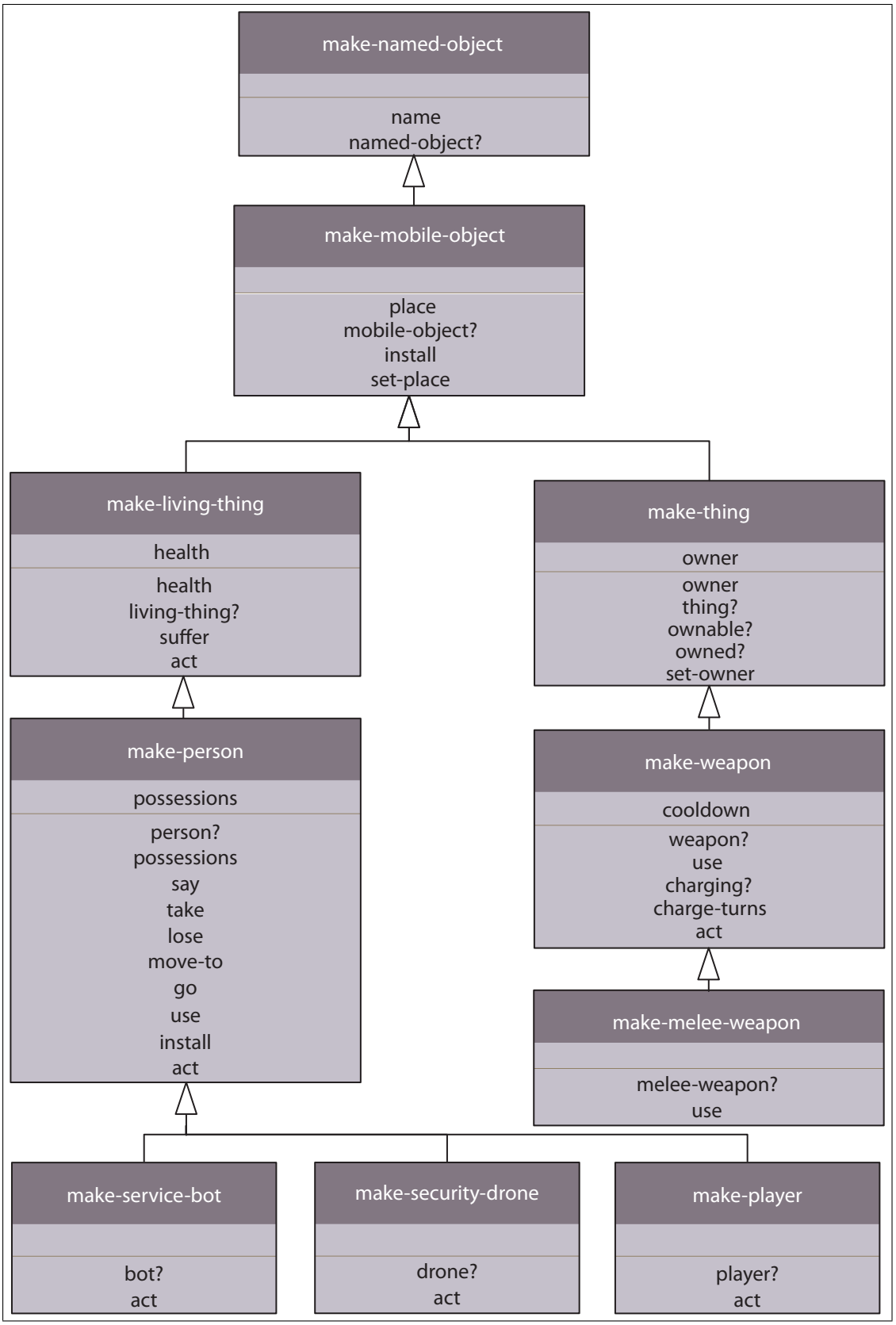
Range: The range of a ranged-weapon, spell, or bomb all start from 0, with 0 referring to the room the weapon is in. A ranged-weapon with range 1, then, can fire at anyone in the current or an adjacent room, a spell with range 1 will hit every enemy in the current and adjacent rooms, and a bomb with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

Ranged-Weapons: A ranged-weapon often has relatively high attack and range, but can only hit one target at a time. The parameter to be supplied when using a ranged-weapon is the intended target, for example, `(ask self 'use laser b1)`.

Spells: 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, `(ask self 'use lightning '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).

Bombs: 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 `(ask bomb 'countdown)` clock-ticks. To plant a bomb, call `(ask self 'use bomb)`.

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



Appendix: Useful Procedures and Methods

General Procedures		
Name	**Parameters**	**Description**
`other-people-at-place`	`person:Person`, `place:Place`	Retrieves all people at `place`, except `person`.
`pick-random`	`lst:list`	Picks a random item in `lst`.
`random-neighbour`	`place:Place`	Picks a random adjacent room.

`split-list`	`pred:procedure`, `lst:list`	Equivalent to
		`(cons`
		`(filter pred lst)`
		`(filter (negate pred) lst))`

Place Methods: `(ask place method . args)`		
`neighbours`		Retrieves a list of adjacent rooms
`exits`		Retrieves a list of directions leading out of the room
`things`		Retrives a list of things inside, whether living or not, owned or unowned
`neighbour-towards`	`dir:direction`	Retrieves the neighbour in that direction, or `#f` if there is none
`accept-person?`	`person:Person`	Checks if `person` can enter
Living-thing Methods: `(ask living-thing method . args)`		
`health`		Retrieves current Health Points
Person Methods: `(ask person method . args)`		
`possessions`		Retrieves list of items owned
`say`	`sentence:list`	Says the sentence
`take` / `pick`	`item:Thing, item...`	Takes all specified items
`lose` / `drop`	`item:Thing, item...`	Loses all specified items
`use`	`item:Thing, ...`	Uses the item. See the items table for more details
`move-to`	`room:Place`	Moves to `room`. Must be adjacent to current location
`go`	`dir:direction`	Moves in that direction
`act`		Called every clock cycle

Weapon Manuals	
`(ask weapon 'charging?)`	Check if a weapon is charging
`(ask weapon 'charge-turns)`	Check remaining charging time
`(ask weapon 'max-damage target)`	Determines max. damage on a target
`(ask weapon 'min-damage target)`	Determines min. damage on a target
Using Weapons:	
`(ask self 'use` **`melee-weapon`** `target)`	Use `melee-weapon` to attack selected `target` in the same room.
`(ask self 'use` **`ranged-weapon`** `target)`	Use `ranged-weapon` to attack selected `target` in range. Ranged attacks cannot penetrate a `protected-room`.
`(ask` **`ranged-weapon`** `'range)`	Retrieves the weapon's range. 0 means it is limited to the current room.
`(ask self 'use` **`spell`** `dir)`	Use `spell` to attack all targets in selected **direction**, including those in the current room. Ranged attacks cannot penetrate a `protected-room`.
`(ask` **`spell`** `'range)`	Retrieves the spell's range. 0 means it is limited to the current room.
`(ask self 'use` **`bomb`**`)`	Sets the bomb in the current room, which will explode in **all** directions, except into a `protected-room`.
`(ask` **`bomb`** `'range)`	Retrieves the bomb's range. 0 means it is limited to the current room.

The Force is with you. Always.