Please do not reply to this email. Contact your lecturer or demonstrator with any queries about your results.

Assignment 2

First iteration

FIT2099 - Object-Oriented Design and Implementation - 1 - 2018

28305728 - LUHAN CHENG 28393414 - MARCUS KEN JIE OOI

Task 1: Deliverables

This section is about the quality of your code and submitted documentation.

Task 1.1: Java code

Task 1.1.1: Functionality

This section covers the quality of the code you submitted. For full marks here, your implementation will need to be robust and stable. It must also meets specification.

Leave Affordance - left object is in actor's location, and can be taken	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
Force Ability - some actors can have it	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
Lightsabers - only actors with lots of force ability can use one as weapon	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
Ben Kenobi can train Luke when the are in the same location	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
Ben training Luke gives Luke sufficient force ability to use a lightsaber	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
Droids can't use force	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
Droids can have an owner	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
Droid movement with an owner follows the logic in the specification	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
Droid movement without an owner follows the logic in the specification	☐ Poor ☑ **Okay** ☐ Good ☐ N/A

Droid loses health when it moves in Badlands	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
Droid with no health can't move	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
Error handling and robustness	☐ Poor ☑ **Okay** ☐ Good ☐ N/A

Task 1.1.1 Comment: Very Good

Mark for task 1.1.1: 4.5 out of 5.0

Task 1.1.2: Style and readability

Variable, attribute, and method names are well-chosen	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
Use of javadoc	☑ **Poor** ☐ Okay ☐ Good ☐ N/A
Quality of comments	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
Readability (Layout, use of whitespace indentation etc.)	☐ Poor ☐ Okay ☑ **Good** ☐ N/A

Task 1.1.2 Comment: Can't see the generated JavaDoc. Also the comments not following the JavaDoc conventions.

Mark for task 1.1.2: 1.5 out of 2.0
Mark for task 1.1: 6.0 out of 7.0

Task 1.2: Supporting documentation

If your design changed during implementation, e.g. because you needed to refactor or discovered that your initial design was unworkable, you needed to keep your documentation up to date. If not, there was no need to change the documentation.

Design documentation matches code	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
New documentation is readable	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
Notation used (if any) is correct	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
Spelling, grammar etc. of written documentation (if any) is correct	☐ Poor ☑ **Okay** ☐ Good ☐ N/A

Task 1.2 Comment: None
Mark for task 1.2: 2.5 out of 3.0
Mark for task 1: 8.5 out of 10.0

Task 2: Design quality

Task 2.1: Design of new system components

This section is about the design as embodied in the code rather than the design as written in the documentation.

Engine code is unchanged (i.e. classes under edu.monash.fit2099)	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
Good use made of existing infrastructure (e.g. Actions, Affordances, Entities)	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
Use of encapsulation (e.g. implementation hiding, use of private attributes)	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
Quality of abstractions (classes contain related data and functionality)	☐ Poor ☐ Okay ☑ **Good** ☐ N/A
Public interfaces are well-designed for new classes/ packages	☐ Poor ☑ **Okay** ☐ Good ☐ N/A

Task 2.1 Comment: None
Mark for task 2.1: 3.5 out of 4.0

Task 2.2: Integration with existing system

The implementation makes appropriate use of the existing framework of actors, entities, actions, affordances, etc.

Task 2.2 Comment: Very Good

Mark for task 2.2: 4.0 out of 4.0
Mark for task 2: 7.5 out of 8.0

Task 3: Meeting submission requirements

Task 3.1: Use of Git and GitHub

Both partners used GitHub	☐ No ☑ **Yes**
Frequency of commits	☐ Just one ☐ A few ☑ **Many**
Quality of commit comments	☐ None ☐ Poor ☑ **Okay** ☐ Good

Task 3.1 Comment: None
Mark for task 3.1: 2.0 out of 2.0
Task 3 Comment: None
Mark for task 3: 2.0 out of 2.0

Task 4: Work allocation

This section is only used if teaching staff become aware that a teammate has significantly breached the Work Breakdown Agreement.

4.1: Work allocation bonus

If you have done significantly more work than is fair, you may receive compensation here.

Task 4.1 Comment:
Mark for task 4.1: None out of 20.0

4.2: Work allocation penalty

If you have done significantly less work than is fair, you may receive a penalty here.

Task 4.2 Comment:
Mark for task 4.2: None out of 20.0
Task 4 Comment: None
Mark for task 4: 0.0 out of 20.0

5: Late penalty

Late submissions are penalized at 10% per working day.

Task 5 Comment:
Mark for task 5: None out of 20.0

Total mark: 18.0