

FIT2099 - Object Oriented Design & Implementation - Semester 1, 2017

**Assignment #3 - Star Wars - Roguelike One
Extension Design Changes - Second Iteration: return(theJedi);**

James Sammut and Michael Carter

TABLE OF CONTENTS

Introduction	Page 3
Changes and Additions to Worlds and Inter-World Travel	Page 3
Changes and Additions to Characters and actors	Pages 3 - 4
Additions to the Game Engine (Scheduler)	Pages 4 - 5
References	Page 6

Introduction

In accordance to the overall assignments of FIT2099, Assignment 1 instructed The Zappers team (consisting of James and Michael) to draft a set of UML and sequence diagrams to prepare implementation of code modifications in Assignment 2. Since the Submission of Assignment 2, the team was tasked into creating various extension that was to be implemented to the Star Wars: Roguelike One game (**Assignment 3 - Second Iteration: return(theJedi))** - which design changes, extensions and feature changes are all documented below.

Changes and Additions to Worlds and Inter-World Travel

The first major design change in the extension of Star Wars: Roguelike One game is the inclusion of a new **worlds** package - which includes the three new SWWorld classes that instantiate the Star Wars Universe. These three worlds are Tatooine (the previous working map), the rebel base at Yavin IV, and of course - the Empire's stronghold of the Death Star. Each world has various differing sizes, as well as different SWActors, entities and locations on each respectful world that can be interacted with (explained below).

These three world are contained in the newly created **SWUniverse class in the starwars package**. This universe class contains an Arraylist of all the worlds that are within it, an Arraylist of Millennium Falcon locations - as well as the active world that is being displayed for the Message Renderer, UI Controller and the Scheduler all handle. The implementation of a SWUniverse Class also means that upon creation of a new world (such as Tatooine), the constructor of the world must contain the universe it belongs to - which seems logical - as a world cannot simply exist in its own universe!

Travel is implemented with the **Millennium Falcon entity** - which resides on each of the three worlds described above. The Millennium Falcon implements **three new flight actions** - which allow travel to and from Tatooine, Yavin IV and the Death Star (and intermittent stop overs) at will. Followers of the player will also travel with the player to the new worlds - and will **not** be duplicated once the actor revisits the world in which they were created.

Changes and Additions to Characters and Actors

Various **new characters** have been added to the game in the extension of worlds and the overall Star Wars universe.

Firstly, on the rebel base on Yavin IV - two new humanoid characters have been added. The first is **Mon Mothma** - who will comment snidely if the player arrives at the base by themselves to 'bring back their General' (Leia), while also calling Luke a 'farm-boy'.

The second character is **General Ackbar** - tactician - and extremely sensitive to planning and organising. Ackbar will sometimes yell it's a trap!' if the player is on the rebel base - otherwise, will be staring at the player unnervingly.

To the new Death Star characters - the first and most numerous of which are **Storm Troopers**. Storm Troopers are the soldiers of the Empire Army - undertaking the Sith Lords' doing - and will attack the player **AND** their associates if contacted with. Being clones, Storm Troopers sometimes wildly attack their opponents - which will result in them missing their attack altogether. What is dangerous about these Troops however, is the possibility of calling in backup - instantiating another Trooper - increasing their numbers on the Death Star and making it -ever-so harder for the player to rescue Leia and beat the game!

Princess Leia (General Organa) also resides on the Death Star initially - albeit unwillingly - as she has been taken captive by the Empire. Leia will be waiting at the opposite side of the Death Star upon arrival - and will soon be made a follower of the player **as soon as they reach her**. She will continuously follow the player in need of safe passage back to Yavin IV - where she can plan the rebel attack on the Empire - along with her resistance.

Possibly the most fearsome character in the game, **Darth Vader** resides on the Death Star - initially placed in the middle of its Grid World. Darth Vader is an extremely powerful Sith - with 10000HP, makes for an incredibly powerful and tough to beat opponent. Vader wanders randomly around the map - in search for any enemy actor that is present. If Vader comes across an enemy, he can use the **Force Choke** ability - which also leads to an **Influence** attempt (for the player) into switching over to the Dark Side. Along with dealing incredible amounts of damage - Vader is tough - and if the player can somehow beat the Sith Lord - will end in victory and winning of the game!

Changes to existing actors consisted first of **R2-D2 holding Death Star Plans** instead of some Droid Parts. R2-D2 AND Leia are pivotal in the rebels able to defeat the Empire - hence Mon Mothma's message! In addition to the specific extension of R2-D2, changes were made **in the immobility of Droids** - where if a Droid became immobile while following the player - they would be removed from the players' followers Arraylists - which is described below.

The player also has two new Arraylists as part of their instance - a list depicting **their followers' Symbols**, as well as another Arraylist **depicting the SWActor followers that are following the player**. This was useful in determining who Luke has in his 'party' - and is essential in developing a winning situation in the game!

Additions to the Game Engine (Scheduler)

Star Wars: Roguelike One, in this extension exercise - was to have the functionality of **winning and losing** - when certain conditions are met. The winning conditions include

defeating Darth Vader or taking Leia and R2-D2 (with plans) to the rebel base at Yavin IV.

Losing the game includes the player and/or Leia being defeated, R2-D2 being disassembled - or the conversion of Luke (player) to the Dark Side by his father, Darth Vader.

The implementation of the winning and losing functionality **was implemented in the games' engine code - and not in the Star Wars client base**. This is due to the nature of the engine to be reused in other games - and not just in Star Wars: Roguelike One. The change to the engine code was **minimalistic** - yet **expansive** on the possibilities that it allows. In the addition to the game engine, **two new functions were added to the Scheduler class in the edu.monash.fit2099.simulator.time package**. The two functions added are the **lossSchedule(Message Renderer)** and **winSchedule(Message Renderer)** functions, explained below.

Both the **lossSchedule** and **winSchedule** functions take in a **Message Renderer** as a parameter for their instantiation. This is to allow the display of messages to the devices' console in a win or lose event - described for Star Wars: Roguelike One above. It is important to note also that it is good design to have this functionality here - as if Star Wars: Roguelike One were to be implemented by a UX team to include sprites and graphics - the underlying console message will still be available for debugging and testing purposes.

The **lossSchedule** function displays a rendering by the passed in Message Renderer that the game has been lost by a circumstance that has made the player lose their goal / task. The function then exits the runtime of Star Wars: Roguelike One by calling in a **System.exit(0)** call - which halts processing and ends the game.

Similarly, the **winSchedule** function displays a rendering by the passed in Message Renderer that the game has been won by a circumstance, action or task that has made the player meet the goal of the game. The function then exits the runtime of Star Wars: Roguelike One by calling in a **System.exit(0)** call - which halts processing and ends the game.

Both functions are available in all functions of the client (Star Wars) packages - which was extremely easy and efficient to insert into. Where the lose and win conditions are met (i.e if a SWActor is dead, if Luke's followers list contains R2-D2 and Leia at Yavin IV) - one of these two functions is called - in the termination of the game (whilst displaying a victory/loss message).

References

Javatpoint 2017, *Java Switch Statement*, viewed 10 May 2017,
<https://www.javatpoint.com/java-switch>

Javin, P 2017, How to find length/size of ArrayList in Java? Example, *Java67*, viewed 22 May 2017,
<http://www.java67.com/2016/07/how-to-find-length-size-of-arraylist-in-java.html>

Stack Overflow 2011, *Creating a new ArrayList in Java*, viewed 22 May 2017,
<https://stackoverflow.com/questions/5915892/creating-a-new-arraylist-in-java>

Stack Overflow 2011, *Getting random numbers in Java [duplicate]*, viewed 10 May 2017,
<http://stackoverflow.com/questions/5887709/getting-random-numbers-in-java>

Stack Overflow 2012, *How to remove first and last character of a string?*, viewed 22 May 2017,
<https://stackoverflow.com/questions/8846173/how-to-remove-first-and-last-character-of-a-string>

Stack Overflow 2010, *In Java, how do I check if a string contains a substring (ignoring case)? [duplicate]*, viewed 7 May 2017,
<http://stackoverflow.com/questions/2275004/in-java-how-do-i-check-if-a-string-contains-a-substring-ignoring-case>

Stack Overflow 2010, *In Java how does one turn a String into a char or a char into a String?*, viewed 10 April 2017,
<http://stackoverflow.com/questions/2429228/in-java-how-does-one-turn-a-string-into-a-char-or-a-char-into-a-string>

Stack Overflow 2010, *Java ArrayList Index*, viewed 22 May 2017,
<https://stackoverflow.com/questions/4313457/java-arraylist-index>

The Internet Movie Database 2017, *Quotes for C-3PO (Character)*, viewed 10 April 2017,
<http://www.imdb.com/character/ch0000048/quotes>