UNIVERSITY OF WAIKATO

COMPX304-19B—Advanced Networking and Cyber Security

Lab 7—Introduction to Software Development in C

Problem Description

You are working as a Chief Engineer for the spaceship U.S.S. Aotearoa II, which is currently posted at the border with the Gamma quadrant. Being near the border, your commanding officer Captain Panos Patros has tasked you to design, implement and run a set of battle simulations to assess our preparedness to resist a Borg invasion. As you are well aware, our ship's supercomputer can be programmed in the only language that survived the Programming Wars of 2153: ANSI C. Being an Engineer, you are also well trained in encapsulation, abstraction and separation of concerns, all of which your solution must satisfy.

A configurable number of battle simulations will be conducted. On initialization, each battle simulation is assigned a unique id and references to two spaceships, created specifically for it. A battle is simulated by successive attacks from one ship to the second and vice-versa until either one ship is destroyed or a configurable number of attacking rounds have passed. At the end of the battle, the memory of both spaceships needs to be released.

Each spaceship is assigned a name, which must be copied from a parameter passed during its initialization. Also on initialization, each spaceship is assigned a pseudorandom number of health points (80-100), attack points (10-20) and defense points (5-10). When a spaceship attacks another spaceship its attack power is determined pseudorandomly as either equal to its attack points (50% of the time) or half its attack points (50% of the time). During an attack, a battleship receives damage, which is subtracted by its health points, equal to the difference between the incoming attack power and its defense points; if the incoming attack power is less than the ship's defense points, no change occurs in its health points. If a ship's health points become zero or less, it is considered detroyed.

A dynamic-length log of all the conducted battles needs to be maintained. Each record in the will store a pointer to the battle struct object, the initial heath points of both ships, its outcome and the attack rounds it lasted. Additionally, each record will also store a pointer to the next record; new records may be added in the front of the log. The log can be iterated and its contents printed on-screen as well as destroyed while ensuring that all memory reserved for records and/or battles is released.

Part 1

Design and develop a module (a pair of a .h header file containing declarations and a .c file containing implementations), for each of the 4 required object types: Spaceship, BattleSimulation, Record and Log. Test each module incrementally.

Start by compiling and modifying the code that the Captain has attached with your orders.

There is a plethora of documentation on the Inter-galactic Network (also known as the Internet), that can be used as a reference, such as https://www.gnu.org/software/gnu-c-manual/gnu-c-manual.pdf and https://devdocs.io/c/.

Part 2

After you write and test each of the modules required, write and test a main function that creates a log, performs and stores the results of 1000 battle simulations, prints the results on screen and releases the log.

----------------------------------Lab 7 Ends----------------------------------