COS 214 Project

1 Introduction

Warfare has appeared throughout human history, and surprisingly remained on average level, even though there might be differences in intensity and scope of each. On average there are 2,000,000 deaths over six major wars every ten years. R.J. Rummel mentions that there were 315 wars since 1815. Extrapolating this statistic, presents one with the mind blowing figure of 13,600 wars since 3,600 B.C. Furthermore the author postulates various causes for war such as: (a) opposing interests and capabilities (specific sociocultural differences and similarities between the parties); (b) contact and salience (awareness); (c) significant change in the balance of powers; (d) individual perceptions and expectations; (e) a disrupted structure of expectations; and (f) a will-to-conflict. For an in-depth study into war refer to the notes of R.J. Rummel `https://www.hawaii.edu/powerkills/UCW.HTM`. [1, 2, 3, 4]

In this project a war simulator software system will be constructed. The design of the software system must be conducted using the design patterns as identified by Gamma et. al.

1.1 Objectives

The objectives of this project will be to:

- identify the requirements for a given software requirement;

- translate the requirements into a design using design patterns;

- highlight aspects of the design that are important in the implementation of the design using UML;

- implement the design in C++ by separating the concerns and providing the class definitions and implementations in separate .h and .cpp files;

- compile a design document that is to be used to understand the design and implementation of the project; and

- provide an opportunity to work in a software development team

1.2 Outcomes

When you have completed this project you will:

- have designed and modelled a system and experienced integrating design patterns,

- be able to say with conviction that you have completed a relatively large project,

- have experience of working in a relatively large design and development team, and

- be ready for COS301 next year.

2 Constraints

1. You must complete this project in teams of 4 to 6.

2. At least 10 design patterns must be included in the design and the implementation thereof.

3. UML diagrams showing the design and aspects of the system that need additional explanation.

4. git must be used as the version control system

5. Doxygen must be used to produce documentation about your API

3 Time line

1. 23 September 2022 - Team registration of 4 to 6 to be completed. One member of the team should be assigned as team lead.

2. 4 October 2022 - Submit pre-initial design.

3. 4 and 5 October 2022 - Meet with team manager and discuss pre-initial design.

4. 11 and 12 October 2022 - Meet with team manager discuss initial design.

5. 18 and 19 October 2022 - Meet with team manager.

6. 25 and 26 October 2022 - Meet with team manager.

7. 1 and 2 November 2022 - Meet with team manager.

8. 8 November 2022 - Submission of project on ClickUP.

9. 8 and 9 November 2022 - Project demos.

4 Mark Allocation

Task	Weights
Practical Assignment 5	0
Design	25
Implementation	25
Report	20
Development Practices	10
Demo & Presentation	20
TOTAL	100

5 Project Description

In this project, you are required to conduct additional reading on warfare. As a starting point, the notes of R.J. Rummel [2] as mentioned earlier in this document and the University of Pretoria's library services, which includes eBooks, journals, articles and more, can be of great assistance.

Throughout history, there have been various wars. These wars have ranged in size from small too big with regards to the amount of military personnel. The geographic scope varied, e.g. a disagreement about a small piece of land, to vast parts of continents being disputed or trying to be unified. In wars, countries have either fought alone, or formed alliances, that is to say groups of countries standing together for a common cause or goal. The countries involved in a war form alliances with other countries within their respective country groups (e.g. Country A forms an alliance with Countries B, C and D, and Country E forms an alliance with Countries F, G and H). Each country in a country group has soldiers / armies / troops, weapons and flags (white = surrender). Countries within a specific country group order attacks against other countries within the opposing country group until only one sole superpower country group remains standing.

5.1 War Engine

In this project, you are required to write a generic war simulator, that models the various elements of wars, as identified by this specification and your additional research. The main driver of the war simulator, is the main event loop referred to as the **war engine**. In simple pseudocode, the war engine can be modelled as a player turn based loop event. **Note:** This code is a very simplified model, **AND** will need to be expanded for your war engine.

```
public void warLoop() {
        while (warIsActive) {
                for i := 0 to n do {
                        country := pickCountry(countryGroup[i]);
                        country.attack(pickOpposingCountry(countryGroup, i))
                }
        }
}
```

5.2 Components of War

This section proceeds to list some of the components of warfare. However you are required to identify additional components for your warfare simulator.

5.2.1 War Theatres

Carl von Clausewitz defines the term Kriegstheater, roughly war theatre, in his book *On War* as:

> "Denotes properly such a portion of the space over which war prevails as has its boundaries protected, and thus possesses a kind of independence. This protection may consist of fortresses, or important natural obstacles presented by the country, or even in its being separated by a considerable distance from the rest of the space embraced in the war. Such a portion is not a mere piece of the whole, but a small whole complete in itself; and consequently it is more or less in such a condition that changes which take place at other points in the seat of war have only an indirect and no direct influence upon it. To give an adequate idea of this, we may suppose that on this portion an advance is made, whilst in another quarter a retreat is taking place, or that upon the one an army is acting defensively, whilst an offensive is being carried on upon the other. Such a clearly defined idea as this is not capable of universal application; it is here used merely to indicate the line of distinction."

From the above definition it can be noted that a war theatre includes the area, such as airspace, land or sea were war events take place. Some of the most famous theatres of WWII (World War Two) are:

- The battle of Normandy (D-Day) being the most famous. The counter-offence to begin the retaking of Europe by the allies.

- The battle of Dunkirk. Any and all planes which could fly in the air overhead. Coordinated and communicating. Fighting for their home against the Axis invaders.

- The battle of Alamein in Egypt. Tank squadron battles in the desert with soldiers on foot in the trenches. Commands being overlaid from generals to the tanks and soldiers fighting for their lives.

5.2.2 Transportation

In any war, the transportation of goods, people and services are critical for any party to ensure victory at best, and survival at worst. Destroying the transport corridors of any country or group, will most surely cripple and be a major set back to the party in question. You are required to model the transport in your warfare engine.

5.2.3 Entities

For any party to engage in warfare, entities need to be entered into the war theatre. Various entities are involved in warfare, but more so in modern warfare. A non-exhaustive list of entities include soldiers, refugees, medics, global organisations and neighbouring countries. Furthermore, entities themselves, may iterate through some state machine. An example of this, would be the status of a citizen of a country, e.g. unlisted, enlisted, deployed, fighting, stationed, dead, returned, etc. Similarly the state of rockets might change over the course of time as from the construction of the rocket, to deployment, to successful or unsuccessful detonation.

5.2.4 Phases of War

A war itself may be broken into various phases. For your reference three different views on the phases of war have been presented below. Note that these are not necessary authoritative views, and you are welcome to draw inspiration from these sources or others in defining your phases of war.

- https://warontherocks.com/2016/11/phases-of-war-and-the-iraq-experience/
- https://www.researchgate.net/figure/Chronological-Phases-of-Conflict-Intensity-Levels-Swanstrom-and fig1_318960397
- https://web.mit.edu/cascon/warend.html

5.2.5 Changes to War Engine

Lastly, as time progresses, by a country or country group might make various changes to how they engage with the enemy. These changes might include, increased spend on R&D to enhance or change existing technologies, adopting no-fire treaties, etc. The way in which countries also respond to provocation and attack, can change over time depending on various factors. Countries might even withdraw from country groups, join existing country groups, change sides or request assistance from other country groups.

5.3 Launch Reenactment

Provide an interface to setup war simulations. It should be possible to run a war simulation in design mode or real mode. Design mode war simulations can be interrupted, tweaked and then allowed to continue. Real mode simulations are setup and run.

6 Tasks

Task 1: Practical Assignment 5 .... (0 marks)
For Practical Assignment 5 you are required to submit a pre-initial UML class diagram design. Refer to Task 2 for a list of steps which can be iterated through to assist you in designing your system. Furthermore refer to the timeline given in Section 3.

Task 2: Design .... (25 marks)

2.1 Identify the functional requirements.

2.2 Design the processes using Activity diagrams.

2.3 Decide on the patterns to address the functionality defined by the functional requirements and processes.

2.4 Design the classes for each of the identified patterns taking their interrelationships into account.

2.5 Draw a class diagram of your system.

2.6 Draw Sequence and communication diagrams showing the message passing between objects.

2.7 Design state diagrams showing how an object (which could also be a composite) changes state.

2.8 Provide at least two object diagrams showing the state of the objects active in the war simulation at a specific point in time.

Task 3: Implementation .... (25 marks)

3.1 Implement your war simulator including **at least** a text based interface.

3.2 Implementation of a graphical interface is not required, however will provide you with bonus marks.

Task 4: Report .... (20 marks)

4.1 You are required to write a brief about your research, ensuring that you reference all sources. Your brief should include what you understand about warfare as well as associated entities, phases, etc. that you will be using as part of you warfare engine. Make sure to document any further assumptions or decisions you made. Lastly ensure to include relevant definitions and explanations where required.

4.2 A report stating how you applied the design patterns to address the functionality required by the system. This report should include UML diagrams to augment the explanation. This Task goes hand-in-hand with the Design task. Much of the design must be reported on in this task.

Task 5: Development Practices .... (10 marks)

5.1 Use git as the Version Control System (VCS). Every member of your **MUST** make at least 10 (ten) commits.

5.2 Document your code using C++ documentation standards.

5.3 Generate documentation for your system using Doxygen.

5.4 Develop your code according to best practices using a contract first design.

5.5 Use an automated unit testing framework to unit test your system. Full coverage is not required, however every member of your team **MUST** implement various unit tests.

Task 6: Demo & Presentation .... (20 marks)

Your entire team is required to be present at the demo. Failure of team members to join the demo will result in a penalty for all team members. Your demo must be professional and well prepared.

7   Submission Instructions

Each team is required to create a Git repository to manage the project. Manage your Git repository with GitHub. Make sure that in the documentation, a link to your GitHub is provided. The following must be available in your repository.

- Your system, this includes all your source files (that is `.h` and `.cpp`) and your Makefile.

- Any data files you may have created and are needed to run your program.

- A readme (`readme.txt`) explaining how to compile and run the program and the placement of any data files you may have created.

- A PDF version of your latest report in a folder named `Report`. The report must be written in Google Docs and a link to the Google Docs version of the report included in the PDF version of the document.

Make a snapshot of your Git repository and submit it as an archive to the ClickUP submission slot before the deadline.

- Place all your system files in a folder called `System`. Running your makefile in the `System` folder should compile and link your system correctly.

- Place your data files, if any, in a `Data` folder.

- Make sure your `readme` is included in the root directory of your archive.

- A PDF version of your report in a folder named `Report`.

Failure to upload the project to ClickUP will result in the team receiving 0. It is the duty of the team lead to upload the file.

You will be required to demonstrate your system during the week of 1 November 2022.

References

[1] F. H. Denton, Mar 2005. [Online]. Available: https://www.hawaii.edu/powerkills/WAR.ROOTS.HTML

[2] R. J. Rummel, "Understanding conflict and war," Nov 2002. [Online]. Available: https://www.hawaii.edu/powerkills/UCW.HTM

[3] ——, "Chapter 16: Causes and conditions of international conflict and war," Nov 2002. [Online]. Available: https://www.hawaii.edu/powerkills/WPP.CHAP16.HTM

[4] ——, "Volume 4: War, power, peace," Nov 2002. [Online]. Available: https://www.hawaii.edu/powerkills/NOTE13.HTM