DSA: DATA STRUCTURES AND ALGORITHM

I. APPLICATION THEME: Game Development

II. RATIONALE

We originally set out to create a video game before the themes we could use were released. So we chose to go make an application under the "Game Development" theme.

III. EXPLANATION OF THE IMPLEMENTED APPLICATION AND ITS FEATURES

Our application is a game where you play as a Wizard with the power of magic and time travel. You battle three elemental demons armed with your spells and ability to rewind time.

There is a combo box of the available spells you can use— each with their own strengths and weaknesses. A button labeled "Cast Spell" allows you to cast that spell against an enemy, after which the enemy will attack you too.

A "Rewind" button sends you back in time prior to your previous action, effectively undoing it. This is implemented using a stack.

The enemies can randomly switch out of battle, becoming dequeued and then enqueued to the back of the enemyQueue. This is guaranteed to happen when they go below critical health.

IV. THREE TEST CASES FOR THE CHOSEN APPLICATION

A. WIN/LOSE CONDITION

- ➢ Check whether or not the Game Over Screen triggers when the Player/All Enemy's health reaches zero.
- ➢ Check if it's the correct Game Over Screen.

B. TYPE INTERACTIONS

- ➢ Check if the type effectiveness follows (Fire >> Grass >> Water >> Fire) and (Normal > All Types (excluding Normal)).

C. TIME TRAVEL ACCURACY

- ➢ Check if "Rewind" button correctly undoes each possible action.

V. CHALLENGES FACED DURING APPLICATION

From the start, choosing a GUI library has been a big obstacle for our development. Qt was a gigantic hassle to install: the Open Source route through GitHub proved to be a nightmare to traverse with all its dependencies, incompatibility with our machine, and bugs, particularly with the Makefile; the Online Installer was also not incredibly slow, even with our fast WiFi, it kept freezing too. Eventually, we settled on Dear ImGui, a very light immediate GUI library which is pretty different compared to what we are usually used to, but relatively easier to implement to our project.

We also had a lot of trouble implementing our own stack and queue as we tried to use std::any to mimic the built-in stack and queue library.

VI.

VII. GROUP MEMBERS/MEMBER ROLE

- Amrei Louis Espares - Developer/ Designer
- Michaela Zapirah Yusilon - Developer/ Bugtester
- Zach Miguel Senado - Developer/ Bugtester