# Program.cs

I am not documenting this shit; even I do not know how it works at all.

# Unit.cs

## Region Variables:

It stores all the variables the unit uses.

## Public Unit constructor:

It’s used to make a new unit.

## Math Methods:

They use math to do damage and receive damage.

## Update Methods:

They’re used to make changes to the existing unit file.

### LoadWholeUnitFile:

Pretty much a constructor, but instead of being used to make a new unit, we update EVERYTHING of an existing unit. (Used by Saving and Loading)

## Information Methods:

They only send information about the unit.

# JSONFighterReader.cs

Used to load the fighterClasses.json into a usable format with c# and Newtonsoft.Json.

# FightManager.cs

## UnitLists class:

Contains all the lists that the units could be in. (allies, enemies, deadAllies, deadEnemies)

## FightManager class:

Handles the interactions between units and teams.

### Special Attacks:

#### AttackDeployer:

Is used to identify what type of special attack the unit used, and then runs that attack.

#### All the attacks:

Some are attacks, some are buffs. Each contains a Cooldown time, and some contain a persist time. Persist is always shorter than the Cooldown, and while the current persist time is less than the defined persist time, its effects are active.

# ActionLogWriterAndReader.cs

## WriteTeamStatusLog:

Write’s the current state of all teams in the game, adds 1 into the current round number, and saves it into a ActionLog.json file.

## LoadTeamStatusLog:

Same as WriteTeamStatusLog but instead of saving, it first removes 1 from the current round number, then updates the current team statuses and removes the round information from the ActionLog.json file.

## LoadTeamStatusBase:

Same as LoadTeamStatusLog but it loads the very first round. It is used instead of LoadTeamStatusLog to prevent crashing.

## ResetLog:

It resets it and is also used to make a new ActionLog.json if it is missing.