# 1.0 Introduction

The goal of the project is to create an asynchronous online turn based Pet simulation game. The game will allow multiplayer battling to achieve experience points, and users can play the mini-game to earn food points that is used to heal the pets.

MonsterSim has three main features:

* Pet Creation: Users can create new pets belonging to one of fire, water, earth type.
* FlappyMon's Moonwalker: A mini-game that allows the user to gain food points used to heal the pets.
* Battle!: Allows the user to search for other users and request them for a battle.

# 2.0 Usage

The website can be accessed through: ***http://1-dot-ece1779test0123.appspot.com/***

The website prompts the user to login through their Google account. After the authentication, if the user has no pets then he/she will be redirected to the pet creation stage else he/she will be redirect to Pet Statistics page (main page).

To create a pet, the user first picks one of three element types. Each type is weak against one other type and strong against one other type ( fire is weak against water and strong against rock). The user then is asked to name the monster (has to be unique) and pick one of the three avatars. In the last stage the user is given ten points that he/she must distribute between health, speed, attack, and defense with minimum of one point for each. Finally, the user is shown the pet card which contains all the statistics of the new pet.

The user can gain food points (fed to pet to heal their health) by playing the mini-game FlappyMon's Moonwalker. The numbers of pipes the pet passed through is aggregated as food points. By clicking the submit button on FlappyMon's Moonwalker main page, the user is directed to the leader board showing the food points they collected as well as the top five scores in the world. Through Feed Pet button, the user can select a food item and heal the pet.

The final/main feature is the battle feature. The user first has to click on Battle! button in Pet Statistics page and he/she will be redirected to Battle Search Page which shows the user's monster and a list of potential opponents. Using the radio buttons, he/she can select an opponent and request them for a battle. The opponent has to accept the battle request for the battle to begin. If the opponent accepts, the algorithm calculates who will go first based on the speed of both the monsters. The user has three attacks: weak attack with guaranteed hit, a medium attack with 30% miss rate, and a strong attack with 60% miss rate.



# 3.0 Feature Description

This section will describe how each of the features were implemented.

## 3.1 Battle!

Once the user’s pet and opponent’s pet is selected, both their names and the “nextattacker” value (which is “nobody” since it hasn’t been calculated yet) are passed to the *battle\_start.jsp* page. The “nextattacker” variable is used to show the state of the battle. On this page, since we passed in “nobody” as the next attacker, it means that the battle has just been initiated. The page will now create a new entity in the table *pet\_battle* with the two pet names and the next attacker variable, which will now change to the “request” state. The pet that is used to start the battle will be in the “you” column of the *pet\_battle* table, so the person who initiated the battle will inherently be known.

Both these pets are now unavailable for other people to select in *battle\_search.jsp*. The initiator pet will see a “wait for response” message when it goes into battle. The pet that has been challenged will now be able to accept or reject the battle. If the opponent rejects the battle, a reject variable is sent to *battle\_start.jsp* which will set the battle to the “done” state removing them from the database.

If the opponent accepts, a response variable is sent to *battle\_start.jsp* which will update the “nextattacker” entity's state to whoever will attack first. The pet with more speed will attack first, and the state will be set to “you” (the initiator of the battle) or “them” (the opponent). The *battle\_start.jsp* page will render three attack buttons for the user whose pet is attacking. The other user will not see the buttons, instead seeing a “wait for opponent” message. The page each pet will see depends on the “nextattacker” state, so if the current user’s pet is in the “you” column and the next attacker is “them”, then the user will see the wait message instead of buttons. Also, at any point in this process either user can exit or go back to the main page as everything is saved, making the battle truly asynchronous.

The attacking pet can choose between three attacks: Weak, Normal or Strong. Clicking any of these buttons will send all stats and pet names directly to the *battle\_attack.jsp* page. Here the game will calculate damage, accuracy, and resulting HP. The base damage is calculated as follows:

damage = (((((2\*user\_lvl/5)+2)\*user\_atk/opp\_def)/50)+2)\*weakness\*rand/100;

User refers to the attacking pet. Weakness is determined by the rock-paper-scissors system: Water does 1.5x damage to Fire, Fire does 1.5x damage to Earth, and Earth does 1.5x damage to Water. The Rand variable uses a generated number 70-100 for randomness. The attack calculation also generates another random number 0-10, which is used for hit rate. Weak will always hit but with 0.5x damage, Normal will hit if the random number is under 7 and will do 1x damage, and Strong will hit if the number is under 4, and 2x damage. The final number is subtracted from the opponent’s (not attacking) initial health, and the resulting number is immediately updated to the database. If the opponent’s new health is more than zero, the *pet\_battle* entity state is set to the other pet (if attacking pet was “you”, state will be “them”, and vice-versa). Now the other user will see the buttons to attack, and the current user will see the waiting message.

When a pet attacks and the opponent’s new health is zero, the *pet\_stats* database will be updated with one more win for the attacking pet, and one loss for the losing pet. The pet will gain:

Experience points = (opponents level)\*4/weakness

If total experience points are greater than (current level)^3, all stats add 1

The next attacker state will be “done”, and the winning pet’s name will be erased from the *pet\_battle* entity, allowing it to battle someone else. The losing pet will now see a “battle over” message when it goes to the battle page, with a button to go back to the home page. When either pet goes to the *battle\_search.jsp* page, they will check for their name in the *pet\_battle* database with a done state, and remove it completely from the table.