Name: Stephen
Date: June 11, 2016
Current Module: SQL Database/Testing
Project Name: Thunderdome

**Project Goals:**

The goal of this project was to create an “arena” type program that would access a database of fighters and have them all fight each other. We would then insert into this database from python to create a list of the fights that took place. We also needed a stats file that would give stats on the fights and a reset file that would clear the fight table

**Considerations:**

- What will happen when a new type is seen?
- What will happen when neither opponent is dealing damage?
- Does the database exist? Can it be connected to?

**Initial Design:**

The initial design for my program was to try calculating all the information without the use of classes. However early on in the design I decided that using classes would help store the information for later use so I moved to using classes. I also would query and insert once the fighters have fought. For the stats file I started with a menu and based on the menu option they would get the information they wanted.

**Data Flow:**

The thunderdome program would connect to the database then run queries against the tables to acquire fighter information. The program would have them then fight and once the fights were decided it would insert into the fight table of the database. The stats file would just query the database and calculate all the information about wins/losses/draws and longest time/shortest time. The rese was the simplest because it just deleted the information from the fight table in the database.

**Communication Protocol:**

- PostgreSQL was used to view the database and changes that were made to it

**Potential Pitfalls:**

Some of the pitfalls I faced was to make sure to handle errors if the database didn't exist of the wrong one was given. Another pitfall was to make sure to insert correctly such as with the timestamp field.

**Test Plan:**

**User Test:**

1. I tested whether the fighting had already taken place
2. I tested that thunderdome connected to the database
3. I tested that the reset file cleared the fighter table
4. I tested each menu options for the stats file

**Automated Test:**

None

**Test Cases:**

1. I ran thunderdome.py to see if I got kicked out because the fighting had already happened
2. I ran the program to see if, given the proper database name, it would connect
3. Tested the sql database to make sure is cleared the fight table after ./reset.py was ran
4. Ran each menu option so see if if would give the results needed

**Expected Result:**

For each of these files I expected that it would connect to the database. Once it was connected it would query or insert into the database and give back the proper information about the fighter/combatant.

**Conclusion:**

The project was exceptionally time consuming, it would difficult to figure out how to use/interpret the information brought back from the queries correctly. Another problem I faced was how to insert the timestamps after the fight was over. The thunderdome file was the most difficult file to deal with since it had the damage calculation and the fighting logic.