**Structure**

The system will consist of two components, a “what” generator and a “why” generator. For the creation of “what” statements, we plan to use a database of common animal names. The program will randomly select these names. Once the “what” has been established, we will utilize the ConceptNet and/or the wordnet knowledge graph API(s) in order to generate our whys. Instead of doing some quantitative comparison, we will exploit the looser semantic relations ConceptNet/wordnet provides in order to generate humor through absurdism. The CN API provides a list of things that the inputted animal is “capable of”, and using an algorithm we will select one of these and state “(Animal 2) is better than (Animal 1) because they are able to (capability) unlike (Animal 1)”.

**Tasks for this week:**

David and Mason: Implement the API within our code. Figure out how to send it an animal and retrieve the list of capabilities returned. Figure out how we are going to select the specific capability we will use.

Sid, Vedant, and Daniel: We will figure out the API before this weekend. Then, before monday we need you guys to write the code that actually uses the information retrieved from the API, this likely entails:

* Grabbing a random animal from a list of animals (there's already one in the code somewhere)
* Inserting the capability into the “why” sentence of the structure
* Write an algorithm that goes between 10 back and forths, each randomly selecting an animal and retrieving its capabilities and putting those into a sentence.
  + Would be AWESOME if you made a list of transition words/sentences that work for all animal/capability combos to make the conversation more natural.