# Behavior

The pet will have a set of values which will go down as time passes. These values can be replenished by interacting with your pet via buttons. After the buttons are pressed, the associated value will be ‘satisfied’ for a short time before starting to decline again.

Each value will not be allowed to drop below zero and will not be able to exceed 50. Each button press will increase the values by 10. After 5 seconds, the values will start to drop by 2 every second.

When a button is pressed, the image of the pet will be replaced with another image for the 5 second timeout and then be reset to the original image.

# Design

## Model

The model will be made of a \_\_init\_\_ function which will create all the attributes and set their default values, setter functions for each value, and a callback function. Each setter will execute the callback function to communicate the change back to the controller

## View

The view will display the current attribute values. The buttons will send callbacks to the Controller, and the Controller will update the view. This keeps the Model from being exposed to the view.

## Controller

The controller will manage the view and the model, while keeping both of them separate. The view will send action commands to the controller, the controller will execute these commands by accessing the model, and then the model will callback when completed. The controller will receive this callback and update the view.

## Application

The application is where the Controller, View, and Model are initialized and the loop is setup.

# Interface



