

Finding Flipper

An Interactive, Hyper-Realistic Biography of My Husband, **Albert**,
and His Favorite Fluffy Feline Friend, **Flipper**.

Overview of the Text Game

- **Objective:** Help Albert find Flipper, who may be hanging out somewhere in the house, and make Albert happy.
- **Classes**
 - **Friend:** Takes form of either albert or flipper. These are the main characters that hold the major attributes like sleepiness. They also hold an instance of a Location to show where each character is.
 - **Locations:** Various places in or around the house that Albert or Flipper can go
 - **Activity:** Activities that Albert and Flipper engage in. Impacts Albert or Flipper's happiness level or sleepiness level.
- **User direction:** [m]ove or [s]tay. User direction triggers **consequences()**.

Class - Instances

Friend

- **albert:** Initiates awake and in the Office
- **flipper:** Initiates asleep in the Backyard.

Activity

Albert

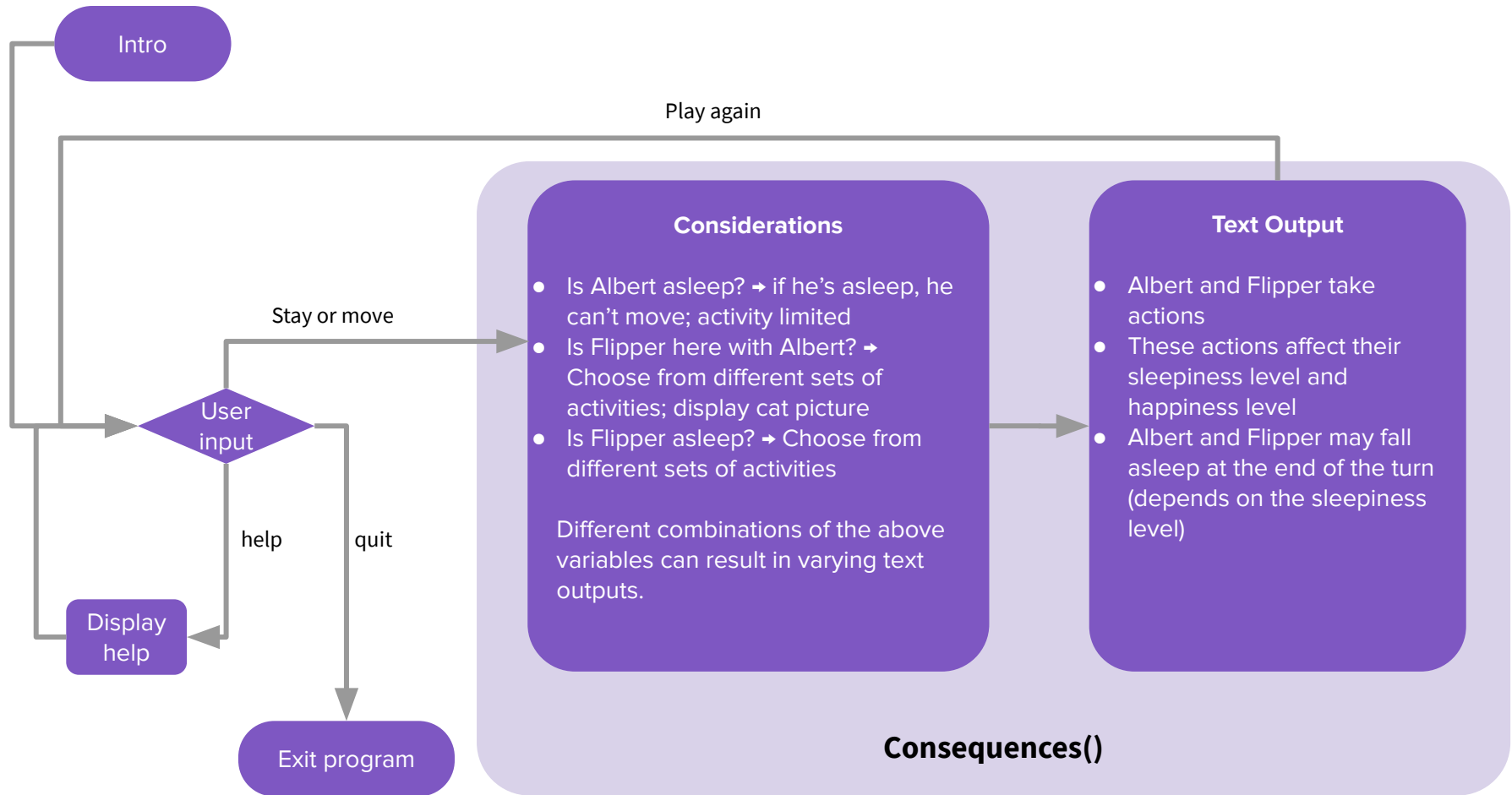
- **Solo:** Work, Sleep
- **With Flipper:** Pet cat

Flipper - We only see her activities when Albert and Flipper are in the same location.

- Chill
- Sleep
- Ask for pets

Locations

- Office
- Front Porch
- Backyard
- Bedroom



Class Structure

Let's explore how the classes
interact with each other.

Classes

Friend

Attributes

- name
- happiness
- sleepiness
- asleep
- location
- emotional_state
- energy

Activity

Attributes

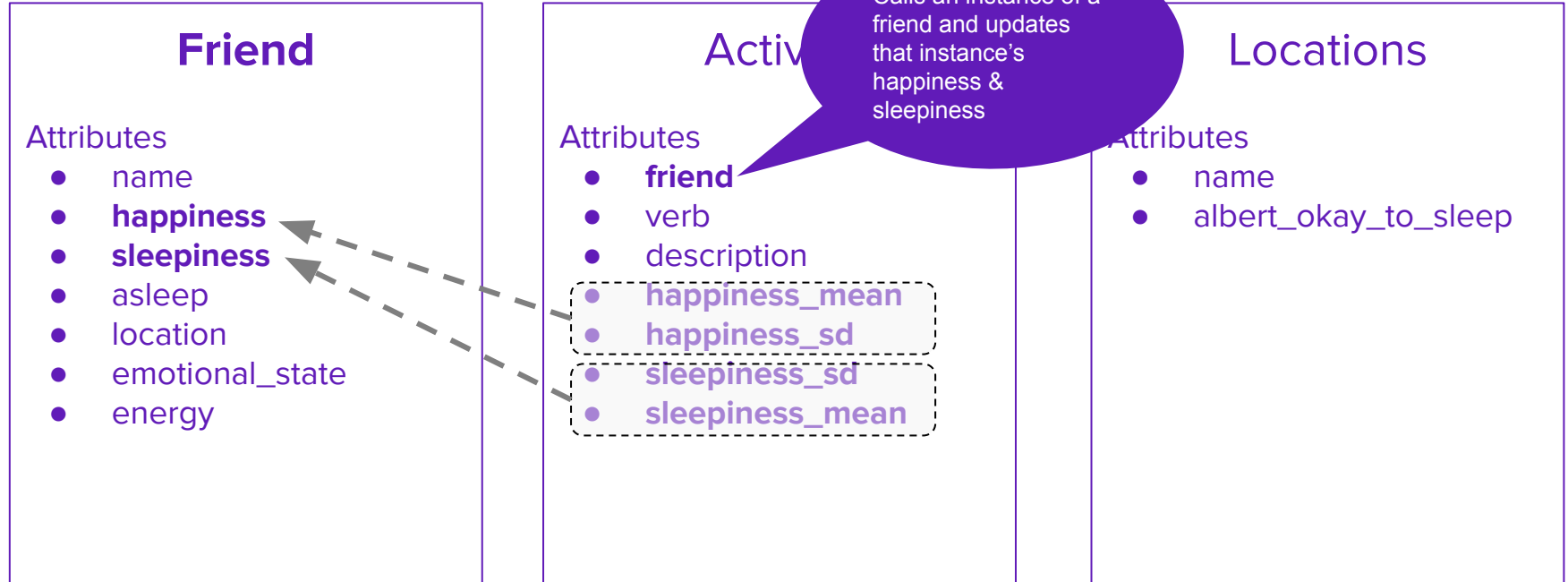
- friend
- verb
- description
- happiness_mean
- happiness_sd
- sleepiness_sd
- sleepiness_mean

Locations

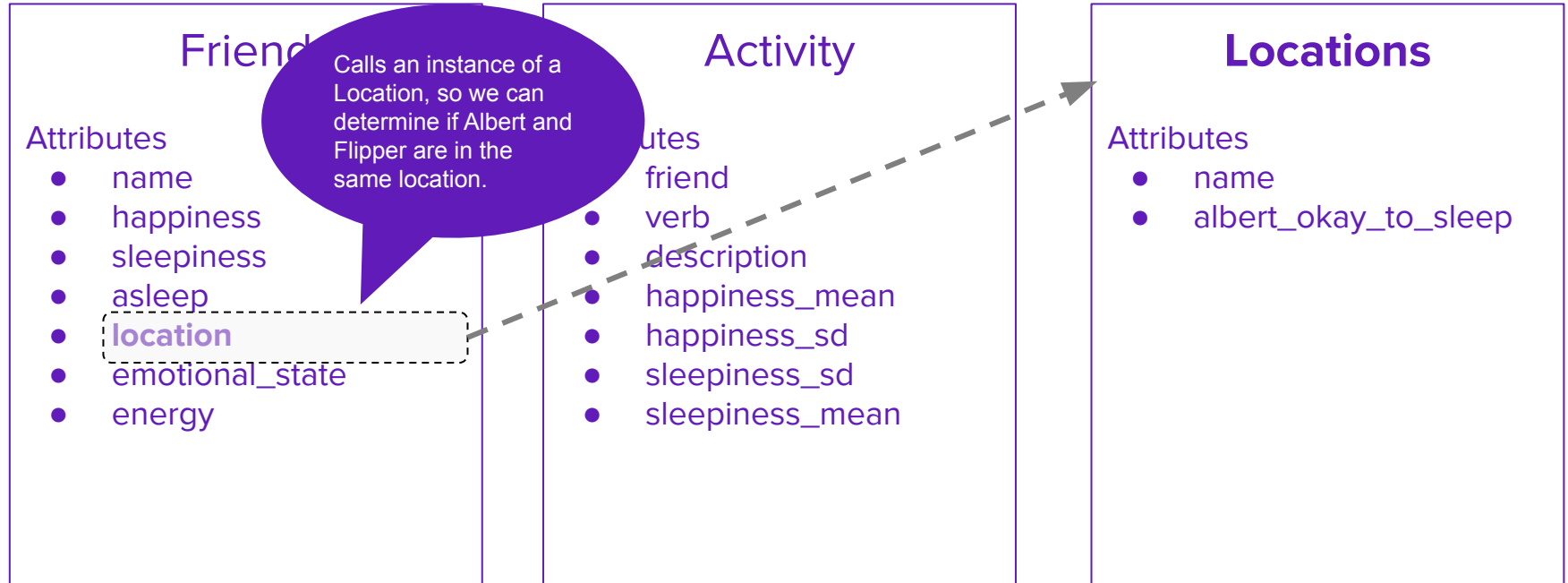
Attributes

- name
- albert_okay_to_sleep

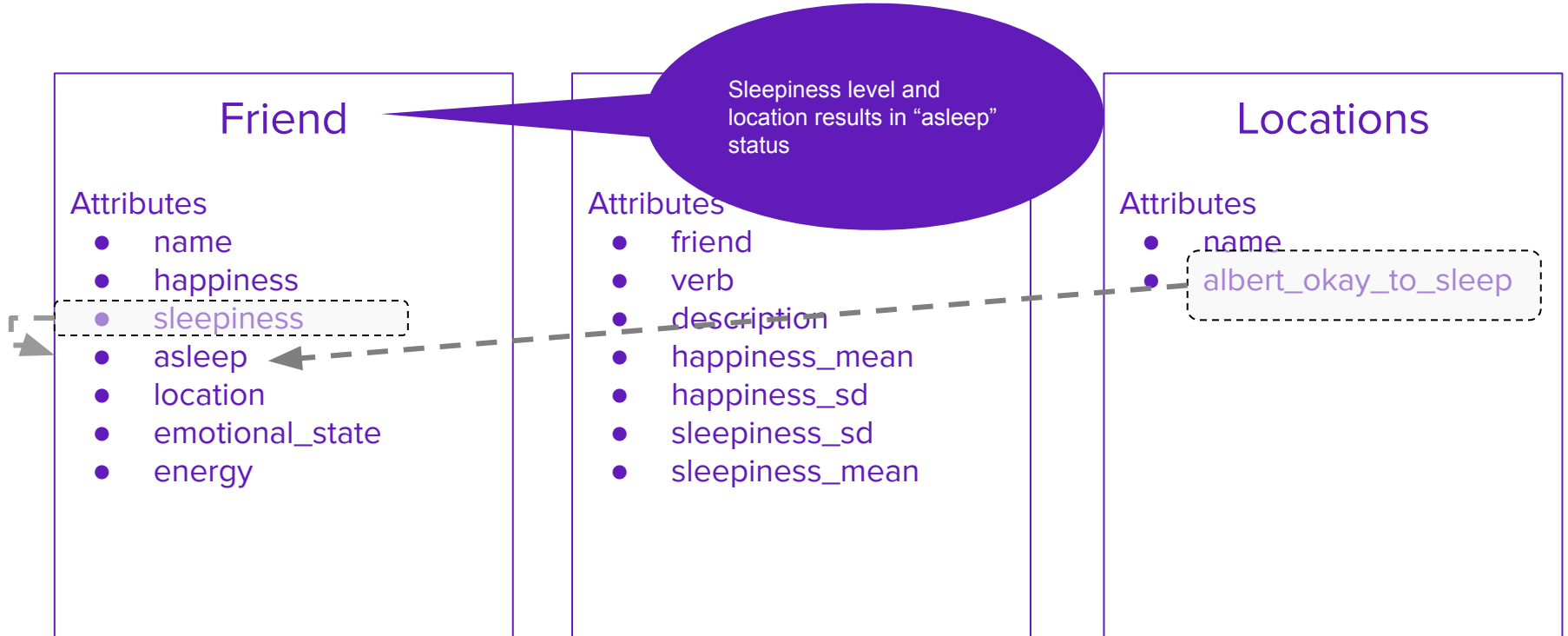
Classes



Classes



Classes



Reflections



Challenges

- **Handling the sleepiness and the sleep status**
 - It was harder than I anticipated. Sleep took various forms: Level of sleepiness; an Activity class that describes the action of sleeping, which in turn affects the sleepiness; and the current and previous state (asleep or awake.) Because these terms were so similar and they closely interacted with each other, it was hard to keep track of these different concepts. Next time, I'll label them better.
- **Random distribution**
 - I used numpy to sample from a normal distribution, because I wanted the incremental happiness or sleepiness to vary from turn to turn. Selecting the right mean and standard deviation was important in order to not make a Friend sleep indefinitely.

Challenges: Narrative

- **Sentences**
 - Because this is a text game, the sentences had to flow naturally. So picking out the correct words or phrases to pass in as variables into string templates required a lot of thought.
- **Sleep Status**
 - It didn't make any sense for Albert to fall asleep again when he was already asleep. So making the narrative and the sentences flow naturally required more logic.
- **Natural Behaviors**
 - The logic of what behaviors are possible or irrational quickly became very complex. For example, it didn't make any sense for Albert to sleep in the backyard, because he was not a cat. It also didn't make sense to inform the user of the change in Flipper's happiness when the user or Albert couldn't even see her. Also, for either Friend to move from one location to another while fast asleep did not make any sense. Dealing with these little nuances took me a long time and it's still a work in progress.

What I Competed vs. Plan to Later

- I was able to successfully pass attributes back and forth between Locations, Activities, and Friends. The random factor that determines the incremental happiness/sleepiness or asleep status was successful, too.
- I would love to spend more time ironing out the language that gets displayed, though. I want to make it sound more natural for human users. For the purposes of this project, which is to be comfortable with the concepts of OOP, this is satisfactory.