

## **Project Proposal**

### **TP3 Updates:**

- Gift method uses algebraic equation based on whether or not cat is visiting their favorite item, and the distance from the cat's favorite position

### **TP2 Updates:**

- Backtracking:
  - When a cat visits their favorite item, they will give the user a lot of goldfish.
  - When all 5 positions are filled, the user is asked to choose 5 cats. Using backtracking, the game tries to create an optimal yard arrangement where all 5 cats visit their favorite item to give as much goldfish as possible.
- Local save:
  - Users can save their progress

### **TP1 Updates:**

None

## **Project Description:**

Cat Collector, a simulation game where the player buys cat toys and food to attract cats to their yard. Cats will reward the player with currency to buy more cat toys that can attract cats of different rarity.

## **Similar Projects:**

Cat Collector is based on the existing game, Neko Atsume. They are similar because they are both cat simulation games with the same goal, creating a cute yard, collecting as many cats as possible, and collecting rare cats. Cat Collector is different from Neko Atsume because instead of the player waiting hours for cats to appear in the original game, there is a sleep mode where the player can speed up time if they are feeling impatient.

## **Structural Plan:**

There will be a class for different cats to indicate if they are currently visiting the yard, their name, breed, personality, rarity, number of visits, most used toys. There will be a super class for extra rare cats with unique features. Methods for the cat class include visiting and generating gifts. There will be a class for the user's inventory of cat toys and food. Methods in this class include buying items, placing toys in the yard based on available space, and removing the toy from the yard.

The game will be displayed with 112 graphics. The user will use mouse presses and keys to navigate through the game.

### **Algorithmic Plan:**

The most difficult part of the game is implementing OOP successfully, and making it more complex than what was taught in the class. I think it will also be tricky to connect OOP to 112 graphics.

### **Timeline:**

**Sunday, November 20:** Finalize all features and make sure OOP is working

**Thanksgiving Break:** Work on drawings

**Sunday, November 27:** Finish UI and flat drawings (no animations):

**Wednesday, November 30:** Successful working demo without bugs

**Wednesday, December 7:** add any extra animations/details if wanted to make smoother user experience

**Version Control Plan:** I can back up my code on my external hard drive. It has enough space because it is meant for photography.

**Module List:** None

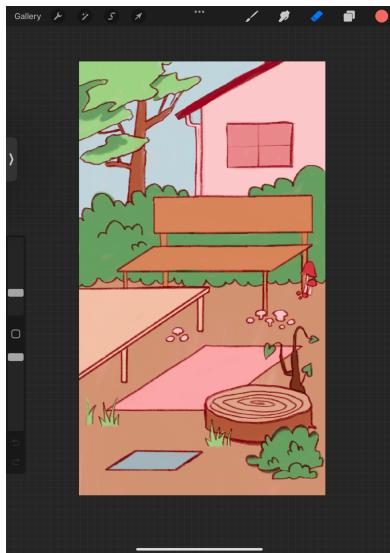
**Preliminary Code:** The file is called “nekoatsume.py”

### **Storyboard:**

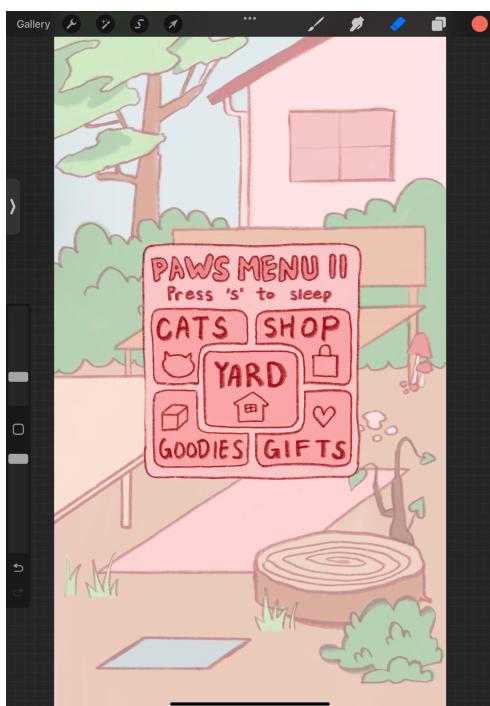
Initial opening/instructions page



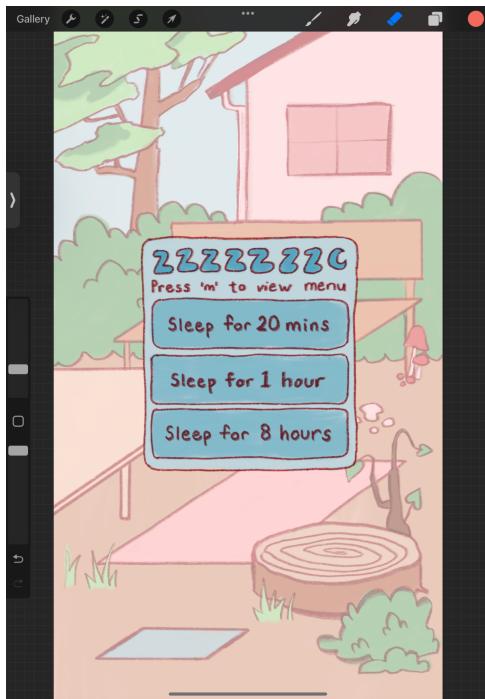
Yard



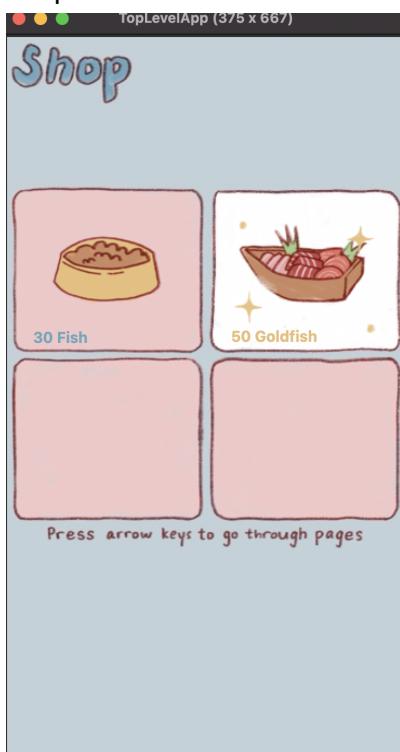
Menu



Sleep menu pop up



## Shop



Cat profile

