

CART 263: Project 02 - Proposal

Concept

As someone who enjoys simple little games, I wish to recreate a very common simulation many may be familiar with. I want to program a simple simulation where the user interacts and takes care of a virtual pet. The user will be able to feed the pet, customize it (room and pet), and do actions (pet, clean, play with a toy, etc). There will be different status bars that move depending on the interactions (ex: can lose or gain happiness/hunger/cleanliness/energy).

The simulation will be presented as multiple windows on a single screen. There will be a window for the visuals of the pet (moving around inside a box), the status bars, a list of user action options, and a log of all the interactions (user actions, pets mood)

I believe this type of simulation is an appropriate theme that encompasses some of the material we have learned this semester. For example, using the [web storage API](#) to store the pet's data (status, name), using [JSON file](#) to organize the lines of text (describing interactions in a log), and a lot of the concepts learned in [jquery-ui](#). I plan to pay close attention to the design and fluidity of the layout & user interface (CSS) and make the program look pleasing and easy to understand.

Challenges

- Linking the sections together (user action -> status bar -> pet action -> log)
- Implementing the "action log" that records/updates user and pet actions with the corresponding lines written (in JSON). Examples:
 - You patted ___ on the head. +5 happiness
 - You fed ___ pizza. -10 hunger, -5 cleanliness
 - ___ is bored... -1 happiness

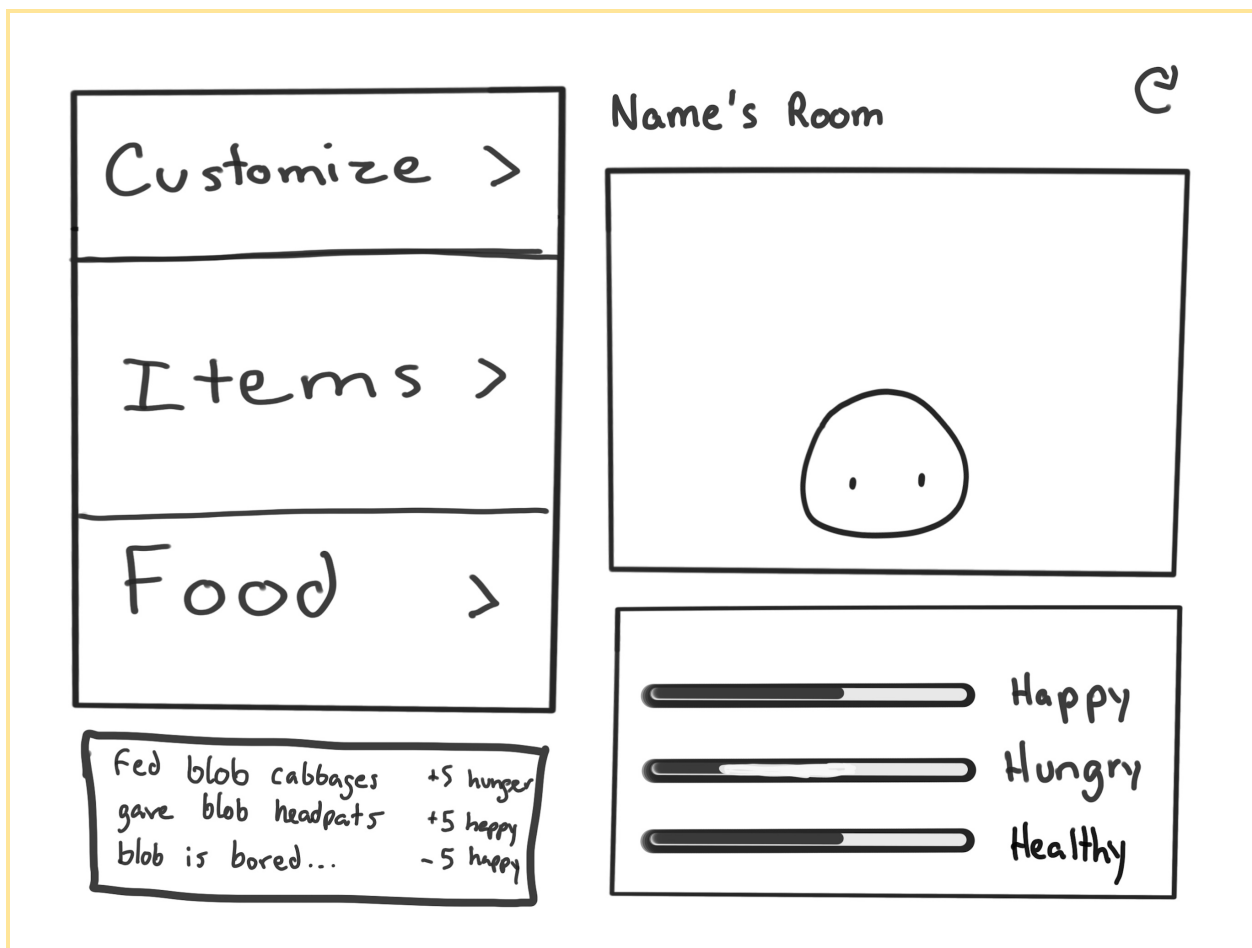
- Coding time-based interactions: having the pet be affected by real-time things
 - Leaving it idle for too long makes it bored/lonely/hungry
 - Having the pet run away/die if left alone for too long (maybe even when the user is not active on the page)
- Implementing animations to the program so it is not too static:
 - Pet animations (moving around, expression change, sleeping if no action)
- [MAYBE] advanced action interaction (drag & drop food/items directly on the pet)

Plan

Main Aspects	<ul style="list-style-type: none"> • Interface (menu): 4 windows <ul style="list-style-type: none"> ◦ Interaction Lists (UI) <ul style="list-style-type: none"> ■ Customization tab ■ Food tab ■ Action tab ◦ Pet's room <ul style="list-style-type: none"> ■ Pet display <ul style="list-style-type: none"> • Different facial expression based on mood • Visual: accessories from user customization ■ Name display <ul style="list-style-type: none"> • Editable text on top of the box (___'s room) ■ Room display <ul style="list-style-type: none"> • Background of the box • Visual: color is chosen by the user ◦ Log <ul style="list-style-type: none"> ■ Add a line of text upon each action or system message (pet mood) ■ overflow: content disappears out (pushed at the top) ◦ Status Bar <ul style="list-style-type: none"> ■ Three "progress" bars ■ Label changes depending on the points it has on 100
--------------	---

	<ul style="list-style-type: none"> • 0-50 = hungry / unhappy / dirty • 50-100 = full / happy / clean • Storing all the data (keeps the same pet when the user revisits) • Link each action to the meters • Reset for a new pet (forced reset, or when the current pet is gone)
Additional Features if time and ability permits	<ul style="list-style-type: none"> • Cooldown so the user cannot spam an interaction • Advanced action interaction (drag & drop food/items directly on the pet) • Pet movement

Sketches



UI for interactions

Customize >

change blob's look
blob's room

Items >

different interactions
(+/- happy, +/- healthy)

Food >

(+/- happy, - hunger)

Log of
events

Fed blob cabbages	+5	hunger	points
gave blob headpats	+5	happy	
blob is bored...	-5	happy	

refresh



Name's Room

← User chooses



Status bar's (depends on interaction)



Happy → toys
(items)



Hungry → food



Healthy → items