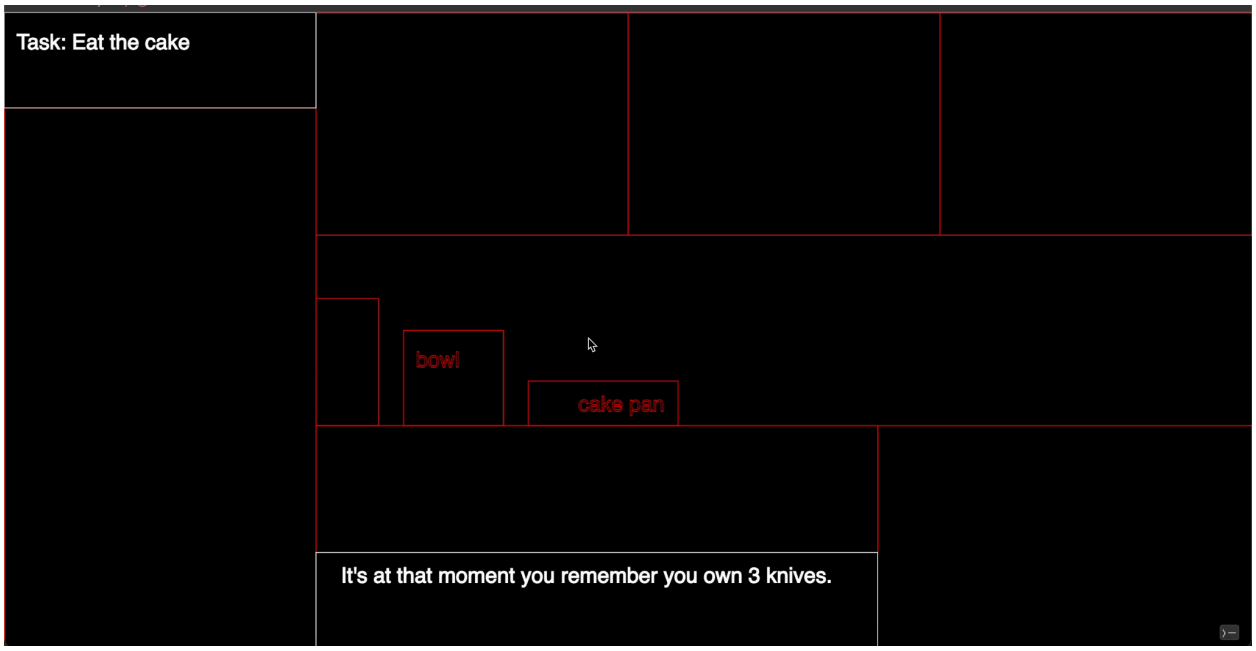
Blind Bake

By Rachel Yang

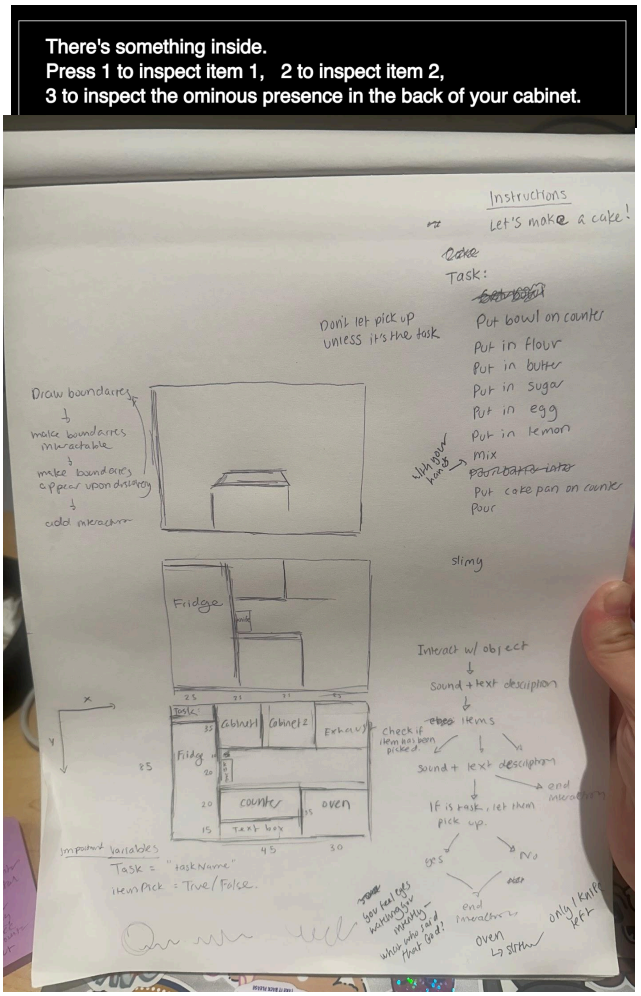


Since this was an audio-focused project, I decided to create a baking game where you rely entirely on audio and tactile descriptions.

What is this about?

Gameplay Video (ft. my friend who playtested) (also note: gross, slimy sounds)

A while back, I brought up the idea of a baking game to my friends. Each of us would be revoked of one sense (hearing, sight, speaking, etc.) and together, we had to bake a cake. While we've yet to do this in real life, I took inspiration for this project where you bake a cake relying entirely on auditory and tactile feedback (though I added the outlines of furniture for a bit of visual appeal). Basically, it's a point and click game where you're given a series of tasks to complete (ie gathering ingredients, baking the cake, etc).
While I initially intended this to be a fairly tame experience, at some point, the thought of an eldritch creature accompanying you cropped up and I worked them into the game. As you rummage through your cabinets for ingredients, you might find the odd thing here and there (creepy breathing, slimy creatures, the works). These are one time encounters and the options to interact with them disappears afterwards, unlike other interactable objects. When my roommate play-tested this, she was very perturbed by the disappearance of the creepy breathing ("I liked it better when he was in the cabinet.") I also noticed that the lack of visuals helped with the horror factor. My roommate was anticipating a jumpscare that didn't exist.



Process

My code for this game can be described as a tangle of wires. I did do a bit of planning before writing any code, but the rest of it was dealing with unforeseen situations, hence the mess. I designed the layout of the kitchen (I originally was going to allow player movement, but decided a static view of a kitchen would be simpler) and drew it in p5.js. Since each furniture is interactable, it helped to know where the boundaries of each object was. I was originally just going to use the boundaries to make debugging easier and remove them for the final game, but then I thought looking at a black screen would be too boring, so I made the boundaries appear permanently once you've discovered a new object. Then, depending on where the player clicked on the screen, they would get a text description of whatever they clicked. I used a function that made one character in the text string appear after the other, creating a kind of typewriter effect.

The rest of the coding was dealing with unaccounted scenarios, which is to be expected for a game like this.

For example, if the player interacted with a container like a refrigerator or cabinet, they would find several items inside and be prompted to pick an item (ie Press 1 for item 1, press 2 for item 2, etc.) I realized that I couldn't take user input within a nested if statement under the if statement that checked which furniture the user clicked since it would require the KeyClicked function, so I had to put the secondary interactions in an entirely different area. The rest of the code followed a similar pattern of bouncing between 5 different places of the script because everything was so interconnected.

I tried restricting choices to reduce unexpected user input. I didn't want to deal with an inventory system, which would've been needed if players picked up an item that they didn't need for their current task. Therefore, I only allowed picking up an item if it was relevant to the task. I had a boolean variable for each task to indicate if said task was active (there would only be one variable that is true at a time). Some actions would only be available if their respective task variable was set to true, which made things easier to control.

After all the game components were done, I added in the audio for the game. Originally, I wandered around my kitchen and recorded all the sound effects myself before quickly realizing that a) the sound quality was bad and b) most of the sounds were not distinct enough to give players much information about the respective item. Therefore, I pulled most of the sound effects off pixabay's sound library. I kept in some of the sound effects I recorded myself. For example, the sound of the butter being deposited into the bowl was my rubber frog toy being dropped onto a table.

bag_flour.mp3
beep.mp3
breathing.mp3
breathing_final.mp3
butter_fridge.mp3
butter_plop.mp3
counter_slide.mp3
cutting.mp3
dishes.mp3
eating.mp3
egg_carton_open.mp3
egg_crack.mp3
exhaust.mp3
exhaust2.mp3
flour_plop1.mp3
fridge_open.mp3
knife.mp3
metal_exhaust.mp3
metal_tray.mp3
mixing_batter.mp3
open_cabinet.mp3
oven_door.mp3
pour.mp3
pouring_sugar.mp3
putting_down_bowl.mp3
slithering.mp3
squelch.mp3
sugar_bag.mp3
ticking_oven.mp3
tray_counter.mp3

Conclusion

Overall, this project wasn't anything new as far as what code I used. It was, however, a lot harder to keep track of everything going on and how it would affect other interactions in the game. If I had more time, I would've made this game more user-friendly. The text descriptions currently change based on a timer, which means the text might change too fast or too slow for the player. Also the text does freak out a bit if players don't wait for the text to play out completely or switch between object interactions too quickly. But as it stands, this game functions without any huge bugs, so I'm satisfied.

Links

Code	https://openprocessing.org/sketch/2908742 https://drive.google.com/file/d/1XyR3goERHm2una9BodzjZ_DfILsdmffu/view?usp=sharing
Final video	https://drive.google.com/file/d/1kj8LPInheWzp9g_WQ9b9q3Xu7MLVUU8S/view?usp=share_link

Sources

Sound effects	https://pixabay.com/sound-effects/