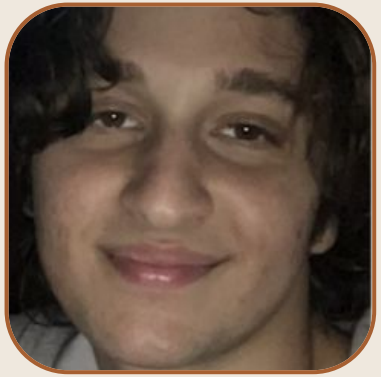
Assignment 8:

Hi-Fi Prototyping

Hamidou Guechtouli
Blaine Wells
Melody Fuentes
Dante Danelian

Team **Arbor**



Hamidou Guechtouli

Blaine Wells

Melody Fuentes

Dante Danelian


arbor

Problem/Solution Recap

Problem

People do not have an **efficient** and **dedicated** method to recall past positive memories.

Proposed Solution

A **virtual** tamagotchi-style **garden** where you can plant and revisit good memories in the form of trees.

TABLE OF CONTENTS

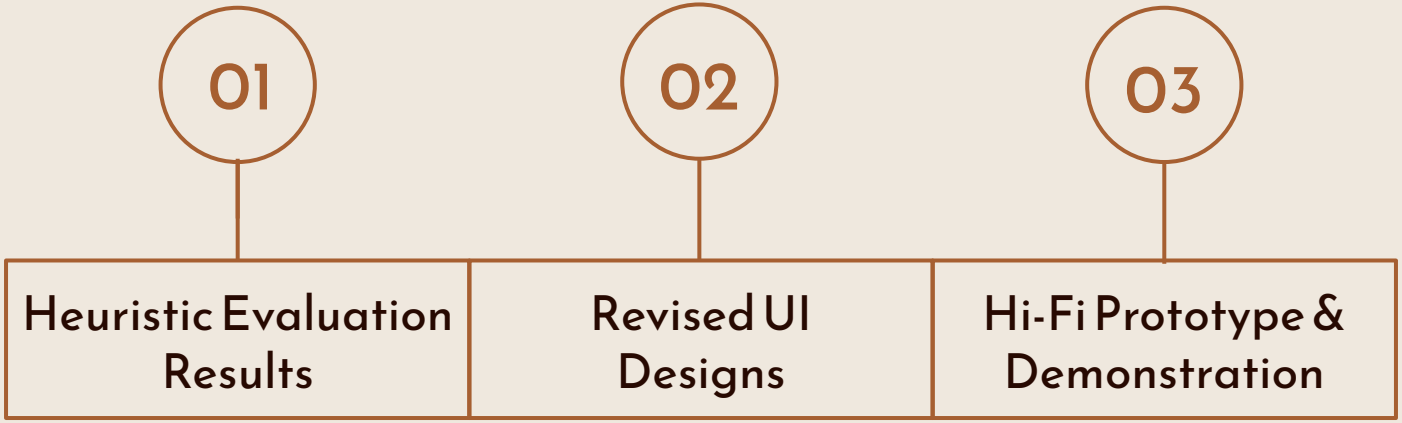
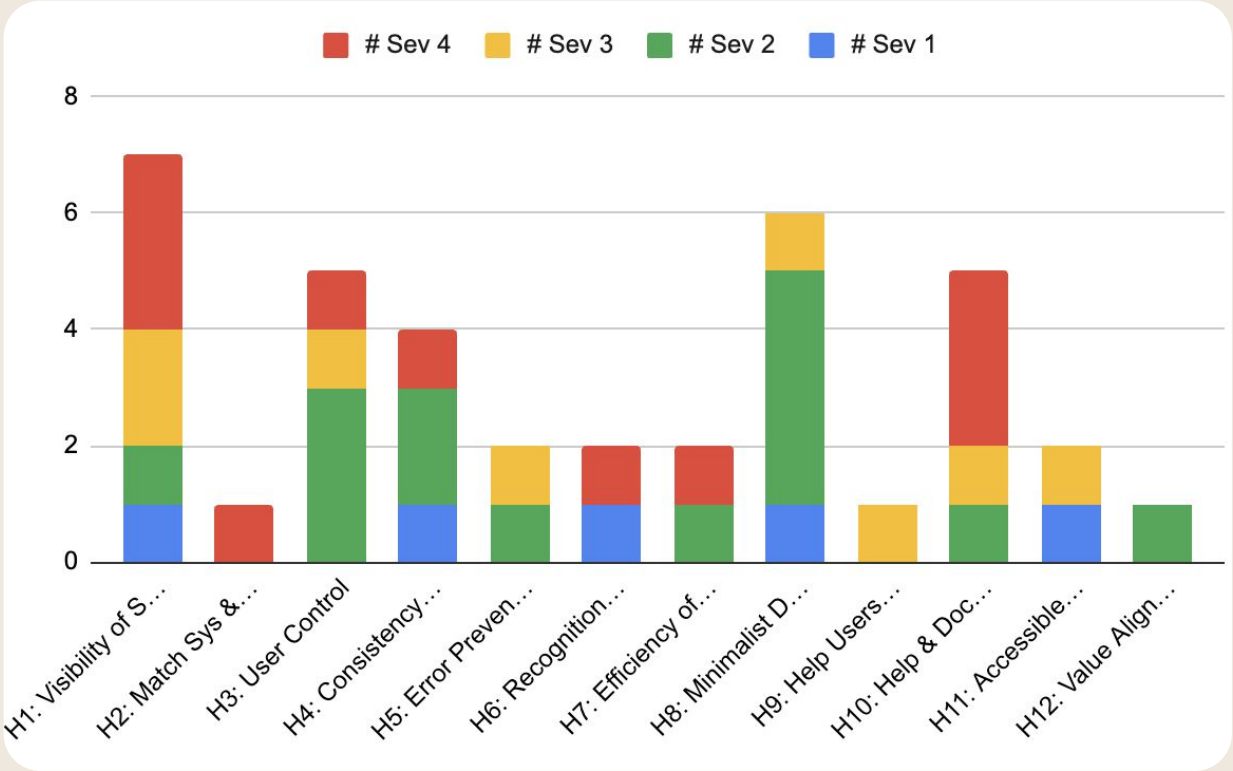


TABLE OF CONTENTS

High-Level Summary



Severity 4: 11

Severity 3: 8

Severity 2: 14

Severity 1: 5

High-Level Summary

Trends display a need to focus on:

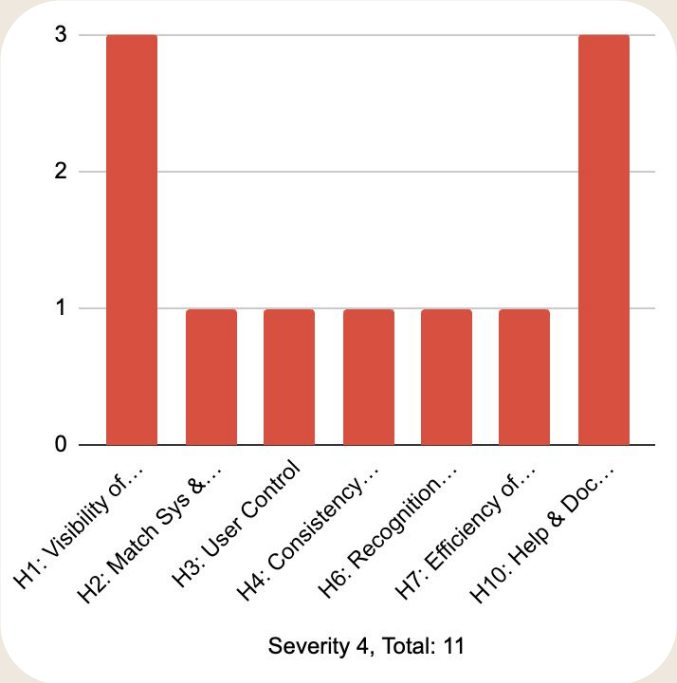
- **H1: Visibility of System status (7)**
  - Guidance/Labels
- **H3: User Control (6)**
  - Unimplemented features
  - Fix exits (back buttons)
- **H8: Aesthetic & Minimalist Design (5)**
  - Changing/adding labels
  - Centering text and sizing in phone
- **H10: Help and Documentation (5)**
  - Guidance, navigation on where to go.

Key Takeaways

The **most common** issue is finding what to do next in the world task-wise.

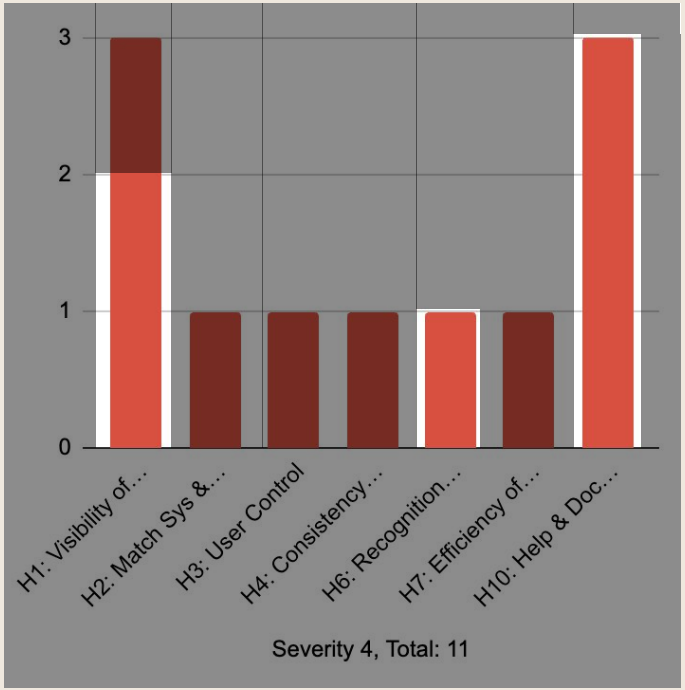
From there, most violations are **cosmetic**, with a focus on labels, aesthetic consistency, exits/buttons, and unimplemented features (i.e. adding friends)

Sev. 4 Trends



Severity 4:
11 violations

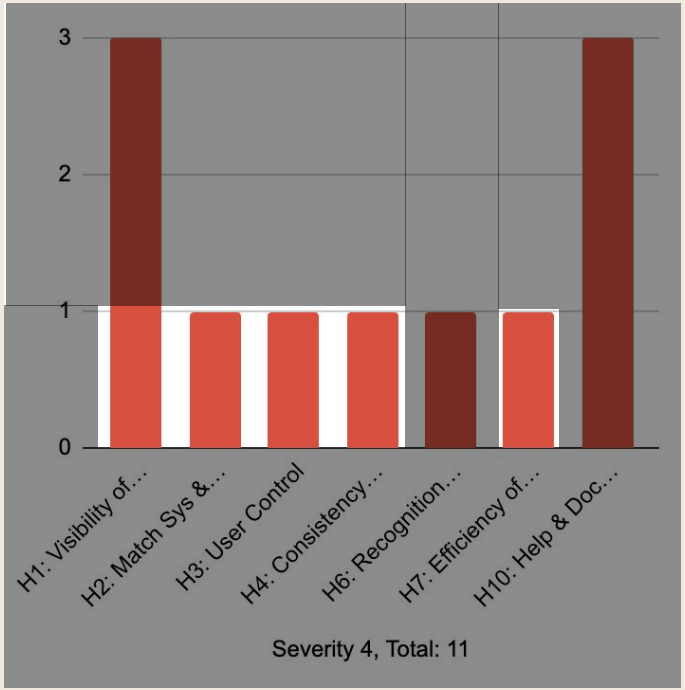
Sev. 4 Trends



Calls for major changes (6): Clarity

- Confusion on where to go and what to do next
  - Need for guidance before/during/after every task (6)

Sev. 4 Trends



Calls for major changes (6): Clarity

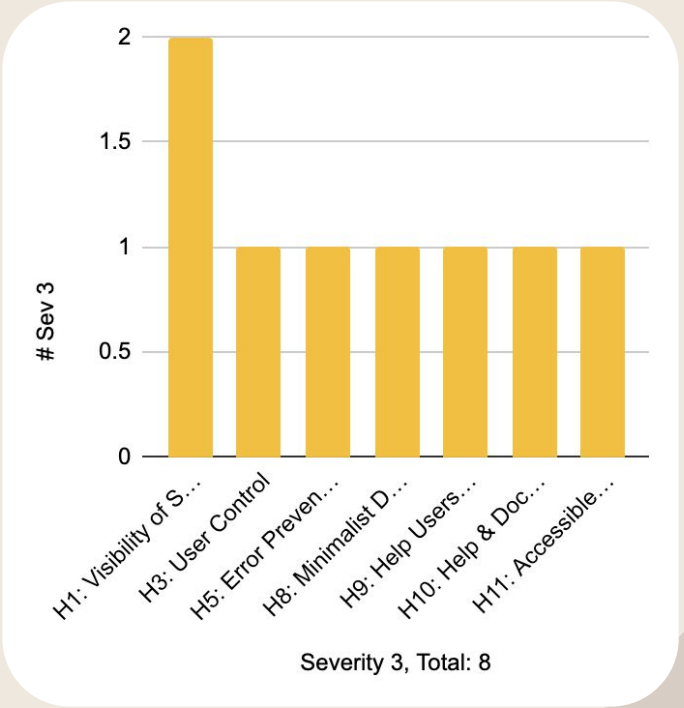
- Confusion on where to go and what to do next
  - Need for guidance before/during/after every task (6)

Calls for minor changes (5):

- Changing one minor UI.
  - Short labels (i.e. “Next” to “Skip” when logging a memory) (3)
  - Where one exit button leads (1)
  - A single animation (1)

Severity 3:
8 violations

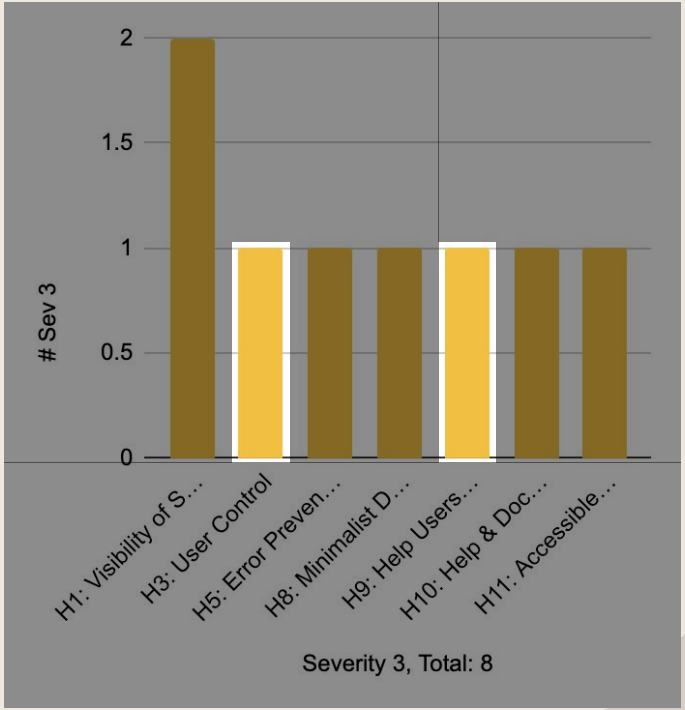
Sev. 3 Trends



Calls for major changes (2): User Options

- Addition of unimplemented features. Great for confirmation.
  - Option to delete memories (1)
  - Buying more than one fruit (1)

Sev. 3 Trends



Calls for major changes (2): User Options

- Addition of unimplemented features. Great for confirmation.
  - Option to delete memories
  - Buying more than one fruit

Calls for minor changes (6):

- Changing one minor UI.
  - One-word labels/names (i.e. Changing name of “Almanac”) (3)
  - Changing image placements (2).
  - Centering Text (1)

Sev. 3 Trends

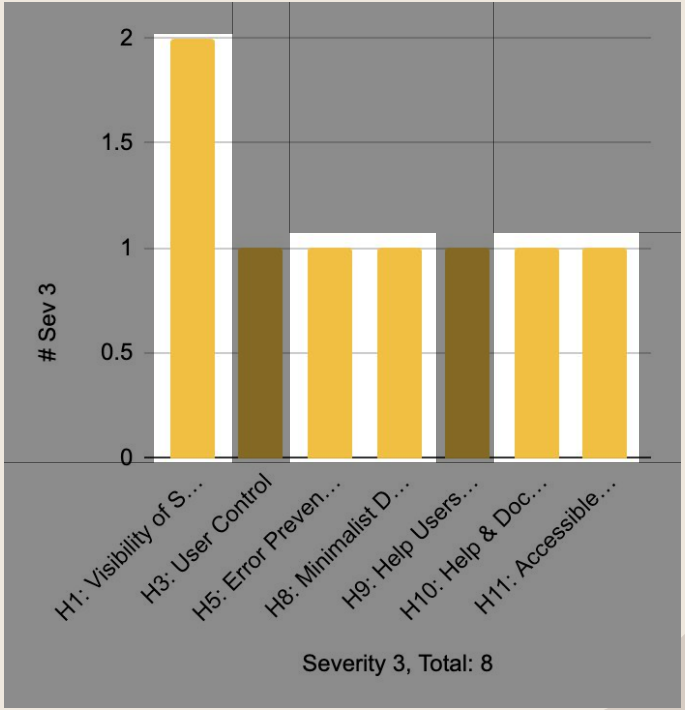
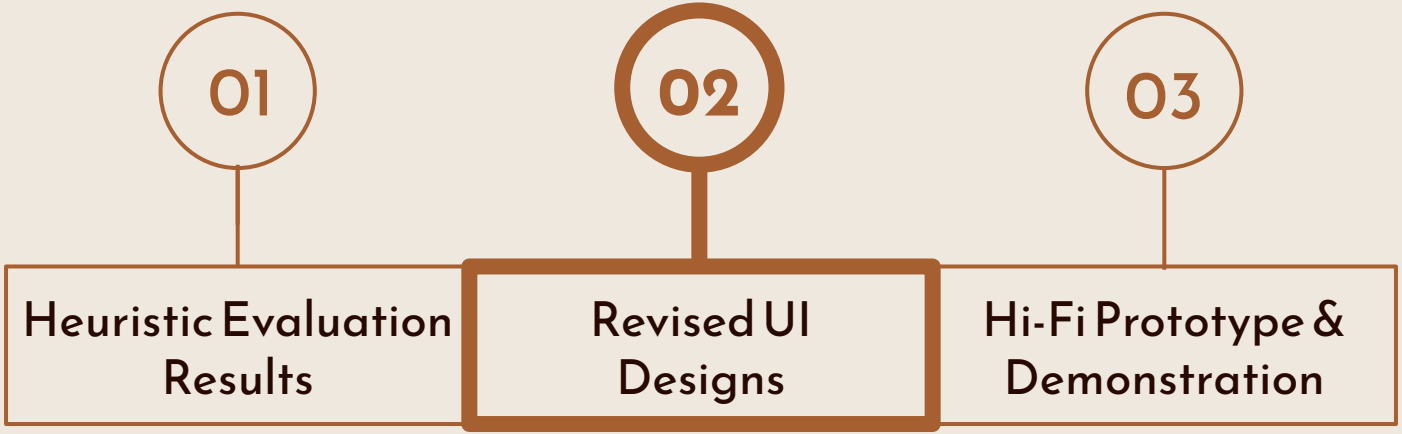


TABLE OF CONTENTS



Our Usability Goals

Efficient

Robust

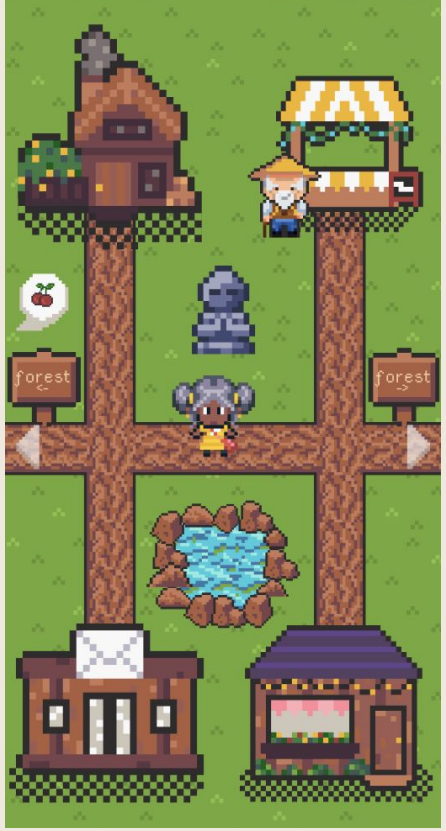
Fun

Product Revisions

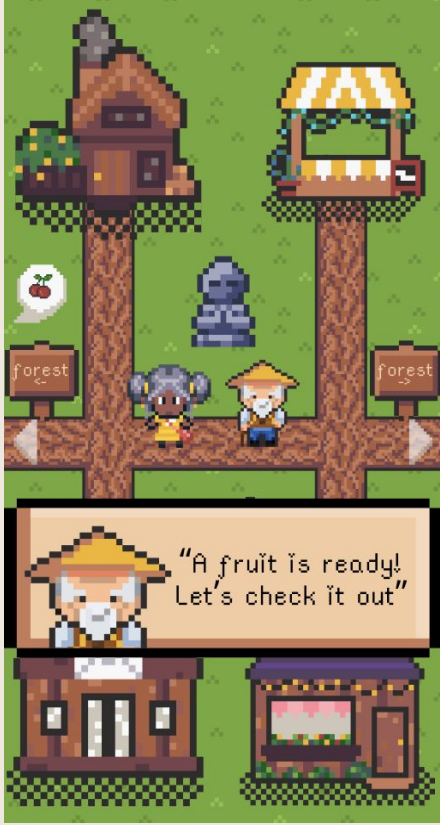
1

Begin Arbor with a **story-like tutorial**, led by the farmer, to introduce the world.

- The most common feedback is not understanding world & what to do.
- This addresses all major changes for sev. 4, as it provides all needed context + next steps.
- Though should only be needed once, players will have the option to ask the farmer within any task for guidance if need be for accessibility (i.e. cognitive disorders)



Before



After

Usability Goal: **Efficient, Robust, Fun**

This change removes all guesswork from user of what to do next and why- allowing for a faster learning curve, fewer errors, and creates an immersive experience into the Arbor world

Product Revisions

2 Animation via Gif

- Arbor wants users to develop a bond with the spite, movement creates a sense of life in the characters and world.
- Also makes clear what is clickable, which addresses common feedback for clarity on what to click in severity 4, 3, and 2.
- Everything clickable in the world will have a small gif, and the character will to walk.

Before

After

Usability Goal: **Robust, Fun**
The clarity of what is able to be interacted with results in less errors by reducing the amount of random-clicking a user may have previously done. Animations are fun, adding an exciting livelihood to the app.

UI Revisions

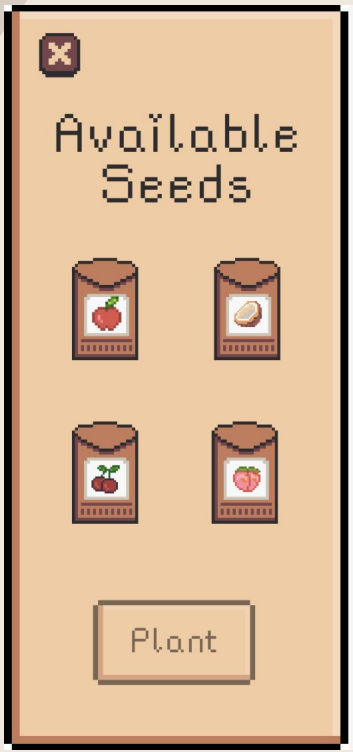
1 Making better use of space by removing unnecessary constraints.
I.e. **Mail task**: screens now a messaging style, switched placement of images for clarity

- Reduces user friction as user needs to scroll less, more important info above-the-fold
- Image layout follows hierarchy of importance- memory first, fruit sent second
- Addresses HE feedback of aesthetic consistency and layouts in Sev. 3, 2, and 1, all screens look more consistent and cleaner.

Usability Goal: **Efficient, Robust**
With a reduction on user friction, the time it takes to slow is severely lessened. Image-switch reduces mixing up memories from different people.

UI Revisions

Before



After

2 Adding better labels.

I.e. **Planting Seed screen**: Labels now includes name and how many seeds a user has in their inventory

- Addresses feedback in Sev. 4, 3; visibility of system status and accessibility, users regardless of ability can know what fruit they have and how many.
- Increases clarity of the app, removes the guesswork from the user, thus creating a smoother, more intuitive experience.

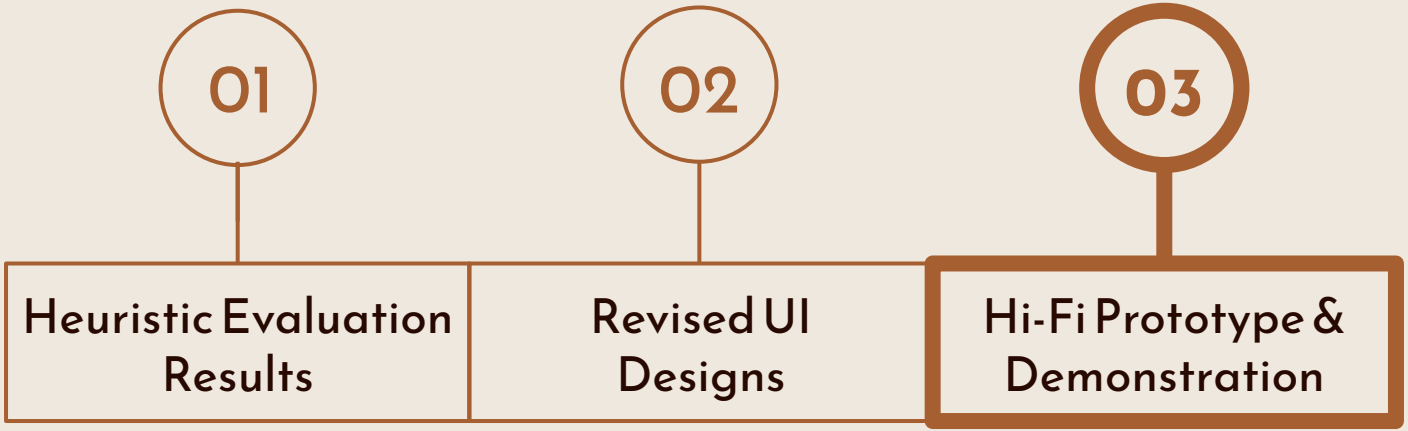
Usability Goal: Robust

This change reduces errors by removing the need for users to remember their inventory. Labelling makes the app more accessible for people of all abilities, reducing overall errors.

Usability Goals Progress

- [x] Efficient
- [x] Robust
- [x] Fun

TABLE OF CONTENTS



Our Tools

- In implementing our Hi-Fi Prototype, we relied on React Native to host our application, and Piskel for the creation and animation of pixel art assets.

Implementation Timeline

Implementation Timeline

Today

- Functionality for simple task (picking fruit)
- Almost complete with moderate task (planting seed)
- Updated graphics and completed animations

To-Be Implemented Features

While we have the assets for all the tasks completed, and are closing in on two thirds of the tasks being functional, there are still features left to implement. Namely:

1. **Functionality for Complex Task:** Also happens to be the simplest to implement of all of our tasks.

2. **Tutorial Dialogues:** Pop-up text boxes for the tutorial, where the farmer greets you and helps you learn how the farm works.

3. **Animation:** We want to animate all tappable objects. Currently, we have created the gifs necessary, and just need to embed them in place of the png files.

Implementation Timeline

Today

- Functionality for simple task (picking fruit)
- Almost complete with moderate task (planting seed)
- Updated graphics and completed animations

Weekend

- Functionality for moderate/complex task (sharing fruit),
- Tutorial dialogues
- Complete animations for tappable objects.

Implementation Timeline

Today	Weekend	Next Week
- Functionality for simple task (picking fruit) - Almost complete with moderate task (planting seed) - Updated graphics and completed animations 	- Functionality for moderate/complex task (sharing fruit), - Tutorial dialogues - Complete animations for tappable objects 	- Finishing touches - Polishing user experience - Updating any assets/animations that seem out of place

Unimplemented Features

There are also some features we do not plan to prototype, as they do not relate to the tasks but would require extensive additional work:

1. **Town marketplace:** We have chosen not to implement the market, where users can buy outfits or pets for their character. This would require the creation of many more assets, as well as a whole new task.
2. **Customization of the Garden:** Our Hi-Fi prototype does not support full customization of the character or garden, as this would require a back end to track these changes.
3. **Almanac:** Typically, the user would be able to access a bank of all their memories in chronological order. However, for similar reasons to the above, we do not have the back end to implement this.

Wizard of Oz Techniques

1. **Tree Growth:** Instead of implementing a back-end method for tree growth, where fruits become naturally ripe over time, the user starts the app with a fruit ready to pick. This helps expedite the demo so the user doesn't have to wait to start the simple task.

2. **Sending Mail:** Since we don't have a user base, the social component of the app is simulated. Thus, sending mail to a friend is a function we are simulating for the sake of the prototype.

Hard-Coded Aspects

- Several aspects of our prototype were hard-coded, so as to maintain the audience's immersion in the product while also illustrating use cases for the product.

1. **Multi-media Inputs:** Our prototype does not have an extensive back end for supporting multimedia inputs, and thus we hard-coded text, photo, and embedded media when planting seeds and reviewing memories.
2. **Social Component:** Since our app does not yet have a user base, we hard-coded friends into the system. Thus, if you tap on the post office, there is hard-coded mail from friends, and a list of contacts you can send seeds to.

Hard-Coded Aspects

3. **Customization of Garden**: Seeing as our Hi-Fi prototype does not support full customization of the character or garden (as this would involve creating a backend to track these changes), the character's outfit and the layout of the garden are hard-coded in, as if the user set up their app this way.

4. **Seed Inventory**: For similar reasons as the previous point, we had to hard-code the seeds in the user's inventory at the time of planting or sending mail, as keeping track of the seeds would require a more significant back-end.

DEMO

Questions?

APPENDIX

Smaller UI Revisions

3. Changing Names
4. Revisit exit buttons, making sure they are appropriately routed and in place when needed
5. Mailing/Receiving fruits instead of seeds
6. Removing unnecessary decoration
7. Ensure all text is centered and font/size consistent
8. Add unimplemented features (customization, add friends, etc.)