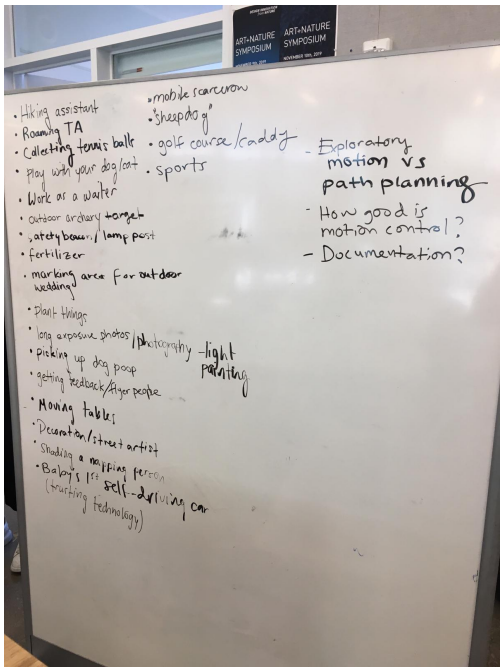
**PROJ04**


Ananya Nandy, Doug Pan, Aaryaman Sen, Lian Song

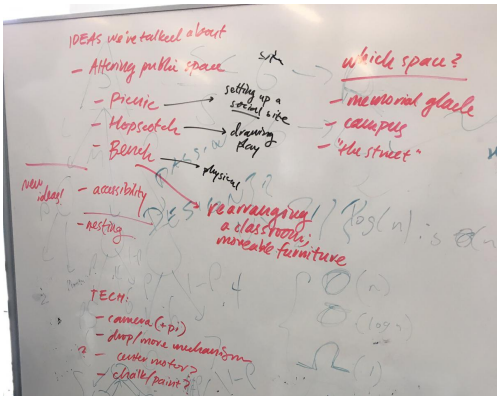
# 01 Introduction

The HoHoHoBot roams through the basement halls of Etcheverry Hall, across the breezeway to Soda Hall, and crosses the street to main campus, spreading holiday cheer across campus. It aims to bring people together through sharing well wishes and sentiments. By allowing people to record messages and listen to messages from other places, the HoHoHoBot gathers messages from one location and brings them to another in the form of a collaborative tree decorating activity.

# 02 Ideation




After our first brainstorm, we narrowed down to 3 main concepts: picnic, hopscotch, or bench. Each of these words represented a larger concept.



**Picnic**: The idea behind this concept was to bring people together physically. This would be accomplished by having the robot go around and layout a picnic blanket on Memorial Glade, bringing food and drink, and incentivizing people to sit down and enjoy time together. Then, we thought about making it be a robot-human picnic instead, in order to promote human-robot interaction and better understanding between robots and humans.

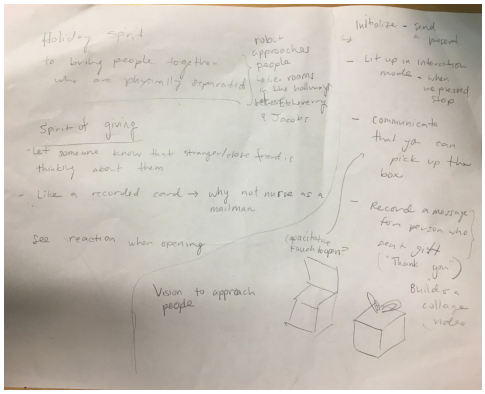
**Hopscotch**: The idea behind this concept was to change normally static public spaces. This manifested in the idea that the robot would go around and generate chalk hopscotch drawings in random public spaces. These would be temporary, adding to the public space landscape and then erasing over time.

**Bench**: The idea behind this concept was to give mobility to a normally static object. For example, if the robot were in the form of a bench, it could move around and find new locations where people might want to sit (for example, the perfect spot with a combination of shade and sun).

In the end, we went with a “Bench” concept (though not using the bench object), because we envisioned a future where robots do not look like what we think of as “robot-like” and instead, are roboticized versions of everyday objects. In addition, we wanted to incorporate aspects of bringing people together. Thinking about the holidays, we decided to use a Christmas tree as the medium for our design, changing it from a static object that people gather around at homes to a dynamic object that can come to people who are physically separated and be used to link the community together. In addition, we wanted to make sure there was an element of “giving” involved in order to incorporate the spirit of the holiday season.

# 03 Design Process

Our initial idea was to focus on giving by physically having people open a box and receive/provide a gift, with that exchange moving across a community. However, we felt that it was not realistic for people to have gifts on hand randomly.



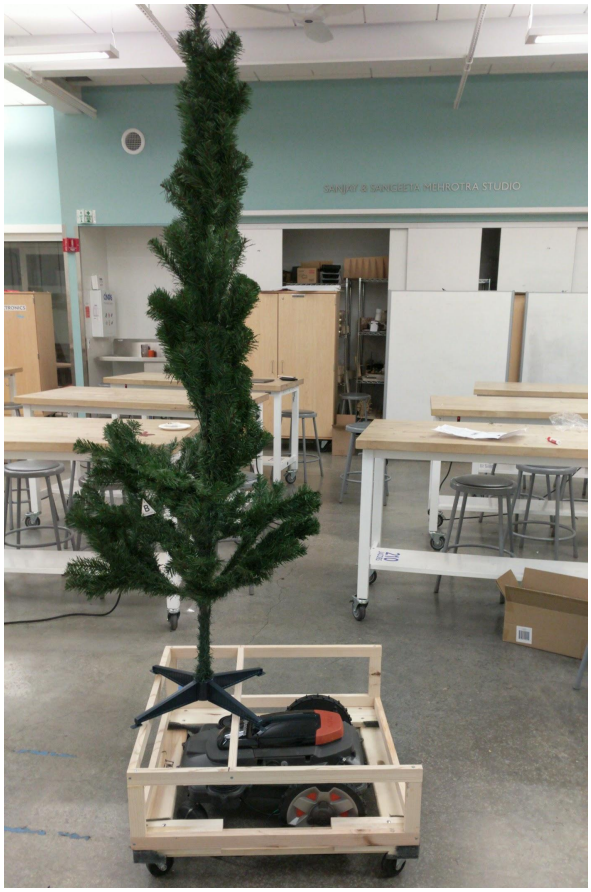
We still wanted the tree to be able to carry something from the people it had interacted with, which motivated the addition of message recording and playback. We decided to make these messages get recorded and played back with capacitive touch as a trigger in order to have the interaction follow the form of the well known activity of tree-decorating.

# 04 Fabrication and Refining the Interaction

## Fabrication

Christmas trees are tall and difficult to maneuver around when attached on top of the Husqvarna robot. We discussed several options of mounting the Christmas tree and eventually narrowed it down to two options. The first was adding several holes to the robot chassis where the tree would be a “horn” on the robot. The second was creating a separate carriage for the tree that would be driven by the robot. We decided to use the second idea because of concerns over the tree tipping over.

The carriage was created by lengthening a furniture dolly and adding a wooden frame that could hide the robot. Access to the robot’s control panel was important so the upper back member of the wooden frame was not added.



The tree and carriage were also designed to make sure it could fit through a standard door.


The carriage was then camouflaged with red felt cloth.

And controllable lights were added to the tree.


## Refining the Interaction

We spent a significant amount of time trying to design the user's primary interaction with the tree. After settling on conveying positive messages as the core purpose of our prototype, we needed a meaningful way to record and play these messages back. We used the act of decoration of the tree as a jumping-off point to allow people to tell their stories. This resonated with us, both because of the skeuomorphic nature of the interaction, but also because it

underscored the idea of community. As the tree moves along, this interaction allows users to participate in a collaborative, communal journey. The community contributes to the decoration of the tree and in doing so, spreads holiday cheer.

**Capacitive touch ornaments**

As our primary mode of interaction, we chose to focus on the ornaments used to decorate the tree. An strung-up ornament, along with the message recorded into it, makes up the quantum of information saved by a single person on the tree. The idea of using a touch of the ornament as a way to control the recording was appealing as it was extremely simple and an intuitive continuation of putting the ornament on the tree.

To implement touch control of the ornaments, we used the capacitive touch sensors on the Crickit Hat. With Adam's help, we turned each bauble into a capacitive touch receptor by filling it up with (conductive) solder wire and connecting this wire to the capacitive touch ports on the Hat. Eventually, we moved to a more robust system using the Adafruit MPR121 which allowed us to use 12 unique sensing points instead of 4. We constructed the circuit and taped it to the stem of the tree and then snaked wires that ended in loops along the branches. Hence, our final interaction was fairly simple. The user takes a bauble that ends in a metallic hook and hooks it onto a loop on the tree, much like a regular ornament. This completes the circuit and makes the ornament interactable. If the ornament is "empty", touching it prompts you to record a message. If it already contains a message, touching it plays it back.

**Sound and lights**

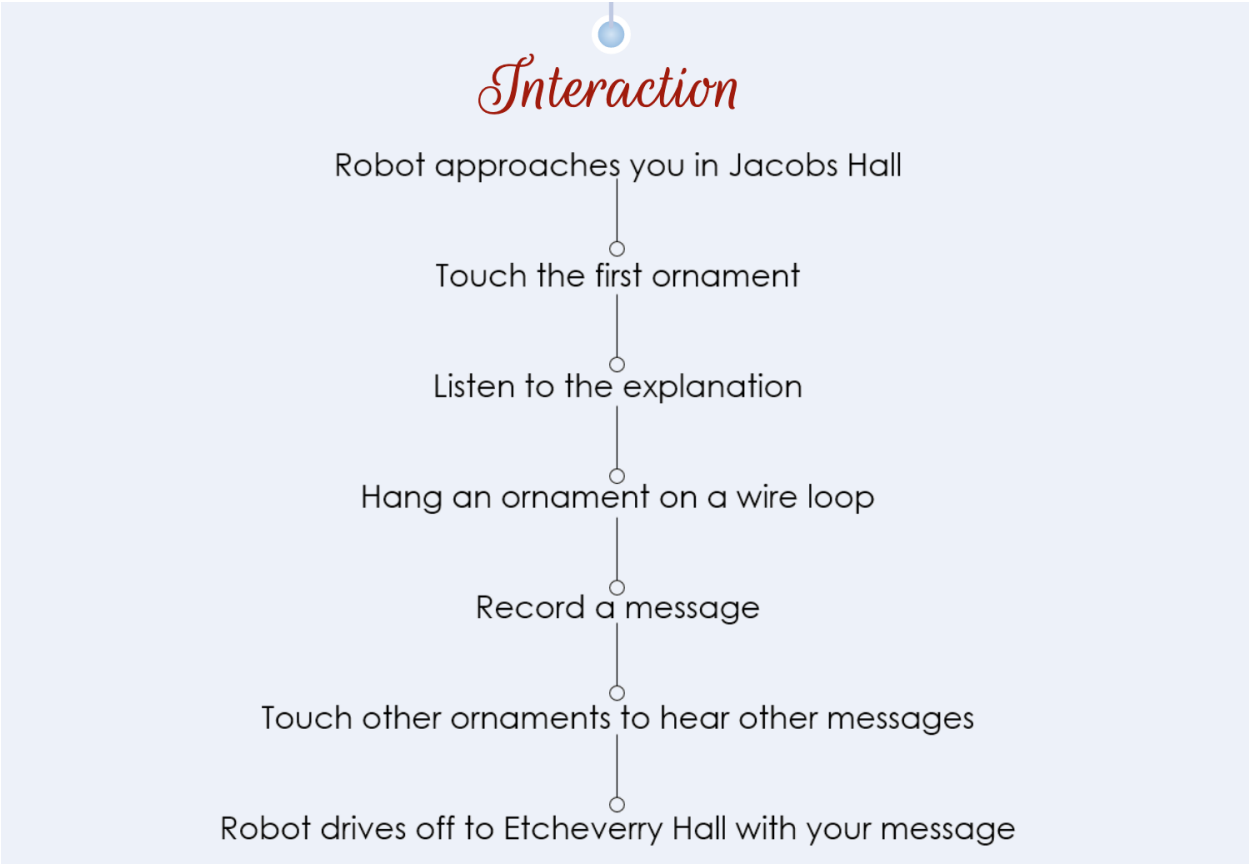
With this interaction, we were mostly concerned with recording audio input and playing it back — what was critical for us was to be able to save the audio appropriately, then play it at the right time (i.e. when an ornament was touched). For us, then, this constituted the two core interactions: one to record, and one to play.

We also included audio that would play once at the beginning of a session (i.e. the first time a user interacted with the tree) that included instructions and a short description of the HoHoHoBot — this was intended to help users quickly understand how to interact the robot, and let them know they had to hang up an ornament to begin the recording process.

Beyond that, we also supplemented voice recordings with a cute, festive backing track — it felt fitting, thematically, and added to the sense of holiday cheer.

Finally, lights were added — after all, no Christmas tree is complete without them. However, we ran into some issues powering and addressing all the lights; there was an issue with I/O bandwidth through the Raspberry Pi. As such, we kept the lights simple — the lights would just change as a block, either to green or red, depending on whether or not an ornament was being touched or being placed, respectively.

A high summary of the interaction can be seen with this diagram:




# 05 Programming

The core of the code for our project came from a previous project, Pillow Talk. What we learned from Pillow Talk was integral to the success of HoHoHoBot — crucially, we were able to adapt the function for saving audio from the microphone stream, which was a key challenge in Pillow Talk. We also largely adapted part of our dialogue system from Pillow Talk, since the recording interaction was roughly similar — that is, a user would be prompted to record their thoughts into the microphone (for Pillow Talk, it was as a message to a partner; for HoHoHoBot, it is as a saved message within the ornament).

However, the implementation for this project still required considerable effort — where Pillow Talk was mostly self-guided (only asking for a voice recording), HoHoHoBot required a great deal more physical input. For instance, one of the most important factors in the HoHoHoBot interaction was whether or not a capacitive wire was occupied by an ornament-recording or not; in order to resolve this, we implemented a dictionary that would be able to save capacitive touch index as the key, with the appropriate audio file as the value. Centering this dictionary data

structure resolved several issues — it allowed us to check if a capacitive touch index was occupied or not, it allowed us to assign the proper audio file (as mentioned), and it allowed us to save this data structure as a file object using the pickle library, enabling persistent mappings that would be able to be reloaded between sessions (gotta make sure to save those audio recordings!).

Future considerations to our software include potentially adding metadata to the message and wrapping it into a bespoke structure. This data could include the location of recorded message, number of plays after the fact and their locations.