

Trashie

Lindsey Chung
Alisha Sonawalla
Srishti Sanya

User Manual

Trashie is designed to be simple to use in order to lower the knowledge barrier for people who want to start sorting their waste properly.

1. Users should first ensure that they have iOS 12 or higher installed on their device, which ensure that they have support for ARKit 2.
2. When the app launches, the user should position their phone in front of the object and wait for the app to tell you what kind of waste the object is: compost, landfill waste, or recyclable waste.
3. If the user is looking for recycling or landfill centers, use the dropdown menu on the top left of the screen to navigate to the map screen by clicking "Map".
4. Once the map screen pops up, the user will be asked to allow the app to use their current location to display facilities in the immediate area.
5. The user can then use the segment controller at the bottom to navigate between landfill centers and recycling facilities nearby.
6. To go back to scanning, simply bring down the menu again and click the "Scan Items" option.

Technical Description

Trashie works by cross referencing the image it sees when the camera is pointed at an object with the ARKit Object scans stored in its files. Once the camera detects that there is an object of interest, it scans the object and attempts to match "features" (often based on strong edges, texture variations, changes in color, or text items) between the object being scanned and the ARKit object scans that were previously created to be the models for object recognition. If the application detects a match between the object being scanned and an ARKit object that it recognizes, then the appropriate SpriteKit scene is shown based on a series of hard-coded values that classify each object. The scene displays the relevant waste info for the object to the user.

The user can also use the menu button in the top right corner to access the map screen. This button is linked to a segue that transitions between the map view and the AR scanner view. Once the user is on the map view, they must allow Trashie to access their location while in the app in order to get locally relevant results, and then they can use the segment controller at the bottom of the screen to toggle between a search for landfill locations and a search for recycling centers. If the user would like to return to the item scanner, they may do so using the options menu.

Technologies Used

Trashie makes use of several types of iOS technologies, including ARKit, MapKit, CoreLocation, and SpriteKit. It was constructed in Swift because of the necessity to keep it on a mobile platform and to integrate various iPhone-specific technologies. ARKit was used to create AR reference objects and for the detection of 3D objects. Once a recognizable object is detected, a SpriteKit label is shown that classifies the waste. SpriteKit scenes were used to display text above trash items that shows the user whether the object is classified as a “recycle”, “compost” or “landfill” object. Once ARKit recognizes the objects, an empty plane is created above the object, and the SpriteKit scene is rendered on top of the plane based on the item ARKit detects.

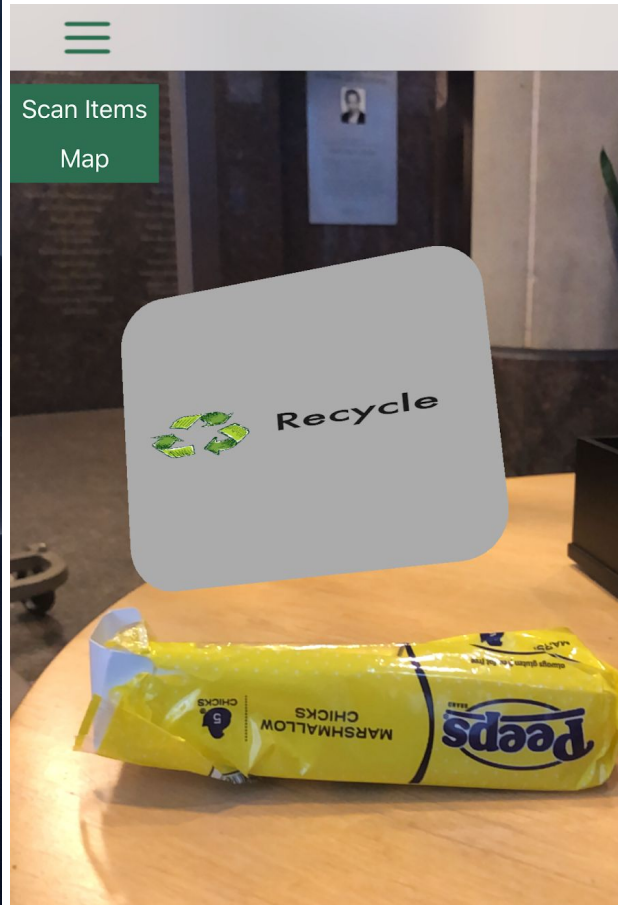
On the mapping side, MapKit was used to create and render the map area, as well as to create and place the point of interest pins that are used to show search results. CoreLocation was used to retrieve the user’s current location and generate a bound within which the search function could generate nearby recycling and landfill drop-off centers.

Code Highlights

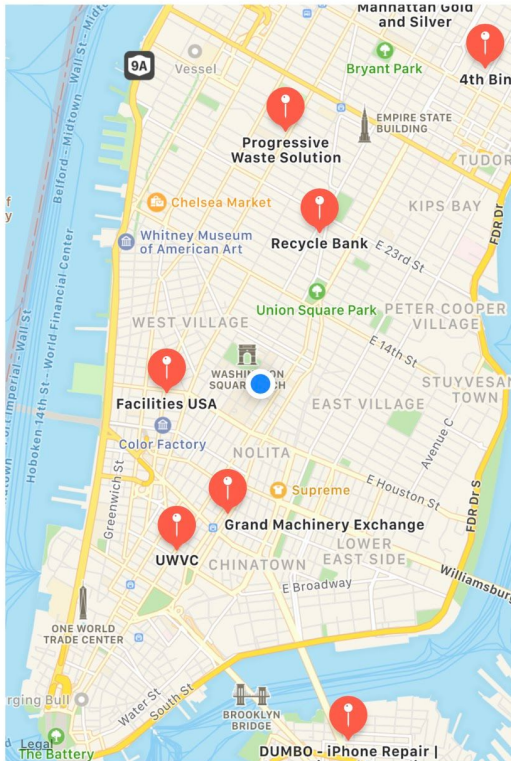
One code highlight is the ARKit renderer that renders the SpriteKit Scene onto the image in the camera, anchors it to the actual object being detected, and positions it appropriately to make sure that the plane is readable when it is rendered. This is located in the `renderer` function in `Trashie.xcodeproj`.

A second code highlight is the search function for the Map that takes the name of the segment currently highlighted in the segment controller on the Map view, and executes a search based on various search criteria. It then adds pins on the map to every location that matches the criteria and the keyword. This is located in the `searchInMap` function in the `MapKitFinal.xcodeproj` file.

Screenshots



[< Back](#)



Recycling Center

Landfill

Team Contributions

- Alisha Sonawalla: Created the AR View and the SpriteKit Scenes for each waste category. Alisha also created the functionality for identifying which object the camera was pointed at, as well as the logic for assigning each object a waste category.
- Srishti Sanya: Created the Map View as well as the search functions on the map and the functionality for detecting a user's location. Srishti also created the controller for the map options, as well as the ARKit Object scans that were used for reference in detecting objects.
- Lindsey Chung: Created the segues between AR views and Map views, and integrated the two halves of the application into one complete app. Lindsey also created the menu that takes the user between the views in the application.

Future Features

We would like to integrate a library of images, such as the IBM Watson image recognition libraries, in order to be able to identify an even wider collection of objects with many variations. This would also lend itself to the incorporation of machine learning functions to ensure an efficient process for object identification. However, we would like to use libraries of ARKit Objects when possible, as that will allow our application to seamlessly integrate these new objects into our identification process. Additionally, the integration of such libraries would allow us to bypass the manual scan process to add items to the collection of options that the application can scan.

In the future, we would also like to implement the ability to customize the look of the plane that appears with the waste type information by allowing for various textures and colors for the plane. There are also cosmetic changes that can be implemented, such as animating the images for landfill, sprite, and compost, and adding sounds with each detection. We would also like to incorporate some form of incentivization, in the form of keeping track of recyclable objects scanned, and potentially comparing and sharing your scan totals with your friends to create a competitive aspect.