**Memo**

To: Professor Pisano

From: Daniellia Sumigar, Juan C. Vecino, Tristen Liu, Ramy Attie

Team: 14

Date: 3/6/2024

Subject: Scan It! Pack It! Arrange It! - Second Prototype Testing

1. **Setup Summary**

The user manual input mode and the SceneKit schematic generation was performed on an Xcode project with an iPhone 15 Pro emulator installed on a MacOS device.

The manual input is done with embedding python into the Swift code used for creating the iOS application. The Manual Input menu appears once the Scan It! Pack It! application is launched. The menu contains empty fields for specifying the container and object width, height, and depth dimensions separately. The dimension values will be stored in an internal variable. The menu also includes an add object button that will add more input fields for a new object and a finish button to complete the manual input process and demonstrates successful python embedding by printing the system path in the console.

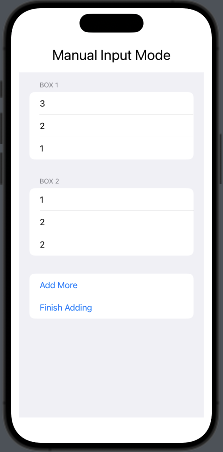
Additionally, the finish button in the manual input mode will enter a new screen that displays the SceneKit packing schematic for testing. The packing schematic is a rotatable figure that depicts the container as a clear, wireframe box with each object color-coded placed inside using an animation to indicate the order in which they should be packed. The schematic test example illustrates the scenario where all objects fit into the container, replicating an experiment done in the previous prototype testing directly with the python packing code.

The packing script test was performed using a python script taking in manual input dimensions of the container and the objects for packing.

**2.0 Experimental Results**

**2.1 User Manual Input Mode**

As the user enters dimensions of the container and the objects, the dimensions are printed in the console. This demonstrates that the embedding of python to construct the manual input mode is working and is compatible with the swiftUI code. By pressing on the ‘Add more’ button, the user is able to specify as many box dimensions as needed for packing. The ‘Finish Adding’ button also completes the manual input mode, triggering the start of the schematic generation based on the supplied dimension data.



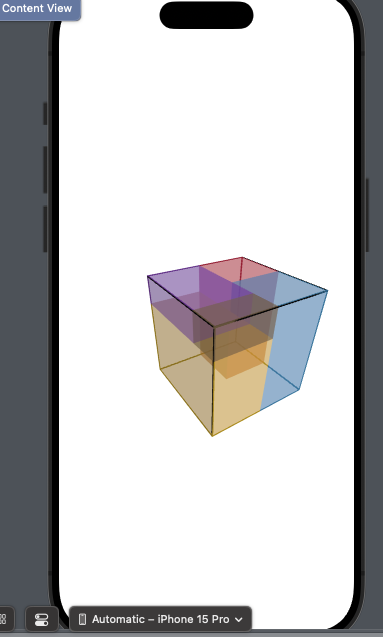
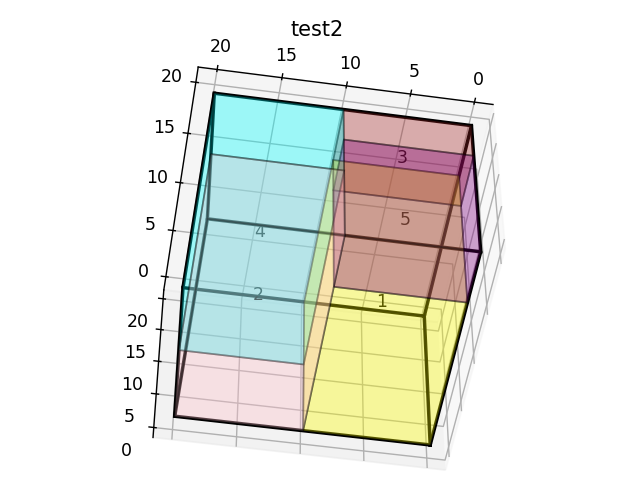
**Figure 1.** Manual Input Mode Menu

**2.2 Packing Script**

After running the packing script, the algorithm returned the arrangement of the objects that could fit within the specified container along with a detailed list of all unfitted objects with their dimensions within 0.2 seconds.

**2.3 SceneKit Schematic Generation**

The packing schematic successfully displays the arrangement of all objects that can fit within the specified container. It is a rotable schematic with an animation of each object to be placed within the container. From the previous prototype testing, the schematic was generated in python. The SceneKit schematic generation accurately replicates the configuration from the python code as shown in **Fig 2**.



**Figure 2**. Comparison of SceneKit to Python Packing Schematic

**3.0 Conclusion**

During this prototype testing, we have demonstrated major pieces of the overall project working individually. The task to create a scanning to packing pipeline remains a work in progress as we aim to replace the manual input mode with a scanning method to supply dimension data to the packing algorithm. We initially wanted to leverage the embedding of python from the manual input mode to run the packing algorithm directly inside the project code. However, we ran into issues installing some of the required python libraries to run the packing algorithm. Since then, we have pivoted our approach toward creating a Flask API and hosting on AWS to run the python backend. The python packing script featured in this testing serves as a framework for querying the relevant packing data needed to generate the schematic.

Based on the results, the schematic generation works sufficiently, illustrating how to place the fitted objects into the container step-by-step. However, it currently does not display any information about the unfitted objects in the schematic. To improve the packing schematic, we aim to add labels and implement buttons to toggle between each individual object, while also addressing the unfitted objects to ensure that the packing process is as clear and intuitive as possible to the user.

We have yet to implement a scanning method that will use LiDAR to automatically detect dimensions of objects - we are currently looking into using ARKit/RealityKit to implement the scanning module.