**Memo**

To: Professor Pisano

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

Team: 14

Date: 02/27/2024

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

1. **Required Materials**

Hardware:

* iPhone
* MacOS Development Device

Software:

* Python Scripts
  + Packing function with Dimension Input
* XCode
  + User Manual Input Mode
  + SceneKit Schematic Generation

1. **Setup**

The testing setup will be split into 3 separate modules: User Input, Packing Script and SceneKit Schematic Generation

The User input test will be a simple test to demonstrate that the input from the User is correctly recognized and stored in an internal variable. It will involve pushing the application to an iPhone device, and reading outputs from the console whenever the buttons on the application are pressed. Three decimal fields are available to input the Width, Height and Depth of an object, and two buttons are available to add new object dimensions and finish inputting dimensions. The add object button will add three new input fields to the page, while the finish input button currently demonstrates the correct embedding of Python by printing the system path to console.

The packing script consists of two functions and a main script. The function pack(width, height, depth) takes in three lists of floats, where the corresponding indexes of the lists represent the dimensions of one object. The function check\_input(values) makes sure that the input values from the main script are valid dimension values that can be processed by the packing algorithm. Finally, the main script prints and accepts inputs from the console, allowing users to input object dimensions and calls the pack() function after all the object dimensions have been inputted. This script is here in place of the not-yet-implemented pipeline for the Swift XCode project to the Python function.

The SceneKit Schematic generation will demonstrate the optimal arrangement of objects within the specified container. This test setup involves the scenario where the container perfectly fits all input objects. It will show the order of how to place objects within the container with a rotatable view. The container is illustrated as a transparent wireframe and each object is portrayed with a distinct color. The SceneKit Schematic generation will be built on the same Xcode project as the User Input test.

1. **Pre-testing Setup Procedure**

User input:

1. Build the XCode project and push it to the testing iPhone device
2. Ensure functionality of all fields and buttons
3. Maintain connection with the MacOS in order to read console outputs

Packing:

1. Clone or update the team github repository
2. Change directory into the directory with packing\_script.py
3. Run the script in terminal or VSC

SceneKit:

1. Build the XCode project (same as User Input) and push it to the testing iPhone device
2. Ensure functionality of buttons
3. Maintain connection with the MacOS to view schematic
4. **Testing Procedure**

User input & Schematic Generation:

1. Input three dimensions into the fields on screen
2. Click “Add More” to add another dimension list on screen
3. Input three more dimensions, and change the previous dimensions
4. Click “Add More” and view console to see that the dimension list is updated
5. Click “Finish Adding” to see the packing schematic and view console to see that Python is correctly embedded.

Packing:

1. Test the dimension checker by inputting a non-numerical value, zero value and non-positive value
2. Input dimensions from any previous tests to see packing output
3. **Measurable Criteria**

The criteria for successful running and output is as follows:

1. The user should be able to manually input dimensions, and the dimensions should be automatically updated in case any previous fields were modified
2. Python should be successfully embedded in the Xcode project, and should be demonstrated through console output and Python system calls
3. The Packing script should successfully filter non-valid dimensions, and the dimension processing should be automatically completed after the user inputs all dimensions
4. The schematic generation should successfully display all input objects and the container. The placement of objects should be clear and user-friendly and the schematic must be rotatable for clarity.

1. **Score Sheet**

| **User Input** | | |
| --- | --- | --- |
| Test # | Test Component | Correct? (Y/N) |
| 1 | Dimensions successfully added |  |
| 2 | Old Dimensions automatically update |  |
| 3 | Python embedding is demonstrated |  |
| **Result** | | % |

| **Packing Script** | | | |
| --- | --- | --- | --- |
| Test # | Object Size (cm) | Container Size (cm) | Correct? (Y/N) |
| 1 | Obj 1 - 15 x 39 x 22  Obj 2 - 25 x 5 x 9  Obj 3 - 53.3 x 40.6 x 10  Obj 4 - 35 x 8 x 12  Obj 5 - 30 x 40 x 10  Obj 6 - 39 x 32 x 10 | 53.3 x 40.6 x 40.4 |  |
| 2 | Obj 1 - 10 x 15 x 20  Obj 2 - 10 x 20 x 10  Obj 3 - 10 x 5 x 5  Obj 4 - 10 x 20 x 10  Obj 5 - 10 x 5 x 15 | 20 x 20 x 20 |  |
| 3 | Invalid dimension values are successfully caught | |  |
| **Result** | | | % |