CamIO Explorer instructions

(Updated 1/2/19)

This version of CamIO uses a stylus and an object without any board (attached barcodes). It assumes the object lies flat on the table top and can thus move with three degrees of freedom (x and y translation and rotation on the table top plane). However, every time you launch explorer.py you have to determine the pose (see below); the pose has to be re-estimated whenever the camera or object is moved.

# Hardware:

* Camera: Logitech C920 (<https://www.amazon.it/Logitech-HD-Pro-Webcam-C920/dp/B006H967FA>
* Object: Southwest airplane (<https://www.amazon.com/gp/product/B000TQXNKE>)

# Software dependencies:

* Python 3.4+
* Numpy
* OpenCV
* gTTS: generates .mp3 TTS utterances
* pygame: plays .mp3 and .wav files

# How to set up and use CamIO:

1. Ground plane marker

This is used to tell the camera where the table top is. Print out the file ground-marker-A4.pdf and feel free to trim it around the narrow square frame (8 cm wide).

1. Stylus
2. This is used to point to locations of interest on the object. Print out the file stylus-A4.pdf and cut out the pattern. It’s designed to wrap around a square dowel (<https://www.amazon.com/gp/product/B01G0TC2FU>) which is ½ inch wide and 12 inch long. Ideally the dowel should be cut to roughly 6-10 inches long (the pattern isn’t long enough to fit the original 12 inch length). The longer the dowel the more accurate it will be in general, but the less convenient to use.
3. The top of the stylus (where a pencil eraser would be) is defined by the single marker above the rows each containing four markers. This top marker can be taped over the flat end of the dowel.
4. The pattern should be trimmed on the bottom if needed to fit the dowel – don’t trim the top (which must be included in the stylus).
5. It’s nice to have a pointed tip at the bottom of the stylus, but this is optional.
6. If you don’t have a dowel, you can just cut out the printed pattern, fold and tape it into a dowel shape. Then you have to hope it holds its shape! It may be helpful to add some sort of hard tip to the bottom.
7. You must measure the length of the stylus, from top to tip, and specify this in the variable stylus\_length (in parameters.py) in units of cm.
8. The stylus won’t work if it’s not visible to the camera. When it’s visible you’ll hear a double pulse (see below). If the stylus is pointing directly towards the camera it may be hard to see; it may help to angle the stylus a bit away from direct pointing.
9. Camera and object

Mount the camera on the tripod about 14 inches high, pointing down at the table where the airplane is placed. It’s helpful to secure the airplane on the tabletop (e.g., with glue dots on the wheels) so it remains fixed unless pushed hard.

1. Over view of CamIO user interface:
2. Ambient pulse sound: single pulse means ground plane visible, double pulse means stylus visible, silence means neither visible
3. Approach hotspot --> begin announcement right away
4. Refusable interface: TTS announcement is terminated immediately if stylus disappears, moves away or goes to another hotspot
5. Keypresses (see below for details): 0 sets ground plane, else 1 and 2 set two views of the stylus on the tabletop; a and b set the two anchor points; 3 finishes by determining pose; 4 prints the current stylus tip XYZ location in 3D (in annotation coordinates, which are only available once the pose is determined). Note that keypresses are ignored unless the OpenCV camera window is active. Each keypress is followed by a TTS confirmation.
6. Determine the object pose
   1. First you must determine the ground plane, either using the ground plane marker (you can remove it after you’re done) or two views of the stylus on the tabletop.
   2. Next you must touch the stylus tip to the first anchor location, which in this case is the airplane nose. Then press the a key.
   3. Next you must touch the stylus tip to the second anchor location, which in this case is the airplane tail (towards the end of the fuselage). Then press the b key.
   4. Finally, press the 3 key to determine the pose.

# Details:

1. Camera focus: I set the camera focus in line 40 of explorer.py to a value that works for me using the C920 camera. You might need to change this. For some reason I couldn’t get the camera autofocus to work for this camera.
2. The first and second anchor locations are defined as the first two hotspots in the object definition.