User Manual

Table of Contents

Required equipment:

- Walking compensator with two mice optic detector at 90° of each other.
- Cylindrical or semi-spherical screen
- Video projector
- Computer

Installation

Unzip the software where you want it to be installed and that's it you can use it.
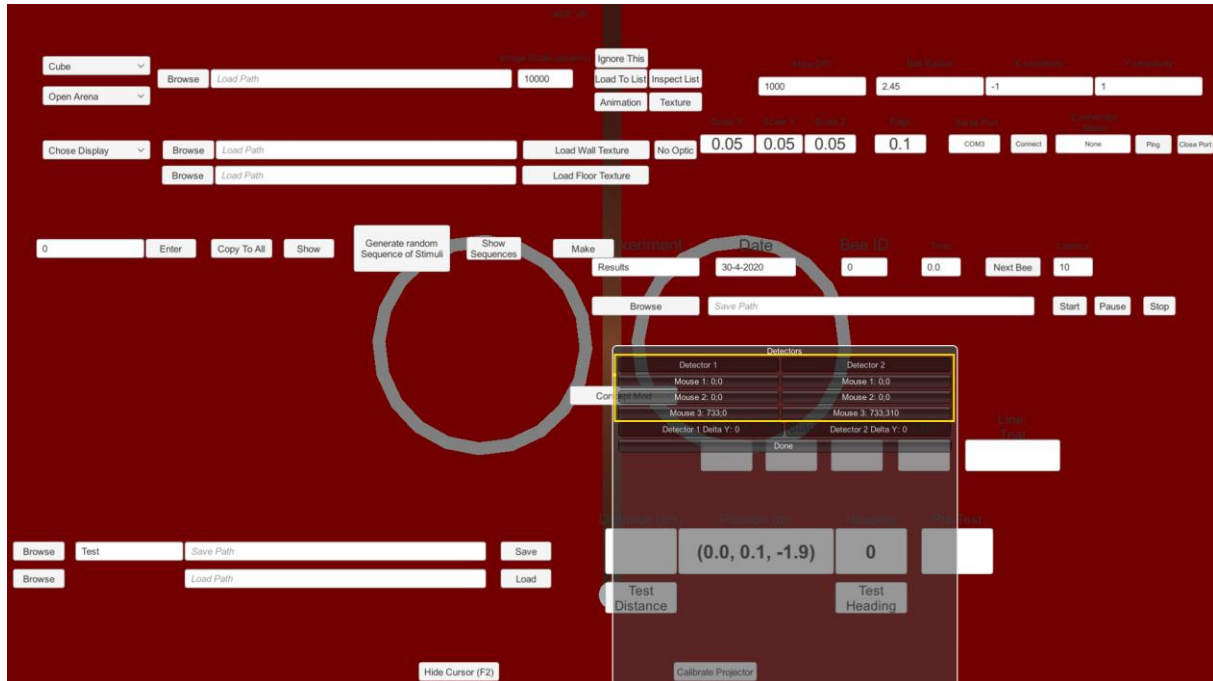
Simple associative conditioning:

Have the walking compensator and the video projector plugged in before starting the software.

Setting up the experiment

- Select Arena Shape
  - **Open Arena** is the recommended setting, a cylindrical arena 20cm in diameter

- Select which mice are the detectors

Move your trackball and note what values are changing, this should tell you which mice are part of the trackball. Click once on each mouse you want as detectors and then click **Done**.

NB: If you make a mistake in this step can click **Re-select Mice** to re-open the selection window.

- Chose two *.png* file for your stimuli
  o Click **Browse** > chose a file > click **Select** > click **Load To List** > repeat
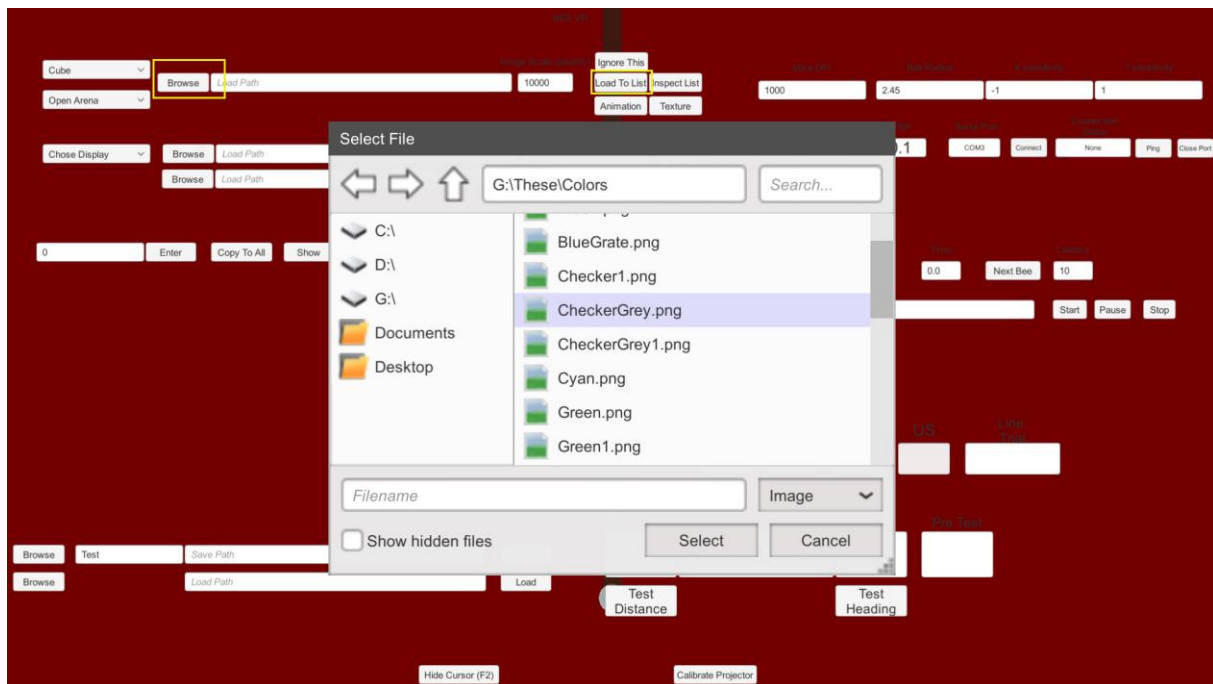
For the texture a 400x400 pixels image works well with the default settings.

If you only want one stimulus you can load one normal texture and one transparent one, and then click **Ignore This** to keep the transparent object from triggering a choice.
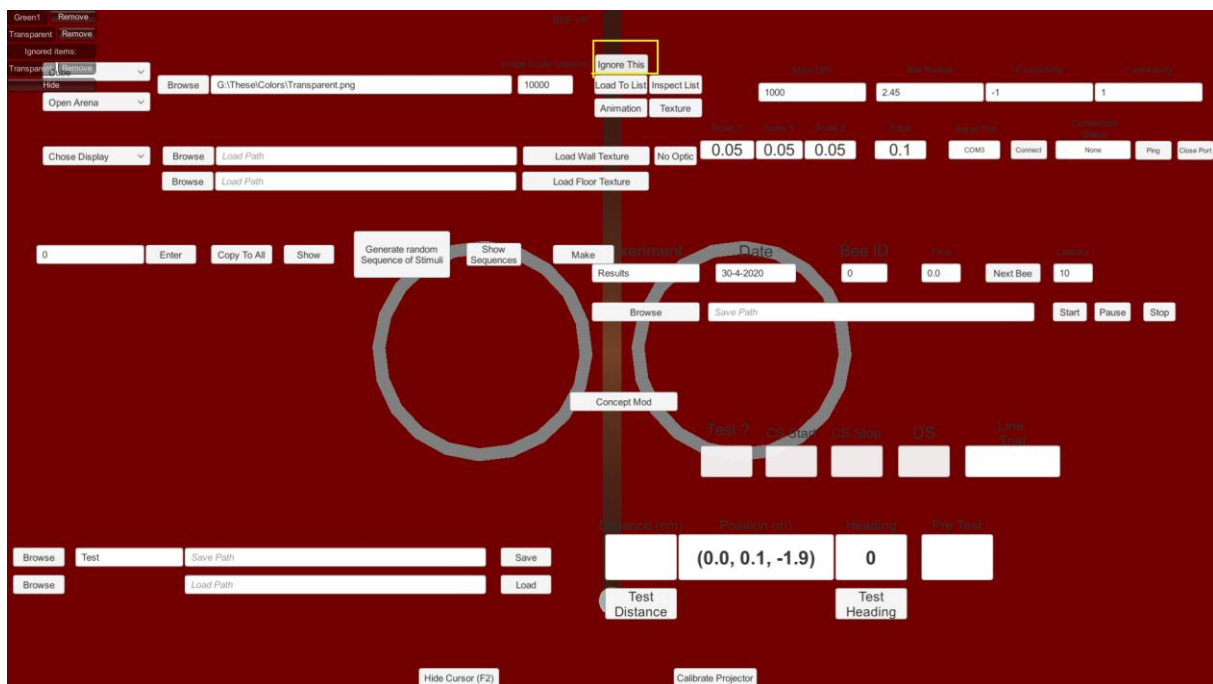
- Chose the texture for the floor and the walls or the arena
  - o Click **Browse** > chose a file > click **Select** > click **Load Wall Texture**
  - o Click **Browse** > chose a file > click **Select** > click **Load Floor Texture**
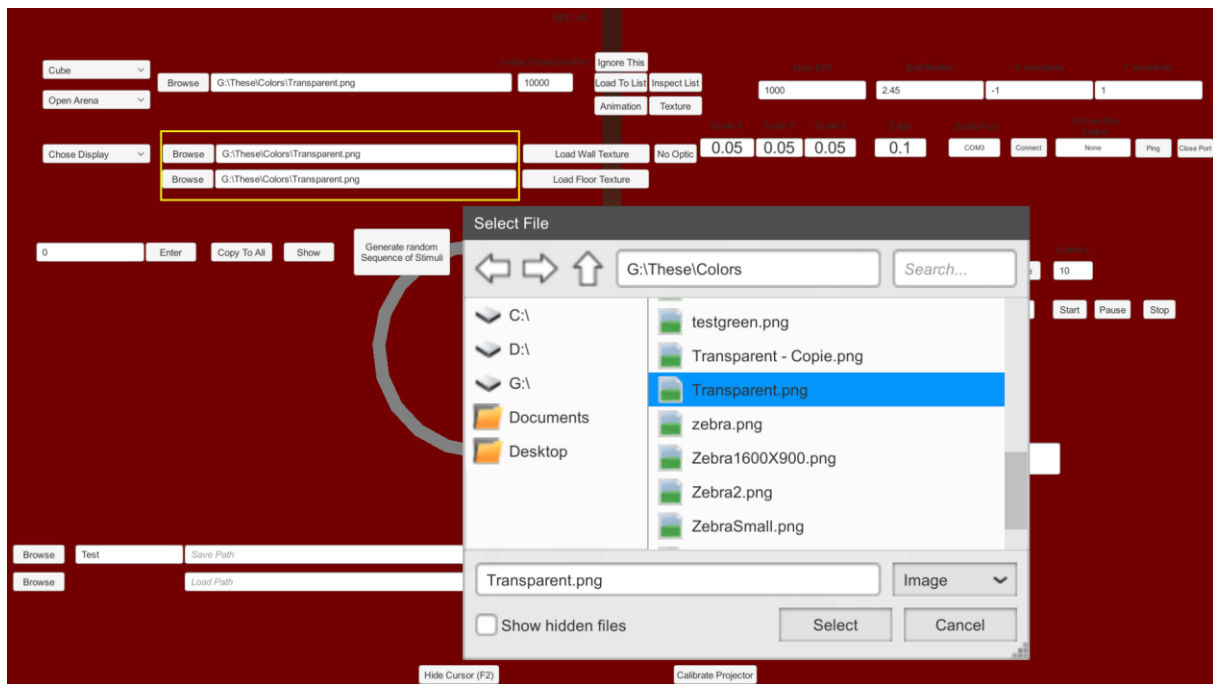
Using a transparent texture works well to set a simple environment

- You can then click **Inspect List** to check if everything was loaded properly and how the environment looks

NB: If you wish to keep the wall and floor transparent you can skip this step. By default, wall and floor are not displayed.
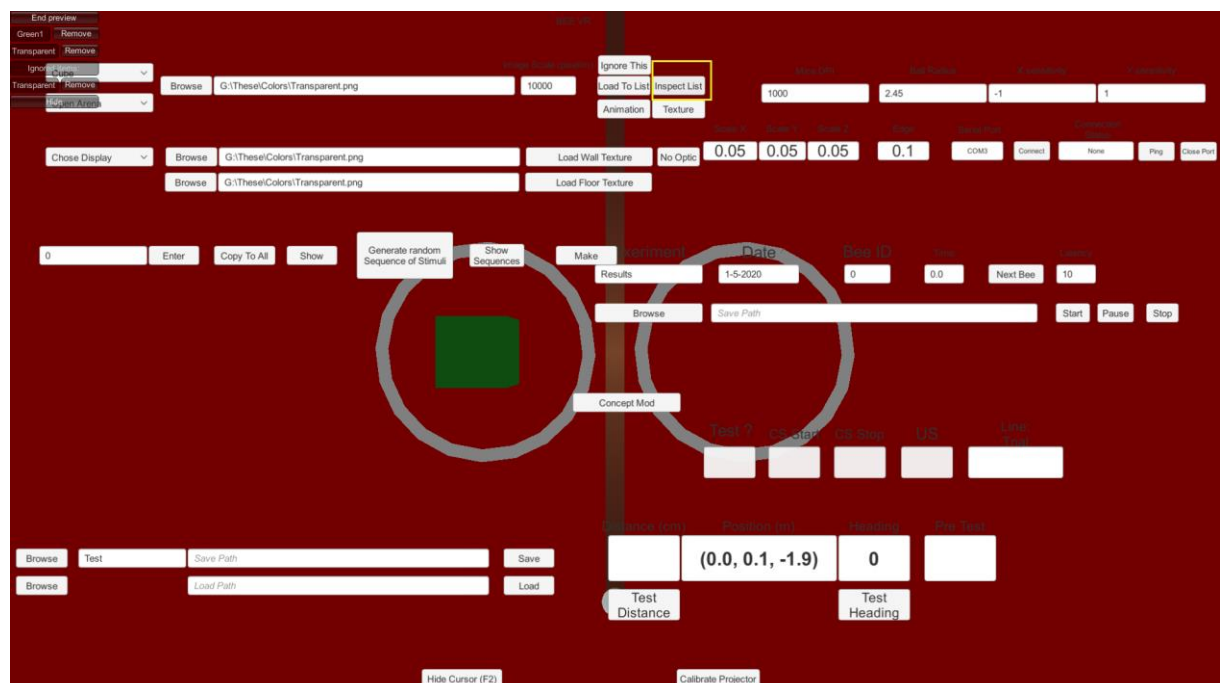
NB: pressing **F11** twice switch the monitors so you can have the VR scene on your main monitor
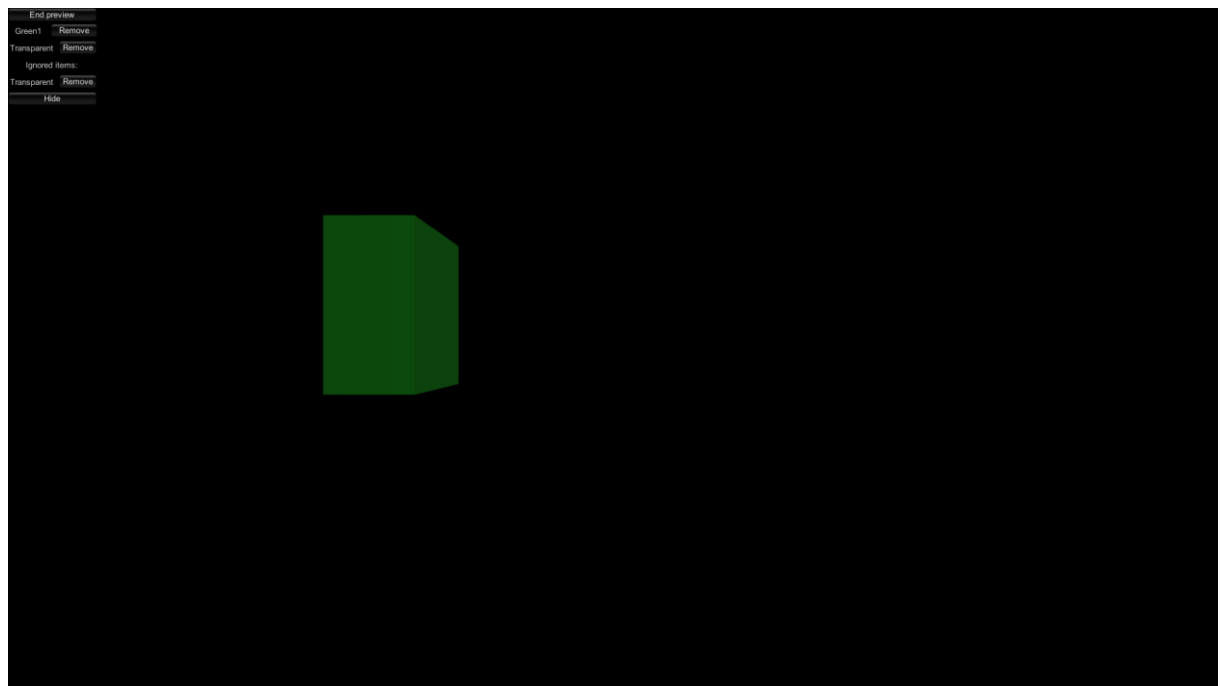
Pressing **F11** again will bring you back to the normal display. Click **Inspect List** one more time to stop the preview.

- Set your experiment's protocol:
  - Enter how many different type of task there will be, for this example we'll chose 1 learning phase and 1 test.

o Fill in the parameters

US Duration: How long, in seconds, will the simulation freeze for the US delivery

Prep Duration: Display either **Stim 1** or **Stim 2**, randomly chosen, on screen for the specified duration. If set to 0 this phase is skipped.

CS Start: How long before the stimuli appear on screen and the bee can start moving in the VR. This is the inter trial interval.

CS Stop: How long before the trial end. This the maximum duration of the trial.

Repetition: How many time should the line occur. For exemple 10 on our first line here means 10 trials.

NB: You can click **Copy To All** to copy **US Duration**, **CS Start**, and **CS Stop** on every line

Stim 1 and **Stim 2**: The name of the stimulations to use for this line. If our texture files are named *Green.png* and *Blue.png*, the entries will be *Green* and *Blue*.

Test ?: Is this line a test? If **No** the simulation will freeze for **US Duration** seconds if the bee makes a choice. If **Yes** the simulation doesn't freeze when the bee makes a choice.

PreTest ?: If **Yes** the software will compute which stimulus the bee spent the most time choosing and display its name in the lower right corner at the end of the trial.

Wall on ?: If checked wall and floor are displayed, defaults to off.

**Concept ?:** /!\ Not fully functional /!\ This option is not fully functional yet and should be left on *none*

Is 2D ? If selected this option will switch the VR to a 2D mode where the bee can only turn on the spot making the stimuli turn around her.

Once you've set your protocol you can click **Hide** to hide the parameter window.

To finalize the protocol, you need to click **Generate random Sequence of Stimuli**, you can also click **Show Sequences** to see the sequences generated.
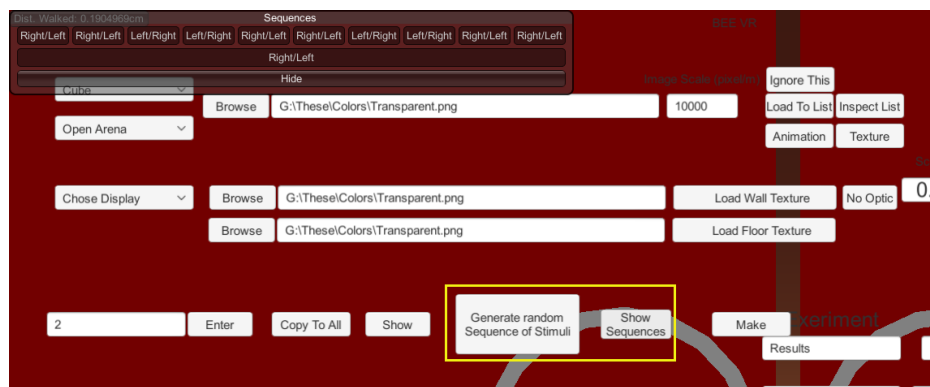
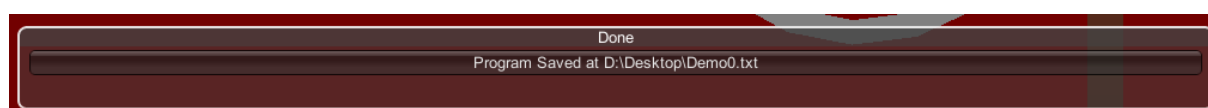NB: This generates pseudo random sequences, following the rule that stimuli must not appear on the same side more than twice in a row.
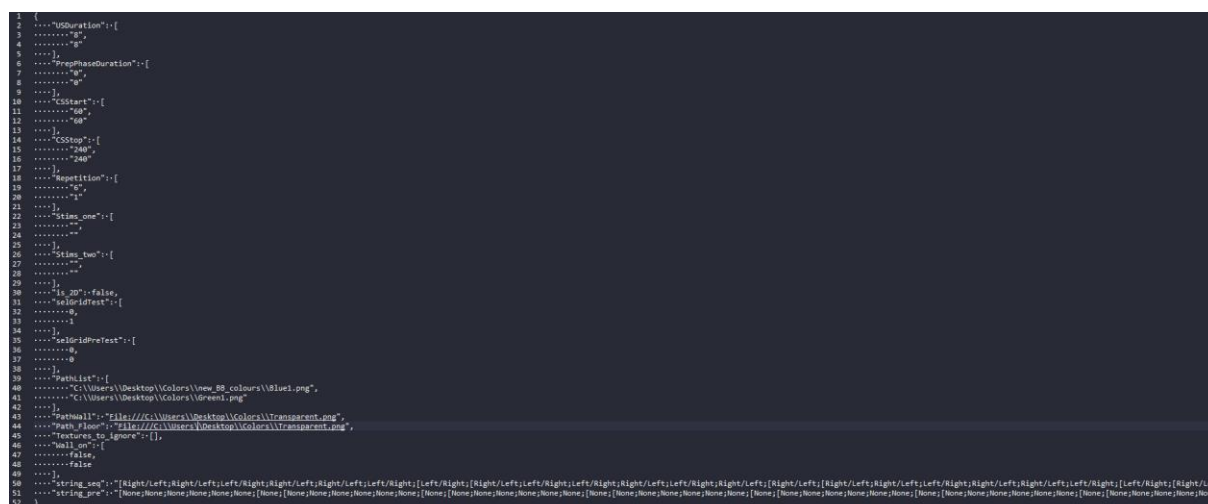
- Save your protocol:
  Enter the name of this protocol in the field next to **Browse**, click **Browse** to choose a folder and then **Save** to save your protocol.

If it works you should get this message

Protocols are saved as *.txt* files, meaning that you can edit them manually.

- Chose where to save the data from this experiment
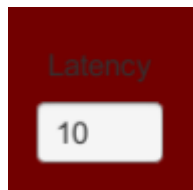
Chose the name of the file, and click **Browse** to choose where to save it.

Upon starting the experiment, the software will create a file with a name following this format: **<Name_of_this_Exp>_<Date>_<Bee_ID>**. Each time a full protocol is completed the Bee_ID will increase by one and a new file is created.

- Set the **Latency**
  By default it's set to 10s, meaning that the experiment will start 10s after you press start.

And you're done, you can now click start and begin your experiment.

During the experiment

Because all the mice control the pointer at the same time you might want to hide it by pressing **F2** so that you don't see it moving around on the screen as the bee walk. You can control the experiment via the keyboard to some extent.

Space will start and pause the experiment.

CTRL+Space will stop the experiment.

SHIFT+Space will increment the Bee-Id by one.

Data output

Data are saved as *.csv* files with semi-colon separators [;]

Excel or R should have no problem opening them, there might be some issue to open them on a Mac since it might expect colons and not semi-colons.

- **Line**: The number of the instruction line in your protocol. In our example above line 0 is the learning trials and line 1 is the test
- **Trial**: The number of the repetition within the line. In our example line 0 has 10 trials and line 1 has 1.
- **Time(s)**: Time in seconds since the beginning of the experiment.
- **PositionX/Z**: coordinate of the bee on the horizontal plane.
- **Rotation**: Heading of the bee, 0 is forward.

- **DistanceTotale**: Total distance walked since the beginning of the trial.
- **Speed**: Mean speed between two data point, the point being very close to each other this should be close to the instantaneous speed of the bee, in m/s.
- **Choice**: Name of the chosen stimulus, if the bee is making a choice.
- **Test**: Is this a test, 0 for no and 1 for yes.
- **Centered**: What object is the bee centering on the screen.
- **Looking_at**: What object is close to the center of the screen.
- **Edge**: Is the bee centering the edge of and object and if yes which one.
- **Edge_coord**: Coordinate of the centered edge

Extra functionalities

Settings and calibrations

- Screen shape
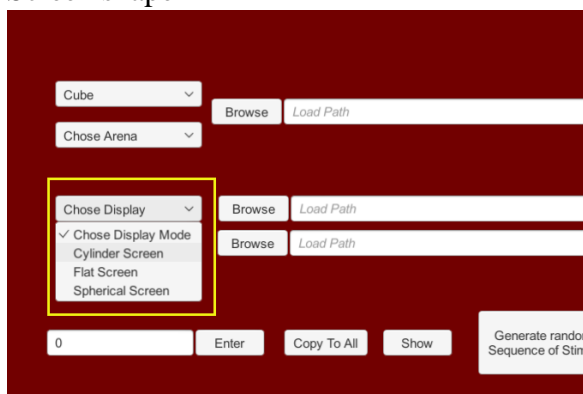
You can set the shape of the screen you're projecting on, by default it's set to Cylinder, but you can also set it for a spherical screen or a flat one.
- Shape of the Stimuli
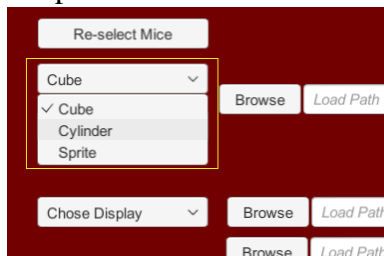
You can chose between **Cylinder**, **Cube** or **Sprite**, by default it's set to **Cube**. **Sprite** will display the stimuli on two flat surfaces that always face the camera.
- Size of the stimuli

You can change the size of the stimuli; the values are in meter.

- Recording frequency:

Time allows you to set the minimum duration in seconds between two data recording, leaving it to 0.0 means that the software records a data point every processor cycle (~every 0.0167 second).

- **Test Distance**

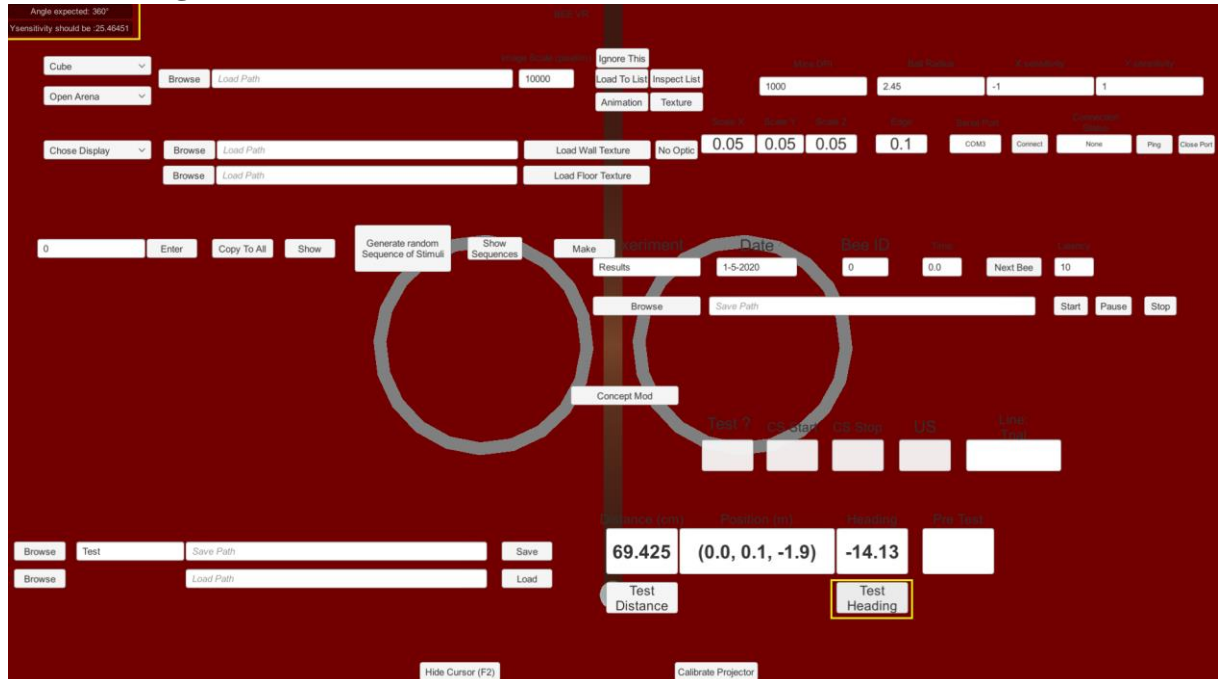After choosing an Arena and setting your detector you can click **Test Distance** to see what mouse sensitivity you should use. Click on **Test Distance** and then have your walking compensator make a complete rotation along the X axis (forward/backward around the horizontal axis). If everything is set correctly the value in **Distance** should be the same as the one displayed in **Distance expected** in the upper left corner, if it's not the software will suggest a new **X sensitivity** to correct the difference however before changing the sensitivity you should make sure that the **Ball Radius** is correct as it is the most likely source of error.

- **Test Heading**:

After choosing an Arena and setting your detector you can click **Test Heading** to see what mouse sensitivity you should use. Click on **Test heading** and then have your walking compensator make a complete rotation along the Y axis (left/right rotation around the vertical axis). If everything is set correctly the value in **Heading** should be the same as the one displayed in **Angle expected** in the upper left corner, if it's not the software will suggest a new **Y sensitivity** to correct the difference however before changing the sensitivity you should make sure that the **Ball Radius** is correct as it is the most likely source of error.

- **Mice calibration:**

You can set the **DPI** of the mice you're using in the walking compensator. You can also set the **Radius** of the ball you're using, by default the software will expect a 2.45cm radius. The default **Sensitivity** should be fine on any setup but you might want to change it if your video projector is behind the bee (inverting the sign) or if you want to create discrepancies between the bee's movements in the real world and the movements in the virtual one.

- **Edge scale:**
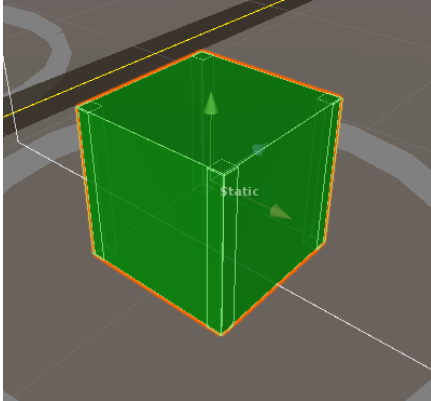
Edge is the scale of the edges of the cube relative to the cube size. Edges are used to detect if the bee is centering the edges of the cube or not.
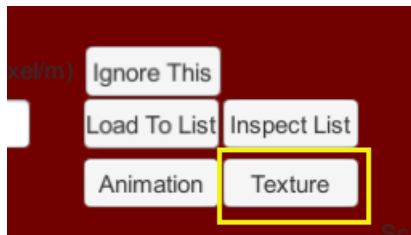
- **Calibrate Projector**:
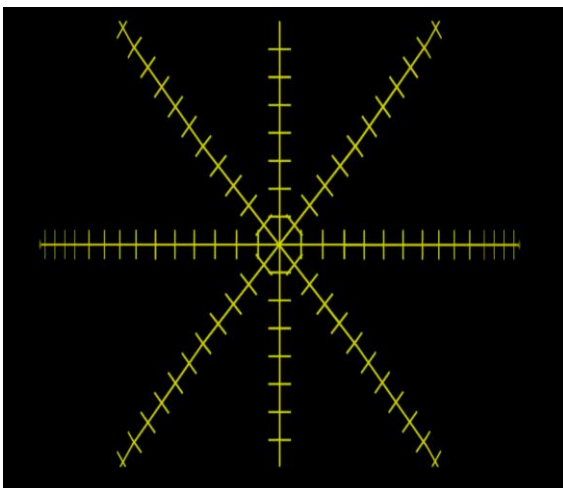
Displaying this target on your screen might help you position your video projector properly in relation to the screen. The target is deformed according to the selected screen shape.

Texture animation

- **Texture**:

  o Target can be **Wall** or **Floor** for the wall and the floor texture respectively.
  o Tiling allows to set how many time the texture is repeated on each axis.
  o Offset allows to set the offset of the texture on each axis
  o Click **Update** to apply your changes

- **Animation**:

  o **Offset increment** allows to set the movement of the texture during each frame in pixels along each axis
  o **Fps** sets the number of time the texture is changed per seconds, this is capped by the number of processor cycle per seconds.
  o **Update** to apply your change.
  o **Play** to start the animation.
  o **Stop** to stop it.

Arduino connection
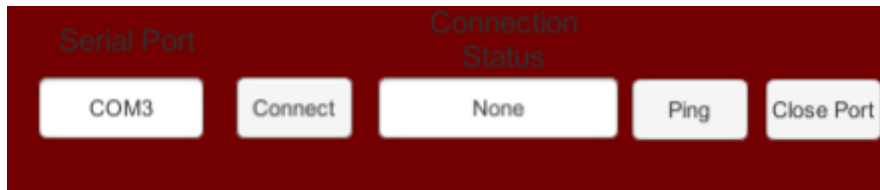
- **Serial Port**: Name of the serial port to which the Arduino is connected, pressing Enter after typing the name will establish the connection
- **Connect**: This button is not working as intended in current version. See **Serial Port** entry above on how to establish a connection.
- **Connection Status**: Port_Name_Open if the port is connected and open, Port_Name_Close if the port is connected but close, None if no attempt as connection has been made, No port if no port is detected after connecting attempt.
- **Ping**: Will send "1" as a signal through the port

If an Arduino is connected to the setup the software will send "1" through the port each time an experiment starts and "0" each time it ends. Those features might be expended in the future.