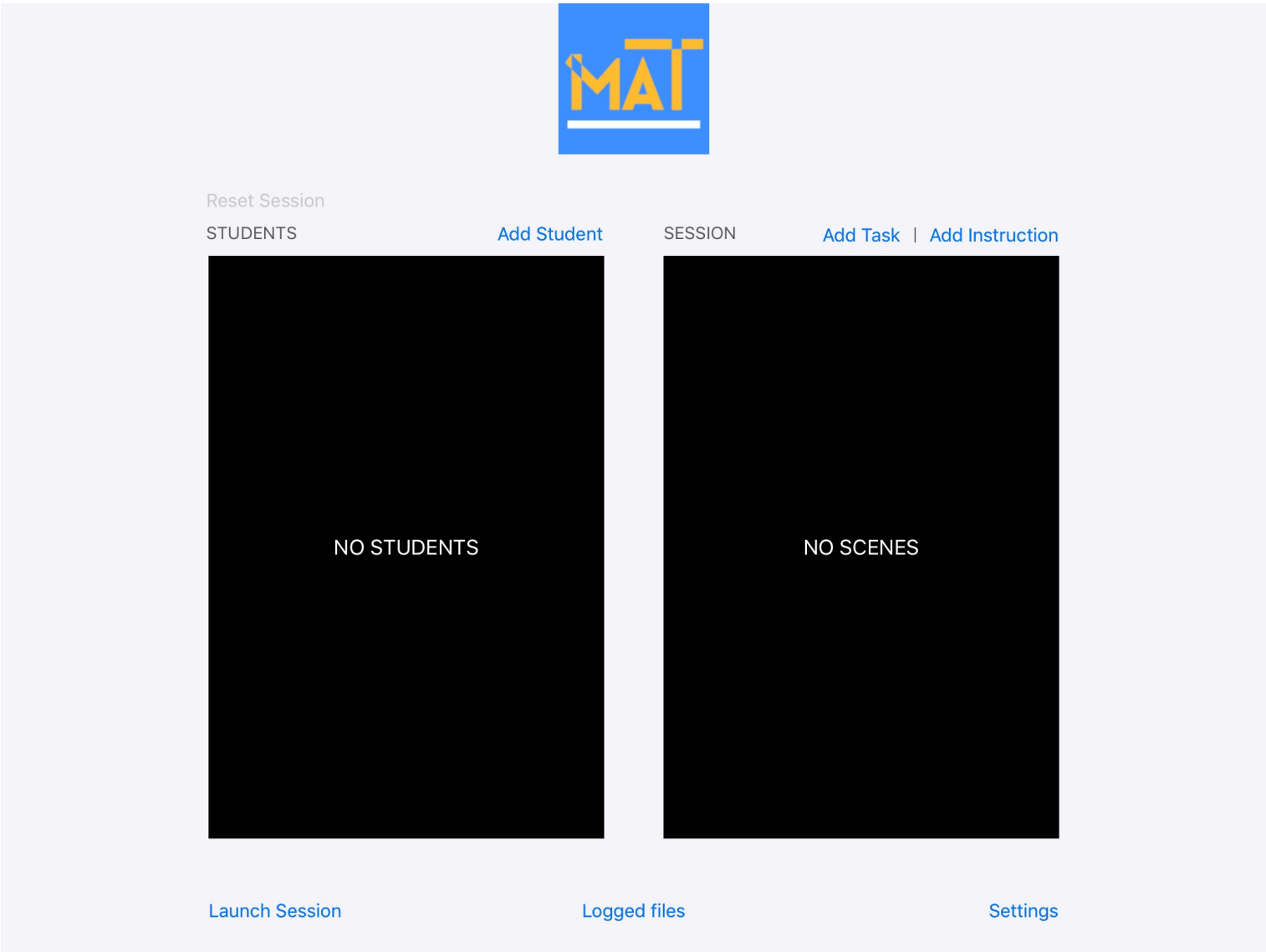
The app starts with two blank pages.

On the left you can add new student's name and information and on the right the set of tasks and instructions which you can run in order.

Even if this is only a test session and not a real experiment, you still need to have one task and one participant. In such trial version, just pretend the participant is a student and enter a name.

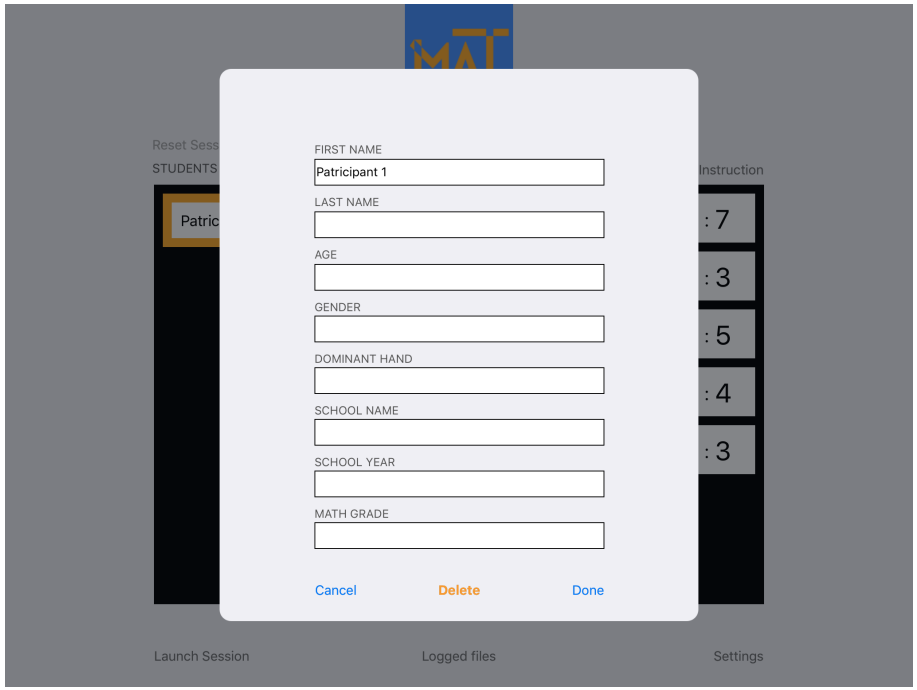
Click on **Add Student**



A data entry form pops up to enter the participant's information.

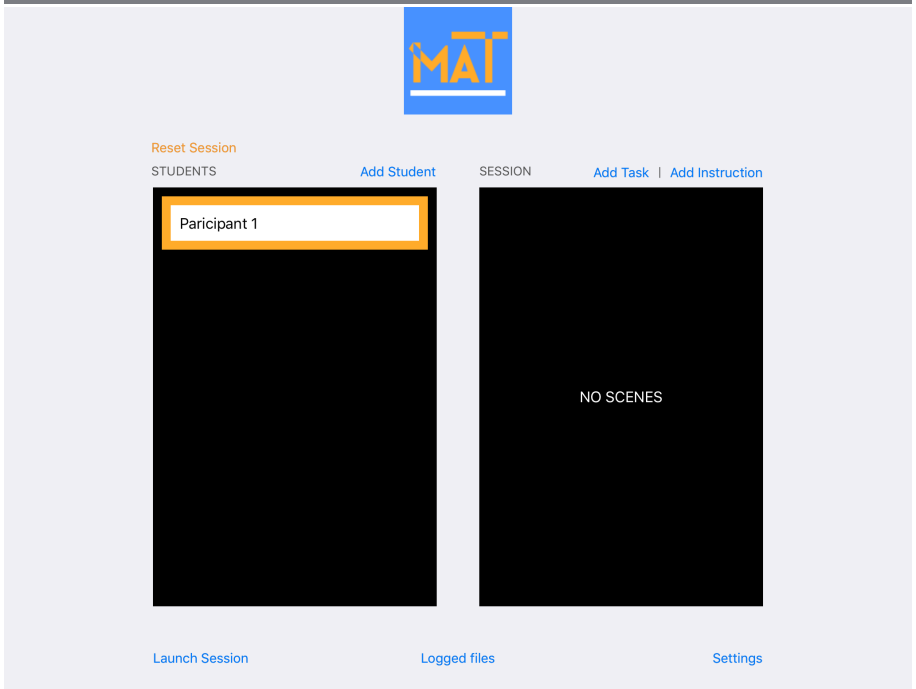
If you just want to have a test session, having a name is enough.

Click on **Done** when you are done

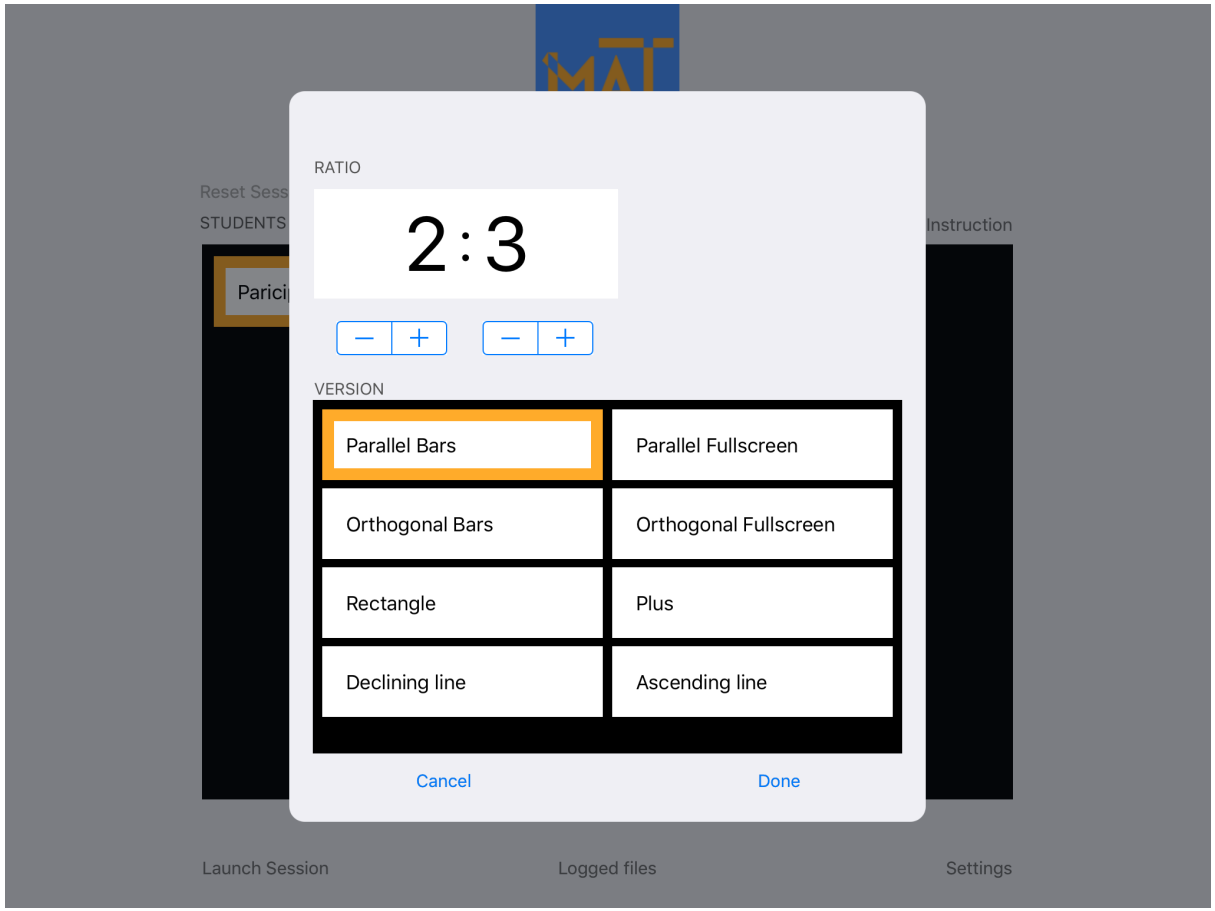


You are back to the main page, with the name of your participant added to the left.

Existing participant's information can be edited by double tapping on the participant's name.It pops up the same entry window and allows for edit or deleting.

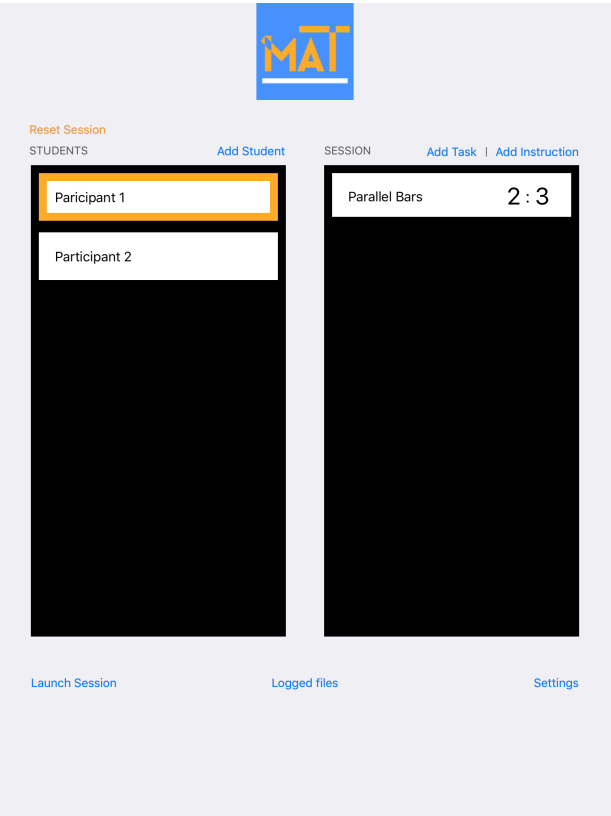


You need to add at least one task on the right plane to be able to start a working session, whether it is a real session or a trial. Click on **Add Task** to choose from different versions of the task.



On the new window that pops up you can set the intended proportion where it says **ratio**. If the ratio is 1:2 it means that the green feedback will be given when the right hand is moving twice as fast the left hand. Of course movement is not necessary to get the green feedback, the pre-set ratio applies to the static hand positions as well.
The difference between all the eight listed tasks is in the visualization of the feedback, as well as the direction of the hand movement.
Click **Done** to continue.

As the name indicates **Parallel Bars** and **Parallel Fullscreen** tasks, involve parallel finger movement on the screen. All the rest involve orthogonal hand movement. For orthogonal tasks, the app has to be started in landscape mode to work properly.

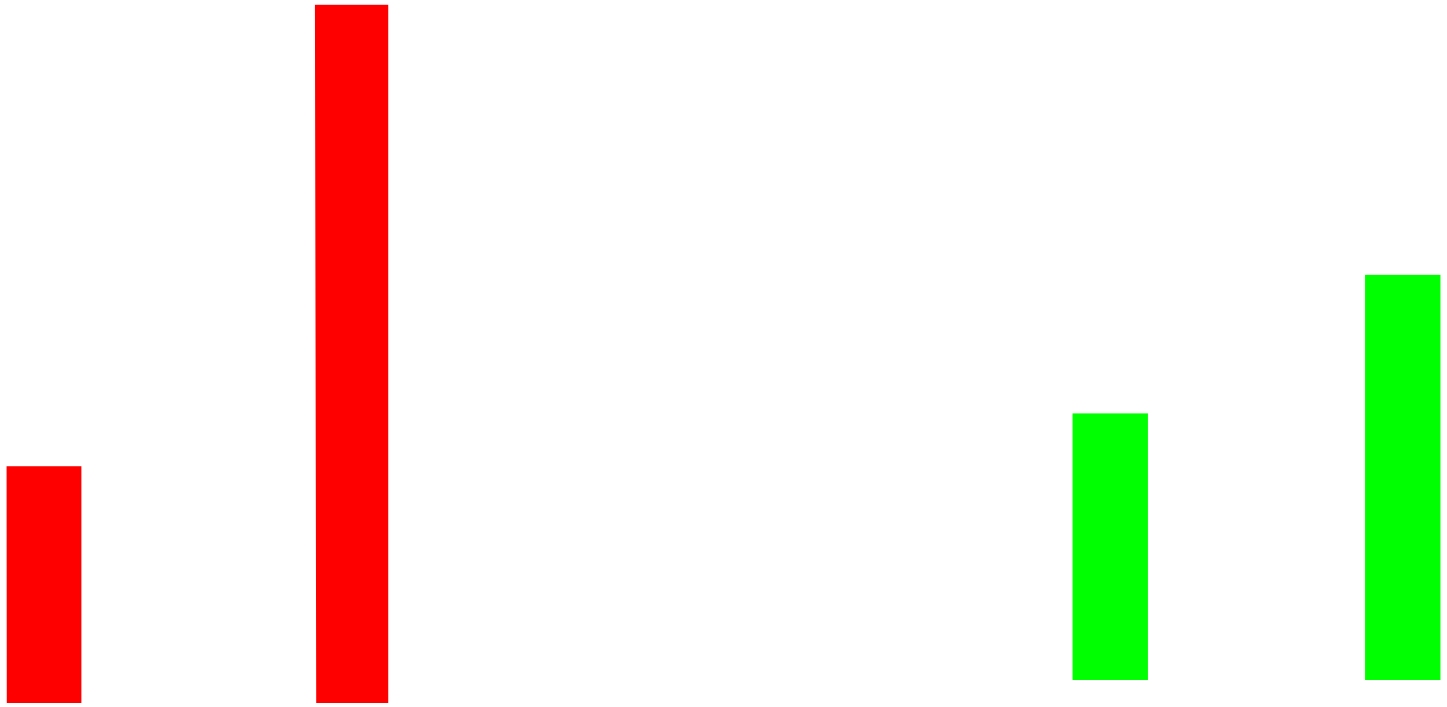


Here we have created a parallel bar task with a setting of 2:3 .It should be noted that due to the dimensions of the iPad, it is best if the app is started in portrait set up when working on parallel tasks, however parallel tasks run in landscape orientation as well.

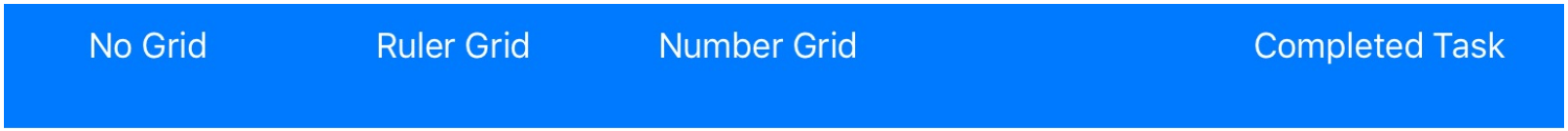
Next, click on the name of the participant that runs the task (Participant 1 here) and click on **Launch Session**

A white screen appears. You have to know roughly where the bars will appear on the screen, put your fingers on those positions and start moving. You can set the X coordinate of the bars as well as the distance between the two bars in **Settings**.

Once you put your fingers on the designated locations, you will see the bars appear in red or green, depending on the position of your fingers. In this example the green feedback appears only at the 2:3 proportional positions. Move your fingers up and down with this ratio and it will be a steady green.



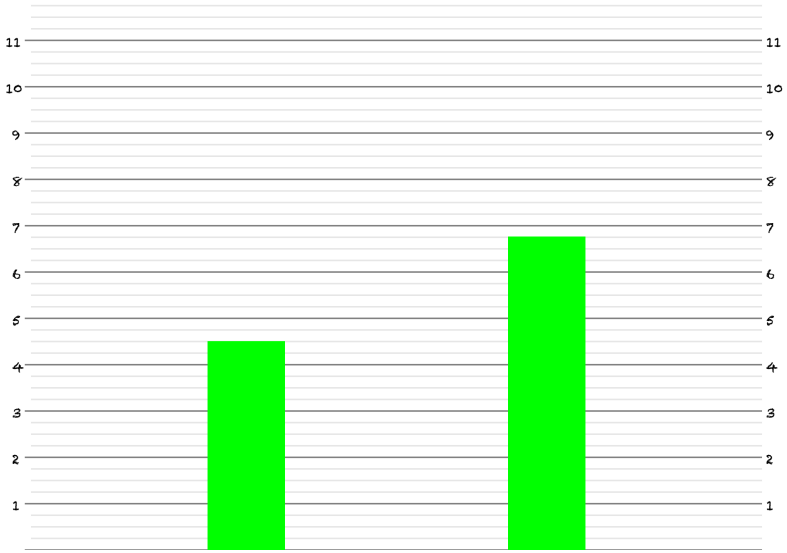
You can change the screen to a **Ruler Grid** view where a grid appears on the screen, enabling a concrete measurement of the height of the bars. It is also possible to add numbers next to the grid by selecting the **Number Grid** view. These options appear by tapping three times on the screen with both fingers. You will see a blue bar appear on the top of the screen with these options. Finally you can finish the task by selecting **Completed Task**.



Add **Ruler Grid**

Add **Number Grid**

Start moving your finger

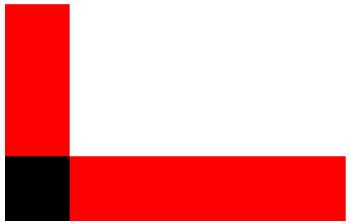


This was one example of a full run of one task. As mentioned before, the idea behind all the other tasks is the same; The two fingers need to move proportional to one another to get the green feedback, otherwise the feedback will turn gradually to red, depending on the error in positioning the fingers.

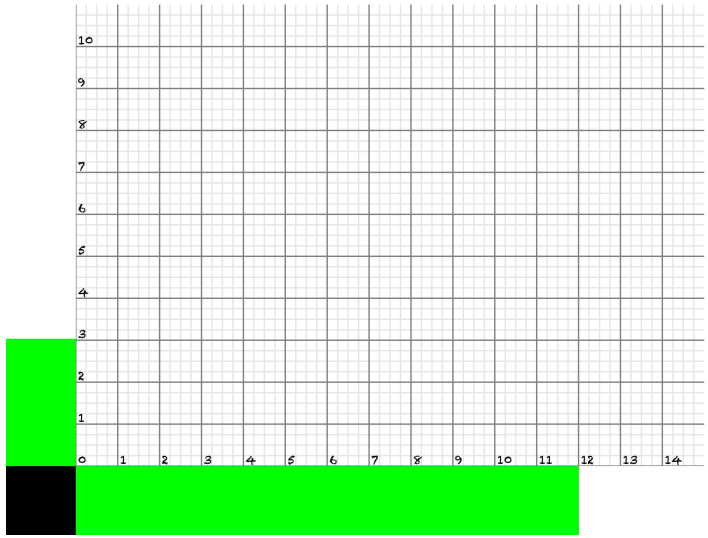
As a different example of a different task and set up, let's try **Orthogonal Bars** instead of **Parallel bars**. First start the app in landscape mode, by turning the iPad to landscape position and then restarting the app.

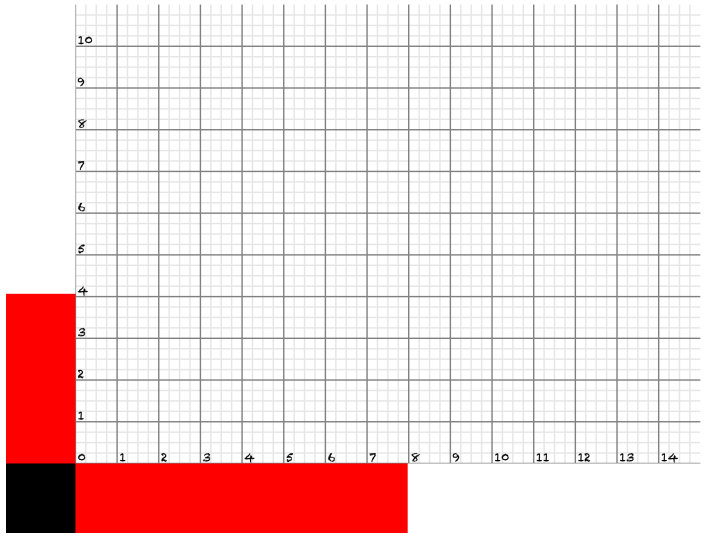
Double tap on the task and change it to **Orthogonal Bars**, change the proportion setting to 1:4 as well. Click on done. Select a participant and click on **Launch Session.**

Here is how you have to move your fingers differently.

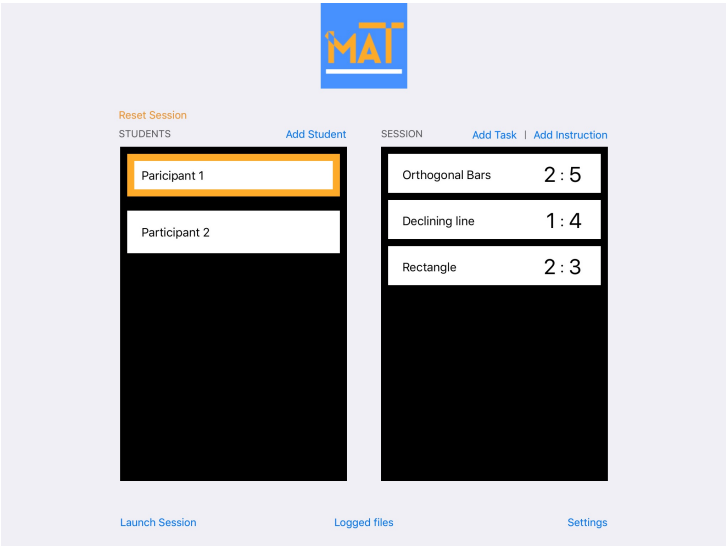


And how it looks like with grid and numbering.

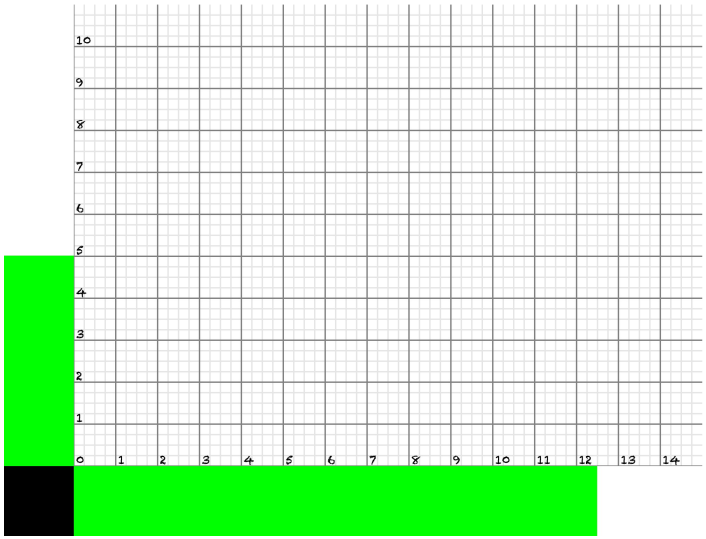




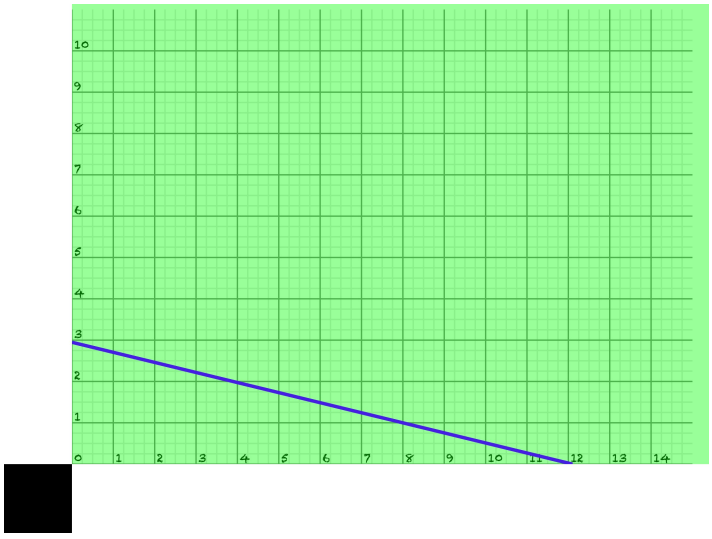
It is also possible to run different tasks one after another, by defining several tasks in a session. You can also add instruction in between tasks by clicking on **Add Instruction**. Notice that in such set up, you have to manually end one task to start the next. You can end a task any time by three times click of both fingers and selecting **Completed Task**.
This example set up will result in what is depicted below in order:



Task 1: **Orthogonal Bars**



Task 2 : **Declining line**

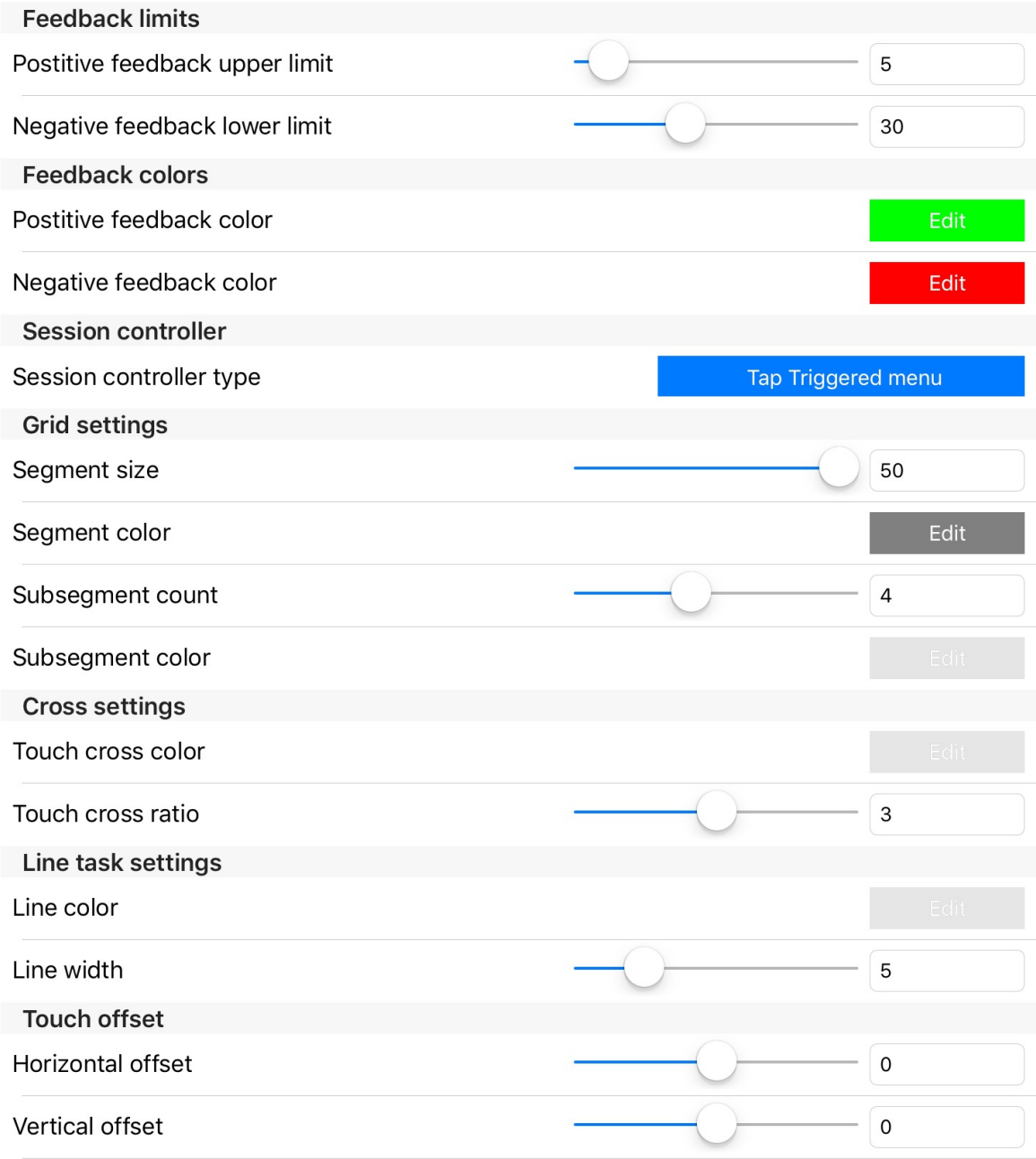


Task 3 : **Rectangle**

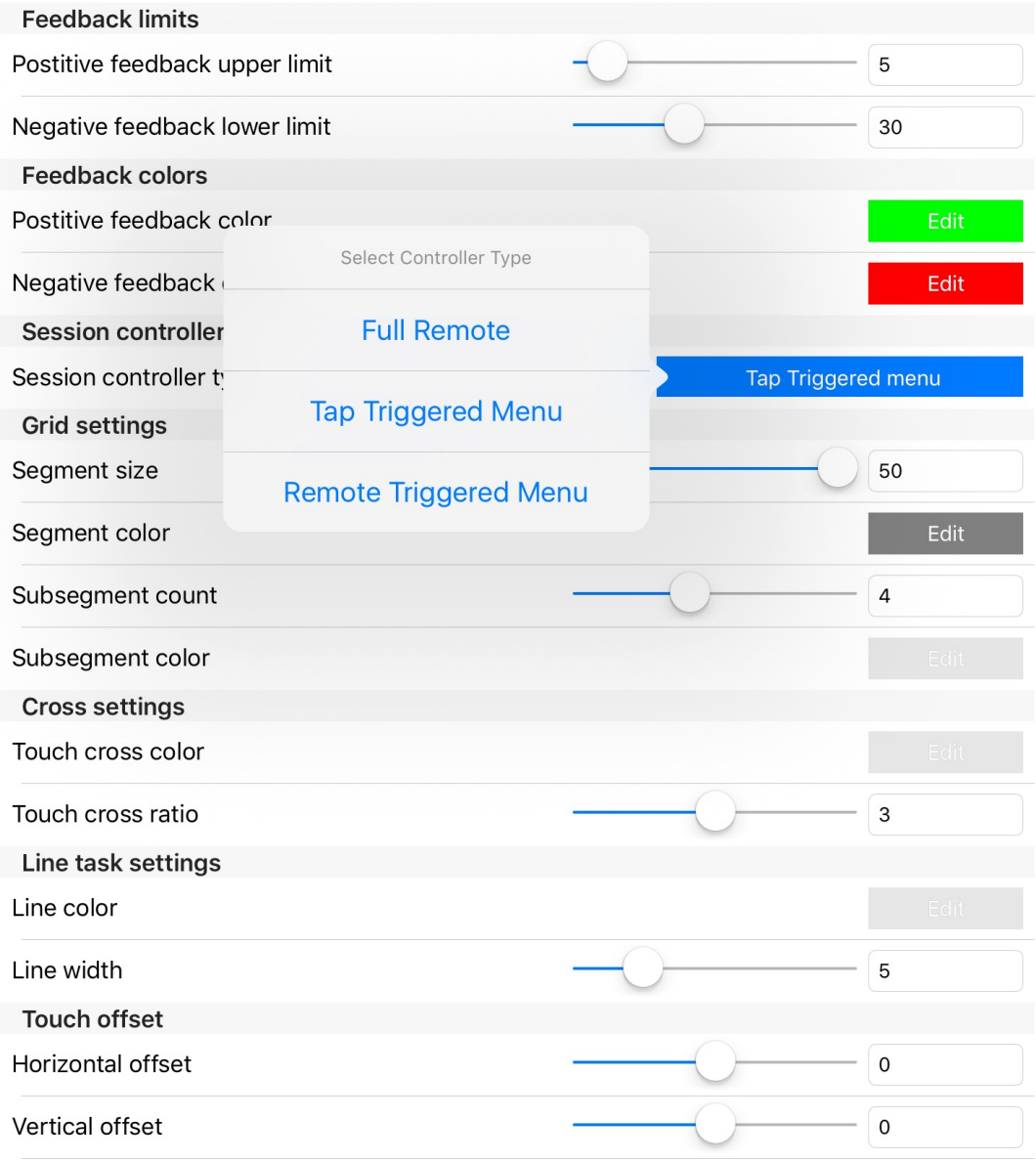
Finally you can change the visual appearance features by clicking on **Settings**.

Feedback limits allows for setting the margin of error and turn the green feedback gradually and smoothly to red.

You can also change the color for **Positive Feedback** and **Negative Feedback** to any other color using the Edit button. The default is *green* for **Positive Feedback** and *red* for **Negative Feedback.** The behavior of the app so far was described based on this default, where green meant positive and red meant negative feedback



Session controller type, allows for choosing a remote controller instead of triggering the blue bar menu and shifting between views or ending the task. In Full Remote type you can use a remote audio controller device or any earphone with audio controller and by clicking on next and previous button shift back and forth between views. To end the task you can double click on next button.
In Remote Triggered Menu

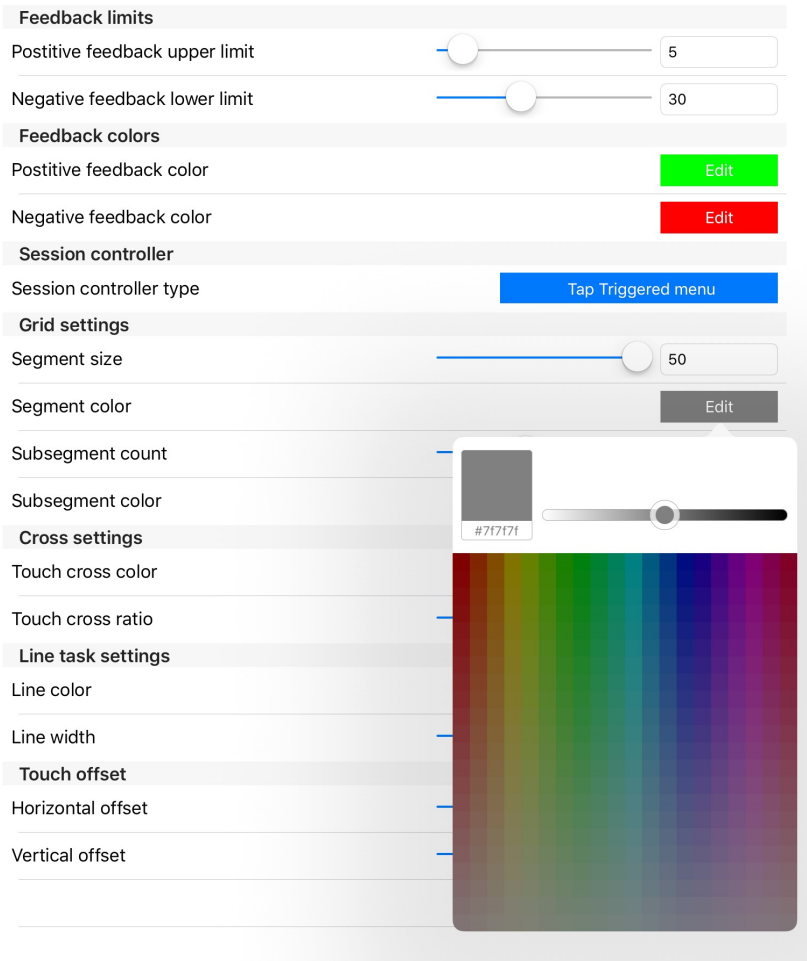


Grid settings is to define the size and color of the *segments* and *subsegments* in grid views. The segmenting unit is in pixels. Number of subsegments can be changed within a segment using *Subsegment count.*

You can change the grid line colors by clicking on Edit, the default is dark grey. Similarly subsegment line color can be changed.

Cross settings is meant for changing the size and color of the little cross that is used as placeholder in *orthogonal* and *parallel fullscreen* tasks.

Line task settings sets the color and width of the line in *declining* and *ascending line* task.



Touch offset sets the margin of error offset around the touching point vertically and horizontally.