# Learning Objectives

1. To work on how we design confounds out of an experiment
2. To practice using PsychoPy Builder
3. To add a practice block
4. To expand on use of conds files

# Activity 5.0

## Designing confounds out of an experiment

In Seminar 2 you were asked to identify any potential experimental confounds in the design of the CRT study. A primary confound was that the negative angry face was male, and the positive happy face was female, therefore if we ran that experiment online, we could not know for certain if any effect we might find was due to the positive/negative conditions, or the confounding male/female issue. To try to remedy this we will develop the experiment further and add more positive/negative stimuli to the program.

1. In the **Activity Set 5** page of the **website** there is a new set of stimuli for you to download. You can use your existing CRT experiment folder and delete the stimuli in the **stims** folder, and replace with these new ones
2. In the **conds** file for the CRT experiment add a **new column** for the sex of the poser in the image and update the columns with the new **faces** image file names (numbered 1-8 happy/angry – *we will be using files ending in a 9 later*), the **poserSex**, and the **corrAns**. Save the Excel file
3. We now have a new IV in this experiment. **What is it?**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Return to PsychoPy Builder and click on the **trialsLoop** to browse for the newly updated **conds** file to update the **Conditions** box. Once uploaded you will see the new conditions and parameters below the conditions box. It should resemble Figure 1

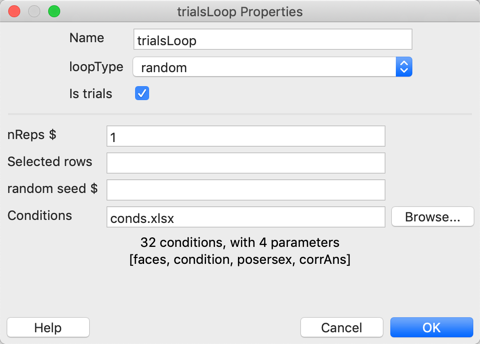


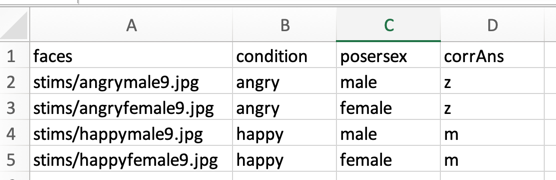
Figure 1

1. Run through the experiment and check that it all works as it should. If it does not run, you must figure out the reason why, and fix the issues before you can proceed
2. Describe your understanding of the rationale for this, and how it has rectified some of the confounds in the experiment \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Activity 5.1**

## Adding a practice block

Many experiments will require a practice session to ensure that participants are responding correctly and understand what it is they must do. One way to do this is to **create a new Excel file** to apply a conds file for the practice block. (there are other methods, but this is the simplest).



**Figure** **2**

You should have your new **CRT.psyexp** open in PsychoPy Builder

1. In Excel create a new conds file called **prac.xlsx** and this will be a shorter version of the **conds.xlsx** file only using the stimuli that end with number 9 (Figure 2)

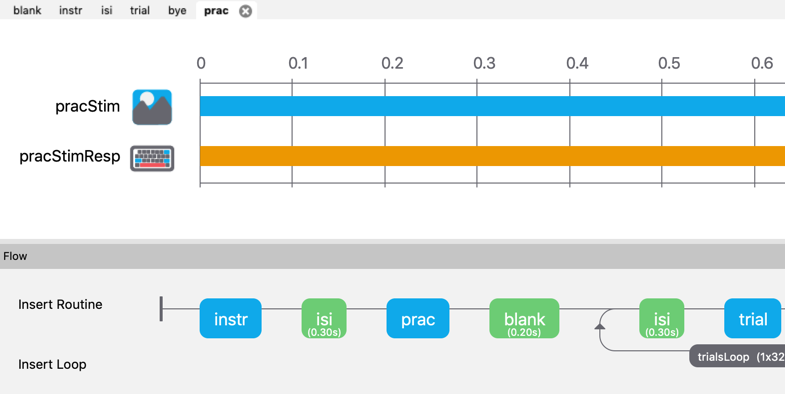
## Copying routines to retain ALL properties

1. On the Flow, you now need to add three more routines, instead of selecting new as you done previously, you can add a copy of two of the existing trial routines (**isi, blank**) by clicking **Routine 🡪[routine\_name].** As you have copied these two routines, you do not need to edit them, as all the relevant properties have been carried across

**NOTE:** With this copy method, if you change the properties in one instance of the routine, it will change the properties in **ALL** instances of the routine with the same name

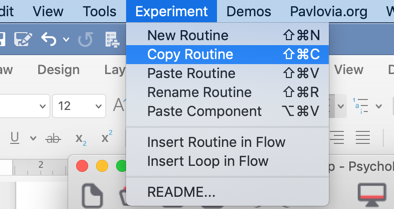
## Copying routines to allow you to change some properties

Next you can copy thetrial routine but in this method it which the properties, but they can be edited.



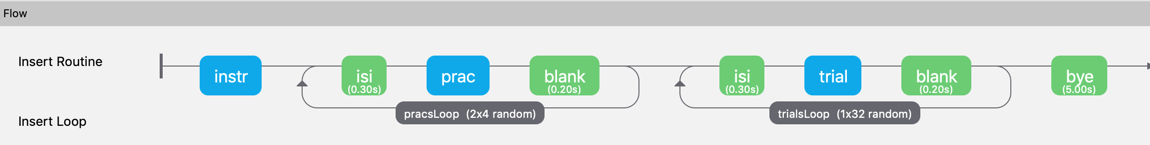
**Figure**

1. Select the **trial routine🡪 go to the top Psychopy Builder menu,** and select **Experiment🡪Copy Routine🡪Paste Routine** and **name it prac** (Figure 4)



**Figure**

1. Now insert a **new Loop** around the three newly created routines, call it **pracsLoop.** Browse for **prac.xlsx** conditions file



**Figure**

1. Change the instructions before the practice to reflect that this block is a practice block. Add instructions before the experimental block. Run the program through to check it works

# Activity 5.2

## More about conditions files

We can use conditions files to do a lot more than just store the stimuli, the condition/s and the correct responses. Most often we would use these if we had several blocks and wanted to repeat certain properties without having to edit them in the component properties for every single component. Here are some possibilities to try out.

## Changing image size using conds

We might want to change the size of the stimuli dynamically. We could go into a photo editing software and resize the images which could take a long time if you have many images, or we can use the conds file to automatically fit the images to the required size

#### Method 1

1. In the conds file add two new columns, call them **width\_x** and **height\_y** and type in a new size in pixels (try 350, 480) and save the conds file
2. Go to the **faceStim properties 🡪Layout** and in place of the pixel sizes specified type in **(width\_x, height\_y)** and **set to every repeat**
3. Save, replace the conds file in the loop and run the experiment

#### Method 2

1. A second way is not to use pixels, but to use the **height spatial unit** instead. Create a column in the conds file called **useHeight**, and type in **(0.5, 0.7)** (50% wide and 70% of the screen in height) and **set to every repeat**
2. Go to **faceStimProperties🡪Layout** and in the **Size** field type **(useHeight)**
3. Make sure the spatial units for the image is set to height too. Height displays stimuli as a percentage of the screen.In this case the images become somewhat pixelated as they are too small to legitimately stretch to 50x70% of the screen. Try some different values to see what is an optimal size for the image quality and make some notes below for what worked and what failed! \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Changing the orientation of a stimulus

1. We might want to display the same stimulus in differing rotations on the screen. In the **conds file** add a column can call it **dynamicOri**. In this column type some values between 1 and 180 to orient from its original position in a 180deg rotation. Change the relevant properties of the stimulus **Orientation** field to reference the **dynamicOri** column and run it. Remember you can use - to orient the image in the other direction
2. What else do you need to do before this will run?
   1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. In adding orientation of the stimuli, we have also introduced a new independent variable. What considerations would you want to describe in a method section regarding this new IV? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Changing colour of stimuli

1. You can specify colours in the **conds files** too. In a new column type in **fixCol** and type in some colour names in the cells below (red, blue, green, yellow for example)
2. In the **fixation component properties🡪Appearance🡪** change the relevant properties for the **fill colour** and **line colour** to correspond to the column in your conds file using **$(fixCol)** in the properties
3. If you copied the fixation component directly from the trial loop, you will need to modify the prac conds file to reflect the changes in the original trial conds for the colour of the fixation cross

## Changing text size using conds

1. We might want to change the properties of a text component dynamically.
2. Add a text component after the blank screen component, call it **goNext**
3. In the conds file type a heading **nextTxt**, and in the column type in simple colour names such as red, green, blue, orange, yellow etc
4. In the foreground colour property change the font colour to reference the **nextTxt** column in the conds file (**$nextTxt**) and **set to every repeat**
5. You can try changing the size of the font, what text will repeat, where it is positioned on screen, play about and try to get it to do a variety of things with small changes in the properties and the conds file. If you hit an error message, try to figure out what the problem is by looking carefully at the error message.

These are some simple examples of manipulations that you can make using a conds file, so while it is primarily useful as the repository for your stimuli and specifying correct answers, you can also use it to specify other properties for your components if they are to be regularly used, or change on each repeat.