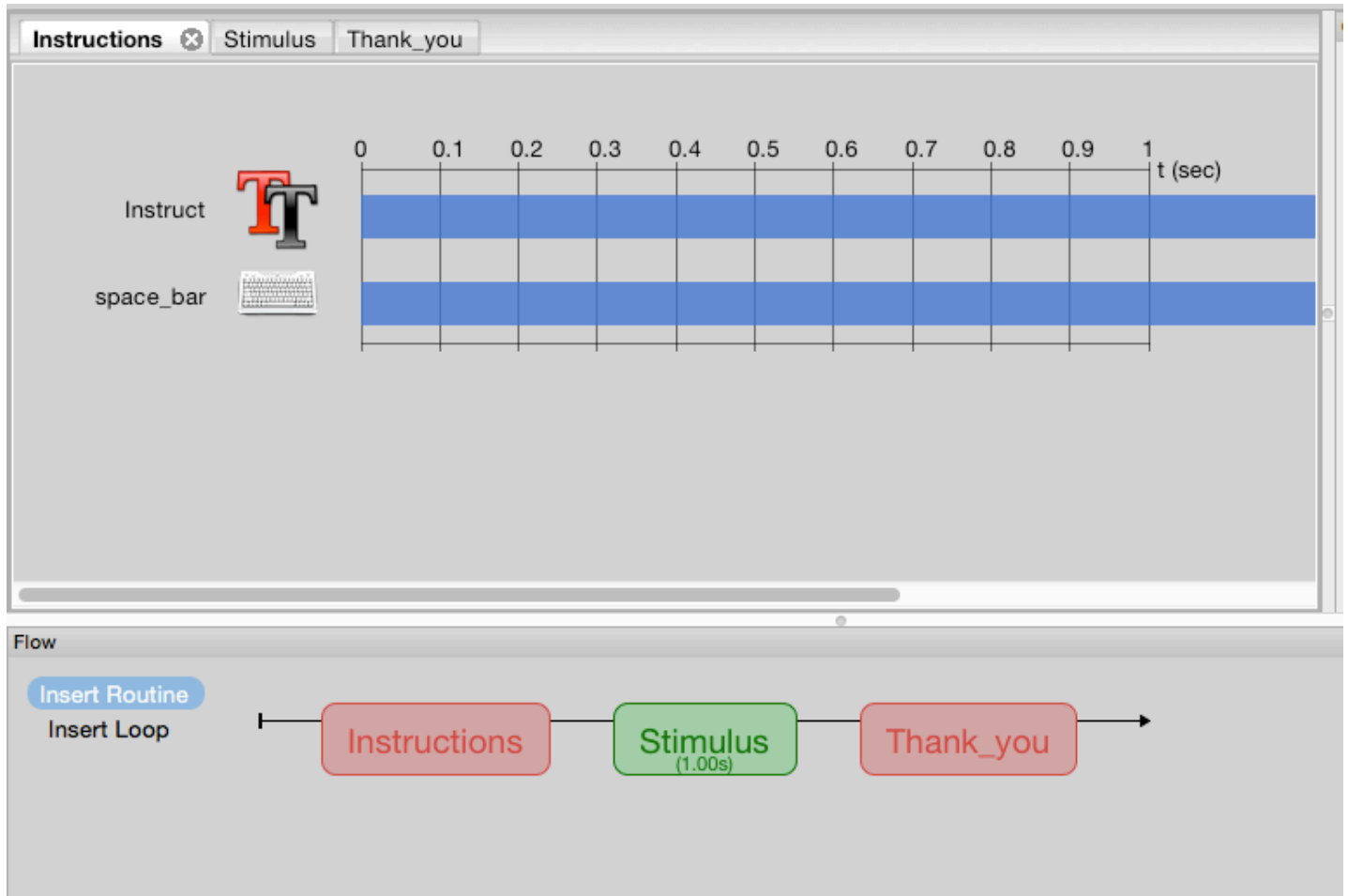



Looping through conditions


Now that we have our movie structure:




we can add the movie's scene *Components*.

Click on *Stimulus* in the **Flow** to open the scene, then click  in the **Components** panel. A box pops up with several options for the image. Notice the default duration of an image is 1 second.

In the *Image* dialog box, we need tell *Psychopy* where to look for this component of the scene. In your *Psychopy* folder, there is a subdirectory called '*images*'. This subdirectory holds the images that can be used in this component. Lets use the *cat_eyes.jpeg* file. In the *Image* dialog box, enter the path to the file like so:

Image  Click OK

We can test to see if this works by running the movie. Click the Run icon  on the top right of the *Psychopy* window. Enter '001' as the participant number.


In your *Psychopy* directory, you will now see a '*data*' subdirectory with the data saved in the form of 001*filenamedate*.csv. We will discuss the contents of the .csv later.


The movie works, but usually there are many scenes in a movie, and in psychological experiments the scene is often repeated.

In the **Flow**, click the *Insert Loop* button. Click the first dot immediately following *Stimulus* and the second dot immediately before it. A dialog box appears with several options. Name it 'trials', the stimulus will loop 5 times by default.


Run  the experiment again.

You will have noticed that there was only one presentation of the kitten scene, any idea why? The kitten scene was actually presented for 1 second, 5 times, but there was no gap between the scenes.


An interstimulus interval (ISI) is required to separate each scene. In the **Components** panel, click *Custom* then the . The default ISI is 0.5 seconds. Click OK, there is now a pink shaded area for 0.5 seconds.

Put the ISI before the image by changing the start time of the  to 0.5 and the duration to 1.0.




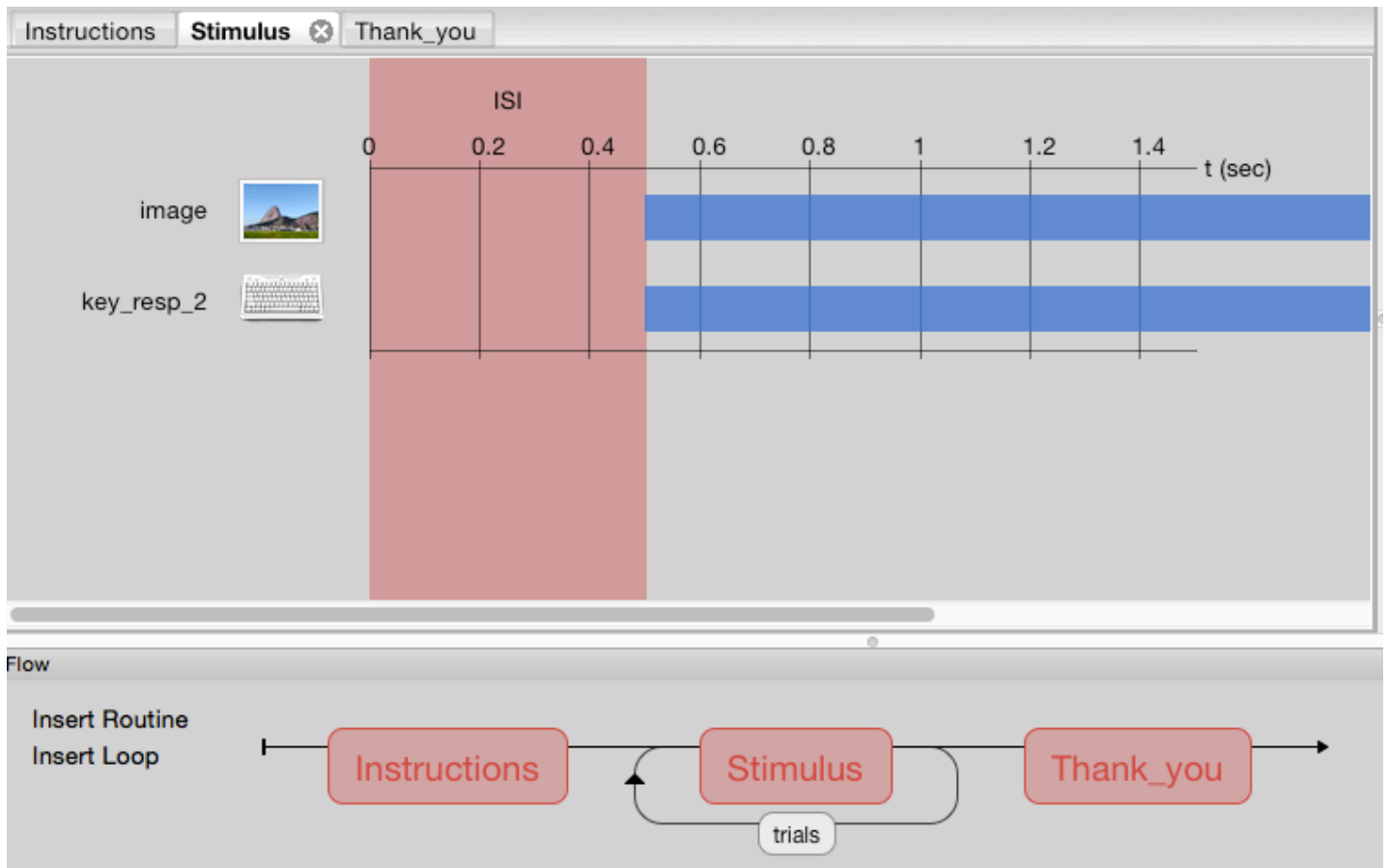
Run  the movie again.


So we have 5 scene presentations, but what about recording a response? Let's add a button press for reaction time and accuracy.

Click on the  in the **Components** panel. There are several options again, type 'space' into the *Allowed*

keys box to indicate a response by spacebar press. Click OK and the key response appears in the workspace.

Click again on the  in the scene, and change the duration of the image to 0. Now the picture will be presented until the spacebar is pressed.



We now have instructions and a scene that flashes for 1 second 5 times, and we can record a response. Click 

This is good, but we can also loop through different scenes.

Let's use a new example. Imagine that *Psychopy* is in the pizza delivery business.

Psychopy needs to collect the pizza order and deliver it to the customer, then receive the money. *Psychopy* looks at the list of pizzas to be delivered, and delivers each in turn until there are no more. Similarly, *Psychopy* can collect an image from a list, deliver it to the participant, and then receives a response, continuing until the list is done.

This list can be saved to an excel spreadsheet. In your *images* subdirectory, there is a file called *datafile.xlsx*. This contains a ready-made list.

	A	B	C
1	correct_ans	images	type_of_image
2	left	images/gorilla_face.jpeg	gorilla
3	left	images/gorilla_fat.jpeg	gorilla
4	left	images/gorilla_tongue.jpeg	gorilla
5	right	images/cat_eyes.jpeg	cat
6	right	images/ear_muffs_cat.jpeg	cat
7	right	images/rat_cat.jpeg	cat
8	up	images/egg_hog.jpeg	hog
9	up	images/icecream_hog.jpeg	hog
10	up	images/marshmallow_hog.jpeg	hog
11	up	images/santa_hog.jpeg	hog
12			

Like the path we entered to the kitten image earlier, the column *images* has a list of paths to images. There are additional columns in the spreadsheet, in the pizza example, these can be thought of as the drink and a dessert that accompany the pizza. The drink and dessert in each row are specific to the pizza in that row.

Lets set it up. In this example we will use images of gorillas, kittens and hedgehogs. The instructions are:

You will see pictures

Press left for gorillas

Press right for kittens

Press up for hedgehogs

Press spacebar to continue

Click on *Instructions* in the **Flow**, and select . Type the above into the *Text* box.

To tell *Psychopy* where to look for the datafile, click on *trials* in the **Flow** and select *Browse....* Select the datafile.xlsx and hit OK. The path to the file is now in the dialog box, and below there will be a summary of what is included. In this spreadsheet there are 10 rows (conditions) with 3 columns (parameters) as shown below.

trials Properties

Name trials

loopType random

Is trials ☒

random seed \$

nReps \$ 5

Selected rows

Conditions images/datafile.xlsx

10 conditions, with 3 parameters
[correct_ans, images, type_of_image]


Now, to tell *Psychopy* to deliver the images from this list, click on *Stimulus* in the **Flow**, and the . In the *Image* box, type '\$images'. The \$ tells *Psychopy* to look for the column called *images* in the excel spreadsheet. Change the associated dropdown box to *set every repeat*. The image Properties should look like this:

image Properties

Basic Advanced

Name image

Start time (s) 0.0
Expected start (s)

Stop duration (s)
Expected duration (s)

Image \$images set every repeat

Position [x,y] \$ [0, 0] constant

Size [w,h] \$ [0.5, 0.5] constant


Orientation \$ 0 constant

Opacity \$ 1 constant

Units from exp settings

Help OK Cancel

Click OK.

We now need to record the amount of money received following pizza delivery, or the button press received following the image delivery. Click the  and type 'left', 'right', 'up' into *Allowed keys*. In the *Store* dropdown, select *first key*, and change the *Start* to 0.5 as shown below.

key_resp_2 Properties

Basic

Name

Start
Expected start (s)

Stop
Expected duration (s)

Force end of Routine ☒

Allowed keys \$

Store

Store correct ☐


Discard previous ☒

sync RT with screen ☒

Help OK Cancel

Your scene should now look like this:



Let's do it! Click 

Data time!