If you have troubles syncing the data online: make sure to start fresh:

A new Pavlovia project name from the Psychopy builder folder

* Delete:
* Html
* Git
* Data
* Gitignore
* Generated javacsript files

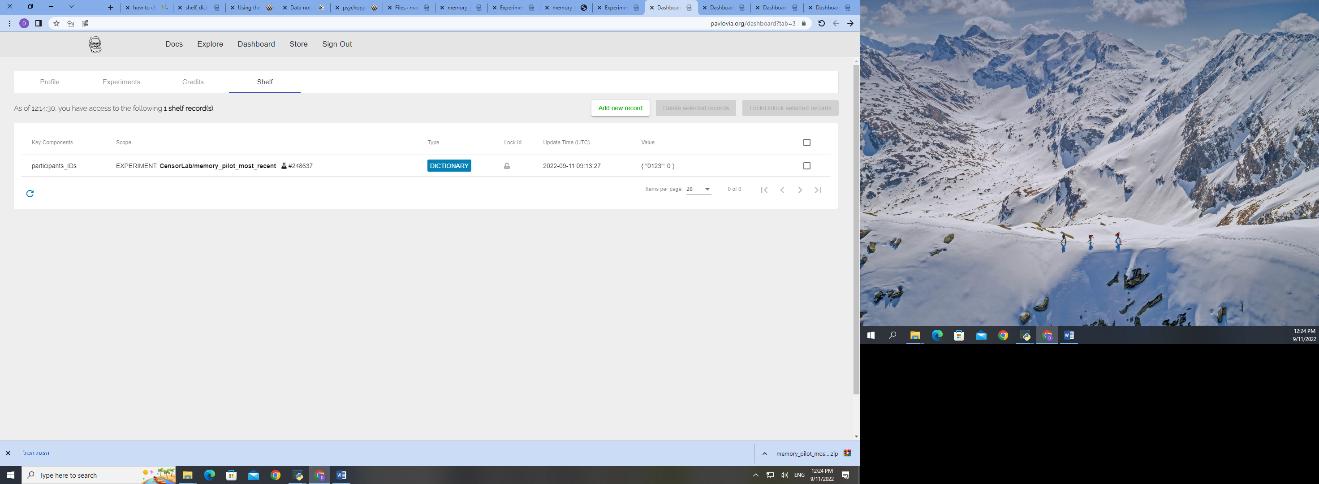
Modify java-script code of uploaded experiment to Pavlovia such that it uses a record of participants (shelf) which allows to serve them individualized data:

1. Above the init function loop (line 260~) define these variables:

var existing\_participants;

var num\_keys\_in\_dict;

var subject\_number;

1. Add the code to check a shelf records (make sure to create one for your experiment, add “participants\_IDs” as key word:
2. Add the following code within the scope of **experimentInit()**

// check the fieldnames in the dictionary - does this participant already exist?

existing\_participants = await psychoJS.shelf.getDictionaryFieldNames({key: ["participants\_IDs"]})

num\_keys\_in\_dict=Object.keys(existing\_participants).length

console.log(num\_keys\_in\_dict)

// if not add a new fieldname to this dictionary

if(!existing\_participants.includes(expInfo['workID'])){

psychoJS.shelf.setDictionaryFieldValue({key: ["participants\_IDs"], fieldName: expInfo['workID'], fieldValue :num\_keys\_in\_dict})

subject\_number=num\_keys\_in\_dict;

}else{//increase the number of sessions completed by this participant

subject\_number = await psychoJS.shelf.getDictionaryFieldValue({key: ["participants\_IDs"], fieldName:expInfo['workID'], defaultValue:'no sessions detected'})

subject\_number = subject\_number;

console.log('session\_number2', subject\_number)

psychoJS.shelf.setDictionaryFieldValue({key: ["participants\_IDs"], fieldName: expInfo['workID'], fieldValue :subject\_number})

}

1. Change in the code that indeed loading depedends on the variable “subject\_number” that is created based on the dictionary:
   1. 1486: trialList: 'sub\_encoding' + subject\_number.toString() + '.csv'
   2. 1552: trialList: 'sub\_test' + subject\_number.toString() + '.csv'
2. Change the code that lets you more forward only on left or right arrow for demo encoding and test encoding:

DEMO ENCODING

////////////// custom code start /////////////////

if (demo\_encoding\_response.status === PsychoJS.Status.STARTED)

{

let theseKeys = [];

console.log(DemoImage)

if ((DemoImage == 'left.jpg')|| (DemoImage == 'right.jpg'))

{ console.log('###################')

console.log('entered demo image')

theseKeys = demo\_encoding\_response.getKeys({keyList: ['left', 'right', 'num\_4', 'num\_6'], waitRelease: false});

console.log(theseKeys)

console.log('###################')

}

\_demo\_encoding\_response\_allKeys = \_demo\_encoding\_response\_allKeys.concat(theseKeys);

if (\_demo\_encoding\_response\_allKeys.length > 0)

{

demo\_encoding\_response.keys = \_demo\_encoding\_response\_allKeys[\_demo\_encoding\_response\_allKeys.length - 1].name; // just the last key pressed

demo\_encoding\_response.rt = \_demo\_encoding\_response\_allKeys[\_demo\_encoding\_response\_allKeys.length - 1].rt;

// was this correct?

if (demo\_encoding\_response.keys == DemoCorrect) {

demo\_encoding\_response.corr = 1;

} else {

demo\_encoding\_response.corr = 0;

}

// a response ends the routine

continueRoutine = false;

console.log(theseKeys);

}

}

TEST ENCODING

////////////// custom code start /////////////////

if (test\_encoding\_response.status === PsychoJS.Status.STARTED)

{

let theseKeys = [];

console.log(target\_image)

if ((target\_image == 'left.jpg')|| (target\_image == 'right.jpg'))

{ console.log('###################')

console.log('entered demo image')

theseKeys = test\_encoding\_response.getKeys({keyList: ['left', 'right', 'num\_4', 'num\_6'], waitRelease: false});

console.log(theseKeys)

console.log('###################')

}

\_test\_encoding\_response\_allKeys = \_test\_encoding\_response\_allKeys.concat(theseKeys);

if (\_test\_encoding\_response\_allKeys.length > 0)

{

test\_encoding\_response.keys = \_test\_encoding\_response\_allKeys[\_test\_encoding\_response\_allKeys.length - 1].name; // just the last key pressed

test\_encoding\_response.rt = \_test\_encoding\_response\_allKeys[\_test\_encoding\_response\_allKeys.length - 1].rt;

// was this correct?

if (test\_encoding\_response.keys == correct) {

test\_encoding\_response.corr = 1;

} else {

test\_encoding\_response.corr = 0;

}

// a response ends the routine

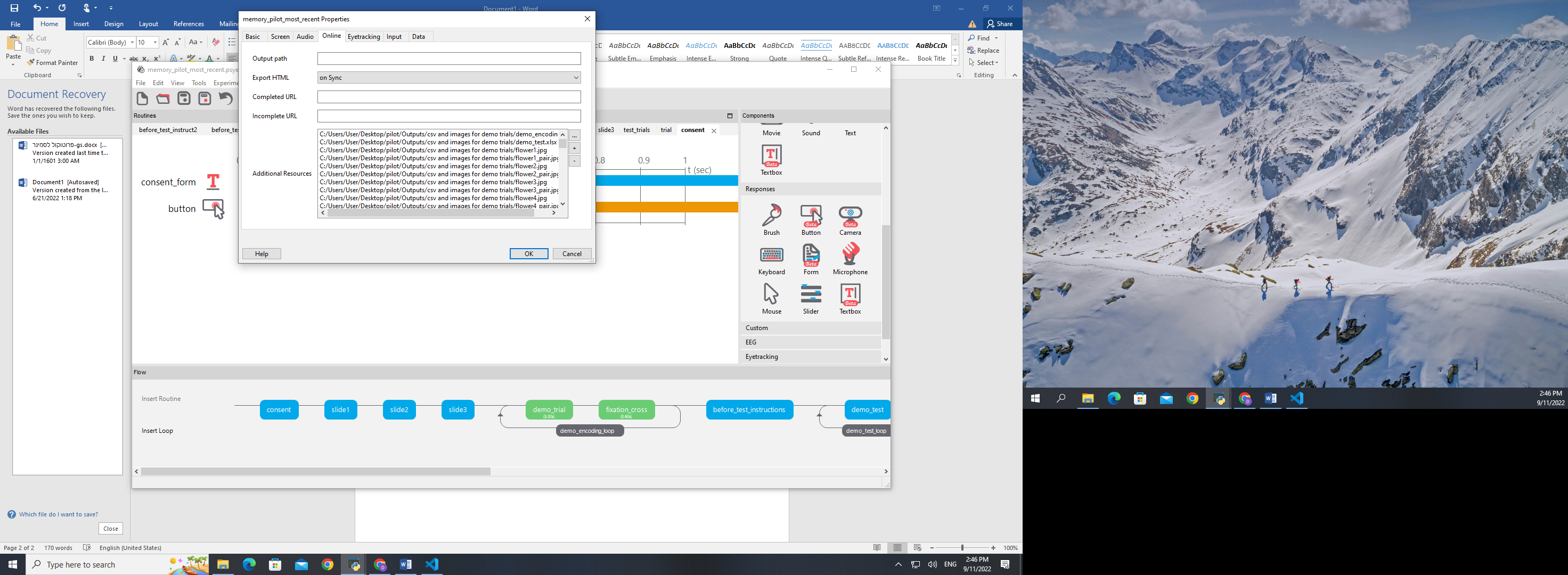
continueRoutine = false;

console.log(theseKeys);

}

}

1. Make sure the experiment has all the required stimuli and condition matrixes in its resources directory
   1. In Psychopy builder press the “settings icon” -> online - > make sure “Output path” is blank
   2. Press the plus icon, and add all the required resources:
      1. (for our experiment it)
         1. all csv conditions for all participant (1 csv per participant)
         2. all possible images (240)
         3. all demo images
         4. demo csv files
         5. attention manipulation checks images (left and right arrow



**Additional corrections especially for our memory experiment:**

1. Don’t allow for keyboard inputs while in the display (unless its an attention check trial)
   1. Add the following code:
2. Add screen size info to participant output:

psychoJS.experiment.addData('windowSize', window.size)