*// Returns a new array with elements of the array in random order.*

*// given some array of your possible combinations of treatments/conditions*

*// see the two arrays “*J\_T\_videos” or “R\_R\_videos” that contain the two possible orders

function shuffle(array) {

var shuffled = Ember.$.extend(true, [], array); // deep copy array

for (var i = array.length - 1; i > 0; i--) {

var j = Math.floor(Math.random() \* (i + 1));

var temp = shuffled[i];

shuffled[i] = shuffled[j];

shuffled[j] = temp;

}

return shuffled;

}

*//*

*// Returns a random element of an array, and removes that element from the array*

*//*

function pop\_random(array) {

if (array.length) {

let randIndex = Math.floor(Math.random() \* array.length);

return array.splice(randIndex, 1)[0];

}

return null;

}

// *this array helps to counterbalance the videos that are used in the video selection later in the study (see: "video-select-1" and "video-select-2")*

*// a larger array containing an array of each and every possible condition/trial/etc order,*

**let J\_T\_videos** = [

// *first possible set of ways to present stimuli*

["JennyFalse-small\_v2", "TinaTrue-small\_v2"],

// *second possible set of ways to present stimuli*

["JennyTrue-small\_v2", "TinaFalse-small\_v2"]

];

// picks one of the two possible combinations/permuations/orders from J\_T\_videos to use

let trial\_videos\_1 = pop\_random(**J\_T\_videos**);

// sets the video of jenny to either be the one

// where she is incorrect (false) or correct (true)

let **Jenny\_video** = trial\_videos\_1[0];

// sets the video of tina to either be the one

// where she is incorrect (false) or correct (true)

let **Tina\_video** = trial\_videos\_1[1];

// *this array does the same for the rote vs. reflection section (see: "video-select-3" and "video-select-4")*

*// a larger array containing an array of each and every possible condition/trial/etc order,*

**let R\_R\_videos** = [

// *first possible set of ways to present stimuli*

["rote-kids-v1\_comp", "Teacher2\_Refl\_small"],

// *second possible set of ways to present stimuli*

["reflective-kids-v1\_comp", "Teacher2\_Rote\_small"]

];

let trial\_videos\_2 = pop\_random(R\_R\_videos);

let **Rot\_Ref\_teacher1** = trial\_videos\_2[0];

let **Rot\_Ref\_teacher2** = trial\_videos\_2[1];

let frames = {

"video-select-1": {

"kind": "exp-lookit-video",

"video": {

"position": "fill",

"loop": false,

"source": **Jenny\_video**,

"controls": true

},

"backgroundColor": "white",

"autoProceed": false,

"pauseWhenExitingFullscreen": false,

"allowUserPause": false,

"requiredDuration": 1,

"requireAudioCount": 0,

"requireVideoCount": 1,

"parentTextBlock": {

"title": "Please try to watch the video all the way through without pausing!",

"text": "Once the video finishes, you may replay it from the beginning if your child would like to watch it again.",

"css": {

"color": "gray",

"font-size": "small"

}

},

"baseDir": "https://raw.githubusercontent.com/ilonabass/lookit-stimuli-template/master/",

"audioTypes": [

"ogg",

"mp3"

],

"videoTypes": [

"webm",

"mp4"

]

},

// second video for general exclusions; shows 2nd teacher correct/true if 1st teacher wrong/false and vice-versa

// breaks here, probably breaks in second generateProperties below, as well -> UPDATE (02/16): no longer breaks here, no longer using generate Properties, using pop\_random()

//

"video-select-2": {

"kind": "exp-lookit-video",

"video": {

"position": "fill",

"loop": false,

"source": **Tina\_video**,

"controls": true

},

"backgroundColor": "white",

"autoProceed": false,

"pauseWhenExitingFullscreen": false,

"allowUserPause": false,

"requiredDuration": 1,

"requireAudioCount": 0,

"requireVideoCount": 1,

"parentTextBlock": {

"title": "Please try to watch the video all the way through without pausing!",

"text": "Once the video finishes, you may replay it from the beginning if your child would like to watch it again.",

"css": {

"color": "gray",

"font-size": "small"

}

},

"baseDir": "https://raw.githubusercontent.com/ilonabass/lookit-stimuli-template/master/",

"audioTypes": [

"ogg",

"mp3"

],

"videoTypes": [

"webm",

"mp4"

]

},

// frame for 3rd video; first rote-reflective task video

"video-select-3": {

"kind": "exp-lookit-video",

"video": {

"position": "fill",

"loop": false,

"source": **Rot\_Ref\_teacher1**,

"controls": true

},

"backgroundColor": "white",

"autoProceed": false,

"pauseWhenExitingFullscreen": false,

"allowUserPause": false,

"requiredDuration": 1,

"requireAudioCount": 0,

"requireVideoCount": 1,

"parentTextBlock": {

"title": "Please try to watch the video all the way through without pausing!",

"text": "Once the video finishes, you may replay it from the beginning if your child would like to watch it again.",

"css": {

"color": "gray",

"font-size": "small"

}

},

"baseDir": "https://raw.githubusercontent.com/ilonabass/lookit-stimuli-template/master/",

"audioTypes": [

"ogg",

"mp3"

],

"videoTypes": [

"webm",

"mp4"

]

},

"video-select-4": {

"kind": "exp-lookit-video",

"video": {

"position": "fill",

"loop": false,

"source": **Rot\_Ref\_teacher2**,

"controls": true

},

"backgroundColor": "white",

"autoProceed": false,

"pauseWhenExitingFullscreen": false,

"allowUserPause": false,

"requiredDuration": 1,

"requireAudioCount": 0,

"requireVideoCount": 1,

"parentTextBlock": {

"title": "Please try to watch the video all the way through without pausing!",

"text": "Once the video finishes, you may replay it from the beginning if your child would like to watch it again.",

"css": {

"color": "gray",

"font-size": "small"

}

},

"baseDir": "https://raw.githubusercontent.com/ilonabass/lookit-stimuli-template/master/",

"audioTypes": [

"ogg",

"mp3"

],

"videoTypes": [

"webm",

"mp4"

]

}

};

let frame\_sequence = [

"video-select-1",

"video-select-2",

"video-select-3",

"video-select-4"

];

// Return the study protocol with the "frames" and "sequence" fields defined above

return {

frames: frames,

sequence: frame\_sequence

};

}