Online Experiments Tutorial 03 – Random Card-Deck Design

Uri Hertz, PhD

Department of Cognitive Sciences

School of Psychological Science, University of Haifa

uhertz@cog.Haifa.ac.il

www.socialdecisionlab.net

אוניברסיטת חיפה
University of Haifa
جامعة حيفا

Social Decision Making Lab

What? Another tutorial?

We already know a lot about online experiments!

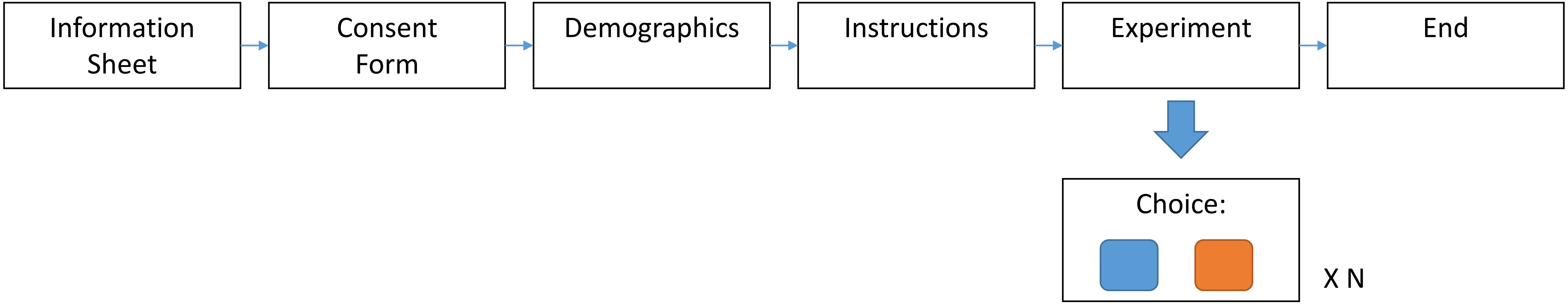
Here we will add some more components to our simple experiment:

1. Randomised experimental conditions
2. Demographics questionnaire using surveyJS
3. Progress bar :)#

The source files for this tutorial is on:

https://github.com/socialdecisionlab/JStutorial/tree/master/Tutorials_Code/TutorialRiskAmb

Design



Information Sheet
Consent Form
Demographics
Instructions
Experiment
End
Choice:
X N

Design

In our experiment participants will make a series of choices between a lottery and a sure sum of money.

There are two conditions in this experiment – risk and ambiguity.

There are two manipulations in each condition – the level or risk/ambiguity (3 levels each) and the sum of money to be won in the lottery (5 sums).

	25%	50%	75%
5$			
8$			
25$			
50$			
125$			

	25%	50%	75%
5$			
8$			
25$			
50$			
125$			

Setting up the variables in js

	25%	50%	75%
5$			
8$			
25$			
50$			
125$			

	25%	50%	75%
5$			
8$			
25$			
50$			
125$			

Just like before, we set up the experiment variables in the beginning of the script, immediately after the document ready check:

```
 var NumTrials = 60;//Number of trials
  var SubID = CreateCode();// random code for each participant


   var SumVec=[5,8,20,50,125]; // conditions in our experiments - levels of sums of money
   var RiskVec=[25,50,75];//levels of risk
    var AmbVec=[25,50,75];//levels of ambiguity
```

Setting up the variables in js

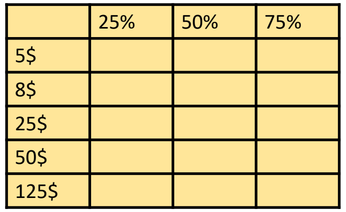
	25%	50%	75%
5$			
8$			
25$			
50$			
125$			

	25%	50%	75%
5$			
8$			
25$			
50$			
125$			

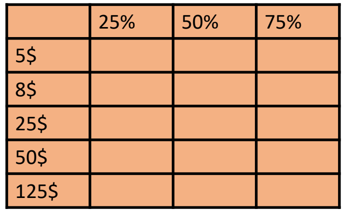
And we then create vectors with all the experimental conditions (cells in the tables):

```
  var RiskTrial=[];
  var AmbiguityTrial=[];
  var SumTrial=[];

  for (i = 0; i < 5; i++) {
     for (j=0;j<3;j++){
        RiskTrial.push(0);
        AmbiguityTrial.push(AmbVec[j]);
        SumTrial.push(SumVec[i]);
     }
  }

  for (i = 0; i < 5; i++) {
     for (j=0;j<3;j++){
        AmbiguityTrial.push(0);
        RiskTrial.push(RiskVec[j]);
        SumTrial.push(SumVec[i]);
  }
}
```

Setting up the variables in js



	25%	50%	75%
5$			
8$			
25$			
50$			
125$			



	25%	50%	75%
5$			
8$			
25$			
50$			
125$			

We actually do it twice, to have each condition repeating twice.

We now generate an ‘order’ vector, which is just the numbers between 1 and 60, and then we shuffle it. It will determine the order in which the experimental conditions are presented to the participant, generating a different order for each participant.

```
 var Order = [];
for (i = 0; i < 60; i++) {
      Order.push(i);
   }
   Order = Shuffle(Order);
```

Setting up the variables in js

	25%	50%	75%
5$			
8$			
25$			
50$			
125$			

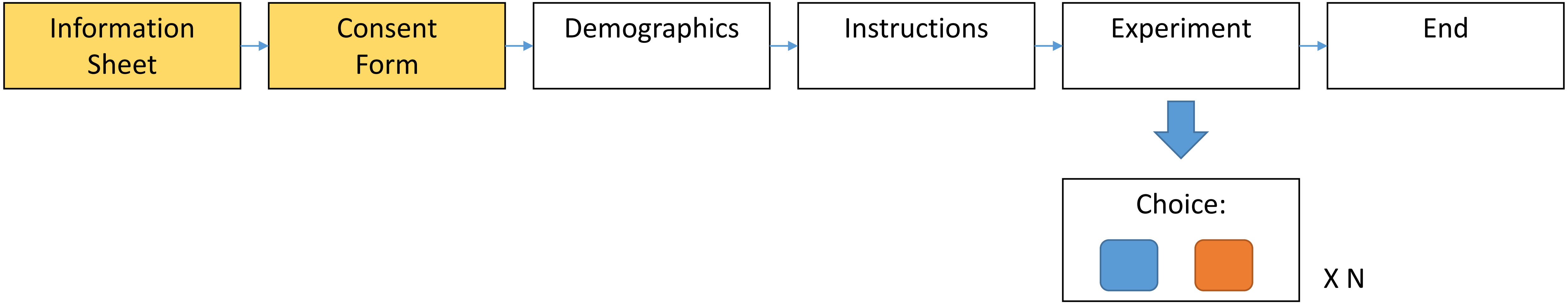
	25%	50%	75%
5$			
8$			
25$			
50$			
125$			

```
function Shuffle(array) {

    var currentIndex = array.length,
      temporaryValue, randomIndex;

    // While there remain elements to shuffle...
    while (0 !== currentIndex) {

      // Pick a remaining element...
      randomIndex = Math.floor(Math.random() * currentIndex);
      currentIndex -= 1;

      // And swap it with the current element.
      temporaryValue = array[currentIndex];
      array[currentIndex] = array[randomIndex];
      array[randomIndex] = temporaryValue;
    }

    return array;
  };
```

Design

We are now ready to go on with the experimental flow – you already know how to move from information to consent.

I actually added some more information now in the information sheet, and a mechanism to check the subjects' worker ID for duplication in the consent form – explore the code.

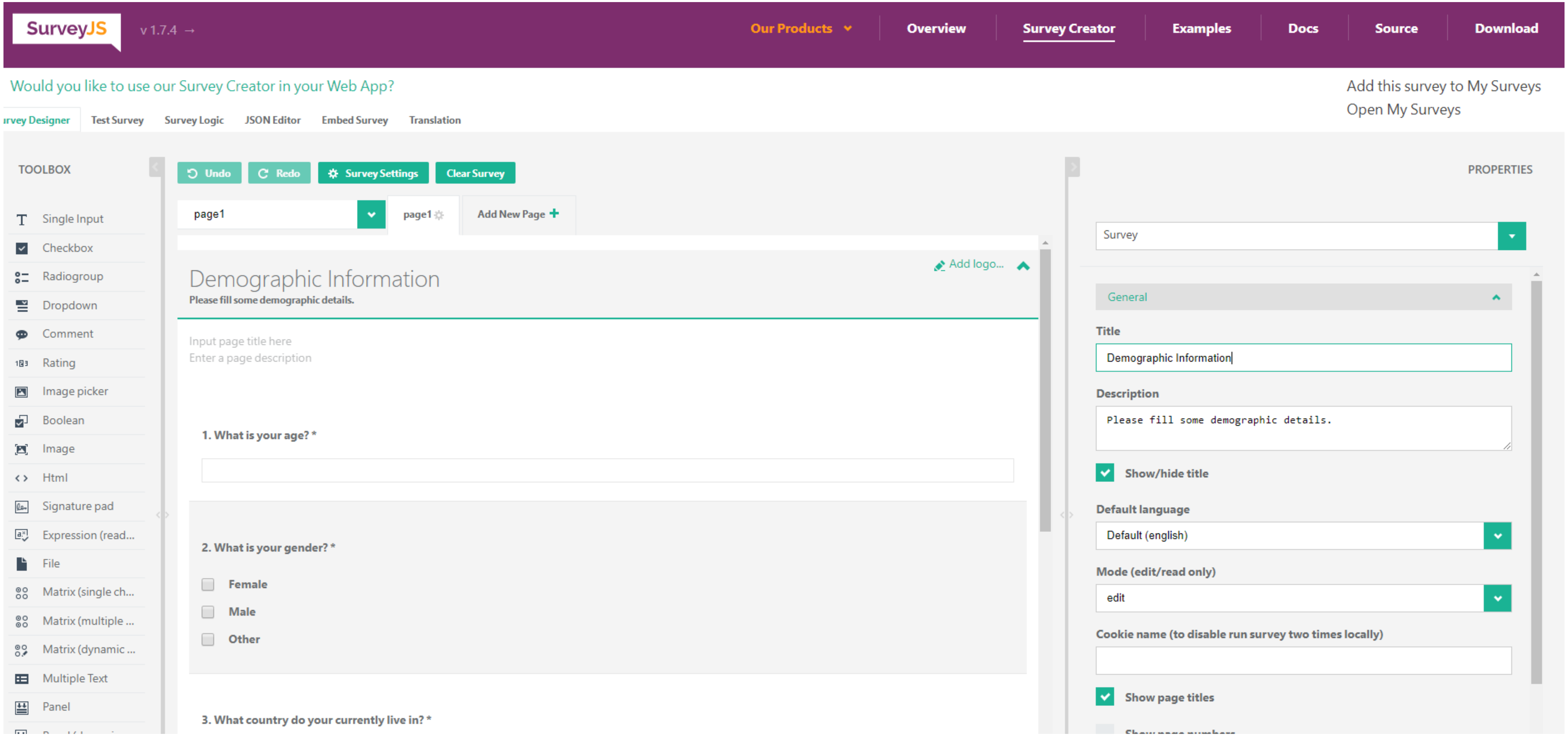


Information Sheet → Consent Form → Demographics → Instructions → Experiment → End

Choice:

X N

Demographics questionnaire – survey.js



Demographic Information

Please fill some demographic details.

Input page title here

Enter a page description

1. What is your age? *

2. What is your gender? *

□ Female

□ Male

□ Other

3. What country do your currently live in? *

Title: Demographic Information

Description: Please fill some demographic details.

☑ Show/hide title

Default language: Default (english)

Mode (edit/read only): edit

Cookie name (to disable run survey two times locally):

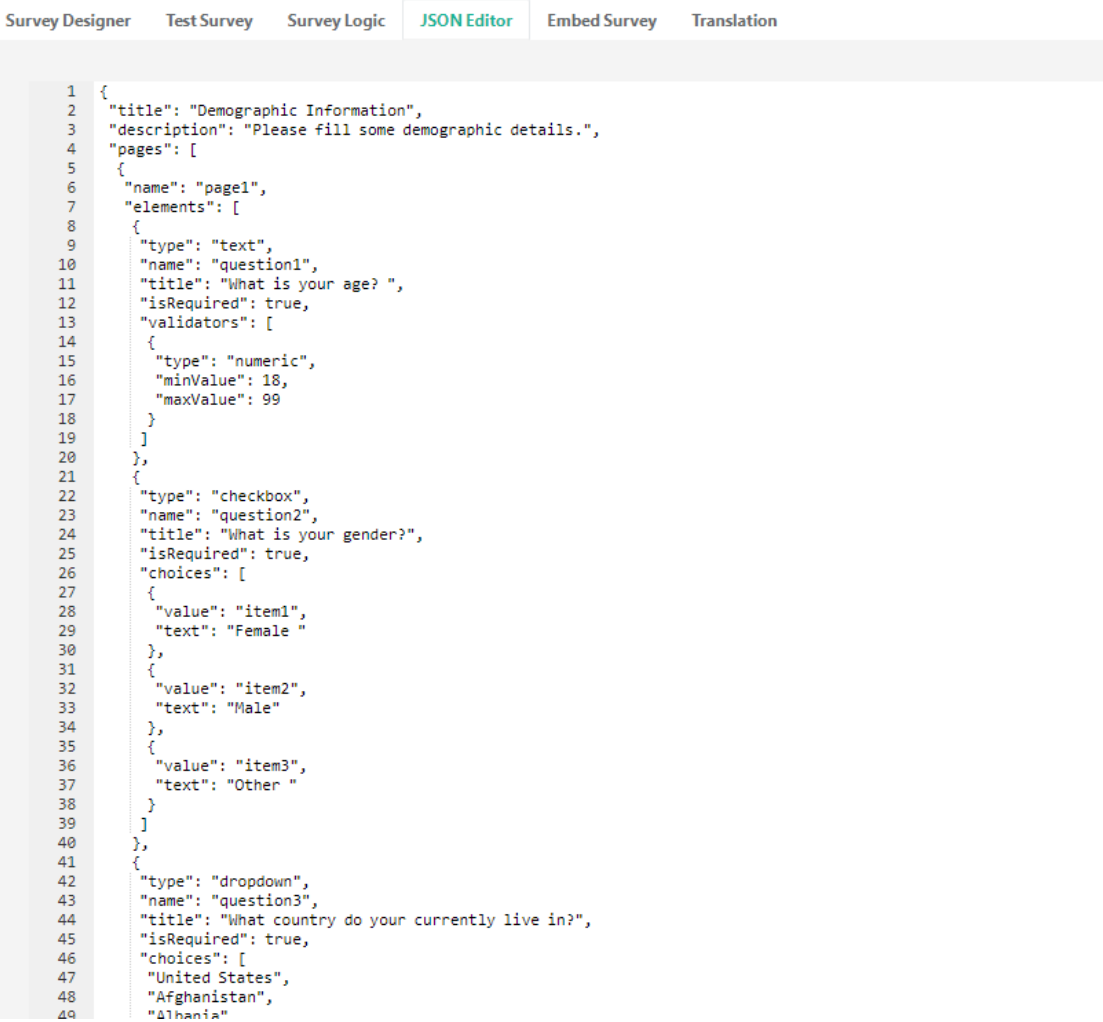
☑ Show page titles

Demographics questionnaire – survey.js

I generated one-page questionnaire in surveyJS, and copied the script from the json editor tab to a DemogJson.json file in the jsons folder.

Json -JavaScript Object Notation – is a standard object that holds data.

There are many tools to open and edit it.



```json
{
 "title": "Demographic Information",
 "description": "Please fill some demographic details.",
 "pages": [
  {
   "name": "page1",
   "elements": [
    {
     "type": "text",
     "name": "question1",
     "title": "What is your age? ",
     "isRequired": true,
     "validators": [
      {
       "type": "numeric",
       "minValue": 18,
       "maxValue": 99
      }
     ]
    },
    {
     "type": "checkbox",
     "name": "question2",
     "title": "What is your gender?",
     "isRequired": true,
     "choices": [
      {
       "value": "item1",
       "text": "Female "
      },
      {
       "value": "item2",
       "text": "Male"
      },
      {
       "value": "item3",
       "text": "Other "
      }
     ]
    },
    {
     "type": "dropdown",
     "name": "question3",
     "title": "What country do your currently live in?",
     "isRequired": true,
     "choices": [
      "United States",
      "Afghanistan",
      "Albania"
```

Demographics questionnaire – survey.js

In order to use the surveyJS tools, I need to add an initialization to my javascript:

```
   // Initiating the survey js module, if you use one
    Survey.StylesManager.applyTheme("bootstrap");
```

And add a link in the head of my html file:

```
<script src="https://surveyjs.azureedge.net/1.5.18/survey.jquery.min.js"></script>
```

Demographics questionnaire – survey.js

Finally, I need to upload my json. I can do it using ajax in the js file, or in a fixed manner in the head of the html file:

In html:

```
<script type="text/javascript" src="jsons/DemogJson.json"></script>
```

Or

In JS:

```
$.ajax({
    'async': false,
    'global': false,
    'url': "jsons/DemogJson.json",
    'dataType': "json",
    'success': function (data) {
        console.log(JsonDetails)
        JsonDetails = data;
    }
});
```

Demographics questionnaire – survey.js

And here is how the demographics presentation function looks like:

```
function SurveyPageDetails(){
      console.log('SurveyDetails');

        $('#Top').css('height', thisHeight / 20);
        $('#Stage').css('min-height', thisHeight * 17 / 20);
       $('#Bottom').css('min-height', thisHeight / 20);

    var JsonDetails =  JSON.parse(JSON.stringify(DemogJson));
       console.log(JsonDetails)

       var survey_details = new Survey.Model(JsonDetails);
   console.log(survey_details)

$("#Stage").Survey({
    model: survey_details,
    onComplete: InsertDemog
});
}
```

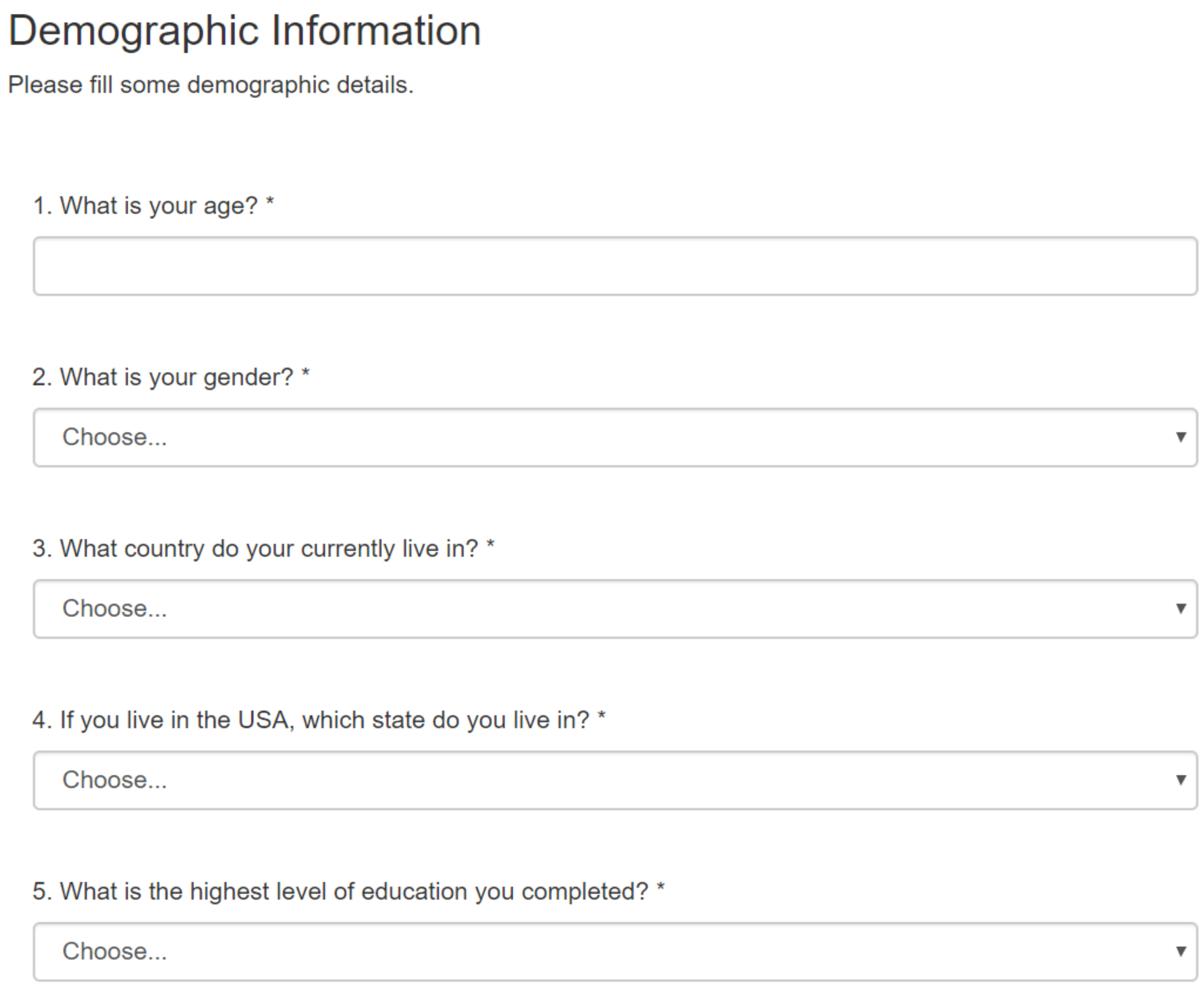
The surveyJS package has a function called onComplete, which tells the survey what to do when the participant is finished – in our case it goes to a function that stores the information - InsertDemog

Demographics questionnaire – survey.js

```
function InsertDemog(survey) {
        $('#TextBoxDiv').remove();
        $('#Stage').empty();

       //send Ajax request to your web server.
var Json1=[survey.data]

var csv = ConvertToCSV_quest(Json1)

console.log("The results are:" +csv)
   Instructions(1);

/* $.ajax({
          type: 'POST',
          data: {ID:SubID,Responses:csv},
          async: false,
          url: 'InsertDemogData.php',
          dataType: 'json',
          success: function(r) {
             if (r[0].ErrorNo > 0) {
                Error();
             } else {
                $('#Stage').empty();
        $('#Bottom').empty();
            Instructions(1);
             };
          }, error: function(XMLHttpRequest, textStatus, errorThrown) {
             alert("Status: " + textStatus);
             alert("Error: " + errorThrown);
          }
        });*/
   }
```

To store information I convert the results to one csv line (comma separated values), and the sql table includes two columns: ID and responses.
I commented the ajax here, but you can test it on your localserver.
Check the ConvertToCSV_quest function in the script.

Demographics questionnaire – survey.js



Demographic Information

Please fill some demographic details.

1. What is your age? *

2. What is your gender? *

Choose... ▾

3. What country do your currently live in? *

Choose... ▾

4. If you live in the USA, which state do you live in? *

Choose... ▾

5. What is the highest level of education you completed? *

Choose... ▾

Experiment – choice screen

```
function Experiment(TrialNum) {


console.log(TrialNum)
        $('#Top').css('height', thisHeight / 20);
        $('#Stage').css('min-height', thisHeight * 17 / 20);
        $('#Bottom').css('min-height', thisHeight / 20);


       var  InitTime = (new Date()).getTime();
```

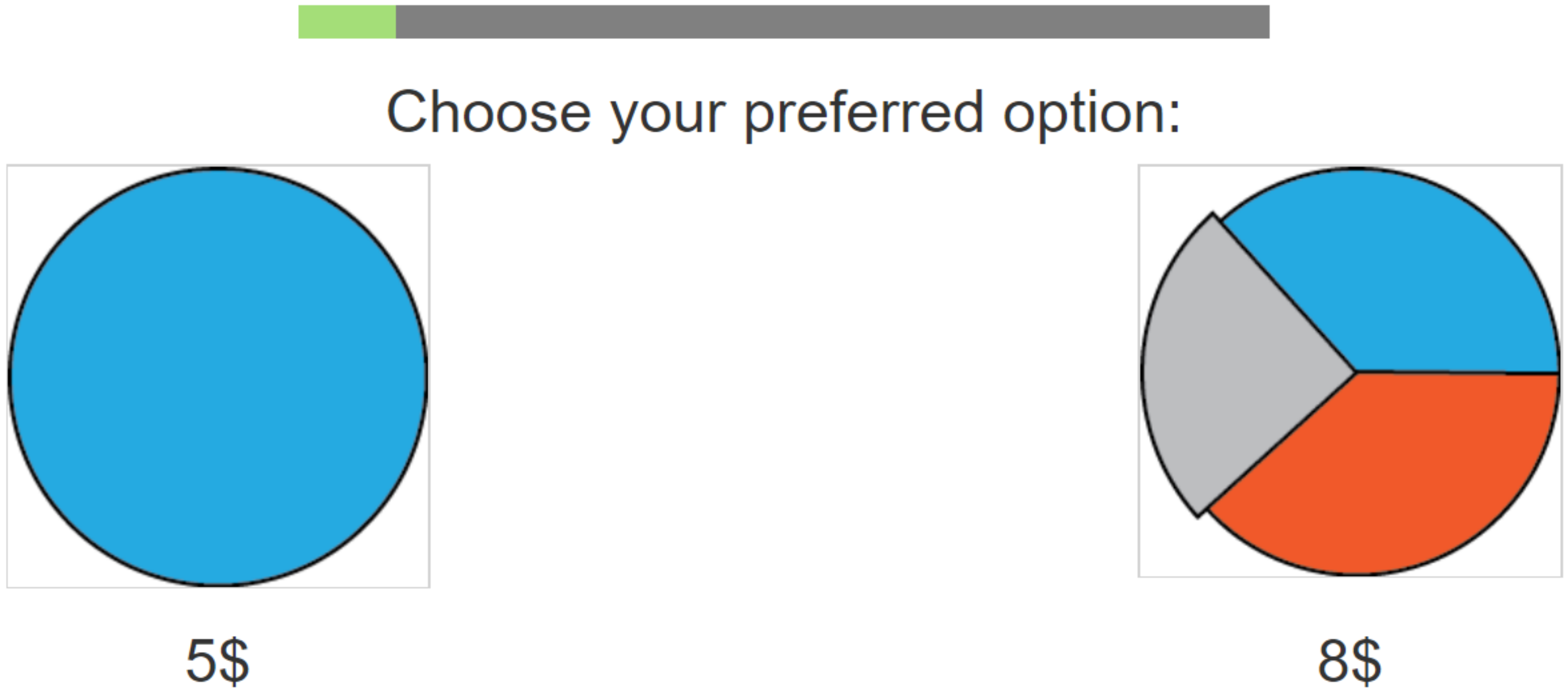
Experiment – choice screen

```
function Experiment(TrialNum) {

console.log(TrialNum)
    $('#Top').css('height', thisHeight / 20);
    $('#Stage').css('min-height', thisHeight * 17 / 20);
    $('#Bottom').css('min-height', thisHeight / 20);

    var  InitTime = (new Date()).getTime();
```

Progress bar – giving our participants some hope…

```
$('#Stage').append('<div class="row"> <div class="col-md-3"></div>  <div id=
"progressBarFrame" class="col-md-6 nopadding"></div> </div>')

     $('#progressBarFrame').css({ "height": thisHeight / 32 + 'px', "background-
color": "grey"});
    $('#progressBarFrame').show();
              var thisWidth = $('#progressBarFrame').width() ;

    CreateDiv('progressBarFrame', 'progressBar');
    $('#progressBar').css({"width": ((TrialNum+1) * thisWidth / (NumTrials)) +
'px', "height": thisHeight / 32 + 'px', "background-color": "#A4DE78"});

    $('#progressBar').show();
```

Progress bar



Choose your preferred option:

5$

8$

Experiment – choice screen

```
     CreateDiv('Stage', 'TextBoxDiv');

     var Title = '<div id = "Title"><H2 align = "center">Choose your preferred option:</H2></div>';
    var DoorName='tmp';
    var Press=0;

   if (AmbiguityTrial[Order[TrialNum]]>0){// ambiguity Trial
                    DoorName='amb_'+AmbiguityTrial[Order[TrialNum]]+'.png';
   }else{//Risk  trial
       DoorName='risk_'+RiskTrial[Order[TrialNum]]+'.png';
   }

       var Door1 = '<img id = "Door1" src="images/sure.png" class="img-responsive center-block
myborder"  >';
       var Door2 = '<img id = "Door2" src="images/'+DoorName+'" class="img-responsive center-block
myborder" >';
       var Sum1 = '5$';
       var Sum2 = SumTrial[Order[TrialNum]]+'$';
```

Experiment – choice screen

```
     CreateDiv('Stage', 'TextBoxDiv');

     var Title = '<div id = "Title"><H2 align = "center">Choose you
    var DoorName='tmp';
    var Press=0;

   if (AmbiguityTrial[Order[TrialNum]]>0){// ambiguity Trial
                   DoorName='amb_'+AmbiguityTrial[Order[TrialNum]]+'.png';
   }else{//Risk  trial
      DoorName='risk_'+RiskTrial[Order[TrialNum]]+'.png';
   }

      var Door1 = '<img id = "Door1" src="images/sure.png" class="img-responsive center-block
myborder"  >';
      var Door2 = '<img id = "Door2" src="images/'+DoorName+'" class="img-responsive center-block
myborder" >';
      var Sum1 = '5$';
      var Sum2 = SumTrial[Order[TrialNum]]+'$';
```

We change the picture of the uncertain option according to the risk and ambiguity characteristics of the trial.
Note that we use the shuffled order vector to set the current trial details.

The sum of money to be won is presented as text under the picture of the lottery.

Experiment – choice screen

```
 var RandPosition = Math.random();
        if (RandPosition < 0.5) {
            var Images = '<div class="row">  <div class="col-md-1"></div>  <div class="col-md-3">' + Door1 + '</div><div id = "Middle" class="col-md-4"></div><div class="col-md-3">' + Door2 + '</div><div class="col-md-1"></div></div>';
             var Sums = '<div class="row">  <div class="col-md-1"></div>  <div class="col-md-3"><h2 align = "center">' + Sum1 + '</h2></div><div id = "Middle" class="col-md-4"></div><div class="col-md-3"><h2 align = "center">' + Sum2 + '</h2></div><div class="col-md-1"></div></div>';
        } else {
            var Images = '<div class="row">  <div class="col-md-1"> </div> <div class="col-md-3">' + Door2 + '</div><div  id = "Middle"  class="col-md-4"></div><div class="col-md-3">' + Door1 + '</div><div class="col-md-1"></div></div>';
             var Sums = '<div class="row">  <div class="col-md-1"></div>  <div class="col-md-3"><h2 align = "center">' + Sum2 + '</h2></div><div id = "Middle" class="col-md-4"></div><div class="col-md-3"><h2 align = "center">' + Sum1 + '</h2></div><div class="col-md-1"></div></div>';
        }

        $('#TextBoxDiv').html(Title + Images+Sums);
```

Experiment – choice screen

```
$('#Door1').click(function() {
        if (Press===0){
           Press=1;
        $(this).css({"border-color": "#CCFF33",
           "border-width": "3px",
           "border-style": "solid"});
    var  ThisTime = (new Date()).getTime();

        InsertData(TrialNum, 1,Sign(RandPosition-0.5),ThisTime-InitTime);
        }
     });
```

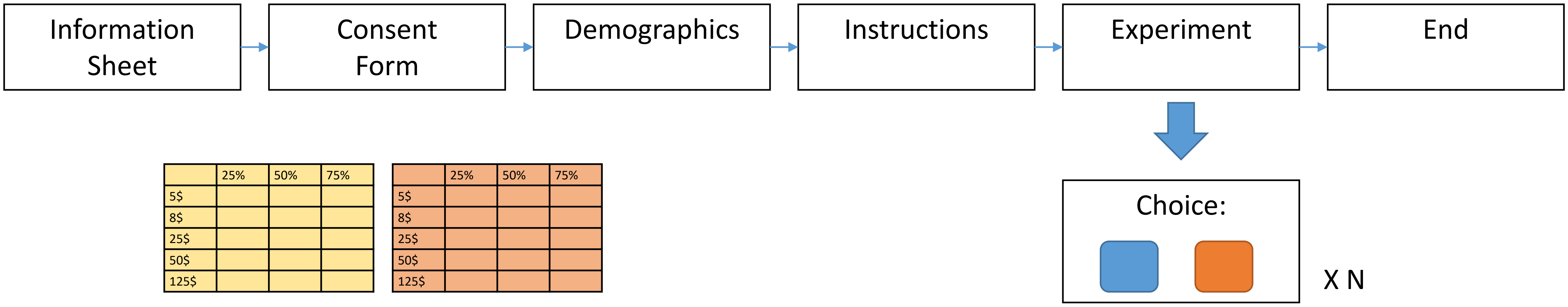
InsertData

```
function InsertData(TrialNum, Choice,Side,RT) {

             if (TrialNum + 1 < NumTrials) {
       //  InsertDataAjax(TrialNum,Choice,Side,RT);
            setTimeout(function() {
                $('#TextBoxDiv').fadeOut(500);
                setTimeout(function() {
                    $('#Stage').empty();
                    $('#Bottom').empty();
                    Experiment(TrialNum + 1);
                }, 750);
            }, 1000);
        } else {
           // InsertDataAjax(TrialNum,Choice,Side,RT);
            setTimeout(function() {
                $('#TextBoxDiv').fadeOut(500);
                setTimeout(function() {
                    $('#Stage').empty();
                    $('#Bottom').empty();
                    End();
                }, 750);
            }, 1000);
        }
}
```

InsertDataAjax

```
function InsertDataAjax(TrialNum,Choice,Side,RT){
     var  ThisTime = (new Date()).getTime();
    console.log(RiskTrial[Order[TrialNum]])

   $.ajax({
        type: 'POST',
        data:
{ID:SubID,TrialNum:TrialNum,Choice:Choice,Side:Side,RT:RT,Amb:AmbiguityTrial[Order[TrialNum]],Risk:RiskTrial
[Order[TrialNum]],Catch:CatchTrial[Order[TrialNum]],Sum:SumTrial[Order[TrialNum]],Time:ThisTime},
        async: false,
        url: 'InsertTrialData.php',
        dataType: 'json',
        success: function(r) {
          if (r[0].ErrorNo > 0) {
            Error();
          }
        }, error: function(XMLHttpRequest, textStatus, errorThrown) {
          alert("Status: " + textStatus);
          alert("Error: " + errorThrown);
        }
      });
  }
```

Summary



	25%	50%	75%
5$			
8$			
25$			
50$			
125$			

	25%	50%	75%
5$			
8$			
25$			
50$			
125$			

We added two main elements in this example.

1. Questionnaire
2. Randomized multi-level experimental design (card-deck)

You can use these elements as is, or expand on them – adding questionnaires and experimental conditions suited to your own experiments.

Future

There are many other tools you can use for your experiments, many are available online for free.

Sliding scales, drawing with your mouse, tracking the mouse, tracking eye movement (using the webcam), keyboard control, force full screen, multiplayer games, game engines, canvas, animation, sounds…

Be creative, think about your design and its goal, and make use of the variety of tools available.