**Instructions:**

There are several steps that have to be taken to implement the sociogram.

**First Step:** Copy the questions into your questionnaire

* Press "Add a block", go to "Import Questions", navigate to the sociogram template, and import the first block of instructions
* Press "Add a block" again, go to "Import Questions", navigate to the sociogram template, and import the second part of the sociogram template (3 questions with the timer, instructions, and the canvas)
* Press "Add a block" again, go to "Import Questions", navigate to the sociogram template, and import the fourth part of the sociogram template (1 Question, with helptext). **This block has to be called "helptext"!**
* Move the instructions block, as well as the drawing block, to the location you would like it in the questionnaire. You can leave the help text

**Second Step:** Adjust the look and feel section of the questionnaire

* Open the "Look & Feel" section of the questionnaire
* In the "General" section, copy the following text into the "header" field:

*<link rel="stylesheet" href="https://ajax.googleapis.com/ajax/libs/jqueryui/1.12.1/themes/smoothness/jquery-ui.css"><script src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.12.1/jquery-ui.min.js"></script>*

* In the "Style" section, copy the following style sheet definitions into the "Custom CSS" field (add any pre-existing CSS definitions at the end):

#saveme,#ellipse,#draw,#label,#undo{

width: 100px;

padding: 5px;

text-align: center;

background: blue;

color: white;

border: 2px solid black;

-webkit-border-radius: 4px 4px 4px 4px;

border-radius: 4px 4px 4px 4px;

}

#saveme:hover,#ellipse:hover,#draw:hover,#label:hover,#undo:hover{

cursor: pointer;

background: red;

}

#options,

#main{

float: left;

height: 820px;

}

#ellipseopt, #lineopt{

position: relative;

left: auto !important;

opacity: 1 !important;

z-index: 1000 !important;

height: auto !important;

width: auto !important;

}

#options{

width: 100%;

}

#timeWarning{

color: red;

font-weight: bold;

text-align: center;

}

#draw\_options,

#undo{

float: left;

}

#undo{

background: #888;

}

#saveme{

float: right;

}

#main{

position: relative;

}

#options div{

display: inline;

margin-right: 10px;

}

body .PageTimer{

padding: 0px 10px 0px !important;

float: right;

}

canvas {position:absolute;left:0;top:50px; border: 1px solid black}

#canvas {background:#eef}

**Third Step:** Adjust the drawing block of the sociogram

* Click on the text in the second question, open the HTML view, and copy the following HTML code in there:

<span style="font-size:13px;"><span style="font-family:Arial,Helvetica,sans-serif;"><b>Please draw your sociogram in the box below.&nbsp;&nbsp;</b></span></span> <img id="helpme" height="16" src="https://kentpsych.eu.qualtrics.com/CP/Graphic.php?IM=IM\_0eaeWelDvdFTiFD" style="width: 16px; height: 16px;" width="16">

* Click on the text in the third question, open the HTML view, and copy the following HTML code in there:

<style type="text/css">.Skin .QuestionOuter {padding-bottom: 0px !important;}

</style>

<div id="main"><canvas id="undocanvas" width="700px" height="600px"></canvas> <canvas id="mycanvas" width="700px" height="600px"></canvas> <canvas tabindex="0" id="tempcanvas" width="700px" height="600px"></canvas></div>

<div style="display:none;" id="timeWarning">${e://Field/time\_out\_text}</div>

<div id="options">

<div id="draw\_options"><input value="ellipse" type="radio" name="drawopt" id="ellipseopt" checked="checked"> ${e://Field/ellipse\_text} <input value="line" type="radio" name="drawopt" id="lineopt"> ${e://Field/line\_text}</div>

<div id="undo">${e://Field/undo\_text}</div>

<div id="saveme">${e://Field/save\_text}</div>

</div>

<div title="Help" id="dialog-one">${e://Field/helptext}</div>

* Click on the "JS" symbol next to the third question, and add the Javascript code from the sociogram template in there:

var saveTextFromArea;

var saveTimeOut;

var saveAll;

var x1;

var x2;

var y1;

var y2

var firstX;

var firstY;

var g\_self\_idx = 0;

var undo\_state = -1;

var undo\_mode = "ellipse"; //Keep track of what the undo state was in ... so we can remove line/ellipse data

var g\_all\_shapes = null;

var g\_num = -1;

var timeMS = parseInt('${e://Field/time\_milliseconds}');

var timeRemind = timeMS - 10000; //Show that only 10 seconds remaining!

// JA summary data classes start

class MyCanvasShapes {

constructor() {

this.all\_ellipses = [];

}

drawNewEllipse(x1, y1, x2, y2 ) {

var el;

g\_num = this.all\_ellipses.length;

el = new newEllipse(x1+0.5\*(x2-x1), y1+0.5\*(y2-y1), 0.5\*(x2-x1), 0.5\*(y2-y1), g\_num);

this.all\_ellipses[g\_num] = el;

}

ellipse\_to\_string(el) {

var st="";

st += el.idx.toString() + ", ";

st += el.self\_ellipse.toString() + ", ";

st += el.label + ", ";

st += el.x.toString() + ", ";

st += el.y.toString() + ", ";

st += (2\*el.radX).toString() + ", ";

st += (2\*el.radY).toString() + ", ";

st += el.area.toFixed(2).toString() + ", ";

st += el.area\_pcent.toFixed(2).toString() + "$$$";

return st;

}

setEllipseSelf(idx, flag) {

this.all\_ellipses[idx].setEllipseSelf(flag);

}

results() {

var data\_string = "el\_id, self\_ellipse, lbl, x\_centre, y\_centre, width, height, area, area\_ratio%$$$";

for ( let e = 0; e < this.all\_ellipses.length; e++) {

this.all\_ellipses[e].setAreaPcent(this.all\_ellipses[g\_self\_idx]);

data\_string = data\_string + this.ellipse\_to\_string(this.all\_ellipses[e]);

}

return data\_string;

}

}

class newEllipse {

constructor(x, y, radX, radY, idx) {

this.x = x;

this.y = y;

this.radX = radX;

this.radY = radY;

this.lineColor = 'black';

this.area\_pcent = 100.0;

this.label = "";

this.area = 0;

this.idx = idx;

this.self\_ellipse = false;

this.setEllipseArea();

}

setEllipseSelf(flag) {

if ( flag ) {

this.self\_ellipse = true;

} else {

this.self\_ellipse = false;

}

}

setEllipseLabel(label) {

this.label = label;

}

setEllipseArea() {

this.area = this.radX \* this.radY \* Math.PI;

}

setAreaPcent(el\_main) {

this.area\_pcent = (100.0 \* this.area) / el\_main.area;

}

}

// JA summary data classes end

Qualtrics.SurveyEngine.addOnload(function()

{

jQuery("#Buttons").hide();

jQuery('#saveme').hide();

jQuery("#dialog-one").hide();

});

Qualtrics.SurveyEngine.addOnReady(function()

{

var mode = "ellipse";

var canvas = jQuery('#mycanvas')[0];

var mainctx = canvas.getContext('2d');

mainctx.fillStyle = "white";

mainctx.fillRect(0, 0, canvas.width, canvas.height);

var ctx = tempcanvas.getContext('2d');

var undoctx = undocanvas.getContext('2d');

var isDown = false;

var w = canvas.width;

var h = canvas.height;

var mouse = {x: 0, y: 0};

g\_all\_shapes = new MyCanvasShapes(); // JA create container for results summary data

jQuery("#helpme").click(function(e) {

jQuery( "#dialog-one" ).dialog({

resizable: false,

draggable: false,

position: { my: "center", at: "top" },

height: "auto",

width: 570,

buttons: {

"Close": function() {

jQuery( this ).dialog( "close" );

},

},

modal: true

});

});

jQuery("#undo").click(function(e) {

//Draw in the old undocanvas onto the main

mainctx.drawImage(undocanvas, 0, 0);

//Hide the undo option right away

jQuery("#undo").hide();

//Kill any text box that may still be active.

jQuery('#textareaTest').remove();

jQuery('#saveText').remove();

//If ellipse was checked, put back into ellipse mode

if (mode == "label")

{

mode = "ellipse";

}

//If the last item was an ellipse, and we now click undo ... need to remove from the list (last element).

if (undo\_mode == "ellipse")

{

g\_all\_shapes.all\_ellipses.pop();

if (g\_all\_shapes.all\_ellipses.length < 2)

{

jQuery('#saveme').hide();

}

}

});

//Hide the undo option right away

jQuery("#undo").hide();

jQuery("#tempcanvas").mousedown(function(e) {

jQuery('canvas').focus();

if (mode == "ellipse")

{

var rect = canvas.getBoundingClientRect();

x1 = e.clientX - rect.left;

y1 = e.clientY - rect.top;

//ADDS

if (isDown == false)

{

firstX = x1;

firstY = y1;

x2 = x1;

y2 = y1;

}

isDown = true;

}

if (mode == "line")

{

if (e.layerX || e.layerX == 0) { // Firefox

e.\_x = e.layerX;

e.\_y = e.layerY;

} else if (e.offsetX || e.offsetX == 0) { // Opera

e.\_x = e.offsetX;

e.\_y = e.offsetY;

}

isDown = true;

firstX = e.\_x;

firstY = e.\_y;

}

return true;

});

jQuery("#tempcanvas").mouseup(function(e) {

if (!isDown) return;

if (mode == "ellipse")

{

// avoid mistaken 'null' ellipses

if ( (x2 == null) || (y2 == null)) {

ctx.clearRect(0, 0, w, h);

isDown = false;

return;

}

//Ensure elipse is a reasonable scale

if (Math.abs(firstX - x2) < 50 ){

ctx.clearRect(0, 0, w, h);

isDown = false;

return;

}

if (Math.abs(firstY - y2) < 30 ){

ctx.clearRect(0, 0, w, h);

isDown = false;

return;

}

//Make sure X and Y is not outside of canvas

if (firstX == null || firstY == null || x2 == null || y2 == null || firstX < 0 || firstY < 0 || x2 < 0 || y2 < 0 || firstX > canvas.width || x2 > canvas.width || firstY > canvas.height || y2 > canvas.height)

{

ctx.clearRect(0, 0, w, h);

isDown = false;

return;

}

//If we are good and have drawn a valid ellipse, save the canvas for undo before we update the current canvas

undoctx.drawImage(mycanvas, 0, 0);

isDown = false;

mainctx.drawImage(tempcanvas, 0, 0);

ctx.clearRect(0, 0, w, h);

//Switch to label mode!

mode = "label";

g\_all\_shapes.drawNewEllipse(firstX, firstY, x2, y2);

g\_all\_shapes.setEllipseSelf(g\_num, false);

if (x2 < firstX) {

temp = x2;

x2 = firstX;

firstX = temp;

}

if (y2 < firstY) {

temp = y2;

y2 = firstY;

firstY = temp;

}

box\_x = firstX + 0.3 \* (x2-firstX); //+ 0.3 \* (x2 - firstX);

box\_y = firstY + 0.5 \* (y2 - firstY); // + 0.25 \* (y2 - firstY);

if (jQuery('#textAreaPopUp').length == 0) {

//Hide the undo option right away at point of text entry

jQuery("#undo").hide();

var rect = canvas.getBoundingClientRect();

var mouseX = e.clientX - rect.left;

var mouseY = e.clientY - rect.top;

//append a text area box to the canvas where the user clicked to enter in a comment

var textArea = "<div id='textAreaPopUp' style='position:absolute;top:"+box\_y+"px;left:"+box\_x+"px;z-index:30;'><input type='text' maxlength='4' id='textareaTest' style='width:50px;'>";

var saveButton = "<input type='button' value='${e://Field/savetextbox\_text}' id='saveText' onclick='saveTextFromArea();'></div>";

var appendString = textArea + saveButton;

jQuery("#main").append(appendString);

jQuery('#textAreaPopUp').on( 'keydown', function( e ) {

// jQuery normalizes the "e" parameter, so you can use:

if ( e.keyCode === 13 ) {

// Do something

saveTextFromArea();

}

} );

} else {

jQuery('#textareaTest').remove();

jQuery('#saveText').remove();

jQuery('#textAreaPopUp').remove();

var rect = canvas.getBoundingClientRect();

var mouseX = e.clientX - rect.left;

var mouseY = e.clientY - rect.top;

//append a text area box to the canvas where the user clicked to enter in a comment

var textArea = "<div id='textAreaPopUp' style='position:absolute;top:"+box\_y+"px;left:"+box\_x+"px;z-index:30;'><input type='text' maxlength='4' id='textareaTest' style='width:50px;'>";

var saveButton = "<input type='button' value='${e://Field/savetextbox\_text}' id='saveText' onclick='saveTextFromArea();'></div>";

var appendString = textArea + saveButton;

jQuery("#main").append(appendString);

jQuery('#textAreaPopUp').on( 'keydown', function( e ) {

// jQuery normalizes the "e" parameter, so you can use:

if ( e.keyCode === 13 ) {

// Do something

saveTextFromArea();

}

} );

}

jQuery('#textAreaPopUp #textareaTest').focus(); // move focus to the text box so we can start typing immediately

}

if (mode == "line")

{

if (e.layerX || e.layerX == 0) { // Firefox

e.\_x = e.layerX;

e.\_y = e.layerY;

} else if (e.offsetX || e.offsetX == 0) { // Opera

e.\_x = e.offsetX;

e.\_y = e.offsetY;

}

if (isDown) {

isDown = false;

//Copy the canvas now for undo... one step behind

undoctx.drawImage(mycanvas, 0, 0);

//Now reshow the undo option

jQuery("#undo").show();

//Update the canvas

mainctx.drawImage(tempcanvas, 0, 0);

ctx.clearRect(0, 0, w, h);

undo\_mode = "line";

}

//Show again once done

jQuery("#undo").show();

}

return true;

});

jQuery("#tempcanvas").mousemove(function(e) {

//As we can move out of canvas, we need to check if the mouse buttons are pressed still to update state.

if(e.which !== 1)

{

isDown = false;

ctx.clearRect(0, 0, w, h);

return false;

}

if (mode == "ellipse")

{

if (!isDown) return;

var rect = canvas.getBoundingClientRect();

x2 = e.clientX - rect.left;

y2 = e.clientY - rect.top;

ctx.clearRect(0, 0, w, h);

drawEllipse(x1, y1, x2, y2);

}

if (mode == "line")

{

if (!isDown) return;

if (e.layerX || e.layerX == 0) { // Firefox

e.\_x = e.layerX;

e.\_y = e.layerY;

} else if (e.offsetX || e.offsetX == 0) { // Opera

e.\_x = e.offsetX;

e.\_y = e.offsetY;

}

if (isDown) {

ctx.clearRect(0, 0, canvas.width, canvas.height);

ctx.beginPath();

ctx.moveTo(firstX, firstY);

ctx.lineTo(e.\_x, e.\_y);

ctx.stroke();

ctx.closePath();

}

}

return true;

});

function drawEllipse(x1, y1, x2, y2) {

var radiusX = (x2 - x1) \* 0.5,

radiusY = (y2 - y1) \* 0.5,

centerX = x1 + radiusX,

centerY = y1 + radiusY,

step = 0.01,

a = step,

pi2 = Math.PI \* 2 - step;

ctx.beginPath();

ctx.moveTo(centerX + radiusX \* Math.cos(0),

centerY + radiusY \* Math.sin(0));

for(; a < pi2; a += step) {

ctx.lineTo(centerX + radiusX \* Math.cos(a),

centerY + radiusY \* Math.sin(a));

}

ctx.closePath();

ctx.strokeStyle = '#000';

ctx.stroke();

}

saveTextFromArea = function(){

var text = jQuery('#textareaTest').val().trim();

//Do not allow empty entries

if (text.trim().length < 1)

{

return null;

}

//Work out x and y

var x = firstX + (0.05 \* (x2 - firstX));

var y = firstY + (0.5 \* (y2 - firstY)) +15;

//Work out central pos of ellipse

var centerX = firstX + ((x2-firstX)/2);

var centerY = firstY + ((y2-firstY)/2);

//get the value of the textarea then destroy it and the save button

jQuery('#textareaTest').remove();

jQuery('#saveText').remove();

//get the canvas and add the text functions

var cw = canvas.clientWidth;

var ch = canvas.clientHeight;

//break the text into arrays based on a text width of 100px

var phraseArray = getLines(mainctx,text,100);

// this adds the text functions to the ctx

CanvasTextFunctions.enable(mainctx);

var counter = 0;

//draw each phrase to the screen, making the top position 20px more each time so it appears there are line breaks

jQuery.each(phraseArray, function() {

//set the placement in the canvas

text = this;

//draw the text

mainctx.textBaseline = "middle";

mainctx.textAlign = "center";

mainctx.fillStyle = "#000000";

mainctx.font = "bold 16px verdana, sans-serif";

mainctx.fillText(text, centerX, centerY);

mainctx.save();

mainctx.restore();

});

//reset the drop shadow so any other drawing don't have them

mainctx.shadowOffsetX = 0;

mainctx.shadowOffsetY = 0;

mainctx.shadowBlur = 0;

mainctx.shadowColor = "rgba(0,0,0,0)";

// save the ellipse label/name into results data structure

g\_all\_shapes.all\_ellipses[g\_num].setEllipseLabel(text);

//Reset the firstX and firstY

firstX = null;

firstY = null;

mode = "ellipse";

undo\_mode = "ellipse";

//Now reshow the undo option

jQuery("#undo").show();

//console.log(g\_all\_shapes.all\_ellipses);

if (g\_all\_shapes.all\_ellipses.length >= 2)

{

jQuery('#saveme').show();

}

}

function getLines(ctx,phrase,maxPxLength) {

//break the text area text into lines based on "box" width

var wa=phrase.split(" "),

phraseArray=[],

lastPhrase="",

l=maxPxLength,

measure=0;

ctx.font = "16px sans-serif";

for (var i=0;i<wa.length;i++) {

var w=wa[i];

measure=ctx.measureText(lastPhrase+w).width;

if (measure<l) {

lastPhrase+=(" "+w);

}else {

phraseArray.push(lastPhrase);

lastPhrase=w;

}

if (i===wa.length-1) {

phraseArray.push(lastPhrase);

break;

}

}

return phraseArray;

}

var CanvasTextFunctions = { };

CanvasTextFunctions.letters = {

' ': { width: 16, points: [] },

'!': { width: 10, points: [[5,21],[5,7],[-1,-1],[5,2],[4,1],[5,0],[6,1],[5,2]] },

'"': { width: 16, points: [[4,21],[4,14],[-1,-1],[12,21],[12,14]] },

'#': { width: 21, points: [[11,25],[4,-7],[-1,-1],[17,25],[10,-7],[-1,-1],[4,12],[18,12],[-1,-1],[3,6],[17,6]] },

'$': { width: 20, points: [[8,25],[8,-4],[-1,-1],[12,25],[12,-4],[-1,-1],[17,18],[15,20],[12,21],[8,21],[5,20],[3,18],[3,16],[4,14],[5,13],[7,12],[13,10],[15,9],[16,8],[17,6],[17,3],[15,1],[12,0],[8,0],[5,1],[3,3]] },

'%': { width: 24, points: [[21,21],[3,0],[-1,-1],[8,21],[10,19],[10,17],[9,15],[7,14],[5,14],[3,16],[3,18],[4,20],[6,21],[8,21],[10,20],[13,19],[16,19],[19,20],[21,21],[-1,-1],[17,7],[15,6],[14,4],[14,2],[16,0],[18,0],[20,1],[21,3],[21,5],[19,7],[17,7]] },

'&': { width: 26, points: [[23,12],[23,13],[22,14],[21,14],[20,13],[19,11],[17,6],[15,3],[13,1],[11,0],[7,0],[5,1],[4,2],[3,4],[3,6],[4,8],[5,9],[12,13],[13,14],[14,16],[14,18],[13,20],[11,21],[9,20],[8,18],[8,16],[9,13],[11,10],[16,3],[18,1],[20,0],[22,0],[23,1],[23,2]] },

'\'': { width: 10, points: [[5,19],[4,20],[5,21],[6,20],[6,18],[5,16],[4,15]] },

'(': { width: 14, points: [[11,25],[9,23],[7,20],[5,16],[4,11],[4,7],[5,2],[7,-2],[9,-5],[11,-7]] },

')': { width: 14, points: [[3,25],[5,23],[7,20],[9,16],[10,11],[10,7],[9,2],[7,-2],[5,-5],[3,-7]] },

'\*': { width: 16, points: [[8,21],[8,9],[-1,-1],[3,18],[13,12],[-1,-1],[13,18],[3,12]] },

'+': { width: 26, points: [[13,18],[13,0],[-1,-1],[4,9],[22,9]] },

',': { width: 10, points: [[6,1],[5,0],[4,1],[5,2],[6,1],[6,-1],[5,-3],[4,-4]] },

'-': { width: 26, points: [[4,9],[22,9]] },

'.': { width: 10, points: [[5,2],[4,1],[5,0],[6,1],[5,2]] },

'/': { width: 22, points: [[20,25],[2,-7]] },

'0': { width: 20, points: [[9,21],[6,20],[4,17],[3,12],[3,9],[4,4],[6,1],[9,0],[11,0],[14,1],[16,4],[17,9],[17,12],[16,17],[14,20],[11,21],[9,21]] },

'1': { width: 20, points: [[6,17],[8,18],[11,21],[11,0]] },

'2': { width: 20, points: [[4,16],[4,17],[5,19],[6,20],[8,21],[12,21],[14,20],[15,19],[16,17],[16,15],[15,13],[13,10],[3,0],[17,0]] },

'3': { width: 20, points: [[5,21],[16,21],[10,13],[13,13],[15,12],[16,11],[17,8],[17,6],[16,3],[14,1],[11,0],[8,0],[5,1],[4,2],[3,4]] },

'4': { width: 20, points: [[13,21],[3,7],[18,7],[-1,-1],[13,21],[13,0]] },

'5': { width: 20, points: [[15,21],[5,21],[4,12],[5,13],[8,14],[11,14],[14,13],[16,11],[17,8],[17,6],[16,3],[14,1],[11,0],[8,0],[5,1],[4,2],[3,4]] },

'6': { width: 20, points: [[16,18],[15,20],[12,21],[10,21],[7,20],[5,17],[4,12],[4,7],[5,3],[7,1],[10,0],[11,0],[14,1],[16,3],[17,6],[17,7],[16,10],[14,12],[11,13],[10,13],[7,12],[5,10],[4,7]] },

'7': { width: 20, points: [[17,21],[7,0],[-1,-1],[3,21],[17,21]] },

'8': { width: 20, points: [[8,21],[5,20],[4,18],[4,16],[5,14],[7,13],[11,12],[14,11],[16,9],[17,7],[17,4],[16,2],[15,1],[12,0],[8,0],[5,1],[4,2],[3,4],[3,7],[4,9],[6,11],[9,12],[13,13],[15,14],[16,16],[16,18],[15,20],[12,21],[8,21]] },

'9': { width: 20, points: [[16,14],[15,11],[13,9],[10,8],[9,8],[6,9],[4,11],[3,14],[3,15],[4,18],[6,20],[9,21],[10,21],[13,20],[15,18],[16,14],[16,9],[15,4],[13,1],[10,0],[8,0],[5,1],[4,3]] },

':': { width: 10, points: [[5,14],[4,13],[5,12],[6,13],[5,14],[-1,-1],[5,2],[4,1],[5,0],[6,1],[5,2]] },

',': { width: 10, points: [[5,14],[4,13],[5,12],[6,13],[5,14],[-1,-1],[6,1],[5,0],[4,1],[5,2],[6,1],[6,-1],[5,-3],[4,-4]] },

'<': { width: 24, points: [[20,18],[4,9],[20,0]] },

'=': { width: 26, points: [[4,12],[22,12],[-1,-1],[4,6],[22,6]] },

'>': { width: 24, points: [[4,18],[20,9],[4,0]] },

'?': { width: 18, points: [[3,16],[3,17],[4,19],[5,20],[7,21],[11,21],[13,20],[14,19],[15,17],[15,15],[14,13],[13,12],[9,10],[9,7],[-1,-1],[9,2],[8,1],[9,0],[10,1],[9,2]] },

'@': { width: 27, points: [[18,13],[17,15],[15,16],[12,16],[10,15],[9,14],[8,11],[8,8],[9,6],[11,5],[14,5],[16,6],[17,8],[-1,-1],[12,16],[10,14],[9,11],[9,8],[10,6],[11,5],[-1,-1],[18,16],[17,8],[17,6],[19,5],[21,5],[23,7],[24,10],[24,12],[23,15],[22,17],[20,19],[18,20],[15,21],[12,21],[9,20],[7,19],[5,17],[4,15],[3,12],[3,9],[4,6],[5,4],[7,2],[9,1],[12,0],[15,0],[18,1],[20,2],[21,3],[-1,-1],[19,16],[18,8],[18,6],[19,5]] },

'A': { width: 18, points: [[9,21],[1,0],[-1,-1],[9,21],[17,0],[-1,-1],[4,7],[14,7]] },

'B': { width: 21, points: [[4,21],[4,0],[-1,-1],[4,21],[13,21],[16,20],[17,19],[18,17],[18,15],[17,13],[16,12],[13,11],[-1,-1],[4,11],[13,11],[16,10],[17,9],[18,7],[18,4],[17,2],[16,1],[13,0],[4,0]] },

'C': { width: 21, points: [[18,16],[17,18],[15,20],[13,21],[9,21],[7,20],[5,18],[4,16],[3,13],[3,8],[4,5],[5,3],[7,1],[9,0],[13,0],[15,1],[17,3],[18,5]] },

'D': { width: 21, points: [[4,21],[4,0],[-1,-1],[4,21],[11,21],[14,20],[16,18],[17,16],[18,13],[18,8],[17,5],[16,3],[14,1],[11,0],[4,0]] },

'E': { width: 19, points: [[4,21],[4,0],[-1,-1],[4,21],[17,21],[-1,-1],[4,11],[12,11],[-1,-1],[4,0],[17,0]] },

'F': { width: 18, points: [[4,21],[4,0],[-1,-1],[4,21],[17,21],[-1,-1],[4,11],[12,11]] },

'G': { width: 21, points: [[18,16],[17,18],[15,20],[13,21],[9,21],[7,20],[5,18],[4,16],[3,13],[3,8],[4,5],[5,3],[7,1],[9,0],[13,0],[15,1],[17,3],[18,5],[18,8],[-1,-1],[13,8],[18,8]] },

'H': { width: 22, points: [[4,21],[4,0],[-1,-1],[18,21],[18,0],[-1,-1],[4,11],[18,11]] },

'I': { width: 8, points: [[4,21],[4,0]] },

'J': { width: 16, points: [[12,21],[12,5],[11,2],[10,1],[8,0],[6,0],[4,1],[3,2],[2,5],[2,7]] },

'K': { width: 21, points: [[4,21],[4,0],[-1,-1],[18,21],[4,7],[-1,-1],[9,12],[18,0]] },

'L': { width: 17, points: [[4,21],[4,0],[-1,-1],[4,0],[16,0]] },

'M': { width: 24, points: [[4,21],[4,0],[-1,-1],[4,21],[12,0],[-1,-1],[20,21],[12,0],[-1,-1],[20,21],[20,0]] },

'N': { width: 22, points: [[4,21],[4,0],[-1,-1],[4,21],[18,0],[-1,-1],[18,21],[18,0]] },

'O': { width: 22, points: [[9,21],[7,20],[5,18],[4,16],[3,13],[3,8],[4,5],[5,3],[7,1],[9,0],[13,0],[15,1],[17,3],[18,5],[19,8],[19,13],[18,16],[17,18],[15,20],[13,21],[9,21]] },

'P': { width: 21, points: [[4,21],[4,0],[-1,-1],[4,21],[13,21],[16,20],[17,19],[18,17],[18,14],[17,12],[16,11],[13,10],[4,10]] },

'Q': { width: 22, points: [[9,21],[7,20],[5,18],[4,16],[3,13],[3,8],[4,5],[5,3],[7,1],[9,0],[13,0],[15,1],[17,3],[18,5],[19,8],[19,13],[18,16],[17,18],[15,20],[13,21],[9,21],[-1,-1],[12,4],[18,-2]] },

'R': { width: 21, points: [[4,21],[4,0],[-1,-1],[4,21],[13,21],[16,20],[17,19],[18,17],[18,15],[17,13],[16,12],[13,11],[4,11],[-1,-1],[11,11],[18,0]] },

'S': { width: 20, points: [[17,18],[15,20],[12,21],[8,21],[5,20],[3,18],[3,16],[4,14],[5,13],[7,12],[13,10],[15,9],[16,8],[17,6],[17,3],[15,1],[12,0],[8,0],[5,1],[3,3]] },

'T': { width: 16, points: [[8,21],[8,0],[-1,-1],[1,21],[15,21]] },

'U': { width: 22, points: [[4,21],[4,6],[5,3],[7,1],[10,0],[12,0],[15,1],[17,3],[18,6],[18,21]] },

'V': { width: 18, points: [[1,21],[9,0],[-1,-1],[17,21],[9,0]] },

'W': { width: 24, points: [[2,21],[7,0],[-1,-1],[12,21],[7,0],[-1,-1],[12,21],[17,0],[-1,-1],[22,21],[17,0]] },

'X': { width: 20, points: [[3,21],[17,0],[-1,-1],[17,21],[3,0]] },

'Y': { width: 18, points: [[1,21],[9,11],[9,0],[-1,-1],[17,21],[9,11]] },

'Z': { width: 20, points: [[17,21],[3,0],[-1,-1],[3,21],[17,21],[-1,-1],[3,0],[17,0]] },

'[': { width: 14, points: [[4,25],[4,-7],[-1,-1],[5,25],[5,-7],[-1,-1],[4,25],[11,25],[-1,-1],[4,-7],[11,-7]] },

'\\': { width: 14, points: [[0,21],[14,-3]] },

']': { width: 14, points: [[9,25],[9,-7],[-1,-1],[10,25],[10,-7],[-1,-1],[3,25],[10,25],[-1,-1],[3,-7],[10,-7]] },

'^': { width: 16, points: [[6,15],[8,18],[10,15],[-1,-1],[3,12],[8,17],[13,12],[-1,-1],[8,17],[8,0]] },

'\_': { width: 16, points: [[0,-2],[16,-2]] },

'`': { width: 10, points: [[6,21],[5,20],[4,18],[4,16],[5,15],[6,16],[5,17]] },

'a': { width: 19, points: [[15,14],[15,0],[-1,-1],[15,11],[13,13],[11,14],[8,14],[6,13],[4,11],[3,8],[3,6],[4,3],[6,1],[8,0],[11,0],[13,1],[15,3]] },

'b': { width: 19, points: [[4,21],[4,0],[-1,-1],[4,11],[6,13],[8,14],[11,14],[13,13],[15,11],[16,8],[16,6],[15,3],[13,1],[11,0],[8,0],[6,1],[4,3]] },

'c': { width: 18, points: [[15,11],[13,13],[11,14],[8,14],[6,13],[4,11],[3,8],[3,6],[4,3],[6,1],[8,0],[11,0],[13,1],[15,3]] },

'd': { width: 19, points: [[15,21],[15,0],[-1,-1],[15,11],[13,13],[11,14],[8,14],[6,13],[4,11],[3,8],[3,6],[4,3],[6,1],[8,0],[11,0],[13,1],[15,3]] },

'e': { width: 18, points: [[3,8],[15,8],[15,10],[14,12],[13,13],[11,14],[8,14],[6,13],[4,11],[3,8],[3,6],[4,3],[6,1],[8,0],[11,0],[13,1],[15,3]] },

'f': { width: 12, points: [[10,21],[8,21],[6,20],[5,17],[5,0],[-1,-1],[2,14],[9,14]] },

'g': { width: 19, points: [[15,14],[15,-2],[14,-5],[13,-6],[11,-7],[8,-7],[6,-6],[-1,-1],[15,11],[13,13],[11,14],[8,14],[6,13],[4,11],[3,8],[3,6],[4,3],[6,1],[8,0],[11,0],[13,1],[15,3]] },

'h': { width: 19, points: [[4,21],[4,0],[-1,-1],[4,10],[7,13],[9,14],[12,14],[14,13],[15,10],[15,0]] },

'i': { width: 8, points: [[3,21],[4,20],[5,21],[4,22],[3,21],[-1,-1],[4,14],[4,0]] },

'j': { width: 10, points: [[5,21],[6,20],[7,21],[6,22],[5,21],[-1,-1],[6,14],[6,-3],[5,-6],[3,-7],[1,-7]] },

'k': { width: 17, points: [[4,21],[4,0],[-1,-1],[14,14],[4,4],[-1,-1],[8,8],[15,0]] },

'l': { width: 8, points: [[4,21],[4,0]] },

'm': { width: 30, points: [[4,14],[4,0],[-1,-1],[4,10],[7,13],[9,14],[12,14],[14,13],[15,10],[15,0],[-1,-1],[15,10],[18,13],[20,14],[23,14],[25,13],[26,10],[26,0]] },

'n': { width: 19, points: [[4,14],[4,0],[-1,-1],[4,10],[7,13],[9,14],[12,14],[14,13],[15,10],[15,0]] },

'o': { width: 19, points: [[8,14],[6,13],[4,11],[3,8],[3,6],[4,3],[6,1],[8,0],[11,0],[13,1],[15,3],[16,6],[16,8],[15,11],[13,13],[11,14],[8,14]] },

'p': { width: 19, points: [[4,14],[4,-7],[-1,-1],[4,11],[6,13],[8,14],[11,14],[13,13],[15,11],[16,8],[16,6],[15,3],[13,1],[11,0],[8,0],[6,1],[4,3]] },

'q': { width: 19, points: [[15,14],[15,-7],[-1,-1],[15,11],[13,13],[11,14],[8,14],[6,13],[4,11],[3,8],[3,6],[4,3],[6,1],[8,0],[11,0],[13,1],[15,3]] },

'r': { width: 13, points: [[4,14],[4,0],[-1,-1],[4,8],[5,11],[7,13],[9,14],[12,14]] },

's': { width: 17, points: [[14,11],[13,13],[10,14],[7,14],[4,13],[3,11],[4,9],[6,8],[11,7],[13,6],[14,4],[14,3],[13,1],[10,0],[7,0],[4,1],[3,3]] },

't': { width: 12, points: [[5,21],[5,4],[6,1],[8,0],[10,0],[-1,-1],[2,14],[9,14]] },

'u': { width: 19, points: [[4,14],[4,4],[5,1],[7,0],[10,0],[12,1],[15,4],[-1,-1],[15,14],[15,0]] },

'v': { width: 16, points: [[2,14],[8,0],[-1,-1],[14,14],[8,0]] },

'w': { width: 22, points: [[3,14],[7,0],[-1,-1],[11,14],[7,0],[-1,-1],[11,14],[15,0],[-1,-1],[19,14],[15,0]] },

'x': { width: 17, points: [[3,14],[14,0],[-1,-1],[14,14],[3,0]] },

'y': { width: 16, points: [[2,14],[8,0],[-1,-1],[14,14],[8,0],[6,-4],[4,-6],[2,-7],[1,-7]] },

'z': { width: 17, points: [[14,14],[3,0],[-1,-1],[3,14],[14,14],[-1,-1],[3,0],[14,0]] },

'{': { width: 14, points: [[9,25],[7,24],[6,23],[5,21],[5,19],[6,17],[7,16],[8,14],[8,12],[6,10],[-1,-1],[7,24],[6,22],[6,20],[7,18],[8,17],[9,15],[9,13],[8,11],[4,9],[8,7],[9,5],[9,3],[8,1],[7,0],[6,-2],[6,-4],[7,-6],[-1,-1],[6,8],[8,6],[8,4],[7,2],[6,1],[5,-1],[5,-3],[6,-5],[7,-6],[9,-7]] },

'|': { width: 8, points: [[4,25],[4,-7]] },

'}': { width: 14, points: [[5,25],[7,24],[8,23],[9,21],[9,19],[8,17],[7,16],[6,14],[6,12],[8,10],[-1,-1],[7,24],[8,22],[8,20],[7,18],[6,17],[5,15],[5,13],[6,11],[10,9],[6,7],[5,5],[5,3],[6,1],[7,0],[8,-2],[8,-4],[7,-6],[-1,-1],[8,8],[6,6],[6,4],[7,2],[8,1],[9,-1],[9,-3],[8,-5],[7,-6],[5,-7]] },

'~': { width: 24, points: [[3,6],[3,8],[4,11],[6,12],[8,12],[10,11],[14,8],[16,7],[18,7],[20,8],[21,10],[-1,-1],[3,8],[4,10],[6,11],[8,11],[10,10],[14,7],[16,6],[18,6],[20,7],[21,10],[21,12]] }

};

CanvasTextFunctions.letter = function (ch)

{

return CanvasTextFunctions.letters[ch];

}

CanvasTextFunctions.ascent = function( font, size)

{

return size;

}

CanvasTextFunctions.descent = function( font, size)

{

return 7.0\*size/25.0;

}

CanvasTextFunctions.measure = function( font, size, str)

{

var total = 0;

var len = str.length;

for ( i = 0; i < len; i++) {

var c = CanvasTextFunctions.letter( str.charAt(i));

if ( c) total += c.width \* size / 25.0;

}

return total;

}

CanvasTextFunctions.draw = function(ctx,font,size,x,y,str)

{

var total = 0;

var len = str.length;

var mag = size / 25.0;

ctx.save();

ctx.lineCap = "round";

ctx.lineWidth = 2.0 \* mag;

for ( i = 0; i < len; i++) {

var c = CanvasTextFunctions.letter( str.charAt(i));

if ( !c) continue;

ctx.beginPath();

var penUp = 1;

var needStroke = 0;

for ( j = 0; j < c.points.length; j++) {

var a = c.points[j];

if ( a[0] == -1 && a[1] == -1) {

penUp = 1;

continue;

}

if ( penUp) {

ctx.moveTo( x + a[0]\*mag, y - a[1]\*mag);

penUp = false;

} else {

ctx.lineTo( x + a[0]\*mag, y - a[1]\*mag);

}

}

ctx.stroke();

x += c.width\*mag;

}

ctx.restore();

return total;

}

CanvasTextFunctions.enable = function( ctx)

{

ctx.drawText = function(font,size,x,y,text) { return CanvasTextFunctions.draw( ctx, font,size,x,y,text); };

//ctx.measureText = function(font,size,text) { return CanvasTextFunctions.measure( font,size,text); };

ctx.fontAscent = function(font,size) { return CanvasTextFunctions.ascent(font,size); }

ctx.fontDescent = function(font,size) { return CanvasTextFunctions.descent(font,size); }

ctx.drawTextRight = function(font,size,x,y,text) {

var w = CanvasTextFunctions.measure(font,size,text);

return CanvasTextFunctions.draw( ctx, font,size,x-w,y,text);

};

ctx.drawTextCenter = function(font,size,x,y,text) {

var w = CanvasTextFunctions.measure(font,size,text);

return CanvasTextFunctions.draw( ctx, font,size,x-w/2,y,text);

};

}

jQuery('#saveme').click(function() {

//Clear any timeouts

clearTimeout(saveTimeOut);

saveAll(g\_all\_shapes);

});

jQuery('#ellipseopt').click(function() {

mode = "ellipse";

});

jQuery('#lineopt').click(function() {

mode = "line";

});

saveAll = function() {

var canvas = document.querySelector('#mycanvas');

var dataURL = canvas.toDataURL("image/jpeg", 0.7);

Qualtrics.SurveyEngine.setEmbeddedData('image\_data', dataURL);

if(g\_all\_shapes.all\_ellipses.length > 0)

{

// write out the summary data as a csv string into the embedded data variable 'summary\_data'

// Only if we have an ellipse!

g\_all\_shapes.setEllipseSelf(g\_self\_idx, true);

data\_str = g\_all\_shapes.results();

Qualtrics.SurveyEngine.setEmbeddedData('summary\_data', data\_str);

}

//console.log( data\_str);

jQuery('#NextButton').click();

return true;

}

function timeReminder() {

//Show warning and only save button

jQuery('#timeWarning').show();

jQuery('#draw\_options').hide();

jQuery('#undo').hide();

jQuery('#saveme').show();

//Hide popup incase its showing

jQuery('#textAreaPopUp').hide();

//Unbind drawing handlers

jQuery("#tempcanvas" ).unbind( "mousedown" );

jQuery("#tempcanvas" ).unbind( "mouseup" );

jQuery("#tempcanvas" ).unbind( "mousemove" );

if(mode == "label")

{

//If we are in label mode, then we must have an ellipse draw, so auto rollback...

mainctx.drawImage(undocanvas, 0, 0);

}

return true;

}

//At the very end of definitions loaded ... set the timers...

saveTimeOut = setTimeout(timeReminder, timeMS); //Auto save and move on once time is done.

});

Qualtrics.SurveyEngine.addOnUnload(function()

{

/\*Place your JavaScript here to run when the page is unloaded\*/

});

**Fourth Step:** Make changes to the survey flow

* The helpdesk question has to be outside of the survey flow; to do this go to "Survey Flow" and press delete for the helptext block.
* Machine generated alternative text:
  Set Embedded Data: 
  Add Below 
  Seta Value Now 
  Add Below 
  Move 
  Move 
  Duplicate 
  Duplicate 
  Add From Contacts 
  Options 
  Duplicate 
  Duplicate 
  Duplicate 
  Options 
  Delete 
  Delete 
  Delete 
  Delete 
  Delete 
  undo_text 
  ellipse_text 
  line_text 
  save_text 
  Undo 
  Oval 
  Line 
  Save 
  Save 
  300000 
  Time out. Please click Save to proceed. 
  savetextbox_text 
  time_milliseconds 
  time_out_text 
  helptext 
  Add a New Field 
  Show Block: Instructions (1 Question) 
  Show Block: Drawing (3 Questions) 
  Show Block: End (1 Question) 
  Set Embedded Data: 
  image_data 
  Value will be set from Panel or URL 
  Add a New Field 
  + Add a New Element Here 
  Add Below 
  Add Below 
  Add Below 
  Move 
  Move 
  Move 
  Add From Contacts We have to define the variables for the sociogram. To do this go to "Survey Flow", add a new section "Set Embedded data", and move it to the top of the questionnaire. Then, add the following variables (as well as an additional variable called “summary\_data” at the end!, not shown in image!):
* For the last variable "helptext", you will have to pipe the value in from the questionnaire. To do so, click on the value field, select "Insert Piped Text", and follow the flow until you find the question containing the helpdesk text. Select it.

Machine generated alternative text:
X My WORKTO DOS-GcX Dashboard -Kent PX 
M Mark sent you a ne 
Survey Flow 
Sociogram drawing 
S 
Set Embedded Data: 
Technical Team - 
Google Keep 
Search... 
Asset loan manage 
x 
RT at a glance 
Technical Team - 
Edit Survey QualtF 
Zoom Out Zoom In 
Cl Show Flow IDs 
Save Flow 
undo_text 
ellipse_text 
You 
line_text 
save_text 
Undo 
Oval 
Line 
Save 
Save 
300000 
Time out. Please click Save to proceed. 
savetextbox_text 
time_milliseconds 
time_out_text 
helptext 
helptext 
Pipe text from a... 
Haa 
Survey Question 
Embedded Data Field 
Geol? Location 
Survey Links 
Date / Time 
Opt-out Link 
Random Number 
Custom Value 
Insert Piped Text 
Add a New Field 
Show Block: Instructions (1 Question) 
Show Block: Drawing (3 Questions) 
Show Block: End (1 Question) 
Set Embedded Data: 
Q3 In this task, you will create what is known a 
a sociogram. This is basically a picture of your 
Qö Timing 
Q4 Please draw your sociogram in the box 
below. 
QI 
St'e:// Field/ellipse_text) 
Q2 This is the end 
helptextStartby putting yourself inan oval, an 
add your initials to it. Next, draw ovals around 
you 
image_data 
Value will be set from Panel or URL. 
Add a New Field 
+ Add a New Element Here 
Seta Value Now 
Panels Field 
Loop & Merge 
Add 
Quota 
Question Text 
Cancel 

* Finally, we need a place to store the image data from the sociogram. For this we add another embedded data block at the end. To do this go to "Survey Flow", add a new section "Set Embedded data", and move it to the end of the questionnaire. Create a variable called "image\_data", but leave the value blank. Also, create a variable “summary\_data”, but leave the value blank.
* The sociogram should not be all set up and ready to use.

**Comments:**

* Data from the sociogram should be included in the exported data qualtrics file.
* It might be better to import to CSV instead of SPSS, since there is a long string of raw image data included that might mess up SPSS formatting.
* The sociogram data links with the identifier give to each participant by Qualtrics. T**hese should be unique within each questionnnaire, but not necessarily between questionnaires.**
* Generall, if text needs to be edited, it is saver to do so in the HTML view of the question than in the rich content editor, as this can mess up the underlying scripts.
* If you preview the task, the block preview can cause some problems and not show the task correctly. Generally, the safest way is to go through the whole questionnaire to test it.
* If you make changes to the questionnaire, do not forget to publish the questionnaire anew.