Procedures for installing necessary tools and running the code:

The purpose is to be able to install and develop code by using BPMN-IO BPMN viewer library. You can find some descriptions on their own web page: https://bpmn.io/ (and you should examine this site, use their examples and forums etc. when you can't figure out something). Below are the descriptions I created and applied myself:

- Install node.js by using the installation file (.mis) from: https://nodejs.org/en/
- Open a command line (CL) and browse to the folder that you want to place your working codes (you can later place the codes in a lower-level folder, that is no problem) (by cd command)
- Install NPM by typing the following on this command line: npm install --save bpmn-js
- Install browserify: npm install -g browserify
- Install Uniq: npm install uniq npm install -g grunt-cli
- Install brfs: npm install brfs
- Install jquery: npm install jquery
- In the same folder, for testing, create a file named main.js with the following content:

```
var unique = require('uniq');
var data = [1, 2, 2, 3, 4, 5, 5, 5, 6];
console.log(unique(data));
```

- Run this command on CL: browserify main.js -o bundle.js
- Create an .html file for testing and write the following inside:

```
<script src="bundle.js"></script>
```

- You should observe that given array is written as output in the console.
- Install dependencies of NPM:

```
npm install -g npm
```

From <http://stackoverflow.com/questions/27637642/npm-throws-error-couldnt-read-dependencies>

- Install Grunt:

```
npm install -g grunt-cli
```

From <http://gruntjs.com/getting-started>

Part 2: Get ready to install jquery and build bpmn-io code:

- Open a node.js command prompt (because you have node.js now, you should be able to do it like you start a new application in windows)
- Change directory to the folder where node_modules folder exists (under this folder, the files you globally installed in the previous steps exist. For me, the path is: C:\users\user_name\node_modules
- Run the following commands:

```
npm install jquery
npm install brfs
```

Change directory to the folder which contains your .js files in your project (i.e. something like prime.js for our project)

- Run the following command in node.js command prompt, for whatever .js you need to run (meaning you need to change the name "prime-web.js") browserify -t brfs prime-web.js > prime-web_bundle.js

- Your code uses the “browserified” javascript file. You need to generate it yourself, everytime you make a change in this code.

Notes about the code

The following main files are common for the projects: prime.js and main.css. Prime.js file is the main file which is all about the logic and details of the html user interface.

prime.js

- This code has two “modes”. appType may be “PRIMEWeb”, which means the js is called from the html file of the web page. Or it can be “Exp2”, then it means that it is called from one of the html files of the experiment.
- When you see some code enclosed with the condition “if(appType.indexOf('Exp2') != -1)”, this means that this code is specific to the experiment. Those parts are usually about the process models and process elements used in the experiment, referred to with their sid’s. You don’t need to understand those parts.
- The animation on the web page can run in two modes: continuous and stepwise. In many places of the code, there is a check like (if(isStepAnimSelected)==true). This is about the mode of the animation.
- All the code about filters, for example the function populateRoleDropdown(), etc. are about Filtering functionality on the web page (seen when clicked to Show Filters button). This feature is experimental and does not properly work. Do not use those.

This is the flow of the code (roughly):

- When js is called from .html, the main code parts are run (approx. line 927-970)
- At the end of this part, openDiagram() is called.
- This function initiates the diagram (as assigned to xmlDiagram variable) and does a lot of other things (its code is at the beginning, starting from line 151)
- Starting from line 257 in this function, an eventBus is registered. So, this part of the code is called whenever a click event (on the canvas-where the bpmn diagram is showed) occurs. Everything till the end of this function (line 914) is about this event. This is how the code continues (calling other functions) – other than other type of user input: button clicks. (beware, there are a lot of codes here only for Exp2, ignore them, don’t get confused)
- From line 978 to 1360, all functions are the ones called when a button on the user interface is clicked. The tags used (such as *open-url-click* are obtained from the html file.
- Starting from line 1366 to 1410, functions adjusting the timers are listed. These are called from others, such as from the eventbus etc. and are very important to make the animation. Pay attention that javascript cannot be stopped like other traditional languages. It always runs. Timeout mechanism is how we make the animation.

- Through the end, there are functions such as setConvergingParallelGatewayArray(), markSeqInOrder(), findNextObject(),findGatewayCouples(),findConvAndofGivenDivAnd(). These contain the algorithm logic. I think you never need to deal with the content of these functions.
- markObjectAndSeqFlow(), markObject(), markCleanObject(), markSeqFlow(), markSeqFlowwithGivenId() are all about painting the process elements when the time comes. I again think you don’t need to consider the details.