980B Project Report

Digital simulation web application

Instructor: Dr. Norvell
Hui Wan
201469897

Table of Contents

Summary

Similitude – A web based digital simulation system. In this system, user can draw logic gates on the canvas and connect gates by using wires. Moreover, in this system user could draw several circuit diagrams at the same time, and every circuit diagram could be a component which we call it compound component in other circuit diagram, we could edit a compound component in one circuit diagram, and those changes in the compound component will affect the original circuit diagram. Besides this, there are some other function has been implemented such as undo, redo function, copy and paste functions.

In this term, I have done lots of functions:

1 Most of editing actions have been implemented, such as:
    1. Move, delete, add components, links, and endpoints
    2. Create new circuit diagram, delete a circuit diagram, editing circuit diagram
    3. Undo, redo functions
    4. Create compound component and display the inner circuit diagram, and editing functions for compound component.
2. Change in program structure
3. Some invisible changes such as changing the controller, changing the model…

Technologies Applied

Until now, only four technologies applied in this project:

1 Haxe: Haxe is a toolkit that can be used to build cross-platform tools and frameworks. The reason why this project chosen Haxe is that Haxe has strong type check mechanisms which javacript does not have.

2 Boostrap: Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web. The reason why we choose Boostrap as the front-page design framework is that this web application could use on the mobile devices, therefore, we need this framework to make this web application easy to design

3 jQuery : it is a fast, small, and feature-rich JavaScript library.

4 HTML5 : It is the fifth and current version of the HTML standard. This web application only support the web browser which support HTML 5, because this project use the some new HTML 5 features

Design

MVC

The project based on the MVC pattern. Only controller could update the model and the major update comes from command pattern, command pattern contains lots of commands, such as add command, delete command, move command, copy command and so on. For model part, in this project, gate can draw itself, only way to draw those gates on the canvas is call CircuitDiagram.Draw() method, then the circuit diagram will tell every component to draw themselves. In the controller, some functions will listen all the mouse action and keyboard action from the web application and step into the corresponding controller state to do the right action. Figure one shows how it works.

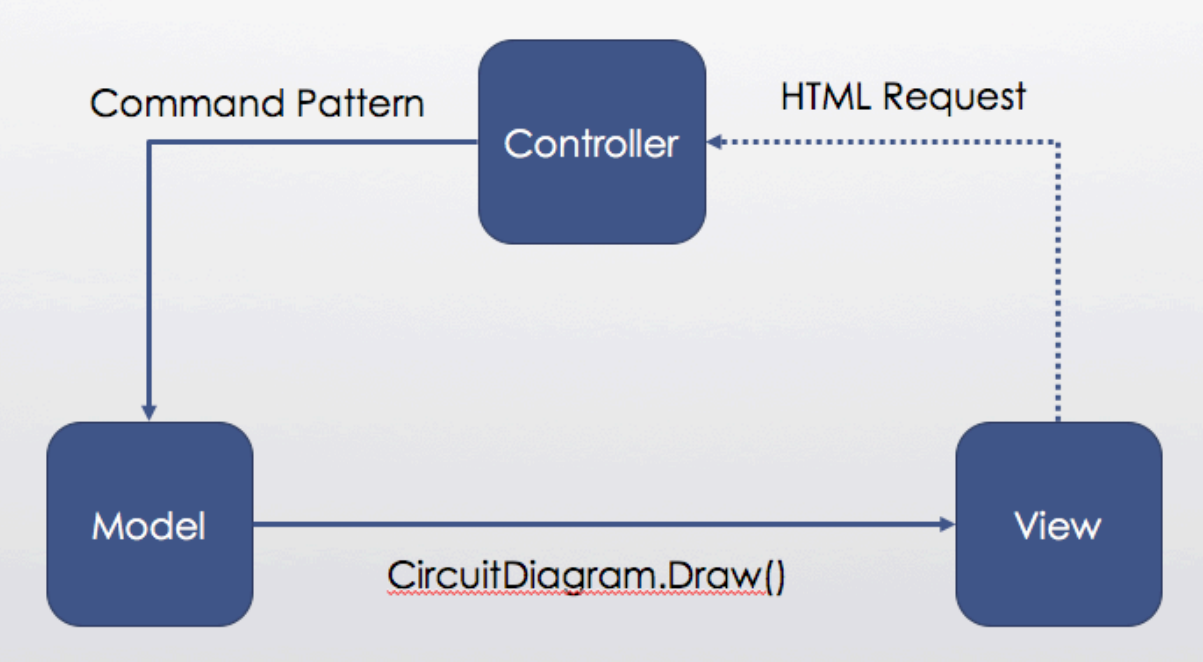

Figure 1: MVC Structure

Controller Class Diagram

In this project, the whole system starts from a class called FolderState, which is the main controller for the whole system. It describers how to switch between different circuit diagram and how to make sure the canvas update the right circuit diagram. After the main() function create a new FolderState class, the FolderState class will automatically generate

the UpdateCanvas class which is used to re-draw the canvas, also generate the UpdateCircuitDiagram class which responsible to connect the model, all of the command action are controlled by this class, also generate the updateToolBar class to make sure the tool bar display correctly, also generate the SideBar class to make sure in which circuit diagram user can draw what component, also will generate the COntrollerCanvasContext class to listen the operations happens on the canvas. Figure 2 shows how does it works.

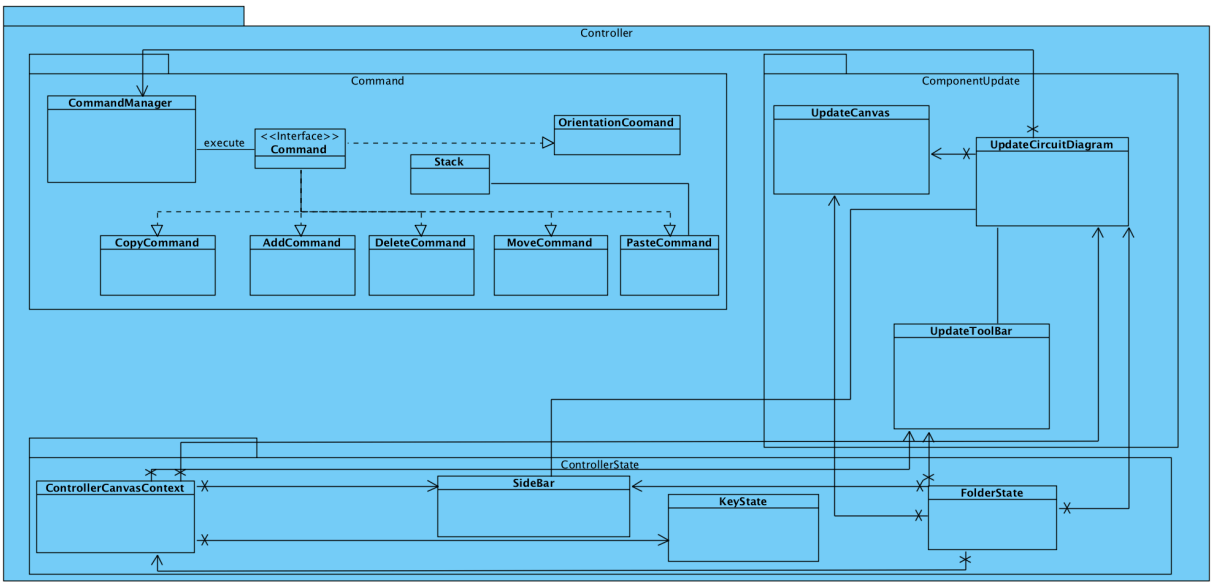

Figure 2: Controller class diagram of the whole system

Model Class Diagram

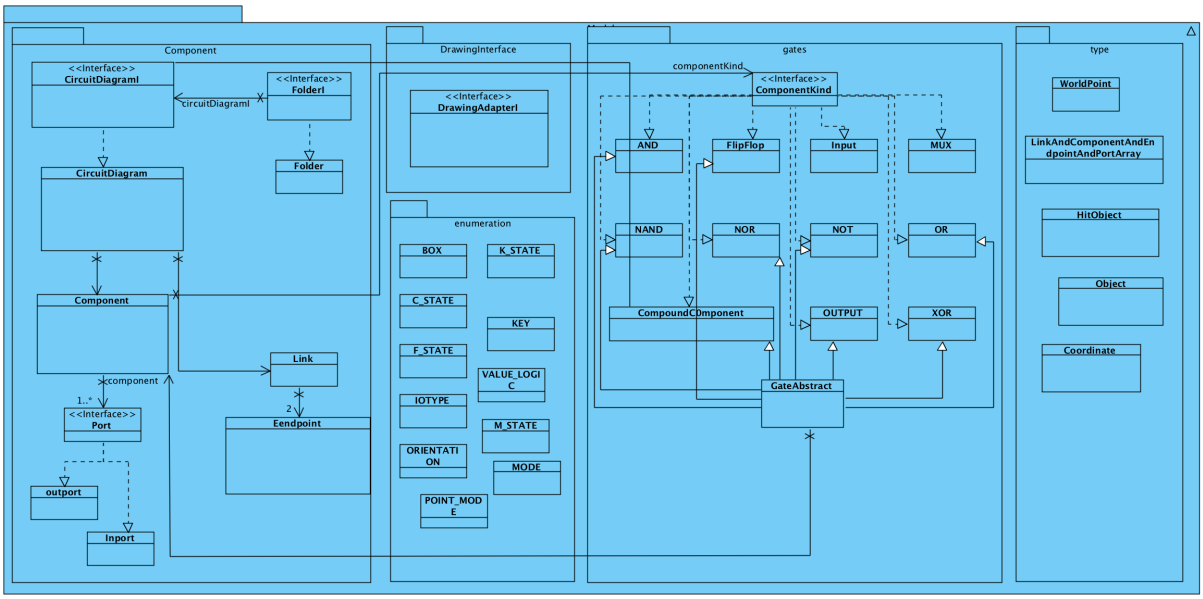

Figure3 : Model class diagram of the whole system

In the model, everything corresponding to the circuit diagram will be stored in here. In the Component package, Folder is on the top level, because Folder has the responsibility to create circuit diagram, every folder should contain one or more circuit diagram. In the circuit diagram, it contains several components and links, each component has more than one port and only one component kind, and each link has two endpoints. Figure 3 shows how does the circuit diagram works. Then we can get the data structure for the whole circuit diagram.

Figure 4 shows the data structure of the whole system.

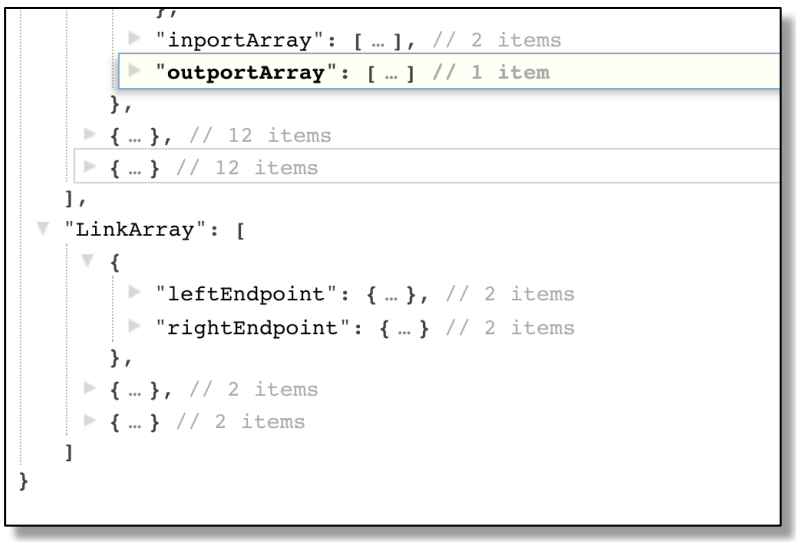

Figure 4: The data structure of the whole system

View Class Diagram

For view class diagram, for the class diagram point of view it is much easier than others because those class only responsible for make sure each component draw themselves correctly. In this system, there is a DrawComponentI to make sure every component knows how to draw themselves, figure 5 shows how does it works.

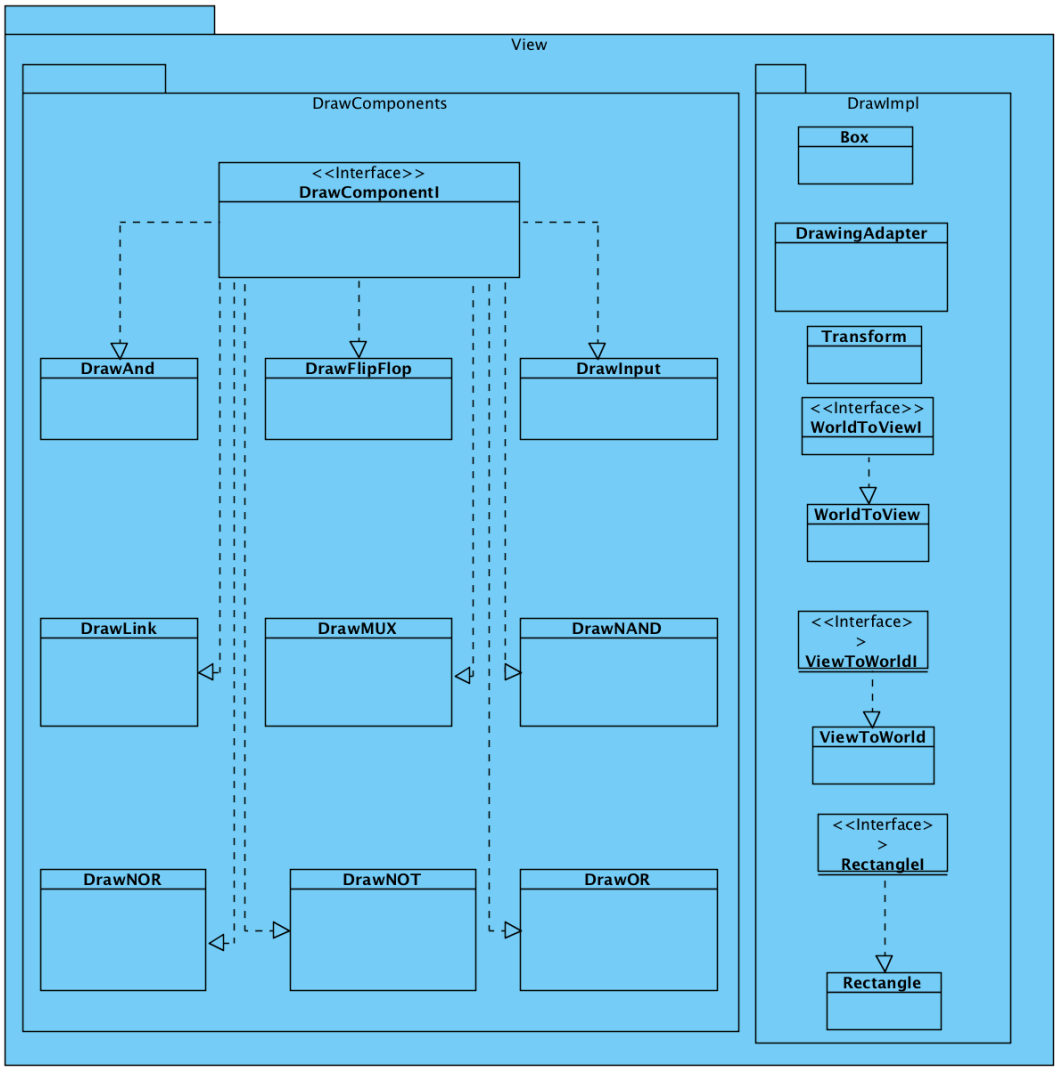

Figure 5: view class diagram of the whole system

World Coordinate system

In order to make sure the compound component draw correctly, in this system, there is a world coordinate system has been established. When we draw a compound component, we need to relocated all the component in the sub- circuit diagram, so we need to separate the world coordinate and view coordinate, that means every coordinate stored in the circuit diagram should be translated into world coordinate, and before the system

draw the circuit, those world coordinate needs to be translated into view coordinate. There is very important function to make the transform correctly which is the makeTransform() function(as shown on figure 6).

```
function makeTransform():Transform{
    var transform:Transform = Transform.identity();
    transform = transform.translate(-circuitDiagram.getCircuitDiagramCenterPoint().get_xPosition(),
                                    -circuitDiagram.getCircuitDiagramCenterPoint().get_yPosition())
                .scale(component.get_width()/circuitDiagram.get_diagramWidth(),
                       component.get_height()/circuitDiagram.get_diagramHeight())
                .translate(component.get_xPosition(), component.get_yPosition());

    return transform;
}
```

Figure 6: the most important function to make sure the world to view function works.

FolderState

As described before, the FolderState is the main class for the whole system. It controls how to create and switch circuit diagram. When the class has been created, it will step into the IDEL state, then create the first circuit diagram and change the state to the current, and then wait for the instructions. If user delete a circuit, the FolderState will notify the Folder to delete one circuit diagram and then go to the previous state to find the previous circuit diagram user accessed and then search the circuit diagram and to open it. If the instruction is search some circuit diagram, the state will go to Search state and find the corresponding circuit diagram. For the Next state now is useless, but it used before, maybe it will be useful in the future, so I leave it there. Figure 7 shows the state diagram for the FolderState.

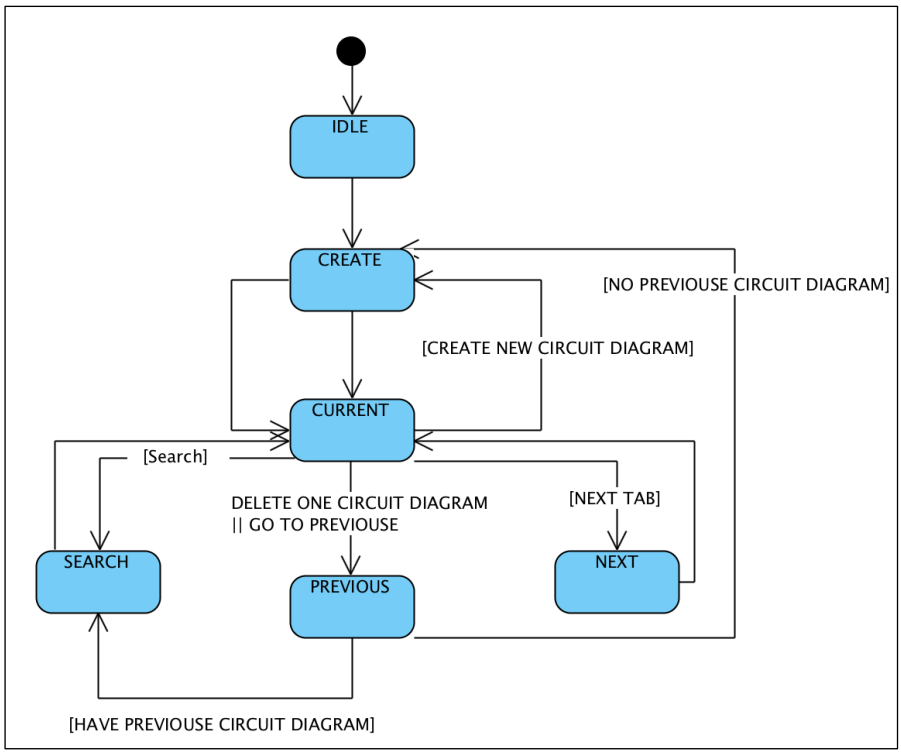

Figure 7: State diagram for FolderState

Editing State

For the editing state, it shows how does the ControllerCanvasContext works. When we create a new ControllerCanvasContext, it will step into IDEL state, and then wait for instructions. If user click on the canvas without press the alt key, the state machine will go to SINGLE_SELECTION state, after mouse down, the program will identify what have you selected, in this project, we assume that the priority of port is higher than endpoint, the priority of endpoint is higher than component and link. Component and link have the same priority. After we got an object (endpoint, port, component and link), we can move or create. If user click mouse down with press alt key, the state machine will step into the MULTI-SELECTION state, In this state, user can selected more than one object, but in this state, user can only move object. In this state diagram, the CREATE_COMPONENT state should be triggered by the outside signal which is SideBar. Figure 8 shows the state diagram for ControllerCanvasContext.

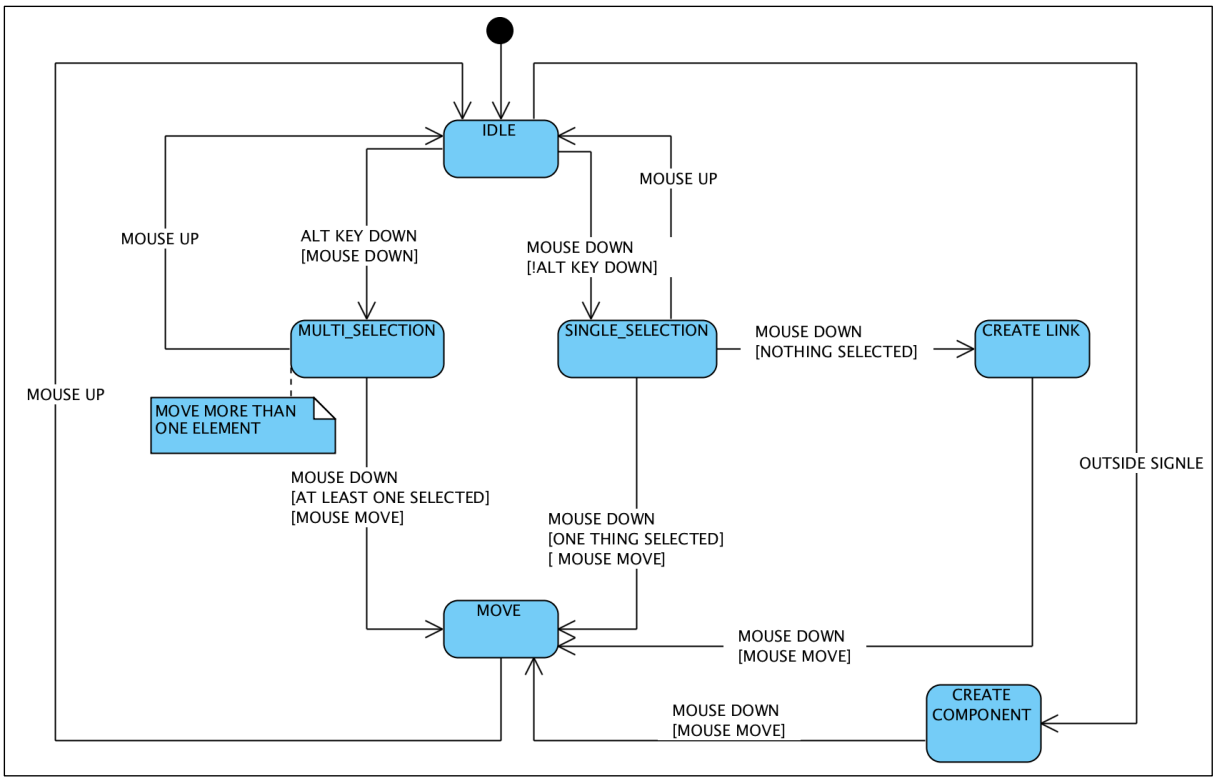

Figure 8: State diagram for ControllerCanvasContext

Testing

I will list the major functions in this section. Some of the function is stilling in test.

Editing Component

Add component is very easy, just need to click the gate you want to create, then move mouse into the canvas, the corresponding component will follow the mouse display on the canvas, and this function works. For moving component or link, user just need to select one component or link, and then move it. For delete a component, user just need to select a component or link, then click the delete button, then the corresponding component or link will be deleted. For rotate a component, just select a component and then click the Orientation button to select one direction. All of these function are working now.

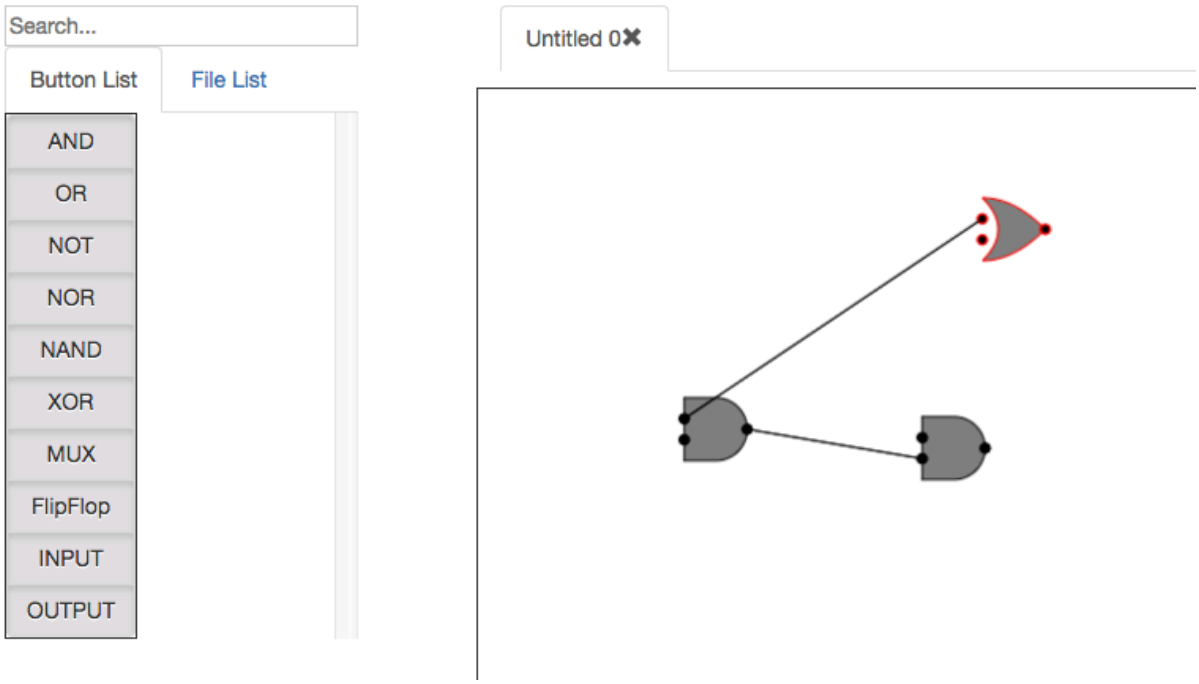

Figure 9: Edit Components

ADD Circuit Diagram

Add a new circuit diagram is very easy, just need to click the add "button" – actually it is a tab, then a new circuit diagram will be created.

Figure 10: Add Circuit diagram

Delete Circuit Diagram

Delete a circuit diagram is also very easy, just need to click the "X" button on the right of the name of this circuit diagram, after click the "X" button, there is an alert will pop up to make sure user want to delete it. Figure 11 shows where is the delete button and figure 12 shows the alert.

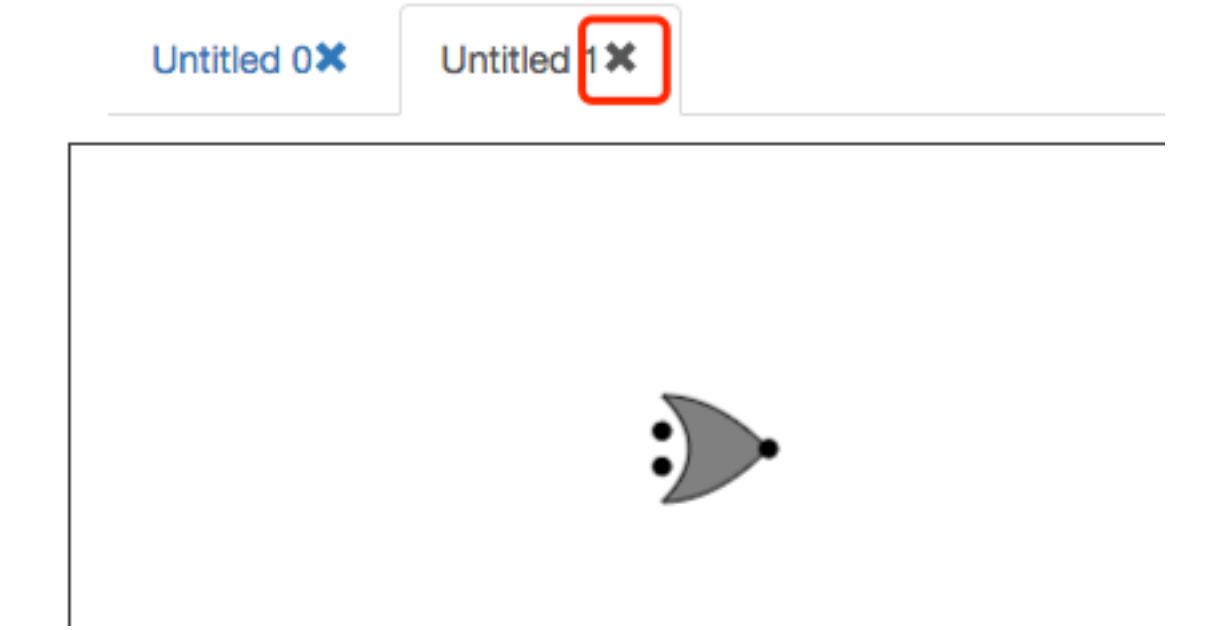

Figure 11: Delete Circuit Diagram button

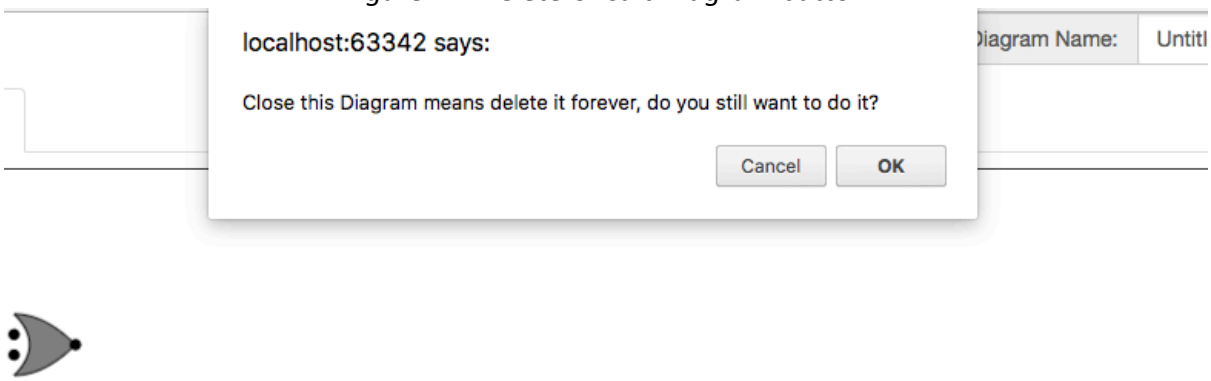

Figure 12: Alert for delete a circuit diagram

Change Name For Circuit Diagram

Change name for circuit diagram is very easy, just enter new name and press enter, then the system will verify whether this name is vailed, if vailed, the input frame will turn green, or it will become red. Figure 13-15 shows how to change name.

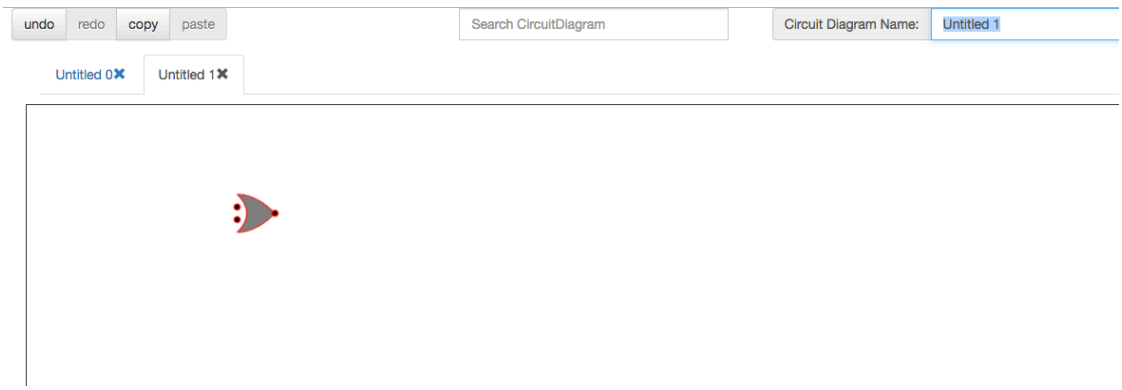

Figure 13:Before change circuit diagram's name

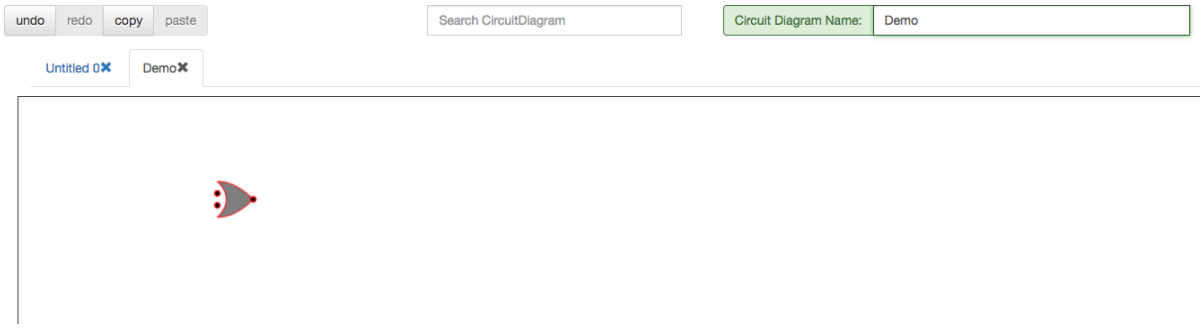

Figure 14:Change name successful

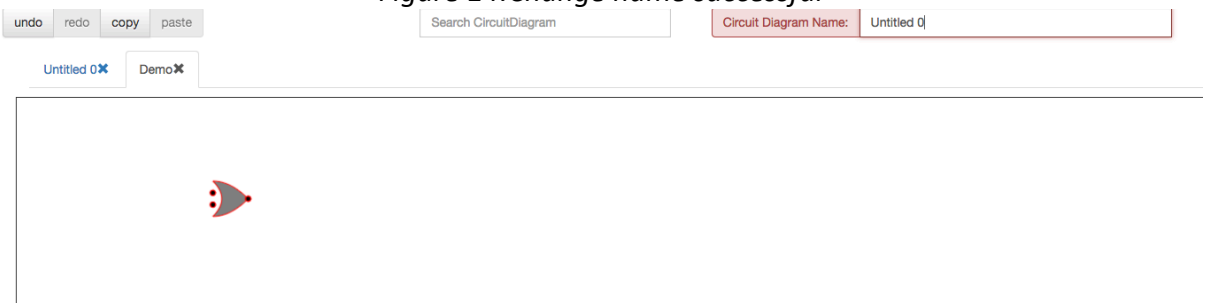

Figure 15: Failed to change a name

Search Circuit Diagram

Search a circuit diagram is very easy, just need to type a part name of the circuit diagram, then there will be some hint to help you to get the circuit diagram. And just click it, the system will switch the circuit diagram. Figure 16-18 shows what happens when we are searching a circuit diagram.

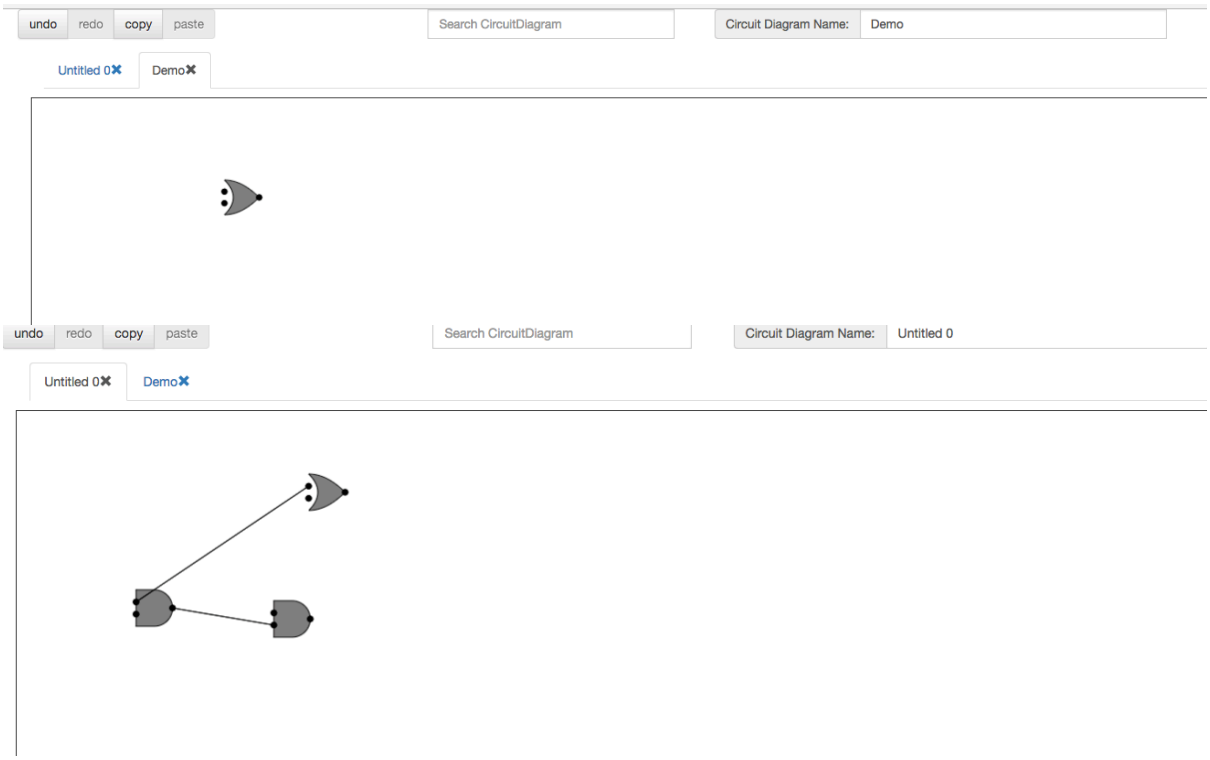

Figure 16: Before search

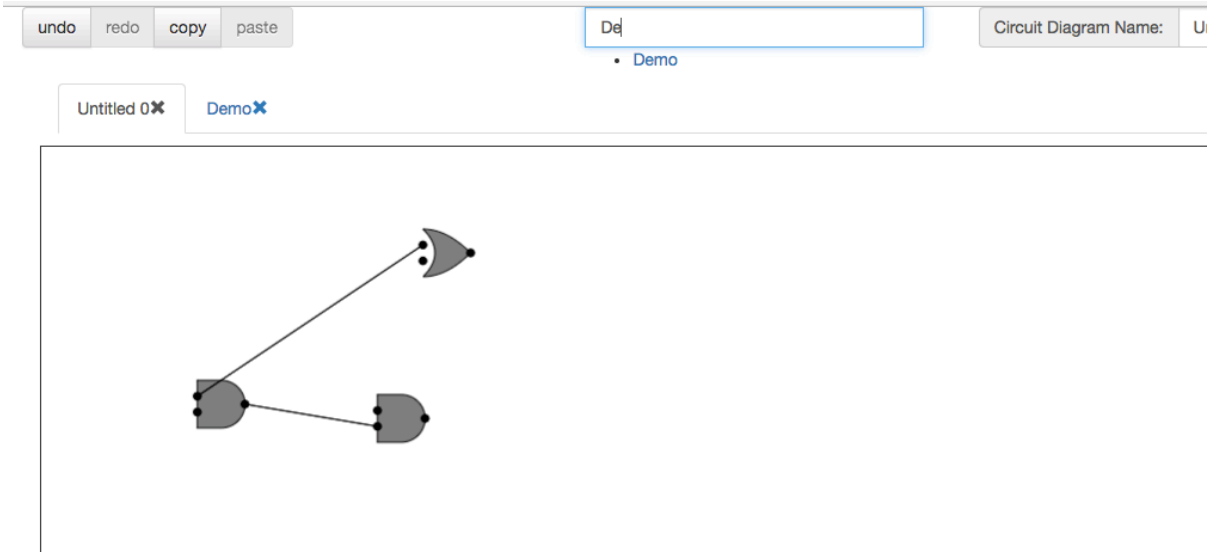

Figure 17: In search

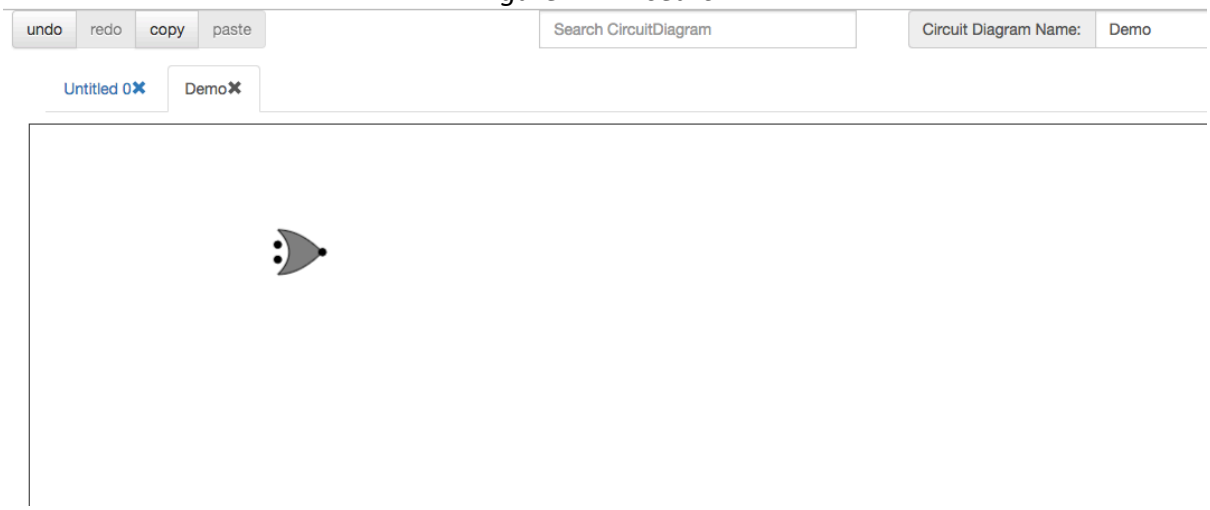

Figure 18: After Search

UNDO, REDO

For undo, redo function, just need click the undo or redo button. Figure 19 -21 shows the undo and redo functions are working.

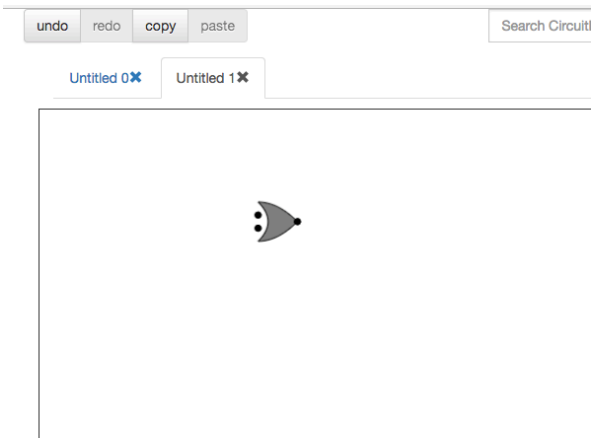

Figure 19: Before undo

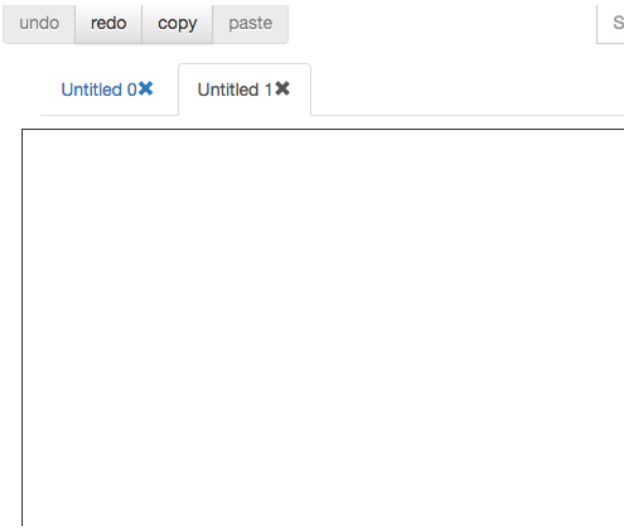

Figure 20: After Undo

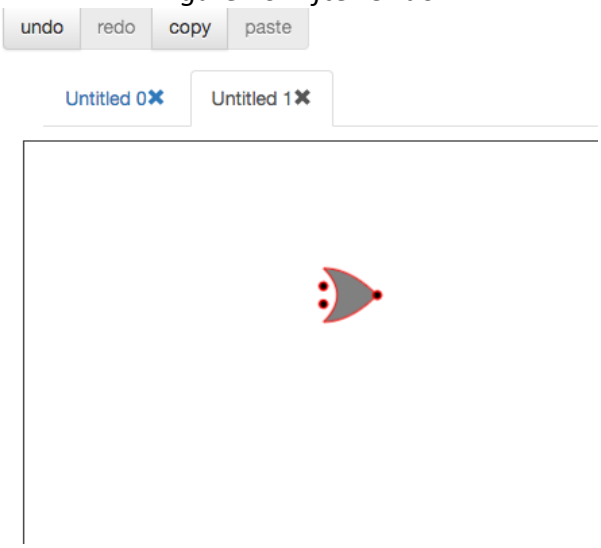

Figure 21: After Redo

COPY and Paste

For copy and paste, user just need to select an object to copy, then click paste button and then move mouse into the canvas, click on the canvas, the copied object will be shows up. Figure 22 and 23 shows the copy and paste.

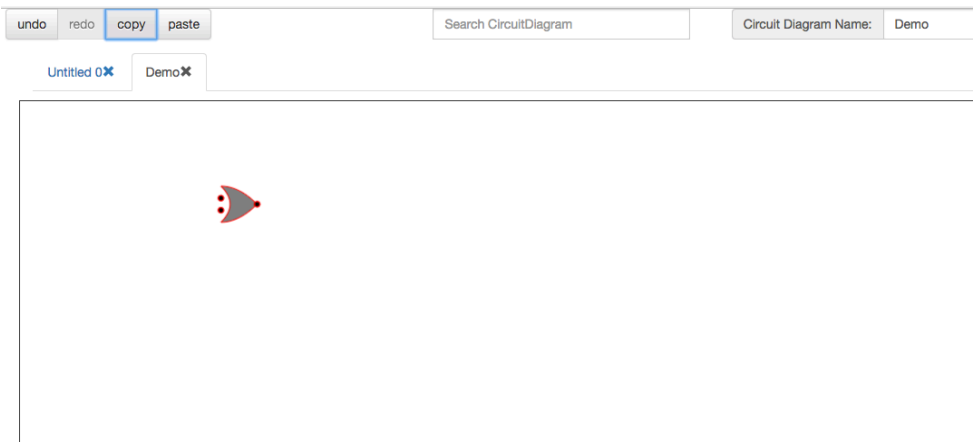

Figure 22: After I click the copy button

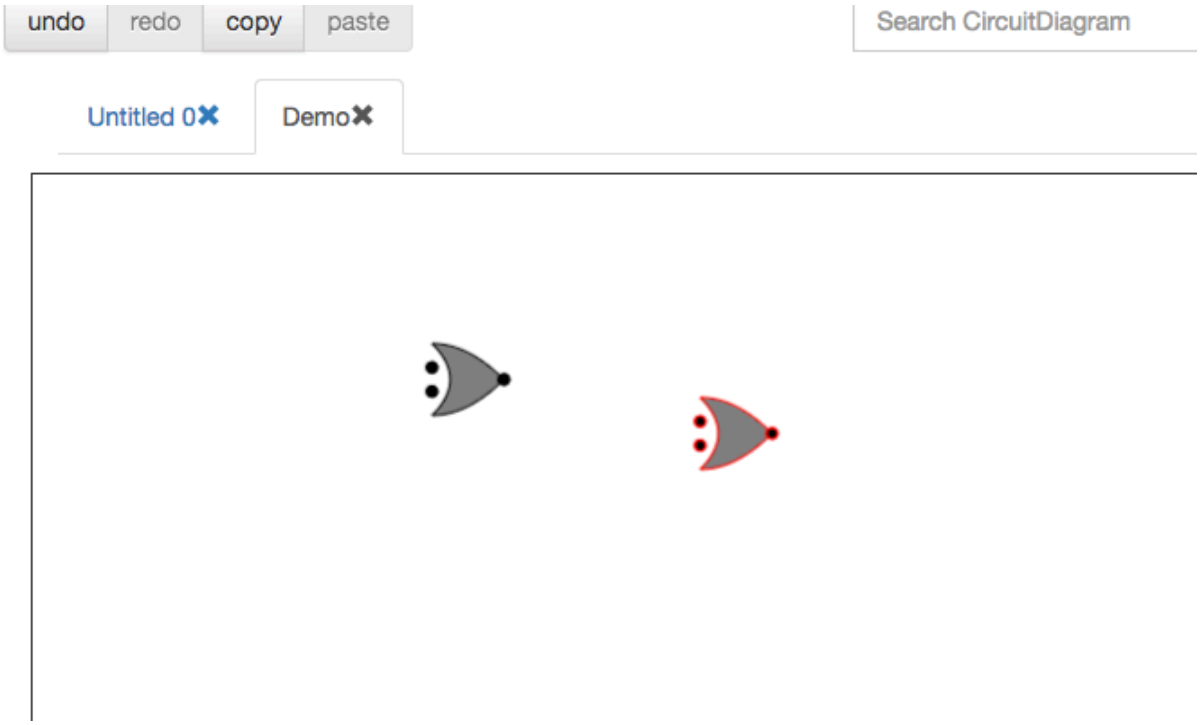

Figure 23:After I click the paste button

Add Compound Component

Add compound Component is the same as add component. Figure 24 shows draw a compound component correctly.

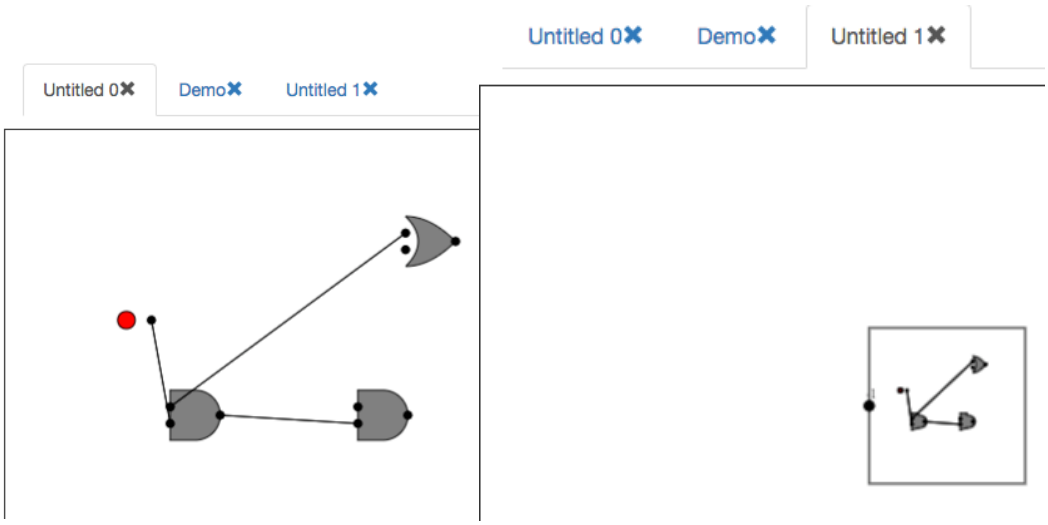

Figure 24: Shows the compound component

Question Encountered

1. Haxe has its own disadvantages, for example:
    1. Its community is not very active, so finding a solution for a specific issue is not easy.
    2. Haxe does not support jQuery very well, some features are not supported by standard Haxe library.
2. Requires knowledge of HTML layout. The layout of the HTML always take times.

Future work

Until now, there are still some bugs need to be fixed, but not too many. The drawing part almost done now, after this the simulation part should be start soon. And the server side also can start now.

Appendix

Github link: https://github.com/theodore-norvell/similitude