**Wiki for Archotech**

**Logic Gate Plugin**

**Group Members**

* 1846346 | Jared Harris Dewey
* 1851234 | Michael-John Brewer
* 1853154 | Jonah Alter
* 1877594 | Erin Ollewagen
* 1934122 | Max Parkin

**Description**

Logic Gate plugin is a plugin for Moodle that allows a lecturer to set a Logic Gate type of question in order to test a learner’s understanding on Logic gates. The Plugin enables a learner to drag and drop logic gates onscreen and create a circuit that should match an output created by a lecturer.

**Features**

* A lecturer should be able to type a question into a text-box for the student to answer
* A lecturer should be able to choose to use a logic gate type of question.
* A lecturer should be able to assign an output to a question (ie the answer for the question)
* A lecturer should be able to choose a set selection of logic gates for the students to use in answering the question
* A lecturer should be able to choose the maximum amount of each logic gate that a student can use.
* A student should be able to drag and drop logic gates onto the canvas
* A student should be able to connect logic gates via connecting wires
* The marker needs to be able to recognize the combination the student put down and compare it to the desired output that the lecturer assigned to the question
* The marker needs to be able to give feedback to the student as to whether their output and the lecturers output matched by returning “correct” or “incorrect”
* A student should be able to answer in any order and to change their answer before submitting
* A student should be able save the state of a question, and be able to return to the question in it’s saved state to work on it.
* A student should be able to submit their answers to the marker
* A lecturer should be able to place optional hints to help the students in answering a given question

**High-level Architecture Overview**

Below is a diagram showing the set of principle designs for the plugin. This diagram consists for 4 views: Development View, Logical View, Physical View and Process View. Each view consists of diagrams, which are demonstrated further down.



**Component Diagram**

Below is a diagram showing the components connected (by interfaces) to the plugin. This plugin is directly connected to 3 other components - namely the Web Browser, the Moodle Database, and the Renderer – and is indirectly connected to the Marker via the Renderer and the Moodle Database. Every component connected to a semi-circle requires human input in order to function. Every component connected to a circle produces information for the connected component.



**State Diagram**

Below is a diagram showing the different states the plugin could be in. The circle in the top left is the starting node. From there, one is either a lecturer preparing a question for a student, or one is a student answering a question prepared by a lecturer. By following the arrows, the process of preparing and marking a question can be observed. The bottom black circle is the ending node where the process terminates.



**Deployment Diagram**

Below is a diagram showing the physical hardware on which the Plugin is run. In this diagram there are 2 device nodes, the User Client and the Web Server. Within in the User Client is a device, the Web Browser, which contains an artifact – HTML5.



**Activity Diagram**

Below is a diagram showing how the students and lecturers make use of the plugin, starting at the starting node at the top, and ending at the node at the bottom.

**Installation**

Download the zip from:

latest (master branch) - [github.com/Jaredrhd/Archotech](http://github.com/Jaredrhd/Archotech)

Unzip it into the question/type folder, and then rename the new folder to “Logic Gate”, if it isn’t that already.

**Links**

* Travis-CI | [travis-ci.org/github/Jaredrhd/Archotech](https://travis-ci.org/github/Jaredrhd/Archotech)
* Github  | [github.com/Jaredrhd/Archotech](https://github.com/Jaredrhd/Archotech)
* Coveralls | [coveralls.io/github/Jaredrhd/Archotech](https://coveralls.io/github/Jaredrhd/Archotech)