Initial Plan

Logic Circuit Builder

Author: Sam Williams Supervisor: David Walker

CM3203 One Semester Individual Project – 40 Credits

Project Description

The following document will lay out the plans for the project that will be developed from Monday 4th February 2019 (Date of initial plan hand-in) – Friday 10th May 2019 (Date of final report hand-in).

For this project the aim will be to develop a software environment for building computer logic circuits from simpler circuits. At the lowest levels the simplest circuits consist of NAND, NOR, NOT, AND, OR and XOR gates, with these we can create a graphical user interface that allows these circuits to be dragged, dropped and connected in any way the user chooses in order to build more complex circuits. With circuits that users have constructed they can then be stored and subsequently accessed to build yet more complex circuits. In addition to this a user will be able to generate truth tables from the constructed circuits.

Logic is the foundation and plays a fundamental role in computer science, so understanding is key. With this software we aim to make understanding logic circuits and how truth tables are generated easier for the user by giving a visual representation that they can edit as they please and generate a truth table from their desired circuit, with this in mind this software is aimed at targeting students looking to study computer science or related fields at a university level, first year undergraduate students studying computer science or related topics (courses that cover logic gates etc) and people looking to improve or test their knowledge on logic circuits. Although some logic circuit builders exist online, there are not many, if none, that offer the ability to generate a truth table from a created circuit. By coupling this with a simple intuitive design for users it will allow for a great teaching tool that can be used in a variety of different situations from allowing lecturers to offer a visual representation of some truth table, to generating a truth table from a complex circuit that would otherwise take a considerable amount of time for someone at a first-year undergraduate level.

Project Aims and Objectives

Aim: Software contains pre-defined logic gates that include at least NAND, NOR, NOT, AND, OR and XOR.

Acceptance Criteria:

- Logic gates should be easily accessible, and each gate should be labelled clearly.
- Symbols for each logic gate should be clear and large.
- List containing the logic gates should be scrollable/dropdown if all symbols do not fit in the window.
- Symbols should match the universal logic gate symbols.

Aim: User interface allows for the logic gates to be dragged, dropped and connected inside a designated workspace within the interface. By allowing the logic gates to be easily dragged, dropped and connected we can offer an intuitive and simple interface for users that give them the freedom to control the components and produce a circuit of any form.

Acceptance Criteria:

- Design should be simple and easy for a new user to create a circuit and generate a truth table.
- Users should be able to select a button that allows them to clear the workspace.
- Users should be able to resize the window.

Aim: Constructed circuits can be stored and easily accessed/loaded into the workspace.

Acceptance Criteria:

- Saved circuits should be saved into a certain file type, these file types can then be called by the software to load the circuit into the workspace. By saving as a file, users can obtain files from other users and load these into the workspace, allowing for different circuits to be shared and edited.
- When a file is loaded into the workspace, all logic gates and corresponding connections between these logic gates should also be generated.
- Loaded files should be editable.

Aim: Truth table can be generated from a constructed circuit.

Acceptance Criteria:

- Truth table should be displayed in a separate window in tabular form with the option of saving/exporting as a file for the user.
- Indication of which circuit corresponds to the truth table.

Work Plan

As stated at the beginning of this document we have been given a time frame of around 12-13 weeks to complete this project, which includes the Easter break and exam period, and we have 5 main aims for producing a software that meets the initial requirements, with this in my mind I will be implementing a agile/scrum methodology while developing where the 12 weeks will be split into 2 week sprint periods where I achieve one of the aims during each 2 week period. This will leave around 2 weeks which can be used for reviewing the project,

creating the final report, more time for any unseen circumstances that arise, external obligations such as exams and adding extra features/components to produce the ideal product.

Since each sprint will contain a two-week period at the end of this two-week period a meeting will be scheduled with my supervisor, referred to as story time in a scrum methodology or a scheduled review meeting, where we review what was achieved during the sprint, if the aim's acceptance criteria were met and prepare for the next sprint. This will allow me to receive feedback and review this feedback in preparation for moving forward; any outside feedback gained will also be reviewed. During each week of these two week sprints I will also arrange a short optional weekly meeting which I can use to discuss any problems I have or require any assistance/feedback with a particular component, since these meetings will be considered optional they will only be arranged if a particular problem arises that cannot be solved easily over email and require a meeting in person or a review and feedback on some component is needed. Finally, for each sprint I will be splitting the main objective into 3 different sub goals where the first goal will allow us to create the minimum viable product (defined as what we know we can produce in the time frame), a target product (defined as what I should be able to produce within the time frame) and an ideal product (defined as what we may be able to deliver within the time frame i.e. additional features), by doing this I can guarantee that the goals set can be met within the 2 week period to create the target product while also allowing the option to further develop the software if any time remains to create the ideal product. This means each week I will complete the minimum viable and target product goals then proceeding to the ideal product goals if time remains.

Below is the work plan layout for the sprints ahead, these sprints will be recorded on Trello Boards since this site allows me to create separate lists which will be labelled Backlog, To Do, In progress, Complete and Tested where each of these lists can be filled with our sub-goals for that sprint in order to achieve the main aim, these sprint boards can then also be shared with multiple users such as my supervisor if required. Backlog will be goals from the previous sprint that were not completed or unable to complete yet and take priority in the next sprint so that they do not create a further backlog; this should be empty most sprints. To do will be tasks that have not yet been started, in progress are tasks that are currently being worked on and complete are tasks I have completed. Finally, the tested will be tasks that are completed, and I have tested against the acceptance criteria to ensure they meet the expected standard.

Sprint 1: Software contains pre-defined logic gates that include at least NAND, NOR, NOT, AND, OR and XOR.

Date: 4th February 2019 – 17th February 2019

Minimum Viable Product	Target Product	Ideal Product
- List containing symbols for each of the logic gates.	- Logic gate symbols should be clear, easily accessible and labelled.	- Workspace should be created within the same window in preparation for the next sprint.

	- List containing the symbols should be scrollable/dropdown if all symbols do not fit in the window.	

Sprint 2: User interface allows for the logic gates to be dragged, dropped and connected inside a designated workspace within the interface. By allowing the logic gates to be easily dragged, dropped and connected we can offer an intuitive and simple interface for users that give them the freedom to control the components and produce a circuit of any form.

Date: 18th February 2019 – 3rd March 2019

Minimum Viable Product	Target Product	Ideal Product
- Window contains list of logic gates and a blank workspace where users will create the circuits.	- Users can drag, drop and connect the logic gates within the workspace. - Workspace can handle lots of components.	- Users can zoom in or out while also re-adjusting the size of the circuit and gates. - Help tab within the window which contains instructions on how to create a circuit and other resources.

Sprint 3: Constructed circuits can be stored and easily accessed/loaded into the workspace.

Date: 4th March 2019 – 17th March 2019

Minimum Viable Product	Target Product	Ideal Product
- Users should be able to store the constructed circuits.	- Users should be able to load their saved circuits and other circuits that are stored in a file on the system. - User can edit a circuit that is loaded into the workspace.	- Stored circuits should be stored in a readable file format such as csv etc. - When users load the circuits, all adjustments made such as resizing should persist. - A repository that can be accessed through the software that allows you to access other users circuits that are stored there.

Sprint 4: Truth table can be generated from a constructed circuit.

Date: 18th March 2019 – 31st March 2019

Minimum Viable Product	Target Product	Ideal Product
- User can generate a truth table from the circuit loaded within the workspace.	- Truth table should display in a separate window when generated. - Users should be given the option to save/export this truth table as a file. - Indication of which circuit the truth table was generated from within the window.	- Users can save/export the file into a chosen file type. - User can input a truth table and generate a random circuit that would fit this truth table.

Sprint 5: Final report, clear any remaining backlog, add features to create the ideal product within reasonable time, Remove bugs, Review Project

Date: 1st April 2019 – 10th May 2019

(This sprint date is longer than 2 weeks as it contains the Easter recess which runs from 13th April – 5th May, after this 5 days remain until the final report deadline so these are included in this sprint as there are not enough days to designate a sprint towards. These days will be used for clearing the backlog, writing up the report etc.)

Minimum Viable Product	Target Product	Ideal Product
- Create the final report ready for submission on 10th May 2019 - Clear any remaining backlog.	- Remove any remaining bugs - Review the project while improving some things I may not feel are up to standard of the acceptance criteria. - Review final report once it has been completed.	- Add features from previous sprints that may not have been possible or ideas that have developed along the way.