Software Engieneering Laboratory
section 2
Project: Logic circuits

Analysis and Requirements

1. Original task:

The aim of the project is to adapt the C++ code written for an obsolete platform. The aim of the implemented program was to simulate the logic circuits – it makes it possible to create the logic circuits using GUI, visualize its structure and computer the circuit's outcome.

The adaptation should make the code compatible and runnable in Visual Studio 2015 (the C++ language must be preserved). Please take into account the following issues:

- the program (and its GUI) work fine under MS DOS,
- the GUI (appearence), workflow and behaviour should not be modified in general,
- any intended modifictations should be discusses and approved by the Laboratory Leader,
- the code should be refactored in terms of separating the model from the view.

2. Code analysis

As first stage of code development code analysis was perfomed by entire team to find all problems with code. Then, when all issues were known, solution were proposed to fix those problems during code implementation stage of code development.

Documents of analysis containing detailed analyses of each file are kept on google drive as well as on Trello on tasks assigned to each team member. Here is a summary of problems to solve:

Analysis:

1. AND
    - 1.1. Minor issues
        - 1.1.1. Obsolete handling of null values
2. BUTTON
    - 2.1. Minor issues
        - 2.1.1. Obsolete syntax
    - 2.2. Obsolete graphics API
    - 2.3. Obolete cursor handling
3. CONSTBUS
    - 3.1. Obsolete graphics API

  14.2. Minor issues
   14.2.1. Obsolete syntax
15. SMALLWINDOW
  15.1. Minor issues
   15.1.1. Obsolete syntax
  15.2. Obsolete cursor handling
  15.3. Obsolete graphics API
16. WIRE
  16.1. Minor issues
   16.1.1. Obsolete syntax
  16.2. Obsolete graphics API
17. WINDOW
  17.1. Obsolete graphics API
  17.2. Obsolete cursor handling
18. XOR
  18.1. Minor issues
   18.1.1. Obsolete syntax

Problems are repetetive which means fewer solutions will be required but they will need to be program-wide except for minor issues which can be fixed extremely easily.

3. Code requirements

Based on code analysis following requirements must be fulfilled by code implementation stage:

Functional requirements:

1. Implement minor fixes – the code is littered with minor incompatibilies caused by changes between original compiler and Visual Studio 2015 as well as changes made to C++ languange syntax itself, as well as regular mistakes made by original authors. Those errors are minor and virtually all of them can be solved by one person during one sitting and if any are missed they can be fixed as they are found along the way.
2. Replace obsolete graphics API – program uses outdated technology of Borland Graphics Interface to function. This technology must be replaced by a newer one.
   2.1. No changes to GUI design itself.
   2.2. Migration from Borland Graphics Interface to Windows API.
3. Remove obsolete cursor handling – program has functions dedicated to handling cursor which are no longer required on modern OS. This functions can be entirely removed.
   3.1. Keep required functionality – during deletion due dilligence must be made to ensure no other functionality included in functions being removed is lost but instead is migrated to other places in the code.

Non-functional requirements:

1. Compatible with Visual Studio 2015.
2. Separation of model from the view.
3. Minimal changes made to code.
   - 3.1. If there are any errors in the code such as empty functions or unused variables in the code but which do not actually prevent compilation they will not be fixed.

Version 1.0 – 29.10.2017
Version 2.0 – 05.11.2017
Version 3.0 – 04.12.2017
Version 4.0 – 19.12.2017