|  |  |  |  |
| --- | --- | --- | --- |
| **CS102** | **Spring 2018/19** | Project Group | 1G |
| Instructor: | **David Davenport** |  |  |
| Assistant: | Gizem Çaylak |  |  |

|  |  |  |
| --- | --- | --- |
| **Criteria** | **TA/Grader** | **Instructor** |
| Presentation |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Overall |  |  |

~ SimCo ~

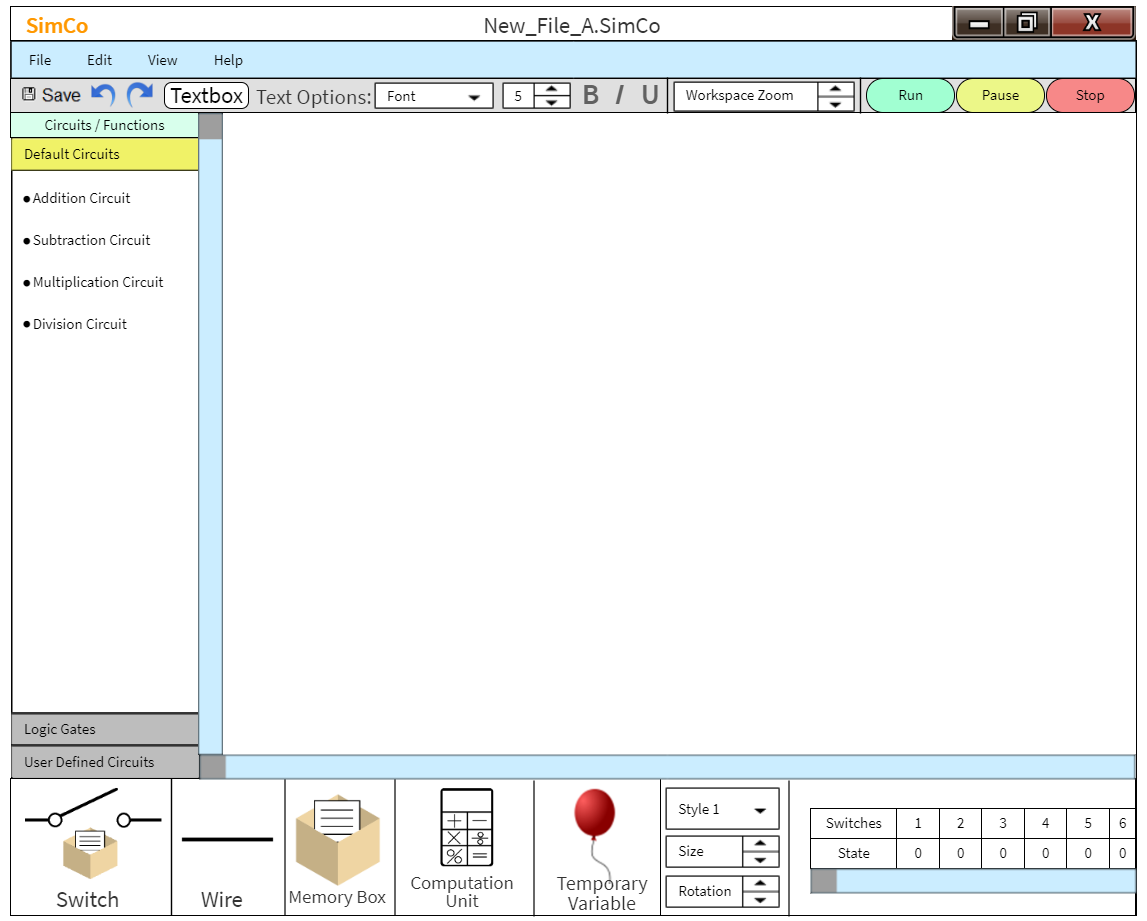
theConstructors

Mustafa Göktan Güdükbay, Musa Ege Ünalan, Doğa Tansel, Uğur Utku Seyfeli, Oğuz Orhun Tüzgen, Yiğit Harun

|  |
| --- |
| Detailed Design Report  ( version 1.0 )  16 April 2019 |

# Introduction

SimCO is an application designed for helping Computer Science instructors to teach the very basics of computer architecture more efficiently. It is a tool for building computer circuits to simplify, visualize and simulate the data flow in a computer. It will allow users to draw basic computer circuit diagrams on a blank canvas where they can add basic parts like wires, switches, memory locations and a computing tool for operations and functions from the menu to form the circuit and animate it.



# System Overview

SimCo is designed and will be implemented as a desktop application as it targets computer science department instructors and their students. It will be a useful tool to use in computer lab sessions and in lectures. SimCo is essentially a client-based application which works offline. The application is going to be built mainly using Java’s default libraries, the AWT and Swing packages and also the Serialization interface for save functionality.

# Core Design Details

* The full look of the UML diagram (Figure 1):

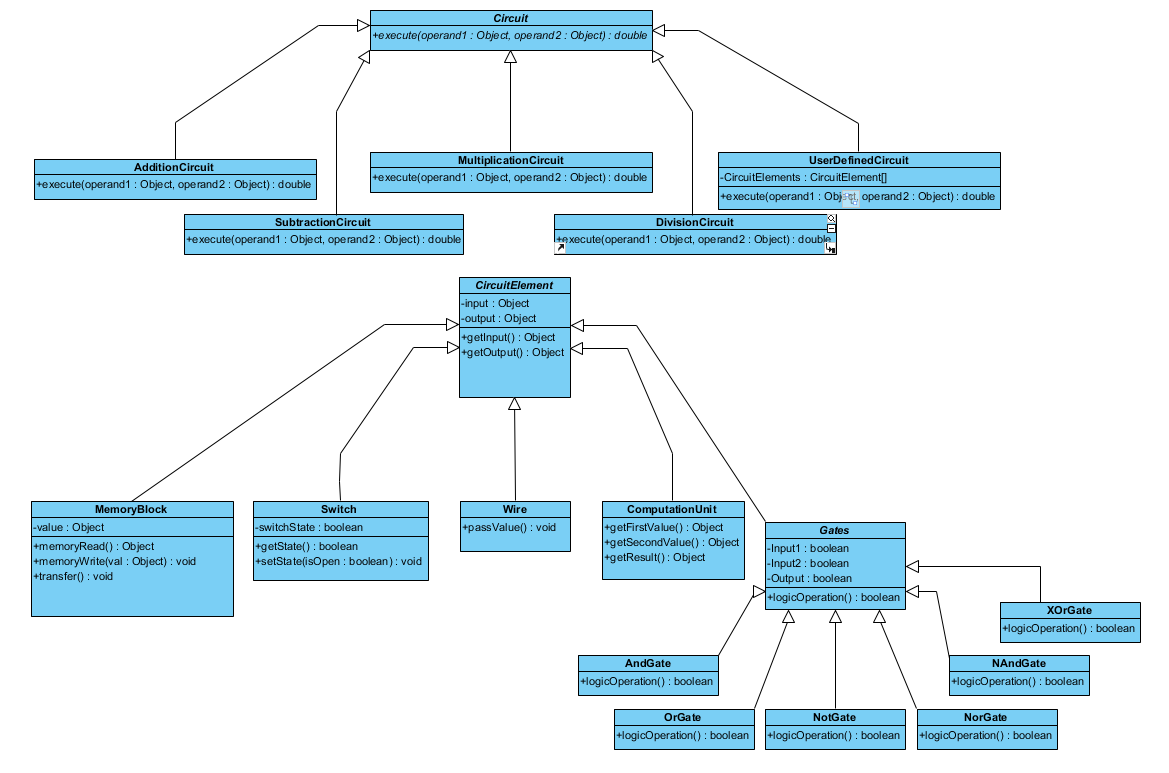


Figure 1

* The Circuit class (abstract) with the 4 default arithmetic circuit classes and the UserDefinedCircuit class, which allows users to save their workspace and reuse it in other workspaces. (Figure 2):

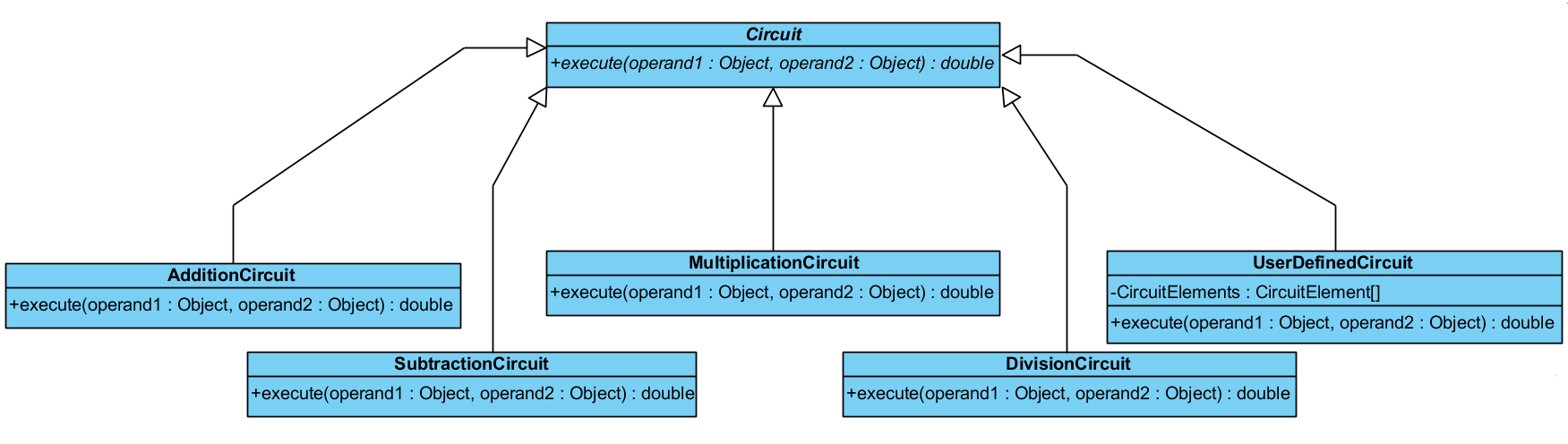


Figure 2

* The CircuitElement class (abstract) and its subclasses MemoryBlock, Switch, Wire, ComputationUnit and Gates (abstract). (Figure 3):

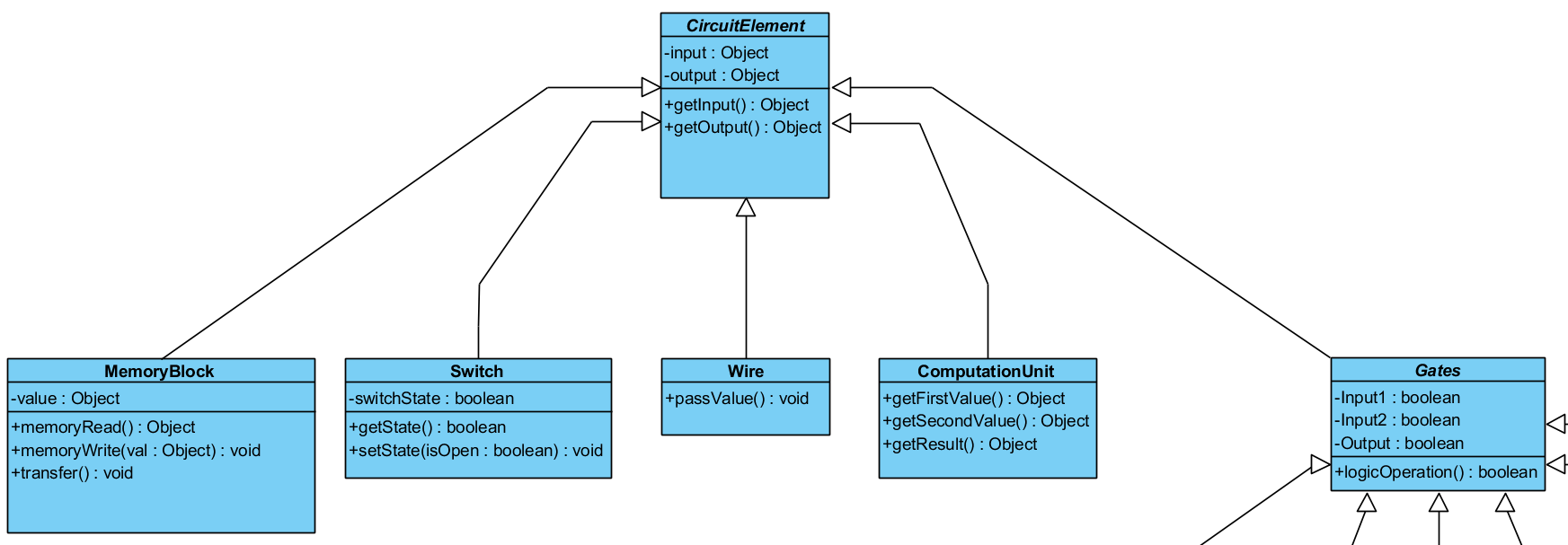


Figure 3

* The Gates class (abstract). (Figure 4):

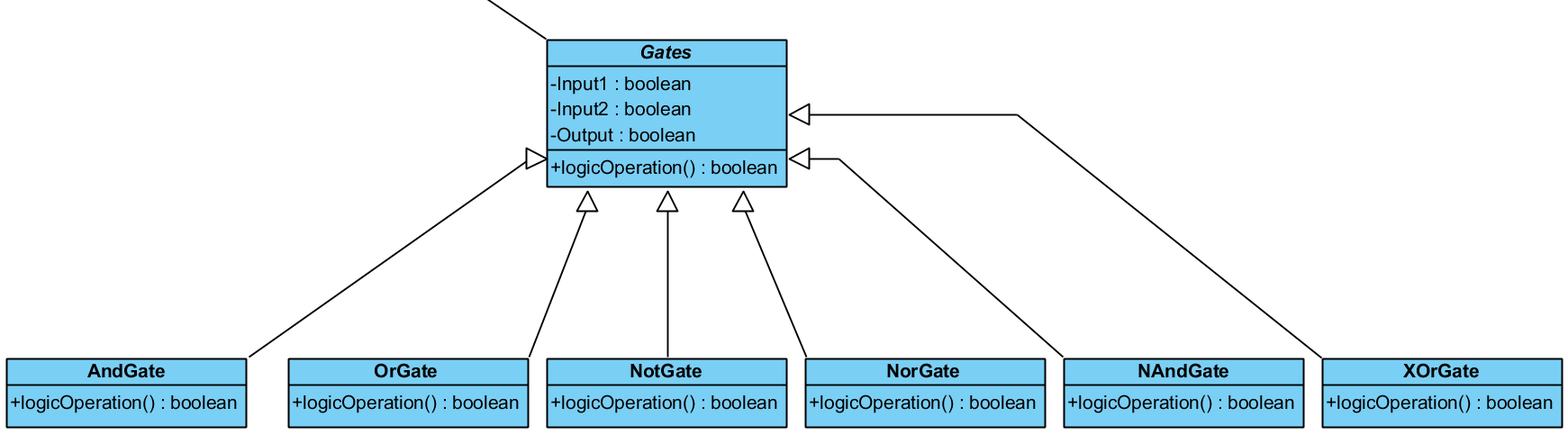


Figure 4

# Task Assignments

* **Utku:** Default Circuits (Add., Sub., Mult., Div.)
* **Goktan:** Circuit, CircuitElement
* **Ege:** UserDefinedCircuit, (Not, Xor) Gates
* **Doga:** MemoryBlock, Switch
* **Oguz:** (And, Or) Gates, Wire, Computation Unit
* **Yigit:** UserDefinedCircuit, (Nor, Nand) Gates