

# Assignment 4

COMP SCI 2ME3, SFWR ENG 2AA4

March 29, 2019

**Assigned:** March 22, 2019

**Spec and Code:** April 9, 2019

**Last Revised:** March 29, 2019

All submissions are made through git, using your own repo located at:

```
https://gitlab.cas.mcmaster.ca/se2aa4\_cs2me3\_assignments\_2019/\[macid\].git
```

where [macid] should be replaced with your actual macid. The time for all deadlines is 11:59 pm.

## 1 Introduction

The purpose of this assignment is to design and specify modules for playing Conway's Game of Life. The modules should cover the Model and View portions of the Model View Controller design pattern. You do not need to write a controller. Your unit tests will take the role of controller.

The rules for Conway's Game of Life can be found at:

Rules

An on-line simulator can be found at:

```
https://bitstorm.org/gameoflife/
```

Your assignment is for the module that stores the state of the game board and the status of the game. You also need a module that can display (view) the state of the game

board using text based (ASCII) graphics. You may also include other modules in your design.

Your specification should be written using LaTeX. All of your code should be written in C++. All code files, except for testing files, should be documented using doxygen. Your test code should use `catch`, version 2.

## Part 1

### Step 1

Submit your design specification, written in LaTeX, of the MIS for the game state module and view module. If your specification requires additional modules, you should include their MISes as well. It is up to you to determine your modules interface; that is, you decide on the exported constants, access programs, exceptions etc. You also determine your state variables and specify the semantics for your access program calls. Your design does not need to concern itself with performance.

The initial state of your game should be provided through a text file. It is up to you to determine the structure of this file. You also need the ability to output the game state to a file after any iteration. The output file should be in the format of the input file, so that a new simulation can be started using the output as initial conditions.

Your design specification should include an overview/critique of your own design. Please be specific. In particular, you should self-assess how your design performs with respect to all of the following qualities: consistency, essentiality, generality, minimality, cohesion and information hiding. You also should use comments in your MIS to explain your intentions. Your goal is to quickly communicate to the marking TA your design decisions, so that they can make a fair assessment.

## Part 2

### Step 2

Submit C++ code that matches the specification given in the previous step. You should also submit code that tests your module(s) using `catch 2`. Document your source code using doxygen. Your code should include a makefile, with rules `make test` and `make doc`. Performance will not be considered in the grading.

## Notes

1. Your git repo is organized with the following directories at the top level: **A1**, **A2**, **A3**, and **A4**. Your specification and code files should be placed in the **A4** folder
2. Please put your name and macid at the top of each of your source files.
3. Your program must work in the ITB labs on mills, as specified for A3.
4. **Any changes to the assignment specification will be announced in class. It is your responsibility to be aware of these changes. Please monitor all pushes to the course git repo.**