# Project Overview

**Taoran Liu, ID: 704605101**

**Gladys Ng, ID: 304643416**

**Qingwei Lan, ID: 404458762**

For our project, we will create a program that plays Conway's game of life on a 16x16 grid, powered by our nexys3 FPGA and displayed through a VGA Cable. The grid will be a flat representation of a toroid, and, at the start of the program, users will be able to set the initial state of the board using the directional keys and the center button in a `set_intial_state` mode. The user will then be able to start and pause the cellular automata, as well as clear the board and reenter the `set_initial_state` mode.

# Modules and Breakdown

**Clock Divider (10%)**

We will need a clock divider to set time intervals for sending data to the VGA output, as well as changing states. We will have a quick clock divider for the VGA output, and a slow one (2-5Hz) for updating the board state.

**Debouncer (10%)**

Since there will be a large amount of physical input to this device, including both buttons and switches, we will include a debouncing

module to ensure that the input get read in properly. We will use debouncer for each of the directional buttons and the center button, as well as one of the switches, which we will use to pause and play the simulation.

**Board Manager (35%)**

This module will maintain a 256 bit array representing the 16x16 grid of the simulation. It will count how many neighbors (out of eight) each cell has and change the cell's next value accordingly. The rules for Conway's Game of Life are as follows:

- If a live cell has 2 or 3 neighbors, then it stays alive in the next generation.
- If a live cell has 4 or more neighbors, then it dies in the next generation, as if by overpopulation.
- If a live cell has fewer than 2 neighbors, then it dies in the next generation, as if by underpopulation.
- If a dead cell has exactly 3 neighbors, then it becomes a live cell in the next generation, as if by reproduction. The board manager will have reset functionality to clear the board and make all living cells dead.

**VGA Controller (20%)**

This module will send the data for the next generation, generated by the board manager, to the VGA cable in the correct format. It will display each cell as a black 30px by 30px square on a 480px by 480px grid.

**set_initial_state Module (20%)**

This module will define the behavior of the `set_initial_state` mode of the program. The user will be presented with a red blinking cursor which

can be moved by the directional keys on the FPGA. When the center button is pressed, the cell highlighted by the cursor will become a live cell (the simulation will be paused to ensure that the cell does not immediately die of underpopulation). This module will also handle pausing and playing the simulation.

**Top Module (5%)**

This module will handle input and output from the board and call other modules to manage high level functionality.