UPPSALA
UNIVERSITET

# Deterministic AI

One definition for deterministic AI is a system where the same input sequence, starting from the same state, always results in the same output sequence. In computer games, the Finite State Machine (FSM) constitutes the most common method for the implementation of AI systems. Cellular automata, such as Conway's Game of Life are as a rule nothing but FSMs coordinating as autonomous agents to produce complex results out of a set of succint rules, not unlike how the laws of physics work on a fundamental level in our universe.

## A1.1: Security System (G) – *Lecture 1*

Complete the security system in F1 (the first lecture), Example 1, so that it becomes fully reversible:

- Establish a transition from $s_6$ (the former "End" state) to $s_2$ about 30 s after the security code is entered once again (this time to activate the alarm system), by using new intermediary state(s) between $s_6$ and $s_2$. Assume that the timer always stops after 30 s when triggered, and that when the right password has been entered, the signal "Authorized Access" is only active during one clock cycle.

Tips:

- Please note that no output-warning signal is expected in s7 (or s8 if applicable). In the PDF form, an output (red field), such as UA, can be set to zero by erasing all field content.
- Since in our system, the output signals of each state are only dependent on the state itself (and not the input signals), make sure that the outputs lines in pages 2–3 of the form are identical for each state. As an example for state 6 in the truth table, if U1 = 0, U2 = 0, and U3 = 1 for line A (according to

UPPSALA UNIVERSITET

the template), then U1–U3 must be set to the same values (U1 = U2 = 0, U3 = 1) on line B.

This assignment is carried out by the completion of a PDF form:

- Download the Adobe Acrobat Reader DC application from https://get.adobe.com/se/reader/otherversions/ or equivalent page and install this application on your computer, if you do not already have a relatively recent version of the software. Else, you could alternatively use PDF readers, such as Google Chrome, or Preview (macOS) to complete the PDF.

- Complete the state diagram, the truth table, and the code for this system in A11_Alt1.pdf *or* A11_Alt2.pdf (in AT.zip, available for download from the course homepage), depending on which form that according to your view gives the best solution. One form gives a succinct solution, while the other is easier to read.

- Save the PDF-form on your hard disk from the Adobe Acrobat Reader and upload this form to the course homepage under "A1.1 – Finite State Machines" within the module "Assignment 1 – Deterministic AI".

*Note: do not include any comments by the submission page or in the submitted files in assignments 1 and 2 that are expected to be read, since the assignment 1–2 submissions are machine graded. However, if you have any questions or comments, do not hesitate to email the course examiner.*

UPPSALA
UNIVERSITET

# A1.2: The Game of Life (VG) – *Lecture 2*

Implement Conway's Game of Life by the completion of the core logic function:

void Game::UpdateFrontBuffer()

in A1/GameOfLife from AT.zip.

The GameOfLife package contains two graphics templates (GLUT and SDL2) and one purely console-based (perhaps simplest to use if you have lack of time and have never used GLUT or SDL before).

To set an appropriate difficulty level for this assignment, only the following information is released about the code:

### 1. Logic

The rules defining the Game of Life are usually expressed from a biological point of view, but for clarification, the following is valid for 2–3 neighbors:

2 neighbors: the cell maintains current value (0->0, 1->1)
3 neighbors: the cell is set to one (0->1, 1->1)

### 2. Matrix Buffer

The grid size of the Game of Life is 25 (rows) x 50 (columns) and the information of each field is saved in the 3D matrix:

mM[2][25][50]

where mFrontBufferIdx in mM[mFrontBufferIdx][m][n], denotes the index (0 or 1) to the current buffer that is shown in that graphics window (vs printed in the console window).

This means that if mFrontBufferIdx is equal to 0, then the front (current) buffer consists of all the information that is located in the 3D matrix mM[0][...][...], and the previous state is found in:

mM[1-mFrontBufferIdx][...][...], i.e., mM[1][...][...], and vice versa.

UPPSALA
UNIVERSITET

## 3. Matrix Allocation

The reason why this matrix is dynamically allocated by the declaration

```
bool (*mM)[GRID_SIZE_Y][GRID_SIZE_X];
```

in Game.h and

```
mM = new bool[2][GRID_SIZE_Y][GRID_SIZE_X];
```

in Game.cpp, is since allocations of large arrays may cause C++ programs to crash on some systems by the creation of memory block overlaps, thereby creating bugs that are difficult to catch. By this method, we may allocate huge arrays in C++ without any glitch.

## 4. Keywords

As a challenge, but also partly to avoid the executed code to fail compilation or crash during testing due to compiler differences, mainly the following C++ keywords are permitted to be used (unless they are commented out) in the submitted method:

unsigned, short, int, float, double, char, bool, true, false, if, else, break, continue, switch, case, default, for, while, do, return, const, enum, void, nullptr, sizeof.

Since printing functions/methods such as printf or cout are not included in the above list, if they are used in the method to be tested during development, they could be commented out before the cpp-file is submitted instead of erased, since the contents in C style and C++ style comments, i.e., // vs /*...*/, are blanked out before compilation.

UPPSALA
UNIVERSITET

### 5. Submission

Upload the file Game.cpp to “A1.2 – Game of Life (VG)”, containing:

```
void Game::UpdateFrontBuffer(){
   //Solution...
}
```

The method UpdateFrontBuffer() in Game.cpp is extracted from the cpp-file before compilation and unit-testing (only this method is extracted) so no need to erase any other methods from the file. The maximum recommended size for submission in A1.2 is 100 kB, but anything below 10 kB is considered to be ideal.

Good Luck!