— слайд 1 **Hello!**

Hello everyone, my name is Daniel and my teammate is Bodgan.

Now we will present our project.

We have chosen and made a “game of life” using Logisim and CdM-8.

— слайд 2 **Table of contents:**

Now you can see table of contents of our project

— слайд 3 **Introduction:**

The Game of Life is a zero-player game.

— слайд 4 **Standard rules:**

In the slide you can see standard rules:

1. Neighbours[нэйбэс] are nearby cells (like a king in chess).
2. 3 Neighbours[нэйбэс] — cell can be born.
3. 2 or 3 Neighbours[нэйбэс] — cell continues to live.
4. Otherwise, the cell dies.

— слайд 5 **Problem Statement:**

— слайд 6 **Subtasks:**

First of all, we have made a work plan and divided the project into subtasks.

— слайд 7

Our project has two parts: Hardware in Logisim and Software (assembler in CdM8).

The Logisim part of the project will be responsible for main functions of the game: Display, System of generation cells, Rules changer.

The CdM8 part of the project will be responsible for the game management.

**========================================================================**

— слайд 20 **Changes** from basic task:

We have made a rules changer for the game. So user can change rules in the Logisim scheme.

— слайд 21 **Improvements** in our project:

We have made a keyboard[кИбоод] for game management.

We have made a cursor[кёрсе] for filling field.

Also we have made a torus. Torus is responsible for the field is collapsed or not, for example, like in the snake game.

— слайд 22 **Software:**

Software is assembly code.

— слайд 23 **About software** (memory and code screenshoots)

We have made it for the CdM8 processor[прэсэсэ] with Von Neumann architecture[Акитэкчэ], by limiting the code size to 240 bytes. The most difficult part was compressing the code, since initially it was 2 times larger than the available one. Also we have difficulties when debugging the program in Logisim: we have found that Mark5 has problems with some branches, so we have switched to the Mark4.

— слайд 24 **Flowchart:**

Our assembly code is responsible for the game management.

In the screenshot you can see the flowchart of our code.

— слайд 25 **Conclusion:**

The end result of our project was the realization of the «Game of Life». In the course of our work, we fixed some bugs and have looked for the best ways to solve problems to optimize the game. Teamwork was useful for soft skills.

Ultimately[алтиметли], we have improved our programming and communication skills.

— слайд 8 **Hardware:**

Let's start with the CDM8+ scheme.

— слайд 9 **Processor:**

First of all, we added a processor with memory for future work, as in the Coco IDE.

— слайд 10-11 **Reading or writing to RAM and external**

Next, we needed a subscheme for reading or writing to RAM (random access memory) and external devices (keyboard[кИбоод] and matrix).

— слайд 12 **Input-output CDM8+**

We receive[рисИв] the keyboard[кИбоод] data, process[прэсэс] it and passes the new state to the matrix.

— слайд 13 **KeyBoard scheme:**

Next important part of our project is the keyboard[кИбоод] scheme. Keyboard[кИбоод]’s logic is written in assembler code for the CdM8, so the keyboard[кИбоод] is connected to the processor[прэсэсэ].

— слайд 14 **KeyBoard in main scheme:**

Using the keyboard, we can control the cursor (up, right, down and left arrows), set a pause, clear the field, set the state of the cell (invert, dead or alive), turn on or off the torus.

If the green LED is on, the processor reads data from the keyboard. This happens only 1 tick.

— слайд 15 **Neighbours scheme:**

The Neighbours[нэйбэс] scheme receives[рисИвс] data on the states of all neighboring[нэйбэрин] cells, converts[канвётс] it into a single number and passes a 25-bit number to the alive scheme.

— слайд 16 **Alive scheme:**

The Alive scheme gets neighbors and look at, birth, save rules. It is responsible for whether the cell will live or not in the next iteration.

— слайд 17 **Line scheme:**

After we get the new state of the cell, we will pass it to the line scheme. In this circuit, we use a single register that stores the values of all cells in a row. Then we pass the register value to the matrix scheme.

— слайд 18 **Matrix scheme:**

Matrix scheme is responsible for working a display in our game. 16 lines from the Line scheme turn into a matrix. We also added 3 presets to this scheme (you could see 1 preset on the “Rules” slide).

I give the floor[не пол, а слово] to a colleague[кОлиг]. *[Передаю слово коллеге]*

**========================================================================**

— слайд 26 **Thanks!**

**Thank** you for your attention. Please ask your questions.