|  | **English** | **Русский** |
| --- | --- | --- |
| Слайд **1**  Настя | The theme of our project is “The game of Life”. In previous years, there was no such theme, so we became interested in implementing this game. In addition, the documentation for the project is in Russian)  The rules of the game are well known, so we will not repeat them.  Before starting the game, it is worth knowing that the player does not actively participate. He only creates an initial configuration of "live" cells, which are then changed according to the rules. Despite the simplicity of the rules, there can be a huge variety of forms in the game. | Тема нашего проекта - игра “Жизнь”. В прошлые годы не было такой темы, поэтому нам стало интересно реализовать эту игру. К тому же, документация к проекту дана на русском)  Правила игры общеизвестны, поэтому повторять мы их не будем.  Перед началом игры стоит знать, что игрок не принимает активного участия. Он лишь создаёт начальную конфигурацию «живых» клеток, которые затем изменяются согласно правилам. Несмотря на простоту правил, в игре может возникать огромное разнообразие форм. |
| Слайд **3**  Настя | In order to successfully complete the project, we have broken down our path into steps:   1. think through a plan of action and the steps of doing the work 2. do the tasks step by step 3. write documentation, speeches and presentations 4. evaluate the work of the team members | Для успешного выполнения проекта мы разбили наш путь на шаги:   1. продумать план действий и этапы выполнения работы 2. выполнять поставленные задачи постепенно 3. написать документацию, текст выступления и сделать презентацию 4. оценить работу участников команды |
| Слайд **4**  Настя | We have succeeded in implementing the game. In the beginning, the player sees the controls and a 16x16 field. The edges of the field are connected to each other, as if "stitched".  There are two circuits connected to the matrix, which are responsible for the memory and the buffer. They are used to write to the cells and output to the matrix. The circuit closest to the matrix is responsible for red blinking of the selected cell. You can select a cell with the help of the navigation buttons on the right. Nearby there is a button, after pressing which the game will be started. | У нас получилось реализовать игру. В самом начале игрок видит элементы управления и поле размером 16х16. Края поля связаны между собой, как бы “сшиты”.  С матрицей связаны 2 схемы, которые отвечают за память и буфер. С их помощью происходит запись в ячейки и вывод на матрицу. Самая ближняя схема к матрице отвечает за мигание красным выбранной ячейки. Выбрать ячейку можно с помощью, кнопок навигации, расположенных справа. Рядом находится кнопка, после нажатия которой будет запущена игра. |
| Слайд **5**  Настя | This slide shows an implementation of CDM. When an attempt is made to write something to RAM, the hardware device intercepts communication with the CDM and RAM. Then certain cell addresses are assigned to the memory and as soon as the address on the address bus matches the address that is defined for a particular device, it appears as a memory cell to the CDM. | На данном слайде представлена реализация CDM. Когда происходит попытка записи чего-либо на оперативную память, то хардовое устройство перехватывает общение с CDM и оперативной памятью. Затем присваиваются определенные адреса ячеек в памяти и, как только адрес на шине адреса совпадает с адресом, который определён для конкретного устройства, то оно представляется ячейкой памяти для CDM. |
| Слайд **6**  Настя | In memory, the values are stored by coordinates, which we set manually at the beginning. For example, let's choose the initial coordinates. Let it be (0;0).  In “Data” comes the value to be written into the memory cell. Record is an input on which the cycles come. On the rising cycle the value is written. And “Return” receives the value that is currently in the cell. | В памяти значения хранятся по координатам, которые мы на начальном этапе задаем вручную. Для примера, выберем начальные координаты. Пусть это будут (0;0)  В Data приходит значение которое надо записать в ячейку памяти. Запись это вход на который приходят такты. На повышающем такте записывается значение. А в return поступает значение которое сейчас находится в ячейке. |
| Слайд **7**  Данила | On the left is the assignment of memory cells to variable labels. The first four lines contain the memory cells to which the hard devices are mapped.  On the right is the code that iterates through the cells. In the first iteration, X and Y are zeroed out, and in subsequent iterations, only X is entered.  The coordinates of the current cell are also entered into the X and Y variables and the state of the current cell is entered into "sost". The state of the cell means whether the cell is alive or dead. | Слева реализована присваивание ячейкам памяти меток для переменных. Первые четыре строки содержат те ячейки памяти, к которым привязаны хардовые устройства.  Справа показан код, который перебирает клетки. В первой итерации зануляются X и Y, а в последующих только X.  Также заносятся координаты текущей клетки в переменные X и Y и заносится состояние текущей клетки в “sost”. Состояние клетки означает живая клетка или мёртвая. |
| Слайд **8**  Данила | Here is implemented the enumeration of neighbors of a cell. The coordinates of neighbors are found, and the edges are checked. For example, if the Y coordinate is 15, then the coordinate one more will be 0. This is how the "stitching" of field edges is implemented. | Здесь реализован перебор соседей клетки. Находятся координаты соседей, при этом проверяются края. Например если координата Y равна 15, то координата на один больше будет равна 0. Так реализовано “сшивание” краёв поля. |
| Слайд **9**  Данила | The left parts of the code check the states of the neighbors of a live cell. If the neighbor is "alive", the variable sums is incremented.  On the right is an extreme optimization of the code so that it fits into memory. | Слева части кода проверяют состояния соседей живой клетки. Если сосед “живой”, то переменная sums инкрементируется.  Справа представлена экстремальная оптимизация кода, чтобы он влез в память. |
| Слайд **10**  Данила | The first screenshot shows the "birth" of a cell. This happens if the number of neighbors and the current state match the rules of cell life.  The second screenshot shows the transition to the next iteration of the cell enumeration. If all cells have already been enumerated, a signal is sent to change the fields and clear the field for recording. Then the enumeration starts over again. | На первом скриншоте представлено “рождение” клетки. Это происходит, если количество соседей и текущее состояние соответствуют правилам жизни клетки.  На втором скриншоте показан переход к следующей итерации перебора клеток. Если все клетки уже перебраны, то отправляется сигнал для смены полей и очистки поля для записи. Затем начинается перебор сначала. |
| Слайд **11**  Настя | To check the correctness of our game, we played it ourselves.  On the left is one of the "everlasting" configurations. It is a glider. Its peculiarity is that it changes only 3 steps, and on step 4 the configuration of the first is repeated, but it is shifted in coordinates. The glider will move across the field and the cells will always be alive.  On the right is the configuration we made up. It has funny shapes at each step. But, unfortunately, the cells will eventually die, although the configurations are fun.  It was interesting for us to do this project.  Firstly, such a theme appeared only this year. Secondly, creating our own game helped us understand how everything works from the inside.  It's an invaluable experience that will remain forever. | Чтобы проверить корректность работы нашей игры, мы поиграли сами.  Слева представлена одна из “вечно живущих” конфигураций. Это планер. Его особенность в том, что он меняется всего 3 эпохи, а на 4 эпохе повторяется конфигурация первой, но по координатам она сдвинута. Планер будет двигаться по полю и клетки всегда будут живыми.  Справа показана нами придуманная конфигурация. У неё забавные формы на каждом шаге. Но, к сожалению, в конечно итоге клетки умрут, хотя конфигурации получаются прикольными.  Нам было интересно выполнять этот проект.  Во-первых, такая тема появилась только в этом году. Во-вторых, создание собственной игры помогло нам понять, как всё работает изнутри.  Это бесценный опыт, который останется навсегда. |
|
| Слайд **13**  Настя | Thank you for your attention, we are ready to answer your questions! | Спасибо за внимание, готовы ответить на ваши вопросы! |