Вова

вступление

1. Good morning, colleagues and audience;
2. We are 1st year students of Deep Robotics Institute;
3. There are my teammates: Vladislav, Arina, Vladimir.
4. We are glad to present our project “Snake”.

Let's begin!

аналоги

It first appeared in 1976 on an arcade machine.
In 1995, Taneli Armanto came to work at Nokia and wrote a game for the Nokia 6110.

игровой процесс

In the snake game you control a snake and you have to collect food to make it grow longer. You need to avoid collisions with the tail of the snake, otherwise you will lose. You can crawl behind the walls and appear on the other side.

Влад

3. There is UI, it consists of thirty by thirty gameplay display with additional colored thirty two by thirty two matrix for apples. Keyboard is used to control the snake movement via WASD keys. However, there are two buttons to control the game process itself like pause and start, score counter and max score counter.

Let’s move on deeper.

4. So, here you can see game process controllers and output circuits.
5. Next, there is software-to-hardware communication:

Cdm processor
Central circuit loads data to processor.
On the bottom right we have the circuit that takes data from processor
And above it we have an I/O manger.

6. There are snake output circuits: pixel and line managers, line consists of pixels,
7. Next you can see output of the snake to the display itself it consists of lines
8. Head and Tail movement is based on coordinates,

This is head movement manager

9. And here is tail movement manager
10. There is random generation of apples position and if an apple spawns inside the snake it generates again.
11. Finally, you can see input and output blocks of previous circuit.

Now, let me show you gameplay.

ANSWERS:

We considered this version of Cdm too complicated for our project.
Excuse me, could you please repeat your question a bit slower/louder.

Арина
in this block We initialize the variables and wait for the game to start.
Here we save a segment if necessary, but we only save segments of non-zero length.
and in this block the program unloads the segments from memory in queue order.
And now let us show you gameplay

in this blok we initialize the variables and wait to the game to start
here we save a segments if necessary but we only save segments of non-zero length
and in this blok the program unloads the segments from memory in queu order
and now let us show you gameplay