

FP7 - Build your own Computer in Minecraft

Christian Hakert, Junjie Shi

Department of Computer Science, Chair 12
TU Dortmund University, Germany

April 14, 2021

Build your own Computer in Minecraft

Minecraft at University?!?

- Minecraft, a popular computer game, implements boolean logic
- AND/OR/NOT/NAND/NOR gates easy to realize
- game environment / creativity (not only functional)

Build your own Computer in Minecraft

Minecraft at University?!?

- Minecraft, a popular computer game, implements boolean logic
- AND/OR/NOT/NAND/NOR gates easy to realize
- game environment / creativity (not only functional)

But why not just use CAD and draw a circuit?

- This FP should mainly reveal if gamified logic circuit building is possible (not to replace CAD)
- Multiplayer / Online games allow competitions
- Looks a bit more fancy :D

Build your own Computer in Minecraft

Minecraft at University?!?

BUT DO WE REALLY BUILD A PC?

- Minecraft, a popular computer game, implements boolean logic
- AND/OR/NOT/NAND/NOR gates easy to realize
- game environment / creativity (not only functional)
- of course not, we start and keep it simple
- some logic gates / simple circuits
- probably a bit more complex circuits
- simple clocks and automata
- probably some programmable parameters
- Multiplayer / Online games allow competitions
- Looks a bit more fancy :D

Minecraft logic in a nutshell

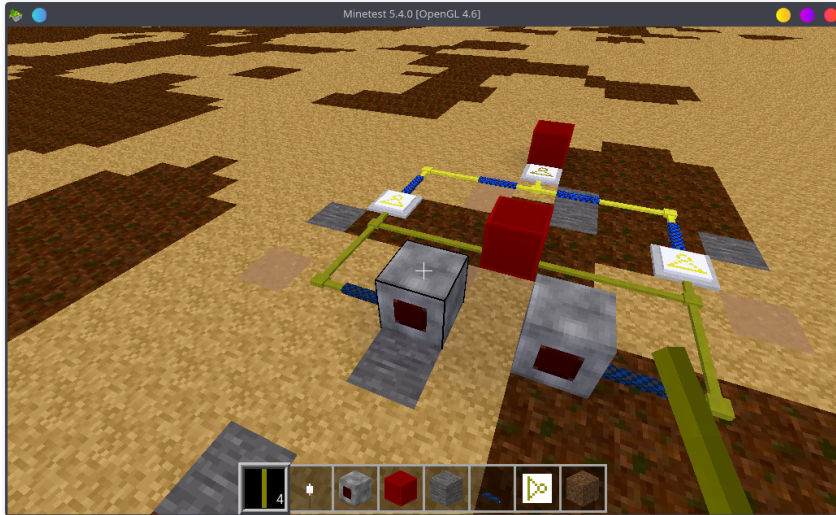
Signal level electricity is implemented

- Binary states, either off or on
- Signals can be passed through wires
- Signals can be generated (e.g. by a switch) and used (e.g. to light up a lamp)

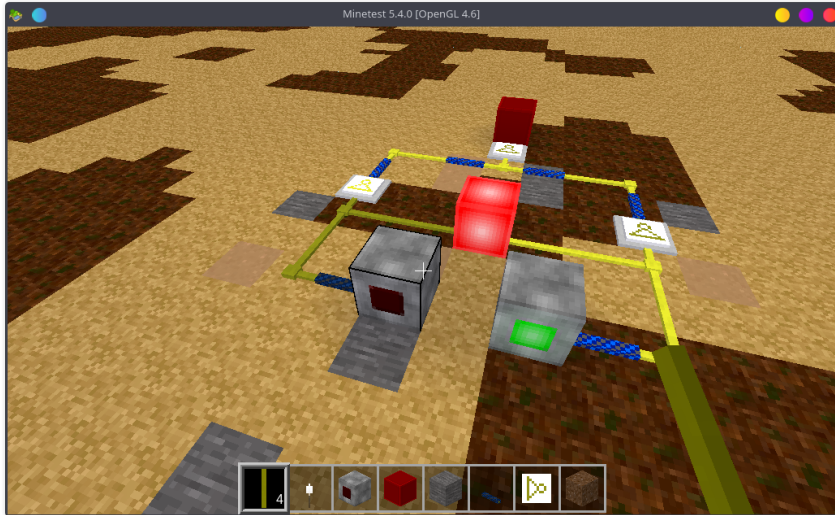
Signal processing

- Wires can be joined (logic or)
- Signals can be inverted (logic not)
- That's sufficient ;) $[a \wedge b = \overline{\overline{a} \vee \overline{b}}]$

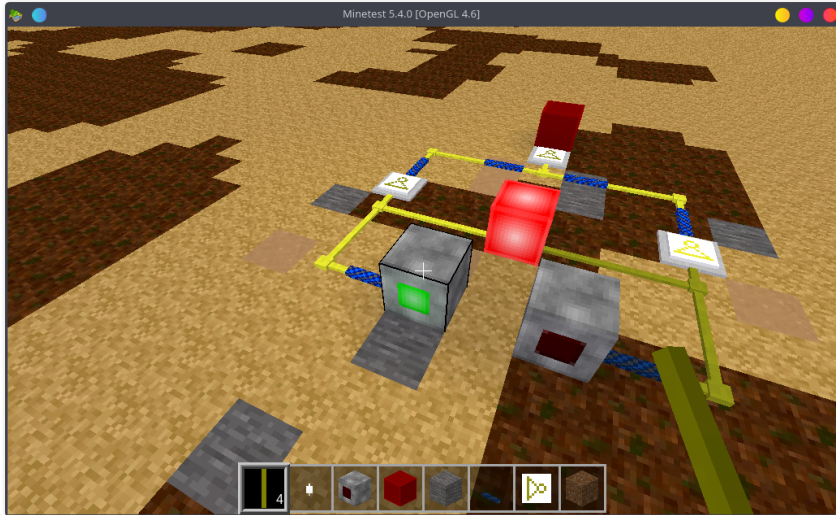
Eyes on



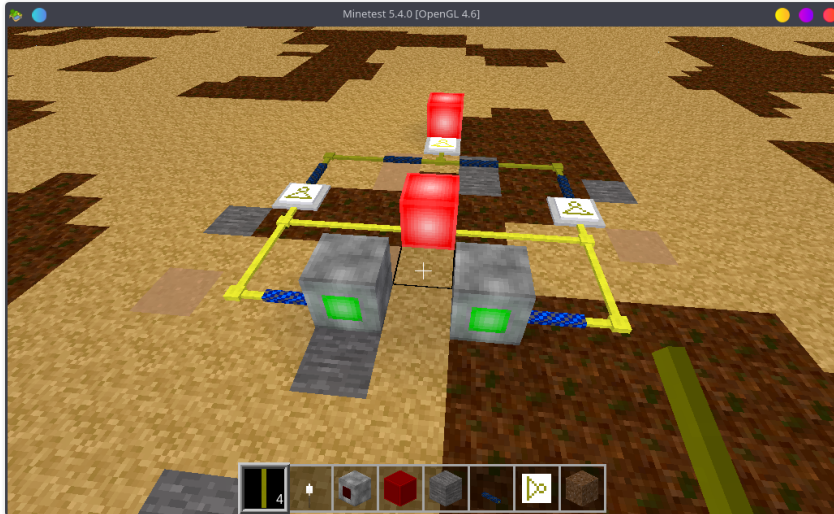
Eyes on



Eyes on



Eyes on



Roadmap for the Fachproject

- ① Seminar Phase to get familiar with the topics
- ② Setting up environments (network, servers, ...)
- ③ Get familiar with simple examples
- ④ Plan bigger circuits in the group
- ⑤ Implement circuits
- ⑥ Provide documentation, report up and downsides

Roadmap for the Fachproject

- ① Seminar Phase to get familiar with the topics
- ② Setting up environments (network, servers, ...)
- ③ Get familiar with simple examples
- ④ Plan bigger circuits in the group
- ⑤ Implement circuits
- ⑥ Provide documentation, report up and downsides

Overall:

- (Bi-)weekly meetings to synchronize
- Prepare a written report (10-15 pages), assess the allover concept of implementing logic in minecraft
- Give a final presentation with demo

Roadmap for the Fachproject (cont.)

- We won't use minecraft, due to licensing
- Minetest is a pretty cool alternative
- Mesecons is a plugin for logic signals

Roadmap for the Fachproject (cont.)

- We won't use minecraft, due to licensing
- Minetest is a pretty cool alternative
- Mesecons is a plugin for logic signals

Seminar topics:

- ① Minetest: Installation, usgae, hidden features
- ② Mesecons: Logic wires, blocks, limitations, pitfalls
- ③ Maps: Exporting and importing maps, replicating structures
- ④ Networking: Multiplayer online gaming, collaborative building
- ⑤ Logics: Recap of boolean algebra, simple circuits (adders, ...)
- ⑥ Logics: Complex static structures, multiplexers, memory, clocking
- ⑦ Logics: Programmable logic, automata, simple processors