

Patrick devine a0101165

Systems and game tech ICA2: hardware report



January 12, 2022

Teesside university

GAV1063-N

Contents

[Description of the aspect of low-level computing that you learned about and implemented, which you found the most interesting or useful 2](#_Toc92796902)

[Discussion of one aspect of computer hardware or structure covered that you thought would help most with your C++ programming 3](#_Toc92796903)

[Conclusion 4](#_Toc92796904)

# Description of the aspect of low-level computing that you learned about and implemented, which you found the most interesting or useful

Implementing the key press events was the most interesting and useful aspect of low-level computing I learned about and implemented. The reason it was useful to implement the key presses was that I had no idea before hand how to go about receiving key events from the keyboard in a C++ environment, so learning how to do so will undoubtedly help me develop my programming skills in the future and give me more options when developing a game from a low-level language such as C++. It was also interesting as the opcode that awaited key presses was probably the most complex to decode (of the opcodes I understood perfectly) and I had to think about what to do rather than just set a register to another register or simply check for if a register was empty and skip the next instruction which is quite simple. It was also interesting to see how the keypad in the emulator translated to the keyboard in which 0-F couldn’t just be bound to 1-15 as there is no key above 9, so I needed to apply the key presses from Q-V as well and bind them to the fifteen hexadecimal values for the keypad. In conclusion, the most interesting and useful aspect of developing the emulator for me was the key presses and the keypad, I feel as if it has given me a valuable experience and knowledge that I will apply in future modules and games.

# Discussion of one aspect of computer hardware or structure covered that you thought would help most with your C++ programming

One aspect of computer structure that I feel would help most with my C++ programming would be emulating the fetch/decode/execute cycle. The reason for this is that every programming language (python to machine code regardless of level) is ran line by line, this is not by choice however and due to the way the CPU works in which each instruction is fetched, decoded and then executed on the CPU. The CPU gets addresses from the RAM or in-built cache memory. C++ is relatively low level when it comes to programming languages but regardless of the level, an understanding of how the fetch/decode/execute cycle works is extremely useful to a programmer. It allows for more freedom when programming and gives more tools as a result when making a game or application in which the sequence of instructions is apparent. In addition, the fetch/decode/execute cycle works with registers such as: The PC, a counter that stores the address of the next instruction to be executed; The MAR, the memory address register which stores the current address being fetched; The ALU, the arithmetic logic unit which handles logical and arithmetic calculations (example of arithmetic operation: addition, example of logical operation: Greater than); The CU, the control unit which sends read and write signals to the RAM allowing the CPU to fetch instructions and then execute those instructions which (depending on the instruction) could change the values stored in the memory.

# Conclusion

In conclusion, I feel as though I learned a lot about programming by creating this emulator. Though most features may be functional, there were a few implementations I could not program this is due to a lack of experience with programming graphics in C++ and I need to reflect on what I could and couldn’t do in the development of this chip 8 emulator. In the future I will continue to research into low level computing and try to implement graphics into my emulator or future projects involving C++. To conclude my report, the process of making this emulator provided me with valuable experience and knowledge and although I could not accomplish everything I set out to do, I feel as though I have learned a lot of useful information that I can apply to future modules and games/applications when developing in C++.