Chip8 Emulator in C

Kirjan Kohuladas/kprime21

Overview

Compact Hexadecimal Interpretive Programming – 8-bit

Components

- Memory - 4kB ram (4096 bytes - 4096 address lines each line is 1 byte)
- Display - 64 x 32 pixels
- Registers
  - Program Counter (16 bits)
  - Index Register (16 bits)
  - Stack - call subroutines and functions (16 bits)
  - Delay timer - decremented at 60Hz (8 bits)
  - Sound timer - decremented at 60Hz (8 bits)
  - 15 General purpose registers - V0 - VF (8 bits)

Memory

- all memory is RAM, 4096 bytes.
  - 4096 addressable lines
  - 12 bits needed
  - each addressable line represents an addresss of 1 byte.
- interpreter located 0x000-0x1FF (not in our case)
- program located 0x200 - 0x...
- font located before program 0x000-0x1FF (popular area - 0x050 - 0x09F)

Font

- font character should be 4px x 5px
- first byte is the character (draw vertically in nvim to see)
- stored in memory, index register set to specific font in memory to draw it

Display

- 60Hz - 60 times per second
- sprite consists of 8 bits
- sprites are between 1 and 15 bytes tall
- 0 bits are transparent and 1 bits flip pixel locations

Stack

- stack(LiFo) to call and return from subroutines
- 16 bit addresses (12 bits useful) are saved here

Timers

- two timer registers - the delay timer and sound timer
- one byte in size and if above 0, decremented by 1 60 times per second (60Hz)
- sound timer beeps as long as it's above 0

Keypad

- 123C
- 456D
- 789E
- A0BF

Fetch/decode/excute loop

- fetch the instruction from memory at current PC (program counter)
- decode the instruction to find what emulator should do
- execute instruction and do what it tells you
- this loop's speed has to be set so that it does not run too fast (700Hz)
- fetch: read instruction in PC, two successive bytes and combine into one 16 bit instruction, increment PC by 2
- decode: switch statement checking first half of instruction
- nibbles after first used for decoding, extract these before decoding from opcode
  - X - second nibble
  - Y - third nibble
  - N - fourth nibble
  - NN - third and fourth nibble - 8 bit immd number
  - NNN - second, third and fourth nibble - 12 bit immd address
- execute: do what each instruction should do in each case of the switch

Main

- Setup directory to ignore makefile outputs
- read through instructions
- make SDL display at 60fps

Reference

- https://tobiasvl.github.io/blog/write-a-chip-8-emulator/