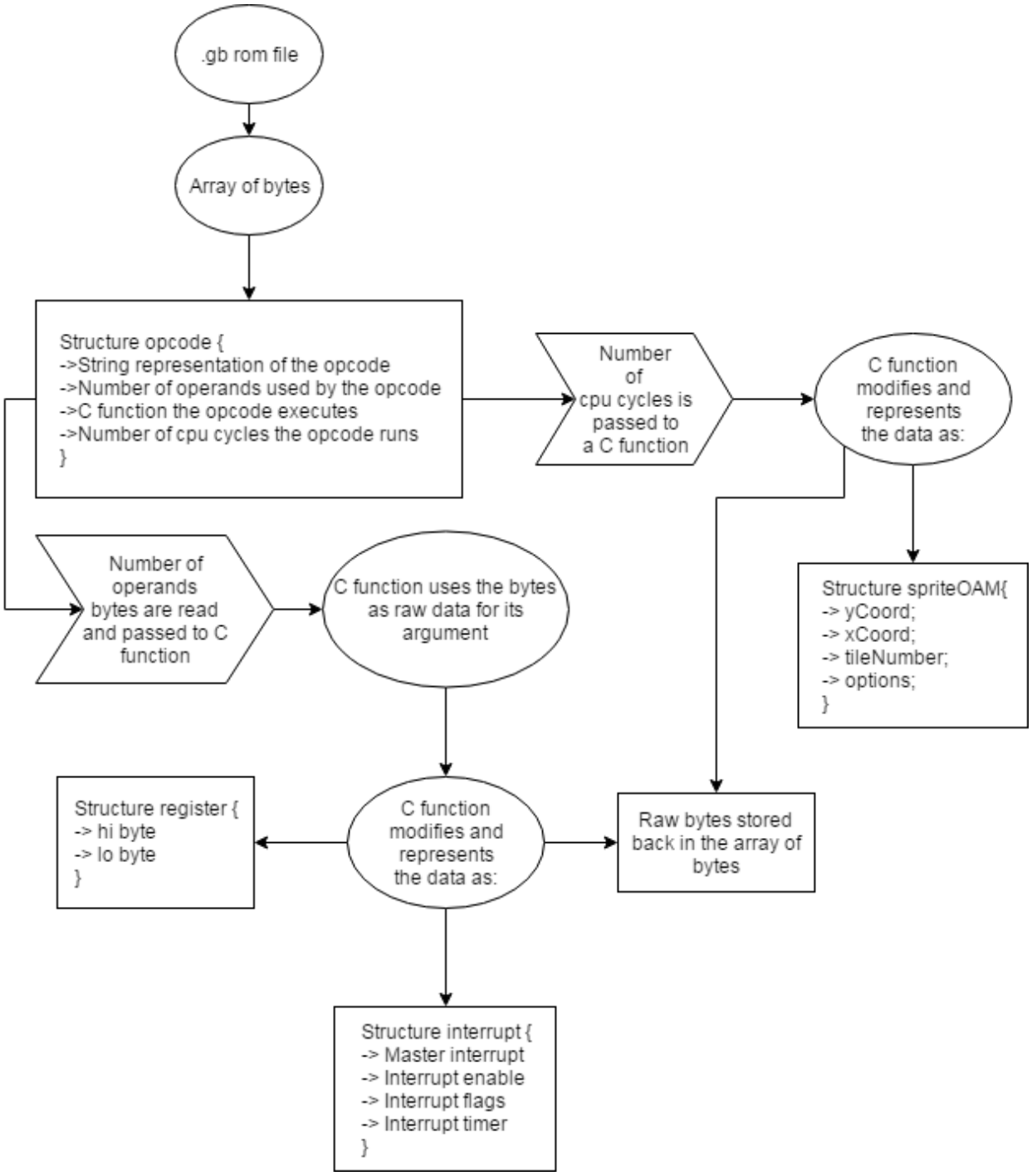
Intermediate Representation



Our intermediate representation is diagramed above. Our input is a .gb rom file. Our code takes that and runs through converting it to a variety of different structures or returning it to an array of data as raw bytes.

How our backend uses these structures to translate to the target language (a running display output)

Structure:

- Opcode:
  - The opcode structure is used for the intermediate representation to further develop the representation of the input data.
- Register:
  - The register structure represents the 8 gameboy registers. If the input data transforms into a register, it is further used in other opcode operations in the future.
- Interrupt:
  - The interrupt structure represents the interrupt switches of the gameboy. If the input data is transformed into an interrupt, it is further used in other operations in the future.
- spriteOAM:
  - The spriteOAM structure is used to convert raw data into a sprite image to be displayed.

Array of Bytes:

- The array of bytes stores all of the original data as well as additional data created by running the original data through the opcode functions as well as data created and modified by any of the gpu functions.

Our code interprets the input .gb rom and stores it in this immediate representation before using it to execute the represented opcode commands which is emulated on our virtual Gameboy hardware.

Currently our code only supports very specific rom settings. All of these rom settings exist inside the test rom (Tetris.gb).

To run the code run the make.bash file. This file will compile the code and then run the Tetris.gb rom. To play Tetris, use the arrow keys for directional input and z to rotate the block. Use the enter key to navigate the menu.