src/**mmu.rs**

Note: This memory is incomplete and only works for our game

```
pub fn read_byte(&mut self, address: u16) → u8 {
        match address {
            0×0000..=0×7FFF ⇒ self.mbc.readrom(address),
            0×8000..=0×9FFF ⇒ self.gpu.read_byte(address),
            0×C000..=0×CFFF | 0×E000..=0×EFFF ⇒
self.wram[address as usize & 0×0FFF],
            0×D000..=0×DFFF | 0×F000..=0×FDFF ⇒ {
                self.wram[(self.wrambank * 0×1000) | address as
usize & 0×0FFF]
            }
            0×FE00..=0×FE9F ⇒ self.gpu.read_byte(address),
            0×FF00 ⇒ self.input.read_byte(),
            0×FF0F ⇒ self.intf | 0b11100000,
            0×FF40..=0×FF4F ⇒ self.gpu.read_byte(address),
            0×FF68..=0×FF6B ⇒ self.gpu.read_byte(address),
            0×FF70 ⇒ self.wrambank as u8,
            0×FF80..=0×FFFE ⇒ self.zram[address as usize &
0×007F],
            0×FFFF ⇒ self.inte,
            _ ⇒ 0×FF,
        }
    }
```

Memory Bank Controllers (MBC)

As the Game Boy 16 bit address bus offers only limited space for ROM and RAM addressing, many games are using Memory Bank Controllers (MBCs) to expand the available address space by bank switching.

These MBC chips are located in the game cartridge
(ie. not in the Game Boy itself).

Retired from https://gbdev.io/pandocs/MBCs.html