

```
pub fn draw(&self, gb: &GameBoy) { unsafe {  
    gl::ClearColor(0.0, 0.0, 1.0, 1.0);  
    gl::Clear(gl::COLOR_BUFFER_BIT);  
  
    gl::TexImage2D(  
        gl::TEXTURE_2D, 0, gl::RGB as i32,  
        gb.width() as i32, gb.height() as i32,  
        0, gl::RGBA, gl::UNSIGNED_BYTE,  
        gb.data().as_ptr() as *const _,  
    );  
    assert_eq!(gl::GetError(), 0);  
  
    gl::DrawElements(  
        gl::TRIANGLES, 6,  
        gl::UNSIGNED_INT, std::ptr::null());  
}
```

```
#[inline]
pub fn load_our_game_rom() → Result<Vec<u8>, Error> {
    use std::{fs::File, io::Read};
    let mut rom = Vec::new();
    let file = File::open("./rom/game.gb");
    file.and_then(|mut f| f.read_to_end(&mut rom))?;
    Ok(rom)
}

fn main() → Result<(), Error> {
    let rom_data = load_our_game_rom()?;
    let gb = GameBoy::new(rom_data);
}
```