src/web.rs

```rust
#[wasm_bindgen]
pub async fn render(rom: Vec<u8>) -> Result<(), wasm_bindgen::JsValue> {
  let mut gb = GameBoy::new(rom);

  let document = window().document().unwrap();
  let game = document.get_element_by_id("game");
  let canvas = document.create_element("canvas")?
      .dyn_into::<web_sys::HtmlCanvasElement>()?;
  game.unwrap().append_child(&canvas)?;
  canvas.set_width(gb.width());
  canvas.set_height(gb.height());
  let context = canvas.get_context("2d")?.unwrap()
      .dyn_into::<CanvasRenderingContext2d>().unwrap();
```

src/web.rs

```
let f_main = Rc::new(RefCell::new(None));
let f_frame = f_main.clone();
*f_frame.borrow_mut() = Some(Closure::wrap(Box::new(move || {
        gb.frame();
        log("Up and running");
        if let Ok(image_data) = ImageData::new_with_u8_clamped_array_and_sh(
            wasm_bindgen::Clamped(gb.data()),
            gb.width(),
            gb.height(),
        ) {
            context.put_image_data(&image_data, 0.0, 0.0).ok();
        }

        request_animation_frame(f_main.borrow().as_ref().unwrap());
}) as Box<dyn FnMut()>));

request_animation_frame(f_frame.borrow().as_ref().unwrap());
```