src/**web.rs**

```
let current_key_code: Rc<RefCell<i32>> = Rc::new(RefCell::new(0));
{
    let key_code = current_key_code.clone();
    let closure =
        Closure::<dyn FnMut(_)>::new(move |event: KeyboardEvent| {
            *key_code.borrow_mut() = event.key_code() as i32;
        });
    add_event_listener(“keydown”, closure.as_ref().unchecked_ref());
    closure.forget();

    let key_code = current_key_code.clone();
    let closure =
        Closure::<dyn FnMut(_)>::new(move |event: KeyboardEvent| {
            *key_code.borrow_mut() = (event.key_code() as i32) * -1;
        });
    add_event_listener(“keyup", closure.as_ref().unchecked_ref());
    closure.forget();
}
```

src/**web.rs**

```rust
let key: RefMut<_> = key_code.borrow_mut();
match *key {
    // A
    65 => gb.keydown(Button::A), -65 => gb.keyup(Button::A),
    // S
    83 => gb.keydown(Button::B), -83 => gb.keyup(Button::B),
    // Z
    90 => gb.keydown(Button::Select), -90 => gb.keyup(Button::Select),
    // X
    88 => gb.keydown(Button::Start), -88 => gb.keyup(Button::Start),
    // Left
    37 => gb.keydown(Button::Left), -37 => gb.keyup(Button::Left),
    // Right
    39 => gb.keydown(Button::Right), -39 => gb.keyup(Button::Right),
    // Up
    38 => gb.keydown(Button::Up), -38 => gb.keyup(Button::Up),
    // Down
    40 => gb.keydown(Button::Down), -40 => gb.keyup(Button::Down),
    _ => (),
}
gb.frame();
```