```toml
[package]
name = “gameboy"
version = "0.1.0"
edition = "2021"

[[bin]]
name = "desktop"
path = "src/desktop.rs"

[lib]
name = "wasm"
path = "src/wasm.rs"
crate-type = ["cdylib"]
```

In this talk we are assuming that:

- [lib] will always be a *cdylib* that targets wasm32
- [lib] contains a completely different code than [[bin]]
- both [lib] and [[bin]] consumes a *gameboy* mod.

This configuration can be quite chaotic for many cases. Majority of the time you want to reuse code between target archs or better organise it (e.g having distributed crates with cargo workspaces).

src/gameboy/**mod.rs**

```
pub enum Button {}

pub struct GameBoy {
    width: u32,
    height: u32,
}
```

```
impl GameBoy {
    pub fn new(rom: Vec<u8>) -> Self {
        Self {
            width: 160,
            height: 144,
        }
    }

    pub fn width(&self) -> u32 { self.width }
    pub fn height(&self) -> u32 { self.height }
    pub fn data(&self) -> &[u8] { &[] }
    pub fn frame(&self) {}
    pub fn keydown(&self, button: Button) {}
    pub fn keyup(&self, button: Button) {}
}
```