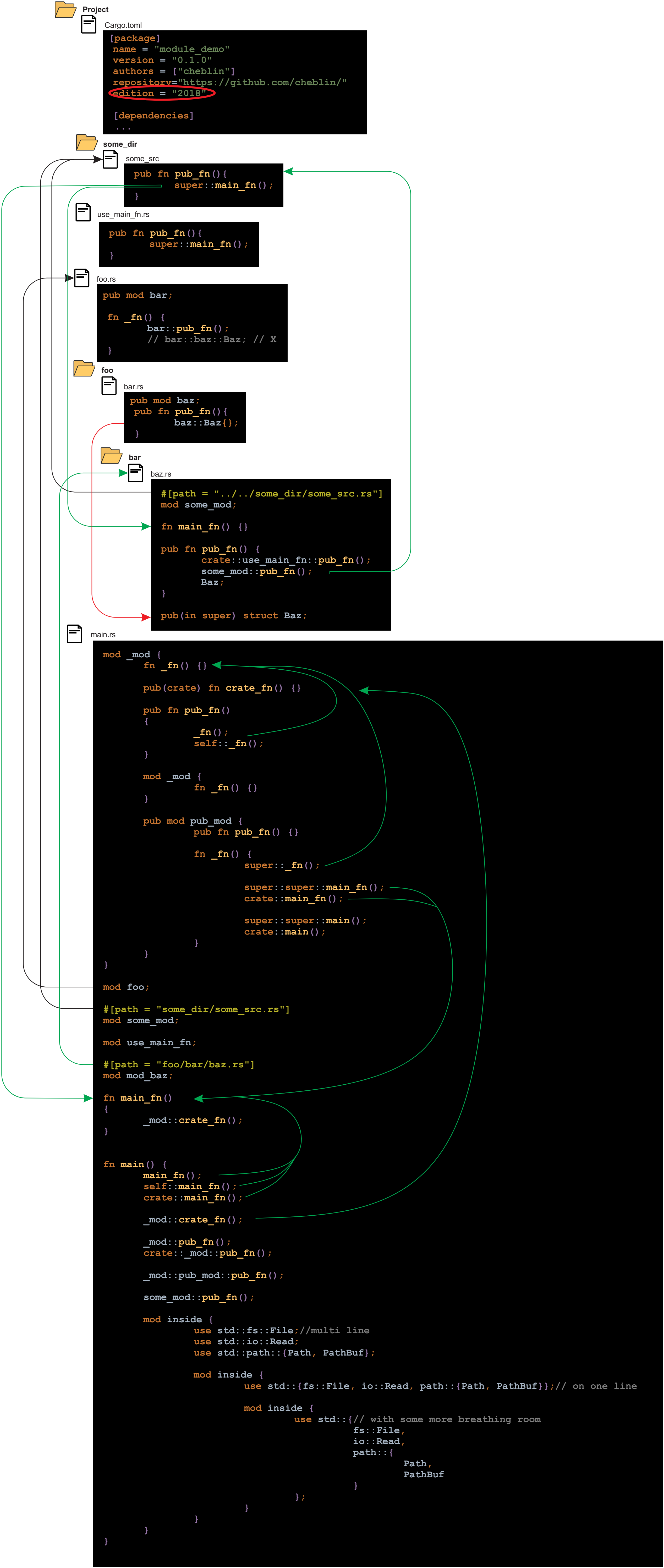

```
[package]
name = "module_demo"
version = "0.1.0"
authors = ["cheblin"]
repository="https://github.com/cheblin/"
edition = "2018"

[dependencies]
...
```

```
pub fn pub_fn(){
        super::main_fn();
}
```

```
pub fn pub_fn(){
        super::main_fn();
}
```

```
pub mod bar;

fn _fn() {
        bar::pub_fn();
        // bar::baz::Baz; // X
}
```

```
pub mod baz;
pub fn pub_fn(){
        baz::Baz{};
}
```

```
#[path = "../../some_dir/some_src.rs"]
mod some_mod;

fn main_fn() {}

pub fn pub_fn() {
        crate::use_main_fn::pub_fn();
        some_mod::pub_fn();
        Baz;
}

pub(in super) struct Baz;
```

```
mod _mod {
        fn _fn() {}

        pub(crate) fn crate_fn() {}

        pub fn pub_fn()
        {
                _fn();
                self::_fn();
        }

        mod _mod {
                fn _fn() {}
        }

        pub mod pub_mod {
                pub fn pub_fn() {}

                fn _fn() {
                        super::_fn();

                        super::super::main_fn();
                        crate::main_fn();

                        super::super::main();
                        crate::main();
                }
        }
}

mod foo;

#[path = "some_dir/some_src.rs"]
mod some_mod;

mod use_main_fn;

#[path = "foo/bar/baz.rs"]
mod mod_baz;

fn main_fn()
{
        _mod::crate_fn();
}


fn main() {
        main_fn();
        self::main_fn();
        crate::main_fn();

        _mod::crate_fn();

        _mod::pub_fn();
        crate::_mod::pub_fn();

        _mod::pub_mod::pub_fn();

        some_mod::pub_fn();

        mod inside {
                use std::fs::File;//multi line
                use std::io::Read;
                use std::path::{Path, PathBuf};

                mod inside {
                        use std::{fs::File, io::Read, path::{Path, PathBuf}};// on one line

                        mod inside {
                                use std::{// with some more breathing room
                                            fs::File,
                                            io::Read,
                                            path::{
                                                    Path,
                                                    PathBuf
                                            }
                                };
                        }
                }
        }
}
```