Translates JetBrains Rider invocations to emacsclient invocations (for Unity)

elizagamedev Bump version to v0.1.1 last year

src	Correctly handle negative line and colu...	last year
.envrc	Initial commit	last year
.gitignore	Initial commit	last year
Cargo.lock	Bump version to v0.1.1	last year
Cargo.toml	Bump version to v0.1.1	last year
README.md	Update readme with important informa...	last year
shell.nix	Initial commit	last year

README

What?

rider2emacs is a command-line utility which translates JetBrains Rider invocations to emacsclient invocations.

Why?

The Unity game engine editor only officially supports four IDEs/editors: MonoDevelop, Visual Studio, Visual Studio Code, and JetBrains Rider. *If and only if* the Unity editor is configured to open source files in one of these editors will it generate sln and csproj files for your code. If you want to use any other editor, including the best text editor, Emacs, with the OmniSharp LSP, you must convince Unity that you're using one of the four editors listed above.

It gets worse: support for each of the above editors each generate project/solution files completely differently from one another, and the ability for OnniSharp to correctly interpret these project and solution files varies just as much. The best of the bunch seems to be the Rider project generation.

Why not just use a shell script or something?

Making this a shell script would more or less make this impossible for Windows users. My first shot at this was done in C for portability reasons and to reduce the dependency tree. However, the amount of `*printf*` calls, string manipulation, and platform-specific code required for what is on paper a very simple shim made me uncomfortable. Unfortunately, this means that installation of unity.el is slightly more complex than I would have liked.

Usage

Now that I have justified why this tool even needs to exist in the first place, here's how to use it.

Install rider2emacs via:

```
cargo intall rider2emacs
```

In Unity, navigate to `Edit->Preferences...`, select the `External Tools` tab in the left-side pane, and select `Browse...` in the drop-down menu for `External Script Editor`.

Navigate to the `rider2emacs` binary. It will have been installed in `$CARGO_INSTALL_ROOT/bin`, which by default is `$HOME/.cargo/bin`. See the documentation on `cargo install` for more details.

Since the file starts with `rider`, Unity will be tricked into thinking that it's actually JetBrains Rider. Congratulations! Any source files you open via Unity should now open in Emacs via `emacsclient`. Ensure you have the Emacs daemon running. See "(emacs)Emacs Server", accessible in the Emacs manual via `{C-h r}` under "Advanced Features".

Troubleshooting

`emacsclient` (or `emacsclientw.exe` on Windows) must be in your `$PATH` for `rider2emacs` to correctly invoke it. On non-Windows platforms, `emacsclient` is invoked via `/bin/sh` so that environment variables set in your `.profile` and similar are taken into account. This is important on macOS, which does not provide a robust way to set `$PATH` for a GUI session, unlike Linux/FreeBSD.

Releases

2 tags

Packages

No packages published

Languages

Rust 96.9% Nix 3.1%