

# **FOUINEJN, un langage Jovial et Naturel**

**or How I learned to stop worrying and love CPS**

Robin JOURDE & Nicolas NARDINO

mai 2021

```

                                `=xvCv!':^czzc|;,
                                .0vsFfU6Qe!,,``~;LE8a6deyk,
                                =yDfz\,r|:                :'?yRCoDR
                                -0Q'                        `raQk
                                wj                            0|
                                Q=   'e$E:                o00z   ~0!
                                ?g'   |8p8B,            `CQaQ|   ;#'
                                ,@c`   ``:w!   `~{:   '=g,
                                @;pu^`   zge8B-   'tu@;`
                                Q\l!:C9c-   ^?+.``e/` Q
                                dai=   ,vp9yo6i~   p"
                                .Q;;e       :z :m       lz
                                `gs `#'       %- 'D       :g
                                '#z   'm       :E B       ,0
                                !B!   .a`       z|%"       vf
                                iQ,       "0`       :#A;       .Q,
                                |Q,       j7       -@!       `R=
                                '@:       fi       -@@`       `0v
                                `#7       9+       ^@{       ue
                                `Uk       %:       dm       ,Q`
                                ~W\       '       #~       Cd       kv
                                vd'       e       `Q. \Q!       \@,
                                -de       ~a`       L% |:g       `9lK/
                                `#$       \l       c9:7,D w|       `0\
                                OD`       :d       =Q8`!6 '@`       `d|
                                lQ`       Q,       .@C =j '@d'       `B-
                                ,@!       |K       fj f\ '@+g; mF
                                pm       `Q!       cEB' rN ze-!|B
                                ,@'       |ka+~,;{CU `B/ +Kya/6
                                t6       B,d>Lw!QDa xd       !dp|`
                                ma       '!=LFQ'!LOdu#=w Q:
                                o$       `:, '':./uoC=`,DO Q:
                                |Q       -+t3@@,R;
                                -@,       'd\ei
                                $y       `Qxe
                                :@,       Rc%
                                % j       Q%k
                                OC       'QWl
                                `~;\uKx` ^R       avQ,
                                `,~luuo3jffttttttfuuCzL=!,`       \@g!       ?dz6
                                -!vCCx=:`       ,B|!@'       'w@0gKKWRpu+`` ``
                                'jjv!`       `:\9W\ $9!       `!CB#kx7l=;ifkBv:;! '~
                                \Ku:   '"``- '~,...':+>>?/zywpp9k9kXmaL!` `Q1",,,,,,'.``       -!L=jQD//+:,`: '``
                                !jo3juujzlvzzzzzvc\?;!!::, '-`       ~|=|>>>?||\lzzzzoe$gQ8e>`

```

À l'origine fut la vitesse, la pure programmation fonctionnelle, le  
«  $\lambda x.e$  »  
Puis l'évaluation décéléra, prit consistance et forme, jusqu'aux  
lenteurs habitables, jusqu'au `let`, jusqu'à vous.  
Bienvenue à toi, lente fouine liée, poussif tresseur de fonctions  
récursives.

---

[Damo7]

## Table des matières

<b>1</b>	<b>Présentation</b>	<b>3</b>
<b>2</b>	<b>Organisation et répartition du travail</b>	<b>4</b>
2.1	Organisation . . . . .	4
2.2	La répartition . . . . .	5
<b>3</b>	<b>Collaboration</b>	<b>5</b>
3.1	git . . . . .	5
3.2	Postérité . . . . .	6

## I Présentation

Foino (Martes foina) aŭ estas specio de marteso kiu havas brilan blugrizan ĝis blubrunan hararon, iom malpli delikatan ol tiu de la arbara marteso; duobla makulo sur la gorĝo estas blanka. [Vik21]

`fouine` est un interpréteur de `FOUINEJN` (Fouine Joviale et Naturelle <sup>1</sup>), sous-ensemble du langage de programmation national, `OCAML`.

`fouine` est programmé principalement `OCAML` (1), et le parser associé est généré avec `MENHIR` (2).

---

1. [Citation Needed]



FIGURE 1 – Exemple de béchamel, dans laquelle on peut trouver `caml`



FIGURE 2 – Exemple de menhir

## 2 Organisation et répartition du travail

### 2.1 Organisation

Comme dit plus haut, `fouine` est programmé principalement en OCAML, avec pas moins de 12 fichiers `.ml`<sup>2</sup> ! Nous les détaillerons plus tard. Ils sont accompagnés d'un `lexer.ml` et `parser.mly`, qui à eux deux, spécifient la grammaire de FOUINEJN.

Dans un ordre peu ou prou chronologique, les fichiers `.ml` sont :

**`expr.ml`** La base de la pyramide, c'est ici qu'on définit les expressions.

**`eval.ml`** Le cœur du travail, c'est là que la fouine travaille

---

2. Source : `ls -al | grep -c "\.ml$"`

**main.ml** L'enrobage

**affichage.ml** Parce que c'est pratique d'avoir un retour sur ce qui se passe

**memory.ml** Pouvoir se souvenir de choses est un avantage évolutif certain, pas étonnant qu'une fouine en bénéficie

**stdLib.ml** Des fonctions tout plein (en particulier, nos opérateurs infixes)

**transformation.ml** Gère la transformation CPS des programmes en entrée

**reduction.ml** Aide le fichier précédent dans sa quête

**types.ml** On type enfin notre langage! Analogue de `expr.ml`

**inference.ml** L'inférence de type monomorphe

**polymorphisme.ml** blablabla polymorphe

**unification.ml** Ze algorithm of unification!

Un, deux, trois... Douze! Le compte y est.

## 2.2 La répartition

Au commencement, il n'y avait rien... Enfin, il y avait le mini-projet de PROJ1 de N. (noms masqués pour préserver l'anonymat), qui a peut-être un peu aidé. Mais le vaillant R. dit : « Que les expressions soient » et les expressions furent. Et avec ces dernières, l'évaluation.

Mais travailler avec des expressions brutes, ce n'est pas pratique, et alors N. parsa. Et parsa. Et parsa. Et il vit que le parser était bien (*spoiler alert*, il n'était pas si bien que ça).

Quand vint le filtrage, un singulier shift/reduce conflict nous embêta des semaines durant. Il s'agissait de pouvoir imbriquer deux `match`. Jusqu'au jour de la Libération, quand le sage D.H. nous en libéra.

Entre-temps, les continuations arrivèrent, et R. entreprit de métamorphoser ce que les vils utilisateurs envoyaient à notre bien-aimée fouine en CPS, tandis que N. passait `eval` en CPS, afin de pouvoir traiter les exceptions. La transformation posa quelques problèmes à R., notamment pour les fonctions récursives, et il dut utiliser la ruse afin de triompher (*i.e.* des réductions).

Mais le triomphe fut court, et le typage pointa le bout de sa truffe. Alors que N. inférait monomorphiquement, R. unifiait les peuples sous sa bannière, et quand ce fut fait, et que le calme retomba, N. put enfin s'atteler aux réparations tant attendues du parser, alors que R. s'adonna à son chef-d'œuvre, le *plusieurs-formes*.

## 3 Collaboration

### 3.1 git

Il était évident que du contrôle de version allait être nécessaire, ainsi qu'un dépôt distant. Nous avons donc créé un dépôt sur `github`, que l'on peut retrouver ici : `github.com/vecteurNabla/proj2`.

Grâce à notre sens de l'organisation hors du commun, notre répartition exemplaire du travail, ainsi que qu'une bonne dose de chance sûrement, il n'y eu que très peu de conflits à résoudre, et

tous furent résolus aisément, dans le plus grand professionnalisme, non sans l'aide de porcelaines comme magit<sup>3</sup>

### 3.2 Postérité

Notre projet est extrêmement populaire, comme le montre le graphique suivant :



FIGURE 3 – Oh c'est joli c'est bleu

On peut observer pas moins de 31 clones en deux semaines !

## Conclusion

Voili voilou [Lan66].

## Épilogue ?

---

3. <https://magit.vc/>

# TCHANA2025 [NTT<sup>+</sup>21]

## Références

- [Dam07] Alain Damasio. *La Horde du contrevient*. Gallimard, 2007.
- [Lan66] Peter J. Landin. The next 700 programming languages. *Commun. ACM*, 9(3) :157–166, 1966.
- [NTT<sup>+</sup>21] Tu Dinh Ngoc, Boris Teabe, Alain Tchana, Gilles Muller, and Daniel Hagimont. Mitigating vulnerability windows with hypervisor transplant. In Antonio Barbalace, Pramod Bhatotia, Lorenzo Alvisi, and Cristian Cadar, editors, *EuroSys '21 : Sixteenth European Conference on Computer Systems, Online Event, United Kingdom, April 26-28, 2021*, pages 162–177. ACM, 2021.
- [Vik21] Vikipedio. Foino — vikipedio, 2021. [Ligite; alirita je 8-majo-2021].