

Bonnes pratiques de développement

Quentin Fortier

July 4, 2022

Conseils personnels pour votre TIPE :

- 1 Utiliser [Visual Code](#)
- 2 Utiliser Git via Visual Code
- 3 Poster votre code sur GitHub (éventuellement en privé, en m'ajoutant en collaborateur)

Pour tout ça, chercher de l'aide sur internet ou me demander (personnellement ou sur [GitHub Discussions](#)).

Choisir un langage de programmation suivant vos besoins et vos connaissances :

- ① Machine learning (pour une voiture autonome, pour une caméra de surveillance...) : Python ou OCaml avec numpy/sklearn/tensorflow...
- ② Création d'un langage : OCaml avec OCamllex et OCaml yacc/Menhir
- ③ Génération de scène 3D par ray tracing (lancé de rayons) : C avec CUDA
- ④ ...

TIPE

- 1 Ajouter un README en expliquant à quoi sert votre code, comment l'utiliser...

- ① Ajouter un README en expliquant à quoi sert votre code, comment l'utiliser...
- ② Faites en sorte de pouvoir compiler/exécuter facilement votre projet :
 - Python : requirements.txt ou poetry
 - OCaml : dune
 - C : Makefile

- ❶ Ajouter un README en expliquant à quoi sert votre code, comment l'utiliser...
- ❷ Faites en sorte de pouvoir compiler/exécuter facilement votre projet :
 - Python : requirements.txt ou poetry
 - OCaml : dune
 - C : Makefile
- ❸ Commenter votre code

- ❶ Ajouter un README en expliquant à quoi sert votre code, comment l'utiliser...
- ❷ Faites en sorte de pouvoir compiler/exécuter facilement votre projet :
 - Python : requirements.txt ou poetry
 - OCaml : dune
 - C : Makefile
- ❸ Commenter votre code
- ❹ Ajouter des tests unitaires (chaque test vérifie une fonctionnalité très précise) :
 - Python : pytest
 - OCaml : dune
 - C : ?

Exemple de `code commenté` :

```
(** [dichotomy a e] returns true if [a] contains [e],
false otherwise. [a] must be a sorted array. *)
let binary_search a e =
  let rec aux i j = (* is e in a.(i), ..., a.(j) ? *)
    if i > j then false
    else let m = (i + j)/2 in
      if a.(m) = e then true
      else if a.(m) < e then aux (m + 1) j
      else aux i (m - 1)
  in aux 0 (Array.length a - 1)
```

Exemple de `test unitaire` :

```
open Mp2i.ArrayUtils
let a = [| -5; 1; 4; 6; 9; 12 |]
let%test "binary_search" =
  binary_search a 1 && not binary_search a 2
```

Template de projets :

- Python
- OCaml
- C : TODO

Optionnel : vous pouvez utiliser des modules (en OCaml) ou des classes (en Python voire C++). Modules en OCaml :

- cs3110.github.io
- Tuto
- Documentation officielle
- Real World OCaml