

*Origine, motivations et  
conception de  
l'application...*

*L'application...*

*Traveler's Tools*



*Go, Go, Go...*

*Go, Go, Go...*

## *1 - Objectif de mon projet*

La genèse de mon idée a pris sa source lors du développement du neuvième exercice nommé « Baluchon ». Pour rappel, ce projet était une application dont le but était d'offrir à un voyageur un traducteur, un convertisseur monétaire, une indication météorologique et tout cela dans une même application.

Dans mon fort intérieur il m'apparaissait évident qu'elle manquait de certains services pour remplir d'avantage sa fonction de couteau suisse du voyageur. J'en liste ici quelques-uns :

- un assistant d'orientation qui le moment venu se montrerait très utile lors d'une immersion dans des lieux inconnus.
- un moyen de lister et de transmettre les coordonnées facilement et rapidement et cela par une manière original d'encodage.
- un outil permettant de faciliter le contact avec la population par la participation au jeux très actuel que sont les jeux de rôle, mais pas que.

Je souhaitais aussi profiter de cette dernière création pour revoir et intégrer tout un ensemble de frameworks qui m'a accompagné tout au long de cette année.

Enfin, je voulais me prouver que j'étais capable de les orchestrer sous une même application, tout en la garantissant stable et évolutive.

J'espère que cette application, accessible en open source sur Github sera installée par le plus grand nombre, afin d'obtenir de nombreux retour m'encourageant à envisager une seconde version. Celle-ci se voudra étoffer des éléments attendus et toujours dans l'esprit de proposer un outil aux globe-trotters d'où son nom « *Traveler's Tools* ».

Je souhaite à tous les utilisateurs d'apprécier sans limite mon travail autant que j'ai eu de plaisir à développer cette application.

# *Page d'accueil*

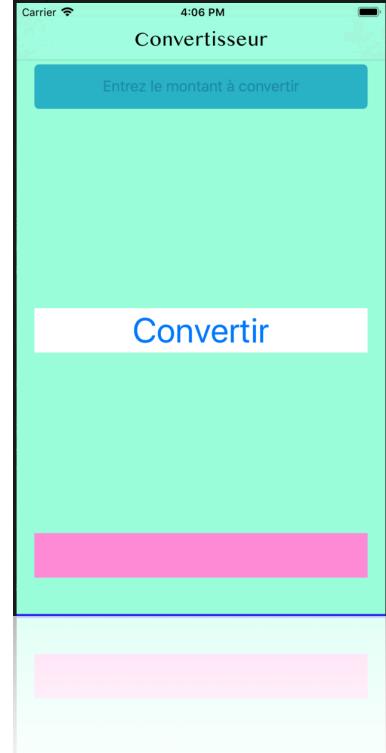
Ma conception d'optimisation me contraint à utiliser toutes les vues. Alors dès la page d'accueil l'application devait avoir son utilité. J'y ai domicilié une calculatrice (un petit clin d'oeil a été fait à CountOnMe) dans les deux premiers tiers de la vue et dans le dernier le convertisseur monétaire, première composante du « Baluchon ».



Ne nous y trompons par, il n'était pas question pour moi de faire un copier coller de mon code ni de mon graphisme. Voyez plutôt...

J'ai tout d'abord converti les appels réseaux des trois éléments vers « Alamofire », créant de fait un protocole et une session pour chacun, puis je les ai factorisés.

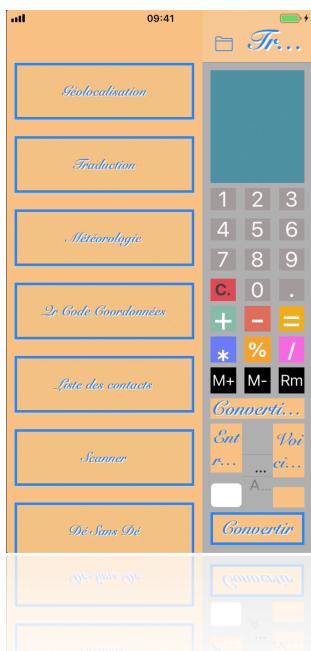
Je lui ai apporté un pickerView offrant un plus grand choix de monnaies, pour finir j'ai animé tous les boutons.



Voici les captures d'écran des deux versions.

## Container de navigation

Au regard de mon ambitieux projet, la Tab-Bar me semblait sous dimensionné alors, j'ai opté pour un container sous la forme d'une « `TableViewController` »



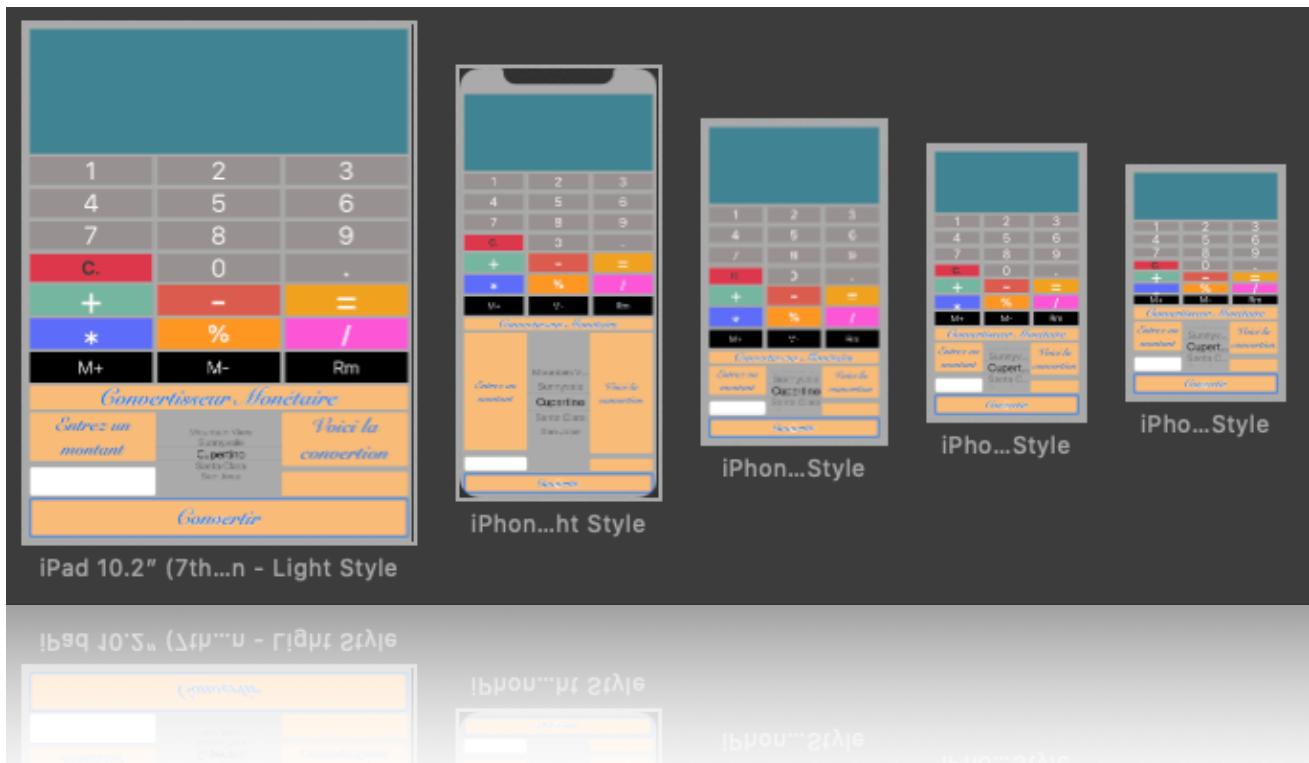
rempli de `tableViewController`. Celle-ci ne devrait apparaître que lorsque l'on clique sur ce bouton



Chacune d'entre elle dirige l'utilisateur vers une partie de l'application.

On pourra remarquer que l'arrivée du container permet toujours l'usage des deux composantes de la vue d'accueil. (calculatrice et convertisseur). Dans mon iPhone 6, l'exercice est périlleux mais sur un écran plus grand tout redéveloppe visible.

Ici nous utilisons un iPhone 6, le rendu n'est pas le même sur les autres tailles d'écran (`autolayout` oblige).



## Géolocalisation

### Le framework « MapKit »

Ce qui me paraissait important dans mon application de voyage était de donner la possibilité à son utilisateur de déambuler là où il se trouve en toute liberté sans avoir à se soucier de retrouver son chemin. J'ai donc intégré une géolocalisation. Pour plus de finesse, j'ai souhaité donner le choix du mode de déplacement, alors j'ai intégré un menu déroulant avec les principaux moyens actuels qui assombrît la vue principale lorsqu'il est déroulé.

```
var dataSource = ["Automobile", "Walking", "Any", "Transit"]
directionsRequest.transportType = .automobile
case "Walking":
    directionsRequest.transportType = .walking
case "Transit":
    directionsRequest.transportType = .transit
case "Any":
    directionsRequest.transportType = .any
```

Une fois renseigné le menu se replie, redonne toute la luminosité à la carte et libère un champs dans lequel l'utilisateur devra indiquer le lieu ou le genre d'établissement qu'il recherche. Une fois terminer il devra valider son choix en pressant sur le bouton du clavier.

Alors, le parcours se **Rechercher** dessine en bleu sur la carte et à chaque changement de direction un rond vert apparaît.

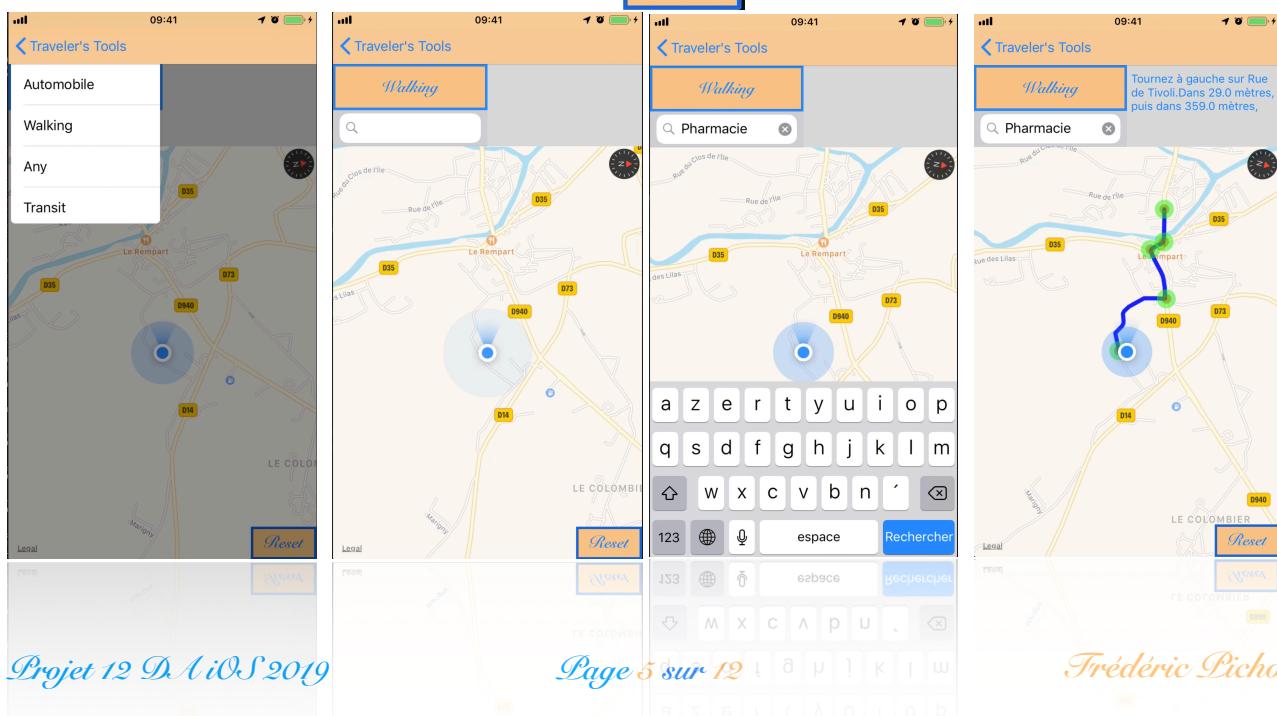
On remarquera aussi qu'en haut à droite le parcours est noté, il évoluera au fur et à mesure du déplacement.

```
let initialMessage1 = "Dans \u2028(self.steps[0].distance) m\u00e8tres,"
let initialMessage2 = "\u2028(self.steps[0].instructions) puis dans \u2028(self.steps[1].distance) m\u00e8tres,"
let initialMessage = "\u2028(self.steps[1].instructions)." + initialMessage1 + initialMessage2
self.directionTextField.text = initialMessage
```

et pour finir sa lecture est audible.

```
98 // sound of the message
99         let speechUtterance = AVSpeechUtterance(string: initialMessage)
100        self.speechSynthesizer.speak(speechUtterance)
101        self.stepCounter += 1
```

Pour toute nouvelle requête un bouton **Reset** est présent.





Traduction

Les deux autres composantes de « Baluchon » ont eu les mêmes transformations de leur code que le convertisseur, nous remarquons tout de même un effort sur la charte graphique.



Sélectionnez la langue

albanais  
allemand  
anglais  
arabe

Traduction



Météorologie

Choisissez une ville

Alexandrie  
Alger  
Athena  
Avignon

Rechercher

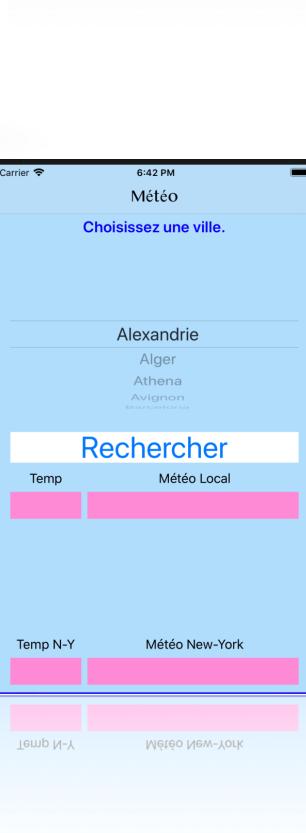
Temp Météo Local Giel



Temp Météo New-York Giel



Giel New-York Météo Temp





L'idée du **Qr code** est venu lors d'un précédent voyage où je souhaitais échanger mes coordonnées avec des rencontres d'opportunités. Malheureusement, en l'absence de papier ou de crayon devenu introuvable sur le moment ou encore dans l'impossibilité de retrouver cesdits papiers, j'ai à mes dépends été dans l'incapacité de les recontacter.



Avec le Qr code, il ne m'arrivera plus ce genre d'étourderie, vous remplissez tout ou parti des champs proposés et vous obtiendrez une concaténation de ceux-ci qui prendra la forme d'un Qr code () .

Vous pourrez à loisirs **l'imprimer**, sous réserve qu'une imprimante compatible « **AirPrint** » soit dans votre entourage, **l'envoyer par email ou par sms** au possesseur du numéro de téléphone ou de l'adresse email (un appel du pied à Instagrid et à OpenQuizz), ou encore conserver les informations en enregistrant les saisies sur votre téléphone par l'intermédiaire du « **Core Data** », un petit clin d'oeil au projet 10.

Ces manipulations se font de différentes manières, soit en actionnant les boutons respectifs, qui s'animent confirmant ainsi l'action, soit en déplaçant le Qr Code.

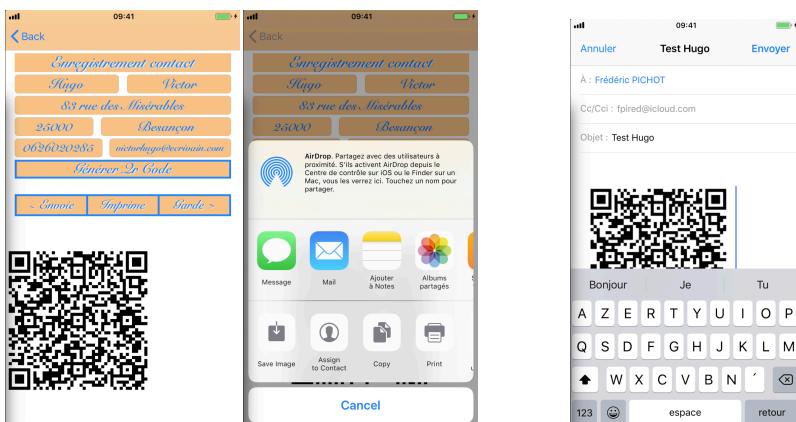


Si vous le glissez vers la droite, vous enregistrez les éléments précédemment entrez dans la base de données et vers la gauche vous l'envoyez par le moyen de votre choix.

Seule l'impression reste attachée au bouton.

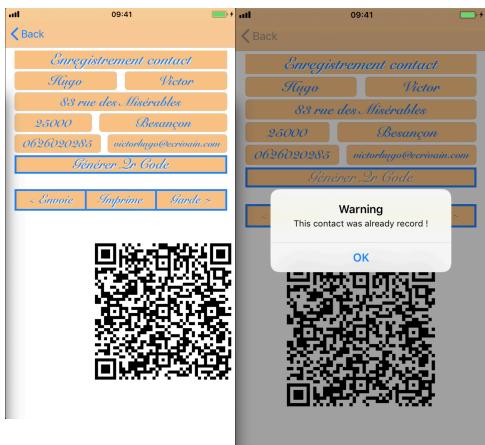
## Exemple avec panGesture :

### Envoyer du QRCode

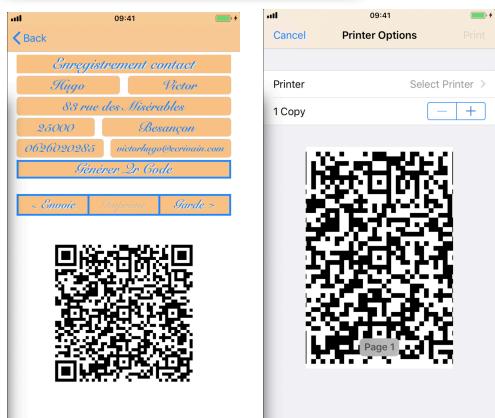


### Sauvegarde dans la base de données

Dans l'exemple ci-dessous une alerte signale que le contact a déjà été enregistré.



### Usage d'AirPrint



## Liste des contacts

La liste des contacts s'est imposée d'elle même, que serait la sauvegarde d'informations importantes sans moyen de les visualiser. J'ai donc tout naturellement pensé à une « **tableView** » option déjà présente dans « Reciplease ».



## Tests de l'application

Les touches « commande + u » ont lancé tous les tests en même temps, ils ont tous réussi. L'application n'a pas d'alarme.

```
let request: NSFetchedResultsController<Contact> = Contact.fetchRequest()
for index in 0..
```

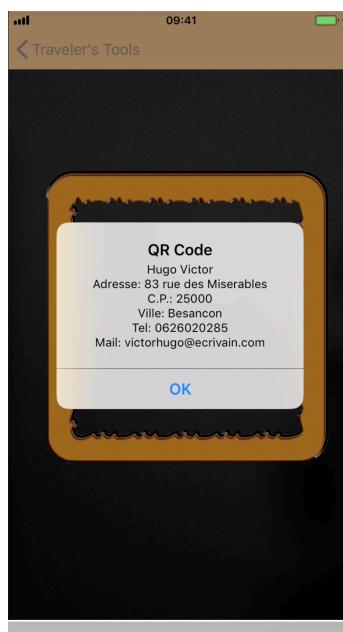
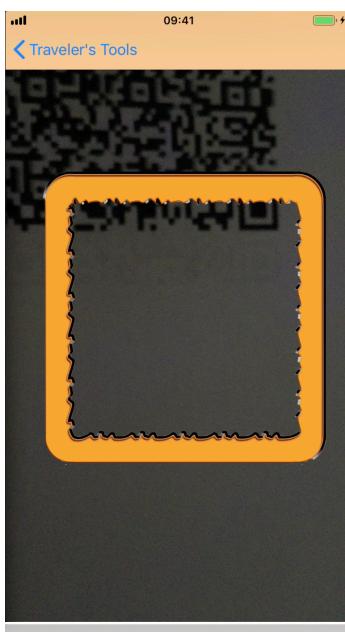
## Scanner

Le scanner lui aussi s'est imposé de lui-même. Pourquoi construire des Qr Codes si nous n'avons pas sur soi un lecteur capable de le déchiffrer. Même si pour cela je devais modifier mon « Info.plist »

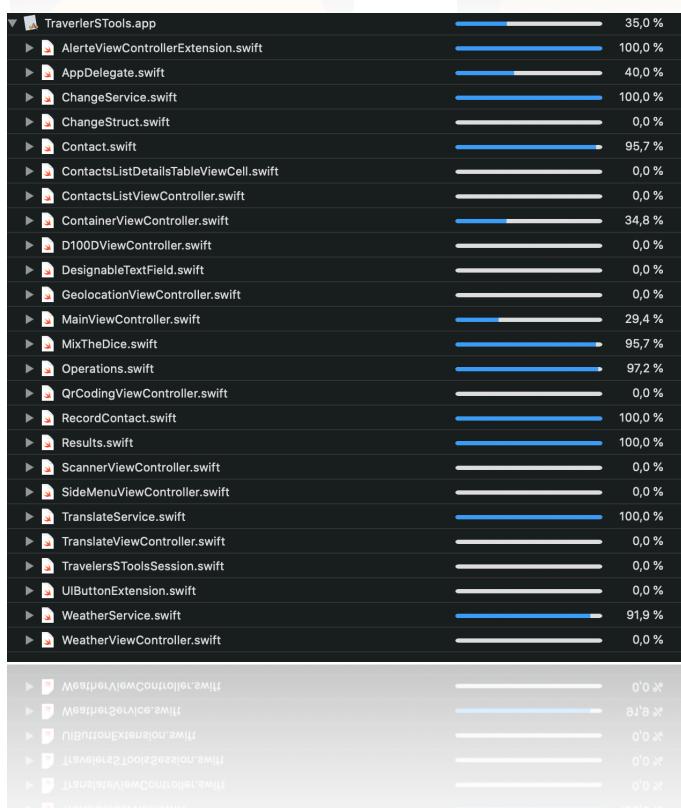
Privacy - Camera Usage Description

String

Used to scan barcodes



## Visualisation du Coverage



## Dé Sans Dé

Pourquoi un dé dématérialisé ou « dé sans dé » D100D ?

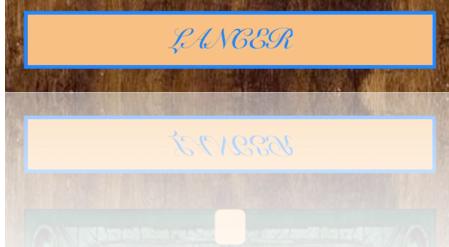
Une tendance très actuelle oriente des communautés de joueurs vers des jeux de rôle. Il existe des déclinaisons importantes de règles du jeux, mais les principes de base restent fondamentalement les mêmes, à savoir que le principal acteur

qui ordonne l'ordre de passage, qui affecte la valeur de la force permettant ou non de détruire le zombie et autres créatures est le dé.

Celui-ci s'est émancipé de l'objet classique que tout le monde connaît, aujourd'hui et c'est là ce qui cristallise les différentes règles de jeu c'est avec combien de dés jouent-on et ceux-ci doivent avoir combien de faces?

Voici un petit florilège de typologie de dés.

Le principal inconvénient de posséder physiquement ce genre d'objet c'est que l'on ne sait jamais si l'on a les bons dés et le bon nombre. Avec le **D100D** cet inconvénient disparaît au profit, si par hasard il faisait défaut, de le proposer à la communauté.



Ainsi vous pouvez sans souci vous faire inviter au sein de ces cercles de jeux en vous rendant incontournable.

Sa particularité ne s'arrête pas là, il peut être secoué en guise de lancement, on écoute le bruit des dés dans le gobelet jusqu'à leur lancement.

## *2 - Bilan de mon parcours.*

J'ai débuté la programmation en 1994 avec les langages C , C++, Pascal etc. A cette époque ce que j'appréciais tout particulièrement était la confrontation avec les lignes de code pour qu'en fine les attendus des progiciels se révèlent au grand jour. Les stratégies et les astuces faisaient partie de l'auto apprentissage et en cela je n'étais pas mauvais.

Cette année, j'ai débuté le parcours de Développeur d'Application iOS avec le même enthousiasme que précédemment. Je peux assurer que de pouvoir infléchir les agissements d'un iPhone ont décuplé mon intérêt à apprendre le langage Swift et l'univers d' Xcode. De pouvoir infléchir les supports mobiles est un piment et un enjeu qui mettaient encore plus en perspective le défit que je me lançais.

Le projet numéro trois du parcours DA iOS a été le premier achoppement avec le monde de la POO. J'ai présenté deux versions de ce jeu, elles étaient solides et fonctionnelles, mais l'examinateur m'a reconduit car mon travail ressemblait trait pour trait à ce que je rédigeais le siècle dernier. Pourtant les cours sur la POO sont très bien fait et graduels, mais il me manquait indéniablement la compréhension du concept de l'objet. L'inertie de mon expérience a quelque peu freiné le déclic tant attendu, maintenant je me sens presque aussi bien qu'avant et je découvre tout les jours de nouvelles possibilités ouvrant des possibilité proche de l'impossible. Mon parcours et l'évolution visible de mon niveau me dynamise encore d'avantage. Je ne sais pas de quoi sera fait mon devenir, mais je le conçois dans ce monde si stimulant qu'est celui de la programmation.

Je mise beaucoup sur l'obtention de cet examen et je crois en mes chances. Il est vrai que j'ai des efforts à produire sur le design et l'animation de mes applications, en cela j'espère trouver de quoi me parfaire dans les parcours proposés par OC, mais je garde à l'esprit que je dois continuer à prendre de l'expérience et éléver encore mon niveau d'étude afin qu'il coïncide d'avantage avec mon âge administratif.

Vous ne trouverez pas à la fin de ce parcours un candidat fatigué et à bout de souffle. Je crois plutôt que les espoirs que je fondais dans ma reconversion professionnelle se concrétisent. J'ai déjà hâte de reprendre mon bâton de pèlerin pour pousser plus loin encore mon aventure. Aujourd'hui je peux sans complexe écrire que je fais ce qu'il me plaît.