

**Bastien Souarn**

Briac De-Rivoyre

Antoine Lunven

## *Dossier projet : ISN*

# *Les Fabuleuses aventures du Pizzaiolo*



# SOMMAIRE

## I) présentation du projet

1. Cahier des charges
2. Moyen mise en oeuvre
3. Structure du projet
4. Répartition du travail

## II) Realisation personnelle

## III) Résultat final

## IV) Bilan personnel et perspective

## I) présentation du projet

### Introduction :

Notre équipe de Pizzaïolo s'est lancée dans un projet en apparence basique (créer un jeu de plateformes) mais qui nécessite de nombreuses connaissances en programmation orientée objet. Nous sommes fiers de vous présenter notre jeu intitulé "Les Fabuleuses aventures du Pizzaïolo" dans lequel notre vedette incontournable du monde de la gastronomie va se lancer à la recherche de la pizza parfaite.

### Pourquoi un tel projet ?

Depuis la création de notre groupe, l'idée de créer un jeu de plateformes était presque évidente et après plusieurs jours de réflexion, nous avons arrêté notre réflexion sur le choix d'un jeu en 2D, inspiré de la célèbre saga "Mario". Nous avons choisi ce projet pour différentes raisons. Premièrement, les jeux Mario ayant bercé nos enfances, c'est l'esprit plein de nostalgie que nous avons décidé de nous replonger dans cet univers de fantaisie. De plus, c'était une occasion pour nous d'étendre nos connaissances en programmation d'une manière ludique.

La charge de travail nécessaire à la création de ce jeu ne nous a pas découragés. D'ailleurs, nous sommes fiers du résultat final. En effet, 5 mois auparavant, nous ne pensions pas être capables de réaliser un tel projet. Ce jeu représente pour nous un apport à la fois personnel et professionnel car sa réalisation n'a pas été aisée, ce qui a permis de renforcer la cohésion au sein du groupe.

La problématique était donc pour nous de créer une interface graphique pouvant accueillir un personnage, se déplaçant de plateformes en plateformes tout en évitant les obstacles

### **1) Cahier des charges**

- Le personnage est un pizzaïolo de couleur rouge et grise.
- Les plateformes sont dans les airs.
- La victoire : récupérer une pizza sans tomber ni toucher l'ennemi
- La défaite :
  - à chaque fois que le pizzaïolo tombe dans le vide (entre deux plateformes)
  - à chaque fois que le pizzaïolo touche un obstacle (un ennemi)
- Utiliser la Programmation orientée Objet
- Créer différents types de plateformes :
  - des plateformes piège qui tombent lorsque le personnage marche dessus
  - des plateformes mobiles qui se déplacent verticalement
- Possibilité de recommencer le niveau lors d'une chute

### **2) Moyens mis en œuvre**

Notre programme a été entièrement codé en Python. C'est effectivement le seul langage de programmation que nous avons appris cette année, et il s'avère qu'il était tout à fait adéquate pour réaliser notre projet de jeu de plateforme.

Néanmoins, subsistait un problème, car il n'était pas possible de créer une interface graphique en utilisant uniquement Python. Il nous fallait alors trouver un plugin ou un logiciel nous permettant de créer une interface graphique à partir de Python.

C'est pour cela, que nous nous sommes directement tournés vers Pygame qui, comme son nom l'indique, nous permet de créer des jeux en Python en permettant la programmation d'une interface graphique.

Pour en faire une présentation rapide : Pygame est une librairie de commandes pour python spécialisée dans la création d'interface graphique et de jeu



Ensuite, il nous fallait un logiciel de support pour coder et lancer notre programme. Nous avons commencé avec Thonny, mais nous nous sommes rendu compte qu'il était vraiment trop basique. Nous avons besoin d'un logiciel plus complet et c'est pour cela que nous nous sommes rapidement tournés vers Pycharm.

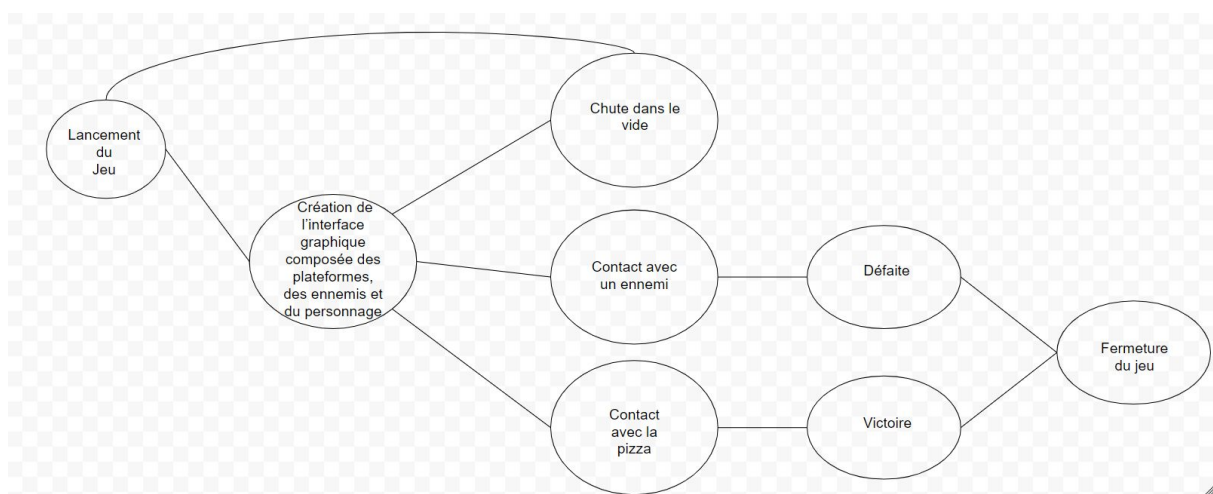


Pycharm est un environnement de développement coloré, permettant un meilleur travail visuel pour différencier les lignes et se retrouver plus facilement dans le code.

Un site a été utilisé pour la réalisation de notre projet, le site Piskel que l'on a utilisé pour le design de nos personnages et de notre plateforme.



### **3) Structure du projet**



#### 4) répartition des tâches

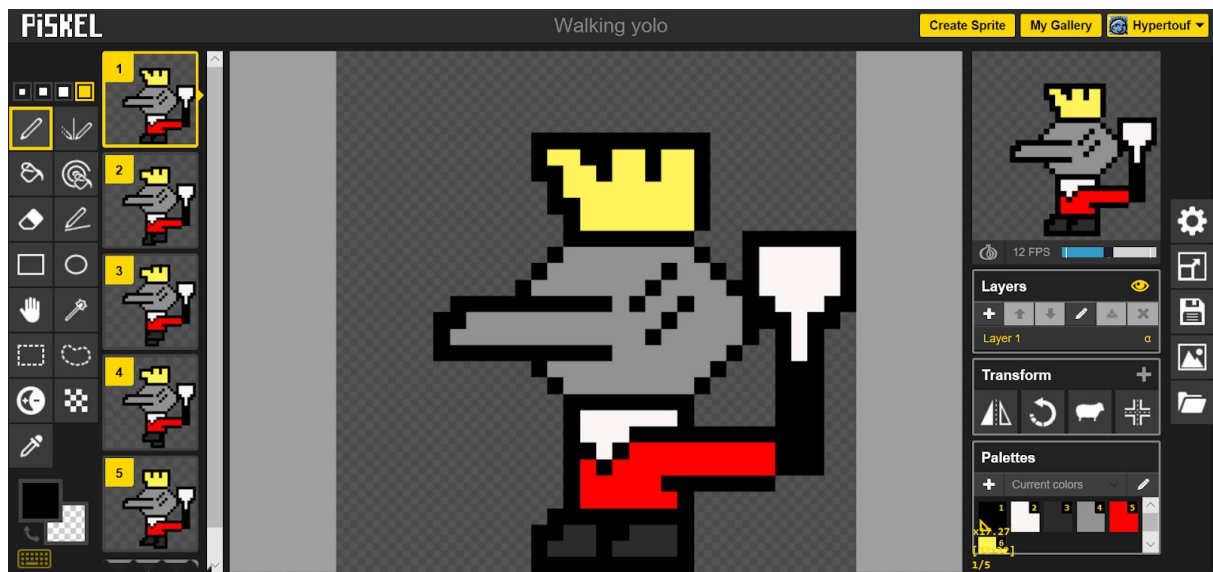
|                     |  |         |       |
|---------------------|--|---------|-------|
|                     | Avancée des Tâches   |         |       |
|                     | Antoine  | Bastien | Briac |
| 10/01/2019          | Choix du sujet : un jeu de plateformes dans lequel un personnage doit franchir différents obstacles afin d'atteindre une pizza   |         |       |
| 20/01/2019          | Comprendre le fonctionnement d'un jeu sous pygame à l'aide de forums et tutoriels. Néanmoins, cela n'est pas suffisant, ce qui nous pousse à découvrir par nous-mêmes les potentialités de création sous Pygame.   |         |       |
| 24/01/2019          | <p>Création de l'interface graphique de base dans laquelle le personnage est immobile.</p> <p>Création des personnages et des plateformes à l'aide du site design Piskel.com</p> <p>Répartition du travail:</p> <ul style="list-style-type: none"><li>-Création d'une interface dans laquelle on peut faire bouger le personnage</li><li>-Création des plateformes</li><li>-création des ennemis et des hitboxes associées</li></ul> |         |       |
| 31/01/2019          | Mise en commun des premières lignes de code et découverte des fonctions de base de Pygame  |         |       |
| 07/02/2019          | recherche d'un objectif de fin de jeu : Design de la pizza   |         |       |
| Vacances de février | Création des différents moyens de mobilité : Implémentation du mouvement et du saut dans le jeu  |         |       |
| 14/03/2019          | Mise en place de dynamisme additionnel : Implémentation de l'animation de marche.  |         |       |
| 03/04/2019          | Mise en place de la programmation orientée objet (création des classes) pour ranger le code.   |         |       |

|            |   |
|------------|---|
| 11/04/2019 | <p>Ajout de beaucoup de contenus et de nouvelles classes : Mise en place de la gravité, création des plateformes, des ennemis et des hitboxes.</p> <p>-&gt; le jeu prend forme.</p> <p>Début de la création d'une page d'accueil</p> <p>Recherche et compréhension des différents outils de Tkinter à mes camarades</p> |
| 18/04/2019 | <p>Mise en place de l'objectif : Ajout de la princesse (et d'un fond d'écran)</p> <p>Design de l'architecture du début du niveau</p>  |
| 09/05/2019 | <p>design de l'architecture de la fin du niveau</p> <p>Mise en place de deux fins possibles au jeu : La mort suite à la chute ou en touchant les ennemis, ou alors la victoire en attrapant la pizza.</p> <p>Problème majeur rencontré : Tkinter incompatible avec pygame</p>   |
| 12/05      | <p>Ajout de commentaires afin d'explicitier le rôle des fonctions et changement du nom de certaines variables</p>   |

## II/ Réalisation personnelle

En somme, ma participation à ce projet consiste en plusieurs partis diverses et variés :

Pour commencer, je me suis occupé du design et de dessiner les différentes entités du jeu et les animations. C'est à dire, de créer le pizzaïolo, les ennemies, la princesse pizza, l'animation de marche, à partir d'un petit site permettant de faire du pixel art (Piskel.com).



Capture d'écran du site de design des personnages du jeu

Il fallait ensuite intégrer ces images dans le jeu, je m'en suis donc donné la responsabilité. J'ai donc fait des recherches sur internet pour trouver comment faire, et j'ai pu au finale ajouter toutes les images nécessaires dans notre jeu.

Comme on peut le voir dans cette partie du code sur lequel j'ai travaillé :

```

4  # on importe les images de l'arrière plan et des plateformes
5  background = pygame.image.load('Background2.jpg')
6  platformimg = pygame.image.load('platform.png')
7  # on importe les images pour notre peronnages principal
8  walkRight = [pygame.image.load('Reverseyolo.png'), pygame.image.load('Ralkingyolo1.png'),
9               pygame.image.load('Ralkingyolo2.png'), pygame.image.load('Ralkingyolo3.png'),
10              pygame.image.load('Ralkingyolo4.png')]
11 # images pour l'animation de marche à droite
12 walkLeft = [pygame.image.load('Pizzayolo.png'), pygame.image.load('Walkingyolo1.png'),
13             pygame.image.load('Walkingyolo2.png'), pygame.image.load('Walkingyolo3.png'),
14             pygame.image.load('Walkingyolo4.png')]
15 # images pour l'animation de marche à gauche
16
17 img = pygame.image.load('Pizzayolo.png')
18 imgmechant = pygame.image.load('Mechantyolo.png')
19 imgprincesse = pygame.image.load('Princesse.png')
20 # image du personnage, de l'ennemi et de la pizza à l'arrêt

```

Ici je demande d'importer les images, que j'ai téléchargé et que j'ai mis dans le dossier du code, grâce à la commande **`pygame.image.load('image.png')`**.

Pour les animations de marche, j'ai créé deux tableau (un pour la marche vers la droite, l'autre pour la marche vers la gauche) qui



rassemble l'ensemble des images de l'animation et qui vont pouvoir être déroulée une par une par la suite.

Maintenant que les images ont été importé, il ne suffisait plus que de les afficher lorsque l'on lance le jeu en utilisant la formule magique **“surface.blit(image, coordonnées)”** :

```
103         surface.blit(imgmechant, (self.x, self.y))
```

Cette commande permet tout simplement d'afficher l'image demandé sur des coordonnées voulu dans le jeu.

Jusque là ce n'était pas compliqué, juste un peu de recherche.

Mais mon objectif suivant était d'importer l'animation, et de faire qu'elle ne se lance seulement quand la touche du clavier adéquate était pressée. Et ce n'était pas facile. Mais avec un peu de recherche sur internet j'ai pu aboutir au code suivant :

```
58         # on prend en compte les touche du clavier pressée et on en déduit quelle animation lancer
59         if keys [pygame.K_RIGHT]:
60             self.left = True
61             self.right = False
62         elif keys[pygame.K_LEFT]:
63             self.right = True
64             self.left = False
65         else:
66             self.right = False
67             self.left = False
68             self.walkCount = 0
69     # on choisi le nombre d'images de l'animation à mettre par pas.
70     if pizzayolo.walkCount + 1 >= 25:
71         pizzayolo.walkCount = 0
72     # on définit les mouvements du personnage et on appelle les images pour l'animation
73
74     if pizzayolo.left:
75         surface.blit(walkLeft[pizzayolo.walkCount // 5], (pizzayolo.x, pizzayolo.y))
76         pizzayolo.walkCount += 1
77     elif pizzayolo.right:
78         surface.blit(walkRight[pizzayolo.walkCount // 5], (pizzayolo.x, pizzayolo.y))
79         pizzayolo.walkCount += 1
80     else:
81         surface.blit(img, (pizzayolo.x, pizzayolo.y))
82         pizzayolo.walkCount = 0
```

J'ai commencé en prenant en compte les touche du clavier grâce à la commande **“keys [pygame.K\_(touche du clavier)]”** et qui permet à l'algorithme de lancer l'animation de marche vers la droite ou la gauche si la personne appuie sur la flèche droite ou gauche.

Ensuite, lorsque l'algorithme a pris en compte le fait que le personnage bouge, il lance l'animation correspondante en utilisant le tableau d'image "walkRight" ou "walkRight". Pour que chacune des 5 image de l'animation se lance à la suite, j'ai créé une variable "walkCount" (qui se traduit par nombre de pas) qui permet de faire défiler les images du tableau.

Mais il y avait un petit problème, les images défilaient trop rapidement. C'est pour cela que j'ai décidé de diviser le temps que les images mettent à défiler par 5.

```
surface.blit(walkLeft[pizzayolo.walkCount // 5], (pizzayolo.x, pizzayolo.y))
```

Enfin, il ne restait plus qu'à faire que l'animation recommence en boucle, et pour cela j'ai fait que lorsque la variable "walkCount" atteint 25 (c'est à dire 5 images \* 5), elle revient à 0 et l'animation recommence.

Une autre partie du code sur lequel j'ai travaillé, est la fonction de saut

```
if not (pizzayolo.isJump):

    # permet au personnage de sauter

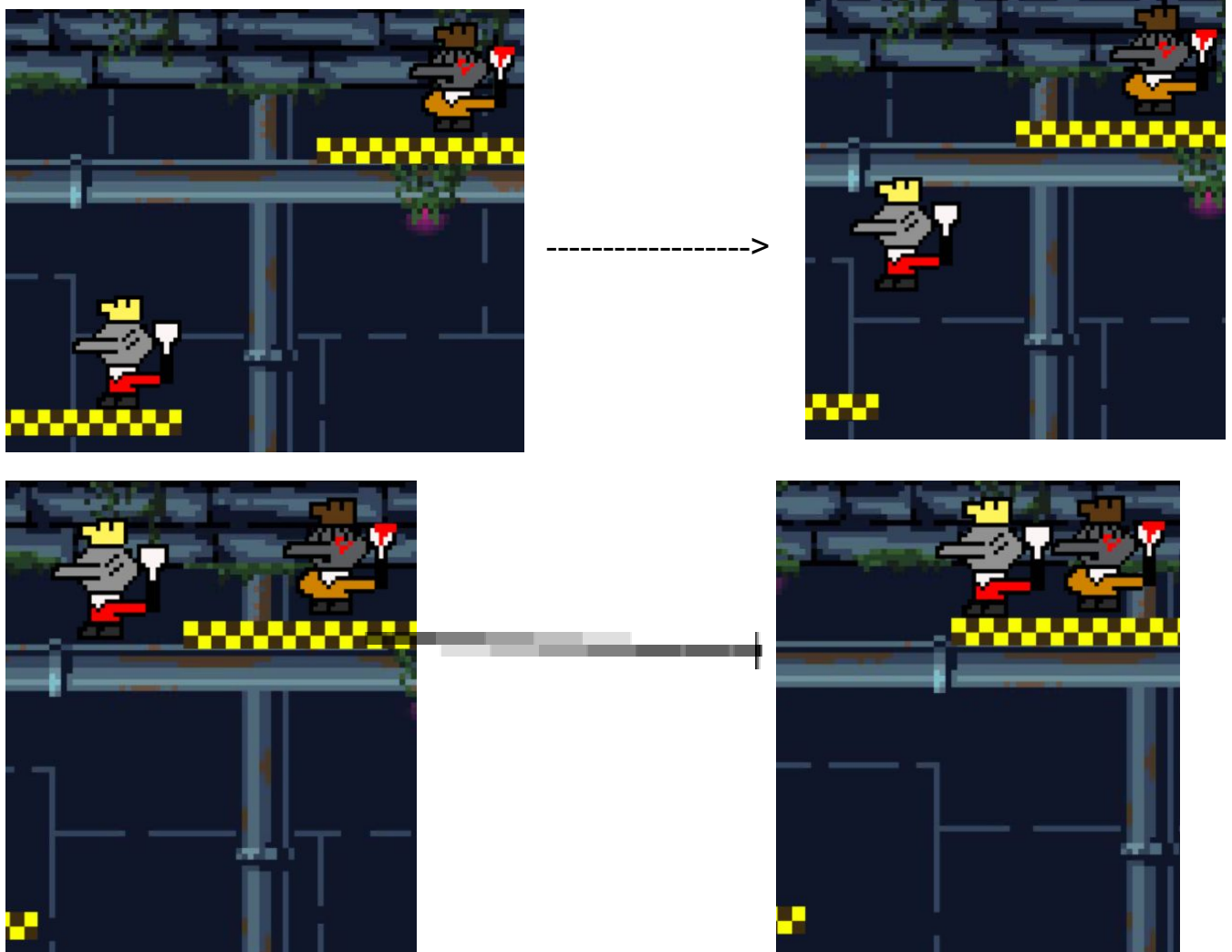
    if keys[pygame.K_SPACE]:
        pizzayolo.isJump = True
        pizzayolo.right = True
        pizzayolo.left = True
        pizzayolo.walkCount = True
    else:
        if pizzayolo.jumpCount >= -12:
            neg = 1
            if pizzayolo.jumpCount < 0:
                neg = -1
            pizzayolo.y -= (pizzayolo.jumpCount ** 2) * 0.5 * neg
            pizzayolo.jumpCount -= 1
        else:
            pizzayolo.isJump = False
            pizzayolo.jumpCount = 13
```

Une fois de plus, je demande au programme de repérer si le joueur appuie sur la barre espace ou non. Et si la personne appuie sur espace alors dans ce cas la fonction saut se déclenche.

Le but de cette fonction saut est tout simplement de faire monter le personnage jusqu'à une certaine distance, puis de le faire redescendre.

Pour cela on utilise une variable "**jumpCount**" qui donne la hauteur du saut. Sauf que en affectant la variable une fois à l'allé et une fois au retour, le saut n'était pas fluide du tout. Ducoup j'ai fais un peu de recherche et j'ai trouvé que l'on pouvait utiliser la formule de l'énergie cinétique ( $0,5*v^2$ ) pour donner un effet d'accélération et ainsi permettre de rendre le saut très fluide. (la variable "**neg**" pour négative, permet changer le signe)

Démonstration :



### III/ Résultat final

#### 1- lancement du jeu



2- Avancer en sautant de plateforme en plateforme tout en évitant les ennemies (à l'aide de la barre espace pour sauter, et des flèches directionnelles pour bouger)



#### 3- Sauver la pizza



Ici on peut voir les différents personnages présents sur l'écran et qui ont tous été dessinés et implantés dans le jeu de mes propres mains.

On peut également voir l'animation de marche en cours et le personnage entrain de sauter.

Et comme prévu, notre jeu respecte le cahier des charges avec notre personnage principale, le pizzaiolo, armé d'une spatule sautant de plateforme en plateforme dans le but de sauver la pizza. Mais bien sûr, le personnage peut également perdre si il tombe dans le vide ou qu'il rencontre un obstacle :





Ce qui résulte à un retour au début du jeu dans le cas de la chute, et à l'arrêt du jeu dans le cas de la rencontre avec un obstacle.

Mais tout cela n'a pas eu lieu d'un seul coup, car il a eu évidemment le besoin de faire des tests. Notamment pour ajuster et réajuster les hitboxes, la hauteur des sauts, la taille des plateformes, la vitesse du personnage.

## **IV ) Bilan personnel et perspective**

### **1) perspective**

Au terme de ces six mois passés sur notre projet, nous pouvons être fier de notre réalisation même si celle-ci n'est pas parfaite, nous avons "appris à apprendre" en découvrant les rudiments de la création d'un jeu.

Pour améliorer ce projet, nous pensons qu'il faudrait créer une fonction rassemblant les caractéristiques des plateformes et ennemis.

Cela nous aurait permis :

- De réduire la taille du code
- De pouvoir faire réapparaître le personnage à sa position initiale suite au contact avec un ennemi.

On peut ajouter que nous avons eu beaucoup de mal à créer le "cœur" du jeu en intégrant la programmation orientée objet. Néanmoins, cela nous a permis de découvrir une nouvelle manière de coder.

Par ailleurs, nous n'avons pas réussi à intégrer une page d'accueil à notre jeu car nous n'avons pas trouvé le moyen d'utiliser Tkinter et Pygame dans un même programme.

## **2) bilan personnel**

En tout cas, cette expérience fût très intéressante, que ce soit dans un cadre collectif ou personnel.

J'ai pu apprendre à utiliser mes compétences en codage et mes recherches sur le sujet pour créer, avec mon équipe, un jeu de plateforme dont je suis fière !

Durant tout ces mois j'ai pu connaître la joie intense de voir que les différentes fonctions que j'ai conçu fonctionnait comme prévu après avoir fais des recherches sur brouillon et/ou sur internet, toujours avec la motivation et le support de mes camarades qui était également présents pour m'assister et me féliciter !

Chaque semaines de nouvelles fonctions étaient implémentées au jeu, et nous avons pu voir le jeu évoluer progressivement pour atteindre le niveau auquel il est désormais. Et chaque semaine, on partageait nos idées et nos avancement, et si quelqu'un avait un problème, on était là pour aider et trouver une solution ensemble, et moi je trouve ça beau !

Sachant que je compte travailler dans l'informatique dans le futur, cette expérience m'est fortement enrichissante, et je le considère comme un bon premier projet sérieux dans le monde de l'informatique, et je suis sûr de m'en souvenir !