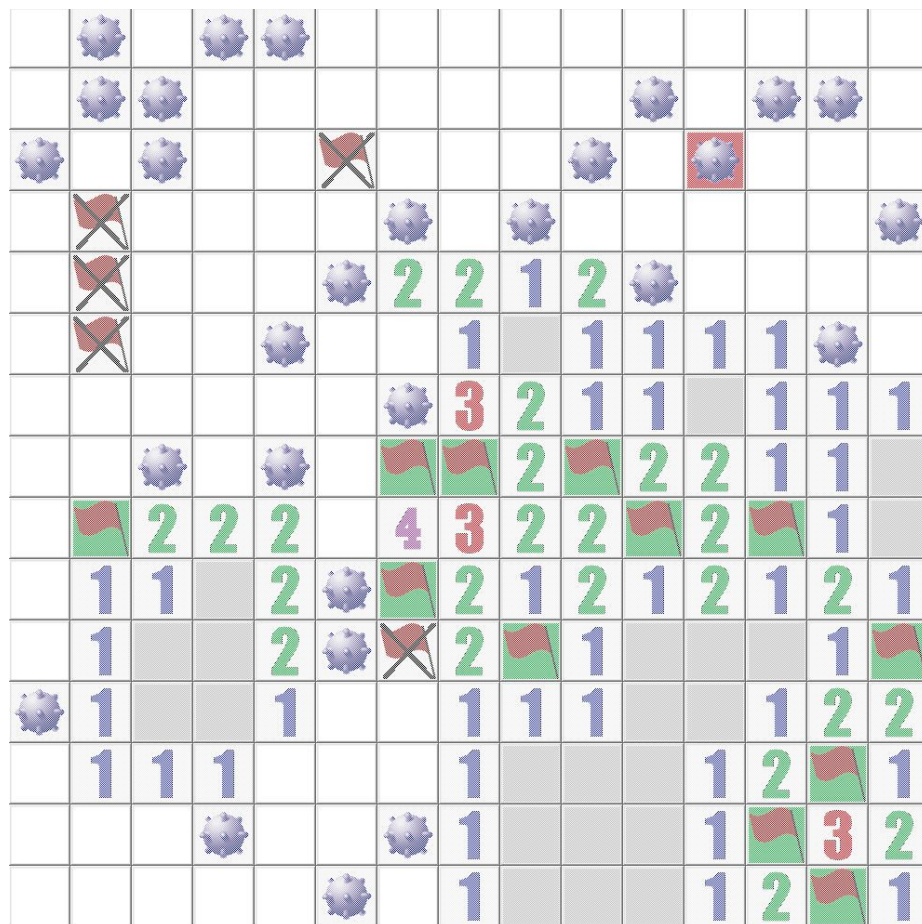


# Projet d'ISN

## D-Mineur



**Raphaël Gradus**, Aristide Leroy et Matéo Boukhobza

29 mai 2019

## Sommaire

1. Sommaire (p 2)
2. Présentation du projet (p 3-4)
3. Cahier des charges (p 5)
4. Répartition des tâches (p 6-7)
5. Réalisation personnelle (p 8-10)
6. Projet final (p 11-12)
7. Points à améliorer sur le projet (p 13)
8. Apport du projet (p 14)

# Présentation du projet

## Pourquoi un démineur ?

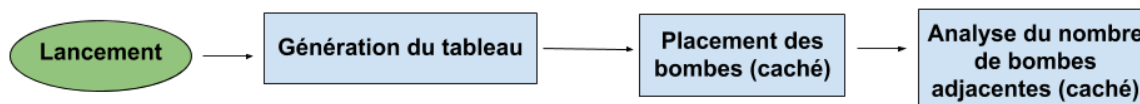
Ce choix ne s'est pas fait directement. En effet, nous avons au début commencé à essayer de faire un algorithme de résolution de rubik's cube. Cependant, au bout de deux semaines, notre faible motivation dans la réalisation de ce projet nous a amené à en changer et opter pour faire un démineur, un projet auquel nous avons déjà pensé auparavant.

Le démineur est un jeu classique, iconique et indémodable auquel nous avons tous déjà joué auparavant, c'est pourquoi nous nous sommes dit que c'était probablement le projet dans lequel nous devions nous lancer.

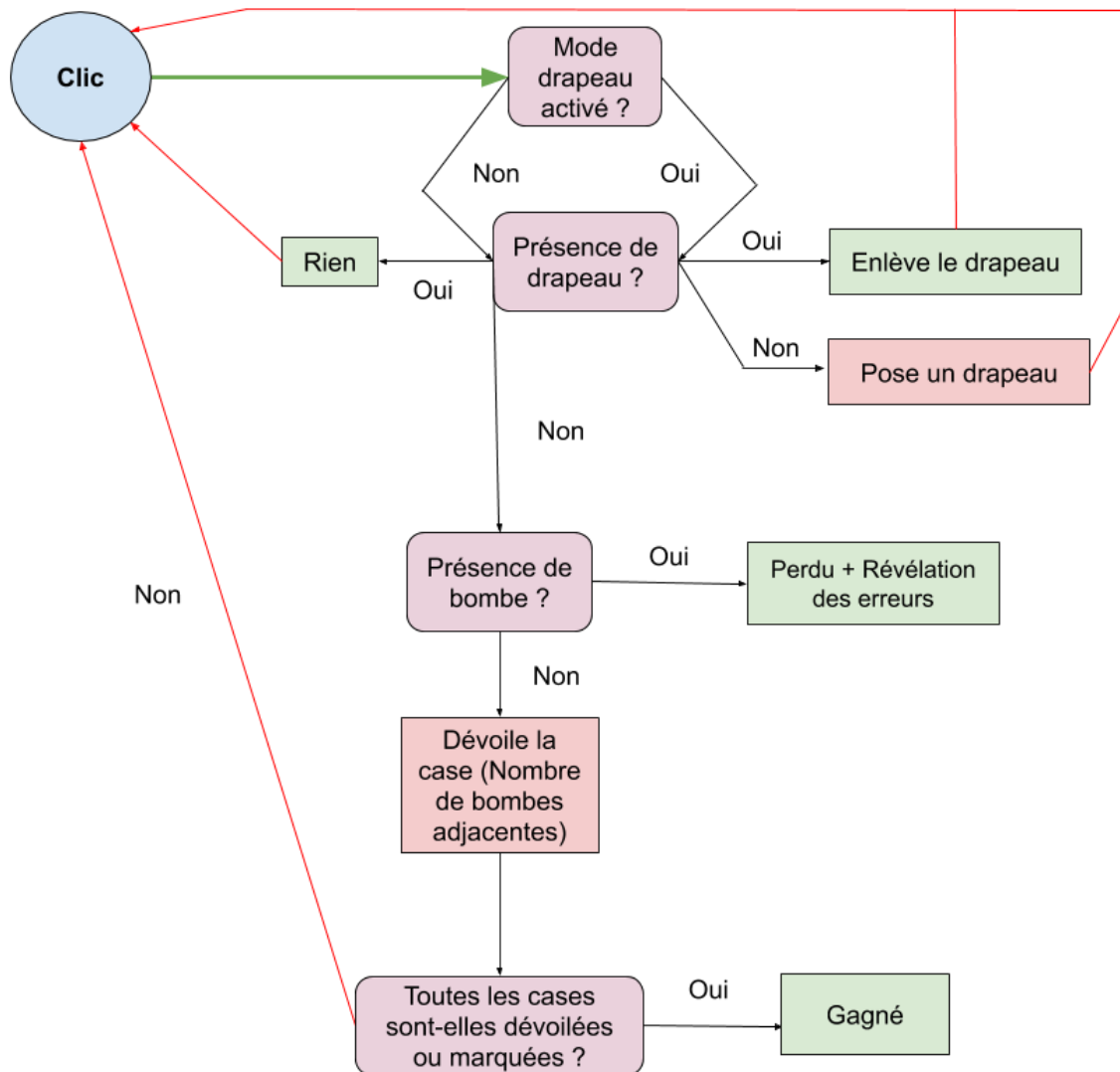
## Quel démineur créer pendant ce projet ?

Nous voulions à tout prix créer un démineur classique tout en y ajoutant une caractéristique supplémentaire qui le démarquera des autres démineurs existants.. En effet, nous voulions apporter à notre démineur une capacité de résolution automatique qui montrerait à l'utilisateur quel mouvement faire à quel moment et pourquoi les faire. Notre programme devait fonctionner en deux parties distinctes : l'initialisation et le fonctionnement en jeu.

Voici deux représentations schématiques du projet que nous voulons réaliser :



### Initialisation

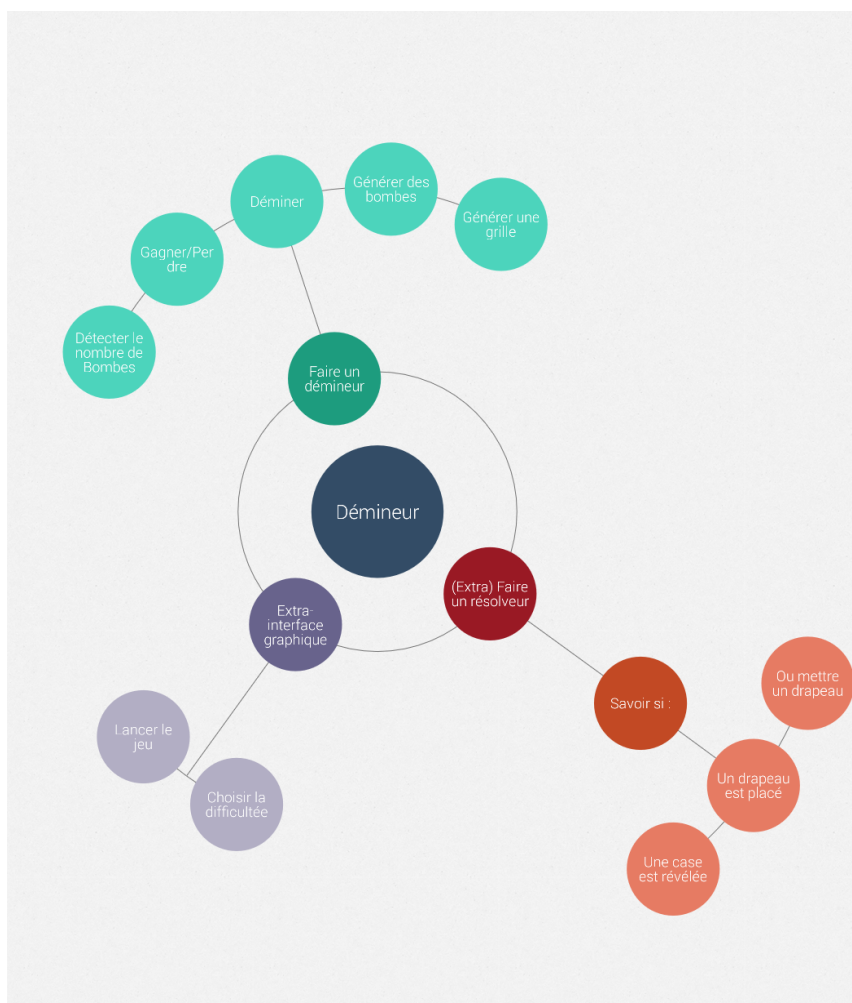


## Fonctionnement de l'algorithme "Clic"

## Cahier des charges

Une fois notre projet choisi et après avoir réfléchi en équipe sur ce que notre démineur devait accueillir comme fonctionnalités pour fonctionner correctement, nous sommes arrivés à en déduire un cahier des charges que nous avons complété au fur et à mesure de l'avancement du projet afin de le perfectionner au maximum.

Nous sommes finalement arrivés à déterminer le cahier des charges ci-dessous :



Ce cahier des charges satisfait parfaitement nos ambitions pour le projet que nous étions sur le point de débiter.

## Partie collaborative :

Avant le début de notre projet, nous nous sommes réparti le travail à faire. Au fur et à mesure que le projet avançait, nous avons été amenés à nous compléter les uns les autres afin que chacun apporte au groupe son savoir-faire pour réaliser au mieux le projet et permettre au groupe d'avancer plus rapidement.

Voici le tableau récapitulatif de ce que chacun a réalisé pendant le projet :

Raphael	Aristide	Matéo
<b>07/02-04/04</b> Entretien du cahier des charges du projet Que va faire notre démineur et comment le faire ?	<b>&lt; février</b> Découverte et compréhension de Tkinter, différents outils, commandes et objets proposés	<b>7 février 2019</b> Création de la grille
<b>14/02-18/04</b> Recherche et compréhension des différents outils de Tkinter et les attributs des objets de Tkinter (aide aux membres de l'équipe)	Disposition des images dans un tableau	<b>14 -&gt; 17 février 2019</b> Génération des bombes
<b>28/02-18/04</b> Commentaires des fonctions et amélioration de la compréhension de ces dernières	Résolution des différents bugs sur le calcul du nombre de bombes adjacentes	<b>v1 : 16-17 février 2019</b> <b>v2 : 23-24 février 2019</b> Générations du détecteur de nombre de bombes adjacentes
<b>07/03-28/03</b> Détection des bugs et résolution de ces derniers (problèmes des cases adjacentes)	Résolution du problème des bombes qui se superposent à la génération du tableau	Semaine avant vacances de fin mars : Récupérer les informations lorsqu'on clique sur une case avec Aristide
<b>21/03-16/05</b> Réalisation de l'interface graphique (menu), inachevée	Résolution du problème : Comment notre programme fait-il pour savoir sur quelle case	<b>21 Mars - 27 Mars</b> Détection des cases sans bombes autour. Partie Résolveur (extra -

	cliquons-nous ?	que l'on a pas eu le temps de finir)
<b>18/03</b> Résolution du problème : Comment rassembler tout les boutons au centre de l'écran (frame)	Développement de la commande de clic	Résolution de divers bug
<b>09/05-20/05</b> Début de résolution du problème : Comment faire démarrer le programme principal à partir du programme du menu.	Développement du "Mode drapeau" pour pouvoir poser des drapeaux sur les cases susceptibles de contenir des bombes	Détection des premières bombes par le résolveur

Pour le travail en groupe et pour échanger plus facilement, nous avons beaucoup utilisé la plateforme en ligne Github pour rester à jour dans nos tâches. Nous avons également utilisé différents outils collaboratifs comme les outils de Google : Drive pour partager des dossiers et y accéder en temps voulu, Docs pour faire du traitement de texte à plusieurs et en simultané. De plus, pour communiquer plus simplement nous avons créé un groupe messenger. Enfin, pour échanger nos nouveautés dans les programmes, il nous arrivait d'utiliser la messagerie de l'établissement : IAL.

# Réalisation Personnelle

## 1) Interface graphique et compréhension de TKinter

Pour ce projet j'ai surtout travaillé sur l'interface graphique. Il a d'abord fallu que je comprenne comment coder une interface graphique, que je m'approprie les différentes fonctions de TKinter. Il m'a fallu du temps pour appréhender la complexité, et je me suis appuyé sur le groupe pour échanger et trouver des solutions. Pour pouvoir comprendre les différentes options que nous présentait TKinter, nous avons lu la documentation sur internet, ce qui fut très utile.

Voici un extrait du code de l'interface que j'ai codé, dans lequel, j'ai utilisé des fonctions de TKinter recherchées au préalable.

```
#titre de la fenêtre
Titre = Label(fenetre, text="D-Mineur", font=("Helvetica", 40), bg='grey', fg='white')
Titre.pack()

#fond d'écran gris pour ne pas faire mal aux yeux la nuit
fenetre.config(background='grey')

#création de la "frame" qui permet de rassembler les niveaux ensembles au milieu de l'écran
boite_Niveaux = Frame(fenetre, bg='grey')

#fonction permettant de lancer le démineur
def play():
    subprocess.run('3.1.py')

#Texte disant de choisir son niveau
Titre_niveau = Label(fenetre, text="Choisissez votre niveau", font=("Helvetica", 30), bg='grey', fg='white')
Titre_niveau.pack(pady= 105)

#Différents boutons demandant à l'utilisateur quel niveau choisir, pas terminé
Niveau1 = Button(boite_Niveaux, text="Niveau 1", font=("Helvetica", 20), bg='white', fg='grey')
Niveau1.grid(pady=5, row=1, column=0, sticky=W)
```

- label() permet d'insérer un texte dans l'interface, on décide ensuite de l'endroit où placer le texte, avec quelle police et quelle taille d'écriture (font), avec quelle couleur de fond et quelle couleur d'écriture
- .config() permet de configurer notre fenêtre, ici, on a décidé de définir la couleur de fond en gris, car cela permet à l'utilisateur d'utiliser le programme la nuit sans



s'abîmer les yeux. (On peut aussi coder les couleurs sous le format hexadécimal en ajoutant un # devant le code)

- `Frame()` permet de créer une sorte de “boîte” rassemblant ensemble plusieurs éléments de l'interface graphique que l'on a décidé de placer dans la grille. C'est comme une sous partie de la fenêtre.
- `subprocess.run()` permet normalement de pouvoir lancer un programme python extérieur en sollicitant une action de l'interface graphique.
- `pady` permet de définir l'écart minimum à garder entre l'élément dans lequel la fonction `pady` est placée et les autres éléments
- `Button()` permet de générer un bouton, comme pour les autres éléments, on peut définir sa police, sa grandeur, son arrière plan etc...
- `.grid()` permet de générer une grille qui va pouvoir servir à placer un élément selon sa colonne et sa ligne.
- `Tk()` permet de générer une fenêtre

Nous avons aussi utilisé d'autres fonctions TKinter mais celles-ci sont les plus importantes.

## **2) Simplification des fonctions**

Pendant le projet, je me suis aussi occupé de simplifier les fonctions et de les rendre plus compréhensibles afin qu'un éventuel “repreneur du projet” puisse comprendre comment fonctionne le démineur et qu'il comprenne comment nos fonctions fonctionnent.

J'ai par exemple donné à des variables des noms plus compréhensibles, j'ai aussi commenté les fonctions pour les rendre compréhensibles.

## **3) Détection et correction des bugs**

J'ai été confronté à de nombreux bugs et au non-fonctionnement de mes fonctions et de celles des membres de mon groupe. J'ai donc appris à détecter et corriger ces bugs. Cela passe d'abord par une analyse du bug. D'où provient-il ? Pourquoi ce bug ? Est-ce seulement une faute de frappe ou une mauvaise utilisation d'une fonction ?

Après avoir analysé, il faut alors détecter la partie du code qui pose problème, et la recorriger.

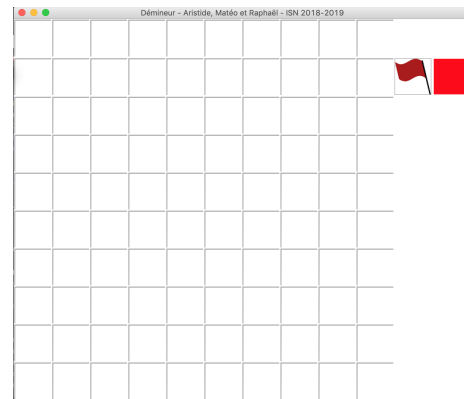
Il peut arriver que le bug soit lié à plusieurs causes. Par exemple, j'ai été confronté à des bugs liés à l'utilisation de la fonction `Frame()`. Une première raison de ce bug venait du fait que j'avais mal positionné le nom de la fonction lorsque j'avais demandé qu'un élément soit compris dans la frame. Ensuite, j'avais mal recopié le nom de la frame, enfin, j'avais oublié le "=" lorsque j'avais défini le nom de la frame. Ces erreurs ne sont pas évidentes à détecter car toutes les parties du code se ressemblent et le code semble correct. Il est presque impossible de détecter instantanément une erreur en regardant le code. Cela m'a beaucoup servi à apprendre à me relire efficacement.

## Résultat final

Après plusieurs mois de travail sur le projet, voici à quoi ressemble la version finale de notre démineur :

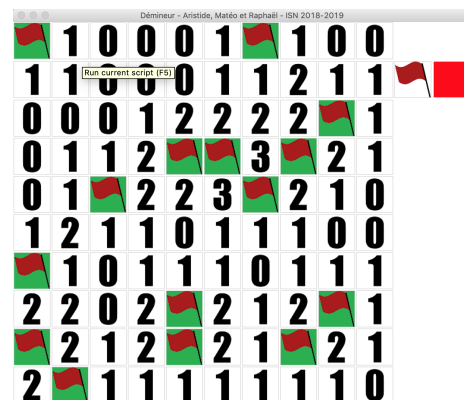
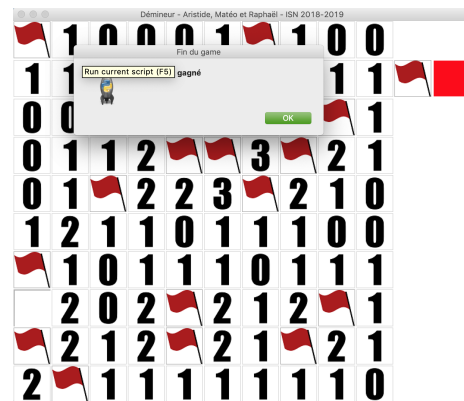
### 1) Début de partie

Lorsqu'on lance le programme python voici la fenêtre qui s'affiche. Un tableau avec un nombre de cases défini par une variable à choisir en lançant le programme.



### 2) L'utilisateur gagne la partie

Lorsque l'utilisateur découvre toutes les cases numérotées et place tous des drapeaux sur toutes les bombes, le message "vous avez gagné s'affiche à l'écran". (il est aussi possible de gagner sans placer les drapeaux et juste en découvrant toutes les cases numérotées mais cela rend le jeu plus compliqué) (photo ci-contre)



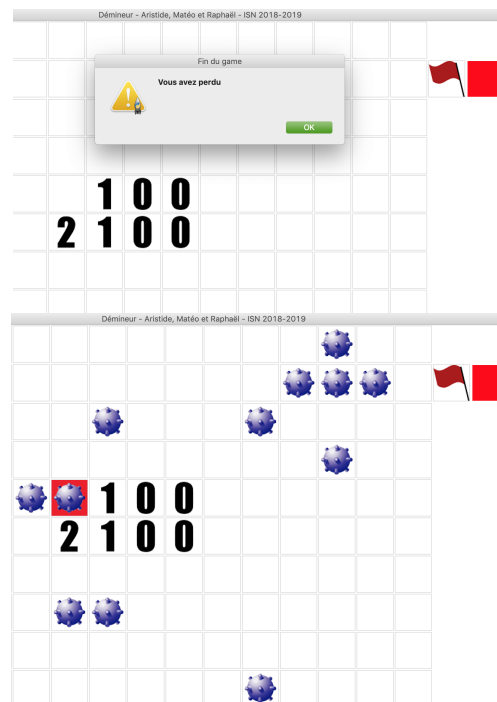
Après avoir cliqué sur "OK", l'écran affiche alors les drapeaux sur fond vert

### 3) L'utilisateur perd la partie

Lorsque l'utilisateur clique sur une bombe alors qu'il n'a pas activé le mode drapeau, le message "vous avez perdu" s'affiche à l'écran. (photo ci-contre)

Après avoir cliqué sur "OK", l'écran affiche alors toutes les bombes qu'il restait à découvrir sur fond blanc ainsi que la bombe qui a été cliquée sur fond rouge. (photo ci-contre)

A noter qu'il est possible de cliquer sur les cases numérotées après la partie.



### 4) Interface graphique

Voici à quoi ressemble l'interface graphique, qui n'est pas fonctionnelle mais terminée.



### 5) Tests à l'usage

Après avoir utilisé le démineur pendant plusieurs jours, je me suis rendu compte que tout fonctionnait convenablement.

## Perspectives futures

Voici quelques points que serait sans doute amenés à reprendre une potentielle future équipe :

- elle devrait d'abord rendre l'interface graphique parfaitement fonctionnelle (lancer le programme lorsqu'on appuie sur jouer, modifier le niveau de difficulté lorsqu'on sélectionne un niveau etc...)
- elle devrait aussi pouvoir créer une fonction permettant de découvrir toutes la cases 0 collées les une aux autres pour ne pas avoir à les découvrir chacune leur tour.
- elle devrait aussi sans doute améliorer le design du programme. Pourquoi pas donner une meilleur couleur de fond que le blanc, mieux intégrer le "menu drapeau" dans la fenêtre etc...?
- enfin, elle pourrait aussi adapter le programme à différentes taille d'écrans et pourquoi pas aussi développer une application pour smartphones.

## Bilan personnel

J'ai beaucoup aimé travailler sur ce projet en groupe. Il m'a permis de voir et d'apprendre ce que en quoi consistait vraiment les mots "travail de groupe". C'est à dire apporter ses connaissances et ses qualités au groupe et au projet. J'ai pu ainsi être utile au groupe même si mes camarades maîtrisent bien plus que moi certaines utilisations des fonctions.

De plus, j'ai pu me familiariser avec le langage informatique. Les compétences que j'ai acquises cette année me serviront beaucoup dans le futur. En effet, l'année prochaine, j'ai décidé de poursuivre mes études dans une école d'ingénieur du numérique à l'ISEP. Même si le programme de l'école prévoit de reprendre à zéro l'apprentissage du langage informatique, je suis convaincu que mes connaissances m'aideront beaucoup pour bien débiter l'année prochaine.