

SOMMAIRE

I-Présentation du projet

II-Cahier des charges du projet

III-Répartition des tâches

IV-Réalisation du projet

V-Améliorations éventuelles

VI-Bilan personnel

VII-Annexe

I-Présentation du projet

Pour certains, la partie la plus dure, lorsqu'on s'inscrit dans une auto-école afin d'obtenir le permis de conduire, est l'examen théorique, c'est-à-dire le code de la route. Nous avons donc décidé de créer un test au code de la route pour aider les personnes comme nous afin qu'ils puissent s'entraîner à l'examen au code de la route. En effet nous avons intégré à notre test des questions avec au minimum deux choix de réponse, ainsi que la réponse correspondante à cette question puis son explication, et enfin pour illustrer le test, nous avons rajouté une image qui est en rapport avec la question.

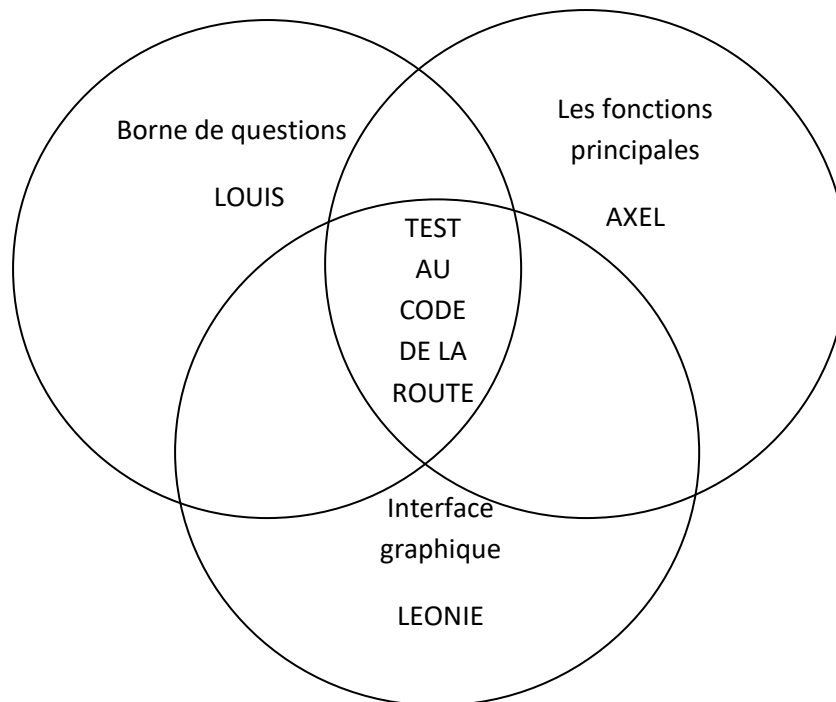
Nous pouvons dire que notre projet est éducatif car il a pour but d'entraîner l'utilisateur du test et lui permet d'apprendre le code de la route. Le projet sert donc de simulateur de série de code.

II-Cahier des charges du projet

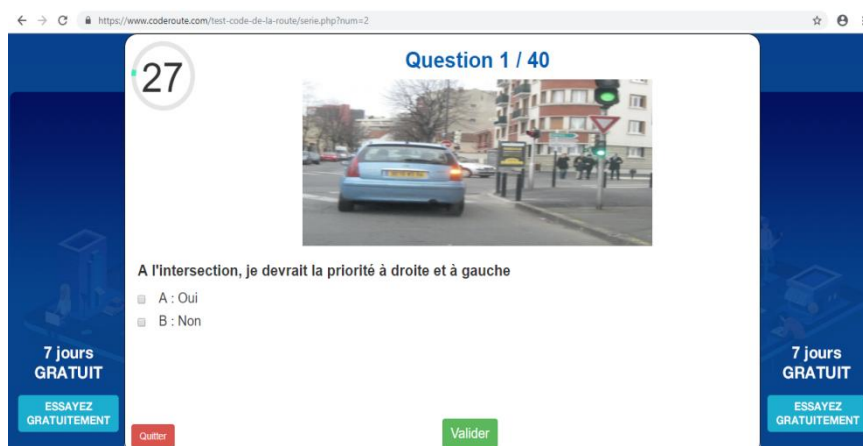
Notre programme doit pouvoir :

- Poser plusieurs questions avec au minimum deux choix de réponses grâce à une borne de questions sans qu'elle ne réapparaisse lors d'une même session de questions
- Savoir quelles sont les bonnes réponses parmi les questions
- Comptabiliser le nombre de bonnes ou mauvaises réponses et de donner un total à la fin de la session
- Afficher une image ou une vidéo correspondant à la question posée
- Expliquer pourquoi cette réponse
- Avoir un temps pour pouvoir répondre comme dans le vrai examen
- Etre sous forme d'appli à la fin du projet

III-Répartition des tâches



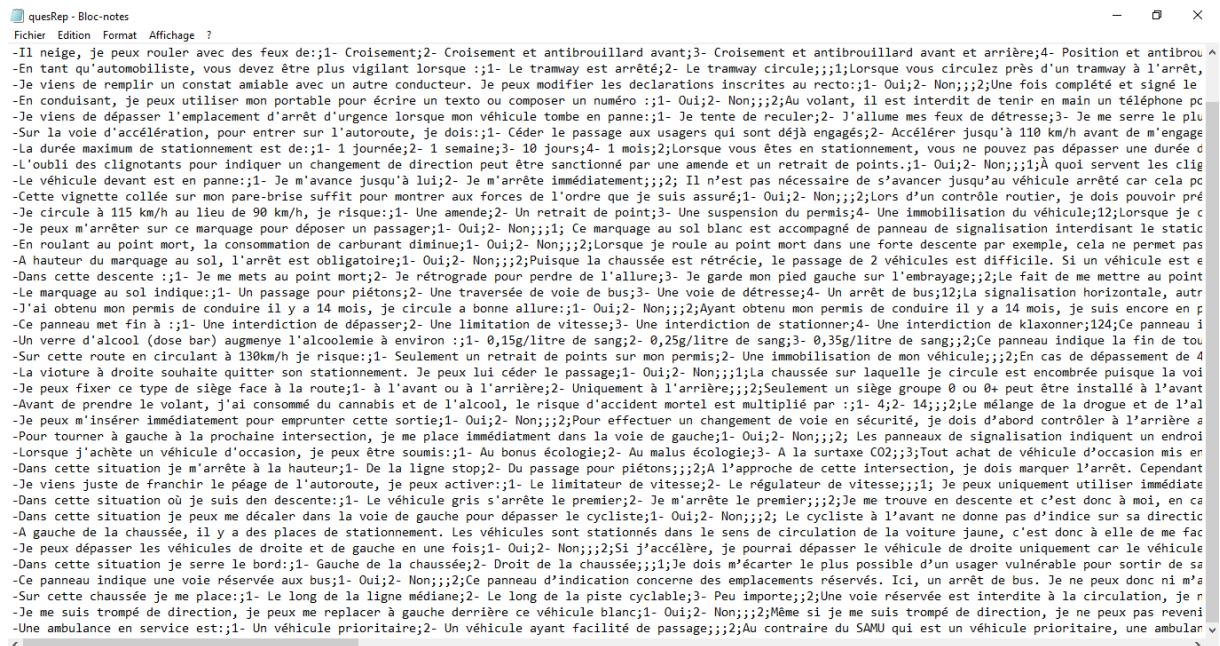
IV-Réalisation du projet



Il nous a fallu tout d'abord trouver des questions, leurs réponses possibles, les bonnes réponses correspondantes ainsi que leurs images. Nous avons trouvé tout ceci majoritairement sur le site, www.coderoute.com.

Ensuite, nous avons réalisé la programmation de notre projet avec les logiciels Python et Thonny

Premièrement, nous avons créé une borne de question pour que notre programme puisse « piocher » dedans. Il y a dans cette liste : les questions, les réponses possibles à cette question, un numéro correspondant à la position qu'a la bonne réponse dans la liste, puis une explication quant au choix de la réponse.



Exemple de la borne de questions

```
def lectureQuesRep(borneQuesRep):
    """Ouvre le fichier des bornes de repo
    fichier = open(borneQuesRep, "r")
    listeQuestion = []
    for ligne in fichier:
        listeQuestion.append(ligne)
    return listeQuestion
```

Par la suite nous avons créé une fonction qui permet d'ouvrir le fichier afin que le programme puisse accéder à la borne de questions.

Puis nous avons écrit une autre fonction qui ordonne au programme de choisir aléatoirement un élément de la liste et la renvoie.

```
def aleaQues(borneQuesRep):
    """choisi un élément aléatoire de la liste et
    listeQuestion = lectureQuesRep(borneQuesRep)
    random.shuffle(listeQuestion)
    return listeQuestion
```

```
def choixQues(paramQues):
    """renvoie une liste dont chaque
    question = paramQues.split(';')
    return question
```

Ensuite la troisième fonction permet de renvoyer une liste dont chaque élément correspond à un paramètre, comme la question, les choix de réponses, la bonne réponse ainsi que les explications associée et son image correspondante.

Afin de pouvoir afficher la question nous avons ajouté une fonction le permettant et deux variable « if » car les questions ont au minimum deux réponses au choix mais peuvent en posséder trois ou quatre.

```
def affichage(question):
    """affiche la question et
    print(question[0])
    print(question[1])
    print(question[2])
    if question[3] != "":
        print(question[3])
    if question[4] != "":
        print(question[4])
```

```
def affichage(question):
    """affiche la question et
    print(question[0])
    print(question[1])
    print(question[2])
    if question[3] != "":
        print(question[3])
    if question[4] != "":
        print(question[4])
```

La fonction « afficheRep » affiche si la réponse est juste ou fausse et elle ajoute l'élément numéro 6 de la liste qui est l'explication à la question. Et la fonction « testReponse » test la réponse pour savoir si elle est bonne ou non.

De plus, pour pouvoir compte les points, le programme possède une fonction qui lorsque la réponse est fausse ajoute un point au compteur.

```
def pointsrepFausses(reponse, repFausses):
    """si la reponse a la question posé est faus
    if testReponse(question,reponse) == False:
        repFausses = repFausses+1
        return repFausses
    else:
        repFausses = repFausses
        return repFausses
```

V-Amélioration éventuelles

Notre projet final répond en majorité au cahier des charges mais le programme ne s'exploite qu'avec la console Python. En effet, nous aurions voulu faire un programme plus jolie visuellement mais nous avons rencontré quelques problèmes avec Tkinter, donc le projet n'est pas fonctionnel avec l'interface graphique. Autrement dit, nous n'avons pas pu intégrer de compte à rebours ainsi que des vidéos des questions et de rendre interactif le programme sous forme d'application par manque de temps et de compétence. Toutefois, sur la petite partie de Tkinter affichant le projet nous avons pu intégrer les images correspondantes aux questions.

VI-Bilan personnel

La spécialité ISN m'a permis de découvrir la programmation, l'interface graphique ainsi que le codage HTML. Grâce au projet que nous avons créé, j'ai pu mettre en application ce que nous avons appris tout au long de l'année et donc de développer mes capacités en informatique. De plus, le travail de groupe qui s'est formé dans l'année pour mener bien ce projet, m'a apporté beaucoup d'atouts tels que la coopération l'entente, l'échange avec les différents membres du groupe. La programmation est la partie qui a été la plus compliquée à élaborer pour moi car cela demande une réflexion plus poussée. Enfin, j'espère que tout ce que j'aurais appris durant l'année grâce à mes professeurs, me servira dans mes années futures notamment pour mes études supérieures.

Gueritas Léonie

Le projet d'ISN sur le thème du test du code de la route était pour moi intéressant puisqu'il mêle le côté pratique et éducatif puisqu'on pouvait réviser son code de la route ou bien même s'entraîner tout en travaillant sur de la programmation. Ce projet qui est pour moi original m'a permis d'en apprendre un peu plus et donc d'approfondir mes connaissances en langage Python et par conséquent en apprendre un peu plus sur la programmation. Ce projet m'a également permis d'avoir plus de faciliter sur les différentes fonctions que l'on peut trouver dans un programme.

Ce projet était aussi un travail autonome en groupe donc cela m'a permis de travailler de façon plus indépendante sans l'aide et le support des professeurs. De plus, le fait que ce soit un travail en groupe nous avons dû faire face à de l'entre-aide mais principalement nous avons appris à travailler en groupe ce qui sera bénéfique dans la suite des études.

Enfin, l'ISN est une nouveauté pour ma part et j'avais une base très faible en programmation et ce projet m'a permis d'enrichir cette base et de réaliser un programme fonctionnel. Je suis donc doté de connaissances en programmation notamment dans le langage Python même si celle-ci ne me seront peu utile dans mes futures études, l'ISN reste toute fois pour moi une bonne expérience.

Lebourg Axel

Cette année en informatique et sciences du numérique m'a permis d'approfondir mes connaissances en programmation et notamment dans le langage Python et Tkinter, qui sont des domaines dans lesquels je ne m'étais encore jamais exercé. Durant la création de notre projet, j'ai appris de nombreuses choses telles que la mise en relation entre un programme Python avec d'autres fichiers comme des fichiers textes ou des photos afin de les afficher avec Tkinter. J'ai également appris à créer divers types de fonctions, comme par exemple une qui permet d'ajouter de l'aléatoire à notre programme ou encore à comparer deux éléments pour que notre programme s'adapte à la situation.

Travailler sur ce projet m'a été très bénéfique, car il renforce le travail d'équipe tout en nous laissant une part d'autonomie, le tout de manière éducative. De plus le fait de voir notre projet avancer et prendre forme petit à petit apporte de la satisfaction et donne envie d'aller jusqu'au bout de nos ambitions.

Pour finir je tiens à dire que l'option ISN est une matière très intéressante et vivante car on y retrouve aussi bien de la théorie que de la pratique, nous

permettant d'évoluer dans le domaine de l'informatique et de la programmation.

Déon Louis

VII-Annexe

Version du programme non fonctionnel sur Tkinter :

```
• paramQues = listeQuestion[numQuestion-1]
• question = choixQues(paramQues)
•40 listeQuestion.remove(paramQues)

def changePhoto():
    """affiche une image"""
    global image, photo
    nomImg = question[7]
    image = Image.open(nomImg)
    photo = ImageTk.PhotoImage(image)
    canvas = tk.Canvas(window, width = image.size[0], height = image.size[1])
    •50 canvas.create_image(0,0, anchor = tk.NW, image=photo)
    canvas.pack()

def afficheQues():
    """affiche la question et les propositions"""
    q = tk.Label(window, text=question[0])
    q.pack()
    a = tk.Label(window, text=question[1])
    a.pack()
    •60 b = tk.Label(window, text=question[2])
    b.pack()
    if question[3] != "":
        c = tk.Label(window, text=question[3])
        c.pack()
    if question[4] != "":
        d = tk.Label(window, text=question[4])
        d.pack()
    reponse = tk.Entry(window, width=40)
    reponse.pack()
    •70 validation = tk.Button(window, text="Confirmer", command=changeQues)
    validation.pack()

def changeQues():
    import random
    •2 from random import randint
    import tkinter as tk
    from PIL import Image, ImageTk

    def lectureQuesRep(borneQuesRep):
        """Ouvre le fichier des bornes de réponses et renvoie une liste dont chaque élément correspond à une question et ses paramètres"""
        •10 fichier = open(borneQuesRep, "r")
        listeQuestion = []
        for ligne in fichier:
            listeQuestion.append(ligne)
        return listeQuestion

    def aleaQues(borneQuesRep):
        """choisi un élément aléatoire de la liste et le renvoie"""
        listeQuestion = lectureQuesRep(borneQuesRep)
        random.shuffle(listeQuestion)
        •20 return listeQuestion

    def choixQues(paramQues):
        """renvoie une liste dont chaque élément correspond à un paramètre"""
        question = paramQues.split(';')
        return question

    def consigne():
        •30 """si le bouton est actionné, écrit les consignes"""
        t = tk.Label(window, text="Pour répondre écrivez le numéro de votre réponse, si vous avez plusieurs choix à faire ne pas mettre d'e")
        t.pack()

    def choixQuestion():
        global numQuestion, paramQues, question
        numQuestion = randint(1, len(listeQuestion))
```

```

• - global window
• - window.destroy()
• - window = tk.Tk()
• - choixQuestion()
• - changePhoto()
•80 - affichageQues()
• -
• - borneQuesRep = "quesRep.txt"
• - nbQues = 0
• - repFausses = 0
• - lectureQuesRep(borneQuesRep)
• - listeQuestion = aleaQues(borneQuesRep)
• -
• - window = tk.Tk()
• - choixQuestion()
•90 - changePhoto()
• - affichageQues()
• -
• - window.mainloop()

```