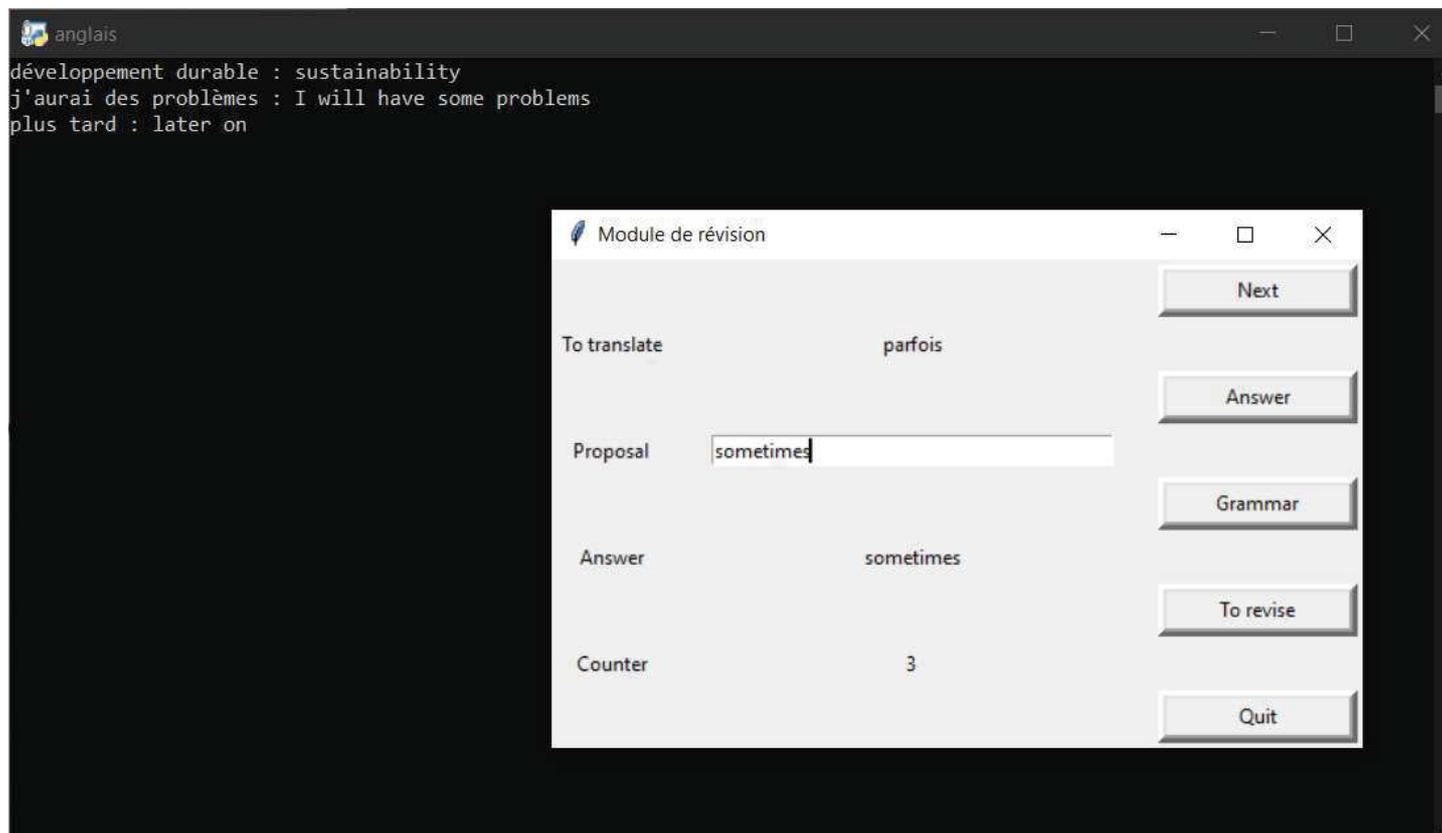


Développement :

Petit module graphique destiné à améliorer le niveau d'anglais de l'utilisateur



Cadre :

Il s'agit d'un projet personnel : mon niveau d'anglais était extrêmement bas, n'ayant jamais eu l'occasion de le pratiquer lors de ma 1ère vie professionnelle, or cette langue est indispensable au métier de développeur.

Support :

Python et l'Idle de Python

Contraintes :

Il était important que l'aspect graphique soit attrayant sans être trop chargé.
Je souhaitais aussi que les interrogations sur le vocabulaire soient totalement aléatoires.

Difficultés rencontrées :

La manipulation du dictionnaire en Python, avec la fonction random, m'a posé quelques soucis : j'ai dû passer par des listes pour la sélection du mot aléatoire

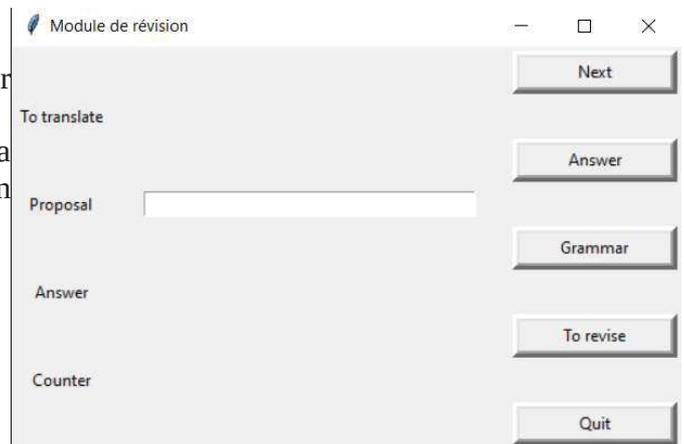
Description détaillée du développement

1. Choix généraux :

J'ai choisi de développer ce projet dans une seule classe, qui gère à la fois l'aspect graphique grâce à la bibliothèque tkinter, et les différentes fonctionnalités.

2. Design de la fenêtre graphique :

Volontairement sobre, il m'a paru approprié d'utiliser l'anglais pour le texte des boutons et des labels. Le code correspondant est dans le constructeur de la petite appli : la coloration du code permet d'identifier en un coup d'oeil les différents objets graphiques.



```
def __init__(self) :
    Frame.__init__(self)
    self.master.title('Module de révision')

    Button(self, text="Next", width=15, borderwidth=5, relief='raised',
           command=self.another).grid(row=0, column=2, padx=3, pady=3)
    Button(self, text="Answer", width=15, borderwidth=5, relief='raised',
           command=self.answer).grid(row=2, column=2, padx=3, pady=3)
    Button(self, text="Grammar", width=15, borderwidth=5, relief='raised',
           command=self.grammar).grid(row=4, column=2, padx=3, pady=3)
    Button(self, text="To revise", width=15, borderwidth=5, relief='raised',
           command=self.revise).grid(row=6, column=2, padx=3, pady=3)
    Button(self, text="Quit", width=15, borderwidth=5, relief='raised',
           command=self.boutQuit).grid(row=8, column=2, padx=3, pady=3)

    Label(self, text="To translate").grid(row=1, column=0, padx=3, pady=3)
    self.lab_trans = Label(self, width=40)
    self.lab_trans.grid(row=1, column=1, padx=3, pady=3)

    Label(self, text="Proposal").grid(row=3, column=0, padx=3, pady=3)
    self.case_proposal = Entry(self, width=40)
    self.case_proposal.grid(row=3, column=1, padx=3, pady=3)

    Label(self, text="Answer").grid(row=5, column=0, padx=3, pady=3)
    self.lab_ans = Label(self, width=40)
    self.lab_ans.grid(row=5, column=1, padx=3, pady=3)

    Label(self, text="Counter").grid(row=7, column=0, padx=3, pady=3)
    self.lab_counter = Label(self, width=15)
    self.lab_counter.grid(row=7, column=1, padx=3, pady=3)

    self.list_dictio = []
    self.dictio_revis = {}
    self.c = 0

    self.liste_fiches = ["Les adverbes.pdf", "Futur (will - going to).pdf",
                        "Les pronoms et déterminants possessifs.pdf",
                        "Les pronoms réfléchis.pdf", "Politesse.pdf",
                        "Les verbes modaux.pdf", "Comparatif - superlatif.pdf",
                        "Nombres Ordinaux.pdf", "Past Perfect.pdf", "Either.pdf",
                        "Phrase avec IF.pdf", "Propositions relatives.pdf",
                        "Avec ou sans article.pdf", "Used to.pdf", "Shall.pdf",
                        "Autres utilisations ING.pdf", "Past perfect progressive.pdf",
```

En parallèle, j'ai un fichier dictio.py dans lequel j'entre mon vocabulaire au fur et à mesure. Il ressemble à ça :

```
voca = {'en tout': 'altogether',
'comment vas-tu ?': 'how are you ?',
'Et toi ?': 'and how about you ?',
'Je viens de France': "I'm from France",
'tu viens des USA': "you're from USA",
'et bien, bon vol !': 'well, have a good flight !',
"je viens d'Australie": 'I come from Australia',
"je prends un jus d'orange": "I'll have an orange juice",
"puis-je avoir l'addition SVP ?": 'can I have the bill please ?',
'combien coûte le gâteau ?': 'how much is the cake ?',
'un café coûte 2 €': 'a coffee costs 2 €',
'3 eaux minérales': '3 mineral waters',
"qu'est-ce que tu veux boire ?": 'what would you like to drink ?',
'je voudrai 9 petits gâteaux au chocolat': "I'd like nine small chocolate cakes",
'alors': 'then',
'quand es-tu libre ?': 'when are you free ?',
'voyons-nous': "let's meet",
'je te vois vendredi': 'see you on Friday',
'aller courrir': 'to go running',
'jouer au tennis': 'to play tennis',
'aller nager': 'to go swimming',
'on se voit bientôt': "let's meet soon",
'il est 8h (pile)': "it's eight o'clock",
'midi': 'midday',
'minuit': 'midnight',
'quart': 'quarter',
'il est 8h55': "it's five to nine",
'es-tu libre ce soir ?': 'are you free tonight?',
'je pense que oui': 'I think so',
'et si on allait au cinéma ?': 'how about going to the cinema ?',
'à quelle heure on se voit ?': 'what time shall we meet ?',
'à tout à l'heure': 'see you then',
'devant le cinéma': 'outside the cinema',
"qu'est-ce que tu veux faire": 'what do you want to do ?',
"que dirais-tu d'aller nager ?": 'how about going swimming ?',
"à quelle heure veux-tu qu'on aille courrir ?": 'what time shall we go running ?',
"que dirais-tu d'aller manger ?": 'how about having diner ?',
"à quelle heure veux-tu qu'on se retrouve ?": 'what time shall we meet ?',
'je te vois à ce moment là': 'see you then',
'sa maman et son papa (à elle)': 'her mum and dad',
'elle est sa cousine': "she's his cousin",
'tante': 'aunt',
```

2. Démarrage de l'application - Compteur

Il faut cliquer sur le bouton “ next ” : un mot, en français, s’affiche alors, il faut proposer une traduction en anglais dans le cadre idoine.

Ce code est placé dans 2 méthodes, la méthode “ another ” pour le clic sur le bouton “ next ”, et la méthode “ answer ” pour le clic sur le bouton du même nom.

```
def another(self) :
    "Proposition d'un mot/d'une phrase en français, à traduire"
    self.lab_ans.configure(text='')

    self.list_dictio = list(voca.items())
    self.list_dictio = choice(self.list_dictio)

    self.lab_trans.configure(text=self.list_dictio[0])

def answer(self) :
    """Traduction du mot/de la phrase choisie, incrémentation du nbre
    d'erreur, et ajout des éléments à revoir au dictionnaire"""
    self.lab_ans.configure(text=self.list_dictio[1])

    proposal = self.case_proposal.get()

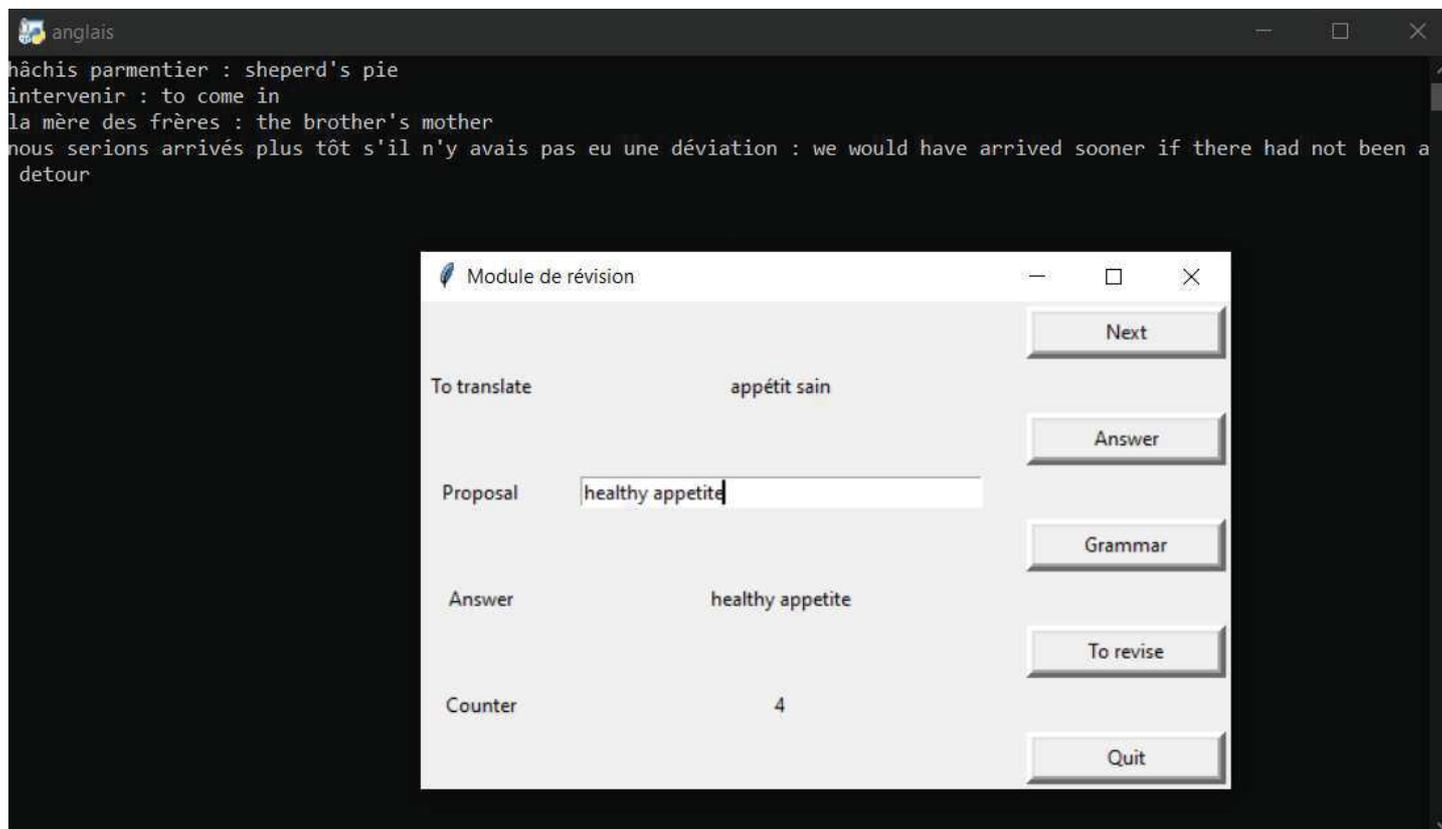
    if proposal != self.list_dictio[1] :
        self.dictio_revis[self.list_dictio[0]] = self.list_dictio[1]
        self.c += 1
        self.lab_counter.configure(text=str(self.c))
```

Afin de ne pas laisser défiler à l’infini des mots qu’on ne saurait pas traduire, j’ai mis un compteur de ‘mauvaises réponses’ : ainsi le compteur s’incrémente au fur et à mesure, cela permet donc d’évaluer en temps réel nombre de mots/phrases proposés et non maîtrisés.

3. Résumé vocabulaire

Il est ensuite aisé d'obtenir, en un clic sur le bouton " To revise ", le vocabulaire qui a fait l'objet d'erreurs :

```
def revise(self) :  
    """Affichage des éléments à revoir"""  
    for fran,ang in self.diction_revis.items() :  
        print(fran, ":", ang)
```



4. La grammaire

En cliquant sur le bouton " Grammar ", on accède à des fiches de grammaire (là encore de manière aléatoire), pour lesquelles les noms ont été préalablement renseignés dans le constructeur.

```
def grammar(self) :  
    "Affiche une fiche de révision conjugaison/grammaire au hasard"  
    choix = choice(self.liste_fiches)  
    startfile(choix)
```

Donc ensuite, il n'y a plus qu'à...

```
if __name__ == "__main__" :  
    appli = Appli_Revis()  
    appli.mainloop()
```