Projet NSI

Application d’Horloge

Louis Guerrero – 1ère8

What : Mon projet est une application d’horloge, permettant évidemment de donner l’heure mais avec aussi la possibilité d’utiliser un chronomètre, un minuteur et de programmer des alarmes.

For whom : Ce projet n’a pas de public cible et donc destiné à tout type de personnes.

Why : et pourquoi pas ? (Je n’ai pas de réelle raison, je n’avais pas d’idée de projet donc j’ai pris la première chose qui m’est passé par la tête.)

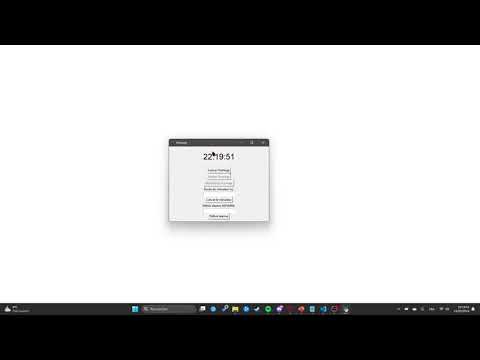
How : Utilisation du langage python et de multiples modules tels que les modules *time* (qui permet de gérer la notion au sein d’un programme python) ou encore *tkinter* (permet d’afficher le programme dans une interface graphique). Les principaux environnements utilisés sont *Capytale (Basthon)* et *Visual Studio Code*.

J’ai longuement hésité entre diverses bibliothèques graphiques pour ce projet, notamment *p5* et *tkinter*. J’ai finalement choisi d’utiliser *tkinter*. Cependant, à cause des limitations de *Basthon*, j’ai dû basculer mon projet sur *Visual Studio Code*, car *Basthon* ne permet pas d’utiliser le module *tkinter* à cause des limitations du navigateur.

When : J’ai tout d’abord programmé sans utiliser d’interface graphique, puis j’ai décidé d’utiliser *tkinter* comme bibliothèque pour pouvoir afficher le programme dans une fenêtre.

J’ai commencé par programmer l’horloge principale, puis j’ai programmé les alarmes et le minuteur et j’ai terminé par le chronomètre. J’avais d’abord programmé les différentes parties séparément, puis j’ai tout mis dans un seul et même programme.

Démonstration :

[](https://www.youtube.com/embed/wwQkHwgp6sw?feature=oembed)

Partie de code commenté :

Une image contenant texte, Logiciel multimédia, Police, capture d’écran

Description générée automatiquement

Une image contenant texte, Police, capture d’écran, blanc

Description générée automatiquementLigne 105 : On définit ici une fonction *set\_alarme* qui permet à partir d’une entrée (qui se trouve dans la fonction *\_\_init\_\_* de la classe) de définir le temps au quel l’alarme devra sonner dans la variable *alarm\_tps*. Cette fonction est rattachée au bouton « Définir alarme » (voir image ci-contre). La ligne 108 permet de vérifier si le format d’heure entré dans la boite de dialogue est correct (si le format n’est pas HH:MM, la fonction renvoi un message d’erreur grâce aux lignes 109 et 110)

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquementLigne 112 : On définit ici une fonction qui va permettre de faire sonner l’alarme. On vérifie d’abord si l’alarme a déjà été lancée grâce à la variable *alarm\_triggered* (qui est *False* si l’alarme n’a pas déjà été lancée et *True* si elle a déjà été lancée. La ligne 114 permet d’afficher l’alarme : un message qui dit « Alarme ! » (Voir image ci-contre)

Ce qu’il reste à faire :

Ce projet a encore besoin d’un peu de travail, notamment pour incorporer le chronomètre dans l’interface graphique, retravailler le minuteur qui ne fonctionne pas très bien ou encore ajouter la possibilité de prendre en compte la date et pas seulement l’heure, qui permettrait donc par exemple de programmer des alarmes autre que pour le jour actuel.

Il faudrait enfin travailler l’apparence de l’interface graphique. On pourrait également ajouter de nouvelles options, comme par exemple la possibilité de changer l’heure, le fuseau horaire, mais aussi d’afficher l’heure dans d’autres pays.

Conclusion :

En conclusion, ce projet est un projet assez simple et qui sert à tout le monde mais qui a encore besoin de travail pour être rendu plus intéressant au sein du grand public. Il faudrait notamment ajouter de nouvelles options et retravailler l’apparence de l’application.