

Projet n°3 : Dessiner une rue aléatoire

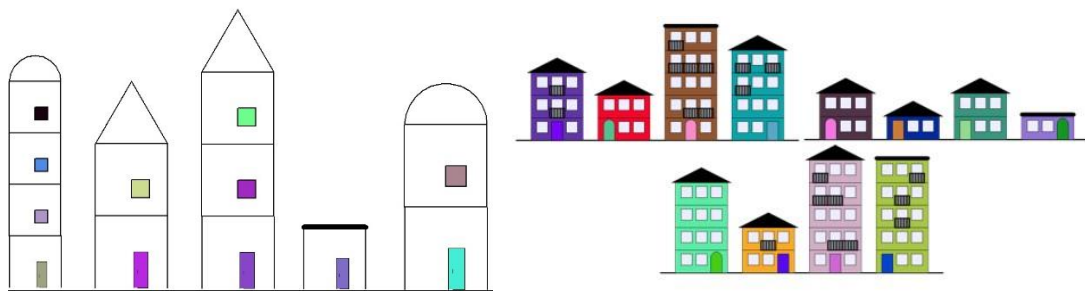
1^{ère} NSI – Collège Episcopal Saint Etienne

Objectifs

- Progresser en méthodologie de projet
- Travailler en équipe : collaborer et répartir les tâches
- Mettre en application les connaissances et compétences travaillées en classe (constructions élémentaires, programmation en langage Python)
- Présenter son travail à l'oral
- Découvrir un module (turtle et random), utiliser une documentation, rechercher des ressources pour progresser et surmonter ses difficultés
- Répondre aux causes typiques de bugs, documenter son code

But du projet

Créer un programme dessinant une rue, composée d'immeubles aléatoires.



Niveau 1

- Dessiner un immeuble comportant entre 1 et 5 niveaux, coloré, avec une porte d'entrée, des fenêtres et un toit.
- Faire en sorte que le nombre d'étages de cet immeuble ainsi que sa couleur soient aléatoires
- Dessiner une rue de 5 immeubles aléatoires

Niveau 2

- Demander à l'utilisateur le nombre d'immeubles dans la rue
- Ajouter des variantes dans les formes et couleurs de porte, de toit, voire d'autres éléments de votre choix (balcons ou autres), qui seront sélectionnés aléatoirement pour chaque immeuble

Niveau 3 : Bonus

- Faire preuve d'imagination pour agrémenter la rue (décor, autres bâtiments, etc.)

Indications

Modules

Vous aurez besoin d'utiliser 2 modules :

- turtle : un module permettant de dessiner. Documentation : <https://docs.python.org/fr/3/library/turtle.html>

- random : un module permettant de faire de l'aléatoire. Documentation : <https://www.python-simple.com/python-modules-math/random.php>

Pour les installer, dans le terminal de Visual Studio Code, entrer la commande suivante :

```
python3 -m pip install [nom du module]
```

Rappel des étapes d'un projet

Pour chaque niveau du projet :

- 1) Réfléchir à la structure du programme : quelles sont les différentes actions qu'il doit effectuer ? Quelles sont celles qui vont être répétées ?
- 2) En déduire la liste des différentes fonctions à implémenter.
- 3) Se répartir le travail en définissant les fonctions que chacun va implémenter, et commencer à coder. Penser à utiliser des noms de variables explicites.
- 4) Faire un point régulièrement sur les avancées de chacun pour effectuer des ajustements si nécessaires.
- 5) Lorsqu'une fonction est terminée, la tester individuellement. Gérer les erreurs. Rédiger sa docstring.
- 6) Une fois chaque tâche effectuée, réunir les différentes fonctions et morceaux de code et tester le fonctionnement global du programme. Gérer les erreurs.
- 7) Analyser le produit final et envisager des améliorations.

Restitution et évaluation

Vous travaillerez en binôme.

Vous n'oublierez pas de commenter votre code et de nommer vos fichiers, variables et fonctions de manière explicite. Pensez également à inclure vos noms en commentaire en début de tout fichier Python.

Chaque binôme produira une vidéo d'une durée comprise entre 3 et 4 minutes. Chacun des membres du groupe doit prendre la parole sur une durée équitable et présenter son travail. Cette vidéo devra expliquer brièvement le fonctionnement du programme. Vous pourrez expliquer plus longuement les difficultés rencontrées et comment vous les avez gérées.

Vos fichiers .py ainsi que votre vidéo devront être partagés dans le devoir Teams créé pour l'occasion.