# FICHE : Choix Technologiques – Laurent Lauret

# Compte rendu de la veille technologique

**Les marque-pages identifiés lors de vos recherches**

*Lien vers une page publique contenant vos marque-pages collaboratifs ou lister les marque-pages directement ici. Pour chaque lien : URL, nom de la page et description sommaire.*

<https://www.python.org/>   
Page officielle de Python : installation et documentation

<https://www.w3schools.com/python/default.asp>   
W3SCHOOL : survol rapide de la syntaxe, des fonctionnalités et des possibilités d’utilisations)

<https://www.sympy.org/en/index.html>   
SYMPY 1.11 : librairie de calcul symbolique, ayant un module de géométrie 3D

<https://geometry3d.readthedocs.io/en/latest/about.html>  
Documentation de la librairie Geometry3D, pour le calcul géométrique 3D, supportant la représentation visuelle de l’espace 3D, et offrant un support pour les formes géométriques simples.

**Les preuves de concept (prototypes)**

*Pour chaque preuve de concept réalisée : identifier le but de la preuve de concept (ce qu’elle vérifie), le lien vers le sous-répertoire de votre dépôt GitHub qui contient le code de la preuve de concept ainsi que les résultats de votre expérimentation, puis, finalement, vos conclusions.*

**Prototype 1** : <https://github.com/Cegep-Rimouski/esp-h23-LLLauret/tree/main/prototypes/Prototype_1>  
Ce prototype vise à tester la possibilité de pouvoir travailler au niveau du pixel dans le canevas d’une interface graphique (*tkinter.Canevas*). **Conclusions** : le temps de réponse est de 4 à 5 secondes pour une résolution de 800x600, ce qui n’est pas ergonomique dans une interface graphique. Dans un premier temps l’application s’utilisera en ligne de commande, et lorsque le processus de calcul sera terminé, une fenêtre s’ouvrira pour avoir un aperçu du fichier BMP

**Prototype 2** : <https://github.com/Cegep-Rimouski/esp-h23-LLLauret/tree/main/prototypes/Prototype_2>  
Ce prototype vise à tester les fonctionnalités offertes par la librairie Geometry3D, à savoir les calculs d’intersections ainsi que les calculs d’angles. **Conclusions** : La librairie est correcte pour calculer l’intersection entre une ligne/sesgment et une figure géométrique, cependant l’implémentation concernant les calculs d’angle n’est pas complète, et certaines fonctions lèvent des exceptions pour du code non implémenté. La solution est de passer par des étapes intermédiaires de calculs, pour arriver à calculer l’angle entre une ligne et un plan par exemple.

Les prototypes visent également à tester les fonctionnalités de rendu proposés par la librairie, permettant d’avoir un aperçu en perspective isométrique de la scène en 3 dimensions.  
**Conclusions** : Le rendu ne fonctionnait pas, sans doute à cause des différentes versions de la librairie Matplotlib, dont elle dépend. Cependant, après avoir modifier un peu le code du *Renderer*, les fonctionnalités rendu sont de nouveaux disponibles.

# Première technologie sélectionnée (la nouvelle)

|  |  |
| --- | --- |
| **Technologie :** | PYTHON |
| **URL :** |  |

**Justifier votre choix technologique pour cette technologie**

*Expliquer à l’aide d’une argumentation rationnelle votre choix technologique. Établir votre justification à l’aide de liens avec les fonctionnalités, contraintes et risques de votre projet. Un tableau comparatif permettant de synthétiser votre réflexion pourrait être un apport judicieux à vos explications.*

J’ai choisi d’utiliser le langage python pour créer mon application car c’est un langage très répandu et que j’aimerais apprendre. Python est un langage de haut niveau utilisé dans le domaine de la recherche scientifique autant que dans le domaine de l’Éducation, c’est pourquoi je pense qu’il pourrait faciliter l’implémentation de mon application en gardant le code lisible, simple. De plus, il existe déjà de nombreuses librairies pour python, permettant de faire du calcul géométrique en 3D, ce qui me permettra de mieux aborder le traitement de données en 3D. Supportant plusieurs paradigmes, Python me permettra de mettre en avant les principes de la POO.

Par-contre, Python est un langage interprété ce qui le rend moins rapide que d’autres langages compilés. Un Ray-Tracer demande beaucoup de calcul ce qui représente un risque, cependant le but n’est pas de créer un rendu dynamique, mais plus d’obtenir une image.

# Deuxième technologie sélectionnée (la connue)

|  |  |
| --- | --- |
| **Technologie :** | Je n’ai pas de technologie connue |
| **URL :** | N/A |

**Justifier votre choix technologique pour cette technologie**

*Expliquer à l’aide d’une argumentation rationnelle votre choix technologique. Établir votre justification à l’aide de liens avec les fonctionnalités, contraintes et risques de votre projet. Un tableau comparatif permettant de synthétiser votre réflexion pourrait être un apport judicieux à vos explications.*

Je n’ai pas de technologie connue, cependant j’utilise des librairies (auxquelles je ne suis pas familier) qui interagissent avec l’interpréteur Python. En autres, il y aura une librairie pour avoir un affichage et une librairie pour simplifier les calculs géométriques.