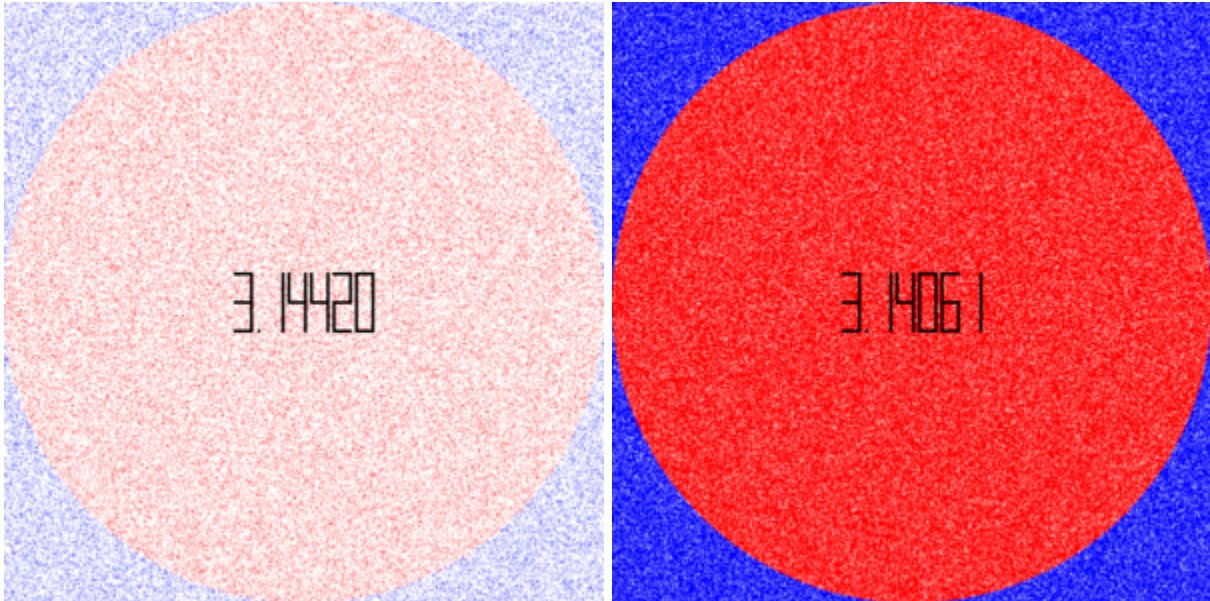


Rapport Projet BPI

TOURDES Yohann Ensimag 1A Groupe 6



Module simulator

On commence par vérifier si l'entrée de l'utilisateur est correcte. On le fait en vérifiant la longueur de `sys.argv`.

Fonction *simu*

On répartit aléatoirement des points dans un carré de $[-1; 1]$ par rapport au nombre de point : on a alors des points d'une précision à $\frac{1}{n}$ le nombre de point entré par l'utilisateur. On ajoute ces points dans une liste avec une seconde variable : un booléen qui nous indique la position du point par rapport au cercle. On divise notre calcul de point par 10 pour nous permettre d'afficher plus tard l'évolution de pi.

Le stockage de pi se fait également dans une liste également pour l'affichage plus tard.

Module `approximate_pi`

Le module permet de générer des images ppm. On ne convertit ces images en gif qu'à la toute fin après avoir exécuter dix fois la fonction `generate_ppm_file`. On utilise alors la fonction `call` de `subprocess` pour executer la fonction **`convert`**.

Fonction `transfo_dimension`

La fonction `transfo_dimension` permet de modifier la valeur des points générés par la fonction `simu` du module **`simulator`**. Ainsi on remet alors les points dans la bonne echelle de coordonnée. On stocke alors nos points dans une nouvelle liste qui contient les points à la bonne echelle ainsi que le bouléen contenant l'information de la présence ou non dans le cercle.

Fonction `generate_ppm_file`

On commence par adapter les décimales de pi pour préparer l'affichage plus tard. On crée une nouvelle image contenant l'information de pi dans son nom grâce aux **`f-string`**.

On construit alors une nouvelle liste contenant les valeurs de la couleur au point donné (1 0 0 s'il est dans le cercle, 0 0 1 sinon). On remplace ensuite les couleurs des points pour afficher les nombres en appelant la fonction `affichage_nombre`.

On écrit ensuite cela dans notre fichier. On convertit ensuite notre ppm écrit en P3 (non compressé) en P6 (compressé) grâce à la commande **`convert`** utiliser avec la fonction `call` du module **`subprocess`**.

Fonction `affichage_nombre`

La fonction affiche le nombre pi dans l'image ppm. On remplace la couleur de ces points par la valeur dans le tableau crée. On utilise la fonction `affichage_nombre` pour simplifier l'écriture de la fonction.

Fonction `affichage_chiffre`

La fonction affiche les chiffres de manière individuelle. Le probleme de cette fonctionne est le nombre de condition pour savoir quel point affiché.