

Ma machine :

Cette machine comporte 2 CPU, 8 GO de RAM, 16 GO de stockage en provisionnement dynamique.

> CPU	1 CPU
> Mémoire	2 Go, 0 Go mémoire active
▼ Disque dur 1	
Capacité	16 Go
Type	Provisionnement dynamique
Emplacement	R202 (2,47 To libre)
> Adaptateur réseau 1	M2102 (Admin syst) (connecté)
> Lecteur CD/DVD 1	Connecté
> Lecteur CD/DVD 2	Connecté
Lecteur CD/DVD 3	Déconnecté
> Carte vidéo	8 Mo
Périphérique VMCI	
Périphérique sur le bus PCI de la machine virtuelle fournissant la prise en charge pour l'interface de communication de la machine virtuelle	
> Autre	Matériel supplémentaire
Compatibilité	ESXi 6.7 et versions ultérieures (VM version 14)

J'ai choisie de mettre 1 cœur de plus et 6go de RAM car j'ai mis en place l'environnement visual studio code sur la machine virtuel, et j'avais besoin de plus de puissance de calcul et de mémoire vive. J'ai déjà eu un crash avec la machine, et sous la permission de M. Matichard, j'ai augmenté le nombre de vCoeur et de RAM.

Mes login / mot de passe sont :

heron / Fr@-1s£ et root / Fr@mb01Se

7. Modification du site web

7.1 analyses

Lorsque l'on trouve le fichier python, on observe qu'il y a une fonction nommé motif().

```
heron@debianHeronSae24: /var/www/html/amath
GNU nano 3.4 koch.py

Exemple : python3 koch.py 3
Crée un fichier png et retourne le nom de ce fichier

Principe : http://math.univ-lyon1.fr/irem/Formation_ISM/formation_recursive/tortue/vonkoch.html

@author: Frédéric BONNARDOT, All rights reserved
This code is given as is without warranty of any kind.
In no event shall the authors or copyright holder be liable for any claim damages or other liability.

"""
# Pour installer le paquet matplotlib :
# apt install python3-pip
# apt install python3-matplotlib

import sys # Pour récupérer les paramètres passés en ligne de commande
import matplotlib.pyplot as plt
import math

def motif(fig,dim,x1,y1,x2,y2):
    if dim==0:
        # Dimension = 0 -> trace une droite
        plt.plot([x1,x2],[y1,y2])
    else:
        # Sinon partage en 4 sections
        deltax=(x2-x1)/3
        deltay=(y2-y1)/3
        xa=x1+deltax
        ya=y1+deltay
        xb=xa+deltax*0.5-deltay*math.sqrt(3)/2
        yb=ya+deltax*math.sqrt(3)/2+deltay*0.5
        xc=x2-deltax
        yc=y2-deltay
        motif(fig,dim-1,x1,y1,xa,ya)
        motif(fig,dim-1,xa,ya,xb,yb)
        motif(fig,dim-1,xb,yb,xc,yc)
        motif(fig,dim-1,xc,yc,x2,y2)

dimension=int(sys.argv[1]) # Premier paramètre : dimension que l'on convertit en entier

# Dessine l
fig=plt.figure()
motif(fig,dimension,0,0,1,0)

# Enregistre la figure
```

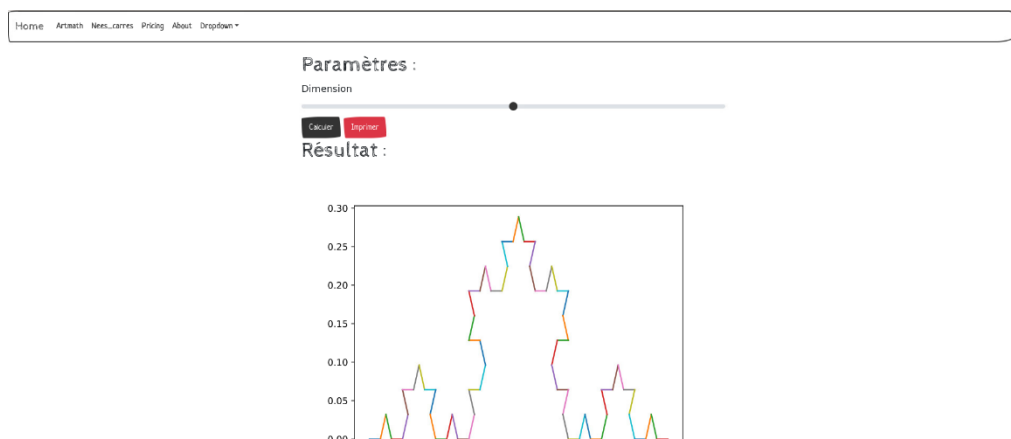
Cette fonction prend en paramètre l'objet figure, la dimension de celle-ci, ainsi que 4 coordonnées.

Cette fonction est récursive, c'est-à-dire qu'elle s'appelle elle-même. La condition d'arrêt est que la dimension diminue de 1 à chaque tour, et lorsqu'elle atteint 0, elle trace une droite, ce qui forme l'œuvre de Koch.

Le programme est appelé par le fichier ArtMathématique.php. Il appelle le fichier koch.py avec python 3 en donnant les dimensions données par l'utilisateur.

```
// Récupère les paramètres issus du formulaire (on indique le champ name)
$dimension = $request -> request -> get("dimension") ;
```

```
// Oui : Appelle le script Python koch.py qui se trouve dans le répertoire /public
$process = new Process(['python3', 'koch.py', $dimension]);
$process -> run();
// Récupère la valeur de retour renvoyé par le script python
$fichier=$process->getOutput();
```



7.2 Adaptation

Grâce à l'exemple présent dans le fichier nees_carre.py, j'ai pu comprendre le fonctionnement du programme.

Exemple : python3 nees_carre.py 0.75 0.5 8 10
Crée un fichier png et retourne le nom de ce fichier

Le programme nees_carre.py attend 4 paramètres de dimensionnement.

```
# Récupère les paramètres
# Premier paramètre : amplitude du hasard entre 0 et 1
# replace permet de transformer 5,4 en 5.4 car python attend un . pour les flottants
hasard=float(sys.argv[1].replace(',', '.'))
# Amplitude des rotations entre 0 et 1
hasardangle=float(sys.argv[2].replace(',', '.'))
# Nombre de lignes et de colonnes nombres entiers
nbcolonnes=int(sys.argv[3])
nblignes=int(sys.argv[4])
```

Le premier permet de régler l'amplitude du hasard en 0 et 1 de la figure.

Le second permet de régler l'amplitude des rotations entre 0 et 1

Le troisième permet de régler le nombre de colonnes avec des nombres entier

Le dernier permet de régler le nombre de lignes avec des nombres entier.

Le programme change le point en virgule car les nombres décimaux en python sont définie avec des virgules.

Le fichier renvoie une image nommé « reponse.png ».

```
# Enregistre la figure
fichier='reponse.png'
pygame.image.save(screen,fichier)
```

J'ai adapté le formulaire pour mettre en place les dimensions de nees_carre.py

[Home](#) [Armath](#) [Nees_carres](#) [Pricing](#) [About](#) [Dropdown](#)

Programme Python nees_carre.py

Paramètres :

Amplitude du hasard entre 0 et 1

Amplitude des rotations entre 0 et 1

Nombre de lignes entre 0 et 50

Nombre de colonnes entre 0 et 50

Calculer

Imprimer

Résultat :

Celui-ci prend les 4 paramètres en compte et permet de les ajuster avec une barre.

J'ai créé une page html propre pour l'affichage de cette figure. Ici, fig2

```
/**
 * @Route("/fig2", name="app_fig2")
 */
public function fig2(): Response
{
    return $this->render('artmath/fig2.html.twig', [
        'fichier_nees' => '',
    ]);
}
```

Cette page permet d'afficher et d'utiliser les ranges

```
<h1>Programme Python nees_carre.py</h1>
<h1>Paramètres :</h1>
<!-- C'est grâce à l'attribut name que l'on peut récupérer la valeur associée dans symfony -->
<form method="post" action="/calculer_nees">
    <div>
        <label for="Hasard" class="form-label">Amplitude du hasard entre 0 et 1</label>
        <input type="range" class="form-range" min="0" max="1" step="0.01" id="dimension_ama" name="dimension_ama">
        <label for="Harsandangle" class="form-label">Amplitude des rotations entre 0 et 1</label>
        <input type="range" class="form-range" min="0" max="1" step="0.01" id="dimension_amr" name="dimension_amr">
        <label for="nombreligne" class="form-label">Nombre de lignes entre 0 et 50</label>
        <input type="range" class="form-range" min="0" max="50" step="1" id="dimension_nbc" name="dimension_nbc">
        <label for="nombreligne" class="form-label">Nombre de colonnes entre 0 et 50</label>
        <input type="range" class="form-range" min="0" max="50" step="1" id="dimension_nbl" name="dimension_nbl">
    </div>
    <div>
        <button type="submit" class="btn btn-primary" name="calculer_nees" value="1">Calculer</button>
        <button type="submit" class="btn btn-danger" name="imprimer_nees" value="1">Imprimer</button>
    </div>
</form>

<h1> Résultat : </h1>

```

Il y a donc 4 ranges car 4 dimensions nécessaires :

dimension_ama : Permet de régler l'amplitude du hasard en 0 et 1 de la figure avec un pas de 0.01.

dimension_amr : Permet de régler l'amplitude des rotations entre 0 et 1 de la figure avec un pas de 0.01

dimension_nbc : Permet de régler le nombre de colonnes avec des nombres entier entre 0 et 50 avec un pas de 1.

Dimension_nbl : Permet de régler le nombre de lignes avec des nombres entier entre 0 et 50 avec un pas de 1.

Enfin, il y a 2 boutons qui renvoient vers les pages « calculer_nees » et « imprimer_nees ». Le résultat sera affiché sous « Résultat : » grâce à la variable « fichier_nees »

Dans le controller, on retrouve la partie qui va exécuter le programme.

Lorsque l'on va appuyer sur le bouton « calculer_nees », il va récupérer les valeurs des ranges selon leurs ID. Il va ensuite transférer les variables au programme python puis l'exécuter. Enfin, la variable « fichier_nees » prend la dernière image générée par le programme python sous le nom de « reponse.png ». Il renvoie ensuite la variable « fichier_nees » à la page html fig2.html.twig

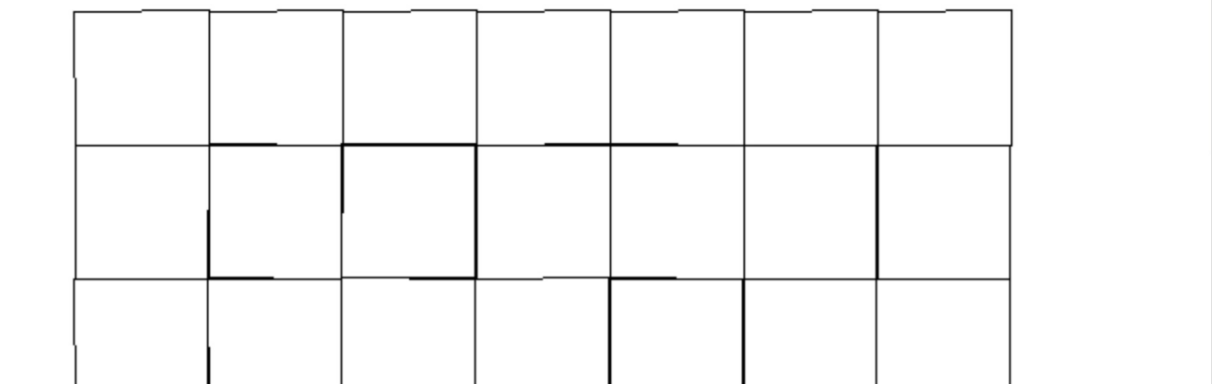
```
/**
 * @Route("/calculer_nees", name="calculer_nees")
 */
public function calculer_nees(Request $request): Response
{
    // Récupère les paramètres issus du formulaire (on indique le champ name)
    $dimension_ama = $request -> request -> get("dimension_ama") ;
    $dimension_amr = $request -> request -> get("dimension_amr") ;
    $dimension_nbc = $request -> request -> get("dimension_nbc") ;
    $dimension_nbl = $request -> request -> get("dimension_nbl") ;
    // Pour les boutons : si appui contenu champ value sinon NULL
    $calculer_nees = $request -> request -> get("calculer_nees");
    $imprimer_nees = $request -> request -> get("imprimer_nees");

    // Oui : Appelle le script Python koch.py qui se trouve dans le répertoire /public
    $process_nees = new Process(['python3', "nees_carre.py", $dimension_ama, $dimension_amr, $dimension_nbc, $dimension_nbl]);
    $process_nees -> run();
    // Récupère la valeur de retour renvoyé par le script python
    $fichier_nees="reponse.png";

    // Retourne un message si l'exécution c'est mal passée
    if (!$process_nees->isSuccessful())
        return new Response ("Erreur lors de l'exécution du script Python :<br>".$process_nees->getErrorMessage());

    // A t'on appuyé sur calculer ?
    if ($calculer_nees!=NULL)
        return $this->render('artmath/fig2.html.twig', [
            'fichier_nees' => $fichier_nees,
        ]);
    else {
        // On a appuyé sur imprimer
        return $this->render('artmath/imprimer_nees.html.twig', [
            'fichier_nees' => $fichier_nees,
        ]);
    }
}
```

Résultat :



7.3 Améliorations :

voici mes pistes d'amélioration :

1) Github

J'ai mis en place une liaison GitHub sur la machine virtuelle en SSH. Cette liaison me permet de modifier le site web depuis mon poste et de push les modifications directement sur le site.

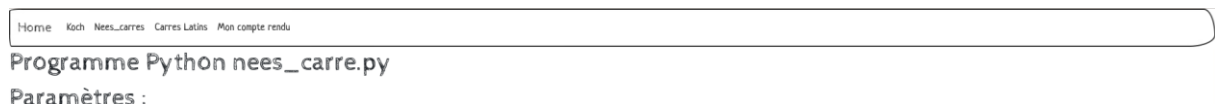
<https://github.com/leonardoheron/SAE24>

```
root@debianHeronSae24:/var/www/html/amath# git pull
astuce: Tirer sans spécifier comment réconcilier les branches divergentes
astuce: est découragé. Vous pouvez éliminer ce message en lançant une des
astuce: commandes suivantes avant votre prochain tirage :
astuce:
astuce:  git config pull.rebase false  # fusion (stratégie par défaut)
astuce:  git config pull.rebase true   # rebasage
astuce:  git config pull.ff only       # avance rapide seulement
astuce:
astuce: Vous pouvez remplacer "git config" par "git config --global" pour que
astuce: ce soit l'option par défaut pour tous les dépôts. Vous pouvez aussi
astuce: passer --rebase, --no-rebase ou --ff-only sur la ligne de commande pour
astuce: remplacer à l'invocation la valeur par défaut configurée.
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 2), reused 4 (delta 2), pack-reused 0
Dépaquetage des objets: 100% (4/4), 364 octets | 182.00 Kio/s, fait.
Depuis github.com:leonardoheron/SAE24
   0ab8262..5745169  main    -> origin/main
Mise à jour 0ab8262..5745169
Fast-forward
 templates/base.html.twig | 2 +
 1 file changed, 1 insertion(+), 1 deletion(-)
root@debianHeronSae24:/var/www/html/amath#
```

Pour le mettre en place, j'ai dû créer des clés SSH publique et privée, puis les mettre sur mon GitHub.

2) NavBar

J'ai mis en place une barre de navigation pour permettre le choix entre les figures depuis n'importe quelle page.



Cette navbar à plusieurs fonctions, elle renvoie à la page d'accueil par le bouton « Home », renvoie sur la page pour utiliser l'œuvre de « koch », renvoie sur la page pour utiliser l'œuvre des « carres_latins » ou encore télécharger mon compte rendu.

Le programme ci-dessous est la navbar pour la page « fig2.html.twig », je n'ai pas mis la navbar dans « base.html.twig » car j'ai mis en place des ancres lorsque la personne se trouve déjà sur la page.

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="/">Home</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarColor03" aria-controls="nav
      <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarColor03">
      <ul class="navbar-nav me-auto">
        <li class="nav-item">
          <a class="nav-link active" href="/artmath">Koch
            <span class="visually-hidden">(current)</span>
          </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Nees_carres</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Carres Latins</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Mon compte rendu</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

3) Téléchargement du compte rendu

J'ai mis en place la possibilité de télécharger mon compte rendu depuis le site web. Il suffit donc de cliquer sur « Mon compte rendu » sur la navbar et celui-ci sera proposé pour être téléchargé.

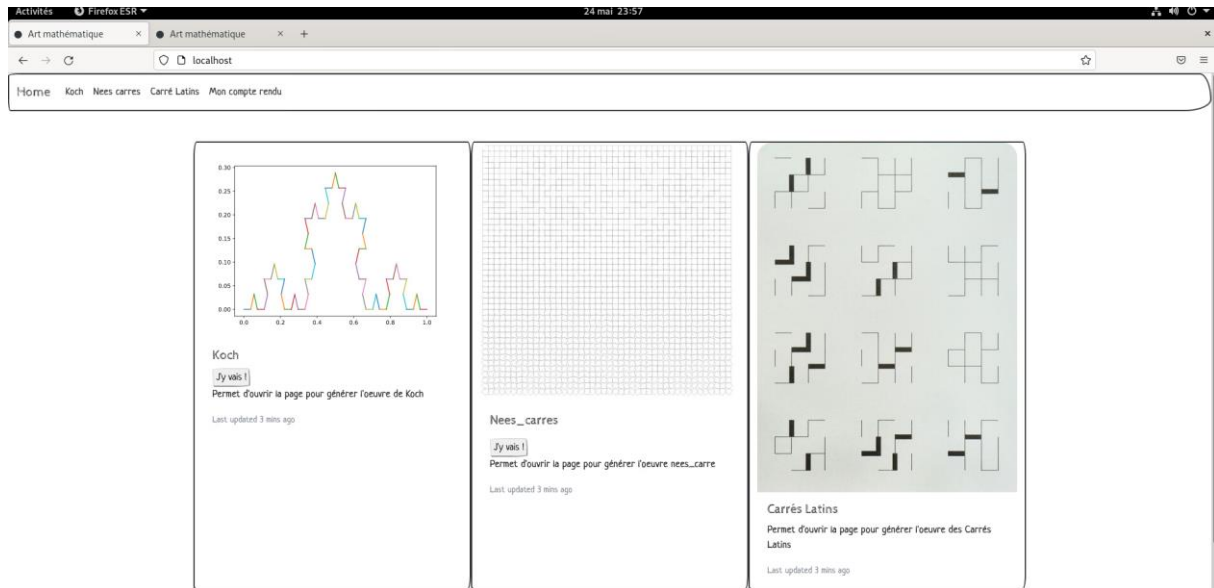
4) Pied de page et copyrights

J'ai mis un pied de page contenant mes contacts linkedin et github. J'ai aussi mis les copyrights selon les termes de la licence creative commons attribution



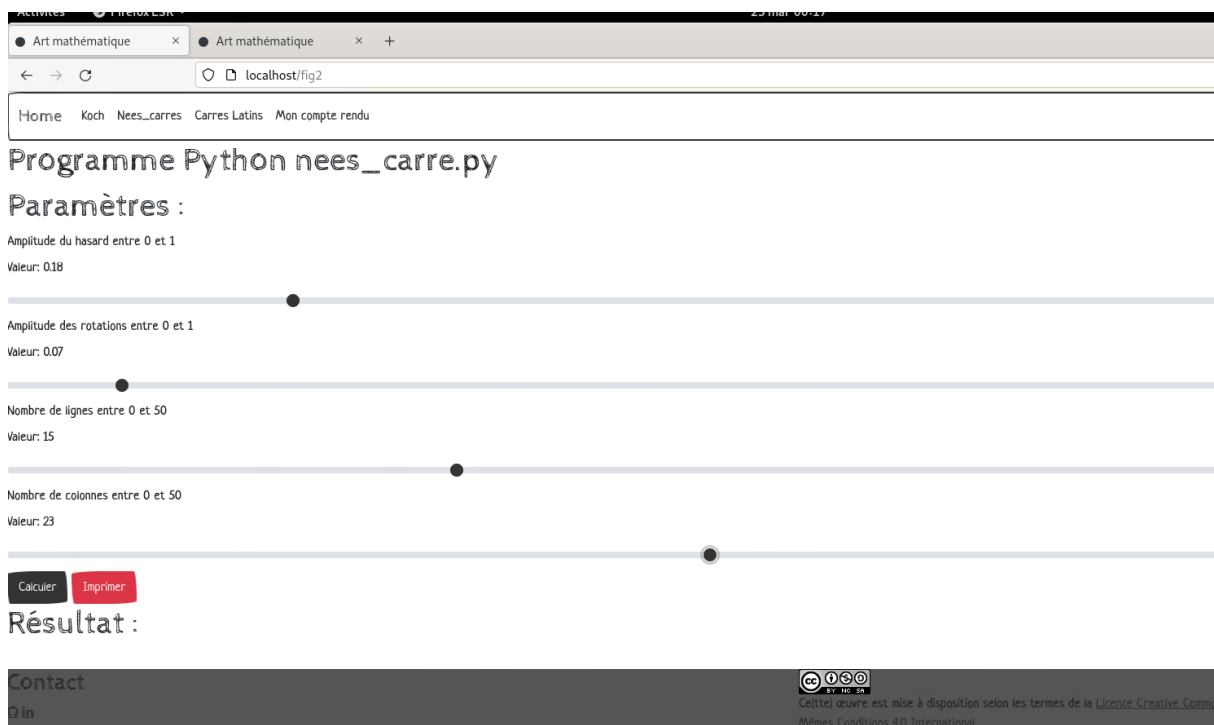
5) Page d'accueil

J'ai fait une page d'accueil permettant de choisir entre les œuvres. Il y a une image et une description pour chacune des œuvres et un bouton qui renvoie vers l'œuvre.



6) Affichage des nombres en direct

Grâce à une petite fonction en JavaScript, je récupère la valeur actuelle de la range, puis je l'affiche en direct à côté de « Valeur : »




```
</label><input type="range" class="form-range" min="0" max="1" step="0.01" id="dimension_ama" name="dimension_ama"></input></div><div><label for="Harsardangle" class="form-label">Amplitude des rotations entre 0 et 1</label><p>Valeur: <span id="val_dim_amr"></span></p><input type="range" class="form-range" min="0" max="1" step="0.01" id="dimension_amr" name="dimension_amr"></input></div><div><label for="nombreligne" class="form-label">Nombre de lignes entre 0 et 50</label><p>Valeur: <span id="val_dim_nbc"></span></p><input type="range" class="form-range" min="1" max="50" step="1" id="dimension_nbc" name="dimension_nbc"></input></div><div><label for="nombreligne" class="form-label">Nombre de colonnes entre 0 et 50</label><p>Valeur: <span id="val_dim_nbl"></span></p><input type="range" class="form-range" min="1" max="50" step="1" id="dimension_nbl" name="dimension_nbl"></input></div></div>
```

```
<script>
var slider = document.getElementById("dimension_ama");
var output = document.getElementById("val_dim_ama");
output.innerHTML = slider.value;

var slider2 = document.getElementById("dimension_amr");
var output2 = document.getElementById("val_dim_amr");
output2.innerHTML = slider2.value;

var slider3 = document.getElementById("dimension_nbc");
var output3 = document.getElementById("val_dim_nbc");
output3.innerHTML = slider3.value;

var slider4 = document.getElementById("dimension_nbl");
var output4 = document.getElementById("val_dim_nbl");
output4.innerHTML = slider4.value;

slider.oninput = function() {
  output.innerHTML = this.value;
}

slider2.oninput = function() {
  output2.innerHTML = this.value;
}

slider3.oninput = function() {
  output3.innerHTML = this.value;
}

slider4.oninput = function() {
  output4.innerHTML = this.value;
}
</script>
```