**Année académique 2020-2021**

**Promotion : B1 Switch IT**



**Campus Contest**

**B1 Switch IT**

**17 et 18 décembre 2020**

Table des matières

[1. Introduction 3](#_Toc59133353)

[1.1 Objectif de l’expérience professionnelle pour l’apprenant 3](#_Toc59133354)

[1.2 Calendrier Campus Contest 3](#_Toc59133355)

[1.3 Modalités 3](#_Toc59133356)

[1.4 Évaluations 3](#_Toc59133357)

[2. Contexte : Christmas Tree 4](#_Toc59133358)

[2.1 Rendus attendus : Team Workflow 4](#_Toc59133359)

[2.2 Niveau 1 – Le sapin à 3 étages 4](#_Toc59133360)

[Algorithm of Level 1: 5](#_Toc59133361)

[2.3 Niveau 2 – Tronc 6](#_Toc59133362)

[Algorithm of Level 2: 6](#_Toc59133363)

[2.4 Niveau 3 – Guirlandes 7](#_Toc59133364)

[Algorithm of Level 3 : 8](#_Toc59133365)

[2.5 Niveau 4 – Boules de Noël 9](#_Toc59133366)

[Algorithm of Level 4 : 9](#_Toc59133367)

[2.6 Niveau 5 – L’étoile finale de la mort ! 11](#_Toc59133368)

[Algorithm of Level 5 : 12](#_Toc59133369)

[2.7 Bonus 1 – Un sapin plus grand ! 14](#_Toc59133370)

[3. Rendu Computer Fundamentals 15](#_Toc59133371)

[3.1 Rendus attendus (Document Word) : 15](#_Toc59133372)

[3.2 Bonus (au choix, fournir POC et doc) : 15](#_Toc59133373)

[4. Grillé de correction 16](#_Toc59133374)

[5. Annexe : Liste des courriels 17](#_Toc59133375)

# Introduction

# Objectif de l’expérience professionnelle pour l’apprenant

Mettre en œuvre les capacités acquises en cours de formation en réinvestissant ce qui a été appris en situation professionnelle en lien avec un besoin simple : concevoir le sapin de noël de Campus Academy.

# Calendrier Campus Contest

17 et 18 décembre 2020.

# Modalités

Par groupe de deux. Livrable en collaboration en utilisant GIT et en déposant votre rendu sur GitHub ou GitLab par exemple. Les groupes ont été générés aléatoirement.

|  |  |  |  |
| --- | --- | --- | --- |
| Nom | Prénom | Promotion | Random |
| CIVILE | Nicola | ANG Switch IT – Bachelor 1 | 0,036 |
| SÉGALEN | Thomas | ANG Switch IT – Bachelor 1 | 0,107 |

# Évaluations

Pour chaque période dédiée, vous serez évalués sur vos avancements et vos livrables. Vous trouverez la fiche d’évaluation finale à la fin de ce document.

# Contexte : Christmas Tree

Noël approche et des animaux se sont introduits dans tous les locaux de Campus Academy pour dérober les sapins de noël. Les animaux eux aussi fêtent noël autour d’un arbre et non pas aimés que les êtres humains découpent leurs arbres. Compatissant, les élèves ont décidé de faire sans sapin de noël provenant de la forêt. Super intelligent, Ils vont utiliser une imprimante 3D pour obtenir leur sapin. Pour cela, l’équipe Switch IT est réquisitionnée pour développer un programme qui va dessiner à l’écran un sapin de noël.

# Critères de réalisation

* Votre programme Python doit pouvoir être exécuter en ligne de commande : Ex : python3 mon\_programme.py Idem avec des arguments Ex : python3 mon\_programme.py argument\_1
* **Interdiction d’ajouter des sapins en « dur » dans votre code**
* L’utilisation du style de code PEP8 et de flake8 est fortement recommandé
* Une forte appréciation sera ajoutée pour un code en anglais (pas de nom de variable avec des accents etc…)
* Pour chaque programme Python3, un rendu d’un algorithme est attendu. Utilisez le pseudo-code appris et/ou tout autre forme permettant au correcteur de comprendre votre démarche.

# Rendus attendus : Team Workflow

Avec votre binôme, vous devez travailler en collaboration. Pour cela, GIT est obligatoire. Veillez également à déposer votre code sur GitHub Votre projet doit comporter : - Des commits réguliers - Un fichier README.md propre et formaté markdown (français ou anglais).

# Niveau 1 – Le sapin à 3 étages

À partir d’une solution algorithmique, écrivez un programme qui permet de dessiner à l’écran un sapin de 3 étages.

Le sapin doit ressembler à la figure suivante :

Une image contenant texte

Description générée automatiquement

Analysez bien la forme du sapin, le nombre de « \* »

### Algorithm of Level 1:

Start

n = 4

height = n

stars\_nb = 1

spaces\_nb = height – 1

For i 0 to height

Print (“ ” + spaces\_nb \* “ “ + stars\_nb \* “\*”)

stars\_nb += 2

spaces\_nb -= 1

stars\_nb = 3

spaces\_nb = height – 1

For i 0 to height

Print (“ ” + spaces\_nb \* “ “ + stars\_nb \* “\*”)

stars\_nb += 4

spaces\_nb -= 1

stars\_nb = 5

spaces\_nb = height – 1

For i 0 to height

Print (spaces\_nb \* “ “ + stars\_nb \* “\*”)

stars\_nb += 6

spaces\_nb -= 1

End

# Niveau 2 – Tronc

Après l’avoir imprimé en 3D, les élèves se sont aperçus qu’il manquait un tronc pour le faire tenir !

Rajoutez au programme la partie pour dessiner un tronc comme la figure ci-dessous :

Une image contenant texte

Description générée automatiquement

### Algorithm of Level 2:

Start

n = 4

height = n

# Method 1

stars\_nb = 5

spaces\_nb = height – 1

For i 0 to height – 1

For j 0 to 2

Print (spaces\_nb \* “ “ + stars\_nb \* “\*”)

# Method 2

For i 0 to height – 1

Print (“ ” + “\*\*\*\*\*”)

End

# Niveau 3 – Guirlandes

Super content, vous parvenez à délivrer les plans des sapins pour l’impression 3D. Et en plus, les sapins tiennent debout !

Néanmoins, un des enseignant fait la remarque suivante : « Il serait plus jolie avec des guirlandes ! »

Vous devez à présent ajouter des guirlandes en bas du sapin :

Une image contenant texte

Description générée automatiquement

### Algorithm of Level 3 :

Start

n = 4

height = n

stars\_nb = 5

spaces\_nb = height - 1

For i 0 to height – 1

If i == 0

For j 1 to 14

If j < 5

Print (” |”, end = ””)

Elif j >= 5 and j <= 9

If j == 5

Print (” \*”, end = ””)

Else

Print (”\*”, end = ””)

Else

Print (” |”, end = “”)

Elif i == 1

Print (“”)

For j 1 to 14

If j < 5

Print (“ 0”, end = “”)

Elif j >= 5 and j <= 9

If j == 5

Print (“ \*”, end = “”)

Else

Print (“\*”, end = “”)

Else

Print (“ 0”, end = “”)

Else

Print (“”)

For i 0 to height – 3

Print (spaces\_nb \* “ “ + stars\_nb \* “\*”)

End

# Niveau 4 – Boules de Noël

Il ne faut pas faire le travail à moitié, maintenant que vous êtes parti pour la décoration, il va falloir décorer le haut !

Ajouter des boules aux extrémités des niveaux (sauf le dernier qui lui a les guirlandes) :

Une image contenant texte

Description générée automatiquement

### Algorithm of Level 4 :

Start

n = 4

height = n

# Method – Stage 2 / Christmas balls

stars\_nb = 3

spaces\_nb = height – 1

For i 0 to height

If stars\_nb == 3

Print (“ ” + spaces\_nb + “ “ + “0 “ + stars\_nb \* “\*” + “ 0”)

stars\_nb += 4

spaces\_nb -= 1

Else

Print (“ ” + spaces\_nb \* “ ” + stars\_nb \* “\*”)

stars\_nb += 4

spaces\_nb -= 1

# Method – Stage 3 / Christmas balls

stars\_nb = 5

spaces\_nb = height – 1

For i 0 to height

If stars\_nb == 5

Print (spaces\_nb \* “ ” + “ 0 ” + stars\_nb \* “\*” + “ 0 “)

stars\_nb += 6

spaces\_nb -= 1

Else

Print (spaces\_nb \* “ ” + stars\_nb \* “\*”)

stars\_nb += 6

spaces\_nb -= 1

End

# Niveau 5 – L’étoile finale de la mort !

Cette fois-ci, sans demande externe, les élèves ont décidé d’ajouter l’étoile de la mort. En réalité, c’est juste une étoile pour le sapin mais les élèves font preuves d’imagination et ont donné le nom d’étoile de la mort à leur étoile.

Rajoute donc cette étoile comme la figure ci-dessous :

Une image contenant texte

Description générée automatiquement

### Algorithm of Level 5 :

Start

n = 4

height = n

stars\_nb = 1

spaces\_nb = height – 1

For i 0 to 7

If i == 0

Print (””)

Print (“ ”, end = “”)

For j 0 to 11

If j == 0 or j == 10

Print (” \* ”, end = “”)

Elif j == 5

Print (”\*”, end = ””)

Else

Print (“ ”, end = “”)

If i == 1

Print (“”)

Print (“ ”, end = “”)

For j 0 to 11

Print (“ ”, end = “”)

If j == 5

Print (“\*”, end = “”)

Else

Print (“ ”, end = “”)

If i == 2

Print (“”)

For j 0 to 11

Print (“ ”, end = “”)

If j == 5

Print (“\*”, end = “”)

Else

Print (“ ”, end = “”)

If i == 3

Print (“”)

Print (“ ”, end = “”)

For j 0 to 11

If j == 0 or j == 2 or j == 4 or j == 6 or j == 8 or j == 10

Print (“\*”, end = “”)

Else

Print (“ ”, end = “”)

If i == 4

Print (“”)

For j 0 to 11

Print (“ ”, end = “”)

If j == 5

Print (“\*”, end = “”)

Else

Print (“ ”, end = “”)

If i == 5

Print (“”)

Print (“ ”, end = “”)

For j 0 to 11

If j == 2 or j == 8

Print (“ \* “, end = “”)

Elif j == 5

Print (“|”, end = “”)

Else

Print (“ ”, end = “”)

If i == 6

Print (“”)

Print (“ ”, end = “”)

For j 0 to 11

If j == 0 or j == 10

Print (“ \* “, end = “”)

Elif j == 5

Print (“|”, end = “”)

Else

Print (“ ”, end = “”)

Print (“”)

End

# Bonus 1 – Un sapin plus grand !

Les enseignants vous trouvant super motivés décident de vous attribuer un bonus si vous parvenez à modifier votre algorithme et donc votre programme pour concevoir des sapins de taille différentes !

Votre programme pourrait accepter un paramètre qui à partir de celui-ci va générer un sapin avec un nombre d’étage correspondant.

Exemple avec la figure suivante :

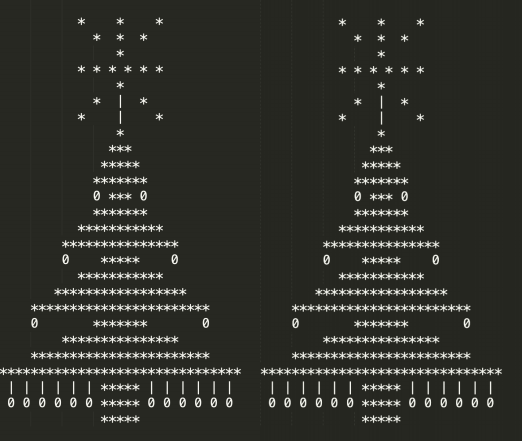
Une image contenant texte

Description générée automatiquement

# Bonus 2 – Beaucoup de sapins !

Pour faire des économies, vous décidez d’imprimer plusieurs sapins d’un coup. Pour cela, votre programme doit pouvoir dessiner plusieurs sapins sur une même ligne !

Modifiez votre programme en conséquence Ex avec la figure suivante :



# Rendu Computer Fundamentals

En plus du Git en binôme, vous décidez pour des raisons de sécurité et de confidentialité de monter votre propre GIT privé. En effet, des rennes accompagnés d’individus habillés de rouge, tentent en cette période de dérober le code source des projets Python type Arbre de Noël. Attention, certaines manipulations demandées ne sont peut-être pas faisables. À vous de justifier.

# Rendus attendus (Document Word) :

* Documentation de l’installation d’un Git sur une VM locale (Windows et/ou Linux) protégée par un routeur PFSENSE – 10 points

Création d’une machine virtuelle : CA-WINDOWS

Une image contenant texte

Description générée automatiquement

Machine virtuelle de Génération 2

Une image contenant texte

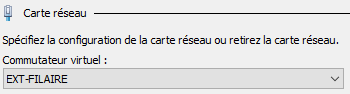
Description générée automatiquement

Mémoire affectée à la machine virtuelle : 4096 Mo

Une image contenant texte

Description générée automatiquement

Mise en réseau via Ethernet : EXT-FILAIRE



Stockage virtuel dédié : 41 Go

Une image contenant texte

Description générée automatiquement

ISO : Windows Server 2016

Une image contenant texte

Description générée automatiquement

Création du routeur virtuel : CA-PFSENSE

Une image contenant texte

Description générée automatiquement

Routeur virtuel de Génération 1

Une image contenant texte

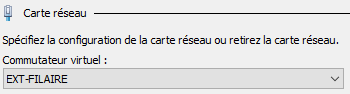
Description générée automatiquement

Mémoire affectée au routeur virtuel : 4096 Mo

Une image contenant texte

Description générée automatiquement

Mise en réseau via Ethernet : EXT-FILAIRE



Stockage virtuel dédié : 40 Go

Une image contenant texte

Description générée automatiquement

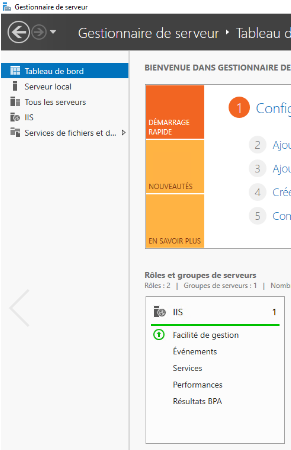
ISO : pfSense 2.4.5

Une image contenant texte

Description générée automatiquement

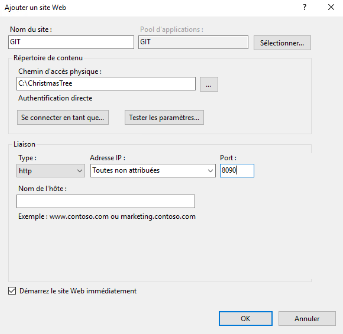
Installation de GIT sur la machine virtuelle :

Création d’un service IIS :



Nom du site : GIT  
Chemin d’accès physique : Répertoire « ChristmasTree » créé dans le disque C:\

Port : 8090



Répertoire GIT : ChristmasTree

Une image contenant texte

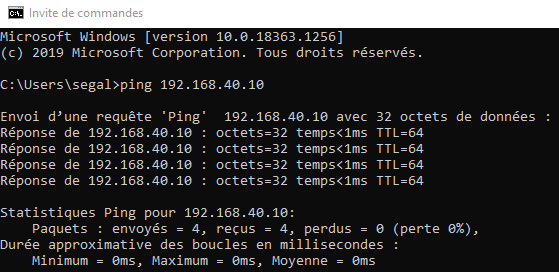
Description générée automatiquement

* Proof of concept de fonctionnement de votre infrastructure – 7 points

Ping sur l’IP 8.8.8.8 afin de tester l’accessibilité de notre routeur



Ping sur l’IP de notre machine virtuelle afin de tester l’accessibilité de celle-ci



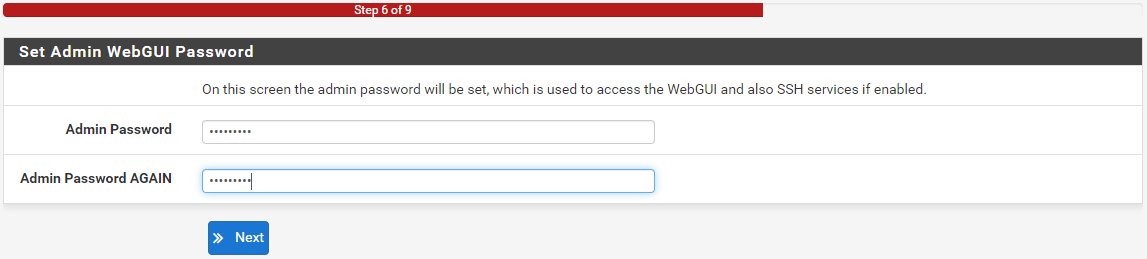
Connexion au panel pfSense :



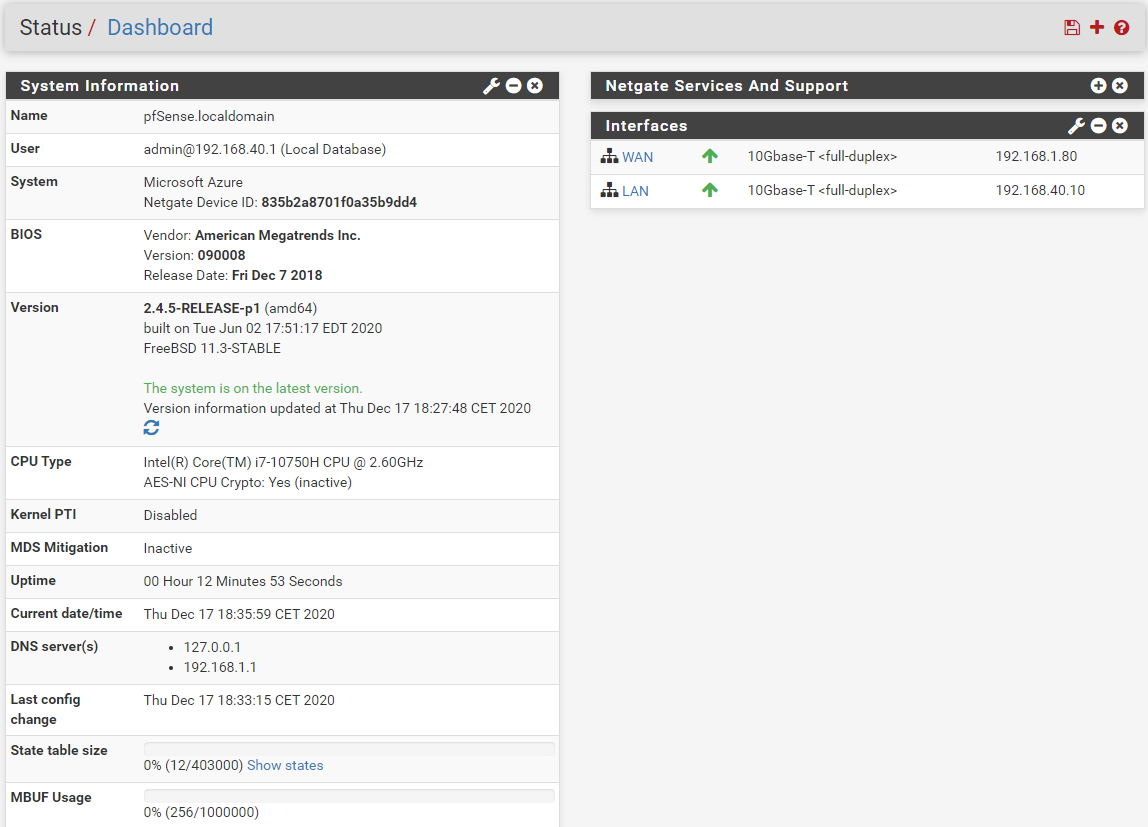
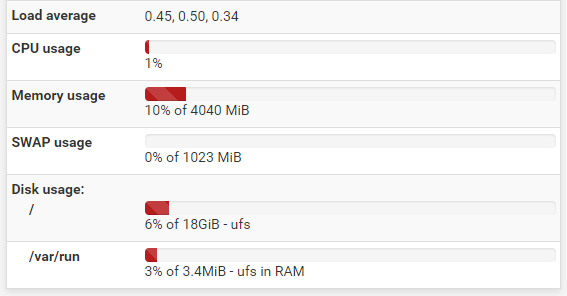
Avec comme identifiant : admin et mot de passe : pfsense



Changement de notre mot de passe



Dashboard pfSense :



* Proof of concept de mise en ligne de votre code – 3 points

Une image contenant texte

Description générée automatiquement

# Bonus (au choix, fournir POC et doc) :

* Le père Noël dans les nuages – Mettre en place un GIT dans une VM en Cloud Azure à partir de portal.azure.com – 5 points
* Noël confiné – À partir du domicile d’un membre du binôme, parvenir à accéder au GIT de l’autre membre à travers sa box – 5 points

# Grille de correction

Promotions : **Bachelor 1**

Filière : **Switch IT**

|  |  |  |
| --- | --- | --- |
| **Livrables fournis :** | **Barème** | **Points** |
| Programme Python3 (3pts) + algo du niveau 1 (1pts) : Le sapin à 3 étages | 4 |  |
| Programme Python3 (4pts) + algo du niveau 2 (1pts) : Tronc | 5 |  |
| Programme Python3 (4pts) + algo du niveau 3 (1pts) : Guirlandes | 5 |  |
| Programme Python3 (4pts) + algo du niveau 4 (1pts) : Boules de noël | 5 |  |
| Programme Python3 (4pts) + algo du niveau 5 (1pts) : L’étoile de la mort | 5 |  |
| Documentation de l’installation d’un Git sur un VM locale (Windows et/ou Linux) protégée par un routeur PFSENSE | 10 |  |
| Proof of concept de fonctionnement de votre infrastructure | 7 |  |
| Proof of concept de mise en ligne de votre code | 3 |  |
| **TOTAL** | **44** |  |

|  |  |  |
| --- | --- | --- |
| **Compétences techniques :** | **Barème** | **Points** |
| Le code est propre | 2 |  |
| Le programme Python3 est fonctionnel | 1 |  |
| Chaque niveau comporte programme python3 + algorithme | 3 |  |
| Le code respecte les bonnes pratiques de développement | 2 |  |
| Des commits réguliers avec des messages courts et appropriés | 3 |  |
| Un fichier README.md propre et formaté markdown pour chaque niveau | 2 |  |
| **TOTAL** | **13** |  |

|  |  |  |
| --- | --- | --- |
| **Compétences transverses :** | **Barème** | **Points** |
| Implication dans le rendu et la collaboration pour mener à bien le challenge | 3 |  |

**Total : / 60**

|  |  |  |
| --- | --- | --- |
| **Bonus :** | **Barème** | **Points** |
| Programme Python3 : Un sapin plus grand ! | 4 |  |
| Programme Python3 : Beaucoup de sapins ! | 4 |  |
| Mise en place du git dans une VM en Cloud Azure | 5 |  |
| Accès d’un git du membre de son équipe à travers sa box | 5 |  |

**Total avec bonus : / 60**

**Note :** Cette grille d’évaluation est liée au travail de groupe ou binôme, mais elle n’oblige pas l’évaluateur à renseigner la même note à tous les membres du groupe. Par conséquent, il est possible de trouver des différences de note d’un apprenant à l’autre.

\* Les documents doivent être accessibles par le correcteur.

# Annexe : Liste des courriels

La liste ci-dessous des courriels sont à **ajouter à vos dossiers GIT** sur GitHub.

Cette liste correspond à vos potentiels correcteurs et pour cela, un accès est obligatoire !

k.niel.pro@gmail.com