

Rapport de projet de réseaux

Le pierre-papier-ciseaux version Raspberry



Le Raspberry

Définition : Le **Raspberry Pi** est un nano-ordinateur monocarte à processeur ARM de la taille d'une carte de crédit conçu par des professeurs du département informatique de l'université de Cambridge dans le cadre de la fondation Raspberry Pi3

Objectif : Ce projet consiste à l'apprentissage et l'utilisation de nos connaissances en réseaux afin de développer une application.

Introduction

Pour ce projet, le Raspberry est utilisé comme serveur. Pour cela, il a été nécessaire d'effectuer plusieurs manipulations pour répondre aux différents critères afin de répondre aux contraintes imposés par le projet.

1. Allumage et configuration du Raspberry

Étape 1 - Les accessoires dont vous aurez besoin

Pour démarrer votre carte Raspberry Pi vous devez également posséder:

[Une alimentation](#) adaptée à votre carte. Si vous optez pour un câble USB, utilisez un adaptateur USB pour le brancher sur secteur.

[Une carte micro SD pré-chargée NOOBS.](#)

[Un câble HDMI vers micro HDMI](#)

[Un écran](#)

[Un clavier et une souris](#)

Étape 2: Raccorder le Raspberry Pi à ses accessoires

Une fois votre Raspberry Pi fraîchement sorti de son emballage, vous pouvez alors commencer l'installation. Vous devez désormais connecter votre Raspberry:

1. Insérer dans le compartiment prévu à cet effet la carte microSD
2. Brancher le clavier et la souris sur les ports USB
3. Connecter l'écran via le câble HDMI.
4. Si vous le souhaitez, vous pouvez vous connecter au réseau via le port Ethernet
5. Brancher des écouteurs si besoin à la prise audio
6. Connecter le Raspberry Pi au secteur via l'alimentation micro USB

Si tout fonctionne correctement la lumière rouge et verte s'affichent sur le Raspberry et NOOBS se lance sur l'écran.

Étape 3: Démarrer votre Raspberry Pi avec votre carte NOOBS

1. Une fois le Raspberry Pi allumé, laissez le s'ouvrir.

2. Lorsque le RPi vous demande le login, rentrer ce que vous voulez et faites la même chose pour le mot de passe.
3. Vous pouvez maintenant découvrir l'aspect graphique de NOOBS.

2. Mise à jour de notre Raspberry

Tout d'abord, la première étape est de vérifier que votre Raspberry est mis à jour. Pour cela, il faut simplement effectuer la manipulation suivante dans le terminal :

sudo apt update

Cette commande permettra au Raspberry de vérifier qu'il est à jour ou alors il vous enverra qu'il existe une liste d'éléments à mettre à jour en utilisant la commande suivante :

sudo apt upgrade

Une fois la manipulation effectuée il est alors temps de manipuler le Raspberry.

3 Configuration de notre Raspberry

Pour continuer sur l'installation de notre Raspberry, il faut ensuite le configurer en effectuant les manipulations suivantes :

1. il faut télécharger l'image de notre OS sur le site :<https://www.raspberrypi.org/downloads/raspbian/>
2. Ensuite, il faut ensuite utiliser **Raspberry Pi Imager** pour installer Raspberry Pi OS. Pour cela, il suffit simplement de charger l'archive préalablement téléchargée avec l'application et après quelques instants voilà un Raspberry configuré.

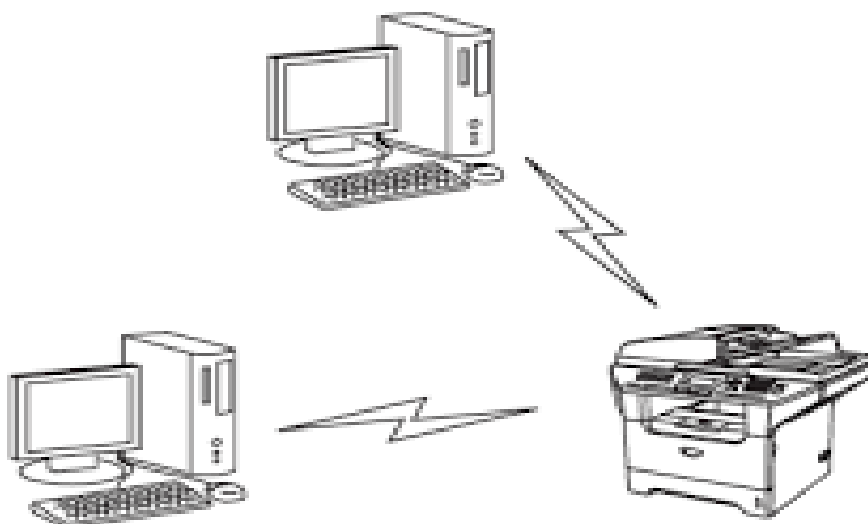
Préparer le Raspberry

Il est important de rappeler qu'à ce stade nous n'avons encore rien fait à part apprendre comment réussir à configurer le nano-ordinateur afin de le rendre utilisable.

Dans cette partie, nous allons voir comment configurer le Raspberry afin de le transformer en serveur. Celui-ci deviendra alors notre nouveau point d'accès contenant la suite du projet.

Tout d'abord, il faut commencer par effectuer quelques manipulations afin de mettre notre Raspberry en réseau Ad hoc.

Définition : Ad hoc est un type de [réseau sans fil](#) décentralisé. Le réseau est ad hoc car il ne s'appuie pas sur une infrastructure préexistante, comme des [routeurs](#) dans les réseaux filaires ou des [points d'accès](#) dans les réseaux sans fil administrés. Au lieu de cela, chaque nœud participe au routage en retransmettant les données aux autres nœuds, de façon que le choix du nœud qui va transmettre les données est opéré dynamiquement sur la base de la connectivité du réseau et de l'algorithme de routage utilisé.



Dans mon cas, étant donné que je suis seul sur ce projet je me suis orienté vers un système un peu plus simple. J'ai configuré mon Raspberry en tant que serveur et ainsi tout autre client qui se connecte à ce serveur peut jouer avec lui.

Maintenant je vais expliquer en plusieurs points comment j'ai opéré durant mon projet.

Raspberry Pi(RPI) mise en réseau ad-hoc

1. La démarche à suivre

Je résume l'installation d'un réseau ad hoc entre un RPI et un ordinateur portable Ubuntu.

Sur la RPI, mettre dans le fichier interface les lignes suivantes:

```
auto lo
iface lo inet loopback

auto wlan0 // Identifier wlan0 avec la commande iwconfig.
iface wlan0 inet static
address 192.168.1.3
netmask 255.255.255.0
wireless-mode ad-hoc
wireless-essid ADHOC // nom du réseau comme il vous plaît
wireless-channel 5
wireless-power on
```

Lorsque ce fichier existe, votre RPI passe en mode ad-hoc. Si vous le supprimer vous passez en mode wifi normal (grâce au fichier dhcpcd.conf)

Il faut ensuite de redémarrer son RPI et le tour est joué. Il suffit désormais de taper dans l'URL de mozilla firefox le mot suivant : localhost

Notre Raspberry fera apparaître un très beau panel de contrôle de notre RPI. Il faut ensuite choisir le mode de notre Raspberry. Pour la simplicité, j'ai décidé de mettre mon IP en statique afin de pouvoir travailler plus aisément durant ce projet.

Voilà pour le petit tutoriel permettant de configurer le Raspberry. Désormais la partie concernant celui-ci est terminé et il est désormais temps de préparer la partie jeu de ce projet.

pierre papier ciseaux

Dans le cadre de ce projet, le but est de concevoir une application implémenté sur le Raspberry qui fera office de serveur sur lequel peut se connecter un autre ordinateur afin de pouvoir jouer avec lui.

Pour ce projet, j'ai décidé de prendre un jeu simple auquel on joue à tous âge et qui permet de départager tout le monde. Pour coder le pierre papier ciseaux, il est nécessaire d'utiliser un langage de programmation que reconnaît le RPI.

Comme dit dans l'[introduction du guide](#), [le Raspberry Pi](#) a été conçu pour permettre aux étudiants d'apprendre la programmation. Le langage principal pour programmer sur Raspberry Pi est Python. Nous allons voir comment mettre en place un environnement de programmation en Python sur le Raspberry Pi.

Qu'est-ce que Python ?

Python est un puissant langage de programmation dynamique qui est utilisé dans différents domaines (application, web, ...) Il est souvent comparé à Perl, Ruby ou encore Java.

Les principaux points forts de Python sont :

- Langage adapté aux débutants
- Les grandes possibilités du langage. Il va aussi bien être adapté aux petits projets qu'aux gros projets.
- Peut-être cross-platform
- Stable, langage qui est assez vieux et qui a fait ses preuves
- Simple à comprendre

Le jeu(JanKenPon pour les adeptes)

De façon plus simple, voici comment ce jeu se joue et c'est sur cette base que je vais programmer mon jeu .



Sans trop rentrer dans le domaine de la programmation de fatiguer les yeux du lecteur avec des montagnes de code je vais surtout expliquer les principes de mon projet et comment j'ai réussi à programmer mon jeu.

Tout d'abord, il faut se rappeler que c'est un ordinateur qui va jouer contre le client et donc il faut toujours penser à lui attribuer une partie du code qui sera utilisé.

Si j'y vais par étape, mon code commence tout simplement par l'augmentation du score selon l'élément joué.

Dans mon cas le pierre papier ciseaux est basé sur 3 coups possibles (1 : pierre, 2 : papier, 3 : ciseaux) . J'ai donc tout simplement créé une fonction qui prend deux variables(mon_coup, ton_coup) qui sont des entiers compris entre 1 et 3. Selon quel coup est joué contre l'autre en respectant le schéma du jeu au dessus, alors j'augmente le score du joueur ou de l'ordinateur 1 si le coup est gagnant. Dans le cas où il y a égalité, alors personne ne gagne de point.

Il faut ensuite en théorie programmer la partie qui va jouer mais comme je voulais faire un jeu graphique, on va commencer par là puis revenir sur la partie jeu ensuite.

Tout d'abord, il existe une librairie sur python qui se nomme tkinter qui vous permet de programmer une interface graphique de votre jeu vidéo de façon dynamique. Une fois que cette librairie est importée, il suffit simplement de créer une fenêtre qui prend en paramètre Tkinter et je peux alors commencer. Il faut ensuite programmer la partie visible du jeu mais en réalité c'est un peu comme mettre des éléments dans des cases afin d'obtenir le format souhaité. Dans mon cas, il s'agit d'une fenêtre très simple avec des images et des boutons.

Pour comprendre il suffit simplement d'imaginer qu'il s'agit d'un tableau à 2 dimensions dans lequel j'ai placé ce que je voulais.



Il faut alors commencer à programmer les fonctions qui seront appelées quand chacun de ses boutons sera cliqué.

La fonction jouer contenant un paramètre(coup) permet selon le bouton qui est joué de charger son image dans la fenêtre du joueur et l'ordinateur va jouer un coup et faire la même chose selon un nombre aléatoire compris entre 1 et 3 et cela permettra alors de lancer l'augmentation du score selon le coup de chacun. À ce stade on peut jouer au jeu à l'infini.

Il manque alors plus qu'à réinitialiser les scores et donc à créer une fonction qui va remettre les scores à zéro et tout simplement vider les fenêtres du joueur ainsi que de l'ordinateur.

Le bouton quitter aura simplement la tâche de détruire la fenêtre du jeu.

L'étape finale du projet

Maintenant que l'on a fini de configurer le Raspberry et que le jeu est prêt il reste une dernière chose à faire. Il faut faire en sorte que ce soit mis sur le serveur et que celui-ci joue contre tout client.

1. Serveur

Pour faire le serveur il fallait faire plusieurs choses :

- Récupérer l'host ainsi que set un port quelconque afin de pouvoir recevoir un client.
- ensuite, il fallait à partir de ses données, créer un socket capable d'écouter tout client qui se connecte à celui-ci
- Il faut ensuite gérer le client qui va se connecter au serveur et pour cela, il faut vérifier qu'il se connecte(Dans mon cas, je suis le seul à me connecter au serveur et donc j'ai choisi de rester sur un unique client qui jouera avec notre ordinateur. Quand le joueur se connecte, le jeu se lance.

Une fois que le client est connecté, on attend qu'il joue et quand c'est fait, le serveur envoie son coup au jeu et la vérification de jeu est faite.

2. Client

Le client va se connecter au socket qui lui sera fourni par le serveur et quand ça sera fait, le jeu se lance et tout ce qu'il a à faire est de jouer avec le serveur.

Essai et conclusion

Durant ce projet, j'ai eu de nombreuses difficultés notamment pour programmer un jeu avec une interface graphique fonctionnelle ainsi qu'un serveur qui puisse jouer avec un serveur. Le tout fonctionne et durant ce projet j'ai appris énormément et notamment l'existence de Tkinter pour les interfaces graphiques ainsi que la facilité avec laquelle python permet de faire du code.