

Débuter avec Minecraft pour Raspberry Pi

Interagir avec Minecraft en Python

Par Fondation Raspberry Pi  - [Laurent Rosenfeld](#) (traducteur) 

Date de publication : 9 février 2016

TOUT PUBLIC

Dans ce tutoriel, vous allez explorer le monde virtuel de Minecraft Pi, l'édition spéciale de Minecraft pour Raspberry Pi.

Vous allez apprendre comment contrôler le joueur, construire manuellement en utilisant des blocs et utiliser l'interface Python pour manipuler le monde autour de vous.

Commentez

En complément sur Developpez.com

- [L'API Minecraft pour Raspberry Pi](#)

1 - Ce que vous allez apprendre.....	3
2 - Ce dont vous aurez besoin.....	3
2-1 - Matériel.....	3
2-2 - Logiciel.....	3
3 - Démarrez.....	3
3-1 - Lancer Minecraft.....	4
3-2 - Utilisation de l'interface de programmation en Python.....	6
3-2-1 - Trouvez votre position.....	7
3-2-2 - Téléportation.....	8
3-2-3 - Placer un bloc.....	8
3-2-4 - Les constantes de bloc.....	9
3-2-5 - Des variables de blocs.....	9
3-2-6 - Blocs spéciaux.....	10
3-2-7 - Blocs multiples.....	10
3-3 - Déposer des blocs au fur et à mesure que vous marchez.....	11
3-4 - Jouer avec les blocs de TNT.....	14
3-5 - S'amuser avec de la lave en fusion.....	15
4 - Aller plus loin.....	16
4-1 - Jeu en réseau.....	16
4-2 - Référence de l'interface.....	17
5 - Notes de la Rédaction de Developpez.com.....	17

1 - Ce que vous allez apprendre



En suivant ce tutoriel avec votre Raspberry Pi, vous apprendrez :

- comment accéder à Minecraft Pi et créer un nouveau monde ;
- comment utiliser l'environnement IDLE de programmation Python pour vous connecter à Minecraft Pi ;
- comment utiliser l'API Python Minecraft pour poster des textes dans la fenêtre de discussion, trouver les coordonnées du joueur, vous téléporter et construire des structures ;
- comment utiliser des variables pour stocker les identifiants des différents types de blocs ;
- expérimenter en plaçant différents types de blocs dotés d'attributs spéciaux.

2 - Ce dont vous aurez besoin

2-1 - Matériel

Aucun matériel supplémentaire n'est nécessaire pour utiliser ce tutoriel, en dehors bien sûr de votre Raspberry Pi et de ses périphériques habituels.

2-2 - Logiciel

Aucun logiciel supplémentaire n'est nécessaire pour utiliser ce tutoriel, en dehors de ce qui est préinstallé dans l'image Raspian actuelle. Pour garantir que votre carte SD est à jour, veuillez consulter la question relative à la mise à jour dans la **FAQ Raspberry Pi**.

3 - Démarrez

Minecraft est un jeu vidéo de construction, à monde ouvert et orienté « **bac à sable** ». Une version libre de Minecraft est disponible sur le Raspberry Pi. Il est également livré avec une interface de programmation. Ceci signifie que vous pouvez écrire des commandes et des scripts en Python pour construire automatiquement des choses dans le jeu. C'est une excellente façon d'apprendre le langage Python.



Dès que le jeu est chargé, cliquez sur **Start Game**, suivi de **Create New**. Vous constaterez que la fenêtre principale de l'application est légèrement décalée. Ceci signifie que pour déplacer la fenêtre, vous devez attraper la barre de titre située derrière la fenêtre Minecraft.



Vous êtes maintenant dans un jeu de Minecraft. Promenez-vous, explorez, bidouillez et construisez des choses.

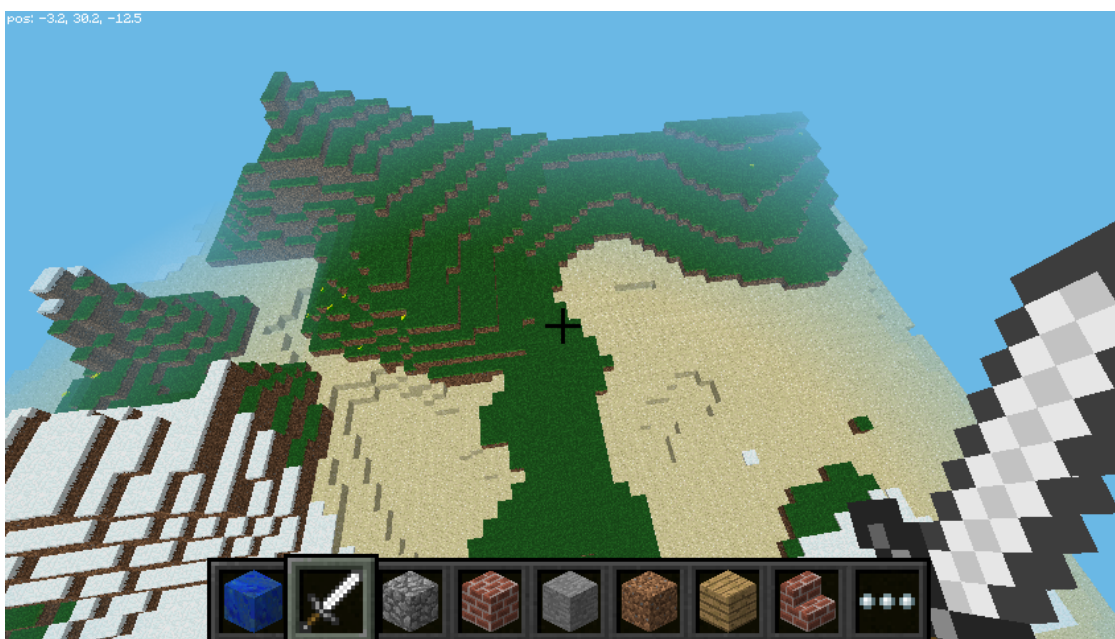
Utilisez la souris pour observer les alentours et utilisez les touches suivantes du clavier :

Touche	Action
W	Avance
A	À gauche
S	Arrière
D	À droite
E	Inventaire
Espace	Saute
Double espace	Vole/tombe
Esc	Pause/menu du jeu
Tab	Libère le curseur de la souris

Vous pouvez choisir un élément dans le panneau de sélection rapide avec la roue de défilement de la souris (ou en utilisant les chiffres) ou en tapant la touche E.



Vous pouvez aussi taper deux fois la barre d'espace, en la gardant appuyée la seconde fois, pour voler en l'air. Vous arrêtez de voler quand vous relâchez la barre d'espace, et si vous tapez à nouveau deux fois, vous retombez sur le sol.



Avec une épée dans votre main, vous pouvez cliquer sur les blocs devant vous pour les faire disparaître (ou pour creuser). Avec un bloc dans votre main, vous pouvez utiliser un clic droit pour placer le bloc devant vous, ou le clic gauche pour supprimer un bloc.

3-2 - Utilisation de l'interface de programmation en Python

Pendant que Minecraft tourne (et une fois le monde créé), amenez le focus hors du jeu en appuyant sur la touche Tab, ce qui libérera la souris. Ouvrez Python 3 à partir du menu application et déplacez les fenêtres en sorte qu'elles soient par exemple l'une à côté de l'autre.

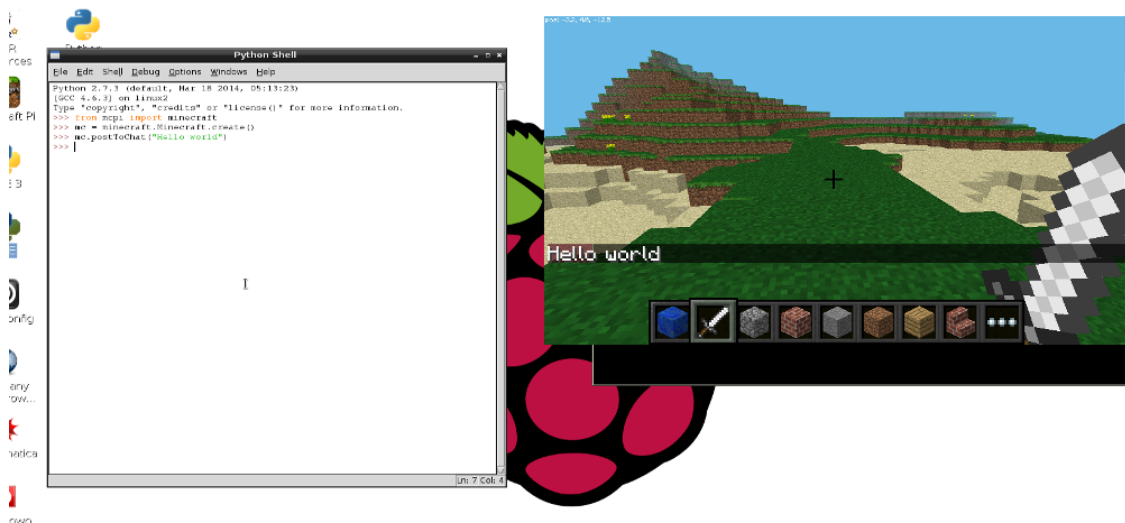
Vous pouvez soit taper des commandes directement dans la fenêtre Python, soit créer un fichier pour sauvegarder votre code et pouvoir le réutiliser.

Pour créer un fichier, allez à **File > New Window** ou **File > New File** de la fenêtre Python et, pour le sauvegarder, **File > Save**. Vous voudrez sans doute le sauvegarder dans votre répertoire personnel ou dans un nouveau répertoire projet.

Commencez par importer la bibliothèque Minecraft, établir une connexion avec le jeu et la tester en postant le message « Hello Word » à l'écran.

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
mc.postToChat("Hello world")
```

Si vous saisissez les commandes directement dans la fenêtre Python, tapez simplement la touche Entrée après chaque ligne. Si vous utilisez un fichier, sauvegardez-le avec **Ctrl + S** et lancez l'exécution avec la touche **F5**. À l'exécution, vous devriez voir votre message s'afficher sur l'écran du jeu.



3-2-1 - Trouvez votre position

Pour trouver votre position, tapez :

```
pos = mc.player.getPos()
```

La variable **pos** contient maintenant votre position ; vous pouvez accéder à chacune des trois coordonnées en utilisant **pos.x**, **pos.y** et **pos.z** :

```
>>> print(pos)
Vec3(0.5,0.0,0.5)
>>> print(pos.x)
0.5
```

Ou sinon, une jolie façon d'obtenir les coordonnées dans des variables séparées est d'utiliser la technique de dépaquetage de Python :

```
x, y, z = mc.player.getPos()
```

Les variables **x**, **y** et **z** contiennent maintenant chacune l'une des trois coordonnées de votre position. **x** et **z** sont les coordonnées horizontales pour la marche (en avant, en arrière, à gauche, à droite) et **y** est l'altitude.

Notez que la fonction **getPos()** renvoie la position du joueur au moment où elle est appelée, et si vous vous déplacez, vous devez appeler à nouveau cette fonction pour connaître le nouvel emplacement.

3-2-2 - Téléportation

Vous pouvez non seulement déterminer votre position actuelle, mais aussi choisir une position vers laquelle vous téléporter :

```
x, y, z = mc.player.getPos()
mc.player.setPos(x, y+100, z)
```

Ceci transportera votre joueur à une altitude de 100 mesures. Ceci signifie que vous vous téléporterez en plein ciel puis retombez là où vous trouviez.

Essayer de vous téléporter ailleurs !

3-2-3 - Placer un bloc

Vous pouvez placer un (seul) bloc à une position matérialisée par ses coordonnées avec la fonction `mc.setBlock()` :

```
x, y, z = mc.player.getPos()
mc.setBlock(x+1, y, z, 1)
```

Maintenant, un bloc de pierre doit apparaître près de là où vous vous trouvez. S'il n'est pas juste devant vous, il se trouve peut-être sur le côté ou derrière vous. Retournez dans la fenêtre Minecraft et utilisez la souris pour tourner sur vous-même jusqu'à ce que vous voyiez un bloc gris devant vous.



Les arguments passés à `setBlock()` sont `x`, `y`, `z` et `id`. Les arguments (`x`, `y`, `z`) sont les coordonnées dans le monde (nous avons spécifié une distance de 1 bloc de l'endroit où se trouve le joueur avec la valeur `x+1`) et `id` est l'identifiant du type de bloc que nous désirons placer. L'identifiant 1 correspond à un bloc de roche.

Vous pouvez essayer d'autres types de blocs :

- Air : 0 ;
- Herbe : 2 ;
- Terre : 3.

Maintenant que vous avez le bloc en vue, essayez de le changer en quelque chose d'autre :


```
mc.setBlock(x+1, y, z, 2)
```

Vous devriez voir le bloc de roche grise se transformer devant vos yeux.



3-2-4 - Les constantes de bloc

Vous pouvez utiliser des constantes de bloc pour placer ou modifier vos blocs, si vous connaissez leur nom. Il vous faut d'abord une nouvelle ligne d'importation :

```
from mcpi import block
```

Maintenant, vous pouvez placer un bloc en écrivant ceci :

```
mc.setBlock(x+3, y, z, block.STONE.id)
```

Les identifiants de blocs sont assez faciles à deviner, il faut les écrire entièrement en lettres CAPITALES. Voici quelques exemples pour vous habituer à leur nom :

- WOOD_PLANKS (planches)
- WATER_STATIONARY (eau stationnaire)
- GOLD_ORE (minerai d'or)
- GOLD_BLOCK (bloc d'or)
- DIAMOND_BLOCK (bloc de diamant)
- NETHER_REACTOR_CORE (cœur de réacteur du Nether)

3-2-5 - Des variables de blocs

Si vous connaissez l'identifiant d'un bloc, il peut être pratique de le mémoriser dans une variable. Vous pouvez utiliser soit le nombre entier de l'identifiant :

```
terre = 3
mc.setBlock(x, y, z, terre)
```

soit le nom du type de bloc :

```
terre = block.DIRT.id  
mc.setBlock(x, y, z, terre)
```

3-2-6 - Blocs spéciaux

Certains blocs ont des propriétés supplémentaires particulières. Par exemple le bloc WOOL (laine, identifiant 35) a une propriété supplémentaire permettant de spécifier la couleur. Pour ce faire, il faut ajouter un quatrième paramètre optionnel à la fonction `setBlock` :

```
wool = 35  
mc.setBlock(x, y, z, wool, 1)
```

Ici, le quatrième paramètre `1` donne une couleur orange. Si le quatrième paramètre n'est pas fourni, il prend la valeur par défaut (`0`), qui est la couleur blanche. Voici quelques autres couleurs :

- 2 : magenta ;
- 3 : bleu ciel ;
- 4 : jaune.

Essayez d'autres valeurs et observez les changements de couleur.

Parmi les autres types de blocs ayant des propriétés, on trouve par exemple :

- le bois (WOOD, identifiant 17) : chêne, épicéa, bouleau, etc. ;
- les herbes hautes (GRASS_TALL, identifiant 31) : arbuste, herbe, fougère, etc. ;
- la torche (TORCH, identifiant 50) : pointant vers l'est, le nord, l'ouest ou le sud.

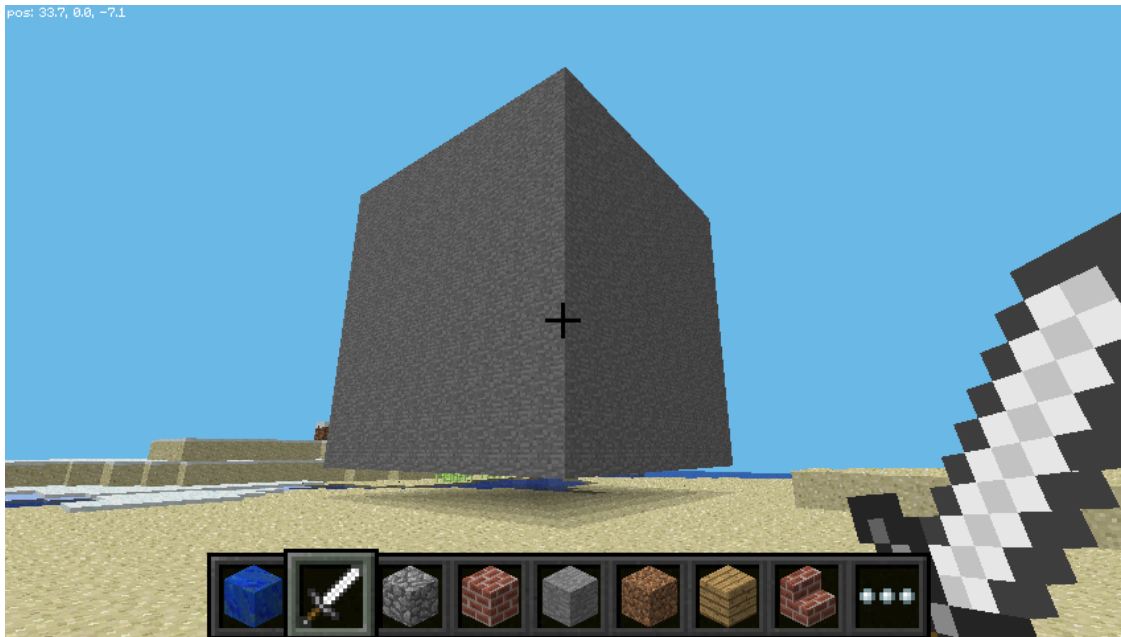
Il y en a bien d'autres. Voir le [guide de référence de l'API](#) pour les détails.

3-2-7 - Blocs multiples

Si la fonction `setBlock` permet de placer ou modifier un seul bloc, la fonction `setBlocks` vous permet de remplir tout un espace d'un seul coup :

```
roche = 1  
x, y, z = mc.player.getPos()  
mc.setBlocks(x+1, y+1, z+1, x+11, y+11, z+11, roche)
```

Ceci créera un volume cubique 10 x 10 x 10 de roche.



Vous pouvez créer des volumes plus grands avec `setBlocks`, mais ça peut prendre un peu de temps à générer.

3-3 - Déposer des blocs au fur et à mesure que vous marchez

Maintenant que vous savez poser des blocs, utilisons notre position mobile pour déposer des blocs au fur et à mesure de notre déplacement.

Le code suivant va déposer une fleur derrière vous chaque fois que vous marchez :

```

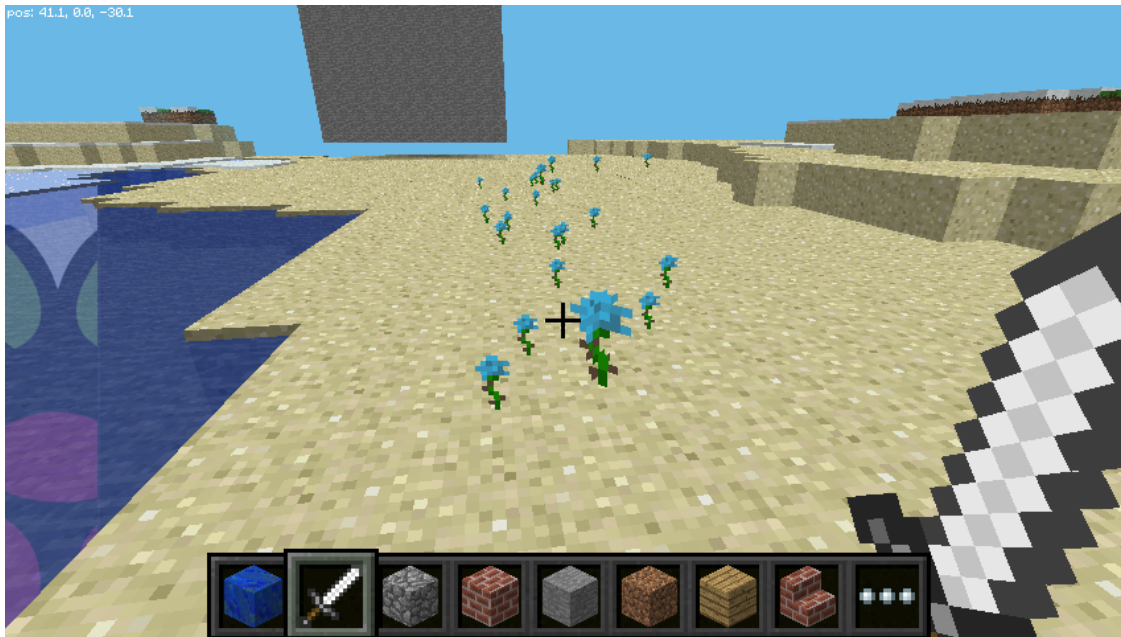
from mcpi.minecraft import Minecraft
from time import sleep

mc = Minecraft.create()

fleur = 38

while True:
    x, y, z = mc.player.getPos()
    mc.setBlock(x, y, z, fleur)
    sleep(0.1)
  
```

Maintenant, promenez-vous pour un moment puis retournez-vous pour voir les fleurs que vous avez semées derrière vous.



Comme nous avons utilisé une boucle `while True`, le code continuera sans fin. Pour l'arrêter, tapez `Ctrl + C` dans la fenêtre Python.

Essayez aussi de voler et observez les fleurs que vous laissez dans le ciel.



Supposons que nous voulions ne semer des fleurs que quand le joueur marche dans l'herbe. Nous pouvons utiliser la fonction `getBlock` pour connaître le type d'un bloc :

```
x, y, z = mc.player.getPos() # position du joueur : (x, y, z)
ce_bloc = mc.getBlock(x, y, z) # Identifiant du bloc
print(ce_bloc)
```

Mais ceci donne la position du bloc *dans lequel* nous nous trouvons (cela donnera 0, car nous nous trouvons *dans* un bloc d'air). Nous voulons trouver le type de bloc *sur lequel* nous marchons. Pour cela, nous retranchons 1 à la valeur y (l'altitude) et utilisons `getBlock` pour déterminer le bloc sur lequel nous nous trouvons :

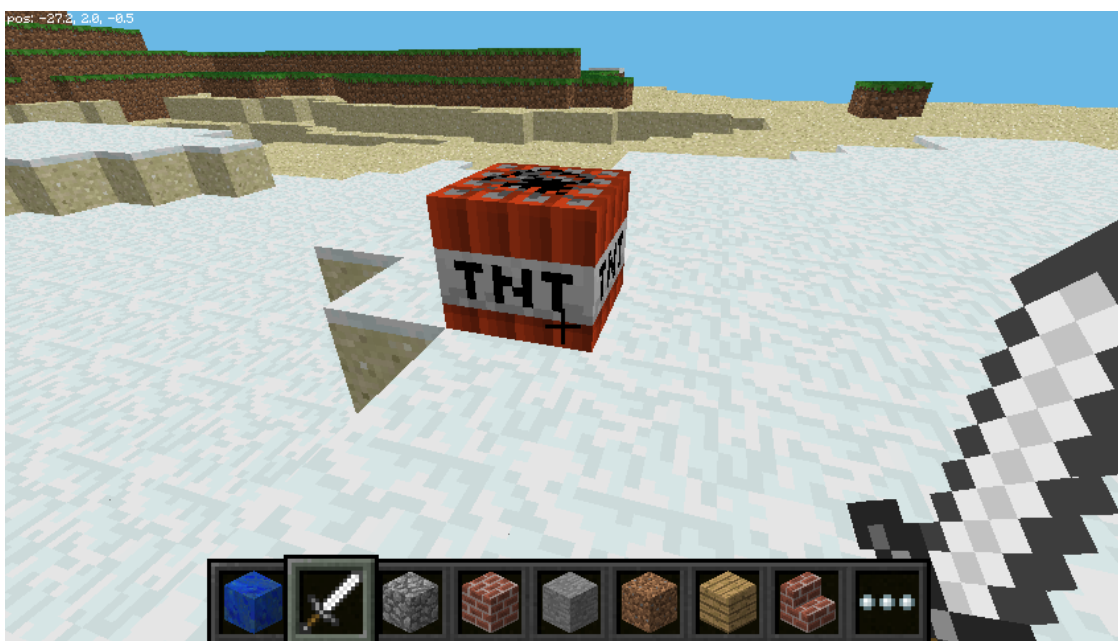
```
x, y, z = mc.player.getpos() # position du joueur : (x, y, z)
```




3-4 - Jouer avec les blocs de TNT

Un autre type de bloc intéressant est le bloc de TNT ! Pour placer un bloc normal de TNT, utilisez :

```
tnt = 46
mc.setBlock(x, y, z, tnt)
```



Ce bloc de TNT n'est cependant pas très intéressant. Ajoutez un paramètre additionnel 1 :

```
tnt = 46
mc.setBlock(x, y, z, tnt, 1)
```

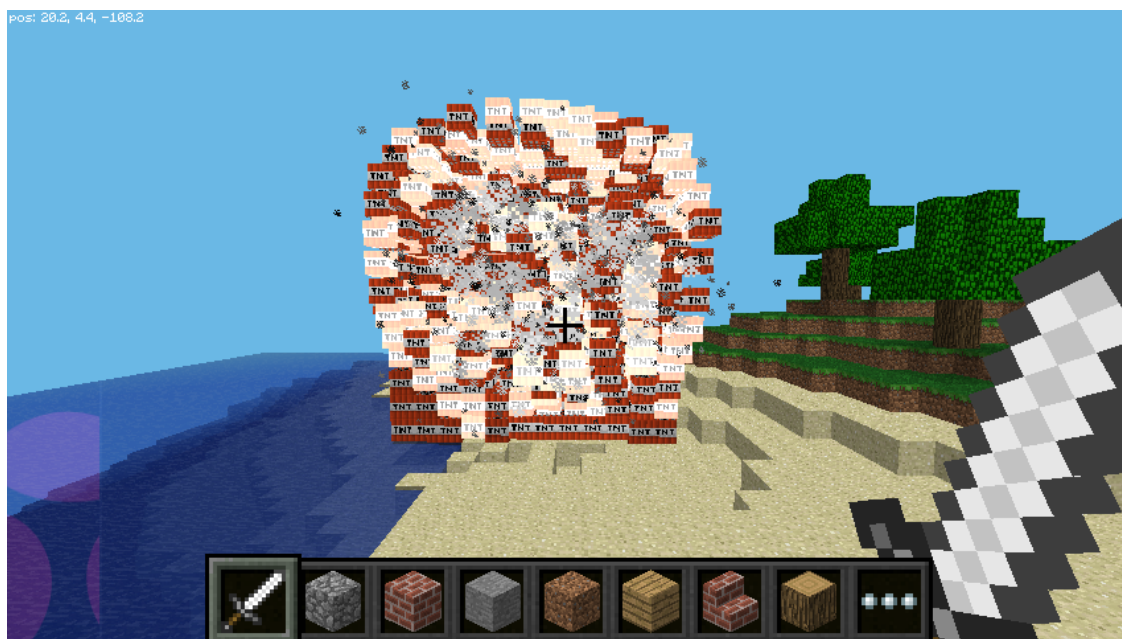
Maintenant, utilisez l'épée et faites un clic gauche sur le bloc de TNT. Celui-ci sera armé et explosera environ une seconde plus tard !

Essayez maintenant de construire un gros cube de blocs de TNT :


```
tnt = 46
mc.setBlocks(x+1, y+1, z+1, x+11, y+11, z+11, tnt, 1)
```



Vous devriez maintenant voir ce gros cube de TNT. Approchez-vous et activez l'un des blocs, puis enfuyez-vous au plus vite et observez la scène. Il se peut que le rendu graphique soit un peu lent, car il y a vraiment beaucoup de choses à changer en même temps.



3-5 - S'amuser avec de la lave en fusion

Un type de bloc très amusant à utiliser est la lave en fusion.

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()

x, y, z = mc.player.getPos()
lave = 10
mc.setBlock(x+3, y+3, z, lave)
```

Trouvez le bloc que vous venez de placer et vous devriez voir de la lave s'écouler du bloc vers le sol.

Ce qui est amusant avec la lave, c'est qu'elle se solidifie et se transforme en roche quand elle refroidit. Déplacez-vous ailleurs dans votre monde et essayez ceci :

```

from mcpi.minecraft import Minecraft
from time import sleep

mc = Minecraft.create()

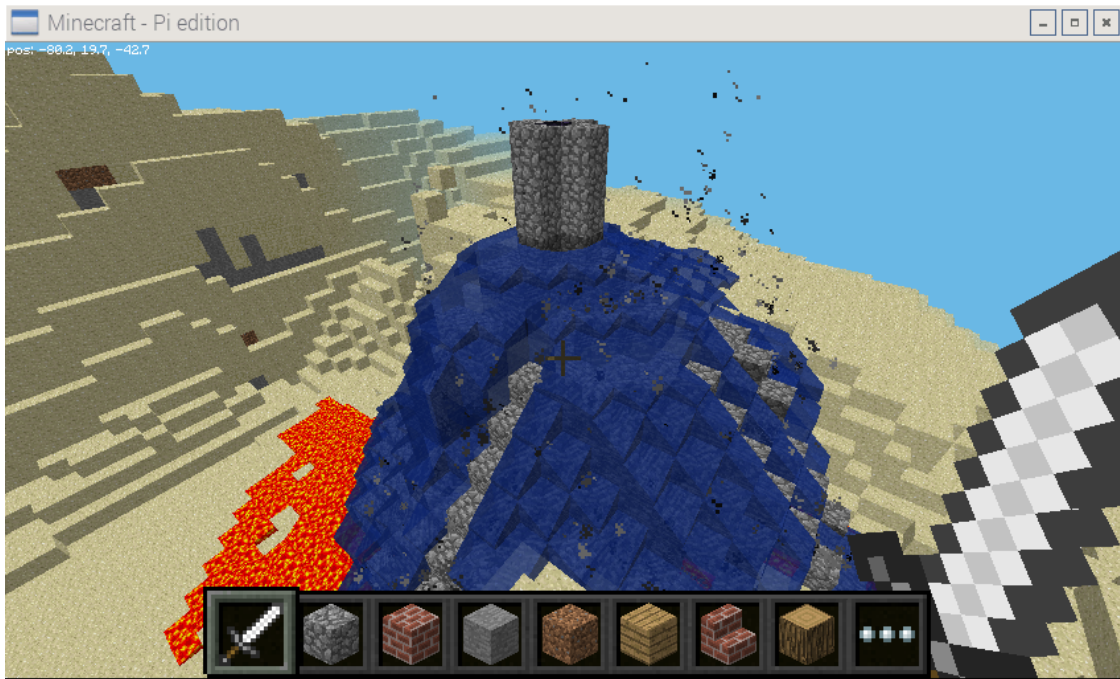
x, y, z = mc.player.getPos()

lave = 10
eau = 8
air = 0

mc.setBlock(x+3, y+3, z, lave)
sleep(20)
mc.setBlock(x+3, y+5, z, eau)
sleep(4)
mc.setBlock(x+3, y+5, z, air)

```

Vous pouvez ajuster le paramètre de la fonction `sleep` pour moduler la quantité de lave qui s'écoule.



4 - Aller plus loin

Il y a plein de choses que vous pouvez faire maintenant que vous savez utiliser le monde de Minecraft et employer l'interface Python.



4-1 - Jeu en réseau

Si plusieurs joueurs connectent leur Raspberry Pi à un réseau local, ils peuvent se connecter au même monde Minecraft et jouer ensemble. Les joueurs se verront les uns les autres dans le jeu.

4-2 - Référence de l'interface

Vous trouverez une documentation plus approfondie sur les fonctionnalités disponibles et une liste des identifiants de blocs dans le **guide de l'API Python**.

5 - Notes de la Rédaction de Developpez.com

Cet article est une traduction du tutoriel issu des ressources du  **site officiel de la Fondation Raspberry Pi**, paru sous le titre  **Getting started with Minecraf Pi**.

Nous remercions les membres de la Rédaction de Developpez pour le travail de traduction et de relecture qu'ils ont effectué, en particulier :

- **Lolo78**
- **f-leb**
- **Claude Leloup**