

The background is a pixelated Super Mario Bros. style scene. It features a green grassy field with a blue sky above and a grey brick floor at the bottom. Scattered throughout the field are several small, pixelated enemies: Goombas (brown with red hats) and Koopas (blue with red hats). There are also some red coins and a single tree on the right side. The title text is centered in the grassy area.

INITIATION À LA PROGRAMMATION EN PYTHON

Bastien Gorissen & Thomas Stassin

READY ?

Press start!

PROLOGUE

“Dernier jour d’initiation à la
programmation...”

NOW YOU KNOW...

Algorithm (noun.)

Word used by programmers when...
they do not want to explain what they did.

JEU DU JOUR

“Pit People”
par Ariane





LA DERNIÈRE FOIS...

On a vu les fonctions...

... et personne n'est mort...

... pas encore du moins.



AUJOURD'HUI, AU MENU:

- Révisions...
- Ajout de monstres dans le donjon
- Conclusion

LEVEL 4-1

Révisions des 100000 ... secondes

EXERCICE 1-1: VIVRE LA VIE DE STANLEY

Écrivez un programme qui demandera à l'utilisateur de taper une lettre de l'alphabet. La lettre que l'utilisateur devra entrer sera choisie au hasard parmi toutes les lettres de l'alphabet.

Le programme demandera cette opération en boucle tant que l'utilisateur ne tape pas la bonne lettre.

EXERCICE 1-2: VIVRE LA VIE DE STANLEY

Faites en sorte que le programme boucle 5 fois avant de s'arrêter.

Au bout de 5 fois le programme affiche le nombre de fois que l'utilisateur a commis une erreur (et donc s'est trompé de touche).

EXERCICE 1-3: VIVRE LA VIE DE STANLEY

Modifier le programme pour qu'après chaque boucle de 5 fois, le programme demande à l'utilisateur s'il veut continuer.

Le programme continue si l'utilisateur entre la lettre 'O' ou 'o'.

EXERCICE 2-1: SURVOLE

Créez un script:

Créez une liste de mots avec les mots suivant dedans:

“disorder”, “enemy”, “humanity”, “is” “of”, “the”, “true”

Faites en sorte que le programme affiche tous les mots de cette liste dans un ordre aléatoire sans jamais répéter un mot.

Un exemple possible de réponse pourrait être:

“the true enemy of humanity is disorder”

EXERCICE 2-2: SURVOLE

Faites en sorte de mettre la partie qui mélange la liste dans une fonction, la fonction prendra en paramètre une liste de mots et retournera une chaîne de caractère avec la phrase créée.

Ajoutez une autre liste de mots et appelez la fonction pour chaque listes.

Nouvelle liste:

“for”, “high”, “let’s”, “new”, “score”, “shoot”, “the”

EXERCICE 3-1: AU PAS DE COURSE

On va simuler un jeu de course.

Dans un dictionnaire, stockez les données de trois personnes:

En clé leur nom et en valeur la distance parcourue représentée par un entier.

Dans un premier temps affichez le nom des personnes qui courent.

(En gros affichez les clés du dictionnaire)

EXERCICE 3-2: AU PAS DE COURSE

Bouclez tant que l'un des joueurs n'a pas parcouru au moins 100 m.

A chaque itération de la boucle, pour chaque coureur, additionnez la distance parcourue à un entier choisi aléatoirement entre 2 et 5.

A la fin de la boucle affichez la distance parcourue par chacun.

EXERCICE 3-3: AU PAS DE COURSE

Ajoutez dans la boucle une partie de code:

Le coureur a une chance sur trois d'avoir un point de côté. Dans ce cas là, il n'avance que de la moitié de la distance prévue et le programme affiche que le joueur ralenti.

HOMework

DEVOIR (PROCHAIN COURS LE 24/01)

Faites un script qui demande à l'utilisateur d'entrer des chiffres au clavier.

Il doit entrer des chiffres au clavier tant que la somme des chiffres entrés n'est pas égale au supérieur à 20.

Après cela le script affichera le nombre de chiffres entrés au clavier ainsi que le plus petit et le plus grand d'entre eux.

LEVEL 4-2

Dungeons & “Dragons”

LE RETOUR DE LA VENGEANCE DU DONJON

Nous allons reprendre le code du donjon, et observer quelques changements...

Par exemple, vous pouvez maintenant bouger le personnage vous-même !

Les flèches gauche et droite font pivoter le personnage de 90°, et la flèche vers le haut le fait avancer tout droit d'une case.

LE RETOUR DE LA VENGEANCE DU DONJON

Le donjon est également moins vide ! Une créature clairement identifiable comme un... enfin une... une créature donc, a élu domicile dans le donjon.

Cette créature, aussi menaçante soit-elle, n'est pas encore le summum de la menace, vous l'aurez constaté.

Notre but va être de la rendre un peu plus maline. De lui créer une intelligence, qu'on pourrait qualifier d'artificielle.

VOYAGE À NEW YORK

Notre but étant de rendre le monstre un peu plus intelligent, nous allons tenter de lui donner un comportement du type :

- Si tu es loin du héros, bouge dans une direction aléatoire.
- Si tu es proche du héros, essaye de t'en rapprocher.

Pour décider si le monstre est proche du héros, nous allons utiliser la “distance de Manhattan”.

What ?

VOYAGE À NEW YORK

La distance de Manhattan doit son nom à la façon dont on décrit la distance dans les villes modernes avec leur joli plan en grille...

On parle dans ce cas de nombre de “blocs” à parcourir dans une direction, et puis dans la direction perpendiculaire.



ON M'A PROMIS UN VOYAGE, MAIS J'AI EU DES MATHS T_T

Pour calculer la distance de Manhattan, il nous faut un tout petit peu de maths (mais promis, c'est très léger).

Si les positions des deux points sont (x_1, y_1) et (x_2, y_2) , la distance sera la “différence” entre les x , ajoutée à la différence entre les y .



ON M'A PROMIS UN VOYAGE, MAIS J'AI EU DES MATHS T_T

Seulement, la différence peut être négative, du coup c'est un peu ennuyant.

Heureusement, il y a en Python (en fait en math aussi, mais chut) une fonction qui permet de rendre un négatif positif.

La “valeur absolue”, ou **abs()**.

Testez dans la console Python:

```
abs(-5)
```

```
abs(3)
```

ON M'A PROMIS UN VOYAGE, MAIS J'AI EU DES MATHS T_T

De façon crue et brutale, la distance de Manhattan est donc:

distance = abs(x2 - x1) + abs(y2 - y1)

Ou bien, si les positions sont dans des tuples par exemple

distance = abs(pos2[0] - pos1[0]) + abs(pos2[1] - pos1[1])

Oh boy.

On va mettre ça dans une fonction, right ? =D

CACHEZ-MOI CES CALCULS !

Vous allez donc définir une fonction qui renvoie la distance de Manhattan entre **hero** et **monster**.

Et nous allons ensuite retourner à notre objectif.

Qui sera votre dernier exercice(*) !

DUN DUN DUUUUUUUN !

(*) en deux parties :p

EXERCICE... PART I

Placez dans le donjon deux monstres. Arrangez-vous pour qu'ils soient sur le trajet du héros.

Codez leur comportement pour qu'ils ne fasse rien sauf si le héros ne se trouve à une distance de Manathan de 4 ou moins d'eux.

Si le héros est proche, faites-les bouger dans une direction choisie au hasard.

IA

On commence à avoir un début d'IA (Intelligence Artificielle).

Même si c'est basique actuellement les monstres “prennent des décisions” en fonction du héros.

On pourrait aller un peu plus loin en simulant l'envie de bouger du monstre.

EXERCICE... PART II

Toujours en considérant la distance de Manhattan.

Faites en sorte que les monstres ne bougent que sur un pourcentage.

Prenez la distance héros - monstre.

Si elle est supérieure à 6, il ne bouge pas.

Entre 6 et 3 il ne bouge que dans 50% des cas

Moins de 3, il bouge à coups sûr.

EXERCICE... PART III (THE LOST LEVELS)

Le top du top, serait évidemment que le monstre se déplace plus intelligemment.

Pour ce faire, plutôt que de choisir une direction aléatoire, il vous faudrait prendre une direction qui le rapproche du héros !

Des idées ?

Double bonus : **`monster.is_tile_free("left")`** renvoie si la tile à gauche est libre. Ca pourrait être utile...

ÉPILOGUE

“Où notre fine équipe s’arrête pour se remémorer, émue, la première étape de leur quête.”

THE STORY SO FAR...

Notre but dans ce cours d'orientation a vraiment été de vous donner le goût de Python et de la programmation en général.

C'est un mode de réflexion un peu particulier, mais on espère que ça vous a plu.

C'est probablement un peu différent de ce que vous imaginiez en pensant “création de jeu vidéo”, mais c'est parce que vous êtes au début de votre chemin.

Voyez-ça comme l'épreuve initiatique.

THE STORY SO FAR...

C'est un peu comme apprendre une nouvelle langue : ici nous avons eu comme objectif de vous donner les bases de la grammaire, et le vocabulaire de base.

Bientôt, vous serez en mesure d'utiliser tout ce que nous avons vu (et allons voir) pour traduire vos idées en scripts qui les réalisent.

Et ça, c'est quand même top...

;))

-Bastien & Thomas