

The background of the slide is a pixelated image of a Sonic the Hedgehog game level. It features a green grassy field with a blue sky at the top and a grey stone path at the bottom. Scattered across the field are several small, pixelated objects: blue and red rings, a red and white checkered flag, a small blue bush, and a small green tree with a yellow top. The title text is centered over the grassy area.

# INITIATION À LA PROGRAMMATION EN PYTHON

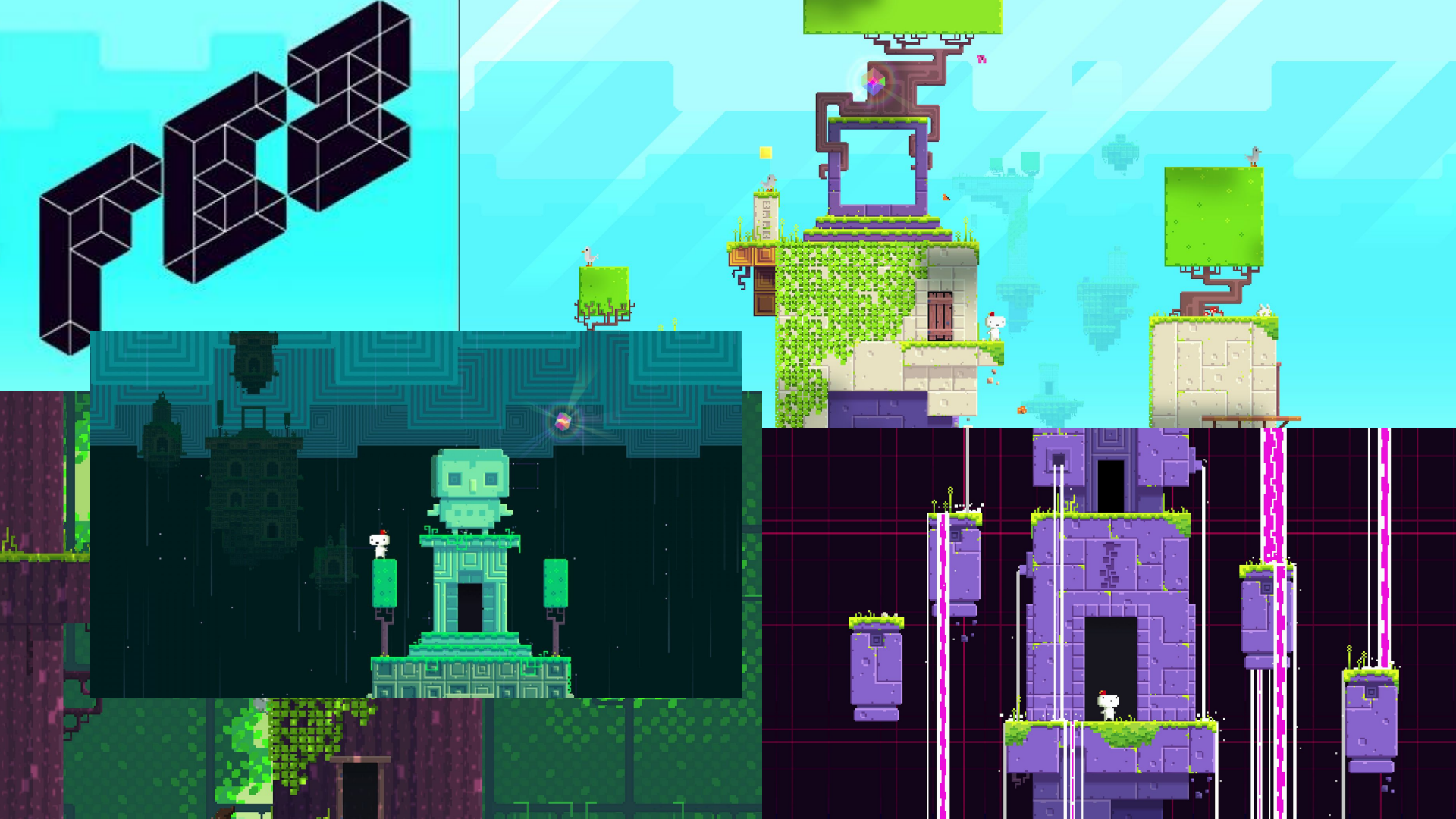
**Bastien Gorissen & Thomas Stassin**

# READY ?

Press start!

# JEU DU JOUR

“FEZ”





# LEVEL 2-5

“BOSS BATTLE”

# A VOUS DE JOUER!

Nous allons réaliser un petit jeu de “pendu”.

Pour cela, il va falloir:

- Une liste de mots à trouver
- Une liste des lettres proposées par l'utilisateur
- Imprimer le mot à trouver avec uniquement les lettres devinées, et des “-” pour remplacer les autres.
- Détecter quand le joueur a trouvé toutes les lettres.

# LEVEL 2-6



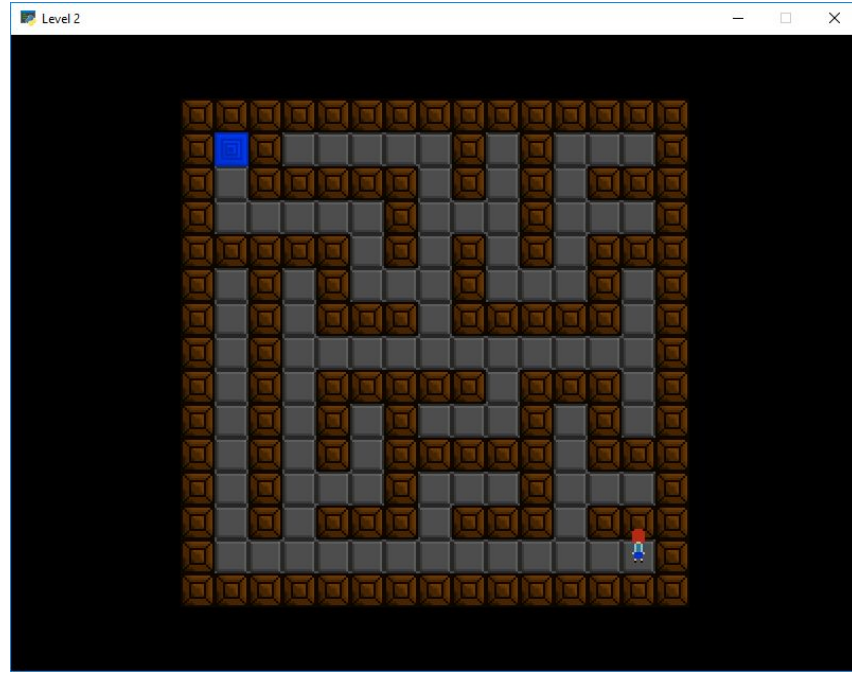


# A VOUS DE JOUER!

En reprenant le code du petit jeu de donjon, nous allons modifier le fichier `world.py`...

- Mettez en commentaire la ligne 214 (pour rappel, un "#" devant la ligne en fait un commentaire ignoré par Python)
- Décommentez la ligne 215
- Lancez le nouveau jeu...

# DUN DUN DUUUUUUN!



# A VOUS DE JOUER!

Le labyrinthe change à chaque fois!

Comment allez-vous faire pour vous en sortir ??

- N'oubliez pas les actions possibles : **hero.turn\_left()**, **.turn\_right()**, **.move()**, **.is\_tile\_free()**
- Avec **hero.is\_at\_goal()** vous pouvez savoir si vous êtes sur la sortie !
- ATTENTION : ne faites pas **while not hero.is\_at\_goal()** !  
Ca semble une bonne idée mais pour des raisons techniques ça va freezer.
- Une technique pour sortir d'un labyrinthe? Toujours tourner dans le même sens si possible à chaque croisement