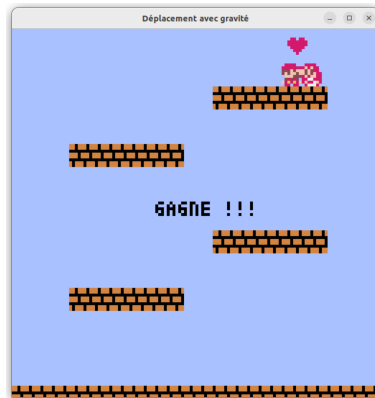


Programmer un jeu vidéo avec Pyxel

Jeu de plateforme

Seconde/Première



Objectifs

L'objectif de ce petit projet est de vous familiariser avec la création de jeux vidéos à l'aide du moteur de jeux « rétro » Pyxel, logiciel libre et open-source.

Objectifs

L'objectif de ce petit projet est de vous familiariser avec la création de jeux vidéos à l'aide du moteur de jeux « rétro » Pyxel, logiciel libre et open-source.

En plusieurs étapes, il s'agit ici de compléter le code d'un programme afin d'obtenir un jeu du type plateforme :

Objectifs

L'objectif de ce petit projet est de vous familiariser avec la création de jeux vidéos à l'aide du moteur de jeux « rétro » Pyxel, logiciel libre et open-source.

En plusieurs étapes, il s'agit ici de compléter le code d'un programme afin d'obtenir un jeu du type plateforme :

- **Étape 1** : Création de la fenêtre de base.

Objectifs

L'objectif de ce petit projet est de vous familiariser avec la création de jeux vidéos à l'aide du moteur de jeux « rétro » Pyxel, logiciel libre et open-source.

En plusieurs étapes, il s'agit ici de compléter le code d'un programme afin d'obtenir un jeu du type plateforme :

- **Étape 1** : Création de la fenêtre de base.
- **Étape 2** : Création du sol et du personnage.

Objectifs

L'objectif de ce petit projet est de vous familiariser avec la création de jeux vidéos à l'aide du moteur de jeux « rétro » Pyxel, logiciel libre et open-source.

En plusieurs étapes, il s'agit ici de compléter le code d'un programme afin d'obtenir un jeu du type plateforme :

- **Étape 1** : Création de la fenêtre de base.
- **Étape 2** : Création du sol et du personnage.
- **Étape 3** : Ajouter les déplacements horizontaux du personnage.

Objectifs

L'objectif de ce petit projet est de vous familiariser avec la création de jeux vidéos à l'aide du moteur de jeux « rétro » Pyxel, logiciel libre et open-source.

En plusieurs étapes, il s'agit ici de compléter le code d'un programme afin d'obtenir un jeu du type plateforme :

- **Étape 1** : Création de la fenêtre de base.
- **Étape 2** : Création du sol et du personnage.
- **Étape 3** : Ajouter les déplacements horizontaux du personnage.
- **Étape 4** : Ajouter la gravité pour faire sauter le personnage.

Objectifs

L'objectif de ce petit projet est de vous familiariser avec la création de jeux vidéos à l'aide du moteur de jeux « rétro » Pyxel, logiciel libre et open-source.

En plusieurs étapes, il s'agit ici de compléter le code d'un programme afin d'obtenir un jeu du type plateforme :

- **Étape 1** : Création de la fenêtre de base.
- **Étape 2** : Création du sol et du personnage.
- **Étape 3** : Ajouter les déplacements horizontaux du personnage.
- **Étape 4** : Ajouter la gravité pour faire sauter le personnage.
- **Étape 5** : Ajouter des plateformes.

Objectifs

L'objectif de ce petit projet est de vous familiariser avec la création de jeux vidéos à l'aide du moteur de jeux « rétro » Pyxel, logiciel libre et open-source.

En plusieurs étapes, il s'agit ici de compléter le code d'un programme afin d'obtenir un jeu du type plateforme :

- **Étape 1** : Création de la fenêtre de base.
- **Étape 2** : Création du sol et du personnage.
- **Étape 3** : Ajouter les déplacements horizontaux du personnage.
- **Étape 4** : Ajouter la gravité pour faire sauter le personnage.
- **Étape 5** : Ajouter des plateformes.
- **Étape 6** : Ajouter des collisions avec les plateformes.

Objectifs

L'objectif de ce petit projet est de vous familiariser avec la création de jeux vidéos à l'aide du moteur de jeux « rétro » Pyxel, logiciel libre et open-source.

En plusieurs étapes, il s'agit ici de compléter le code d'un programme afin d'obtenir un jeu du type plateforme :

- **Étape 1** : Création de la fenêtre de base.
- **Étape 2** : Création du sol et du personnage.
- **Étape 3** : Ajouter les déplacements horizontaux du personnage.
- **Étape 4** : Ajouter la gravité pour faire sauter le personnage.
- **Étape 5** : Ajouter des plateformes.
- **Étape 6** : Ajouter des collisions avec les plateformes.
- **Étape 7** : Ajouter un scénario de jeu.

Principes généraux des jeux vidéos

Un jeu vidéo peut se résumer ainsi :

Principes généraux des jeux vidéos

Un jeu vidéo peut se résumer ainsi :

Une boucle infinie fait progresser le jeu, à chaque tour :

1. On écoute les interactions du joueur ;
2. On met à jour l'état du jeu ;
3. On dessine les éléments à l'écran ;
4. On attend quelques millisecondes

Principes généraux des jeux vidéos

Un jeu vidéo peut se résumer ainsi :

Une boucle infinie fait progresser le jeu, à chaque tour :

1. On écoute les interactions du joueur ;
2. On met à jour l'état du jeu ;
3. On dessine les éléments à l'écran ;
4. On attend quelques millisecondes

Dans Pyxel,

- la boucle infinie est implicite, et l'attente des quelques millisecondes déjà prise en charge.

Principes généraux des jeux vidéos

Un jeu vidéo peut se résumer ainsi :

Une boucle infinie fait progresser le jeu, à chaque tour :

1. On écoute les interactions du joueur ;
2. On met à jour l'état du jeu ;
3. On dessine les éléments à l'écran ;
4. On attend quelques millisecondes

Dans Pyxel,

- la boucle infinie est implicite, et l'attente des quelques millisecondes déjà prise en charge.
- des fonctions prédéfinies gèrent les actions 2. et 3.

Mettre à jour l'état du jeu → `update()`

Dessiner les éléments à l'écran → `draw()`


Le squelette d'un programme Pyxel

```
1 # on importe la bibliothèque pyxel
2 import pyxel
3
4 def update():
5     """ met à jour les variables (30 fois par seconde) """
6     pass
7
8 def draw():
9     """ crée les objets (30 fois par seconde) """
10    pyxel.cls(6)  # Efface l'écran avec une couleur bleu clair
11
12 # on initialise de la taille de la fenêtre 128x128 pixels
13 pyxel.init(128, 128, title="Déplacement avec gravité")
14 # on lance l'exécution du jeu
15 pyxel.run(update, draw)
```

Le squelette d'un programme Pyxel

En route vers l'étape 1


Sur le bureau de votre ordinateur :

- ouvrir le dossier `activite_mario` ;
- double-cliquer sur le fichier `etape1.py` ;
- exécuter le programme en appuyant sur  .

Le squelette d'un programme Pyxel

En route vers l'étape 1

Sur le bureau de votre ordinateur :

- ouvrir le dossier `activite_mario` ;
- double-cliquer sur le fichier `etape1.py` ;
- exécuter le programme en appuyant sur  .

On code

Changer, à votre guise, la couleur du fond en utilisant la palette de couleur.

0 H000000 0,0,0	1 H2B335F 43,51,95	2 H7E2072 126,32,114	3 H3E9DA2 62,157,162
4 HBB4B52 139,72,82	5 H466495 70,100,149	6 HA9C1FF 169,193,255	7 HEEEEEE 238,238,238
8 HD41B6C 212,24,108	9 H0D996E 221,153,110	10 HEECF5D 238,207,93	11 H70C6A9 112,198,169
12 H67BA04 103,138,212	13 H969696 150,150,150	14 HE69AA2 230,154,162	15 HEDC78D 237,199,176

Étape 1 : Création de la fenêtre de base

Dans notre grille de dessin :

- l'origine de la grille est en haut à gauche ;
- l'axe des abscisses est orienté vers la droite ;
- l'axe des ordonnées est orienté vers le bas.

Étape 1 : Création de la fenêtre de base

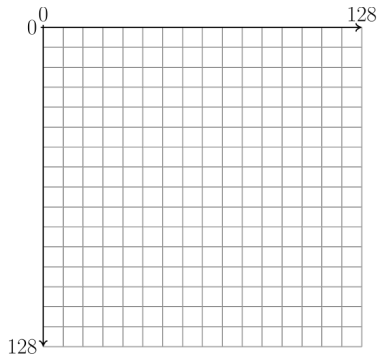
Dans notre grille de dessin :

- l'origine de la grille est en haut à gauche ;
- l'axe des abscisses est orienté vers la droite ;
- l'axe des ordonnées est orienté vers le bas.

Exercice

Dans notre jeu, la grille a pour dimensions 128x128 pixels et 16 cases par 16.

De combien de pixels une case est-elle composée ?



Étape 1 : Création de la fenêtre de base

Dans notre grille de dessin :

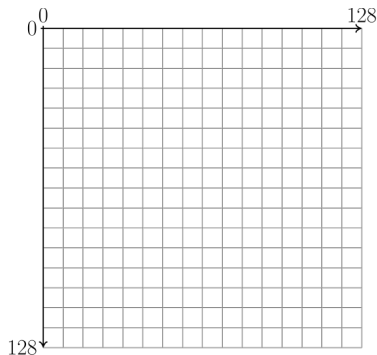
- l'origine de la grille est en haut à gauche ;
- l'axe des abscisses est orienté vers la droite ;
- l'axe des ordonnées est orienté vers le bas.

Exercice

Dans notre jeu, la grille a pour dimensions 128x128 pixels et 16 cases par 16.

De combien de pixels une case est-elle composée ?

Chaque case est un carré de côté 8 pixels.



Étape 2 : Création du sol et du personnage

Au départ du jeu, notre personnage sera représenté par un carré bleu de côté 8 pixels centré sur une plateforme de hauteur 4 pixels.

Exercice

Quelles sont les coordonnées du sommet en haut à gauche du carré représentant le personnage ?

Représenter notre personnage sur le document donné.

Étape 2 : Création du sol et du personnage

Au départ du jeu, notre personnage sera représenté par un carré bleu de côté 8 pixels centré sur une plateforme de hauteur 4 pixels.

Exercice

Quelles sont les coordonnées du sommet en haut à gauche du carré représentant le personnage ?

Représenter notre personnage sur le document donné.

Les coordonnées du sommet en haut à gauche doivent être (60 ; 116).

Étape 2 : Création du sol et du personnage

Au départ du jeu, notre personnage sera représenté par un carré bleu de côté 8 pixels centré sur une plateforme de hauteur 4 pixels.

Exercice

Quelles sont les coordonnées du sommet en haut à gauche du carré représentant le personnage ?

Représenter notre personnage sur le document donné.

Les coordonnées du sommet en haut à gauche doivent être (60 ; 116).

On code

Dans la zone d'initialisation du programme, créer deux variables (dites *globales*) `player_x` et `player_y` initialisées toutes les deux avec les valeurs 60 et 116.

Ajouter, ensuite, la variable : `GROUND` initialisée à (0, 124, 128, 4).

Exécuter, enfin, votre programme. Que se passe-t-il ?

Étape 2 : Création du sol et du personnage

Au départ du jeu, notre personnage sera représenté par un carré bleu de côté 8 pixels centré sur une plateforme de hauteur 4 pixels.

Exercice

Quelles sont les coordonnées du sommet en haut à gauche du carré représentant le personnage ?

Représenter notre personnage sur le document donné.

Les coordonnées du sommet en haut à gauche doivent être (60 ; 116).

On code

Dans la zone d'initialisation du programme, créer deux variables (dites *globales*) `player_x` et `player_y` initialisées toutes les deux avec les valeurs 60 et 116.

Ajouter, ensuite, la variable : `GROUND` initialisée à (0, 124, 128, 4).

Exécuter, enfin, votre programme. Que se passe-t-il ?

Rien apparaît !

Étape 2 : Création du sol et du personnage

Avec Pyxel pour dessiner un rectangle, on utilise l'instruction :

```
pyxel.rect(x, y, longueur, largeur, couleur)
```

- x et y sont les coordonnées du sommet en haut à gauche du rectangle ;
- longueur et largeur sont les dimensions du rectangle ;
- couleur est sa couleur de remplissage (*cf.* palette de couleur).

Étape 2 : Création du sol et du personnage

Avec Pyxel pour dessiner un rectangle, on utilise l'instruction :

```
pyxel.rect(x, y, longueur, largeur, couleur)
```

- x et y sont les coordonnées du sommet en haut à gauche du rectangle ;
- longueur et largeur sont les dimensions du rectangle ;
- couleur est sa couleur de remplissage (cf. palette de couleur).

On code

En utilisant les variables `player_x` et `player_y`, compléter le code de la fonction `draw()` pour dessiner le carré.

Étape 2 : Création du sol et du personnage

Avec Pyxel pour dessiner un rectangle, on utilise l'instruction :

```
pyxel.rect(x, y, longueur, largeur, couleur)
```

- x et y sont les coordonnées du sommet en haut à gauche du rectangle ;
- longueur et largeur sont les dimensions du rectangle ;
- couleur est sa couleur de remplissage (cf. palette de couleur).

On code

En utilisant les variables `player_x` et `player_y`, compléter le code de la fonction `draw()` pour dessiner le carré.

Ajouter dans le code de la fonction `draw`, la ligne suivante pour dessiner le sol :

```
pyxel.rect(GROUND[0], GROUND[1], GROUND[2], GROUND[3], 13)
```

Étape 2 : Création du sol et du personnage

Avec Pyxel pour dessiner un rectangle, on utilise l'instruction :

```
pyxel.rect(x, y, longueur, largeur, couleur)
```

- x et y sont les coordonnées du sommet en haut à gauche du rectangle ;
- longueur et largeur sont les dimensions du rectangle ;
- couleur est sa couleur de remplissage (cf. palette de couleur).

On code

En utilisant les variables `player_x` et `player_y`, compléter le code de la fonction `draw()` pour dessiner le carré.

Ajouter dans le code de la fonction `draw`, la ligne suivante pour dessiner le sol :

```
pyxel.rect(GROUND[0], GROUND[1], GROUND[2], GROUND[3], 13)
```

Exécuter votre programme. Que se passe-t-il ?

Étape 2 : Création du sol et du personnage

Avec Pyxel pour dessiner un rectangle, on utilise l'instruction :

```
pyxel.rect(x, y, longueur, largeur, couleur)
```

- x et y sont les coordonnées du sommet en haut à gauche du rectangle ;
- longueur et largeur sont les dimensions du rectangle ;
- couleur est sa couleur de remplissage (cf. palette de couleur).

On code

En utilisant les variables `player_x` et `player_y`, compléter le code de la fonction `draw()` pour dessiner le carré.

Ajouter dans le code de la fonction `draw`, la ligne suivante pour dessiner le sol :

```
pyxel.rect(GROUND[0], GROUND[1], GROUND[2], GROUND[3], 13)
```

Exécuter votre programme. Que se passe-t-il ?

Tout est dessiné.

Étape 3 : Déplacement horizontal du personnage

En route vers l'étape 3

Sur le bureau de votre ordinateur :

- ouvrir le dossier `activite_mario`;
- double-cliquer sur le fichier `etape3.py`.

Étape 3 : Déplacement horizontal du personnage

Il s'agit ici de modifier la fonction `update()` qui est appelé 30 fois par seconde.

Étape 3 : Déplacement horizontal du personnage

Il s'agit ici de modifier la fonction `update()` qui est appelé 30 fois par seconde.

On code

Remplacer l'instruction `pass` de la fonction `update()` par :

```
global player_x, player_y,  
player_x, player_y = deplacement(player_x, player_y)
```

Il faut maintenant compléter la fonction `deplacement()`.

Étape 3 : Déplacement horizontal du personnage

Il s'agit ici de modifier la fonction `update()` qui est appelé 30 fois par seconde.

On code

Remplacer l'instruction `pass` de la fonction `update()` par :

```
global player_x, player_y,  
player_x, player_y = deplacement(player_x, player_y)
```

Il faut maintenant compléter la fonction `deplacement()`.

Étape 3 : Déplacement horizontal du personnage

En Pyxel, pour détecter l'appuie sur une touche, on utilise l'instruction :

```
pyxel.btn(pyxel.KEY_RIGHT) # pour la touche "flèche droite"
```

Étape 3 : Déplacement horizontal du personnage

En Pyxel, pour détecter l'appuie sur une touche, on utilise l'instruction :

```
pyxel.btn(pyxel.KEY_RIGHT) # pour la touche "flèche droite"
```

ou encore : KEY_LEFT ; KEY_UP ; KEY_DOWN ; KEY_SPACE ; ...

Étape 3 : Déplacement horizontal du personnage

En Pyxel, pour détecter l'appuie sur une touche, on utilise l'instruction :

```
pyxel.btn(pyxel.KEY_RIGHT) # pour la touche "flèche droite"
```

ou encore : KEY_LEFT ; KEY_UP ; KEY_DOWN ; KEY_SPACE ; ...

On code

Double-cliquer sur le fichier etape3_a.py.

- Compléter les pointillés dans la fonction `deplacement()`.
Attention : le personnage ne doit pas sortir de la fenêtre !
- Exécuter votre programme. Que se passe-t-il ?

Étape 3 : Déplacement horizontal du personnage

En Pyxel, pour détecter l'appuie sur une touche, on utilise l'instruction :

```
pyxel.btn(pyxel.KEY_RIGHT) # pour la touche "flèche droite"
```

ou encore : KEY_LEFT ; KEY_UP ; KEY_DOWN ; KEY_SPACE ; ...

On code

Double-cliquer sur le fichier etape3_a.py.

- Compléter les pointillés dans la fonction `deplacement()`.

Attention : le personnage ne doit pas sortir de la fenêtre !

- Exécuter votre programme. Que se passe-t-il ?

Le personnage se déplace avec les touches directionnelles.

Étape 4 : Gestion de la gravité

En route vers l'étape 4

Sur le bureau de votre ordinateur :

- ouvrir le dossier `activite_mario` ;
- double-cliquer sur le fichier `etape4.py`.

On code

Compléter la partie « Gestion du saut » de la fonction `deplacement()`, en programmant l'algorithme suivant :

```
si la barre espace est pressée :  
    la variable vy prend la valeur -7
```

Étape 5 : Plateformes et sortie

En route vers l'étape 5

Sur le bureau de votre ordinateur :

- ouvrir le dossier `activite_mario` ;
- double-cliquer sur le fichier `etape5.py`.

Il s'agit ici de rajouter quatre plateformes et une porte de sortie.

Étape 5 : Plateformes et sortie

Pour faciliter le traitement des plateformes, nous allons créer une liste de plateformes.

On code

- La variable `GROUND` a été remplacé par la variable de type liste `PLATFORMS`. Compléter la boucle `for` de la fonction `draw()` afin de dessiner les plateformes.
- La variable `EXIT` contient les coordonnées de la porte de sortie. Compléter la ligne permettant de dessiner la porte de sortie en marron.

Étape 6 : Gestion des collisions

En route vers l'étape 6

Sur le bureau de votre ordinateur :

- ouvrir le dossier `activite_mario` ;
- double-cliquer sur le fichier `etape6.py`.

Il s'agit de la partie la plus technique du projet qui peut être passée dans une première lecture.

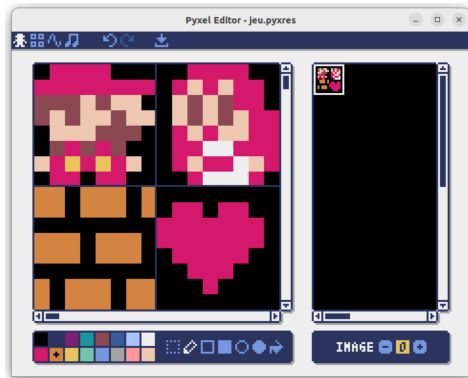
On code

Le code est documenté, compléter les pointillés de sorte que le personnage puisse sauter sur les plateformes sans les traverser.

Le jeu est fini lorsque le personnage touche la sortie.

Étape 7 : Créer un scénario

Le fichier `jeu.pyxres` contient des objets créer pixel par pixel, ils vont nous servir à créer un scénario.



Étape 7 : Créer un scénario

En route vers l'étape 7

Sur le bureau de votre ordinateur :

- ouvrir le dossier `activite_mario` ;
- double-cliquer sur le fichier `etape7.py`.

On code

Juste après l'instruction du titre, ajouter : `pyxel.load("jeu.pyxres")`
Cela permet de charger le jeu d'images.

Étape 7 : Créer un scénario

Pour placer une image dans la fenêtre, il faut copier une partie de la banque d'images vers une position (x, y) dans la fenêtre :

```
pyxel.blit(x, y, numéro banque, x(image), y(image),  
           taille horiz, taille vert, [couleur de transparence])
```

On code

- Remplacer le carré représentant la sortie par :
`pyxel.blit(EXIT[0], EXIT[1], 0, 8, 0, 8, 8, 0)`
- Que se passe-t-il ? Oh ! La belle princesse !
- À vous de remplacer le joueur par le bonhomme, de modifier les plateformes avec « les briques », ...

THE END

A bientôt en NSI