Introduction au développement de jeu vidéo

GDE-101 - 18.04.2023

La Programmation

Ne fuyez pas, pauvres fous.

SPOILER

Soit en ligne :

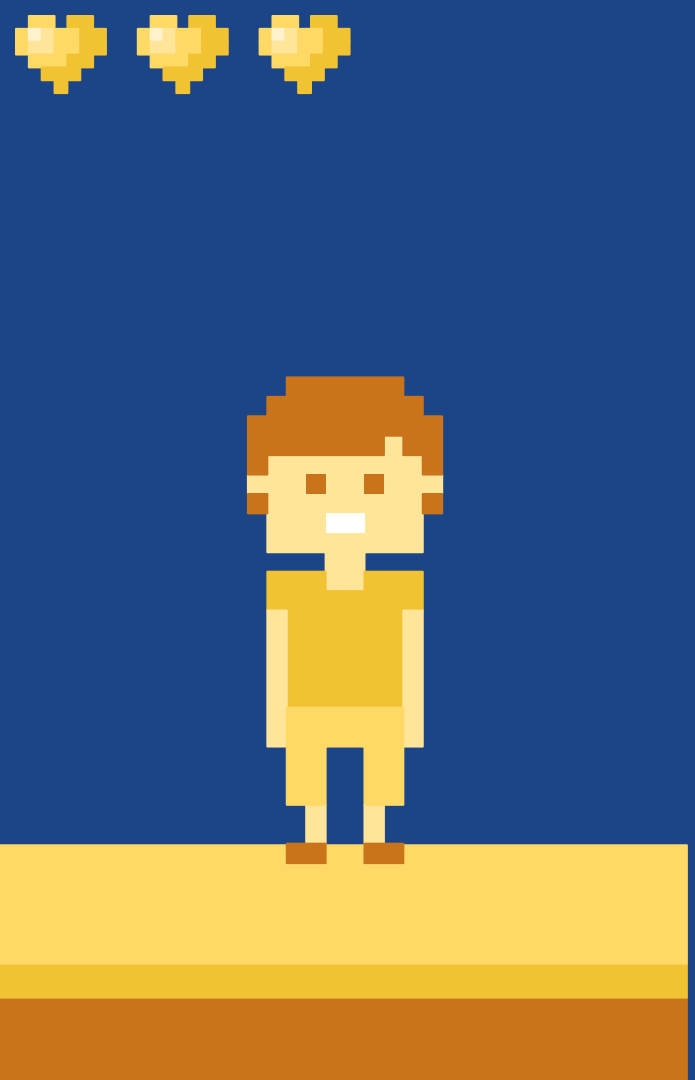
https://repl.it/languages/lua

- Facile et rapide.
- On vous recommande ceci pour cette semaine de cours.

Soit en téléchargeant :

https://studio.zerobrane.com/

- Demande un peu plus de maîtrise.
- Permet de faire un peu plus de choses.



Plan

Introduction

Structures de contrôle

Le Lua

Programmation en direct

Exercice

Introduction

Dans un trou vivait un programme

Principe

Input(s) (entrées)

- Nombres
- Tableau
- Phrases
- …

Programme

Output(s) (sortie(s))
Résultat

Définitions

- **Langage de programmation** : notation conventionnelle pour décrire un programme
- **Variable** : Un nom associé à une valeur
- **Constante** : La même chose, mais qui est fixe
- **Type de donnée (“data type”)** :
  - Booléen (“bool”) : “Vrai” ou “Faux”
  - Nombre (“int”) : Un nombre entier
  - Nombre (“float”) : Un nombre réel
  - Chaîne de caractères (“string”) : Du texte

La programmation, c’est appliquer des éléments les uns après les autres.

On respire

Structures de contrôle

Trois pour les gouverner toutes

Les explications suivantes s’appliquent à tous les langages de programmation.

Trois structures principales

La séquence

“D’abord je bois la potion, ensuite je gagne de la vie, ensuite je n’ai plus la potion”

La condition

“Si j’ai une potion, alors je peux me soigner, sinon, je ne peux pas me soigner”

La boucle

“Tant que je n’ai pas regagné ma vie, je bois la potion”

Le Lua

On dirait de l’elfique

C’est quoi ?

- Langage de script.
- Langage embarqué.
  - Particulièrement apprécié pour l'embarqué, le développement réseau et les jeux vidéo.
- Utilisé dans de nombreux jeux !
  - Transformice, World of Warcraft, Roblox, Garry's Mod, ...
- Utilisable pour PICO-8, mais aussi CryENGINE et LÖVE.
  - Futurs cours ?
- Facile à prendre en main, facile à comprendre et à utiliser.

Pour PICO-8, le Lua utilisé est “allégé”.
Pour apprendre le Lua, on va essayer de s’y tenir.

Programmation en direct

Alors en quoi consiste cette mission… Quête… Chose ?

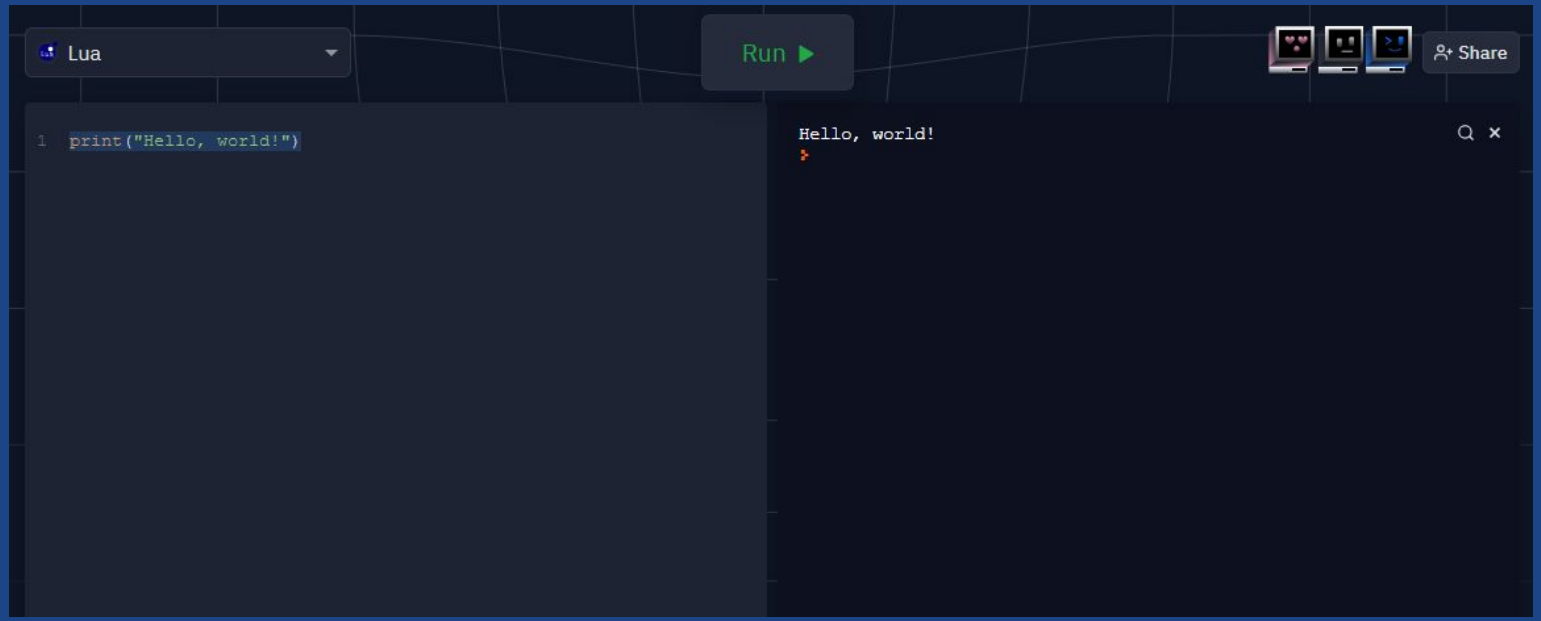
https://repl.it/languages/lua

Essayez de faire des essais en même temps que les slides ! Ouvrez le site Internet et suivez !

ZeroBrane a un chouette tutoriel intégré si vous voulez le suivre !

La tradition

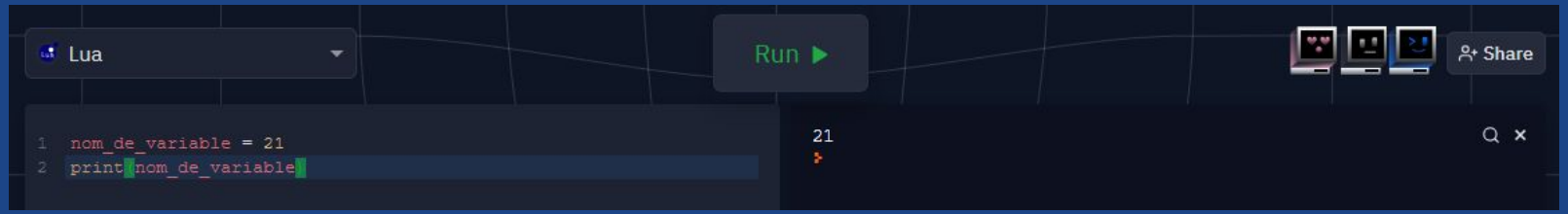
- Tapez `print("Hello, world!")`.



- Si ça vous écrit quelque chose dans la partie de droite, bravo !

Les variables


```
age = 21
```



Quelques opérations...

- **Négation** : `-a`
- **Addition** : `a + b`
- **Soustraction** : `a - b`
- **Multiplication** : `a * b`
- **Division** : `a / b`
- **Division euclidienne (entière)** : `a // b`
  - Exemple : `5 // 2 == 2`
- **Reste de la division entière (modulo)** : `a % b`
  - Exemple : `5 % 2 == 1`
- **Exponentiation** : `a ^ b`

Exercice !

Créer un programme, avec une variable qui contient un nombre entre 2 et 10.

Multiplier dans une nouvelle variable ce nombre par 2, et ajouter le chiffre 5.

Enfin, affichez le résultat avec print

Condition

Simple “Si”

```
if condition then
    something
end
```

“Si” / “Sinon”

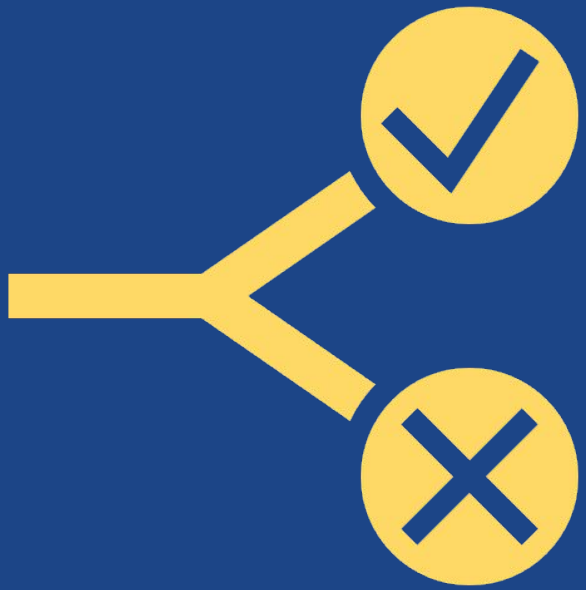
```
if condition then
    something
else
    something_else
end
```

“Si” / “Sinon si”

```
if cond1 then
    something
elseif cond2 then
    something_else
end
```

Les opérateurs de relation

- En gros, les trucs pour comparer.
  - a strictement plus petit que b : `a < b`
  - a strictement plus grand que b : `a > b`
  - a plus petit ou égal à b : `a <= b`
  - L'autre sens : `a >= b`
  - a strictement égal à b : `a == b`
  - a strictement pas égal à b : `a ~= b`



Les opérateurs logiques !

- **Opérateurs**
  - **Et** : `a and b`
  - **Ou** : `a or b`
  - **Non (le contraire de)** : `not a`
- **Calcul booléen**
  - pour un “et”, le premier “faux” pousse le résultat à “faux”
  - pour un “ou”, le premier “vrai” pousse le résultat à “vrai”

Table de la loi **ET**		
b\a	**0**	**1**
0	0	0
1	0	1

Table de la loi **OU**		
b\a	**0**	**1**
0	0	1
1	1	1

J’ai perdu tout le monde ?

Exercice !

Petit programme, qui prend en entrée votre âge :

```
variable = tonumber(io.read())
```

Puis qui indique si vous êtes majeur·e à Madagascar (21 ans), sinon en Algérie (19 ans), sinon en Suisse (18 ans), sinon répond que vous êtes mineur·e.

Vous pouvez aussi utiliser `elseif` !

Boucles

While

```
x = 0
while x < 5 do
    print(x)
    x += 1
end
```

Repeat

```
x = 0
repeat
    print(x)
    x += 1
until x > 4
```

For

```
for i = 0, 4, 1 do
    print(i)
end
```

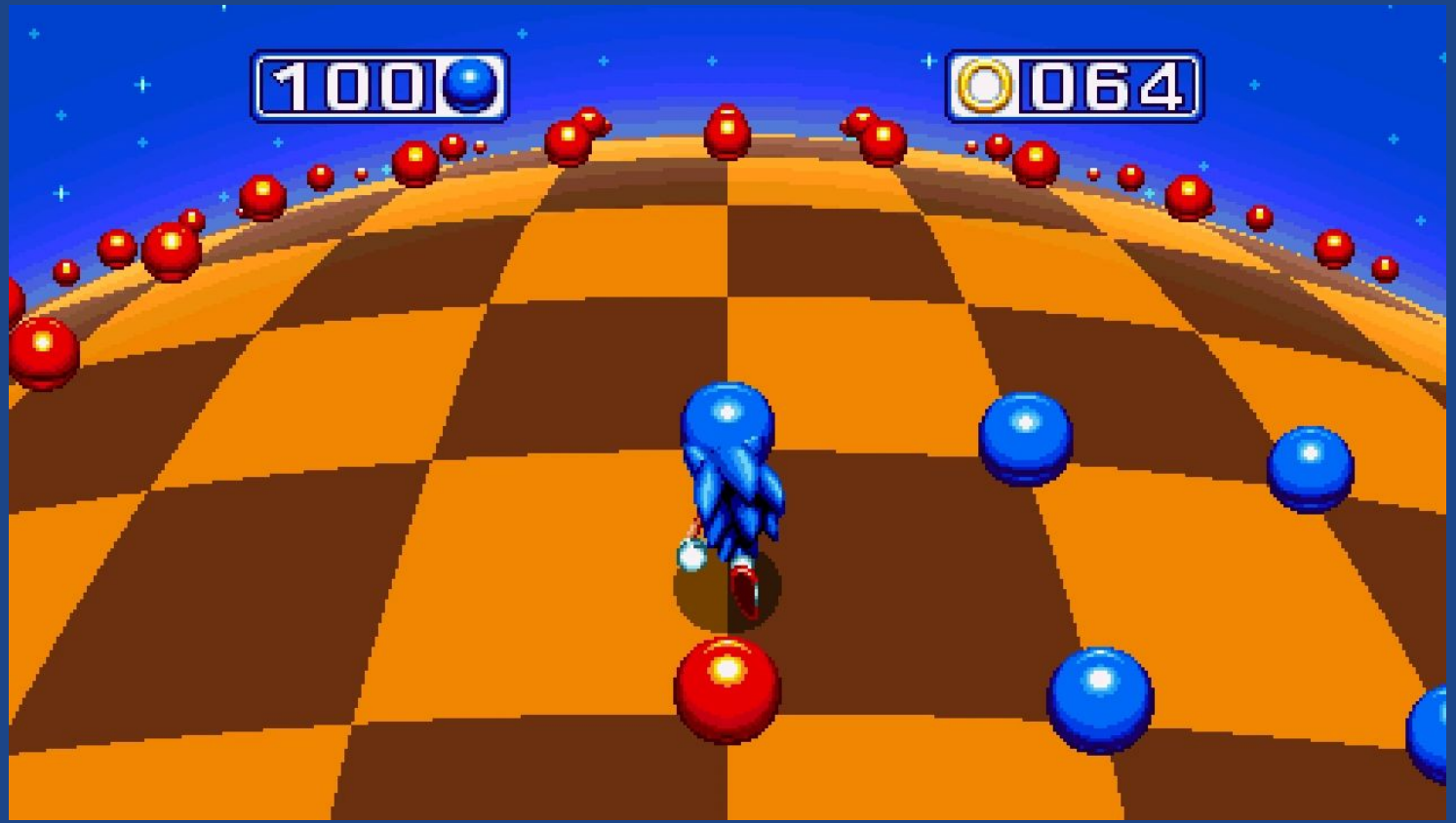
Exercice !

Une boucle, qui demande à l’utilisateur de taper le chiffre 81, jusqu’à ce que le chiffre entré soit effectivement 81.

Fonctions

```
function multiplier(x1, y1)
  return x1 * y1
end
```

Bonus !



Les chaînes de caractères

- Également appelées `string` en anglais.
- Comme une variable normale, avec des guillemets :
  `texte = “Des trucs là-dedans.”`
- Pour fusionner du texte, il faut utiliser deux points : `..`
  - Exemple : `print(“Voici ”..” un “..” texte”)`
- On peut calculer la longueur d’un texte avec `#`.

Des tableaux pour s’organiser

- Un tableau (table), c’est une variable qui contient d’autres variables (ou constantes).
  - Utilisation : `tableau = {1, 2, 3}`.
- Pourquoi c’est utile ? Car on peut nommer les variables !
  - Exemple :

    ```
    joueur = {}
    joueur.vies = 5
    print(joueur.vies)
    ```
  - On peut aussi utiliser `joueur["vie"]`.
- On utilise # pour avoir la taille du tableau.

Parcourir les tableaux

- Un tableau peut être parcouru avec… Une boucle !
  - Exemple :

```
for k,v in pairs(x) do
  print(k.." = "..v)
end
```

- Le tableau peut également être utilisé pour des séquences !
  - Très pratique pour parcourir un inventaire, par exemple.

Des questions ?

discord.gg/8tVCZJG

Par écrit : canal #en-direct

Par oral : lever la main (:

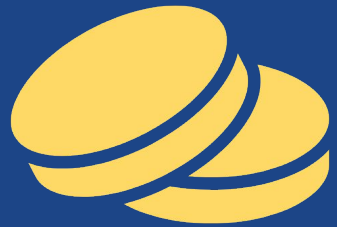
Programmation - Exercice

Les bonnes bases


PICO-8
C’est cool.

L’un des trois

Snake

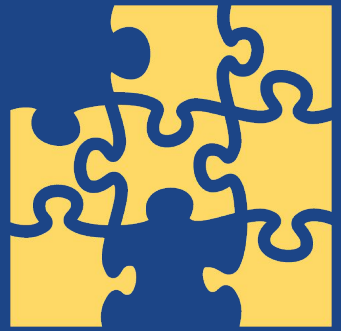


Puissance 4

Memory

Les deux

Remplir les fonctions à trous



Étendre le jeu existant

Snake

Un classique !

Les règles de base

- Le·a joueur·euse contrôle, à l’aide des 4 flèches directionnelles (touches), une ligne ou un “serpent”, qui s’allonge au fur et à mesure.
- Le but du jeu est de faire manger au serpent la pomme, sans quitter les bords de la grille et sans heurter les obstacles (dans la version du jeu qu’on vous propose, le seul obstacle est le serpent lui-même).
- Il n’y a que 4 touches à utiliser ! Tu fais bouger le serpent à l’aide des flèches du clavier, dans les limites de la grille. Si tu heurtes les bords, ou si la tête de ton serpent touche un bout de son corps : c’est perdu !
- Pas de conditions de victoire

Fonctions à trous

- Fonction “Flux” : changer les coordonnées de la tête du serpent avant de la déplacer (regardez les coordonnées “previous” !)

- Fonction “Loop” : lorsque le·a joueur·euse perd, une petite animation est mise en place : le corps du serpent est supprimé graduellement à commencer par la queue, et des particules sont éparpillées au moment où la partie du corps est supprimée. Il faut compléter la partie du code qui gère cela

- Fonction “If” : le but de la fonction est d’expliciter les commandes afin de déplacer le serpent uniquement dans les limites de la grille du jeu.

Modifications possibles

- Les particules qui augmentent de rayon de dispersion selon la longueur du serpent
- La vitesse du serpent qui augmente toutes les 5/10/20 pommes
- Des fruits avec des effets spécifiques quand ils sont mangés
- …

Puissance 4

Manipulation de jetons

Les règles de base

- Un·e joueur·euse (‘X’, couleur bleue) affronte un·e autre joueur·euse (‘O’, couleur rouge) dans un tableau (taille 8x8).
- À tour de rôle, chaque joueur·euse dépose un pion (taille 1x1) dans une des 8 colonnes. Le pion tombe tout en bas de la colonne et les pions se superposent s’il y en a plusieurs dans une même colonne.
- Pour gagner, il faut qu’un·e joueur·euse parvienne à créer une ligne (horizontale, verticale, diagonale) de longueur 4 avec son symbole/sa couleur.
- Dans cette version du jeu, le·a joueur·eyse ‘X’ affronte une IA ‘O’ qui joue ses coups de manière aléatoire.

Fonctions à trous

- Fonction “Flux” : implémentez la fonction printhand() qui affiche une main au-dessus de la colonne choisie par le·a joueur·euse ‘X’ (regardez `spr(num. sprite, x, y)` et `selectedcolumn`)

- Fonction “Loop” : dans `winner(board)`, on vérifie si un·e joueur·euse est parvenu·e à faire une ligne de longueur 4. Implémentez la vérification de la diagonale de la forme en vous aidant du reste.

- Fonction “If” : implémentez, à l’aide d’une boucle for, la fonction `iscolumnfull(board, col)` qui vérifie si la colonne numéro `col` du tableau de jeu `board.tiles` est pleine ou pas.

Modifications possibles

- Compter le nombre de victoires/défaites et l'afficher.
- Modifier la taille du tableau de jeu et/ou la longueur de la ligne à effectuer. On pourra par exemple faire un Puissance 5 sur un tableau 10x10.
- Remplacer l'IA par un deuxième joueur.
- Variante Pop-Out. A chaque tour, le·a joueur·eyse 'X' possède deux choix : il peut soit déposer un pion dans une colonne, soit supprimer un de ses propres pions qui se trouverait tout en bas d'une colonne (ça revient à enlever ce pion et décaler toute la colonne). Par simplicité, pas besoin de donner cette possibilité à l'IA.
- Modifier simplement l'IA du jeu.
- …

Memory

Il ne faudrait pas l’oublier

Les règles de base

- Le jeu commence avec seize cartes, donc huit paires de cartes, que l’on dispose, face cachée, sous forme de carré, sur le plateau.
- Le·a joueur·euse va retourner une première carte, puis une deuxième carte. Si les deux cartes ont la même face, le·la joueur·euse gagne un point. Si les deux cartes ont des faces différentes, il faut de nouveau les retourner face cachée.
- La condition de victoire est de trouver toutes les paires le plus vite possible.
- Il n’y a pas de condition de défaite.

Fonctions à trous

- Fonction “Flux” : Il faut retourner au state wait, puis retourner les cartes cote verso.

- Fonction “Loop” : Il faut afficher les cartes qui sont dans le tableau avec le bon côté (recto ou verso).

- Fonction “If” : Il faut ajouter les trois autres boutons pour déplacer le curseur.

Modifications possibles

- Changer les constantes (e.g., taille de la grille)
- Trouver des trios plutôt que des pairs
- Tu ne peux retourner toutes les cartes qu’une seule fois
- …

Des questions ?

discord.gg/8tVCZJG

Par écrit : canal #en-direct

Par oral : lever la main (:

Merci de votre attention !

discord.gg/8tVCZJG

Par écrit : canal #en-direct

Par oral : lever la main (: