#  Programmation - Projet 3

## Introduction

Après avoir travaillé sur notre game design document et avoir les nombreux détails de notre jeu en tête, il est temps de passer à la programmation ! Cependant, pour commencer, on va vérifier que les acquis de base sont, et bien, acquis.

Durant ce projet, vous allez réaliser un jeu textuel simple, mettant en pratique les trois aspects de base de la programmation, à savoir la séquence, la boucle et la condition. Ces jeux pourront par la suite être augmentés avec des fonctionnalités textuelles, et, pourquoi pas, lors de notre session pixel art, des graphismes.

Veuillez noter que l’on vous donne des solutions plus bas. Ces solutions sont très, très simples, et il est bien plus intéressant pour vous de tenter de coder par vous même !

## Objectifs

* Prendre en main les bases de la programmation.
* Créer un jeu textuel simple avec retour utilisateur.
* Implémenter de nouvelles idées pour un jeu simple.

## Instructions

### Généralités

Si vous avez la flemme de télécharger des choses sur votre ordinateur, vous pouvez faire les exercices en ligne, par exemple sur [Lua Online Compiler & Interpreter](https://repl.it/languages/lua).

Nous allons tenter, dans la mesure du possible, d’utiliser du code “spécial PICO-8”, mais quelques exceptions vont apparaître pour ce projet. Si vous avez besoin de documentation, c’est ici : [Lua cheatsheet](https://devhints.io/lua).

Par ailleurs, voici quelques fonctions utiles pour les exercices.

#### Sélectionner un nombre au hasard

| -- Magie noire à ignorer !math.randomseed(os.time())-- On sélectionne un nombre entre 1 et 100number = math.random(1, 100) |
| --- |

#### Lire une chaîne de caractères du joueur

| reponse = io.read() |
| --- |

#### Convertir une chaîne de caractères en nombre

| nombre = tonumber(pas\_nombre) |
| --- |

#### Sélectionner une seule lettre d’une chaîne de caractères

| mot = "fleur"-- On prend la troisième lettre du motlettre = string.sub(mot, 3, 3) |
| --- |

#### Convertir un tableau en chaîne de caractères

| -- Convertir un tableau en chaîne de caractèresfunction gostring(thing) ret = "" for i=1, #thing do ret = ret..thing[i] end return retendmon\_tableau = {"s", "a", "l", "u", "t"}mot = gostring(mon\_tableau) |
| --- |

#### L’inverse

| -- Convertir une chaîne de caractères en tableaufunction gotable(thing) ret = {} for i=1,#thing do ret[i] = string.sub(thing, i, i) end return retendmon\_mot = "loutre"tableau = gotable(mon\_mot) |
| --- |

### Plus ou moins

L’ordinateur sélectionne un nombre au hasard entre 1 et 100, et le·la joueur·euse doit trouver le bon nombre le plus vite possible. Technique cool : la recherche dichotomique !

#### Comment ?

1. Le·la joueur propose un nombre.
2. L’ordinateur répond si c’est strictement plus ou si c’est strictement moins.
3. Et on boucle jusqu’à trouver le bon nombre.

### Pendu

L’ordinateur choisit un mot au hasard, et le·la joueur·euse doit trouver le mot en 10 (ou 11) coups. Technique cool : l’analyse par fréquence !

#### Comment ?

1. Le·la joueur propose une lettre.
2. L’ordinateur montre les lettres du mot si le mot contient la lettre, ajoute une barre au pendu sinon.
3. Et on boucle jusqu’à trouver le mot ou bien ne plus avoir de coups disponibles.

## A rendre (et/ou)

* Le code, fonctionnel, d’un Plus ou moins.
* Le code, fonctionnel, d’un Pendu.

## Mettre en commun !

Si vous le souhaitez, vous pouvez sans souci partager vos codes sur le Discord du cours ou sur le Forum Moodle, pour que d’autres étudiant·e·s puissent faire des commentaires - il est même encouragé de faire des retours et d’échanger !

# Solutions proposées

## Plus ou moins

| -- Magie noire à ignorer !math.randomseed(os.time())-- On sélectionne un nombre entre 1 et 100number = math.random(1, 100)-- On définie un tableau "player"player = {}--[[ Dans ce tableau, on indique que la valeur "guess" commence à -1--]]player.guess = -1-- On veut aussi compter le nombre d'essais du joueurplayer.tries = 0print("Devine un nombre entre ".."1".." et ".."100".." !")-- Tant que le joueur n'a pas le bon nombrewhile player.guess ~= number do -- On compte le nombre d'essais ici player.tries = player.tries + 1 -- Magie noire pour lire la réponse ! player.answer = io.read() -- On convertit la réponse en nombre entier player.guess = tonumber(player.answer) -- Si le joueur a donné un nombre trop haut if player.guess > number then print("Plus bas !") elseif player.guess < number then print("Plus haut !") else print("Bien joué ! Le nombre était bien "..number..".") print("Tu as mis "..player.tries.." essais !") endend |
| --- |

## Pendu

| -- Convertir un tableau en chaîne de caractèresfunction gostring(thing) ret = "" for i=1, #thing do ret = ret..thing[i] end return retend -- Magie noire à ignorer !math.randomseed(os.time())words = { "tableau", "peinture", "art", "chemise", "toilettes", "ecran", "pedoncule"}--[[ On sélectionne un nombre entre 1 et et la taille du tableau des mots--]]choice = math.random(1, #words)-- Le mot est choisiword = words[choice]-- On définie un tableau "player"player = {}-- On veut compter le nombre d'essais restants au joueurplayer.tries = 10-- Enfin, on prépare le mot camouflé à deviner !player.answer = {}for i=1, #word do player.answer[i] = "?"endprint("== Devine le mot ! ==\n")print("Le mot actuel est '"..gostring(player.answer).."'.")-- On souhaite répéter l'action...repeat print("Quelle lettre ?") -- Magie noire pour lire la réponse ! letter = io.read() -- On veut voir si la réponse est correcte... success = false  -- Est-ce que la lettre est dans le mot ? for i = 1, #word do -- On prend une lettre du mot à deviner, et on compare local c = string.sub(word, i, i) -- Si la lettre est présente... if c == letter then -- Si au moins une lettre est juste, c'est un succès success = true -- Et on remplace dans la réponse du joueur player.answer[i] = c end end -- Si la bonne lettre a été trouvée, on indique le mot if success then print("Bravo !") print("Le mot actuel est '"..gostring(player.answer).."'.") -- Sinon, on enlève une vie else print("Pas de chance...") player.tries = player.tries - 1 end  print(player.tries.." essais restants !\n")--[[ ... Tant que le joueur a toujours des essais, ou bien que le mot n'a pas été trouvé.--]]until player.tries == 0 or gostring(player.answer) == wordif gostring(player.answer) == word then print("Félicitations !")else print("Tu as perdu !")endprint("Le mot était '"..word.."'.") |
| --- |

Hey ! N’oubliez pas qu’il est très facile d’améliorer les exemples précédents ! N’hésitez pas à tenter l’aventure ;p.