

Synthèse semaine 1 :

Procédure d'installation de love2D et de Lua

→ Voir support de cours .

Les variables en Lua/Love2D

Les variables numériques :

Définition : une variable est un emplacement mémoire identifié par un nom qui contient une valeur, les variables sont de 3 types :

- Entières et flottantes : Ce sont simplement des nombres...

Ex :

Energie = 100

Longueur = 36.27 (attention c'est un point pas une virgule).

- Booléennes : Ces variables nous serviront pour les structures de contrôle que nous aborderons la semaine 2 ... Elles peuvent contenir que deux valeurs : « true » ou « false »

Ex :

GameOver = false

PlayerIsAlive = true

- Chaîne de caractères : Ces variables nous serviront pour afficher du texte à l'écran.

Ex :

NomJoueur = « Zoltan »

ClassePerso = « le Barbare »

Les opérations sur les variables :

Sur les variables numériques :

- Addition
 - $\text{NombreDeVie} = \text{NombreDeVie} + 1$
- Soustraction
 - $\text{NombreDeTir} = \text{NombreDeTir} - 1$
- Multiplication
 - $\text{Bonus Dégat} = \text{DégatMissile} * 1.1$
- Division
 - $\text{Sante} = \text{BonusVie} / 2$

Attention à la priorité de calcul : bien penser à appliquer PEMDAS

- 1) **P**arenthèse
- 2) **E**xposant
- 3) **M**ultiplication/**D**ivision
- 4) **A**ddition/**S**oustraction
- 5) Ensuite de gauche à droite...

Ex : $3 + 4 * 2 = 11$

$$\begin{aligned} &\rightarrow 3 + \underline{4 * 2} \\ &\rightarrow 3 + \underline{8} = 11 \end{aligned}$$

Si l'on veut donner une priorité à une opération on place des parenthèses

Ex : $(3 + 4) * 2 = 14$

$$\begin{aligned} &\rightarrow \underline{(3 + 4)} * 2 \\ &\rightarrow \underline{7} * 2 = 14 \end{aligned}$$

Sur les chaînes de caractères :

On peut « concaténer » 2 chaînes caractères, cela revient à les joindre

Ex :

String1 = « Zoltan »

String2 = « Le Barbare »

String3 = String1 .. « : » .. String2

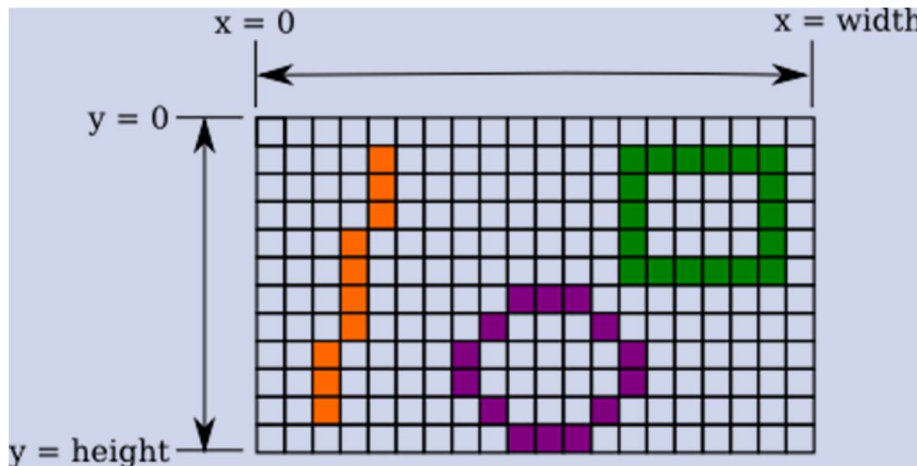
Ce qui donnera si l'on affiche la chaîne String3 à l'écran :

Zoltan Le Barbare

Position des textes à l'écran :

La taille de l'écran est définie par sa résolution (ex : 1024 x 768), ce qui signifie que l'écran peut afficher 1024 points par ligne sur 768 lignes, un « point » est appelé **PIXEL**. En anglais on parle de largeur à l'aide du terme **WIDTH** et la hauteur est exprimée par **HEIGHT**.

Le coin supérieur gauche de l'écran correspond aux coordonnées **0,0**. le zéro ROUGE est la valeur en X et le zéro ORANGE est la valeur en Y.



Intro à Love2D

Il y a 3 fonctions de base à utiliser dans lua :

1) love.load() :

Celle-ci va être utilisée pour charger toutes les variables nécessaires à notre jeu. Elle sera automatiquement lue 1x au démarrage du jeu.

2) love.update() :

Celle-ci sera utilisée pour mettre à jour tous les éléments du jeu. Elle est appelée environ 60x par seconde (les fameux 60 FPS que l'on retrouve dans le vocabulaire des jeux vidéo 😊)

3) love.draw() :

Celle-ci sera utilisée pour afficher tous les éléments du jeu. Elle est appelée également environ 60x par seconde.

Nous devons impérativement comprendre qu'un jeu vidéo est une boucle qui s'exécute à l'infini, tant que nous ne quittons pas le jeu ou tant que l'on éteint pas la console de jeux.

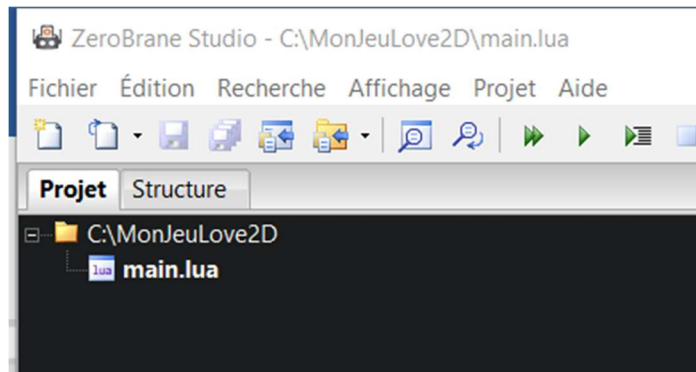
Programme love de base :

On créer un répertoire qui va contenir notre jeu

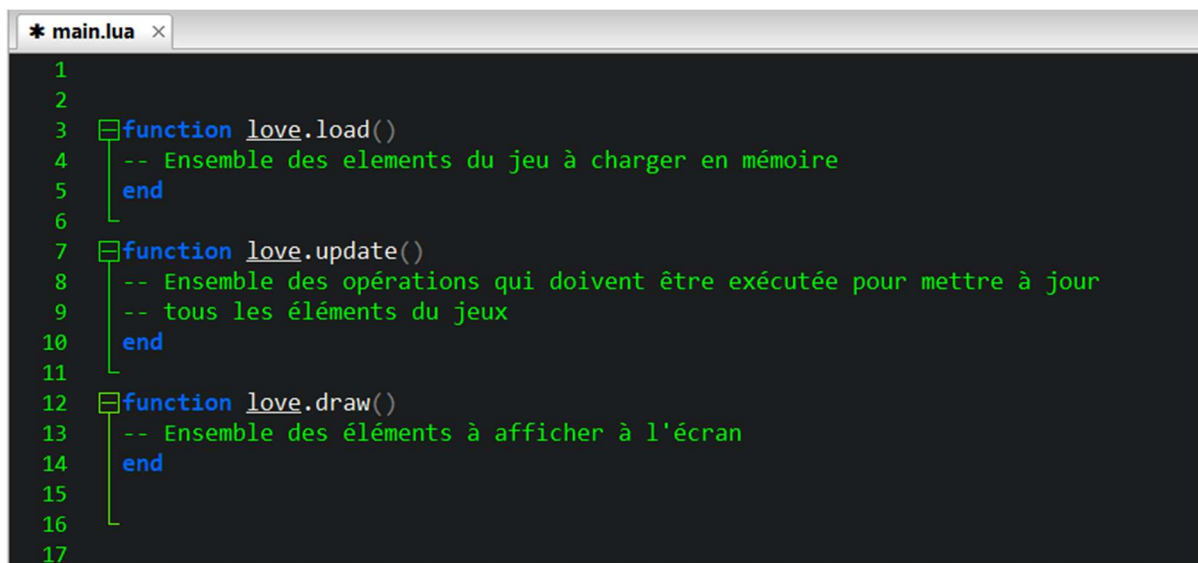
Ex : C:\MonJeuLove2D

On créer à l'intérieur de ce répertoire un fichier appelé « Main.lua ».

Ex : C:\MonJeuLove2D\Main.lua



Celui va contenir la structure de base suivante :



On remarque pour déclarer une fonction on utilise les mots **function** et **end** qui servent à délimiter le contenu de la fonction.

Instructions love2D abordées durant la séance :

Love.graphics.print(«texte » ou Variable, Position en X de l’affichage, Position en Y de l’affichage)

Ex :

```
1
2
3 function love.load()
4   -- Ensemble des elements du jeu à charger en mémoire
5   String1 = "Zoltan"
6   String2 = "Le Barbare"
7 end
8
9 function love.update()
10  -- Ensemble des opérations qui doivent être exécutée pour mettre à jour
11  -- tous les éléments du jeux
12 end
13
14 function love.draw()
15  -- Ensemble des éléments à afficher à l'écran
16  love.graphics.print(String1 .. " : " .. String2, 100, 80)
17 end
18
19
20
```

Donnera à l’exécution :

