**Stage de programmation Codeur Commando**



Table des matières

[Installation des logiciels nécessaires : 3](#_Toc128263526)

[Installation du moteur de jeu Love2D. 3](#_Toc128263527)

[Installation sur Windows 3](#_Toc128263528)

[Installation sur macOS 3](#_Toc128263529)

[Installation sur Linux 3](#_Toc128263530)

[Installation de Zerobrane Studio : 5](#_Toc128263531)

[Installation sur Windows 5](#_Toc128263532)

[Installation sur macOS 5](#_Toc128263533)

[Installation sur Linux 6](#_Toc128263534)

[Tutoriel de base pour l’apprentissage du Moteur Love2D 8](#_Toc128263535)

[Découvrir la structure d'un jeu Love2D 8](#_Toc128263536)

[Écrire notre premier programme 8](#_Toc128263537)

[Ajouter une image 9](#_Toc128263538)

[Ajouter un personnage 10](#_Toc128263539)

[Déplacer le personnage 12](#_Toc128263540)

[Ajouter un ennemi 14](#_Toc128263541)

[Ajouter un ennemi 15](#_Toc128263542)

[Ajouter un objet bonus 17](#_Toc128263543)

[Afficher un message de victoire 19](#_Toc128263544)

[Ajouter de la musique 21](#_Toc128263545)

[Ajouter des effets sonores 22](#_Toc128263546)

[Jouer des effets sonores 23](#_Toc128263547)

[Ajouter un score 24](#_Toc128263548)

[Ajouter un écran de fin de jeu 25](#_Toc128263549)

# Installation des logiciels nécessaires :

## Installation du moteur de jeu Love2D.

### Installation sur Windows

1. Allez sur le site officiel de LOVE2D : <https://love2d.org/>
2. Cliquez sur le bouton "Download" pour accéder à la page de téléchargement.
3. Cliquez sur le lien de téléchargement correspondant à votre version de Windows (32 bits ou 64 bits).
4. Enregistrez le fichier d'installation sur votre ordinateur.
5. Double-cliquez sur le fichier d'installation pour lancer l'assistant d'installation.
6. Suivez les instructions de l'assistant d'installation pour installer LOVE2D sur votre ordinateur.
7. Une fois l'installation terminée, vous pouvez utiliser LOVE2D en double-cliquant sur l'icône de l'application.

### Installation sur macOS

1. Allez sur le site officiel de LOVE2D : <https://love2d.org/>
2. Cliquez sur le bouton "Download" pour accéder à la page de téléchargement.
3. Cliquez sur le lien de téléchargement correspondant à votre version de macOS.
4. Enregistrez le fichier d'installation sur votre ordinateur.
5. Double-cliquez sur le fichier d'installation pour ouvrir l'image disque.
6. Faites glisser l'icône de l'application LOVE2D dans le dossier Applications.
7. Une fois l'installation terminée, vous pouvez utiliser LOVE2D en ouvrant le dossier Applications et en double-cliquant sur l'icône de l'application.

### Installation sur Linux

1. Ouvrez un terminal et tapez la commande suivante pour installer les dépendances requises :

**sudo apt-get install liblove0 liblove-dev love-doc**

1. Téléchargez le fichier d'installation de LOVE2D depuis le site officiel : <https://love2d.org/>
2. Décompressez le fichier d'installation en utilisant la commande suivante :

**unzip love-11.3-linux-x86\_64.zip**

1. Copiez l'application LOVE2D dans le dossier /usr/local/bin avec la commande suivante :

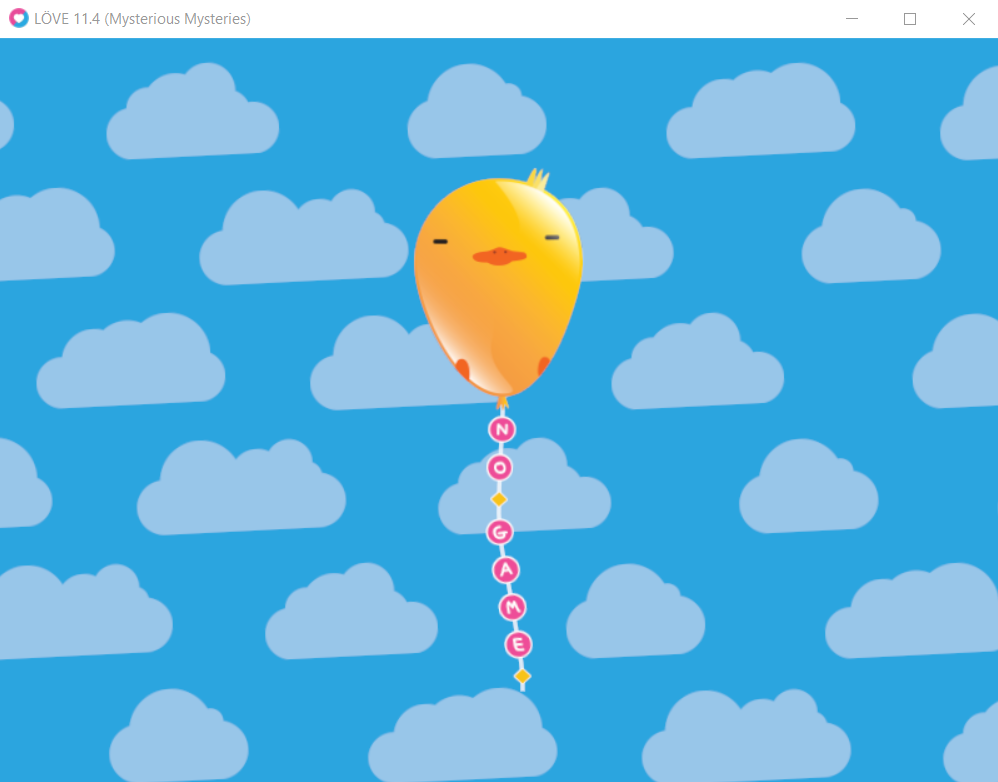
**sudo cp love-11.3-linux-x86\_64/love /usr/local/bin**

1. Vous pouvez maintenant utiliser LOVE2D en exécutant la commande suivante dans le terminal :

**love chemin/vers/votre/jeu**

1. Remplacez "chemin/vers/votre/jeu" par le chemin d'accès vers votre jeu LOVE2D.

Si l’installation s’est bien déroulée, lorsque vous exécutez la commande love2D, vous devriez obtenir ceci :



## Installation de Zerobrane Studio :

### Installation sur Windows

1. Allez sur le site officiel de Zerobrane Studio : https://studio.zerobrane.com/
2. Cliquez sur le bouton "Download" pour accéder à la page de téléchargement.
3. Cliquez sur le lien de téléchargement correspondant à votre version de Windows (32 bits ou 64 bits).
4. Enregistrez le fichier d'installation sur votre ordinateur.
5. Double-cliquez sur le fichier d'installation pour lancer l'assistant d'installation.
6. Suivez les instructions de l'assistant d'installation pour installer Zerobrane Studio sur votre ordinateur.
7. Une fois l'installation terminée, vous pouvez utiliser Zerobrane Studio en double-cliquant sur l'icône de l'application.

### Installation sur macOS

1. Allez sur le site officiel de Zerobrane Studio : https://studio.zerobrane.com/
2. Cliquez sur le bouton "Download" pour accéder à la page de téléchargement.
3. Cliquez sur le lien de téléchargement correspondant à votre version de macOS.
4. Enregistrez le fichier d'installation sur votre ordinateur.
5. Double-cliquez sur le fichier d'installation pour ouvrir l'image disque.
6. Faites glisser l'icône de l'application Zerobrane Studio dans le dossier Applications.
7. Une fois l'installation terminée, vous pouvez utiliser Zerobrane Studio en ouvrant le dossier Applications et en double-cliquant sur l'icône de l'application.

### Installation sur Linux

1. Ouvrez un terminal et tapez la commande suivante pour télécharger le fichier d'installation :

**wget** [**https://download.zerobrane.com/ZeroBraneStudioEduPack-1.90-linux.sh**](https://download.zerobrane.com/ZeroBraneStudioEduPack-1.90-linux.sh)

1. Rendez le fichier d'installation exécutable en utilisant la commande suivante :

**chmod +x ZeroBraneStudioEduPack-1.90-linux.sh**

1. Exécutez le fichier d'installation en utilisant la commande suivante :

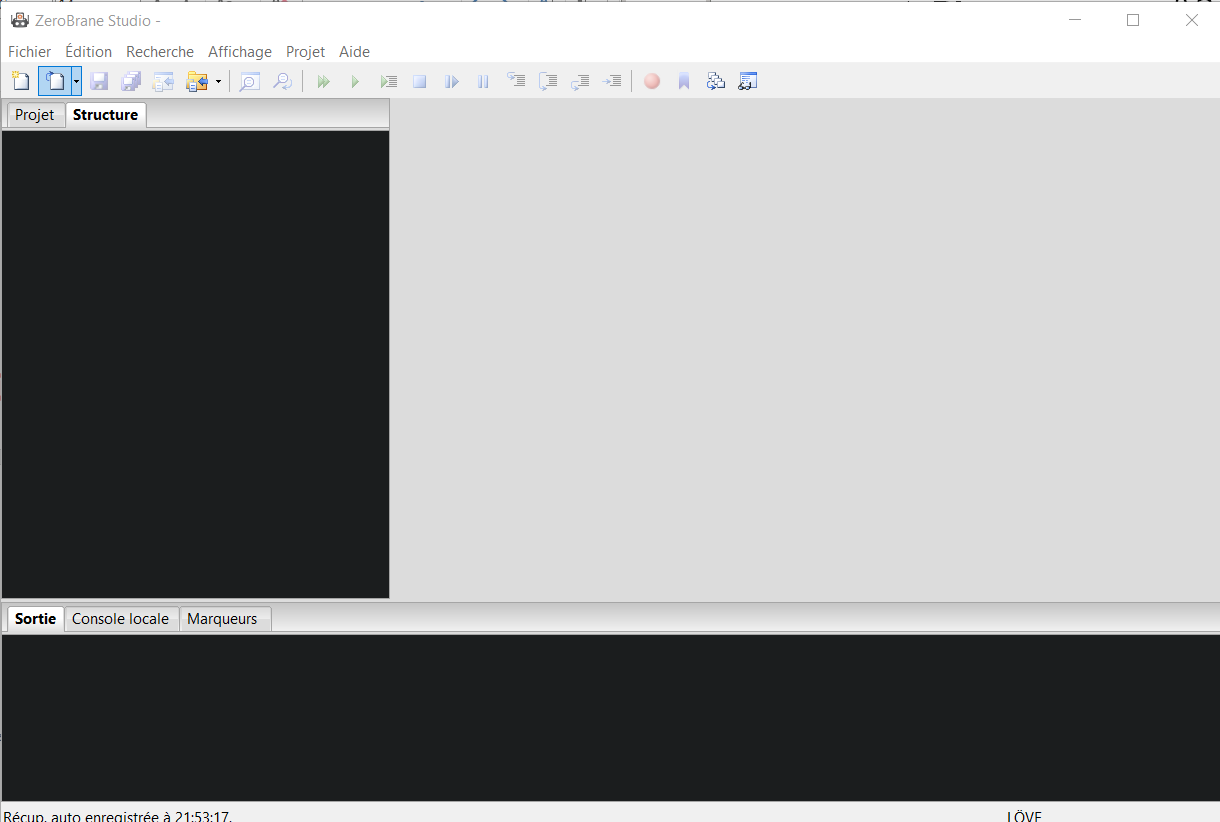
**./ZeroBraneStudioEduPack-1.90-linux.sh**

1. Suivez les instructions de l'assistant d'installation pour installer Zerobrane Studio sur votre ordinateur.
2. Une fois l'installation terminée, vous pouvez utiliser Zerobrane Studio en exécutant la commande suivante dans le terminal :

**zerobrane-studio**

Voilà, j'espère que ces instructions vous aideront à installer Zerobrane Studio sur votre ordinateur !

Si l’installation s’est bien déroulée, lorsque vous cliquez sur l’icône Zerobrane Studio, vous devriez obtenir quelque chose de similaire à ceci :



# Tutoriel de base pour l’apprentissage du Moteur Love2D

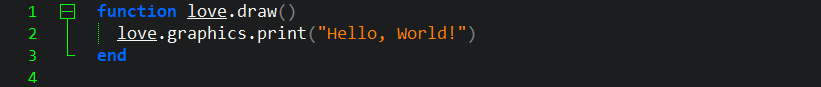
## Découvrir la structure d'un jeu Love2D

Avant de commencer à créer notre jeu, examinons la structure d'un jeu Love2D. Chaque jeu Love2D est constitué de deux parties principales :

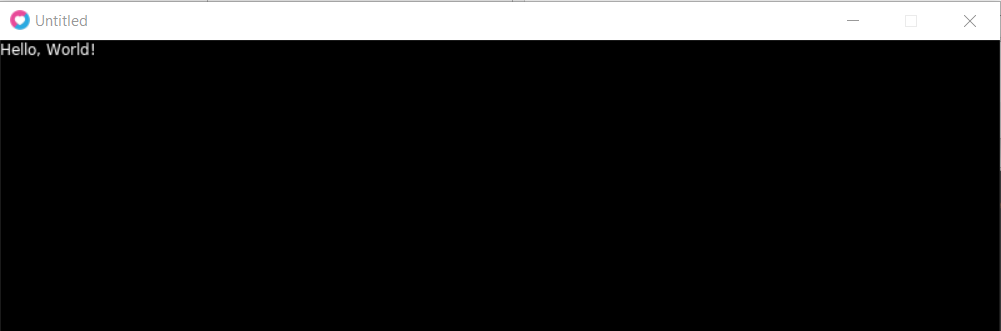
* Le fichier main.lua : c'est le point d'entrée de votre jeu. C'est ici que vous allez écrire le code qui va être exécuté lorsque le jeu démarre.
* Le dossier assets : c'est l'endroit où vous stockez tous les fichiers nécessaires à votre jeu, tels que les images, les sons, les polices, etc.

## Écrire notre premier programme

Créez un nouveau dossier pour votre jeu et créez un nouveau fichier main.lua à l'intérieur. Ouvrez ce fichier dans un éditeur de texte et entrez le code suivant :

****

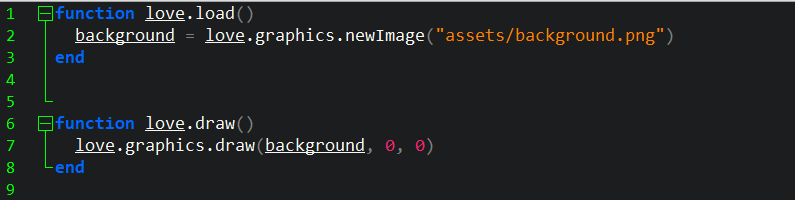
Sauvegardez le fichier et exécutez-le en double-cliquant sur l'icône Love2D. Si tout fonctionne correctement, vous devriez voir la chaîne de caractères "Hello, World!" s'afficher à l'écran.



## Ajouter une image

Créez un nouveau fichier dans le dossier assets appelé "background.png". C'est une image qui sera utilisée comme fond pour notre jeu. Téléchargez ou créez votre propre image pour cette étape.

Ajoutez le code suivant à votre fichier main.lua :

****

Dans cette étape, nous avons créé un objet Image à partir de notre fichier d'image et nous l'avons affiché à l'écran en utilisant la fonction love.graphics.draw.



L’image étant plus petite que l’affichage nous allons apprendre comment :

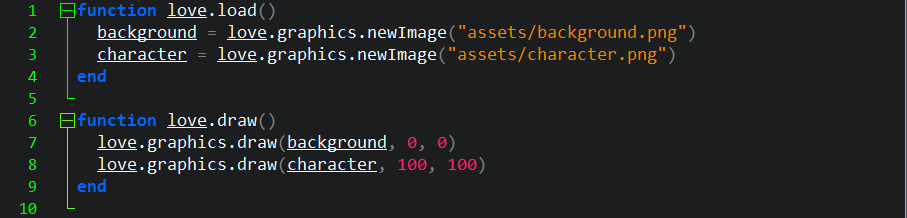
* Soit redimensionner notre fenêtre de jeu (Dimensionner la fenêtre ainsi que pleins d’autres options hyper intéressantes).
* Soit redimensionner notre image pour qu’elle s’adapte à notre fenêtre de jeu (Agrandir, Rétrécir ou déformer l’image).

## Ajouter un personnage

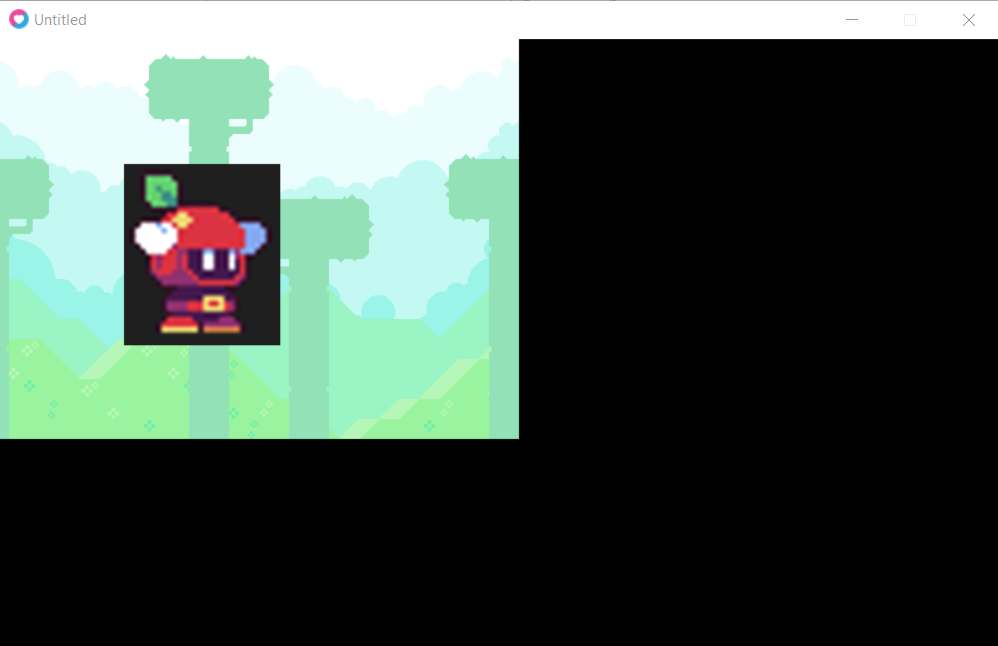
Maintenant que nous avons ajouté un arrière-plan, nous allons ajouter un personnage.

Ajoutez une nouvelle image dans le dossier assets appelée "character.png". C'est une image qui représente notre personnage. Téléchargez ou créez votre propre image pour cette étape.

Ajoutez le code suivant à votre fichier main.lua :

****

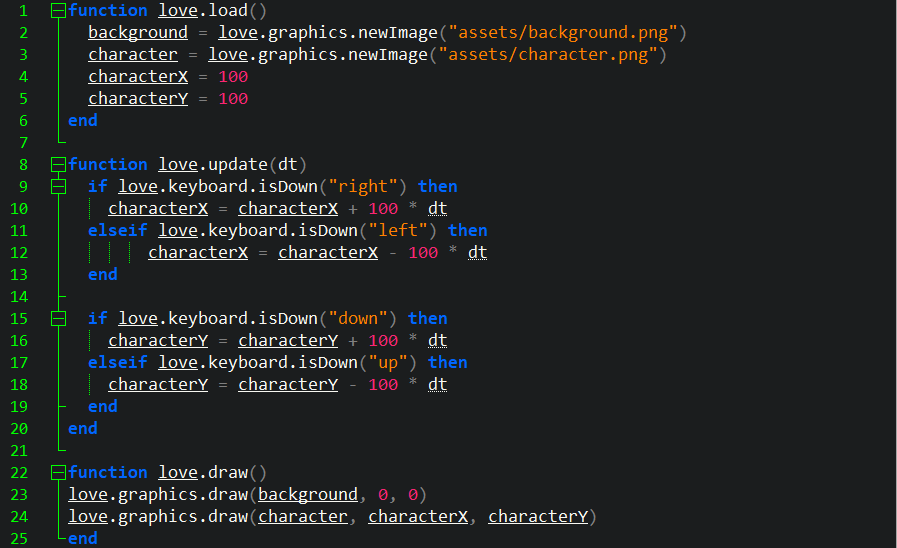
Dans cette étape, nous avons chargé l'image du personnage dans notre programme et l'avons dessiné sur l'écran à une position spécifique à l'aide de la fonction love.graphics.draw(). Vous pouvez modifier les coordonnées de la fonction love.graphics.draw() pour ajuster la position du personnage sur l'écran.



## Déplacer le personnage

Maintenant que nous avons ajouté un personnage, nous allons le déplacer en réponse aux touches du clavier.

Ajoutez le code suivant à votre fichier main.lua :

****

Dans cette étape, nous avons ajouté deux variables characterX et characterY pour stocker la position du personnage. Nous avons également ajouté une fonction love.update(dt) qui est appelée à chaque frame et qui met à jour la position du personnage en fonction des touches du clavier.

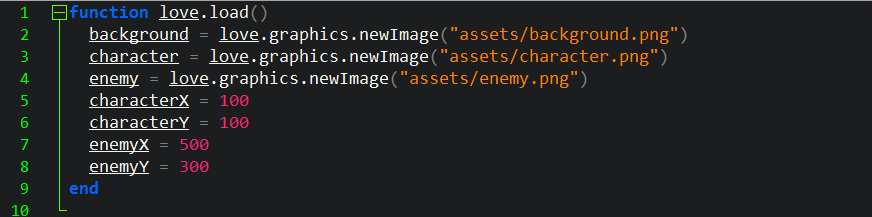


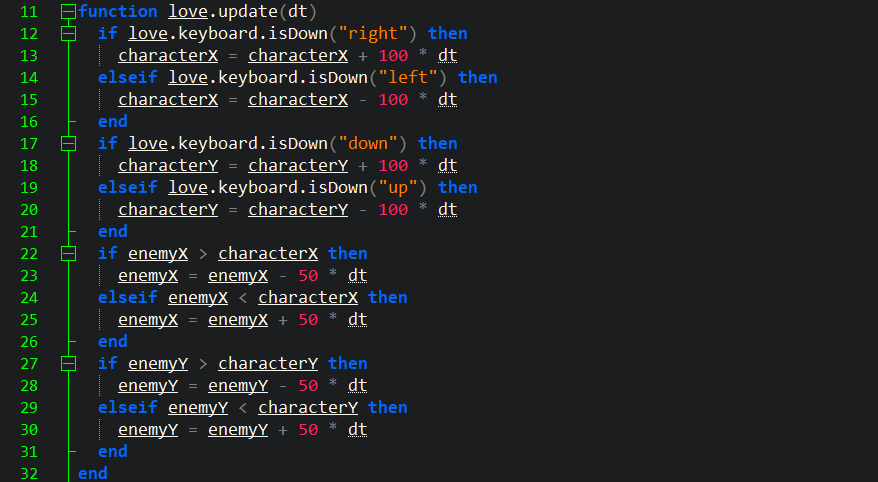
## Ajouter un ennemi

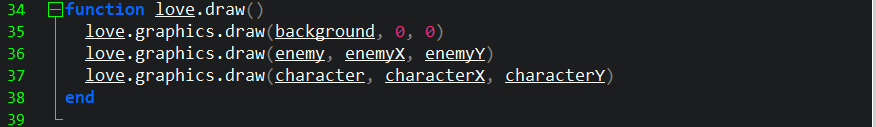
Maintenant que nous avons ajouté un personnage et que nous pouvons le déplacer avec les touches du clavier, nous allons ajouter un ennemi qui suit le personnage.

Ajoutez une nouvelle image dans le dossier assets appelée "enemy.png". C'est une image qui représente notre ennemi. Téléchargez ou créez votre propre image pour cette étape.

Ajoutez le code suivant à votre fichier main.lua :

****

****

****

Dans cette étape, nous avons chargé l'image de l'ennemi dans notre programme et l'avons dessiné sur l'écran à une position spécifique à l'aide de la fonction love.graphics.draw(). Nous avons également ajouté deux variables enemyX et enemyY pour stocker la position de l'ennemi et une fonction love.update(dt) qui est appelée à chaque frame et qui met à jour la position de l'ennemi en fonction de la position du personnage.

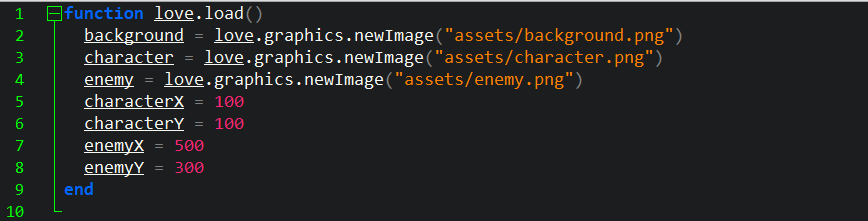
Maintenant, lorsque vous déplacez le personnage avec les touches du clavier, l'ennemi le suit.

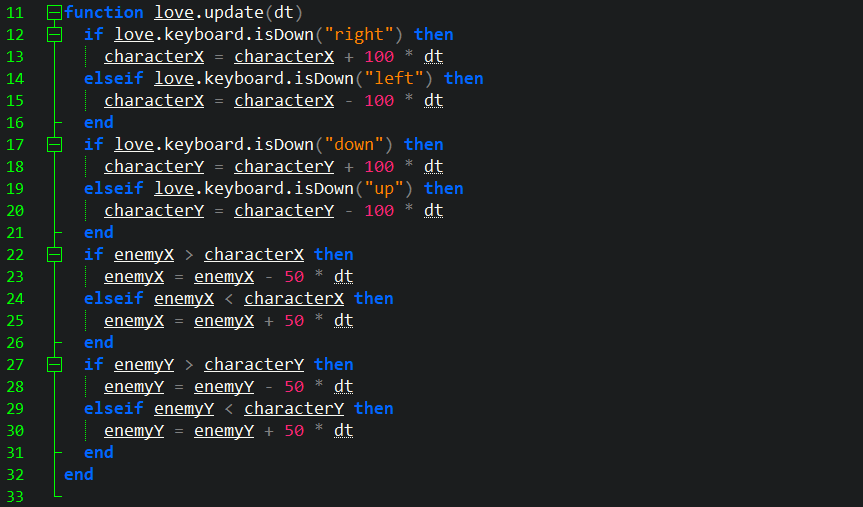
## Ajouter un ennemi

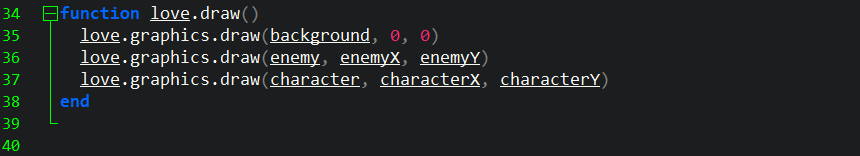
Maintenant que nous avons ajouté un personnage et que nous pouvons le déplacer avec les touches du clavier, nous allons ajouter un ennemi qui suit le personnage.

Ajoutez une nouvelle image dans le dossier assets appelée "enemy.png". C'est une image qui représente notre ennemi. Téléchargez ou créez votre propre image pour cette étape.

Ajoutez le code suivant à votre fichier main.lua :



****

****

Dans cette étape, nous avons chargé l'image de l'ennemi dans notre programme et l'avons dessiné sur l'écran à une position spécifique à l'aide de la fonction love.graphics.draw(). Nous avons également ajouté deux variables enemyX et enemyY pour stocker la position de l'ennemi et une fonction love.update(dt) qui est appelée à chaque frame et qui met à jour la position de l'ennemi en fonction de la position du personnage.

Maintenant, lorsque vous déplacez le personnage avec les touches du clavier, l'ennemi le suit.

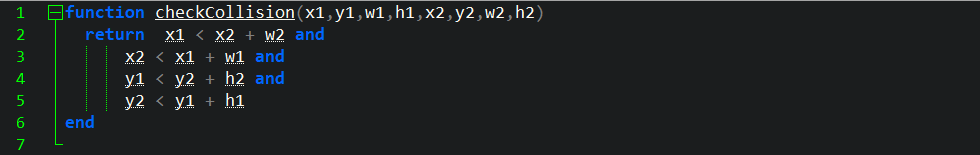


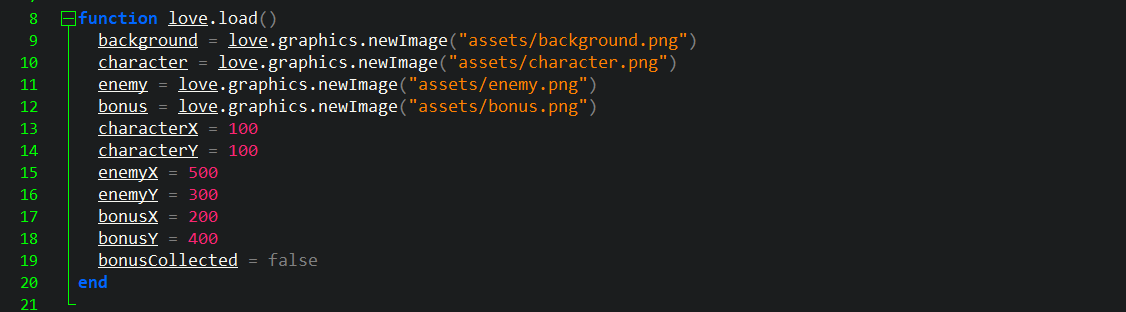
## Ajouter un objet bonus

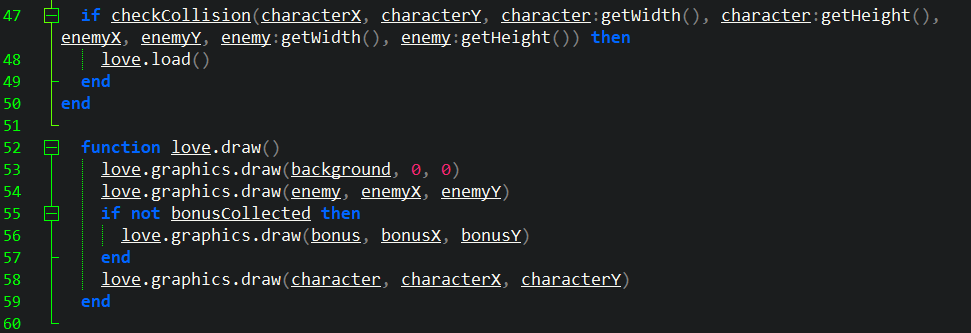
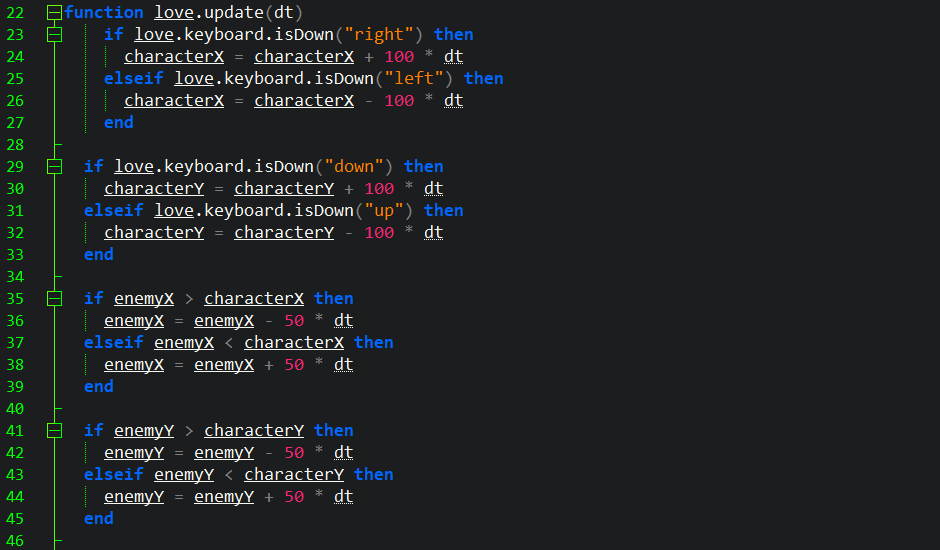
Maintenant que nous avons ajouté un personnage, un ennemi et une collision, nous allons ajouter un objet bonus qui peut être collectés par le personnage.

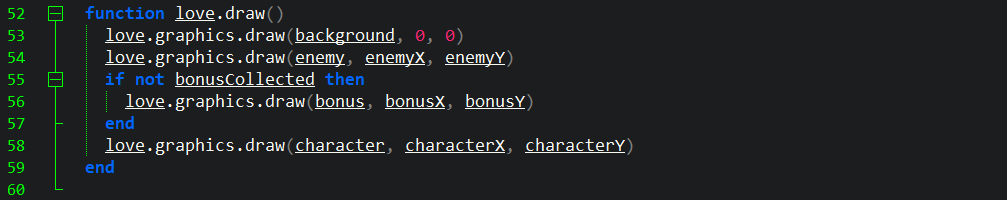
Ajoutez une nouvelle image dans le dossier assets appelée "bonus.png". C'est une image qui représente notre objet bonus. Téléchargez ou créez votre propre image pour cette étape.

Ajoutez le code suivant à votre fichier main.lua :

****

****

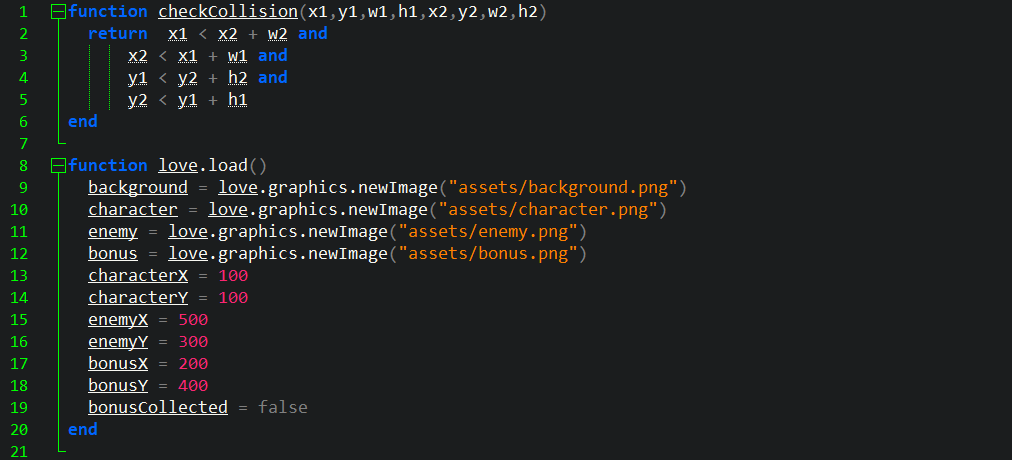
****

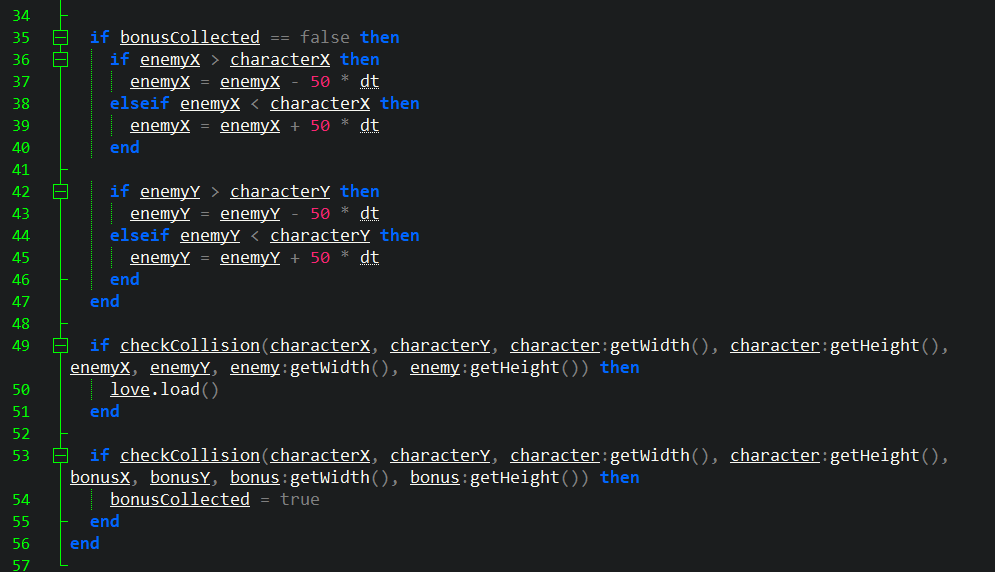


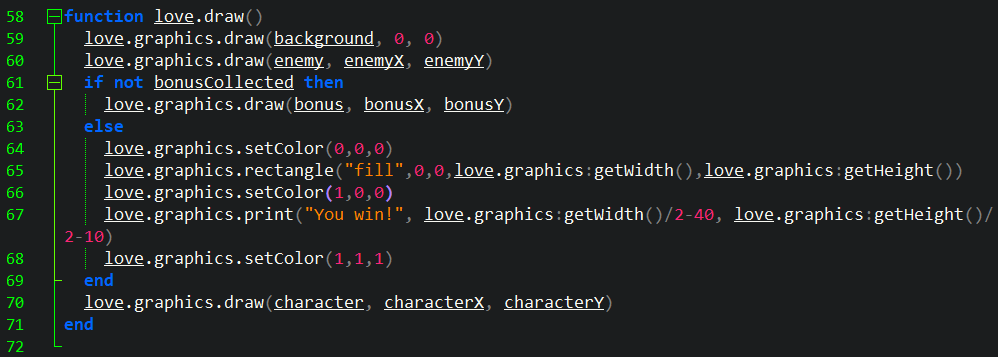
## Afficher un message de victoire

Maintenant que nous avons ajouté un objet bonus, nous allons ajouter un message de victoire pour indiquer que le joueur a réussi à collecter l’objet bonus.

Ajoutez le code suivant à votre fichier main.lua :

****





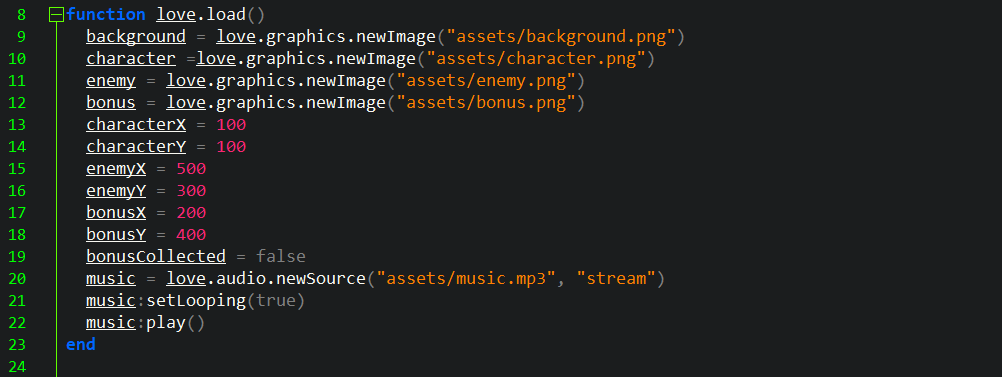
Dans cette étape, nous avons modifié love.update() et ajouté une condition à la fonction love.draw() pour afficher le message "You win!" si la variable bonusCollected est true. Nous avons utilisé la fonction love.graphics.print() pour dessiner le texte centré à l'écran.

## Ajouter de la musique

Maintenant que nous avons ajouté des objets bonus et un message de victoire, nous allons ajouter de la musique pour ajouter une ambiance sonore à notre jeu.

Ajoutez un nouveau fichier audio dans le dossier assets appelé "music.mp3". Téléchargez ou créez votre propre fichier audio pour cette étape.

Ajoutez le code suivant à votre fichier main.lua :

****

Dans cette étape, nous avons ajouté une nouvelle variable music qui est initialisée avec le fichier audio "music.mp3". Nous avons également utilisé la fonction love.audio.newSource() pour créer un objet de source audio et nous avons utilisé les fonctions setLooping() et play() pour faire jouer la musique en boucle.

## Ajouter des effets sonores

Maintenant que nous avons ajouté de la musique, nous allons ajouter des effets sonores pour ajouter des sons à notre jeu.

Ajoutez deux nouveaux fichiers audios dans le dossier assets appelés "collect.wav" et "hit.wav". Téléchargez ou créez vos propres fichiers audios pour cette étape.

Ajoutez le code suivant à votre fichier main.lua :

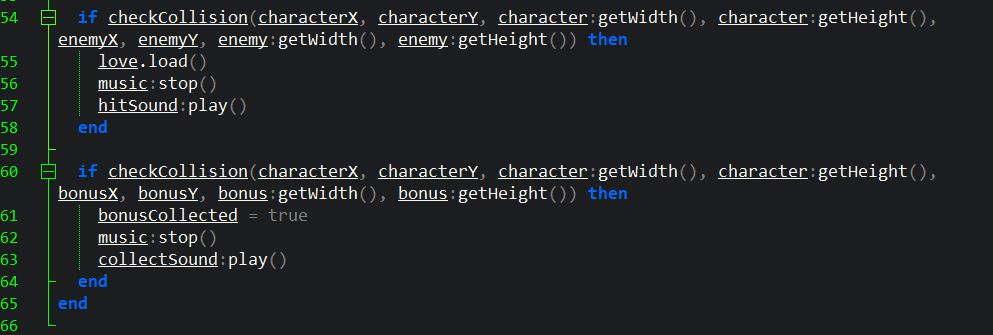
****

Dans cette étape, nous avons créé deux nouvelles variables collectSound et hitSound qui sont initialisées avec les fichiers audio "collect.wav" et "hit.wav". Nous avons également utilisé la fonction love.audio.newSource() avec le deuxième argument "static" pour créer des objets de source audio qui ne sont pas en streaming.

## Jouer des effets sonores

Maintenant que nous avons ajouté des effets sonores, nous allons les jouer lorsque le joueur collecte un objet bonus ou touche l'ennemi.

Ajoutez le code suivant à votre fichier main.lua :

****

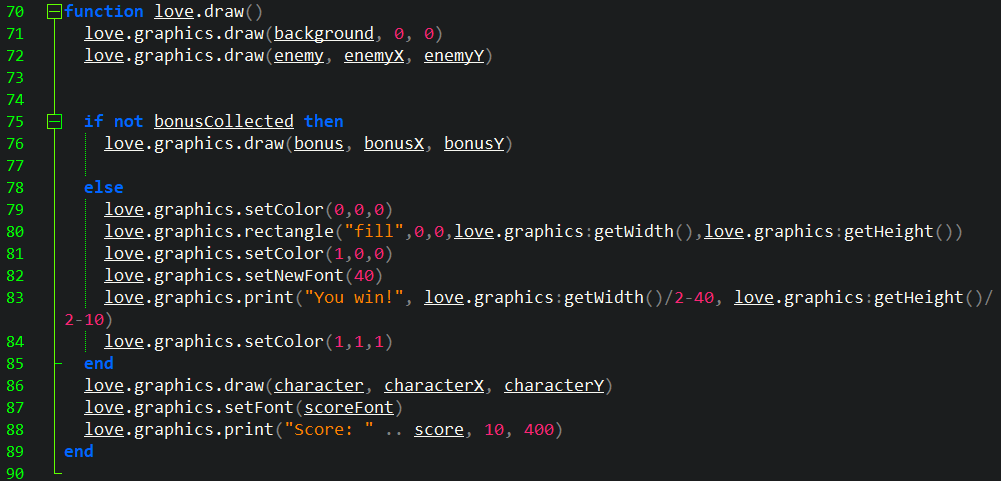
Dans cette étape, nous avons ajouté du code à la fonction love.update() pour jouer les effets sonores collectSound et hitSound lorsque le joueur collecte un objet bonus ou touche l'ennemi. Nous avons utilisé la fonction checkCollision() pour détecter les collisions entre les objets.

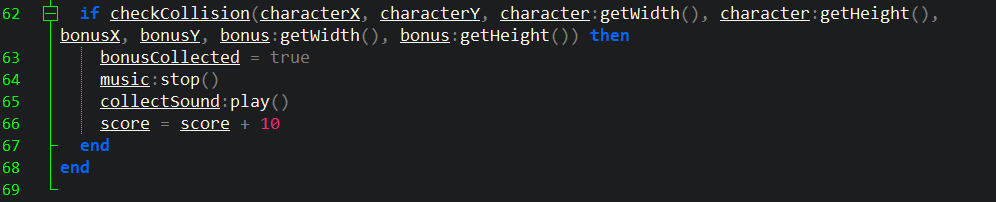
## Ajouter un score

Maintenant que nous avons ajouté des effets sonores, nous allons ajouter un score pour suivre combien d'objets bonus le joueur a collecté.

Ajoutez le code suivant à votre fichier main.lua :

****

****

****

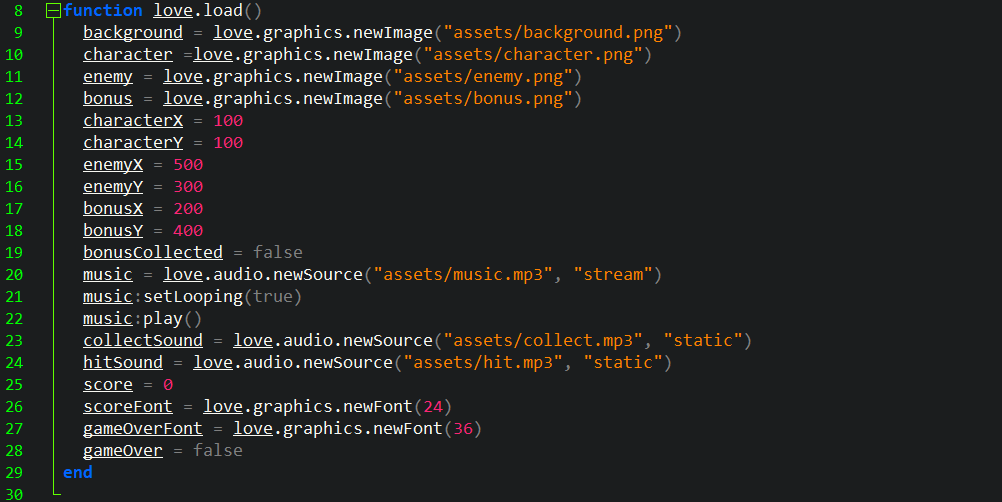
Dans cette étape, nous avons créé deux nouvelles variables score et scoreFont pour suivre le score du joueur et afficher le score à l'écran. Nous avons également ajouté du code à la fonction love.draw() pour afficher le score à l'écran.

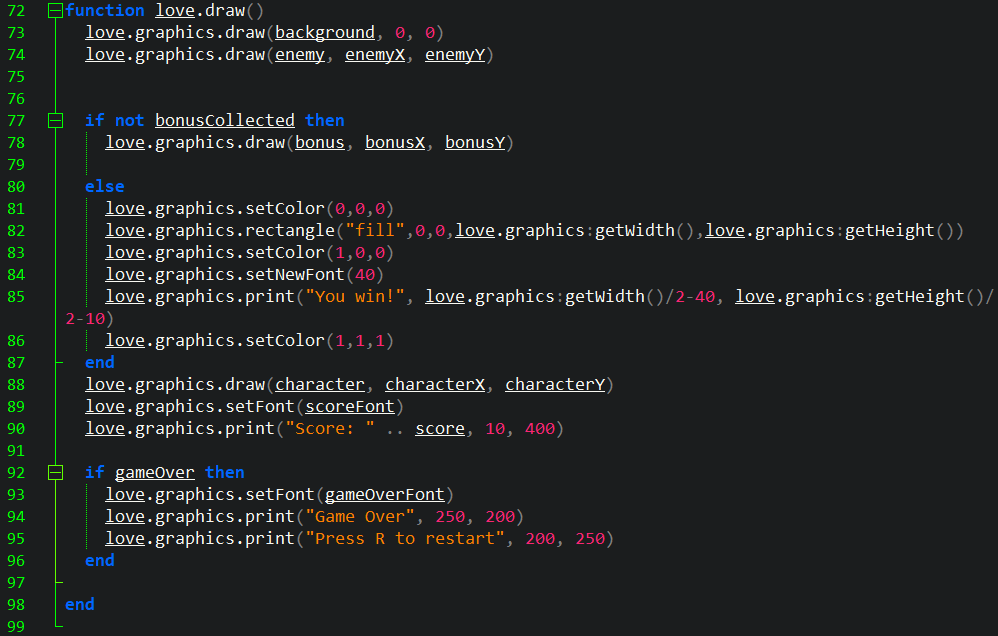
Dans la fonction love.update(), nous avons augmenté le score de 1 chaque fois que le joueur collecte un objet bonus. Le score est affiché à l'écran avec la fonction love.graphics.print().

## Ajouter un écran de fin de jeu

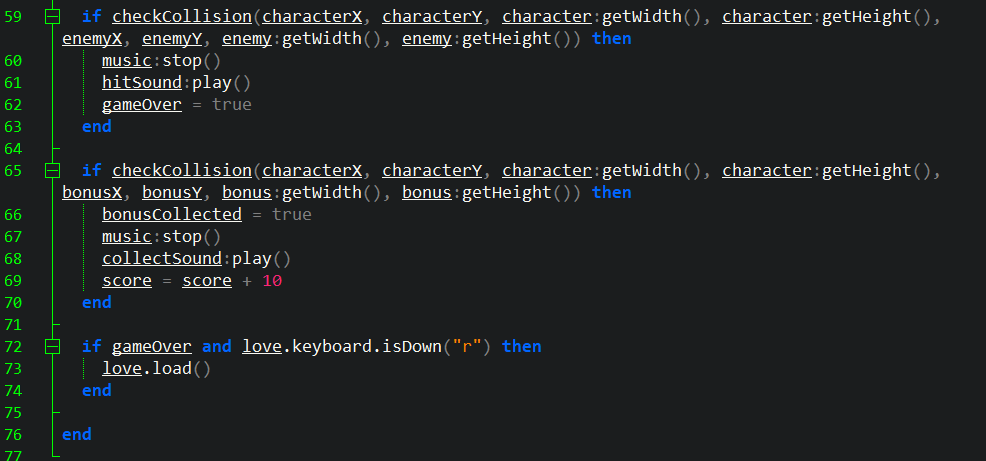
Maintenant que nous avons ajouté un score, nous allons ajouter un écran de fin de jeu qui s'affiche lorsque le joueur est touché par l'ennemi.

Ajoutez le code suivant à votre fichier main.lua :

****

****

****

****

Dans cette étape, nous avons ajouté une nouvelle variable gameOver et une nouvelle police de caractères gameOverFont pour afficher l'écran de fin de jeu. Dans la fonction love.draw(), nous avons ajouté du code pour afficher l'écran de fin de jeu lorsque la variable gameOver est définie sur true.

Dans la fonction love.update(), nous avons ajouté du code pour définir la variable gameOver sur true lorsque le joueur est touché par l'ennemi. Nous avons également ajouté du code pour redémarrer le jeu lorsque le joueur appuie sur la touche "r" après la fin de partie.

Félicitations, vous avez maintenant créé un jeu complet en utilisant Love2D ! N'hésitez pas à expérimenter avec le code et ajouter des fonctionnalités supplémentaires pour améliorer le jeu d’autant que notre jeu contient plusieurs bugs que je vous laisserai corriger à votre aise chez vous…

Ceci clôture, le tutoriel de base. Grâce à lui, nous avons les outils nécessaires pour démarrer notre projet de jeux de plateforme.

Il est important de noter que ce guide est purement pédagogique et n’a pour but que de vous apprendre les notions de base du framework Love2D.

Nous verrons durant ce stage, beaucoup de notions complémentaires comme l’animation des personnages, l’affichage d’un tilemap, la notion de camera, et bien d’autres effets graphiques amusant pour que vous puissiez rendre vos futurs jeux des plus amusants et graphiquement sublime….

Bon courage les Codeurs Commandos.

## Remerciements :

Liolabs tiens particulièrement à remercier la maison des jeunes de Dour sans qui ce stage n’aurait pas avoir eu lieu.

Si d’aventure vous souhaitiez participer à d’autres stages de programmations de jeux Vidéos. N’hésitez pas à en faire part à la maison des jeunes ou en me laissant un petit message sur l’email : **liolabs1973@gmail.com**