



# TP2 : Le Tile Mapping

## Table des matières :

- 1. Le tile Mapping : ..... 1
- 2. Tileset ..... 2
- 3. L'éditeur de carte TILED ..... 4
  - a) Création de la carte : ..... 4
  - b) Import du tileset ..... 4
  - c) Création des couches (layers) ..... 5
  - d) Export JSON ..... 7
- 4. Reconstitution de la carte avec Processing ..... 8
  - a) Lecture du fichier JSON ..... 8
  - b) Extraire une tuile du tileset ..... 9
  - c) Reconstruire la carte ..... 10

### 1. Le tile Mapping :

Le tile Mapping est une technique de création de carte utilisée depuis la création des premiers jeux vidéo.

Afin d'éviter de gaspiller une grande quantité de mémoire avec l'utilisation de grandes images pour réaliser le décor d'un jeu, le tile mapping va reconstruire le décor à partir de motifs nommés **tiles**. Le concept de Tile Mapping est de placer côte à côte des "**tiles**" (des tuiles en anglais) dans une fenêtre de taille prédéfinie. L'écran est alors divisé en une grille dont la taille de chaque case correspond à celle d'une tuile.



Sur cette image ci-dessus, Mario, l'ennemi, l'étoile, et les nuages, sont des **sprites** que l'on insère sur le décor composé de tuiles. Il y a les tuiles uniques, comme les briques et les points d'interrogation, et les tuiles composées, comme le pot de fleur.

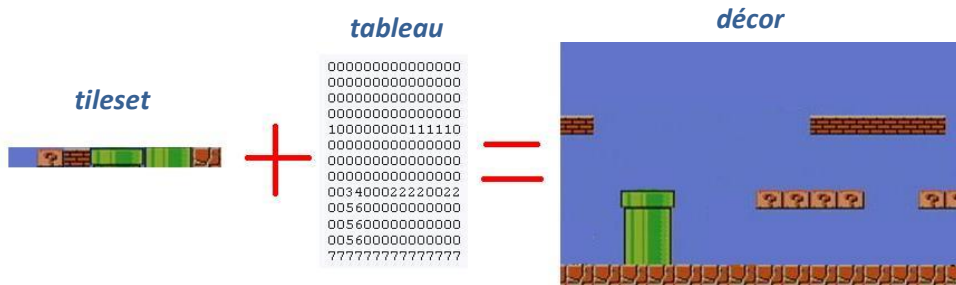
Le décor est donc composé de 8 tuiles différentes :

- 1 tuile ciel
- 1 tuile sol
- 1 tuile point d'interrogation
- 1 tuile brique
- 4 tuiles du pot de fleurs

Cet ensemble forme notre **tileset** :



La dimension du décor est ici de 13\*15 cases soit 195 cases. Nous pouvons donc imaginer un tableau de 195 cases où la valeur 0 correspond à la tuile ciel, la valeur 1 correspond à la tuile brique et ainsi de suite.



Pour faire ce décor, nous avons donc besoin d'un tileset et du tableau.

Si l'on considère que nous avons 256 tuiles différentes (1 octet de 8 bits soit  $2^8 = 256$ ), nous pouvons compter un octet par case soit 195 octets.

Considérons, un niveau entier de 300 cases de long sur 20 cases de haut :



Cela nous donne  $300 * 20 = 6000$  octets.

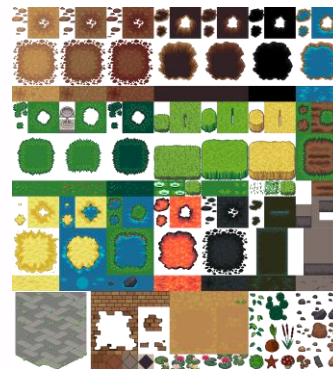
Sans cette méthode de tile mapping, nous aurions utilisé une grande image d'une dizaine de MégaOctets.

**Travail n°1 :**  
Si le tileset est composé de 65536 tuiles et que l'on souhaite créer une carte de 600 tuiles de long sur 30 tuiles de haut, calculez la taille du fichier en octet.

## 2. Tileset

Un **tileset** est une image qui contient un ensemble de tuiles pour créer le décor de votre jeu.

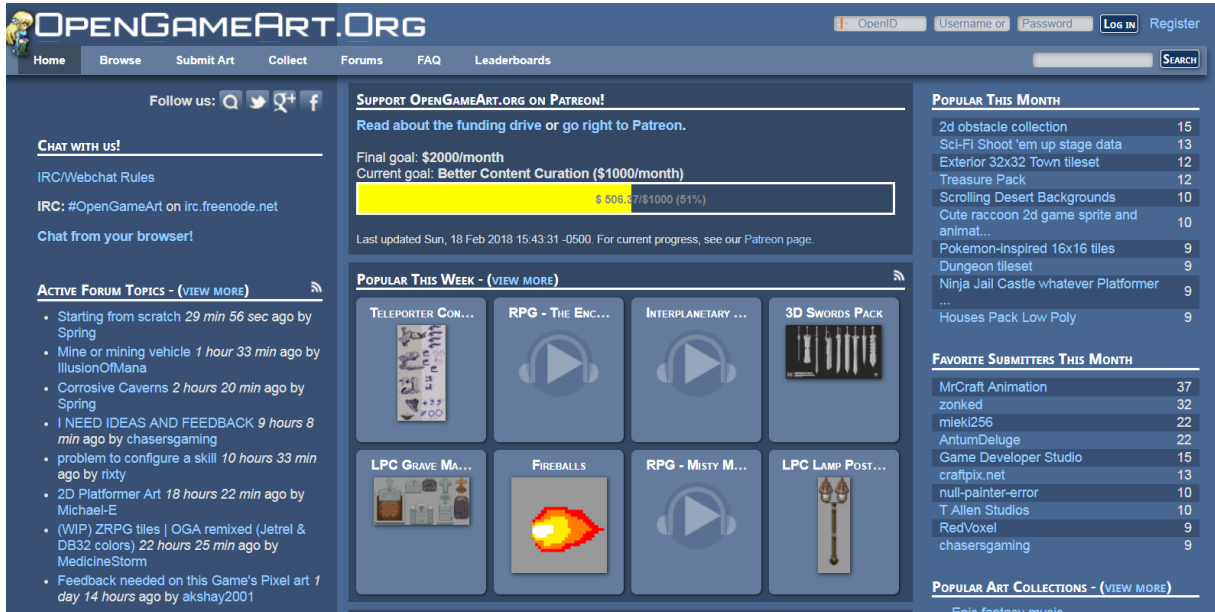
### Exemple de tileset :





On trouve facilement des tilesets sur internet. Il faut cependant penser à vérifier les licences pour pouvoir les utiliser dans votre jeu. Sinon vous ne pourrez pas diffuser votre projet.

Le site OpenGameArt.org propose de nombreuses ressources libres de droit (tilesets, sprites, effets sonores, ...).



Vous trouverez le tileset ci-contre à l'adresse suivante :

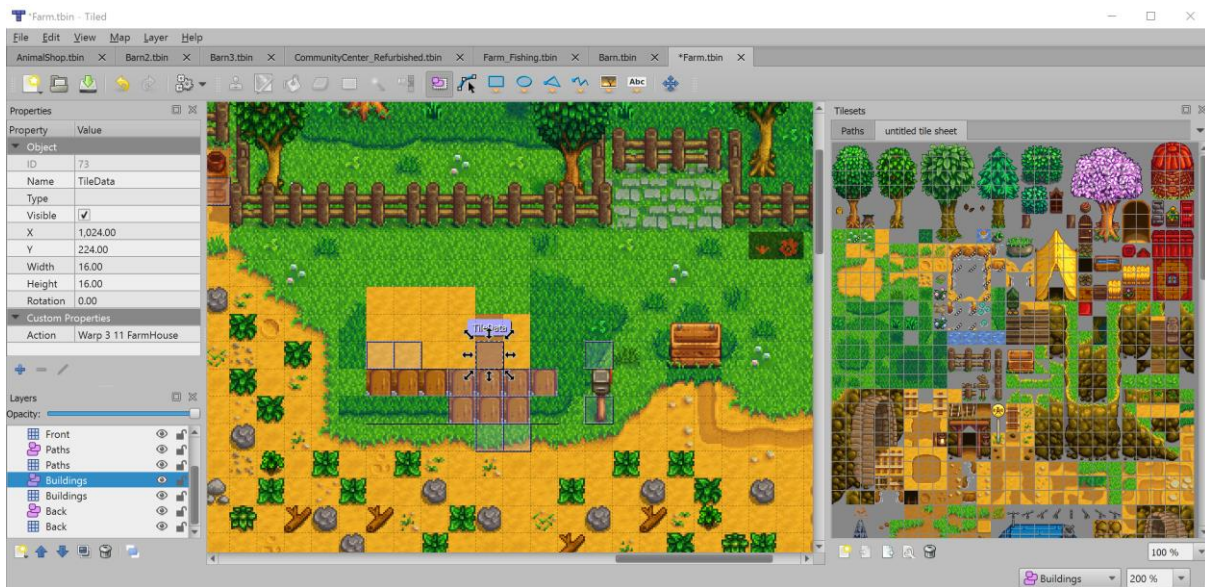
<https://opengameart.org/content/orthographic-outdoor-tiles>

Le tileset est composé de tuiles de 16px\*16px.



### 3. L'éditeur de carte TILED

**Tiled** est un logiciel gratuit qui permet de créer des cartes en vue 2D classique ou 2D isométrique. Il est open source et possède plusieurs formats d'export, dont le **JSON**.

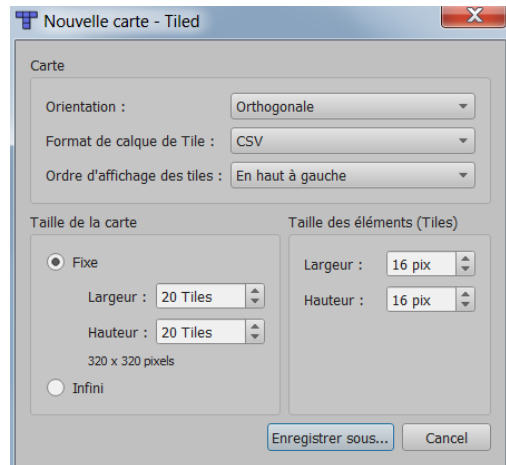


#### a) Création de la carte :

Sélectionnez « **Fichier/Nouveau/Nouvelle carte...** » et créez une carte de type Orthogonale (2D classique) avec une largeur et une hauteur de 20 tiles. Fixez aussi la taille des éléments (les tiles) à 16 pixels.

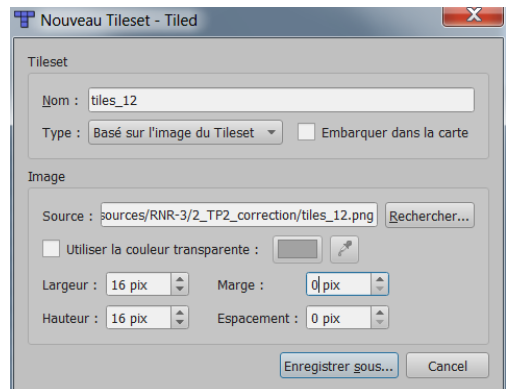
Sélectionnez le format du calque CSV pour faciliter l'export JSON.

Enregistrez votre nouvelle carte (test.tmx) dans un dossier et ajoutez-y le tileset téléchargé précédemment.



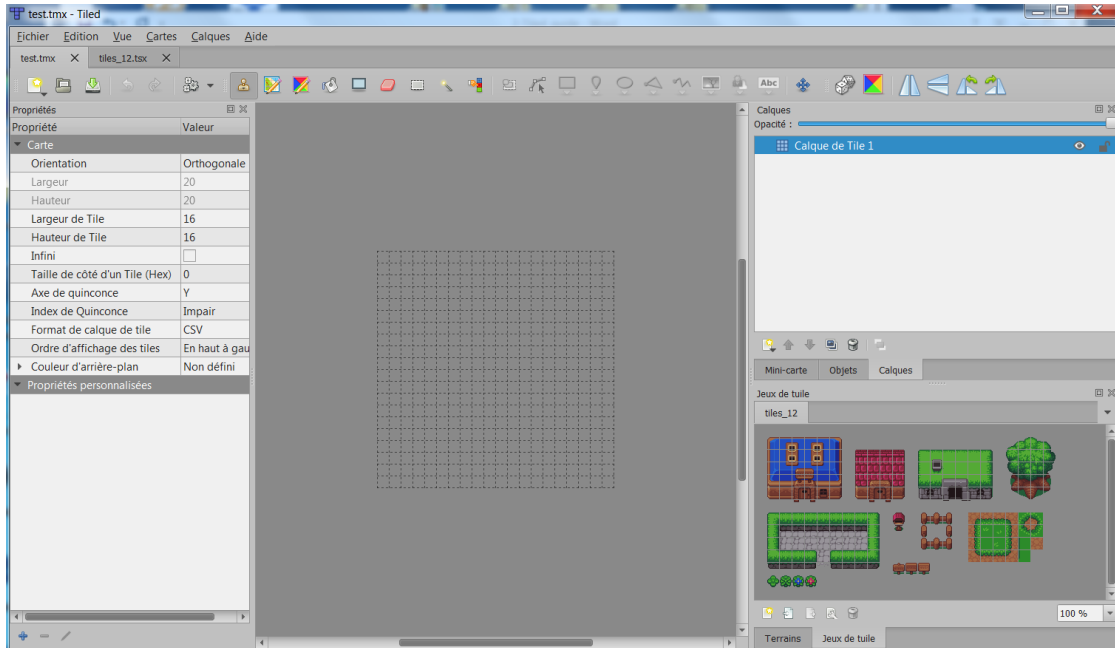
#### b) Import du tileset

Sélectionnez « **Fichier/Nouveau/Nouvelle tileset...** » et sélectionnez votre **tileset** téléchargé précédemment. Puis paramétrez votre tileset (16 pix de largeur et de hauteur).



Algorithmes, Codage numérique, Traitement d'images, Traitement du Son, Cryptage de données, Cloud, Robotique, Applications Web, Stéganographie, Open DATA, Algorithmes, Codage numérique, Traitement d'images, Traitement du Son, Cryptage de données, Applications Web, Algorithmes, Cloud, Open DATA, Codage, Stéganographie

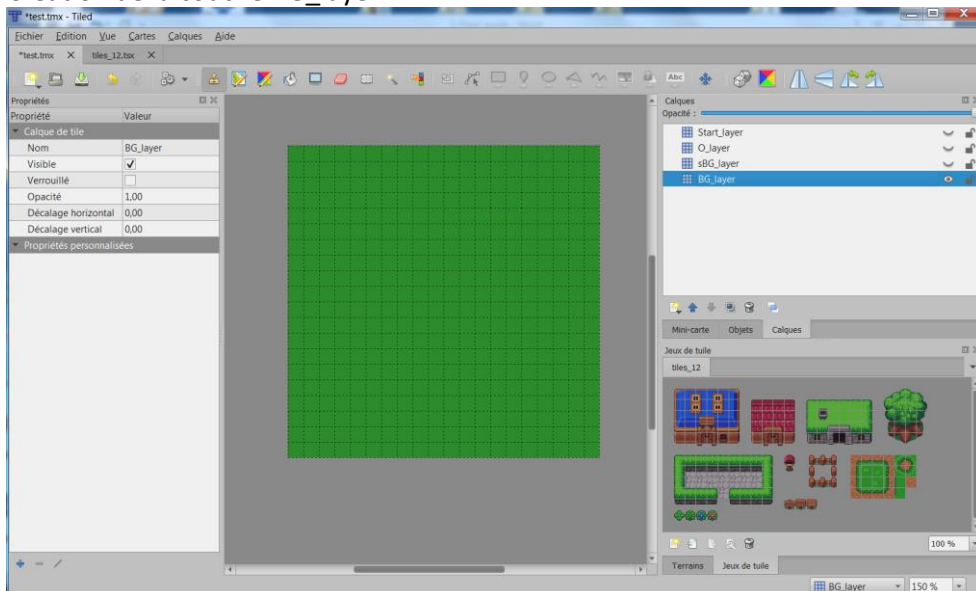
Vous obtenez le résultat suivant :



### c) Création des couches (layers)

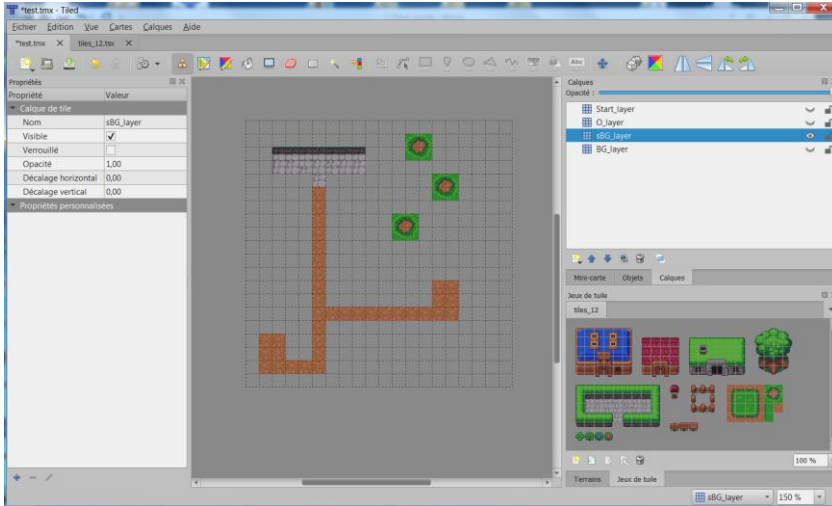
Pour créer votre carte, vous allez procéder par couche. Une couche de fond (BG\_layer), une sur-couche (sBG\_layer) pour personnaliser le décor et une couche d'obstacles (O\_layer) qui correspond aux zones où le personnage ne peut pas se déplacer. On peut rajouter une autre couche pour placer la tuile de départ (Start\_layer). Sélectionnez simplement la tuile dont vous avez besoin sur la droite et effectuez un drag&drop avec la souris pour la placer à l'écran. La barre d'outils en haut donne accès à la gomme et à différents outils de sélection.

Création de la couche BG\_layer :

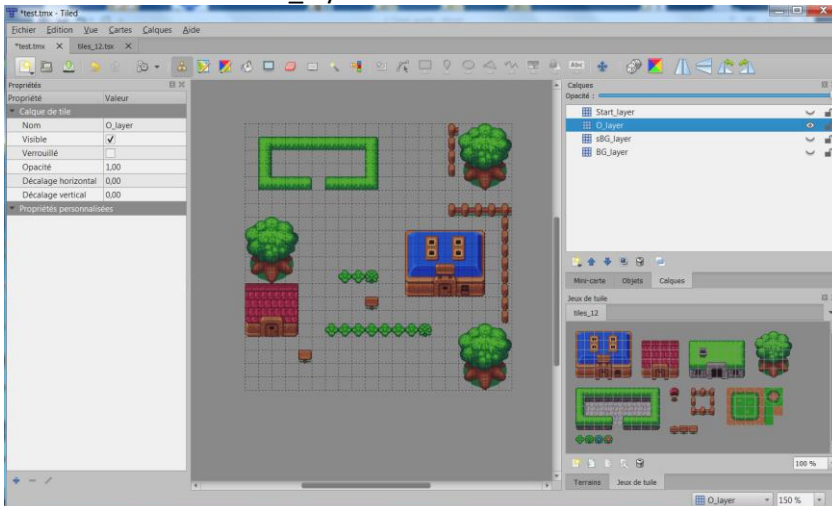


Algorithme, Codage numérique, Traitement d'images, Traitement du Son, Cryptage de données, Cloud, Robotique, Applications Web, Stéganographie, Open DATA, Algorithme, Codage numérique, Traitement d'images, Traitement du Son, Cryptage de données, Applications Web, Algorithme, Cloud, Open DATA, Codage, Stéganographie

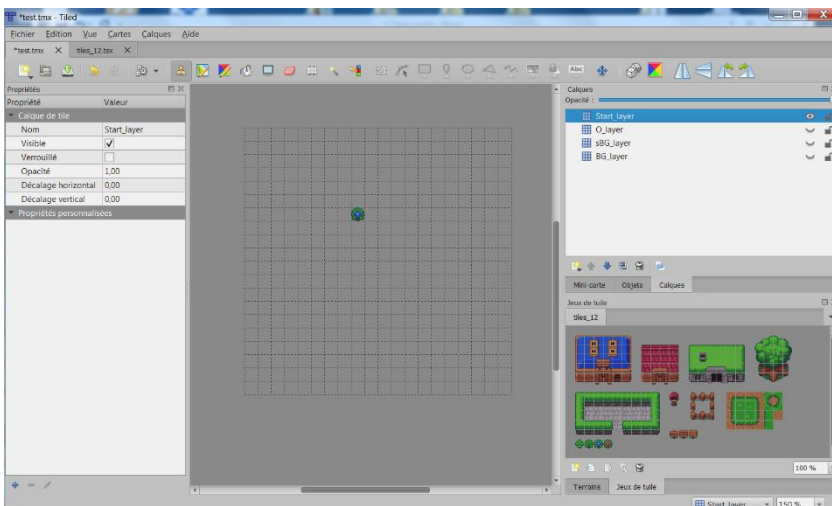
Création de la couche sBG\_layer :



Création de la couche O\_layer :



Création de la couche Start\_layer :





Algorithme, Codage numérique, Traitement d'images, Traitement du Son, Cryptage de données, Cloud, Robotique, Applications Web, Stéganographie, Open DATA, Algorithme, Codage numérique, Traitement d'images, Traitement du Son, Cryptage de données, Applications Web, Algorithme, Cloud, Open DATA, Codage, Stéganographie

La tuile utilisée pour le background a son coin supérieur gauche en position 288 :144 sur l'image du tileset.



### Travail n°2 :

Après avoir réalisé l'ensemble des étapes précédentes et exporté le fichier JSON, justifiez que l'on retrouve bien la valeur 244 du fichier JSON en partant des coordonnées 288 :144 de tuile dans le tileset.

## 4. Reconstitution de la carte avec Processing

A l'aide de Processing nous allons recréer la carte, couche par couche, à partir du fichier JSON.

### a) Lecture du fichier JSON

Pour travailler avec le format JSON, Processing va utiliser les objets de type *JSONObject* ou *JSONArray*.

```
JSONObject map;
int nb_tuile_x, nb_tuile_y, taille_tuile_x, taille_tuile_y;
int[] BG_layer;

void setup()
{
  size(320, 320);
  map = loadJSONObject("test.json");

  taille_tuile_x= map.getInt("tileheight");
  taille_tuile_y= map.getInt("tilewidth");
  nb_tuile_x = map.getInt("width");
  nb_tuile_y = map.getInt("height");
  println("Les tuiles sont de dimensions : "+taille_tuile_x+"px par "+taille_tuile_y+"px");
  println("La carte a pour dimension : "+nb_tuile_x+" tuiles de long sur "+nb_tuile_y+" tuiles de haut");
  println("Soit un total de : "+nb_tuile_x*nb_tuile_y+" tuiles");

  JSONArray layers = map.getJSONArray("layers");
  println("La carte est composée de "+layers.size()+" couches");

  JSONObject layer0 = layers.getJSONObject(0);
  JSONArray data0 = layer0.getJSONArray("data");
  BG_layer = data0.getIntArray();
  println("=====");
  println("les "+nb_tuile_x*nb_tuile_y+" Valeurs de la première couche : ");
  println("=====");
  println(BG_layer);
}

void draw() {}
```

### Travail n°3 :

Recopiez le code ci-dessous dans un sketch Processing. Testez et analysez le code à l'aide de la documentation Processing (<https://processing.org/reference/JSONObject.html>)

Réalisez ensuite un programme permettant d'afficher dans la console processing les valeurs des 4 couches qui constituent la carte.





Algorithme, [Codage numérique](#), Traitement d'images, [Traitement du Son](#), Cryptage de données, [Cloud](#), [Robotique](#), [Applications Web](#), [Stéganographie](#), [Open DATA](#), Algorithme, [Codage numérique](#), [Traitement d'images](#), [Traitement du Son](#), [Cryptage de données](#), [Applications Web](#), [Algorithme](#), [Cloud](#), [Open DATA](#), [Codage](#), [Stéganographie](#)

### c) Reconstruire la carte

L'objectif final de ce TP est de reconstituer dans Processing la carte réalisée sous Tiled. Pour cela, il faut combiner le code permettant de lire le fichier JSON et le code permettant d'extraire les tuiles du tilset.

#### **Travail n°5 :**

En combinant les codes du travail 3 et 4, réalisez un programme Processing capable de reproduire la carte réalisée sous Tiled avec les 4 couches de tuiles (BG\_layer, sBG\_layer, O\_layer et Start\_layer).

