Atelier Tilemaps : Les bases

Langage : Lua
Framework : Love2D

Introduction

Vous trouverez dans ce support de cours, différents "snippets" (extraits) de code pour gérer vos Tilemaps.

Les projets sont disponibles ici :
https://www.dropbox.com/sh/wxlag1u958eahyj/AAD_dsd-DZHoh003wJ69RRXta?dl=0

Déclarer une Tilemap

On utilise un tableau à 2 dimensions pour stocker des id (identifiants) correspondant au type de tuile (terrain, mur, etc.)

Exemple pour 10 lignes de 10 colonnes :

```
Game.Map = {}
Game.Map =  {
              { 2,4,4,2,2,2,2,2,2,2 },
              { 2,1,2,2,0,0,2,2,1,2 },
              { 2,2,2,2,2,2,2,2,2,2 },
              { 2,5,2,2,2,2,2,2,2,2 },
              { 2,2,2,2,3,3,2,2,2,2 },
              { 2,2,2,2,3,3,2,2,2,2 },
              { 2,2,2,2,2,2,2,2,2,2 },
              { 2,2,2,2,2,2,2,2,2,2 },
              { 2,1,2,2,2,2,2,2,1,2 },
              { 2,2,2,2,2,2,2,2,2,2 },
            }
```

Accéder à une cellule de la map

```
id = Game.Map[ligne][colonne]
```

Exemple pour accéder à l'id de la cellule en ligne 2, colonne 3 :

```
id = Game.Map[2][3]
```

Mais dans quasiment 100% des cas on n'accède pas directement à une cellule avec des coordonnées "en dur", on utilise des variables.

Charger des textures pour les tiles

Une méthode simple consiste à utiliser un tableau dont les éléments sont indexés sur les id des tuiles :

```
Game.TileTextures[0] = {}
Game.TileTextures[0] = nil
Game.TileTextures[1] = love.graphics.newImage("images/grassCenter.png")
Game.TileTextures[2] = love.graphics.newImage("images/liquidLava.png")
Game.TileTextures[3] = love.graphics.newImage("images/liquidWater.png")
Game.TileTextures[4] = love.graphics.newImage("images/snowCenter.png")
Game.TileTextures[5] = love.graphics.newImage("images/stoneCenter.png")
```

Parcourir la map pour l'afficher

```
function Game.Draw()
  local c,l

  for l=1,MAP_HEIGHT do
    for c=1,MAP_WIDTH do
      local id = Game.Map[l][c]
      local tex = Game.TileTextures[id]
      if tex ~= nil then
        love.graphics.draw(tex, (c-1)*TILE_WIDTH,
(l-1)*TILE_HEIGHT)
      end
    end
  end
end
```

Exemple :
https://www.dropbox.com/sh/ixwq9dfyzs8ujhu/AAB877hou_MFVm8fZiNmMR30a?dl=0

Connaître la tile à une coordonnées données

```
local c = math.floor(x / TILE_WIDTH) + 1
local l = math.floor(y / TILE_HEIGHT) + 1
local id = Game.Map[l][c]
```

Si la valeur de id est null (id) alors c'est que les coordonnées sont hors de la map.

Voici une version plus avancée, basée sur les coordonnées de la souris. Le code est plus détaillé et comporte un test de validité des coordonnées :

```
local x = love.mouse.getX()
local y = love.mouse.getY()
local c = math.floor(x / TILE_WIDTH) + 1
local l = math.floor(y / TILE_HEIGHT) + 1
if l>0 and c>0 and l <= MAP_HEIGHT and c <= MAP_WIDTH then
  local id = Game.Map[l][c]
  -- Ici la suite de votre code en fonction de l'id
end
```

Exemple :
https://www.dropbox.com/sh/bned0r4288mnb76/AAAiVloohpIHlB0ggTW17jZua?dl=0