



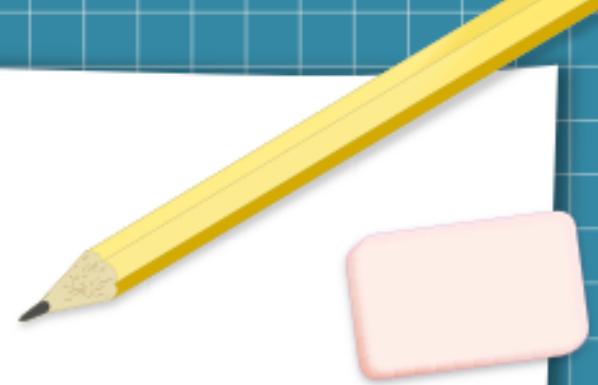
# PyGame

Meetup Python Nantes – Juin 2018

Céline Libéral ([celine.liberal@mail.com](mailto:celine.liberal@mail.com))  
Zoé Belleton ([zobelleton@gmail.com](mailto:zobelleton@gmail.com))

# Sommaire

- Qu'est-ce que PyGame ?
- Pourquoi Pygame ?
- Initialisation de PyGame
- « GameLoop » de base
- Spritesheet & sprite, kesako ?
- Edition de niveau
- Gestion des événements
- Gestion des collisions



# Qu'est-ce que PyGame ?



- Un framework Python orienté vers le jeu
- Surcouche complète à la SDL :
  - SDL Image : manipulation d'images
  - SDL Mixer : manipulation de sons / musiques
  - SDL TTF : utilisation de polices d'écriture
  - SDL Net : utilisation du réseau
  - ...
- Compatible avec Python 3
  - Module « movie » non porté

# Pourquoi PyGame ?



- Compatible Windows, Linux, \*BSD, MacOS, AmigaOS ...
- Plutôt bas niveau (contrairement à Unity ou UnrealEngine)
- Permet de choisir le driver pour le rendu
- Permet de fournir un contexte OpenGL pour des développements en 3D
- Permet de produire très vite un POC :
  - Utile dans les LudumDare ou autre

# Initialisation de PyGame



- Nécessite une initialisation
  - Linux :

```
Python Console
>>> import pygame
>>> pygame.init()
(6, 0)
>>> print(pygame.display.get_driver())
x11
```

- Windows

```
PyDev console: starting.


Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
>>> import pygame
>>> pygame.init()
(6, 0)
>>> print(pygame.display.get_driver())
windib
```

# Initialiser le mixer audio

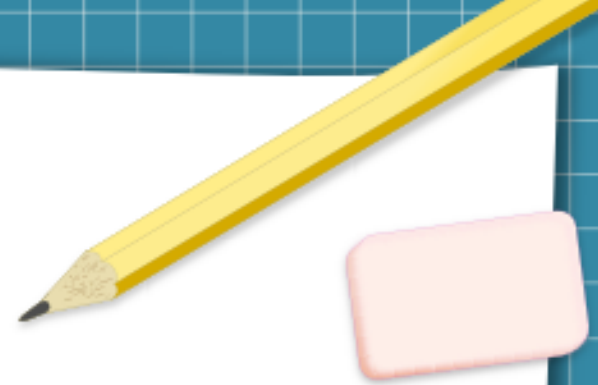


- Chargement de musique de fond :
  - Compatible MP3, OGG et WAV
  - Lecture de la musique en boucle
  - Code :
    - `pygame.mixer.music.load(« fichier »)`
    - `pygame.mixer.music.play()`
- Chargement des bruitages (alias SFX) :
  - Compatible OGG et WAV
  - Code :
    - `sfx=pygame.mixer.Sound(« fichier »)`
    - `sfx.play()`

# Initialiser l'affichage

- Définition d'un titre et de dimensions :
  - `pygame.display.set_caption("Meetup Python Nantes")`
  - `screen = pygame.display.set_mode((800, 600))`
- Possibilité d'être en plein écran
- Possibilité de définir une icône :
  - Par défaut : 
  - Un des avantages de PyGame / SDL !

# Afficher la fenêtre



- PyGame repose sur le « blitting »
- Initialisation d'une « game loop » basique :
  - Effacer l'écran (ou le remplir selon le point de vue)
  - Coller les divers éléments à afficher
    - Image
    - Dessin (figures géométriques fournies)
    - Texte
  - Récupérer les événements :
    - Clavier, souris, joystick, timer ...
  - Mettre à jour l'affichage
  - Afficher



# Spritesheet / Sprite

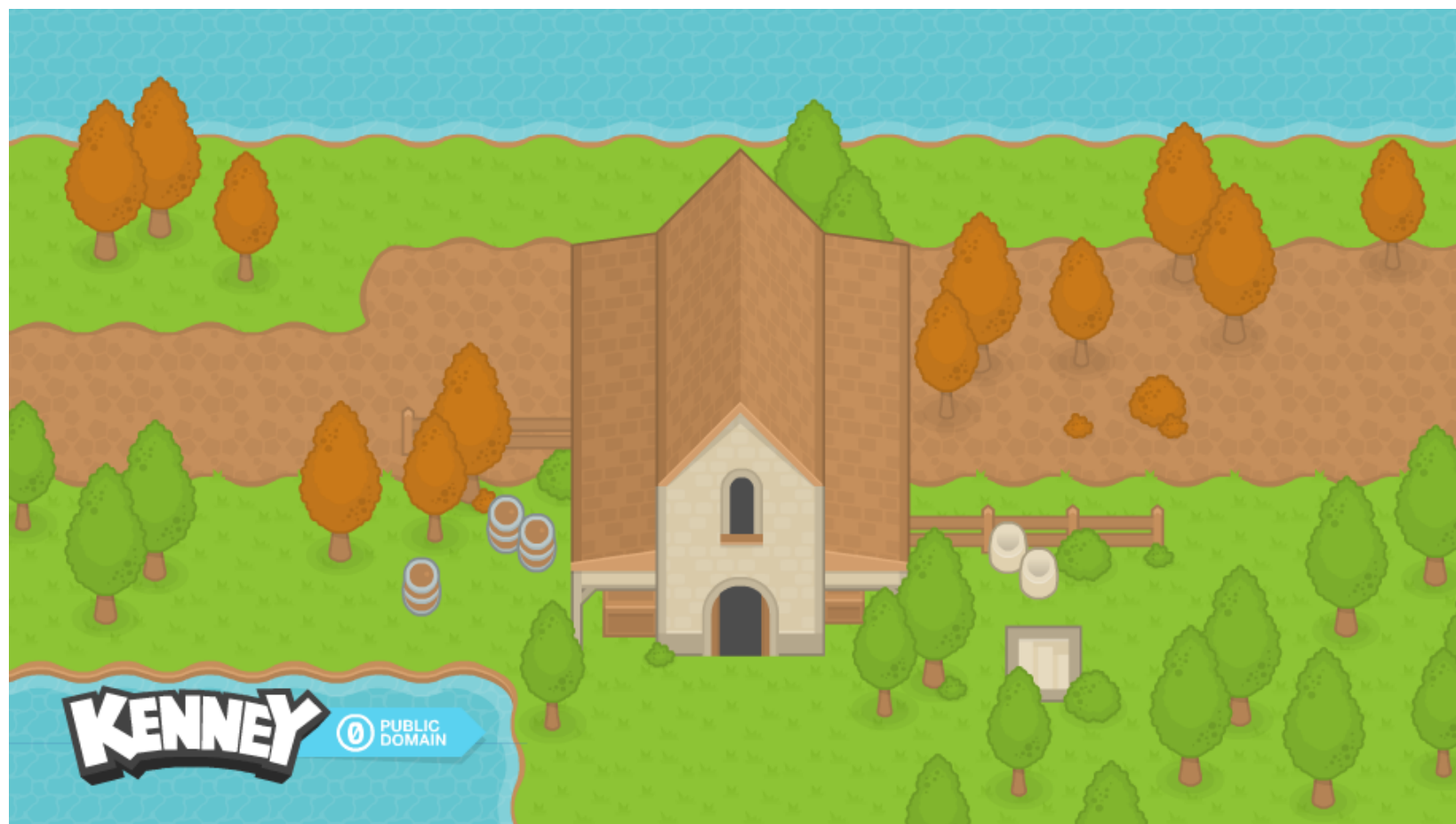


- Spritesheet = ensemble de sprites en 1 fichier image
- Sprite = une partie de la spritesheet :
  - Le personnage X en position Y
  - Un élément de décor
  - ...
- Dans PyGame il existe la classe `pygame.sprite.Sprite` dont on peut faire hériter les différents objets à afficher à l'écran
- Ressource libre :
  - <https://opengameart.org/>

# Exemple de spritesheet



# Exemple d'utilisation

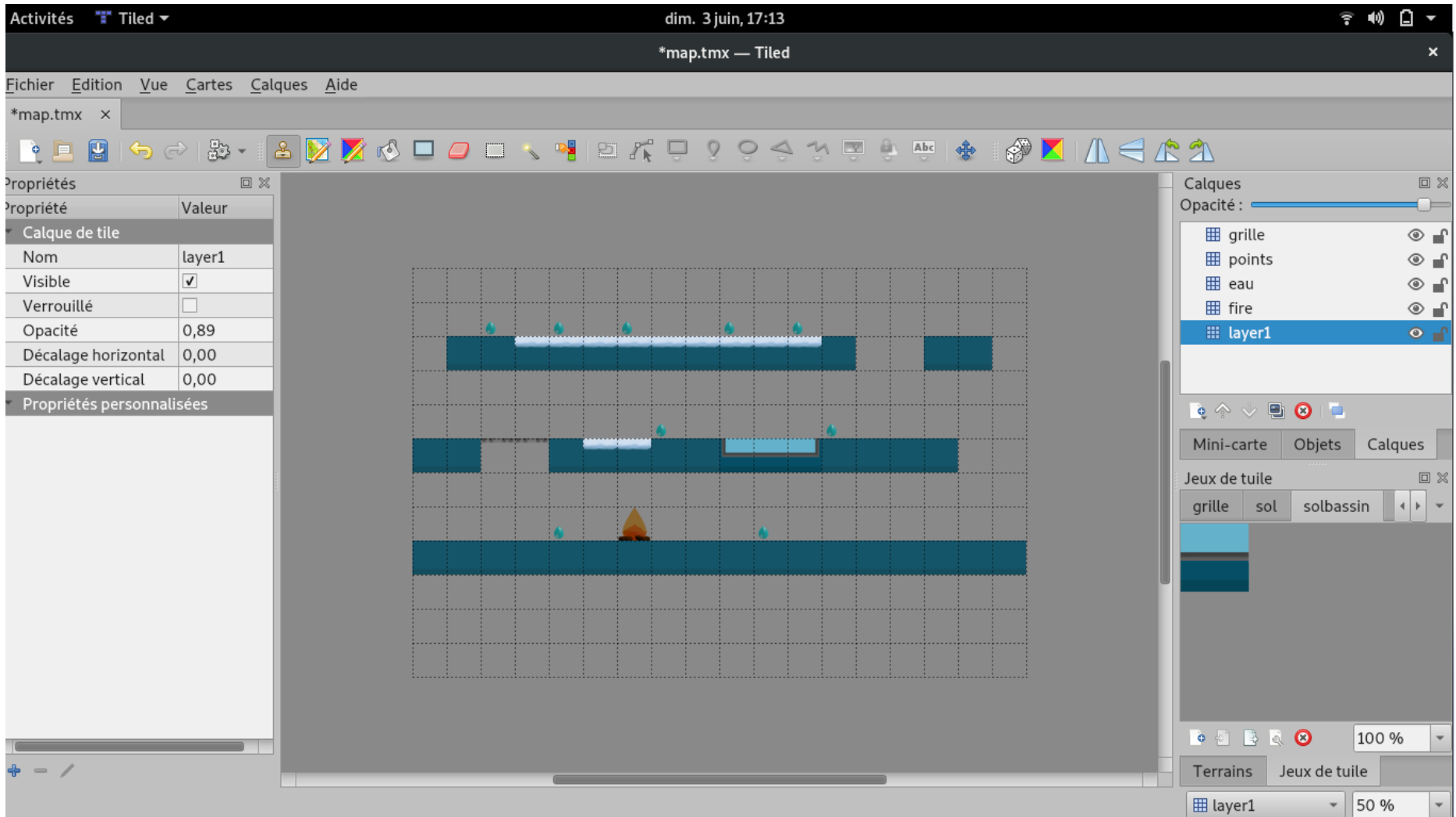


# Création d'un niveau

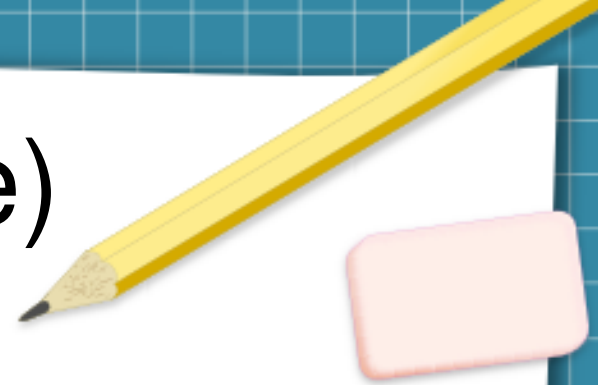


- A l'ancienne :
  - Tableau de chiffres auquel on associe des sprites
- En mode visuel avec Tiled :
  - Logiciel libre sous GPL
  - Site officiel : <http://www.mapeditor.org/>
  - Génère un fichier .tmx (XML)
    - Utilisable via la lib pytmx
    - Compatible avec d'autres frameworks :
      - PhaserJS (en JavaScript)

# Tiled (vue éditeur)



# Tiled (vue PyGame)



# Gestion des événements



- Les événements peuvent provenir :
  - De périphériques :
    - Clavier, souris, joystick, capteur
    - Utile pour :
      - Déplacements
      - Saisie d'information
  - D'événements internes :
    - Timer `pygame.USEREVENT` :
      - Déclencher des sons d'ambiance à intervalle régulier :
        - Exemple « plip-plop » d'une goutte d'eau
      - Chute ou animation d'objets
      - ...

# Gestion des collisions



- Chaque sprite affiché à l'écran possède un objet `Rect()`
- `Rect = Position X, Position Y, Longueur, Hauteur`
- `pygame.sprite.collide_rect` permet de vérifier s'il y a contact entre un `rect` et un autre.
- Permet de créer :
  - des sols,
  - des objets à ramasser,
  - des dangers,
  - ...





# Démo & Questions

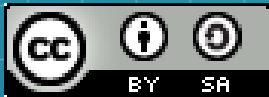
Merci !

Céline Libéral ([celine.liberal@mail.com](mailto:celine.liberal@mail.com))  
Zoé Belleton ([zobelleton@gmail.com](mailto:zobelleton@gmail.com))



La spritesheet présentée aux slides 10 et 11 est dans le domaine public et a été téléchargée sur OpenGameArts (<https://huit.re/oYUKFTEb>)

Le template de ce document est sous licence Creative Commons Attribution-ShareAlike 3.0 Unported License et a été réalisé par Mateus Machado Luna.



Céline Libéral ([celine.liberal@mail.com](mailto:celine.liberal@mail.com))  
Zoé Belleton ([zobelleton@gmail.com](mailto:zobelleton@gmail.com))