

15 JANVIER 2018



SFML

DANS LE CADRE DU COURS DE VEILLE TECHNOLOGIQUE

NICOLAS MAHIEU LA150301

HELHA MONS

Table des matières

- Description 1
 - Général 1
 - Historique 1
 - Fonctionnalités 2
- Domaine d’application 3
- Avantages 3
- Inconvénients 3
- Technologies concurrentes 3
 - SDL 3
 - Allegro 3
 - Godot, Unity, Unreal Engine, Coco 2dx,Game Maker 4
- Code source commenté : 4
- Conclusion 5
- Bibliographie..... 6

Description

Général

Simple and Fast Multimedia Library (SFML) est une librairie multiplateforme permettant d'accéder aux différents composants de l'appareil.

Ecrit en C++, elle peut être utilisée via différents langages, tels que C#, C, Go, Java et encore bien d'autres.

La librairie est disponible sur <https://www.sfml-dev.org/> et est sous licence zlib, en voici les termes :

- La librairie est livrée tel quel, les auteurs ne sont pas responsables des différents dommages qui pourrait découler de son utilisation.
- L'origine du logiciel ne doit pas être falsifié, il est interdit de revendiquer l'écriture du logiciel.
- Les dérivés du logiciel doivent mentionner le logiciel source, il est interdit de revendiquer le logiciel dérivé comme l'original.

Le créateur est Laurent Gomila, et est accompagné d'une équipe (appelé SFML Team).

Cette librairie a été dans le but que les débutants en C++ puissent développer très rapidement un jeu. La librairie est considérée par ses créateurs comme un passeport au développement de jeu vidéo.

Historique

- **.0** (Juillet 2007)
 - **1.1** (18 septembre 2007)
 - **1.2** (16 janvier 2008)
 - **1.3** (22 juin 2008)
 - **1.4** (7 janvier 2009)
 - **1.5** (4 juin 2009)
 - **1.6** (6 avril 2010) : corrections de bogues, les développeurs commençaient déjà l'architecture de la version 2.0
- **2.0** (30 avril 2013) : Refonte complète de la bibliothèque, Laurent Gomila a souhaité une version robuste, stable et mature¹
 - **2.1** (27 juillet 2013) : Correction de bugs ⁴.
 - **2.2** (17 décembre 2014)
 - **2.3** (3 mai 2015)
 - **2.3.1** (27 juin 2015)
 - **2.3.2** (24 août 2015)
 - **2.4.1** (21 juin 2016)
 - **2.4.2** (8 février 2017)

¹ Lu dans la préface du livre SFML Game Development de Packt Publishing

Fonctionnalités

SFML est une librairie multimédia multiplateforme permettant d'accéder aux différents composants de l'appareil.

Il comporte 5 modules distincts :

- Module système
- Module de fenêtrage
- Module graphique
- Module audio
- Module réseau

Le module système se charge des différents threads et de la gestion du temps, il permet également un accès simplifié et rapide aux flux de données.

Le module de fenêtrage permet de gérer la fenêtre ainsi que ses événements. En ce qui concerne les événements ; il s'agit bien des événements sur la fenêtre (redimensionnement, focus, ...) mais également des entrées des périphériques (clavier, souris, joysticks, ...)

Le module graphique est la plus grosse partie de la librairie. Elle permet de dessiner à l'intérieur dans la fenêtre. On peut y dessiner différents éléments (sprites², texte, formes géométriques, ...) ; mais y appliquer également des transformations (rotation, échelle, position, ...).

Il intègre également un système de shader³, qui permet de changer le rendu de l'image dessiné. Ce qui est très utiles pour faire des effets visuels.

Et finalement, il intègre un système de caméra 2D, qui permet d'afficher différentes vues au sein d'une même fenêtre (minicarte, écran scindé, zoom).

Le module audio permet de gérer les flux audios et d'y appliquer des modifications. Il permet :

- De charger et de lire des fichiers audios.
- De modifier la tonalité et le volume du son joué.
- De mettre en pause, se déplacer sur la piste audio.
- De répéter en boucle un son joué.
- D'enregistrer un son
- De faire des flux audios personnalisés.
- De gérer la spatialisation du son joué.

² Le sprite est un élément graphique constitué d'une zone d'affichage, ou on y dessine une texture.

³ Le shader est un petit programme dialoguant directement avec la carte graphique.

Le module réseau permet d'envoyer et de recevoir des données via des sockets (TCP OU UDP).

Il offre sa propre interface pour la manipulation des paquets, ce qui permet :

- D'étendre les paquets pour envoyer différents types de données dans un seul paquet.
- De compresser ou de chiffrer les paquets avant l'envoi.

Le module réseau permet d'envoyer des requêtes http et propose aussi un client FTP.

Domaine d'application

Le principal domaine d'application de la librairie est dans le cadre de création de jeux vidéo, elle a été principalement développée pour ça.

La librairie est couramment utilisée lors de game jam, car elle est simple d'utilisation et utilise très peu de lignes de code pour arriver à un résultat convaincant.

Ensuite, elle est aussi utilisée dans les universités et les écoles dans le cadre de l'apprentissage d'un langage de programmation ou de simulations.

Avantages

Le principale avantage de SFML est sa simplicité, le site officiel est très bien fourni en tutoriels et a une API bien structurée. De plus, la communauté active produits des ressources par elle-même (vidéo, livres, blogs).

Ensuite, elle est multi-plateforme, elle peut être facilement exporté sur divers appareils, tels que Android. De plus, elle supporte plusieurs langages de programmation.

Finalement, la librairie n'est pas que graphique, elle intègre divers outils tels qu'un gestionnaire de paquets, qui permet de créer applications connectées aisément, sans pour autant que cela soit abstrait.

Inconvénients

Le seul inconvénient est qu'il n'est pas possible de faire réellement des jeux vidéo en 3D, sauf si on utilise du raycasting.

Technologies concurrentes

SDL

Le principal concurrent de SFML est SDL (Simple DirectMedia Layer) ; elle est très ancienne (1998), multiplateforme et peut être utilisé avec divers langages.

Elle est très similaire à SFML, cependant, elle est plus compliquée et propose moins de fonctionnalités (gestion de paquets, gestion de thread maison, ...).

Elle a une très forte communauté et est déjà utilisé dans beaucoup de domaines et de projets.

Allegro

Allegro est une librairie open source dédié à la création de jeux vidéo.

Elle est très complète et permet de gérer l'affichage 3D, de plus, elle permet d'aller très bas dans le fonctionnement, ce qui permet faire une bonne optimisation.

Elle était beaucoup utilisé dans le domaine de l'enseignement, car elle est simple et comporte beaucoup d'exemple. Cependant, face à la montée des moteurs de jeux et des autres bibliothèques, sa communauté commence à se restreindre et les mises à jour deviennent de plus en plus tardives.

Godot, Unity, Unreal Engine, Coco 2dx,Game Maker

Tous ces logiciels sont des moteurs de jeux, ils sont utilisés par les professionnels tout comme les particuliers. Ils sont une boîte à outils complète pour produire des jeux vidéos et utilise souvent un langage de script pour animer les interactions. La plupart de ces logiciels sont soit payants, ou demande un pourcentage sur les ventes du jeu vidéo.

Ils sont maintenus par des grosses entreprises, qui animent une communauté autour de leurs produits afin de devenir leader du marché.

Code source commenté :

Le code ci-dessous concerne le Texture Holder, il permet de charger les textures en mémoire pour ensuite qu'elles soient utilisées par différents sprites.

Le TextureHolder est une classe en Singleton et contient une `std::Map`, ce qui permet d'associer un nom à une texture

Ici, le nom de la texture est celui du chemin.

TextureHolder.h

```
///Load texture : TextureHolder::loadTexture(...);
///Get loaded structure : *TextureHolder::getTexture(...);
class TextureHolder {
public:
    ///Singleton instance initializer/getter (we only need our m_Textures loaded once)
    static TextureHolder* getInstance();
    ///amount of m_Textures loaded
    int getCount();
    ///gets texture by name
    sf::Texture& getTexture(const std::string& filePath);
    ///loads a texture
    void loadTexture(const std::string& filePath);
private:
    TextureHolder();
    ~TextureHolder();
    ///Singleton
    static TextureHolder* s_Instance;
    ///Maps m_Textures to their path
    std::map<std::string, sf::Texture> m_Textures;
};
```

Texture.cpp

```
TextureHolder* TextureHolder::s_Instance = nullptr;
TextureHolder::TextureHolder() {
}
TextureHolder::~TextureHolder() {
    delete s_Instance;
    s_Instance = nullptr;
}
TextureHolder* TextureHolder::getInstance() {
    if(!s_Instance)
        s_Instance = new TextureHolder;
    return s_Instance;
}
int TextureHolder::getCount() {
    return m_Textures.size();
}
Texture& TextureHolder::getTexture(const string& filePath) {
    // Check if texture exists
    if(m_Textures.find(filePath) == m_Textures.end())
        loadTexture(filePath);
    return m_Textures[filePath];
}
void TextureHolder::loadTexture(const string& filePath) {
    //m_Textures[name] assigns a new key with <filePath> to <tmpText> if it doesnt exist in the map
    auto& tmpTexture = m_Textures[filePath];
    tmpTexture.loadFromFile(filePath);
}
```

Conclusion

En conclusion, SFML est une librairie permettant d'obtenir des logiciels graphiques de hautes qualités, et ce, très simplement et très rapidement.

Il contient tous les outils nécessaires à la création de programmes divers, mais est surtout destiné aux jeux vidéo.

La librairie permet d'apprendre la création de jeu vidéo ainsi que les différents patterns utilisés dans l'industries. C'est un passport permettant ensuite d'utiliser d'autres librairies sans être perdu.

Bibliographie

RAIMONDAS, Papius. SFML Game Development By Example. Create and develop exciting game from start to finish using SFML. Packt Publishing, 2015.

<https://www.sfml-dev.org>

https://fr.wikipedia.org/wiki/Simple_and_Fast_Multimedia_Library

<https://en.sfml-dev.org/forums/index.php?topic=76.0>