

TP C++ Avancé – Gashamon #1

Lors de ces deux semaines, nous allons réaliser une petite application graphique basée sur les Gashamon en utilisant la SDL.

La SDL est une bibliothèque connue et éprouvée supportant un grand nombre de plateformes et est très éprouvée dans le milieu professionnel.

Néanmoins cette bibliothèque ne dispose que d'une interface C, obligeant la gestion de ressource manuellement (par exemple une fonction retournant un pointeur vers `SDL_Surface` vous demandera à appeler `SDL_FreeSurface` pour libérer celle-ci quand vous n'aurez plus besoin d'elle).

Nous allons donc dans un premier temps réaliser un wrapper à la SDL, pour la rendre « C++-friendly », l'objectif sera de concevoir plusieurs classes englobant correctement les différentes fonctionnalités de la SDL dont nous avons besoin.

En plus de ça, pour préparer la suite, nous allons mettre en place quelques classes :

Classe Sprite

- Le rôle de la classe Sprite est de représenter une partie d'une texture de la SDL (une région de celle-ci, troisième paramètre de `SDL_RenderCopy`) et de fournir une méthode permettant de l'afficher.
- Attention à la façon dont Sprite est lié à la classe Texture, quel outil que vous avez vu cette semaine convient le mieux ?

Classe SpriteManager

- Le rôle de cette classe est d'associer un nom de sprite à un Sprite en question, que nous pourrions ensuite récupérer.
- Associez le nom et le sprite en utilisant une map (par exemple [`std::unordered_map`](#)).
- Faites une méthode permettant d'enregistrer un sprite avec un nom, et une autre permettant de récupérer le sprite à partir du nom
- Faites une méthode permettant d'enlever les sprites qui ne sont pas utilisés par d'autres classes (purge).
- Enregistrez les différents Gashamons sous forme de sprites dans le SpriteManager

Classe Gashamon

- Une classe ne contenant pour l'instant que des informations :
 - Un nom (`std::string`)
 - Un Sprite
 - Des points de vies et points d'attaque (`unsigned int`)
- Pour l'instant un constructeur suffira, cette classe sera utilisée plus tard.

Objectifs (globaux) :

- Ayez une classe (.hpp + .cpp) par objet de la SDL
- À la fin du programme (en cas d'erreur ou non), toutes les ressources de la SDL doivent être automatiquement libérées
- Plus aucune fonction SDL ne doit être appelée directement dans le main.cpp
- Gérez correctement le mouvement pour les objets
- Les structures de la SDL (exemple : `SDL_Color`, `SDL_Rect`) et ses énumérations n'ont pas besoin d'être encapsulées.
- Évitez les copies inutiles

- Évitez les allocations de mémoire inutiles
- Ne faites pas de new/delete directement.
- Réfléchissez bien à ce qui devrait être une référence (constante ou non), ce qui devrait être un pointeur (intelligent ou non), etc.
- Implémentez les constructeurs de copie / mouvement et opérateurs d'assignation par copie / mouvement pour toutes vos classes (quitte à mettre = default/= delete dans le header).

Le rendu devra être fait avant lundi 20h, envoyez le projet au format nom_prenom.zip à l'adresse jerome.leclercq@ext.devinci.fr