

Introduction à SFML

420-V31-SF – Programmation de Jeux Vidéo III

Tout d'abord

- Correction Exercice 1

Clean-up de projet

- S'assurer de l'installation de SFML
- Faire un tour de la documentation de code

Clean-up de projet – TRÈS IMPORTANT

- Pour des remises pas trop massives
 - Enlever le répertoire .vs
 - Enlever .VC.bd
 - Enlever tous répertoires debug et release du projet.

420-V31-SF – Programmation de Jeux Vidéo III

Documentation du code

Documentation du code

- Votre code devrait être toujours correctement documenté.
- Ceci dit en industrie, qu'en est-il?
- Ça va de:
 - - *Les commentaires dans le code ne sont plus à la mode, le code devrait être auto-explicatif*
- À:
 - - *Il faut commenter le code au maximum. Si on peut le faire ligne par ligne alors tant mieux*

Documentation du code

- Votre code devrait être toujours correctement documenté.
- Une bonne règle devrait être (et ce qu'on va utiliser dans ce cours)
- *1 - On commente donc l'entête des classes, les attributs et les méthodes*
- *2 - Les algorithmes les plus complexes devront tout de même être commentés au maximum (pas à pas pour les plus complexes)*

Documentation des entêtes

- Ça sert à:
- Aider le programmeur à comprendre d'un regard à quoi sert un attribut ou une méthode
- Augmenter l'intellisense, si l'environnement de développement el permet (aujourd'hui ils le permettent tout)
- Générer de la documentation en HTML / XML qui devient autonome de l'environnement.

Documentation du code

- Outre une description précise des classes, attributs et méthodes, voici ce qu'on veut ajouter
 - *Paramètres d'entrées*
 - *Valeurs de retour*
 - *Références à d'autres classes/méthodes utilisées*
- La syntaxe est plus large, mais on va à l'essentiel

Documentation du code

- <https://msdn.microsoft.com/en-us/library/5ast78ax.aspx>
- Avouons-le, même si c'est nécessaire et c'est une bonne habitude à prendre, documenter, c'est plate!
- Voici une solution magique
- <http://submain.com/products/ghostdoc.aspx>

On va l'essayer

420-V31-SF – Programmation de Jeux Vidéo III

Présentation de SFML

Le mot de Maître Shigeryu

- Bienvenue, jeune Padawan! Aujourd'hui tu commence une nouvelle étape dans ton apprentissage de développeur de jeux.



La réplique de Darth Newell

- N'écoute pas ce fou! Et vient me rejoindre dans le côté vaporeux de la Force! Avec moi, c'est le mystique 3^{ème} aspect de la demi-vie qui t'attends
- Et... merde non! Non! Non! je ne suis pas Peter Griffin!!!



SFML c'est quoi?

- Wrapper 2D autour de Open GL
- <https://en.wikipedia.org/wiki/OpenGL>
- [https://en.wikipedia.org/wiki/Simple and Fast Multimedia Library](https://en.wikipedia.org/wiki/Simple_and_Fast_Multimedia_Library)
- Comme le dit le nom c'est simple et rapide.

SFML c'est quoi?

- SUPER IMPORTANT
- <http://www.sfml-dev.org/tutorials/2.4/index-fr.php>
- GARDEZ LE LIEN, C'EST UNE TRÈS BONNE RÉFÉRENCE.

Principes généraux de division du code en jeu

- 1- Initialisation du jeu et/ou du niveau et chargement des ressources
- Constructeur vs Init
- Constructeur: ce sans quoi le jeu ne peut pas fonctionner
 - *Si la fenêtre principale ou la vue ne se chargent pas correctement, on a pas de jeu*
- Init: chargement des assets
 - *Si un sprite donné ne se charge pas, on peut toujours essayer de s'en sauver*

Principes généraux de division du code en jeu

- 2 - Boucle de jeu principale (Main Game Loop)
 - *Lecture des entrants (Input reading)*
 - *Mise à jour des données (Update Data)*
 - *Affichage à l'écran (Draw Screen)*

Principes généraux de division du code en jeu

- 3- Fermeture du jeu et/ou du niveau et libération des ressources
 - *Effacer les pointeurs*
 - *Fermer les liens réseaux ou BD existant*
 - *Sauvegardes*
 - *Etc.*
- Méthode unload; on en a pas dans les projets présentés jusqu'ici, mais pourrait s'ajouter facilement.

Principes généraux de division du code en jeu

- NOTEZ BIEN

- La division en

- init / boucle principale (input-update-draw) / unload

-

tel que présenté dans les template fourni est un bonne méthode de conception pour améliorer la lisibilité et la maintenance du code, mais en théorie, tout pourrait aller dans un immense "main"

420-V31-SF – Programmation de Jeux Vidéo III

Boucle de jeu principale

Principes généraux de division du code en jeu

- Rappel: Boucle de jeu principale (Main Game Loop)
 - *Lecture des entrants (Input reading)*
 - *Mise à jour des données (Update Data)*
 - *Affichage à l'écran (Draw Screen)*
- Attention, ceci est un patron de conception (Design Pattern)
- En résumé, une recette, une structure de code efficace et éprouvée pour arriver à un résultat (nous y reviendrons...)

Lecture des entrants

- La lecture des entrants et la mise à jour des données tendent parfois à être confondues
- Par contre il faudra les séparer correctement
 - *Quitte à mettre un filtre sur la saisie d'entrants*
- Pourquoi les séparer?

Lecture des entrants

Parce que:

- Qualité et clarté du code.
- Masque de commande: niveau où le joueur peut déterminer quels contrôles ont quelles fonctions
 - *Changement de contrôles / boutons dans un jeu*

Affichage à l'écran

- Ici, on ne fait que strictement afficher à l'écran
- Seules opérations permises: conditionnelles simples qui détermine si on affiche tel graphique ou non.

Mise à jour des données

- Fait pratiquement tout le reste.
- A parfois une sous division qui s'occupe d'échanger avec les ressources de persistances (fichier, BD, cloud, etc.)

Un mot sur les techniques d'affichage

- TRÈS IMPORTANT:
- Le plus possible: jeu indépendant de la résolution!
- Mise à l'échelle par la suite de ce qui est mis à l'écran.
- Plus le jeu est indépendant de son affichage, plus il est portable d'une plate-forme à l'autre.

(Modèles exemplaires récents... Street Fighter IV, Super Smash Brothers).

Un mot sur les techniques d'affichage

- Par le passé: plusieurs sets de sprites.
- Aujourd'hui: 3D La vue de la caméra va "créer la scène".
- Pour le 2D: Résolutions "Standard" et on étire ou on compresse (demande du jus, peu importe lequel).
- Ou alors c'est l'écran qui est ajusté. (SFML ajuste l'écran, mais visez des résolutions standards ET répandus – 800 x 600 c'est encore un standard, mais ce n'est plus très répandu. 3840 x 2160 n'est pas encore très répandu... mais l'est de plus en plus)

420-V31-SF – Programmation de Jeux Vidéo III

Exemples SFML

Une console dans SMFL???

- Avant tout: pourquoi la console?
- Très pratique pour le débogage: permet d'afficher des infos en même temps qu'on joue
- Vieux jeux PC sur Windows, cette console "trainait souvent"
- Si vous souhaitez l'enlever:
 - *Clic-droit sur le projet-> Propriétés de configuration->Éditeur de liens->Système->Sous-système*
 - *Changez Console pour Windows*

Les exemples

- On se lance
- Regarder le code des exemples: beaucoup d'infos qu'on y trouve pourrait être sujet d'examen.
- Soyez attentif.
- Prenez des notes.

Exercice 2

- Le principe: Tout ce que vous allez faire pourrait servir à un TP ou à un prototype.
- Donc gardez et étudiez ce que vous aller faire.
- On commence à se rapprocher que ce que vous souhaitez faire... à vous d'en prendre acte.
- Si ça ne vous tente pas... posez-vous des questions...