

Compte rendu projet C++

There is no Planet B



Introduction	2
Description de l'application	2
Commande	2
Diagramme UML de l'application	3
Procédure d'installation & exécution du code	4
Parties de l'implémentation dont nous sommes les plus fiers	5
Conclusion	6

Introduction

Au cours de notre deuxième année en parcours d'Electronique d'Informatique et de Système Embarqué, nous avons fait face aux nouvelles problématiques et possibilités des langages de programmation orienté objet, notamment le C++.

A travers un certain nombre de TP et de cours nous avons appris à utiliser ce langage qui est primordial dans le monde de l'informatique. Pour nous mettre à l'épreuve et ainsi tester nos connaissances dans ce langage, un sujet de projet nous a été donné il y a un mois, en décembre 2021. Si tous nos Travaux Pratiques étaient encadrés et avec un sujet bien précis, ce n'est cependant pas le cas de notre projet. En effet, un sujet très vaste nous a été donné: There is no planet B, ce qui a pu nous laisser l'embarras du choix pour programmer quelque chose. Nous aurions pu programmer une application ou d'autres mais nous avons choisi de faire un jeu vidéo dans le thème imposé qui était l'écologie.

Description de l'application

Au cours de votre expérience sur notre jeu, vous incarnerez un personnage ayant une tâche principale: rendre le monde meilleur. Voyager et admirer les décors que vous allez croiser tous inspirés de la réalité, mais gare à vous, votre voyage ne sera pas de tout repos. Affronter les pollueurs qui polluent en leur lançant un duel ! Faites une bonne action pour la planète et plantez des arbres, ils vous permettront également de regagner de la vie, que vous auriez pu perdre lors d'un duel ou autre

Vous aurez donc 3 objectifs:

- Planter les 31 arbres dispersé sur la map
- Faire votre discours devant la caméra
- Dernier objectif à découvrir !

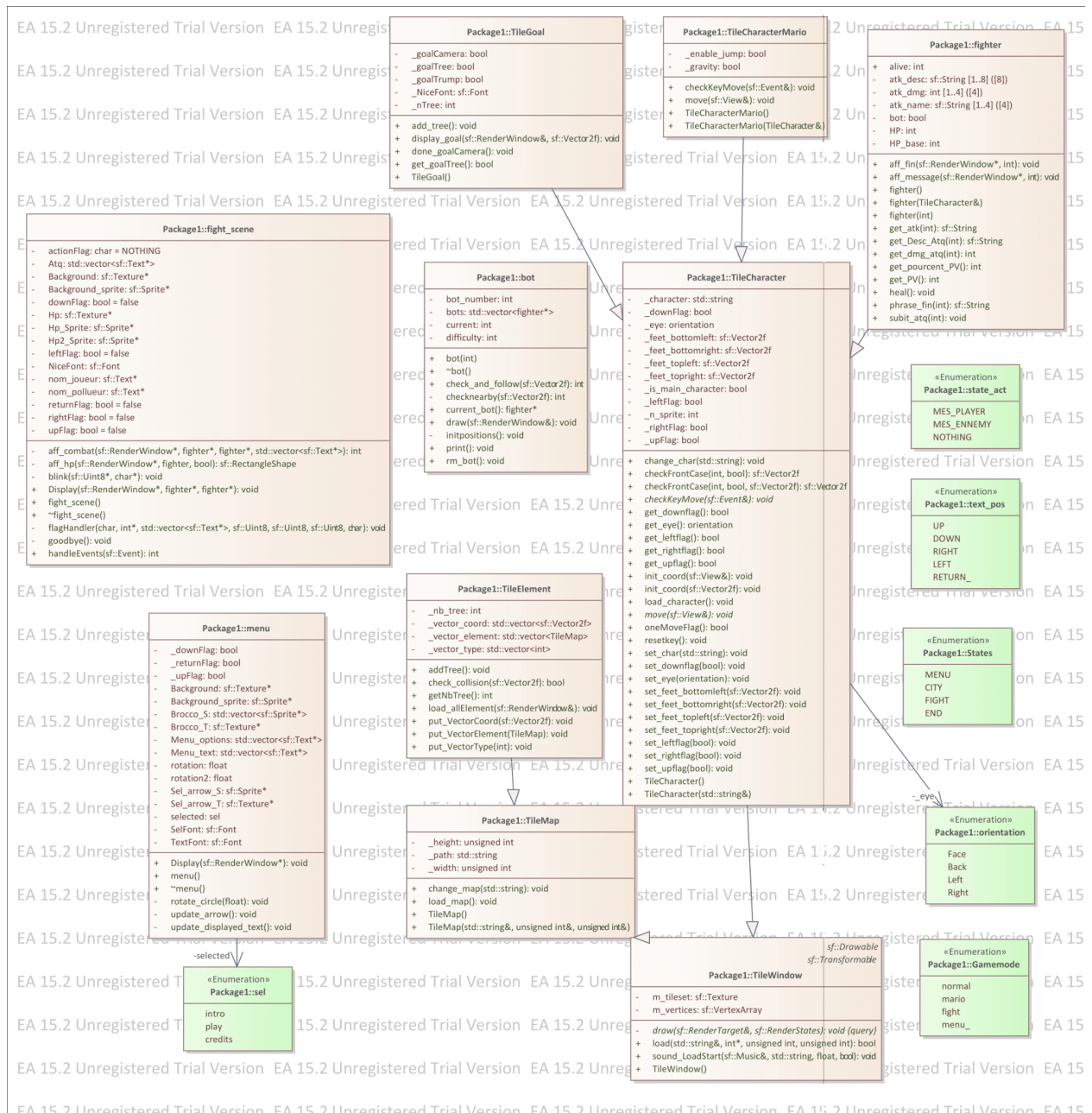
De plus, il y a de nombreux easter eggs dans le jeu qu'il sera mieux de découvrir par soi-même que de le lire dans le code, dont deux event spéciaux cachés ! Gare à la triche et à ne pas ouvrir Obstacle.cpp ni les images de la map pour découvrir toute la beauté de notre jeu.

Commande

Le jeu se joue uniquement avec les touches directionnelles ainsi que la barre d'espace qui permet d'effectuer les actions dans le jeu. Vous pourrez bien entendu à tout moment appuyer sur la touche "échap" afin de quitter le jeu.

Diagramme UML de l'application

Voici notre diagramme UML:



(voir le fichier diagramme.png pour plus visibilité)

Ce diagramme a été fait sur Enterprise Architect en version d'essai ce qui explique malheureusement le filigrane.

Vous pourrez remarquer une hiérarchie sur 3 niveaux, la classe TileWindow qui est la mère de toutes presque dans notre projet, puis la TileMap (pour la gestion des éléments de la map) ainsi que le TileCharacter (pour afficher et déplacer tous les personnages du jeu). Ensuite nous avons les classes fighter (pour gérer les combats, points de vie, dégâts...), TileGoal pour faire le HUD du joueur pour afficher les objectifs.

La classe bot permet de gérer les différents bots dans le jeu, elle contient un vecteur de personnages ainsi que toute la gestion des coordonnées de proximité pour la détection.

Les classes `menu` et `fight_scene` sont toutes deux des classes qui n'ont vraiment rien à voir avec les autres: elles servent à l'affichage de différentes scènes.

Nous les avons d'abord conçus afin qu'ils fonctionnent sur des threads (pour afficher le menu quand on est en jeu par exemple), cependant avec une perte de performance assez conséquente, nous avons décidé de ne pas les utiliser sur des threads mais dans la boucle principale de l'application.

Procédure d'installation & exécution du code

Installer notre jeu et pouvoir y jouer est très aisé! En effet nous utilisons une librairie très portable si vous souhaitez build le jeu depuis la source, sinon vous pourrez prendre la release sur github précompilée!

Pour build depuis la source:

Il faut installer SFML:

```
sudo apt-get install libsFML-dev
```

Il faut maintenant télécharger notre code:

```
git clone git@github.com:PierrrelouisL/reDesigned-Octo-Broccoli.git
```

Pour build notre application:

Il suffit de lancer le Makefile et le tour est joué ! Lancer notre jeu en activant le son pour avoir une meilleure expérience !

Parties de l'implémentation dont nous sommes les plus fiers

Le plus important pour nous :

Nous nous sommes fortement amusés à confectionner de A à Z notre jeu avec toutes les ficelles qui en découlent. Les sprites et le son étaient un véritable plaisir à créer et à manipuler. De plus, manipuler une librairie telle que SFML nous a permis de mieux comprendre les possibilités et les concepts en C++.

La chasse au easters eggs est aussi un élément de décors et de jeu très important pour nous à la conception du jeu. Un jeu sans easter eggs manque de saveur et nous le savons ! C'est pourquoi penser et mettre en œuvre des easter eggs était primordial pour nous.

Tous les les pixels arts ont été fait à la main sur le logiciel ASEPRITE, le travail a été long et laborieux mais nous sommes fiers du résultat.

[SPOILER]

Pour parler un peu du code que nous avons produit, nous avons fortement aimé la façon de passer du mode normal au mode mario. En effet, nous avons réussi à utiliser la même boucle qui permet de gérer les actions et le monde autour de lui pour les deux modes grâce à la fonction virtuelle. De ce fait, le code est beaucoup plus léger et on peut interchanger facilement entre les deux modes de jeu où la physique n'est pas la même. En mode normal, nous avons une 2D avec une profondeur et en mode mario seulement une 2D avec de la gravité.

[SPOILER][SPOILER]

Si vous voulez visiter la map avec une meilleure vue, allez donc chercher le glitch de la ville!

Conclusion

Ce projet en C++ à été très amusant et intéressant pour notre apprentissage. Certains concepts vu en cours pouvaient être un peu flou ou pas 100% acquis mais grâce à ce projet qui nous a poussé à utiliser tous ces concepts, nous en avons donc beaucoup appris.

De plus, nous savions que ce projet pourrait être très long, nous avons alors bien mieux gérer et anticiper nos actions en répartissant les tâches et en commençant très tôt.

Nous n'avons malheureusement pas pu implémenter tout ce que l'on voulait. Nous avons tout d'abord voulu avoir plusieurs personnages jouables, qui seraient tous des figures historiques qui ont participé à une révolution écologique, ou encore avoir plus de boss dans le jeu, les PDG de grandes entreprises participant au réchauffement climatique par exemple. Ensuite, nous n'avons également pas pu implémenter des combats plus dynamiques avec des animations et des effets sonores.

Si vous avez aimé notre jeu, nous pourrions développer en V2 comprenant toutes nos idées non implémentées !