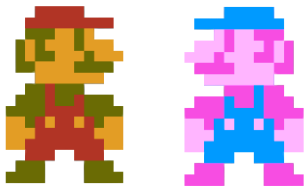
HMIN317 : Moteur de jeux

Documentation : Moteur de jeux UnCommon



Préparé par

Iradukunda Valentin

Janvier 2020

INTRODUCTION

Pourquoi le nom **UnCommon ?**

Il est tiré d'un livre appelé **CAN'T HURT ME** de David Goggins qui parlent du fais que on doit toujours chercher à être the « THE MOST UNCOMMON AMONGST THE UNCOMMON « ou le plus rare parmi les plus rares.

Au début de ce projet, je voulais que cet engine soit unique d'où le nom **unCommon**

Crédits

la plupart de fonctionnalités réalisé dans ce Game Engine ont été faite en suivants des tutoriels réalisé par **Gabriel d'Ambrosio** qui possède une chaine YOUTUBE appelé **GAMESWITHGABE .**

Il a m'a beaucoup tout au long du projet, je lui posé des question hyper stressant alors que c'est un étudiant comme moi il se donne la peine de me répondre que ça soit sur discord ou directement sur YOUTUBE

Et je me suis servis de certains livre mise à notre disposition Par **Mme Faraj**, qui est la responsable de cette UE, qui n'a pas cesser de nous suivre et nous accompagner malgré les conditions actuel lié à la crise sanitaire

Langage de Programmation

Ce game engine est écrit en langage JAVA et non en C++ comme la plus part des game engines ou simplement des logiciels où on a besoin une accès rapide à la mémoire pour optimiser le rendu, j'ai choisi de le développer en java pour les raisons suivantes :

- Je suis plus à l'aise en JAVA qu'en c++
- La plupart de tutoriel qui traiter différentes fonctionnalités que je souhaitais implémenter était en JAVA.
- La possibilité de pouvoir utiliser la librairie OpenGL même en JAVA.
- Le fait que je n'aime pas

Ces trois raisons qui on fait que je décide d'écrire ce moteur en JAVA malgré que les tps sont en c++ .

Outils et Librairie utilisés

- Le développement de ce moteur a été réaliser grâce à L'IDE **INTELLIJ en** utilisant l'outil de build des projets Automatisé pour JAVA **GRADLE**

- Les interfaces ont été réaliser grâces à ImGui qui est une librairie Open source qui permet de réaliser des interfaces utilisateurs de manières rapides et simple

- L'implémentation de la physique a été faite en intégrant un moteur de simulation open source **BOX2D**

- Et le rendu comme indiqué précédemment a été faite grâce à OpenGL

Les fonctionnalités du moteur

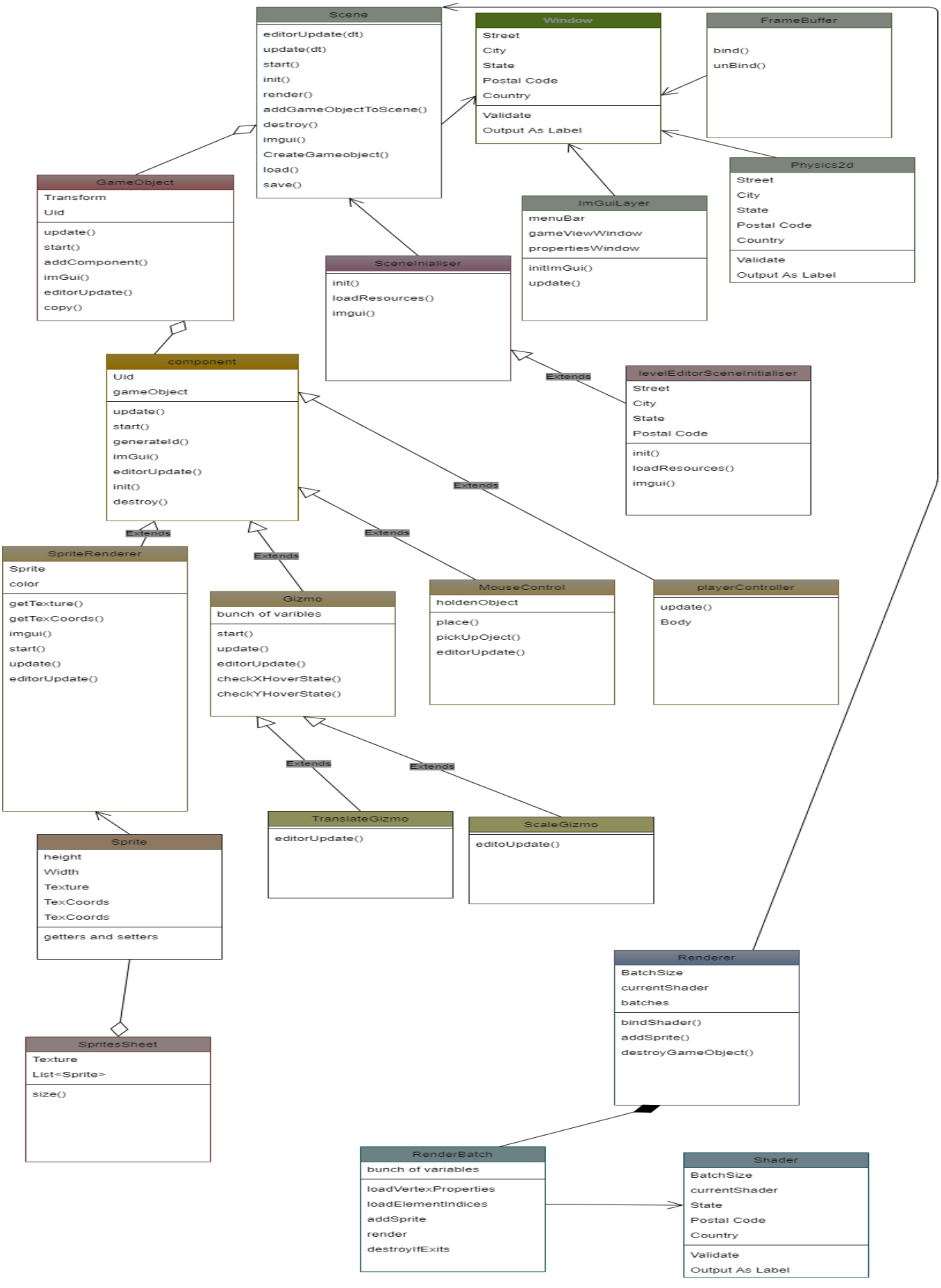
Ce moteur de Jeu en 2D permettra une fois terminer , la création des jeux en 2D, à l'état actuel du moteur on peut essayer de gribouiller quelque chose qui ressemble à un jeu mais c'est ne pas encore trop ça, mais malgré ça le moteur présentent des fonctionnalités qu'on retrouve dans d'autres moteur de jeux connus., il permet

- L'ajout ,duplication et suppression de gameObjects
- La translation
- La mise à l'échelle
- La rotation
- Le moteur possède un Système de Gizmo pour faciliter les opérations précédentes(déplacer ,modifier les games objects)
- Le moteur permet l'Edition des niveaux en attribuant des sprites aux quads du maillage de la scène
- Le moteur possède une caméra avec des fonctionnalités de pouvoir Zoomer et de Zoomer
- Le moteur possède un moyen de pouvoir ramener la camera au centre du monde avec une interpolation linéaire fluide
- Le moteur permet de faire des animations à l'aide des Sprite Sheets et dirtyFlags
- Le moteur comporte un moteur physique pour détecter les collisions et faire bouger les objets grâce aux forces ou impulsion
- Le moteur possède un moyen de sauvegarder l'état actuel de la scène grâce à la sérialisation
  En utilisant **GSON** qui est une bibliothèque Java open source pour sérialiser et désérialiser des objets Java en **JSON**.

- le moteur permet le Z index
- il permet une ajout de component sur les objets faciles
  par exemple pour ajouter une fonctionnalités à un game objet il suffit juste de créer une classe d'étendre la classe **Component** et de rajouter ton nouveau composant à ton game Object tout simplement

Tout ces fonctionnalités peuvent être améliorer ou mieux architecturer

Pseudo Diagramme de classe du Moteur



Le Jeu MARIO

A la base l'idée principale était de faire un moteur de jeu 2D et après réaliser le jeu super Mario like à la fin, l'objectif finale n'a pas était atteint ,étant donné différentes fonctionnalités qui manque au moteur mais au moins on a Un Mario qui bouge et saute dans toute les sens , pouvant interagir avec différents objets (ou pseudo ennemie)du décors ,

Amélioration

Comme j'ai le préciser au début ce moteur manque différentes fonctionnalités utiles pour pouvoir faire un jeu complet et intéressent, et pourrait être mise à jour dans le futur.

Les différentes fonctionnalités qui lui manquent sont ;

-les rendu du texte pour pouvoir réaliser des menus,

-l'éclairage

-le son

-un système de particule

-un système de lancer de rayon

Conclusion

La réalisation de ce moteur n'a pas étais facile, ça a était une période pleine d'angoisse ,des bugs, de stresses mais à la fin je suis content d'avoir au moins quelque chose à présenter. Niveau apprentissage ça était une exercice très enrichissante et si c'était à refaire je signerai .