

Rapport du moteur de jeux

Méténorage

Noémy Artigouha - Matthieu Le Boucher - Grégoire Boiron - Florian Vidal

Lien du projet : <https://github.com/Meight/Metenorage>

I - Architecture et caractéristiques du moteur	1
II - Jeu de démonstration	1
III - Editeur	2
IV - Post-mortem	4

A noter : Nous n'avons pas réussi à générer l'exécutable à cause des fichiers scripts qui sont dans un package en dehors du dossier source. Si vous souhaitez tester le jeu, lancer le main de *FindYourWay* et si vous souhaitez lancer l'éditeur, lancer le main de *Editor*. Nous vous prions de nous excuser pour ce problème.

I - Architecture et caractéristiques du moteur

Nous avons développé le moteur en Java avec l'IDE IntelliJ. Nous nous sommes servis de Maven et GitHub pour la gestion de versions et de développement de notre moteur.

Nous avons utilisé la bibliothèque *Lightweight Java Game Library* (LWJGL) afin d'avoir accès aux APIs natives comme OpenGL (graphique) et OpenAL (audio). Pour la partie physique de notre moteur nous avons utilisé *JBullet*.

Le moteur est développé selon une architecture entités-composants-systèmes, ce qui lui fournit une très grande modularité. Tout au long du projet, il a été très simple d'apporter des améliorations car il suffisait au choix de créer un nouveau composant ou un nouveau système. Nous avons développés différents composants ajoutables à des entités comme : un rigidbody, de l'audio et des scripts.

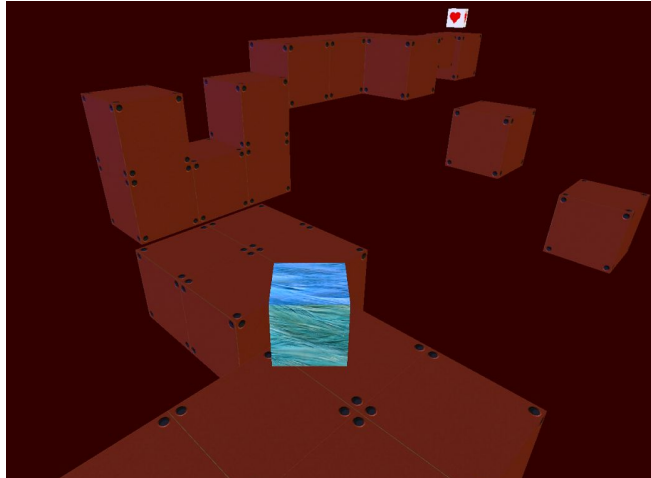
II - Jeu de démonstration

Le jeu que nous avons développé avec notre moteur est **FindYourWay**. Vous incarnez un petit cube qui doit aller d'un point A à un point B en se déplaçant et en sautant.

Contrôles :

- WASD (ZQSD) : déplacement du cube
- QE (AE) : orientation horizontale du cube
- BARRE D'ESPACE : faire sauter le cube

Des scripts sont accrochés à différentes entités du jeu pour ajouter des particularités et des règles. Ainsi, l'entité "caméra" possède par exemple un script pour lui faire suivre l'entité "joueur" et ainsi avoir une vue à la 3ème personne. Le joueur dispose de 3 vies ; il en perd une à chaque fois qu'il tombe. Si jamais il épuise toutes ses vies, la partie est finie et il doit en relancer une autre. Un bonus vie est récupérable dans le niveau. Quand le joueur arrive sur le palier final, il a gagné et les contrôles du joueur se bloquent puisqu'il a fini le jeu et n'a plus rien à faire dessus.

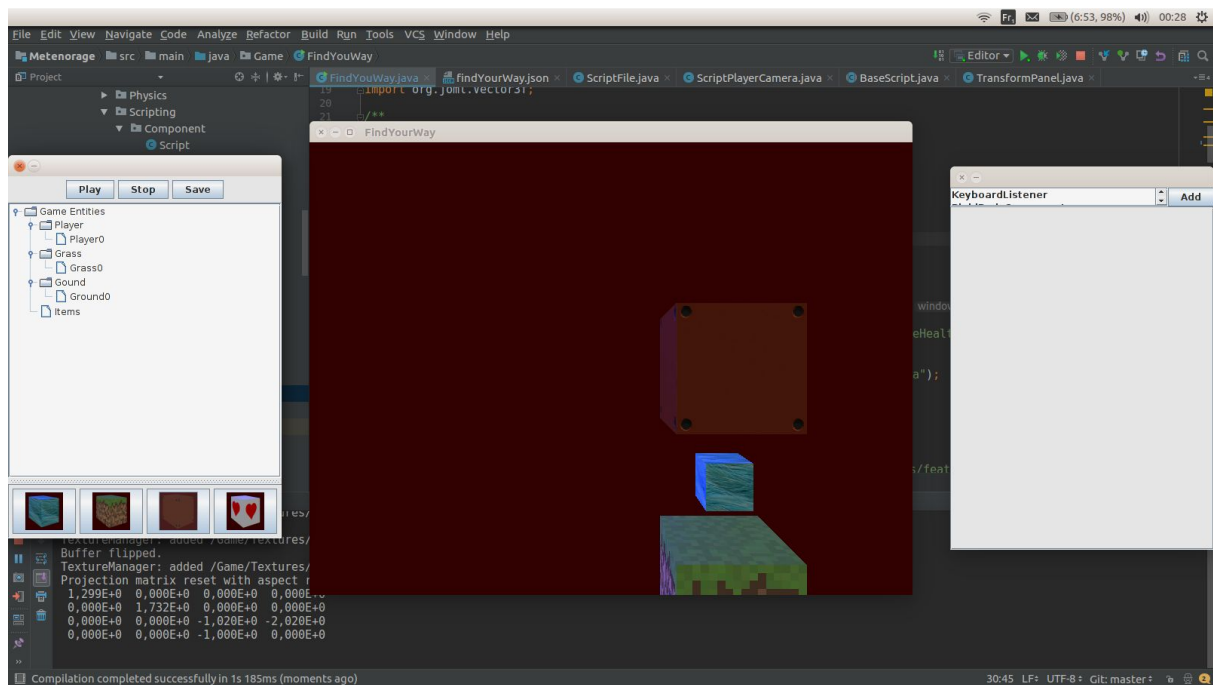


Aperçu de FindYourWay

III - Editeur

Une fois l'éditeur lancé, il est possible d'ajouter des blocs au jeu parmi les quatre disponibles :

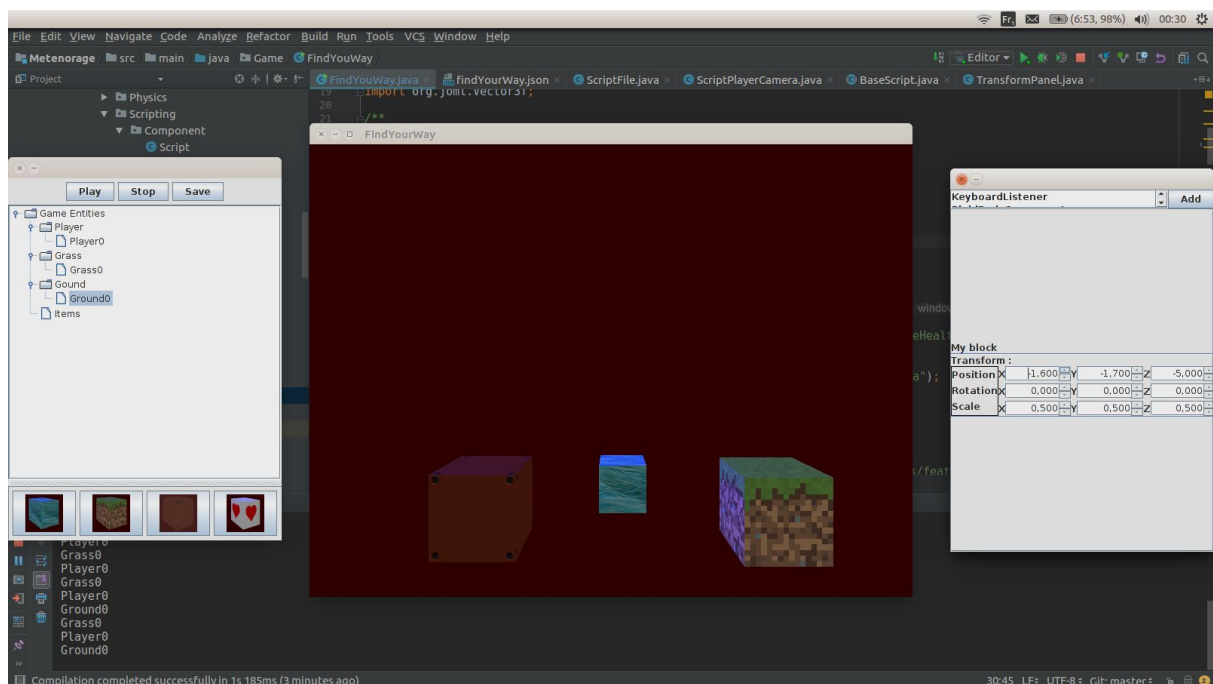
- Un bloc bleu (joueur)
- Un bloc terre (grass)
- Un bloc basique (bloc)
- Un item de vie (coeur)



Editeur avec la création des trois objets au départ.

Une fois l'item ajouté, il apparaîtra dans la hiérarchie et il sera possible de modifier son attribut transforme en cliquant dessus dans la hiérarchie du panel de droite et en modifiant sa transforme dans le panel de droite.

A noter que tous les éléments se créent mais qu'ils se positionnent à des positions "aléatoires". L'item de soin par exemple se positionne à l'extérieur de l'écran.

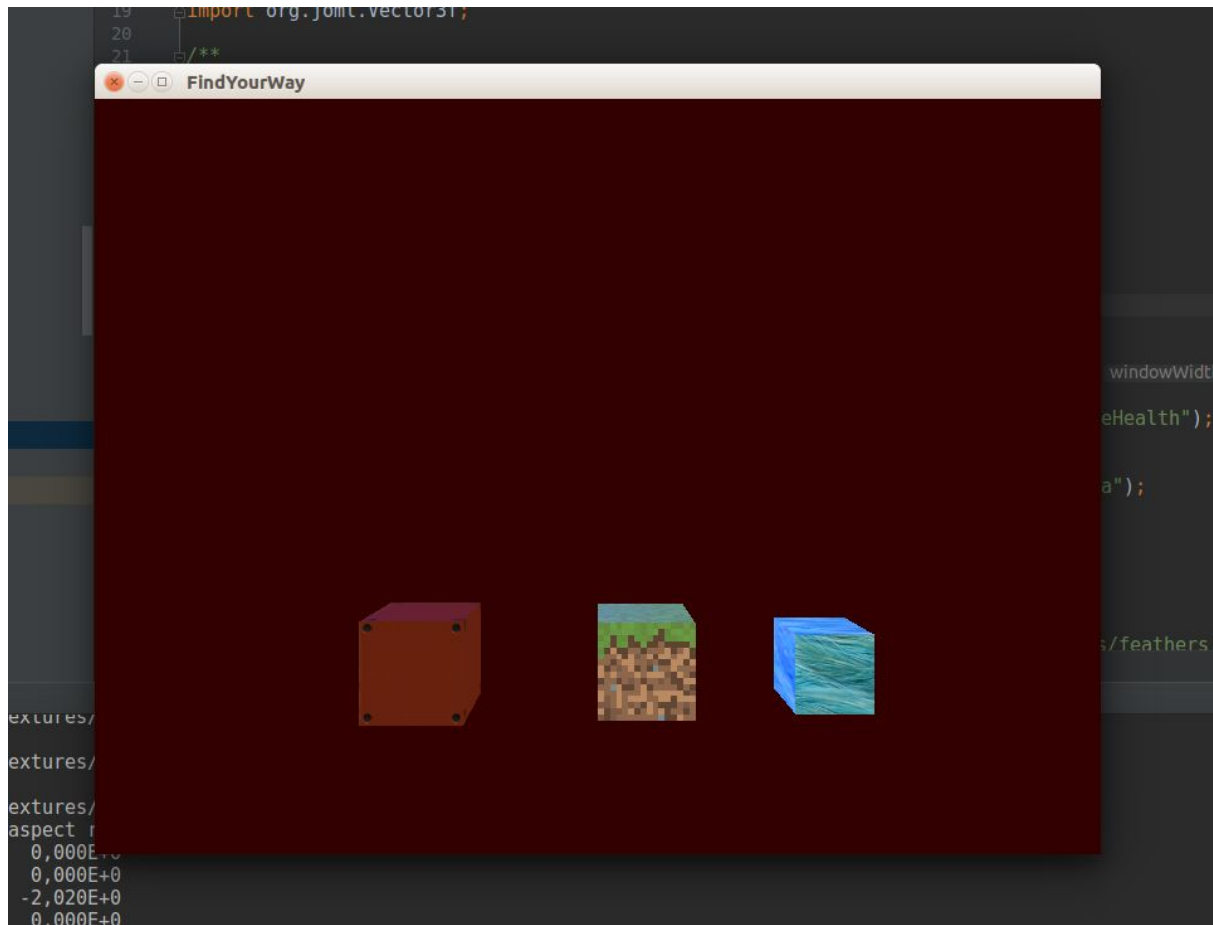


Editeur avec les mêmes trois objets mais avec repositionnement.

Une fois que l'on a obtenu le jeu souhaité, il est possible de l'exporter en format .json pour le charger par la suite par le moteur. Pour exporter le jeu, il suffit de cliquer sur le bouton "Save" du premier panel de gauche.

Une fois le bouton cliqué, un fichier "gameEditor.json" va se créer dans l'arborescence (resources/Game/gameEditor.json) qui représentera les entités du jeu sous forme de données.

Il est possible ensuite de charger ce fichier directement ce fichier dans le main "FindYourWay". Pour se faire, il suffit de modifier, dans le code, la variable booléen "testParser" pour la mettre à "true". Avec cette modification, le jeu qui provient de l'éditeur devrait être joué.



Exemple de jeu lancé, initialement créé par l'éditeur. Aucun des composants ne possède de Rigidbody, car cette fonctionnalité n'est pas encore prise en charge par l'éditeur.

IV - Post-mortem

Si ce projet venait à se continuer nous ferions quelques ajustements pour encore améliorer les fonctionnalités du moteur :

- Plus d'entités différentes disponibles : cercle, polygones
- Afficher du texte à l'écran
- Améliorer l'éditeur

Pour ce qui est de l'organisation de l'équipe, nous aurions dû mieux nous organiser en nous répartissant au préalable les tâches et en nous mettant d'accord sur la façon d'implémenter le moteur. Il a été assez dur de rentrer dans le code des autres car nous ne savions pas comment ces derniers avaient été implémentés.

Nous sommes cependant satisfait de ce que nous avons réussi à développer avec le temps dont nous disposions. Si c'était à refaire nous demanderions juste plus de temps afin de pouvoir nous réunir pour réfléchir aux développements des différents éléments et avoir le temps de rendre le moteur vraiment complet.