

Documentation TCHANZ 296190_331471

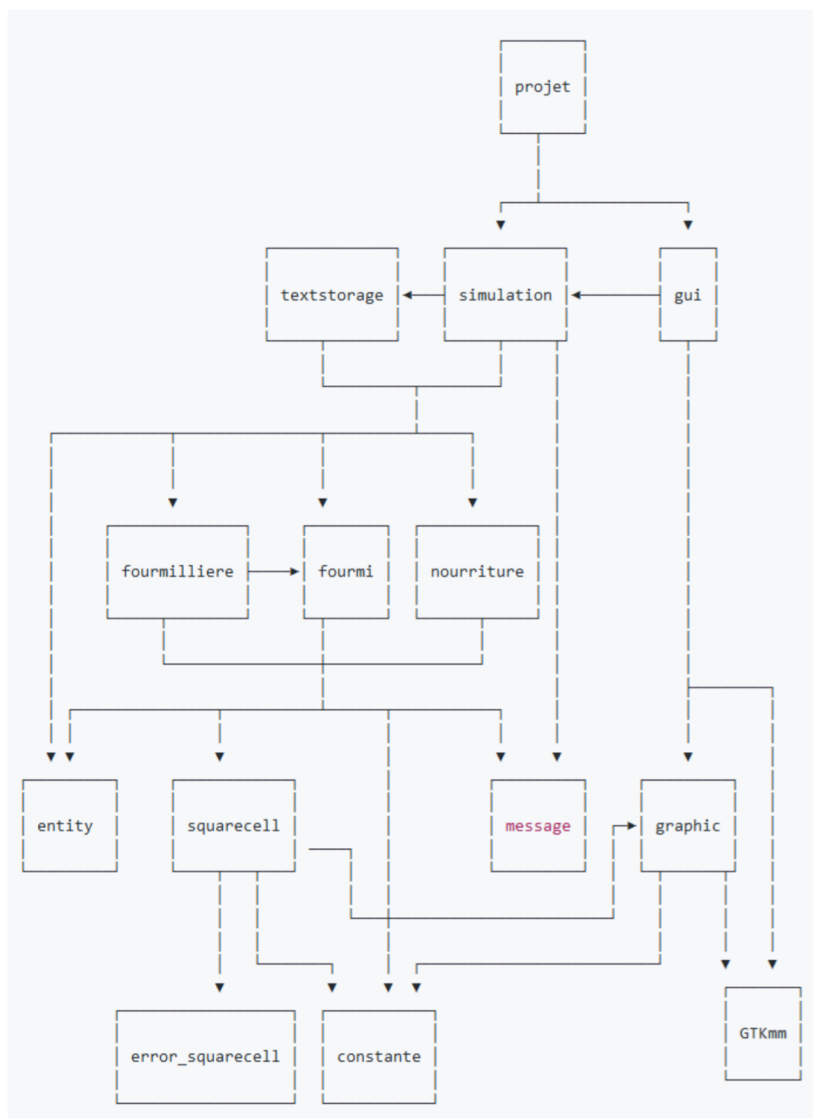
Documentation du fonctionnement général de TCHANZ

organisation des classes du modèles

La hiérarchie de classe du modèle et des fourmis est organisé comme suit :

un module **entity.h** fut créé pour profiter pleinement du mécanisme de polymorphisme. Fourmillière, fourmi et nourriture en hérite car ils possèdent tous une forme et une position ainsi que le besoin d'être affiché à l'écran et d'être update à chaque tick de la simulation. Entity fournit donc des méthodes virtuelles que chaque entité redéfinit pour être dessinées ainsi qu'update et exporter ses paramètres. Une série de fonctions "normales" fournissent un accès aux propriétés commune tel que la position, la taille ou le flag qui indiquent si l'entité doit être détruite.

Fourmi.h possède une classe fourmi générique dans lesquelles sont regroupées les propriétés communes à toutes les fourmis tel que l'âge et un buffer contenant les prochains pas à effectuer. (La plus part des fonctions plus intéressantes étant déjà dans entity, il n'y a pas cas de les redéfinir ici)



structuration des données des entités du modèles

Voici la liste des attributs et fonction principale disponibles dans chaque classe:

- Fourmilliere contient le nombre de chacune de ses fourmis, ainsi qu'un vecteur de pointeur vers ses fourmis membre. En plus des accesseurs et des méthodes virtual héritées d'entity et de la méthode qui permet d'importer une fourmillière à partir de ces paramètres. La méthode update va appeler update sur chaque fourmi membre. De plus elle contient des méthodes pour générer de nouvelles fourmis au hasard et l'expansion de la fourmillière.
- Nourriture ne contient aucune propriété supplémentaire et se contente simplement de redéfinir les fonctions de dessin et d'export dont elle hérite ainsi qu'une fonction pour créer une instance de nourriture depuis un fichier externe.
- Entity contient un champ squarecell qui représente la surface occupé par l'entité. De plus entity stock un booléen endOfLife qui indique si une instance doit être détruite et un id qui permet de représenter l'appartenance à un groupe ou l'ordre d'apparition dans le fichier. Entity stock aussi le RNG en static et specie qui encode l'espèce de l'entity (utile pour caster une entity vers sont type plus spécifique).
- Simulation contient un vecteur de pointeur vers les nourritures présentes sur le plateau ainsi que les fourmillières. Il dispose d'une interface pour faire simuler un tick au modèle et de retourner des statistiques sur celui-ci (nombre de fourmillière, nourriture, etc) ainsi qu'une méthode pour importer une simulation depuis un fichier et exporter la simulation courante vers un autre fichier.

Le choix de déléguer l'importation et l'exportation à un module spécifique a été fait pour rendre l'exportation plus générique (si on décide d'exporter dans un type de fichier ou dans le cloud par exemple).

représentation des entités dans simulation

Comme indiqué précédemment simulation contient un vecteur de pointeur vers la nourriture et les fourmillières. Les fourmillières contiennent un vecteur de pointeurs vers leurs fourmis respectives. Fourmillière fournit une fonction qui retourne la liste de ses fourmis. Le but étant que l'intégralité des entités du modèle puissent être passées en argument à la méthode update de chaque entity qui pourra prendre des décisions basées sur les autres entités.

Brève description des types mis en œuvre dans squarecell :

Squarecell contient un point qui indique la position d'une entité ainsi qu'un flag booléen qui précise si la position est au centre ou dans le coin inférieur gauche. De plus squarecell contient la largeur et la hauteur de chaque entité et la position de son coin inférieur gauche et supérieur droit (ce qui permet de s'affranchir de la problématique de l'origine centrée ou non).

Squarecell contient aussi un char qui représente le calque sur lequel un carré se trouve pour rendre plus flexible les calculs de collision.

Finalement squarcell contient un vecteur de char static qui représente la grille pour vérifier à moindre coût si une région donnée est libre ou non.