

Optimisation de l'évacuation en cas d'urgence

11 juin 2017

1 Étude des foules et évacuation

L'étude du comportement des foules est un domaine important pour améliorer la sécurité lors des événements publics ainsi que dans les bâtiments publics. Mais l'étude de ces foules par l'expérience coûte chère car il est nécessaire de mobiliser un grand nombre d'individus, lors d'un simple festival de musique il peut y avoir plus de 7000 personnes. De plus les personnes prenant part aux expériences ne seront pas dans les conditions réelles, ils ne seront pas paniqués par exemple, l'expérience peut alors produire des résultats erronés.

La conception de bonnes expériences permettant l'étude des foules étant difficile, l'étude se fait principalement grâce à des simulations se basant sur des vidéos et des descriptions d'événements réels ainsi que de résultats d'expériences.

2 Objectifs

L'objectif du TIPE de mon tronc fut de développer un programme permettant d'étudier le mouvement d'une foule en cas d'évacuation dans différentes situations. Puis d'en déduire des optimisations possibles pour l'évacuation, particulièrement des optimisations concernant le plan d'évacuation et la disposition des obstacles.

Le TIPE s'est découpé en deux simulations à des échelles différentes ainsi qu'à une étude des résultats produits par les simulations, une des simulations se base sur un graphe à flux variant et est à l'échelle d'un bâtiment, l'autre simulation se base sur les simulations multi-agents et est à l'échelle d'une salle.

Je me suis concentré sur la modélisation du mouvement des agents dans la simulation multi-agents.

3 Simulation multi-agents

Dans le domaine de la modélisation des foules les simulations multi-agents sont les simulations les plus présentes. Contrairement aux modélisations se basant sur des automates cellulaires et celles se basant sur le mouvement de particules qui modélisent le mouvement de la foule dans son ensemble, les modélisations

multi-agents modélise chaque agents individuellement le comportement de la foule étant un résultat du comportement de chacun de ces agents.

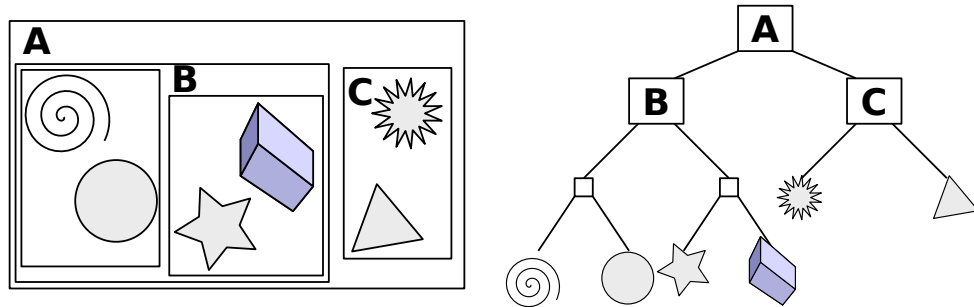
4 Outils utilisés

Nous avons codé les simulations en Python car c'est le seule langage connue par l'ensemble de notre trinôme, de plus Python viens avec un grand nombre de modules permettant de faciliter le développement de notre simulation.

Nous avons utilisé un module de la librairie standard pour manipul  des fichiers en JSON que nous avons utilis  comme fichiers de configurations pour nos simulations permettant ainsi de facilit  l' tude de diff rentes situations.

Nous avons aussi utilis  le moteur physique en deux dimensions Pymunk, qui est une enveloppe autour du moteur physique Chipmunk cod  en C, pour  viter de passer trop de notre temps voire tout notre temps sur la mod lisation physique des obstacles et des agents. Nous avons utilis  en plus de Pymunk la librairie graphique Pygame pour afficher la simulation.

La structure de donn e utiliser par Pymunk pour ses calculs est une hi rarchie de volumes englobants, une structure couramment utilis  pour la d tection de collisions dans un espace, c'est un arbre binaire dont chaque n uds est un volume qui englobe les volumes des n uds enfants, les feuilles de cette arbre sont les objets de l'espace Figure 1. Ainsi pour chercher les objets intersectant un objet donn e il suffit de rechercher r cursivement dans les sous arbres dont le volume englobant racine intersecte l'objet donn e. Pymunk utilise des rectangles align  aux axes des abscisses et des ordonn es couramment nomm  AABB – axis-aligned bounding box – comme volume englobant. Pour avoir des recherches efficaces Pymunk  quilibre l'arbre en minimisant la surface occup  par chaque n uds.



5 Description d'une salle

Les salles dans lesquelles sont étudiées le mouvement sont délimité par une AABB représentant les murs. Des trous dans les murs représentent les différentes sorties. Figure 2

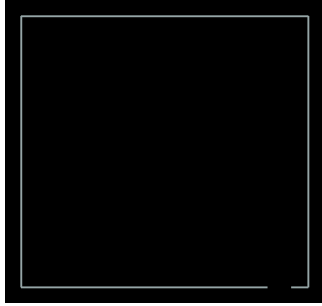


FIGURE 2 – Un exemple de salle, cette salle a une sortie en bas à droite

Les obstacles présent dans la salles – rangs d'une classe, pilier, etc... – ont d'abord était représentés par des AABB Figure 3 car ce sont des formes très simples et donc ont permis de rapidement développer une modélisation correcte. Puis ces obstacles ont était représentés par des polygones convexes quelconques Figure 3 pour atteindre une modélisation plus générale, la convexité ne pose pas de problèmes dans la généralité car tout polygones peut se décomposer en une union de polygones convexes – il existe une triangulation pour tout polygones simples – on peut donc représenté tout obstacle en accolant des polygones convexes ensembles mais ce n'est en générale pas nécessaire.



FIGURE 3 – Des exemples d'obstacles, une AABB à gauche et une polygone convexe quelconque à droite

6 Description des agents

Beaucoup de simplifications ont été faites pour les agents. Au mieux les agents devrait être représentés par des ellipses souples car en vus de dessus les agents ont grossièrement la forme d'ellipse et peuvent prendre plus ou moins de place selon la pression qui leurs est appliqué – ils plient les bras pour laisser de la place par exemple –. Mais une ellipse est une forme qui engendre des calculs de géométrie compliqué et lent à exécuté d'autant plus si l'ellipse est souple. Nous avons donc choisis de représenter les agents par des disques rigides Figure 4, cela permet d'avoir une simulation rapide sans perdre trop de réalisme, les disques étant une assez bonne approximation d'une ellipse et la différence de rayons entre les cas extrêmes et petite par rapport au rayon moyen.

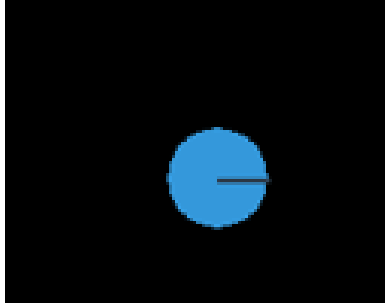


FIGURE 4 – Un agent

7 Modélisation du mouvement d'un agent