

Rapport de synthèse du sujet 1

Simulation de foule

Sommaire

Déroulement du projet.....	p.3
Composants logiciels utilisées.....	p.4
Evolution de la modélisation au cours du développement.....	p.4
Déroulement du développement.....	p.5
Sources.....	p.5
Conclusion et perspectives.....	p.6

I. Déroulement du projet

Le projet java a commencé le 16 avril 2021 et s'est terminé le 20 mai 2020, date de livraison des travaux réalisés sur cette période de cinq semaines. Le projet s'est globalement bien déroulé avec une répartition du boulot entre chaque élève du groupe, que nous avons mis en place lors de la phase analyse. Nous avons en plus attribué des rôles à chaque membre du groupe, à savoir LOSCO Axel le chef de la partie analyse, RIVIERE Baptiste le chef de la partie code, et CORNU Antoine le chef de la partie synthèse. Le rythme horaire de travail quotidien pendant ces 5 semaines a été continu.

Nous avons pour projet initial de créer deux types de simulation de foule. La première une simulation visuelle où l'on voit les individus sortir de la salle, avec en amont un paramétrage de salle, choix de la taille de la salle, du nombre de personnes, une simulation avec ou sans collision. Et la seconde simulation statistique, avec un paramètre en plus qu'avant, le nombre de simulation à effectuer, où on ne voit pas les simulations mais juste le résultat statistique du temps de sortie des individus de la salle, ce temps dépend juste de la position des individus dans la salle car ils sont positionnés aléatoirement à chaque simulation. De plus les individus et les sorties sont actualisés à chaque étape N+1, l'individu, voit sa position ou son état modifié, s'il avance ou s'il tombe à terre, et la sortie peut se fermer ou s'ouvrir en fonction de ce que choisie l'utilisateur pendant la simulation.

Pour la simulation de foule, nous avons comme objectif de départ d'affilier une vitesse à nos individus pour avoir des différences et se rapprocher de quelque chose de réaliste, mais nous l'avons enlevé pour nous faciliter la tâche, car d'après notre code et la réalité, nos individus se devaient d'être attribué une vitesse de déplacement en fonction de ce qu'ils incarnent dans la simulation. On avait trois choix qui s'offrait à nous, soit ne pas prendre en compte la vitesse comme on a fait avec des individus tous égaux, soit attribuer le paramètre vitesse à un individu de façon aléatoire, soit créer des classes qui fixe ce paramètre comme personnes handicapés, personnes normales ou personnes sportives.

Etant trois étudiants sur le projet et étant notre premier projet java, les parties analyse, code et rédaction n'ont pas été faites l'une après l'autre. Nous avons commencé la partie analyse et on a débuté le code très tôt, pour avoir notre base de l'application, qui sont les packages coordonnee, foule, espace et simulation, que nous avons très vite déterminé. De plus, le code pour l'interface graphique de l'application on était fait avant de faire le diagramme UML car aillant aucune notion d'interface graphique sur java, on voulait coder avant pour savoir quelle méthode il nous fallait pour pouvoir faire marcher notre interface graphique. Cela nous a permis d'être plus efficace et de faire directement le code java puis de faire le diagramme UML qui en découle.

Pour la présentation des données statistiques calculer, on a choisi de les afficher sur une nouvelle fenêtre par l'application.

II. Composants logiciels utilisés

StarUML a été notre outil de conception pour les diagrammes pour la partie analyse.

Eclipse IDE for java Developers 2020-12 a été notre unique outil pour la conception du code et l'exécutable de celui-ci.

Les bibliothèques java :

-**javax.swing ; java.awt** ont permis de mettre en place notre interface graphique de l'application.

-**java.util.ArrayList** a permis le stockage de donnée et de pouvoir ensuite les réutiliser ou les modifier en fonction de leur utilité.

GitHub.com a été notre outil de suivi, d'échange de code, de mise en commun au niveau d'une branche dev qui découle de master, éditée par le chef de la partie Code.

III. Evolution de la modélisation au cours du développement

Le projet initial de la modélisation de foule a été amené à être modifié et ajouté de nouvelle simulation. Tout d'abord nous avons ajouté une nouvelle forme de simulation aux deux autres, une simulation donnant la possibilité de choisir la salle. Ce qui permet de simuler des évacuations de foule de façon plus réalistes comme par exemple dans une salle de classe ou encore dans un avion, cela nous a permis d'étendre les possibilités de simulation que d'avoir que des salles rectangulaires vides. Par la suite on a construit dans des répertoires, une base de données comportant des txt des différentes salles et une base de données comportant les images des différentes salles pour le choix de la salle. Cela nous a permis de réduire considérablement les méthodes et lignes de code qui créent les différentes salles.

Nous avons aussi modifié, certaines méthodes pour les rendre plus efficaces ou plus réalistes. Comme la méthode `deplacementAvecCollision()`, où nous y avons rendu nos individus un peu plus intelligents comme passer à côté d'une personne à terre, et pas attendre derrière qu'elle se relève. Nous avons aussi peaufiné la méthode permettant d'attribuer les chemins les plus rapides pour se rendre à une sortie pour chaque individu en leur donnant un chemin plus linéaire. Et on a mis en place dans certaines simulations la possibilité de mettre un obstacle devant la porte permettant de réguler le flux, d'après nos sources.

Ainsi, la pratique, la programmation et les problèmes rencontrés lors de nos tests, nous ont forcé à revenir sur notre modélisation pour trouver des solutions.

IV. Déroulement du développement

Mise en place d'un projet git, où chaque étudiant créait ça branche avec un nom explicite sur quoi il travaillait, puis une fois terminer « merger » ça branche récupérait les éventuels changements apportés par les autres puis recréait une branche pour commencer une nouvelle chose ou apporter des nouvelles modifications à quelque chose.

L'utilisation de GitLab pour mettre le projet java en commun a permis de réduire les conflits au moment des « merger » et de voir directement où sont les erreurs avec les autres codes grâce aux comparateurs de code, ce qui est plus simple que de faire des copier-coller et chercher des erreurs pendant des heures.

Nous avons codé en premier le corps de l'application, la simulation, l'espace, la foule et les coordonnées, puis nous nous sommes répartie le travail en deux pôles. Un sur la conception d'un graphe pour coder Dijkstra, qui permet de calculer les chemins des différents individus, et l'autre sur la conception de l'interface graphique (pour les détails Baptiste s'occuper de Dijkstra et Axel et Antoine de l'interface graphique).

La fin du développement c'est terminer par la création de nouvelle salle et l'implémentation de la base de données qui va avec, et la mise en place de trois petits easter-egg dans l'application.

V. Sources

Pour réaliser ce projet, nous nous sommes inspirés d'une thèse réalisée par Philippe PECOL à l'Université Paris-Est Ecole Doctorales Science Ingénierie et Environnement, dont le sujet est : « Modélisation 2D discrète du mouvement des piétons – Application à l'évacuation des structures du génie civil et à l'interaction foule-passerelle ». Cela nous a permis de mieux comprendre le déplacement de la foule, son interaction avec l'espace qui l'entoure et avec elle-même.

De plus, dans nos expériences nous nous sommes inspirés d'une vidéo YouTube de foulescopie sur le dilemme de l'évacuation,

[Le dilemme de l'évacuation | Format Extra-Court #3 - YouTube](#)

Cela nous a permis d'ajouter des paramètres à nos simulations, celui d'un obstacle devant la sortie, permettant une évacuation plus rapide de la foule.

Pour finir, le package javax.swing et java.awt qui nous ont servi à l'affichage graphique, nous avons suivi les vidéos YouTube données dans le sujet plus quelque autre vidéo complétant celle-ci, pour par exemple utiliser un timer.

VI. Conclusion et perspectives

Notre application pour la simulation de l'évacuation de foule est maintenant terminée au bout de 5 semaines. Ce projet nous a permis de mettre en œuvre nos premières connaissances en programmation orienté objet, et de les approfondir avec la manipulation de nouveaux outils, la mise en place d'interface graphique ou la programmation de Dijkstra. Avec les conditions sanitaires et le télétravail, même si on avait des cours que l'on pouvait assister en présentiel, notre groupe composé de 2 personnes pouvant venir en présentiel et un troisième étant en distanciel, nous avons donc beaucoup utilisé les moyens de communication (Messenger, Teams...). Ce projet nous a donné l'occasion d'appliquer les méthodes de travail en équipe et de gestion de projet, que nous avons déjà développé et mis en place sur de plus petit projet.

Notre application peut être modifiée, pour l'apport d'amélioration dans la simulation, comme par exemple l'ajout de l'attribut vitesse à nos individus, ou encore l'amélioration des méthodes de déplacement car d'après les expériences que mène l'application que nous avons crée et de nos sources. Nous obtenons une contradiction, notre simulation nous montre que sans obstacle devant la sortie les individus sortent plus rapidement de la salle que quand il y a un obstacle, alors que nos sources nous montrent le contraire par des expériences. Cela montre que notre application à encore beaucoup de défaut et peut-être amélioré. Comme aussi le calcule statistique réalisée qui est une simple moyenne faite, et qui peut-être changer en réalisant avec une Map la récupération de plus de données et faire des moindres carrées sur nos données pour avoir des valeurs plus justes. Toutefois notre application permet de bien visualisé le déplacement de la foule dans son espace.