

SimFoule

Documentation et théorie

Version 0.1

Auteur :
Stephan RUNIGO

15 janvier 2018

Résumé

Objet : Ce document (en cours de construction), accompagne le programme SimFoule (lui même en cours de développement).

Contenu : Il contient un manuel d'installation et d'utilisation ainsi que quelques développements théoriques liés à ce programme de simulation.

Public concerné : Ce document s'adresse aux enseignants et aux étudiants des sections sciences physiques et informatique. Il s'adresse également à tout les passionnés d'informatique et de la langue de Molière.

SimFoule est un simulateur numériques de foule offrant une représentation graphique et une interaction dynamique avec certains paramètres. Destinés à un usage pédagogique, il permet de visualiser la simulation d'une foule lors de l'évacuation d'un bâtiment. Cette documentation accompagne ce programme.

Le premier chapitre présente le simulateur, fourni une procédure d'installation et décrit les commandes permettant l'interaction avec le programme.

Les deux chapitres suivants développent les modèles physiques, mathématiques et numériques liés au simulateur.

Enfin, le dernier chapitre rassemble les informations liées à la structure du programme.

Table des matières

1	Installation et utilisation	1
1.1	Présentation du simulateur	1
1.2	Installation du simulateur	1
1.3	Commande du simulateur	2
1.3.1	Résumé des options	2
2	Modèles physiques	3
2.1	Les humains	3
2.1.1	Vitesse souhaité	3
2.1.2	Expressions des forces	3
2.2	La foule	3
2.2.1	Chaîne	3
2.2.2	Évolution	3
2.3	Le batiment	3
2.3.1	Cellule	4
2.3.2	Étage	4
2.3.3	Batiment	4
3	Modèles mathématiques	5
3.1	Discrétisation	5
3.1.1	Discrétisation des dérivées	5
3.1.1.1	Dérivé symétrisée	5
3.1.2	Discrétisation de la relation fondamentale de la dynamique	5
3.2	Algorithme de plus court chemin	6
3.2.1	Discrétisation de l'équation iconale	6
4	Développement	7
4.1	Langage	7
4.1.1	C	7
4.1.2	SDL 2	7
4.2	Modèle Vue Controleur	7
4.2.1	Les répertoires du simulateur	7
4.2.2	Le modèle	7
4.2.3	La vue	8
4.2.4	Le controleur	8
4.3	Diagrammes	8
4.3.1	diagramme de classes	8
4.3.2	diagrammes de séquences	9
4.3.2.1	Simulation graphique	9
4.3.2.2	Réinitialisation	9
4.4	Valeurs implicites	10
4.4.1	Réglage de dt, durée et pause	10
4.4.1.1	Limitation des valeurs des variables	10

4.4.1.2	Paramètres physiques	10
4.4.1.3	Paramètres dynamiques	10

Bibliographie	11
----------------------	-----------

Chapitre 1

Installation et utilisation

1.1 Présentation du simulateur

Le simulateur SimFoule est un programme informatique écrit en C et qui utilise la librairie SDL2. Il donne une représentation graphique d'une foule se déplaçant dans un bâtiment. Une évacuation du bâtiment peut être déclenchée.

1.2 Installation du simulateur

Cette section traite de l'installation du simulateur SimFoule sur un système d'exploitation de type debian. Le téléchargement se fait avec un navigateur internet, la compilation et l'exécution se font dans un terminal. L'installation des outils de compilation nécessite les privilèges du super-utilisateur.

Installation des outils de compilation

```
sudo apt-get install gcc make libsdl2-dev
```

Téléchargement des sources

Télécharger le fichier `.zip` sur github

```
https://github.com/runigo/SimFoule/archive/master.zip
```

Décompresser le fichier `.zip`

```
unzip SimFoule-master.zip
```

Compilation

La commande `make` dans le répertoire des sources produit un fichier exécutable :

```
SimFoule
```

Exécution

En ligne de commande, avec d'éventuelles options

```
./SimFoule [OPTION]
```

Ou par exemple

```
./SimFoule duree 76 nervosite 29 dt 0.059 masse 9.1
```

La fenêtre graphique donne une représentation de la simulation, le terminal affiche les informations.

1.3 Commande du simulateur

Cette section traite des interactions entre le programme et l'utilisateur. Certaines de ces interactions ne seront actives que dans les prochaines versions (été 2018).

1.3.1 Résumé des options

option	valeur	commande
fond	<code>fond>0 & fond<255</code>	Couleur du fond
dt	<code>dt > 0.0 & dt < DT_MAX</code>	discrétisation du temps
duree	<code>0 < duree < DUREE_MAX</code>	Nombre d'évolution du système entre deux affichages
mode	<code>mode = + ou - 1</code>	Mode -1 : Wait, 1 : Poll
pause	<code>5 < pause < 555</code>	temps de pause en ms entre les affichages
initial	<code>INITIAL_MIN < initial < INITIAL_MAX</code>	Numéro du fichier d'initialisation
masse	<code>MASSE_MIN < masse < MASSE_MAX</code>	Masse des humains
nervosite	<code>NERVOSITE_MIN < nervosite < NERVOSITE_MAX</code>	Nervosité des humains
aide	<code>()</code>	Affiche l'aide
help	<code>()</code>	Affiche l'aide

Chapitre 2

Modèles physiques

Ce chapitre traite des modèles physiques liés au simulateur de foule.

2.1 Les humains

2.1.1 Vitesse souhaitée

La vitesse souhaitée modélise la volonté d'un humain à se diriger dans une direction, elle est choisie en fonction de la direction vers la sortie et de l'encombrement dans cette direction par d'éventuels obstacles.[\[2\]](#)

2.1.2 Expressions des forces

Un humain exerce une force motrice en fonction de la différence entre sa vitesse réelle et sa vitesse souhaitée. Les murs exercent sur les humains une force d'impénétrabilité, les humains exercent entre eux des forces réciproques, également d'impénétrabilité.

2.2 La foule

Cette section traite de la modélisation de la foule.

2.2.1 Chaîne

La chaîne d'humain rassemble les humains présents.

2.2.2 Évolution

Une évolution consiste à exécuter

- une incrémentation des positions
- un calcul d'une nouvelle position en fonction de l'actuel et de l'ancienne.
- une mise à jour des forces s'exerçant sur les humains dans la situation nouvelle,
- un éventuel calcul d'un changement de décision.

2.3 Le bâtiment

Cette section traite de la modélisation d'un étage de bâtiment

2.3.1 Cellule

Un étage du bâtiment est discrétisé en cellule (de l'ordre de $1\text{ m} \times 1\text{ m}$). Une cellule peut contenir un mur ou être libre aux humains. Elle peut également être une sortie de l'étage. Elle contient une information (sens et direction à suivre) permettant de diriger les humains vers la sortie la plus proche. Une autre information pourrait être la deuxième sortie la plus proche donnant à l'humain un choix à faire.

2.3.2 Étage

Un étage est l'ensemble des cellules, il consiste en un plan en deux dimensions des murs de l'étage. Un étage peut posséder une ou plusieurs sorties. Les sorties des étages supérieurs sont des escaliers donnant sur les entrées de l'étage inférieur. Un algorithme permet de déterminer le plus court chemin pour accéder à la sortie la plus proche et initialise cette orientation dans les cellules.

2.3.3 Bâtiment

Un bâtiment est composé de plusieurs étages reliés par des escaliers. Il dispose d'un étage 0 pouvant posséder une ou plusieurs sorties. Les sorties des étages supérieurs amènent à des escaliers donnant accès à l'étage inférieur.

Chapitre 3

Modèles mathématiques

Ce chapitre traite des modèles mathématiques et numériques liés au simulateur de foule.

3.1 Discrétisation

La discrétisation de l'équation du mouvement se fait à l'aide de l'algorithme de Verlet. Cet algorithme consiste à symétriser la dérivée par rapport au temps puis d'obtenir une expression de $x(t + dt)$ en fonction de $x(t)$ et $x(t - dt)$. Cette expression permet de simuler de proche en proche le comportement du système physique. La solution discrète se rapproche de la solution analytique si la valeur de dt est convenablement choisie. En dehors d'un certain encadrement de dt , la solution discrète s'éloigne de la solution analytique.

3.1.1 Discrétisation des dérivées

3.1.1.1 Dérivée symétrisée

Par définition, la dérivée symétrique est :

$$\frac{dx}{dt} = \frac{x(t + dt) - x(t - dt)}{2 dt}$$

On en déduit l'expression suivante de la dérivée seconde :

$$\frac{d^2x}{dt^2} = \frac{x(t + 2 dt) - x(t) - x(t) + x(t - 2 dt)}{4 dt^2}$$

Le changement de variable $dt' = 2 dt$ simplifie cette expression :

$$\frac{d^2x}{dt^2} = \frac{x(t + dt) - 2x(t) + x(t - dt)}{dt^2}$$

Une expression disymétrique de la vitesse peut être utilisée pour l'évaluation des forces de viscosité ainsi que pour le calcul de l'énergie cinétique avant le calcul de la nouvelle valeur de $x(t + dt)$.

$$\frac{dx}{dt} = \frac{x(t + dt) - x(t)}{dt}$$

Après l'incrément, $\frac{dx}{dt} = \frac{x(t) - x(t - dt)}{dt}$

3.1.2 Discrétisation de la relation fondamentale de la dynamique

Dans le modèle d'Helbing ([2]) la relation fondamentale de la dynamique donne :

$$m_i \frac{\mathbf{x}_i(t + dt) - 2\mathbf{x}_i(t) + \mathbf{x}_i(t - dt)}{dt^2} = m_i \frac{\mathbf{v}_i^0(t) - \mathbf{v}_i(t)}{\tau_i} + \sum_{j \neq i} \mathbf{f}_{ij} + \sum_w \mathbf{f}_{iw}$$

avec

m_i représente la masse d'un piéton

$\mathbf{v}_i^0(t)$ la vitesse qu'il souhaite atteindre dans une direction

$\mathbf{v}_i(t)$ sa vitesse actuelle

τ_i un paramètre temporel,

\mathbf{f}_{ij} les forces d'interaction auquel il est soumis avec les autres humains,

\mathbf{f}_{iw} les forces d'interactions auquel il est soumis avec les murs.

On en déduit

$$\mathbf{x}_i(t + dt) = 2\mathbf{x}_i(t) - \mathbf{x}_i(t - dt) + dt^2 \frac{\mathbf{v}_i^0(t) - \mathbf{v}_i(t)}{\tau_i} + \frac{dt^2}{m_i} \sum_{j \neq i} \mathbf{f}_{ij} + \frac{dt^2}{m_i} \sum_w \mathbf{f}_{iw}$$

La force d'interaction avec un autre humain est :

$$\mathbf{f}_{ij} = \left(A_i \exp \frac{r_{ij} - d_{ij}}{B_i} + kg(r_{ij} - d_{ij}) \right) \mathbf{n}_{ij} + Kg(r_{ij} - d_{ij}) \Delta v_{ij}^t \mathbf{t}_{ij}$$

avec

$A_i \exp \frac{r_{ij} - d_{ij}}{B_i}$ Force d'interaction répulsive

r_{ij} somme des rayons des deux humains,

d_{ij} distance entre les centres de gravité des deux humains,

A_i B_i constantes.

$kg(r_{ij} - d_{ij})\mathbf{n}_{ij}$ Force de contact qui empêche l'interpénétration des piétons

\mathbf{n}_{ij} vecteur normalisé pointant de l'humain i à l'humain j.

$Kg(r_{ij} - d_{ij})\Delta v_{ij}^t \mathbf{t}_{ij}$ Force de friction

Δv_{ij}^t est la vitesse tangentielle relative,

\mathbf{t}_{ij} est la direction tangentielle.

3.2 Algorithme de plus court chemin

Un algorithme simplifié permet de trouver le plus court chemin vers la sortie la plus proche.

3.2.1 Discrétisation de l'équation iconale

Chapitre 4

Développement

Ce chapitre traite de la structure et du développement des programmes de simulation

4.1 Langage

4.1.1 C

SimFoule est écrit en C [1] [3]

4.1.2 SDL 2

L'utilisation de la librairie SDL permet la réalisation d'une interface efficace avec l'utilisateur.

4.2 Modèle Vue Contrôleur

4.2.1 Les répertoires du simulateur

- **donnees** : Inclusion des librairies standard, définitions des constantes et des valeurs initiales de la simulation.
- **modele** : Système simulé.
- **graphisme** : Représentation graphique et affichage. Utilisation de la librairie SDL2
- **controle** : Liaison entre le système et l'interface graphique
- **objet** : Nécessaire à la compilation
- **documentation** : Documentation du simulateur et bibliographie.

4.2.2 Le modèle

Le système est constitué d'une foule et d'un bâtiment. La foule est constituée d'humains, le bâtiment est constitué d'étage et d'escalier, eux même constitués de cellule.

- **batiment** : Ensembles d'étages et d'escaliers. Fonctions de calcul d'interaction entre humains et d'évolutions des coordonnées des humains.
- **etage** : Ensemble de cellule. Fonctions de calcul de plus court chemin vers les sorties.
- **escalier** : Liaison entre les étages.
- **cellule** : Contient des informations (direction vers la sortie, densité d'humain...) et un statut (vide, mur, sortie, entrée)
- **foule** : Ensemble des humains.
- **humain** : nouvelles coordonnées, ancienne et actuel. Forces appliquées. Caractère (nervosité)

4.2.3 La vue

Construit une représentation graphique du système et affiche celle-ci.

4.2.4 Le controleur

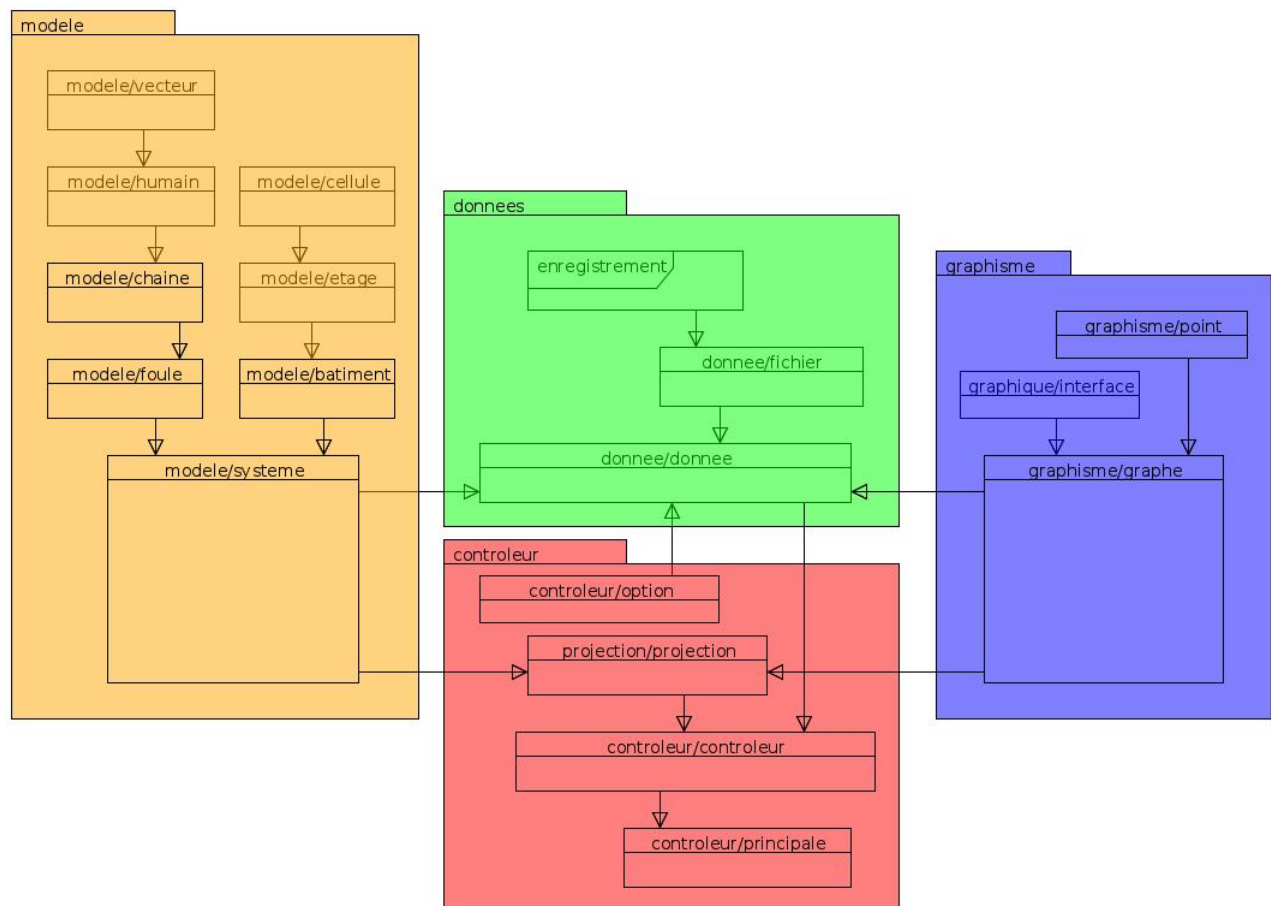
Exécute alternativement

- l’affichage de la vue
- l’évolution du modèle
- les actions du clavier.

4.3 Diagrammes

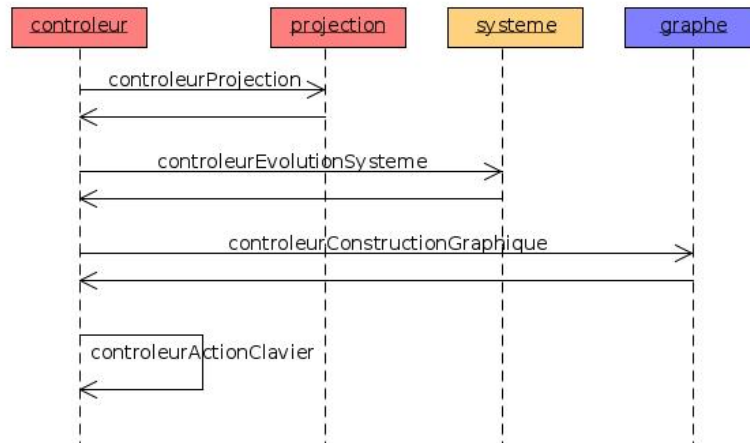
4.3.1 diagramme de classes

Ce diagramme présente les classes de SimFoule et indique l’inclusion des fichiers.

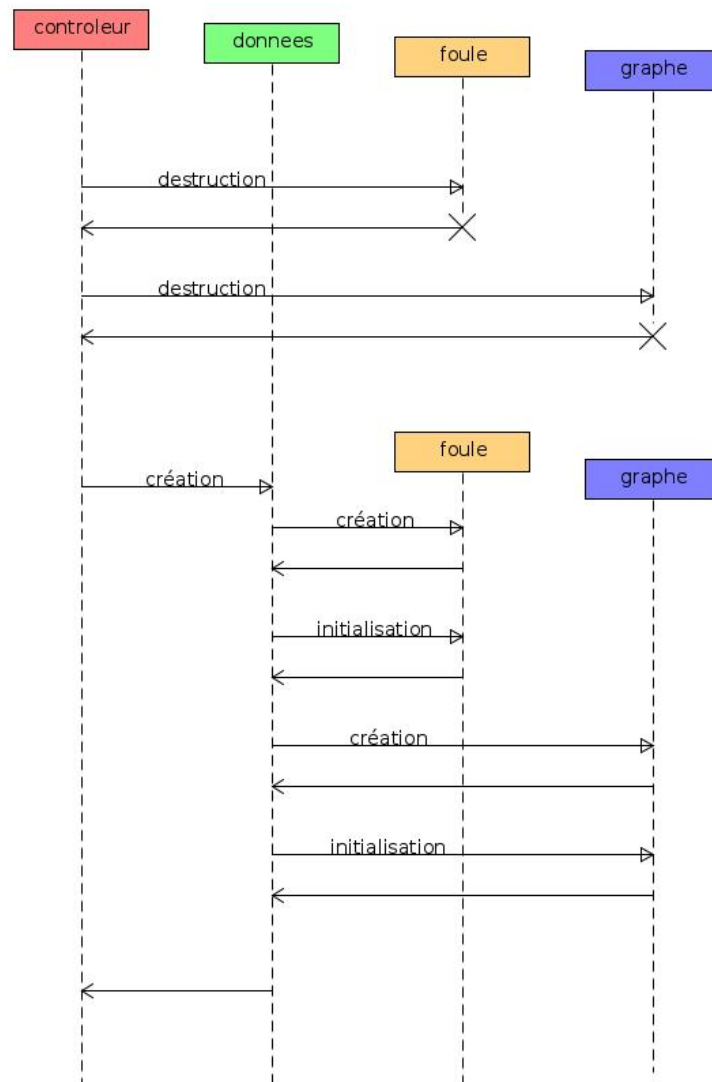


4.3.2 diagrammes de séquences

4.3.2.1 Simulation graphique



4.3.2.2 Réinitialisation



4.4 Valeurs implicites

4.4.1 Réglage de `dt`, durée et pause

Une incrémentation du système correspond à une avancée dans le temps de `dt`. Empiriquement, un affichage graphique par 30 ms, est obtenue avec une pause de l'ordre de 25 ms. Si à chaque affichage correspond à une dizaine d'incrémentations de `dt`,

$$dt \times \text{duree} = \text{delay}$$

$$dt \times 10 = 0,03$$

$$dt = 0,003$$

La valeur de `duree` peut être changée dynamiquement avec les touches `F11` et `F12` afin de faire varier la vitesse de la simulation. La valeur de `dt` peut être réglée avec l'option `dt` au démarrage du programme, celle de `delay` par l'option `pause`.

Dans `SimFoule`, les valeurs implicites de `dt` et `duree` sont égales à 0,003 et 11, celle de pause est égale à 25. Ces valeurs peuvent être affinée suivant les objectifs et les capacités de l'ordinateur.

4.4.1.1 Limitation des valeurs des variables

Au delà de certaines valeurs de certains paramètres dynamiques, la simulation s'éloigne du comportement physique.

Pour des raisons de discrétisation

En raisons de possibles erreurs d'algorithme

En raisons de possibles erreurs d'écriture

4.4.1.2 Paramètres physiques

4.4.1.3 Paramètres dynamiques

Au delà de certaines valeurs de certains paramètres dynamiques, la simulation s'éloigne d'une évolution physique réaliste.

Vitesse des mobiles

Distance entre les mobiles

Énergie des mobiles

Bibliographie

- [1] Delannoy Claude. *Le guide complet du langage C*. Eyrolles, 2014. [ISBN : 978-2-212-14012-5] [4.1](#).
- [2] Lemerrier Samuel. *Simulation du comportement de suivi dans une foule de piétons à travers l'expérience, l'analyse et la modélisation*. Université Rennes 1, 2012. [NNT : 2012REN1S028] [2.1](#)
[3.1.2](#).
- [3] Zhang Tony. *Le langage C*. campusPress, 2002. [ISBN : 2-7440-1518-0] [4.1](#).