

SimFoule

Documentation et théorie

Version 0.1

Auteur :
Stephan RUNIGO

3 février 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
1.3.2	Résumé du clavier	2
1.3.3	Fichier d'initialisation	2
1.3.3.1	Éditer les fichiers d'initialisations	3
2	Modèles physiques	4
2.1	Les mobiles	4
2.1.1	Vitesse souhaité	4
2.1.2	Expressions des forces	4
2.2	La foule	4
2.2.1	Chaîne	4
2.2.2	Évolution	4
2.3	Le batiment	4
2.3.1	Cellule	5
2.3.2	Étage	5
2.3.3	Batiment	5
3	Modèles mathématiques	6
3.1	Discrétisation	6
3.1.1	Discrétisation des dérivées	6
3.1.1.1	Dérivé symétrisée	6
3.1.2	Discrétisation de la relation fondamentale de la dynamique	6
3.2	Algorithme de plus court chemin	7
3.2.1	Discrétisation de l'équation iconale	7
4	Développement	8
4.1	Langage	8
4.1.1	C	8
4.1.2	SDL 2	8
4.2	Modèle Vue Controleur	8
4.2.1	Les répertoires du simulateur	8
4.2.2	Le modèle	8
4.2.3	La vue	9
4.2.4	Le controleur	9
4.3	Vision des mobiles	9
4.3.1	Modélisation	9
4.3.2	Champ de vision	9
4.3.2.1	Limitation des valeurs des variables	9
4.4	Diagrammes	10

4.4.1	diagramme de classes	10
4.4.2	diagrammes de séquences	10
4.4.2.1	Simulation graphique	10
4.4.2.2	Réinitialisation	11
4.5	Valeurs implicites	11
4.5.1	Réglage de dt, durée et pause	11
4.5.1.1	Limitation des valeurs des variables	12
4.5.1.2	Paramètres physiques	12
4.5.1.3	Paramètres dynamiques	12

Bibliographie		13
----------------------	--	-----------

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.

1.3.1 Résumé des options

option	valeur	commande
fond	$0 < \text{fond} < 255$	Couleur du fond
dt	$DT_MIN < dt < DT_MAX$	discrétisation du temps
duree	$0 < \text{duree} < DUREE_MAX$	Nombre d'évolution du système entre deux affichages
mode	mode = + ou - 1	Mode -1 : Wait, 1 : Poll
pause	$5 < \text{pause} < 555$	temps de pause en ms entre les affichages
initial	$INITIAL_MIN < \text{initial} < INITIAL_MAX$	Numéro du fichier d'initialisation
masse	$MASSE_MIN < \text{masse} < MASSE_MAX$	Masse des mobiles
nervosite	$NERVOSITE_MIN < \text{nervosite} < NERVOSITE_MAX$	Nervosité des mobiles
dessineAngle	= 0 ou 1	Dessine ou non les directions
dessineMur	= 0 ou 1	Dessine ou non les murs
dessineMobile	= 0 ou 1	Dessine ou non les mobiles
aide		Affiche l'aide
help		Affiche l'aide

1.3.2 Résumé du clavier

option	clavier	commande
duree	$0 < F9, F10, F11, F12$	Nombre d'évolution du système entre deux affichages
mode	Entrée	Avec ou sans attente
initial	Maj a..t et q..g	Réinitialisation
dessineAngle	F6	Dessine ou non les directions
dessineMur	F7	Dessine ou non les murs
dessineMobile	F8	Dessine ou non les mobiles

1.3.3 Fichier d'initialisation

Les fichiers d'initialisations se trouvent dans le répertoire **donnees/enregistrement/**. Ce répertoire et ces fichiers sont nécessaires pour activer les possibilité de réinitialisation.

1.3.3.1 Éditer les fichiers d'initialisations

Les fichiers d'initialisations peuvent être édités afin de créer des situations. Ils doivent être adaptés aux constantes `BATIMENT_X` et `BATIMENT_Y` définies dans le fichier `donnees/constantes.h`. Il contiennent des entiers séparés par des espaces correspondant au statut des cellules. Une ligne doit contenir `BATIMENT_X` entiers, et le fichier doit contenir `BATIMENT_Y` lignes. Les entiers reconnus lors de l'initialisation sont

entiers	0	1	2	9
statut	vide	mur	sortie	mobile

Chapitre 2

Modèles physiques

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

2.1 Les mobiles

2.1.1 Vitesse souhaitée

La vitesse souhaitée modélise la volonté d'un mobile 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 mobile exerce une force motrice en fonction de la différence entre sa vitesse réelle et sa vitesse souhaitée. Les murs exercent sur les mobiles une force d'impénétrabilité, les mobiles 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 de mobile rassemble les mobiles 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 mobiles 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 mobiles. Elle peut également être une sortie de l'étage. Elle contient une information (sens et direction à suivre) permettant de diriger les mobiles vers la sortie la plus proche. Une autre information pourrait être la deuxième sortie la plus proche donnant au mobile 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 mobiles,

\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 mobile 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 mobiles,

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

$A_i B_i$ constantes.

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

\mathbf{n}_{ij} vecteur normalisé pointant du mobile i au mobile j.

$g(x) = x$ si $x < 0$, $g(x) = 0$ sinon.

$Kg(r_{ij} - d_{ij})\Delta v_{ij}^t \mathbf{t}_{ij}$ 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 de mobiles, le bâtiment est constitué d'étages, eux même constitués de cellule.

- **système** : Contient la foule et le bâtiment. Fonctions de calcul des couplages entre la foule et le bâtiment, fonctions de mise à jour des forces et de la foule.
- **batiment** : Ensembles d'étages. Fonctions d'initialisation des étages.
- **etage** : Ensemble de cellule. Fonctions de calcul de plus court chemin vers les sorties.
- **cellule** : Contient des informations (direction vers la sortie, densité de mobile...) et un statut (vide, mur, sortie, entrée).

- **foule** : Chaîne de mobiles. Fonctions de calcul des couplages entre les mobiles, fonctions d'évolutions des coordonnées des mobiles.
- **mobile** : Coordonnées, forces appliquées, paramètres (taille, masse, 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 Vision des mobiles

Dans la version 1.3, les choix de directions ne se font que dans les cellules ayant été initialisées suivant une direction "diagonale". La direction souhaitée est alors choisies suivant le nombre de mobiles présents dans les cellules à atteindre. La version 1.4 doit mettre en place :

- Un choix pour les cellules dont la direction initialisée n'est pas "diagonale".
- Une diminution du module de la vitesse souhaitée si les cellules à atteindre sont occupées.

4.3.1 Modélisation

Une cellule contient les informations suivantes :

```
int statut; // 0 :libre, 1 :mur, 2 :sortie, 3 :entrée, 9 :mobile lors de l'initialisation
int visite; // 1 si initialisé, -1 si visite en cours
int distance; // Distance à la sortie
int angle; // Direction et sens des cellule à atteindre [0..7]
int dx; // Direction et sens des cellule à atteindre dx int dy; // Direction et sens des cellule à
atteindre dy
vecteurT sens; // Direction et sens à suivre, normalisé par l'initialisation. vecteurT sens1; //
Direction et sens à suivre, normalisé par l'initialisation. vecteurT sens2; // Direction et sens à suivre,
normalisé par l'initialisation. float norme; // Norme de sens avant la normalisation.
int nombre; // nombre de mobile.
```

4.3.2 Champ de vision

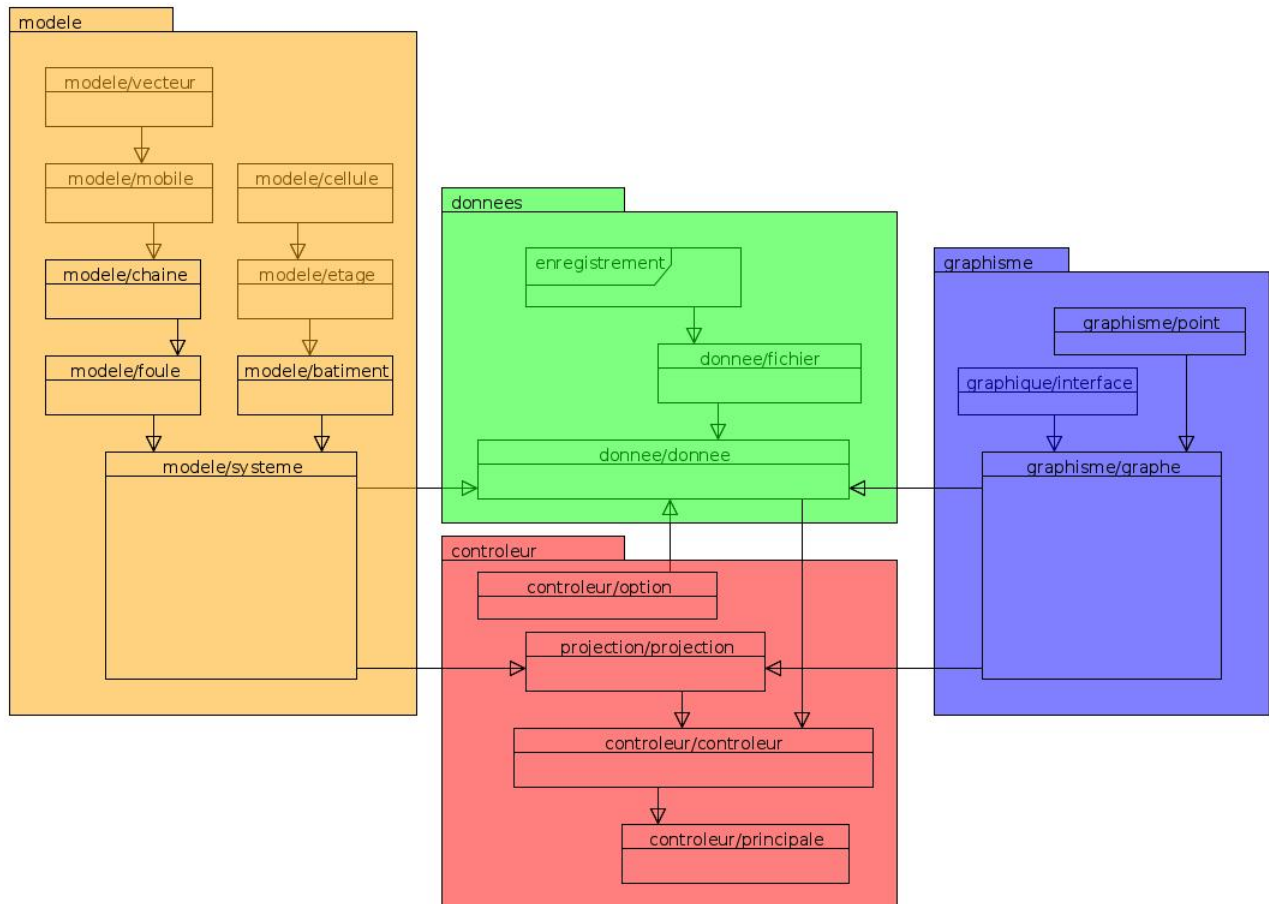
4.3.2.1 Limitation des valeurs des variables

donnees/constantes.c

4.4 Diagrammes

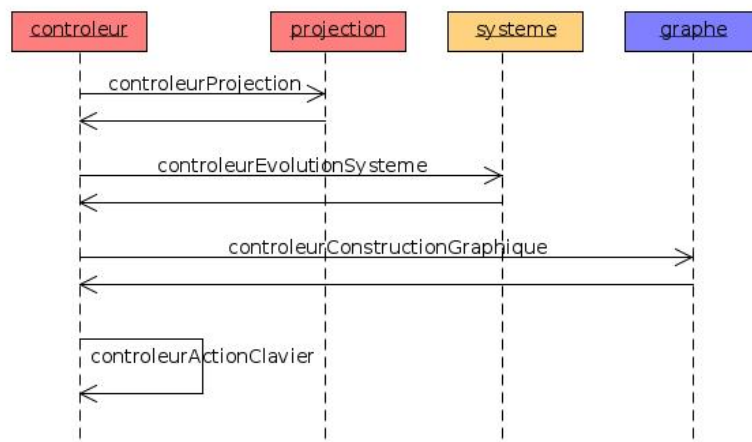
4.4.1 diagramme de classes

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

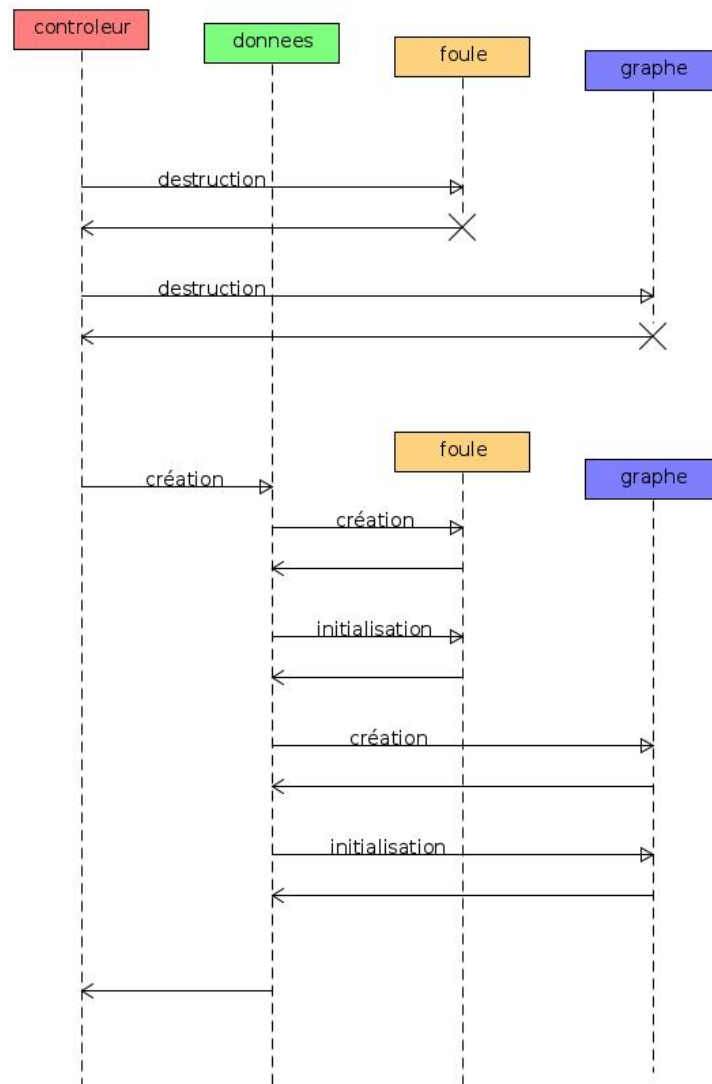


4.4.2 diagrammes de séquences

4.4.2.1 Simulation graphique



4.4.2.2 Réinitialisation



4.5 Valeurs implicites

4.5.1 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 **F9**, **F10**, **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, la valeurs implicites de pause est égale à 25, les valeurs implicites de dt et **duree** sont réglées dans le fichier `donnees/donnes.c`. Ces valeurs peuvent être modifiées suivant les objectifs et les capacités de l'ordinateur.

4.5.1.1 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 raison de possibles erreurs d'algorithme (Le calcul de plus court chemin est réalisé de manière basique et contient un certain nombre d'imperfection...)

En raison de possibles erreurs d'écriture

Aussi, le fichier `donnees/constantes.c` contient des valeurs maximale et minimale de certains paramètres. Ces bornes permettent de consolider le comportement du programme.

4.5.1.2 Paramètres physiques

4.5.1.3 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](#).