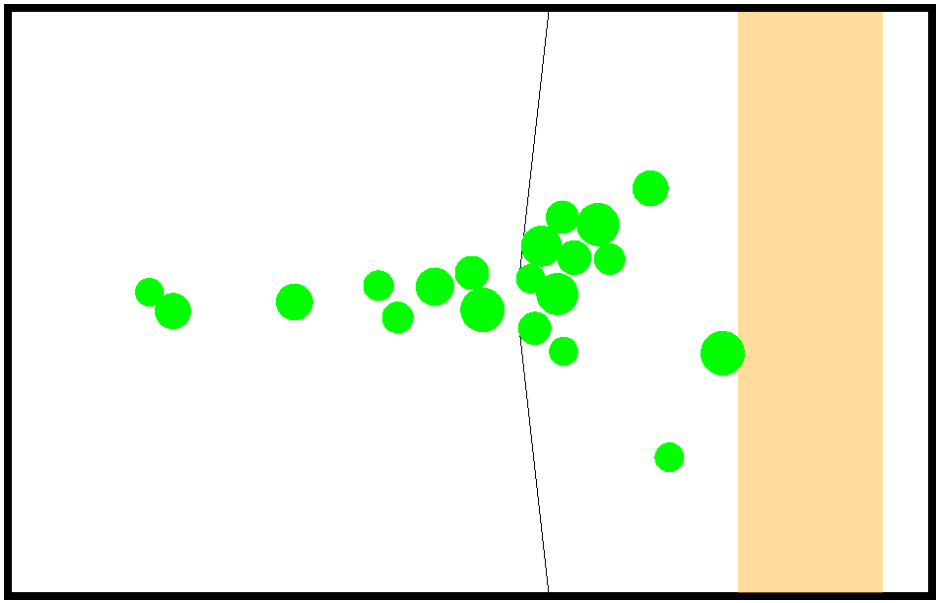
Simulation informatique des mouvements d'une foule.

Arnaud PELISSIER



Le mouvement des foules

But et différents types de modèles

Objectifs :

- Eviter les accidents
- Optimiser le temps d'évacuation

Modèles :

- Modèles cellulaires
- Modèles macroscopiques
- Modèles microscopiques

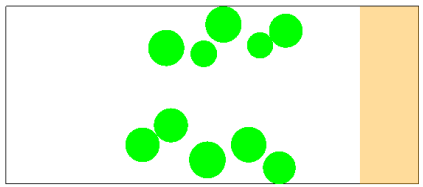


FIGURE – Modèle microscopique

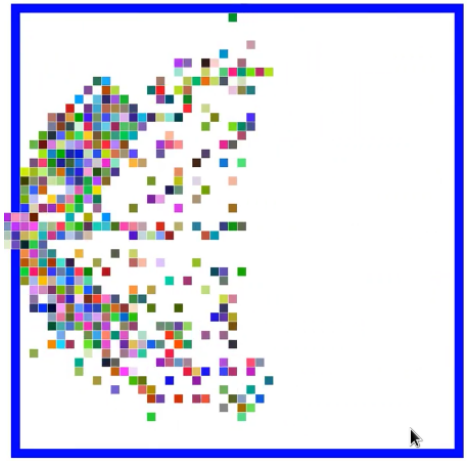


FIGURE – Modèle cellulaire

Fonctionnement des modèles microscopiques

- Chaque piéton possède un état $(\overrightarrow{v_i}, \overrightarrow{x_i}, \dots)_{i \in \mathbb{I}}$
- Le modèle est mise en mouvement par intégrations successives :
  - $\overrightarrow{a_i}(t + \Delta t) = \frac{1}{m_i} \overrightarrow{F_i}(t + \Delta t)$ ($2^{ème}$ *loi de Newton)*
  - $\overrightarrow{v_i}(t + \Delta t) = \overrightarrow{v_i}(t) + \Delta t \times \overrightarrow{a_i}(t)$
  - $\overrightarrow{x_i}(t + \Delta t) = \overrightarrow{x_i}(t) + \Delta t \times \overrightarrow{v_i}(t + \Delta t)$
- Gestion des collisions

Le modèle choisi

Caractéristiques

How simple rules determine pedestrian behaviors and crowd disasters

- Modèle heuristique
  - Les piétons cherchent le chemin le plus direct vers leur destination.
  - Les piétons ralentissent devant des obstacles pour éviter la collision.
- Implémentation en C++
  - Plus pratique que du C (bibliothèque standard / algorithmes / classes)
  - Plus rapide que du python

Le modèle choisi

Fonctionnement

Représentation d'un piéton

- Position $\overrightarrow{x}_i$, vitesse $\overrightarrow{v}_i$
- Rayon $r_i = \frac{m_i}{320}$, où $m_i \in [60, 100]$ est la masse du piéton
- Destination O_i
- Vitesse de croisière v_{0i}

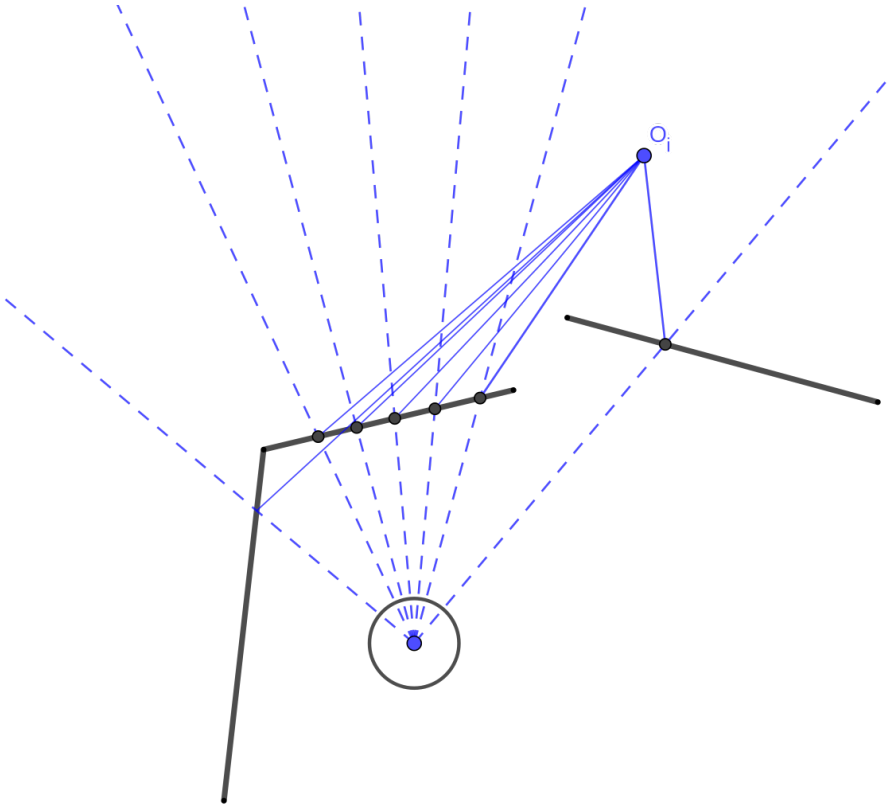
Représentation du monde

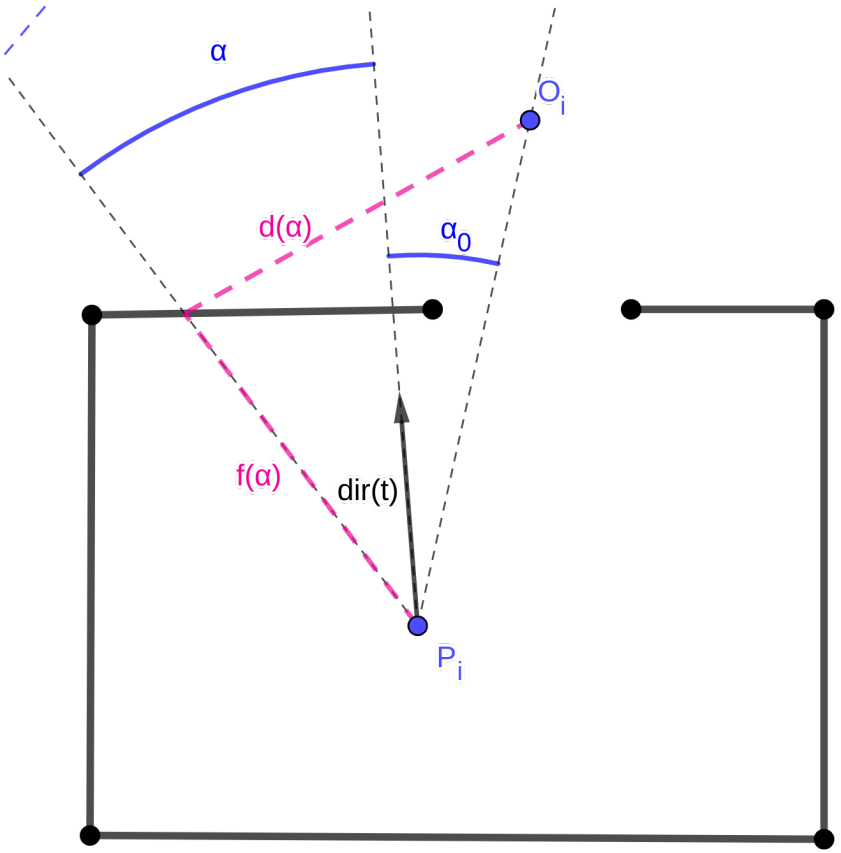
- Les murs sont un ensemble de segments $(p_{0,i}, p_{1,i})_{i \in \mathbb{I}}$
- une zone de départ (x, y, w, h)
- un point d'arrivée

- Force de répulsion lors de la collision entre deux piétons
- Force de répulsion lors de la collision entre un piéton et un mur
- Recherche de l'angle / de la direction optimale pour un piéton

Implémentation

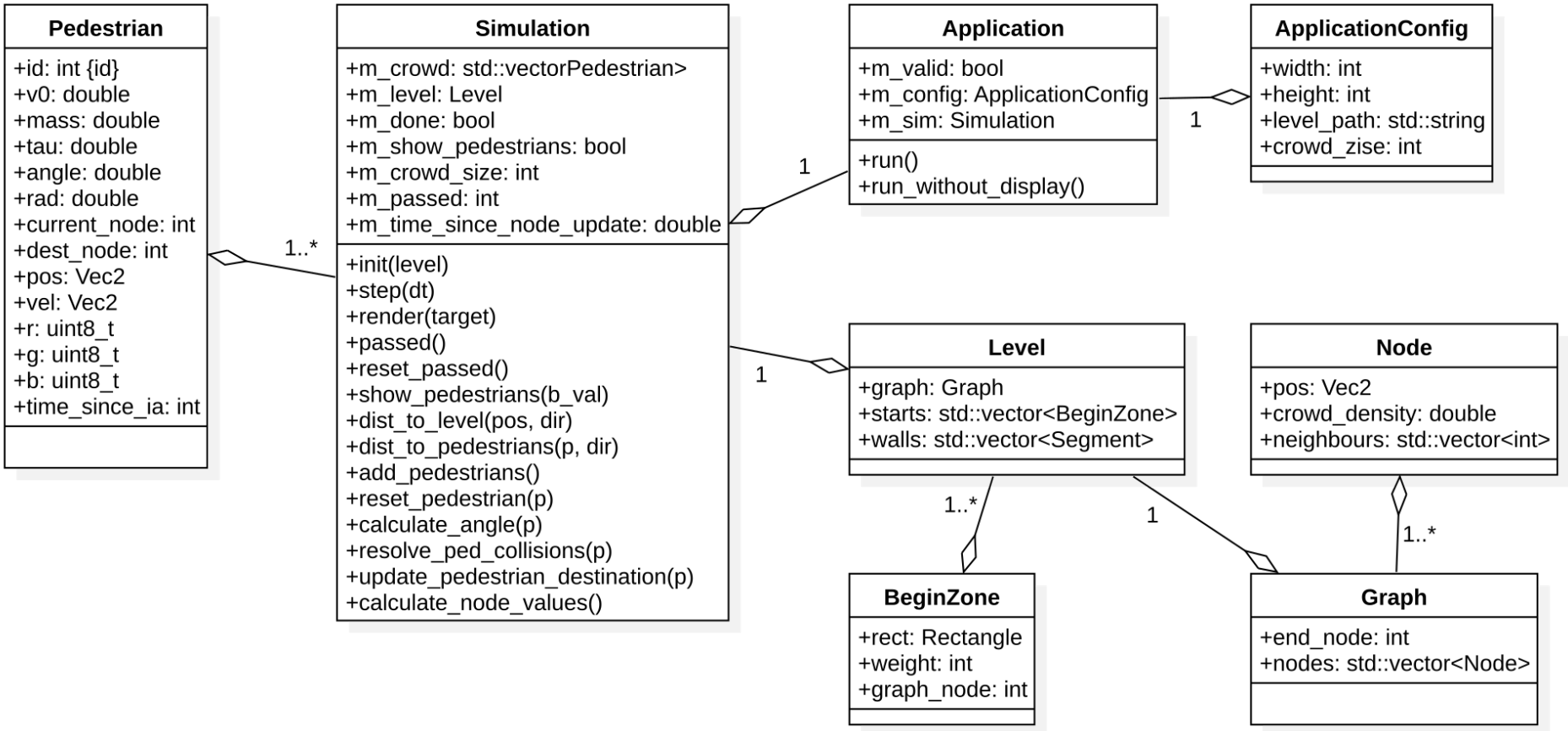
Calcul de l'angle d'un piéton





Implémentation

En C++



Simulation, résultats

Validation du modèle : obstacle en face d'une sortie

Ce modèle prédit-il une heuristique observée empiriquement ?

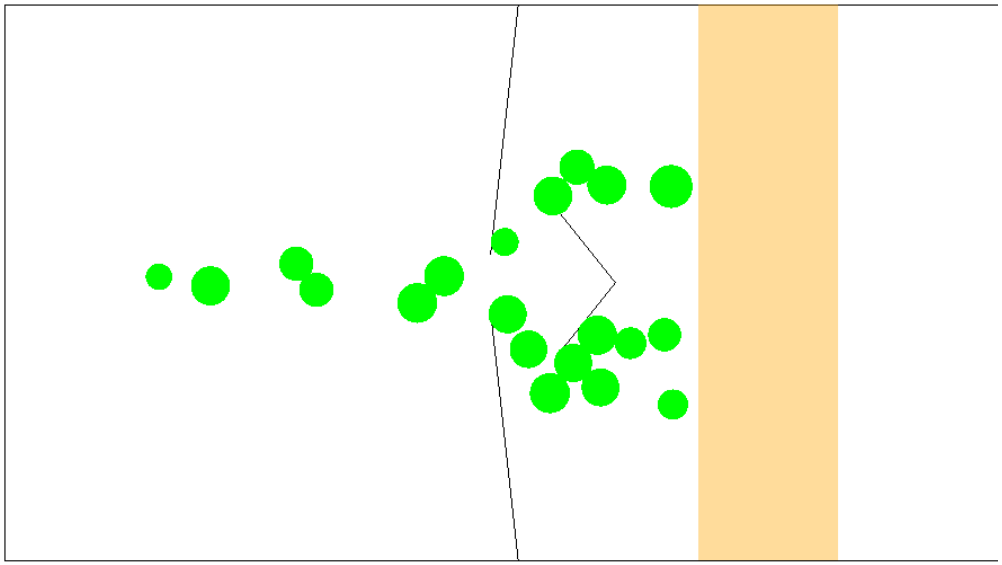
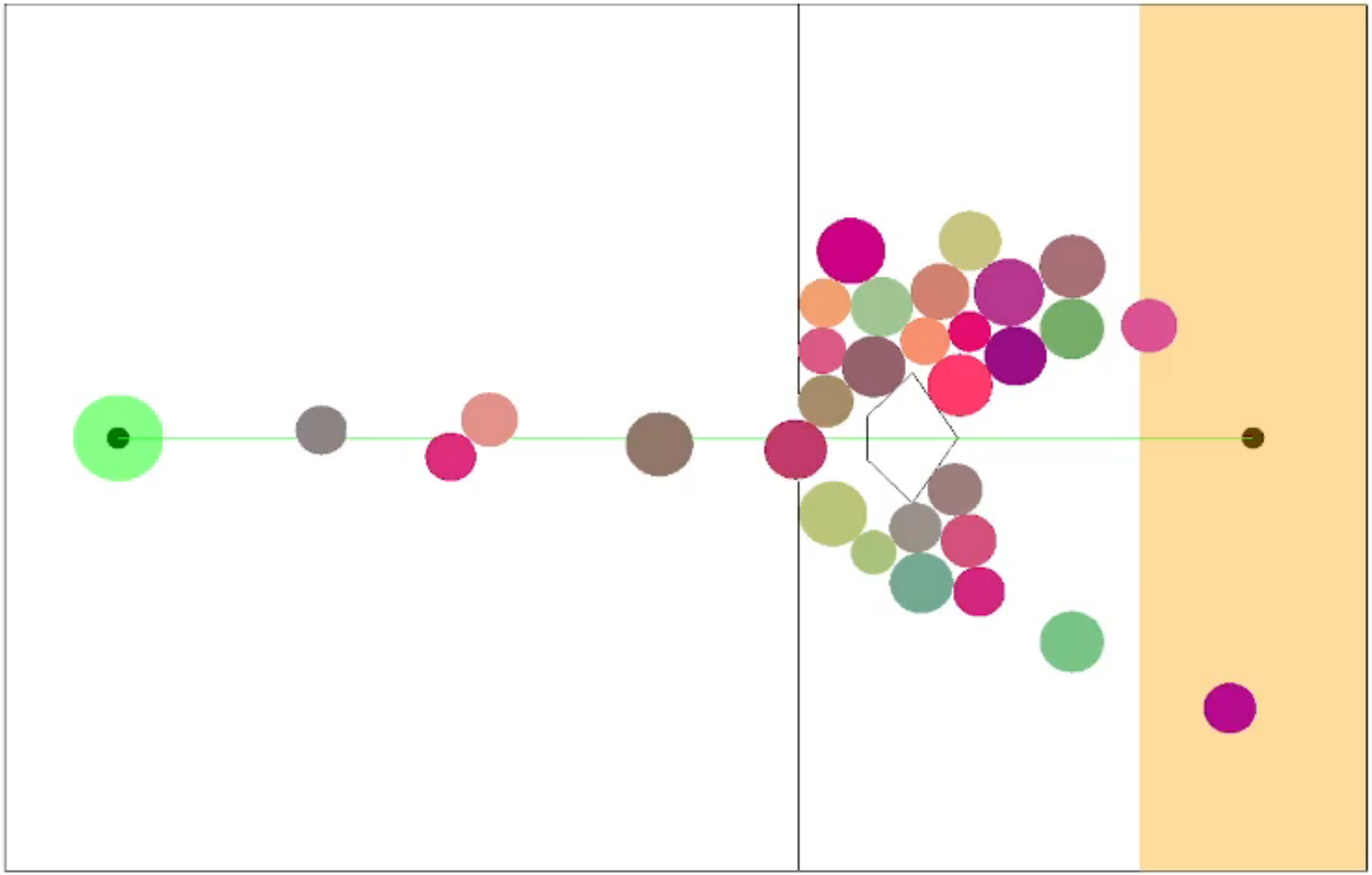
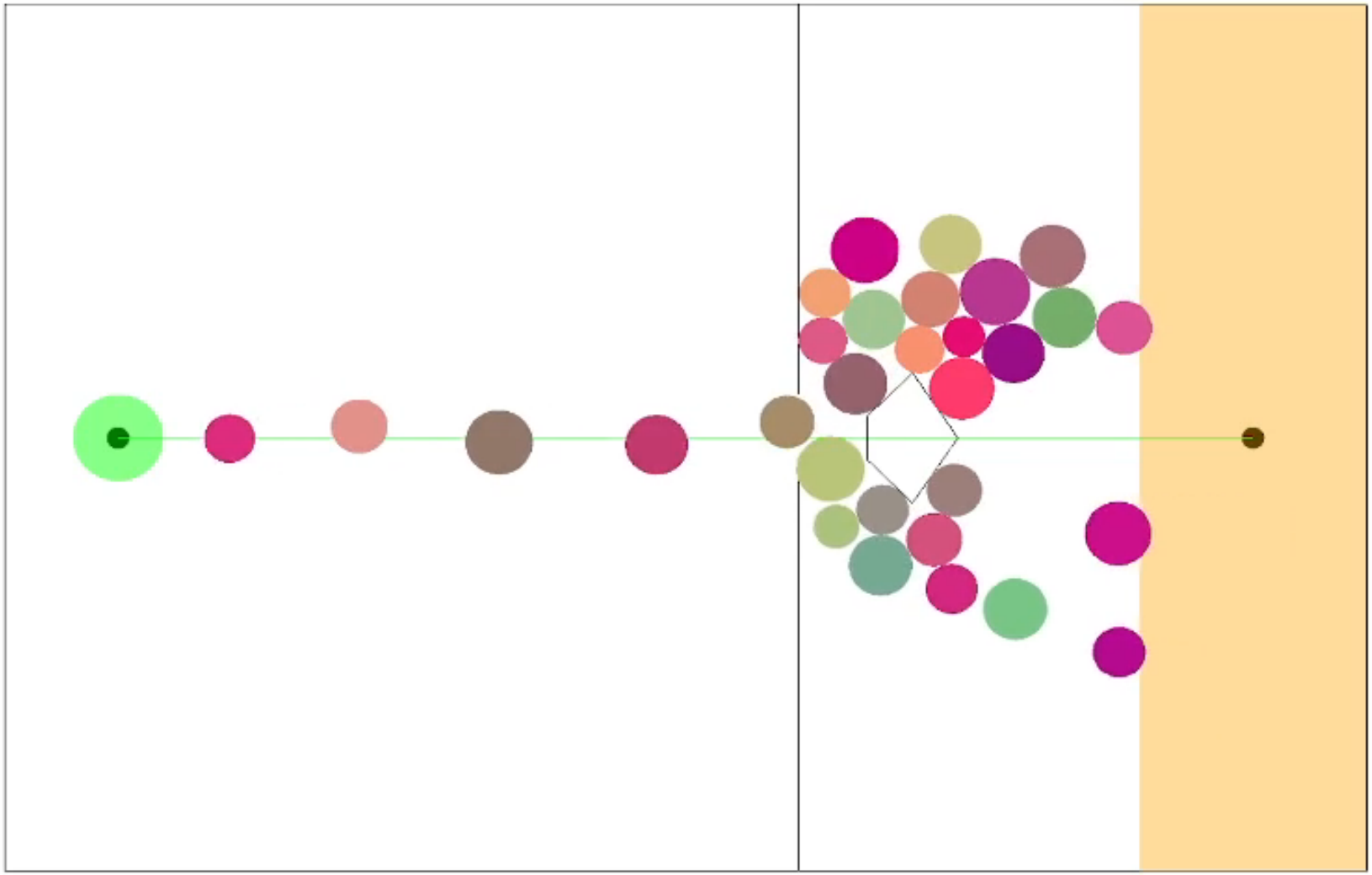
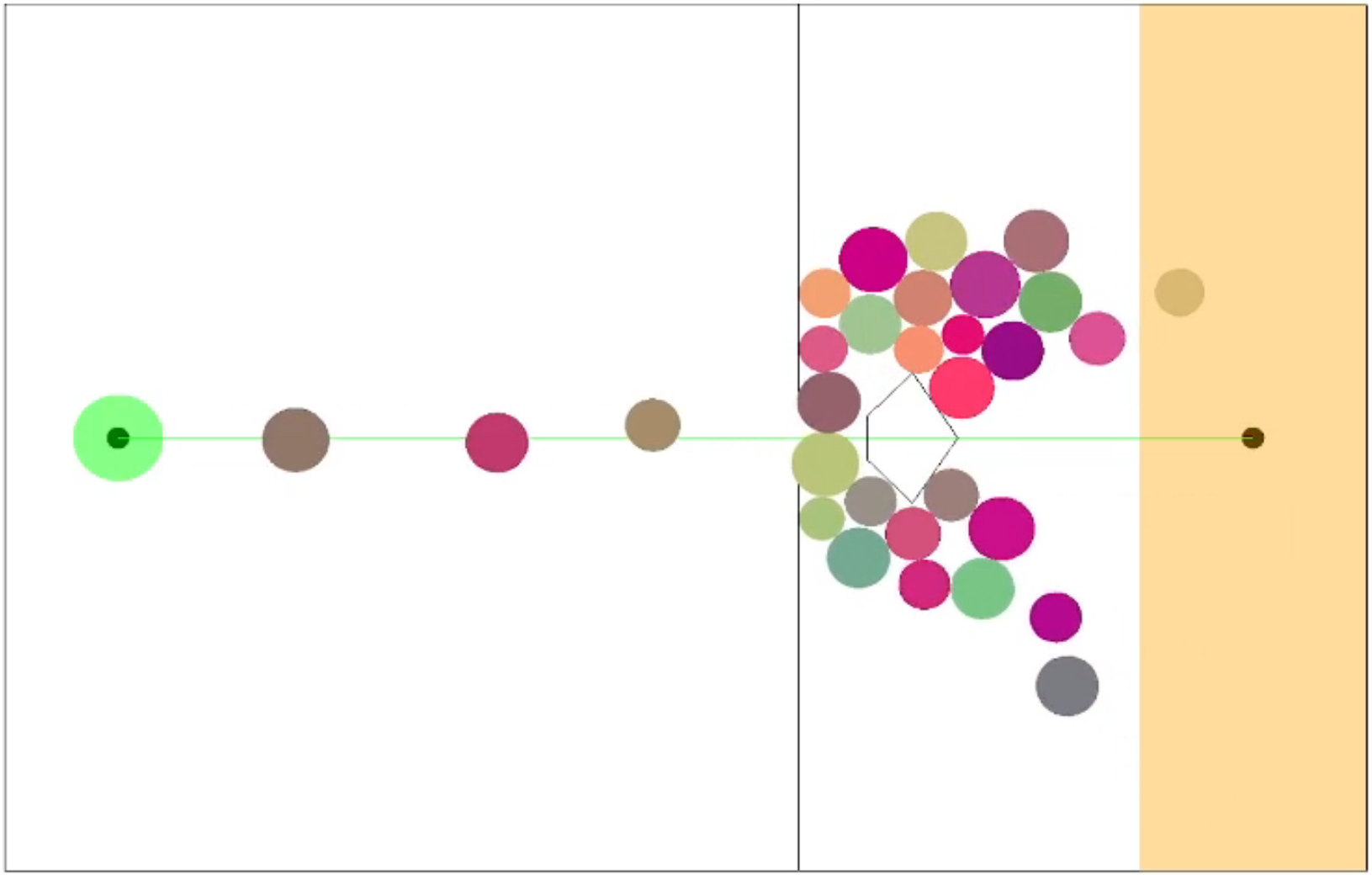
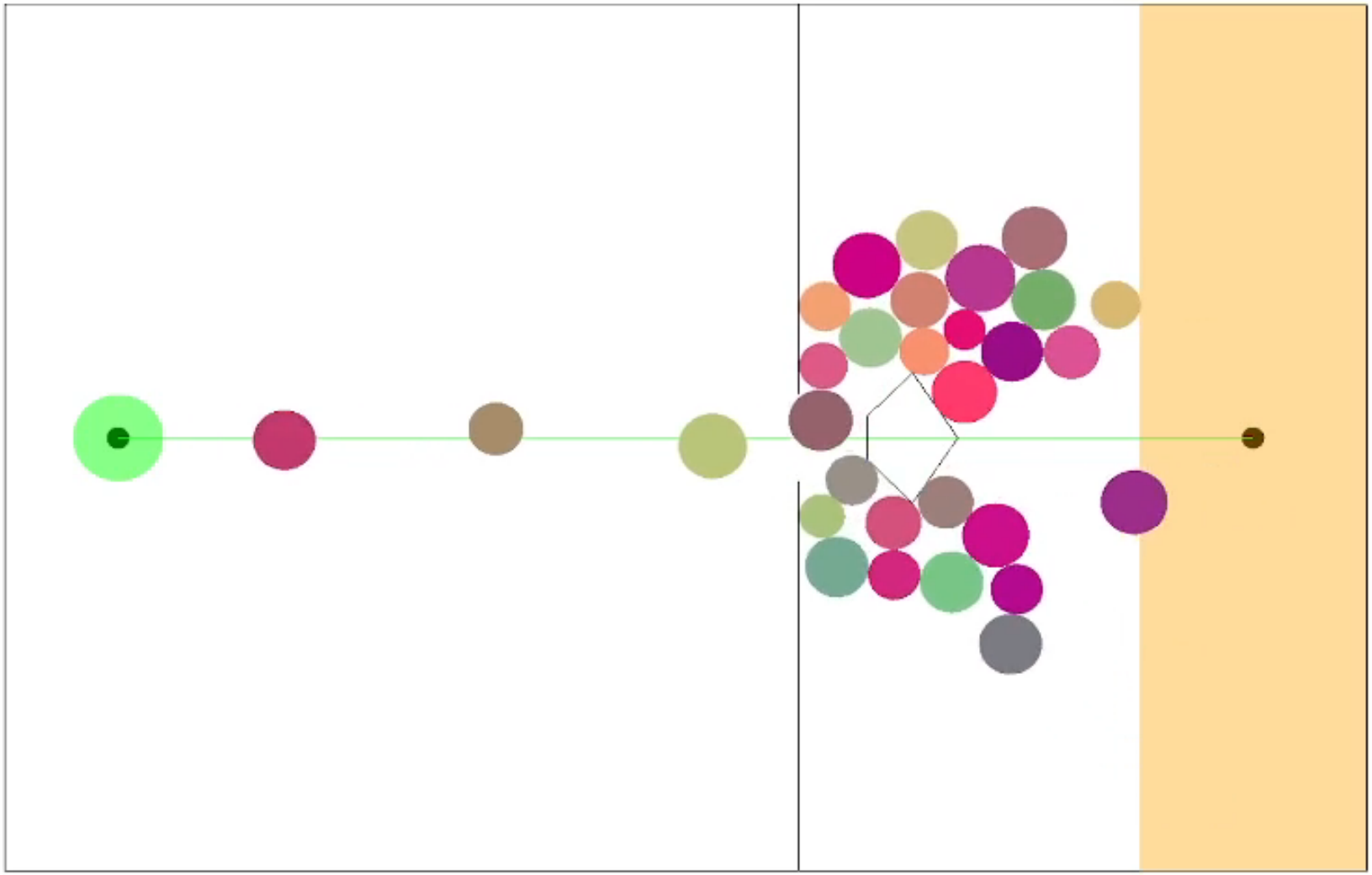


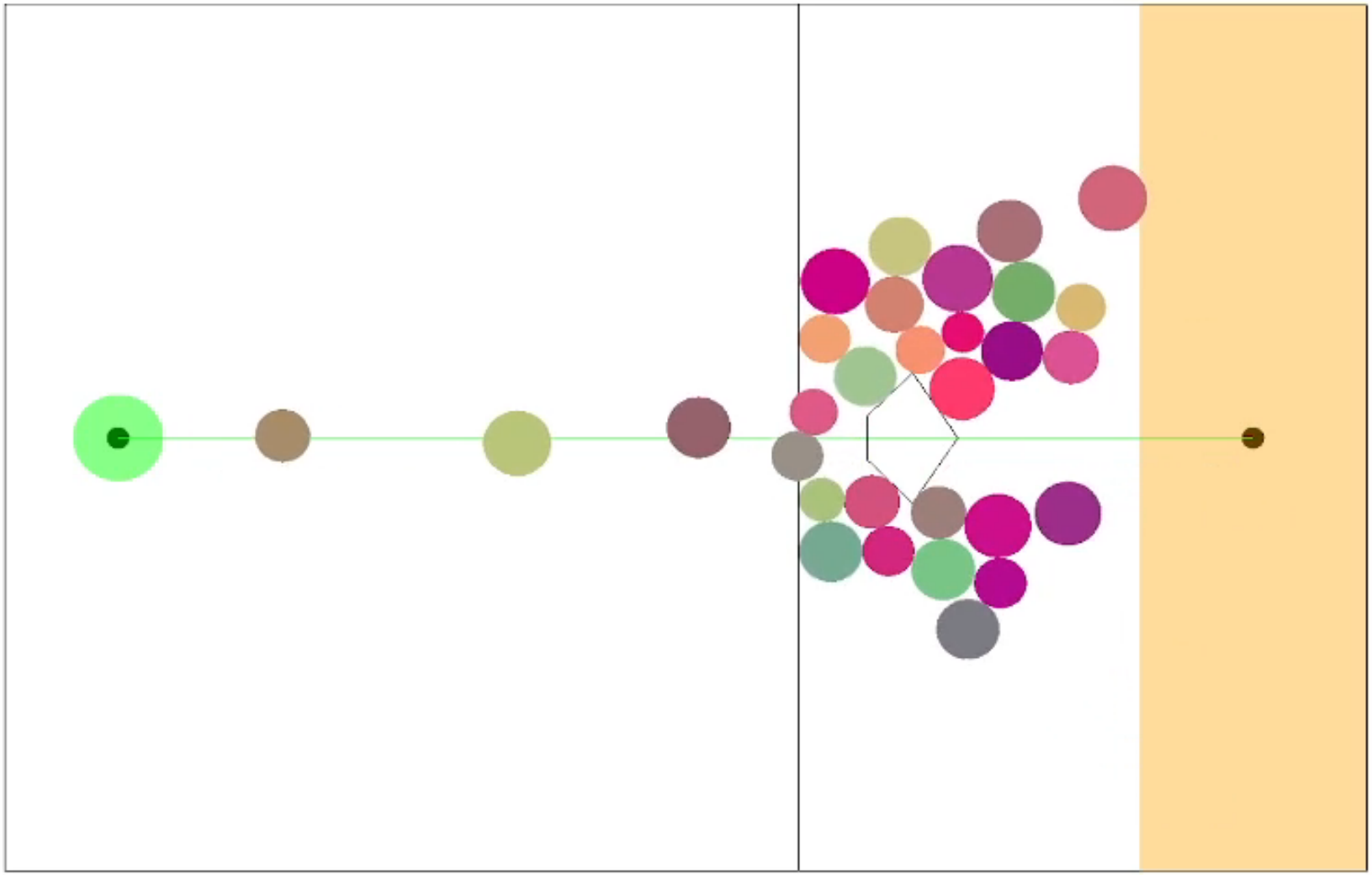
FIGURE – Première simulation

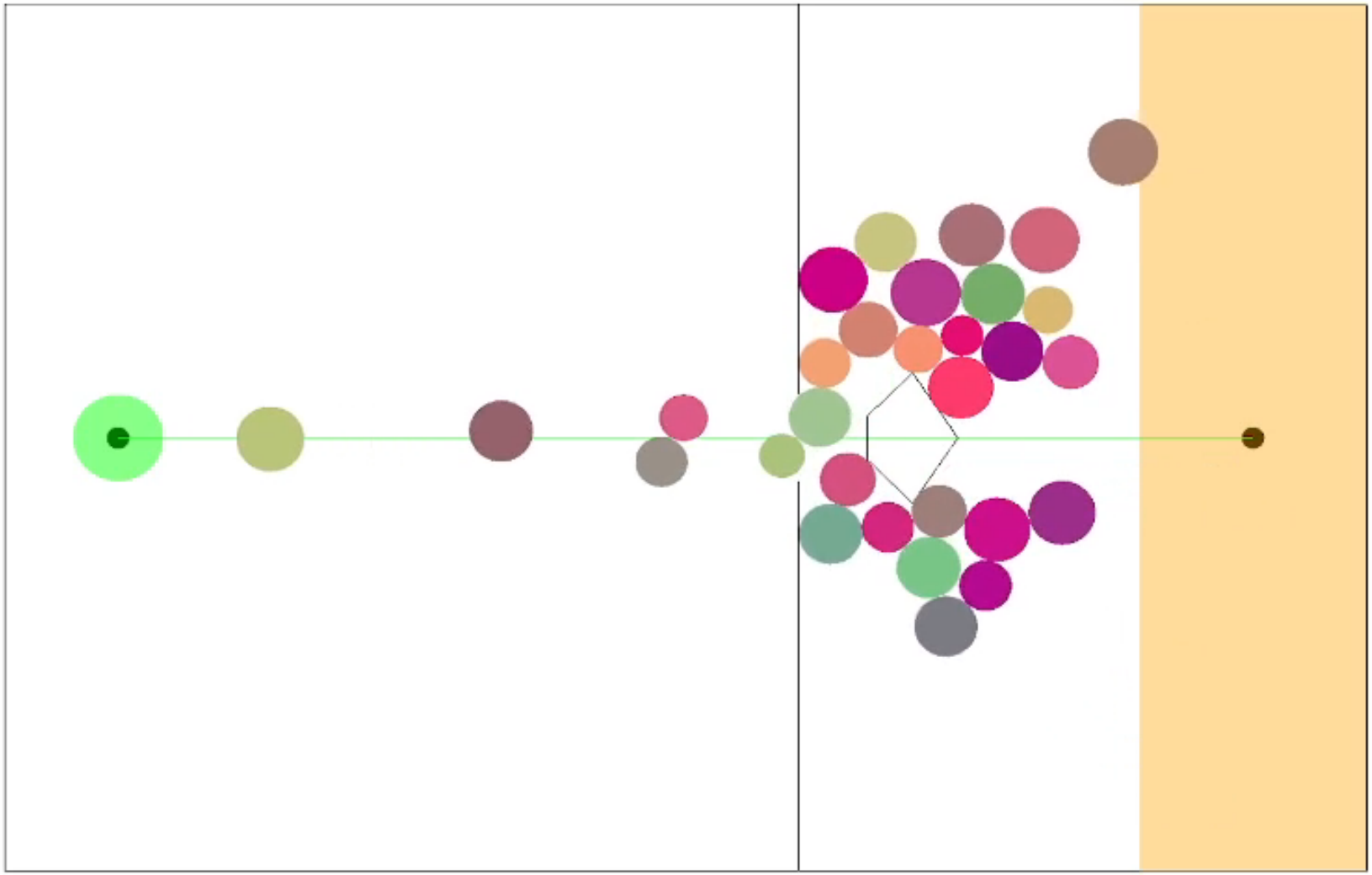


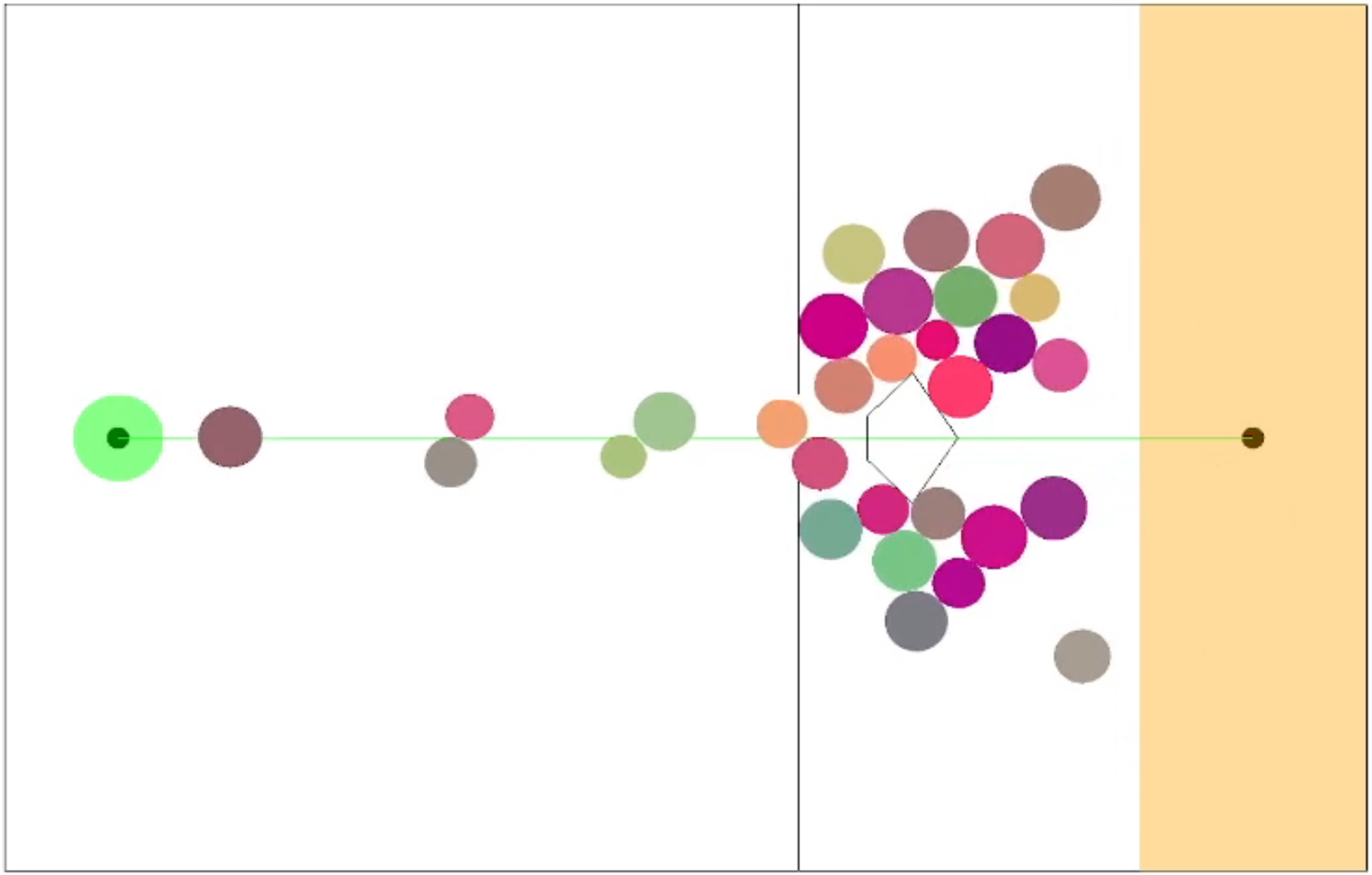


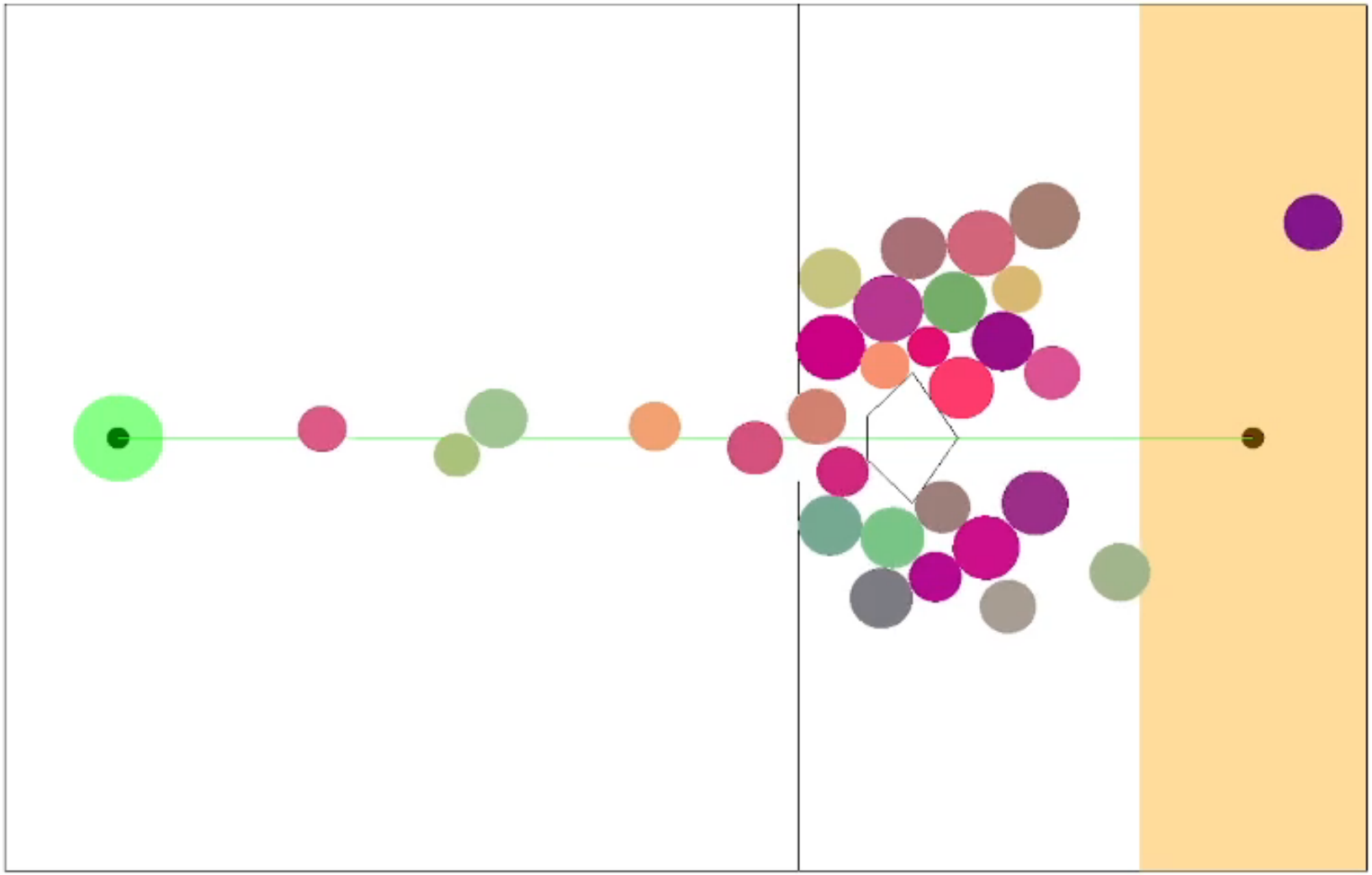


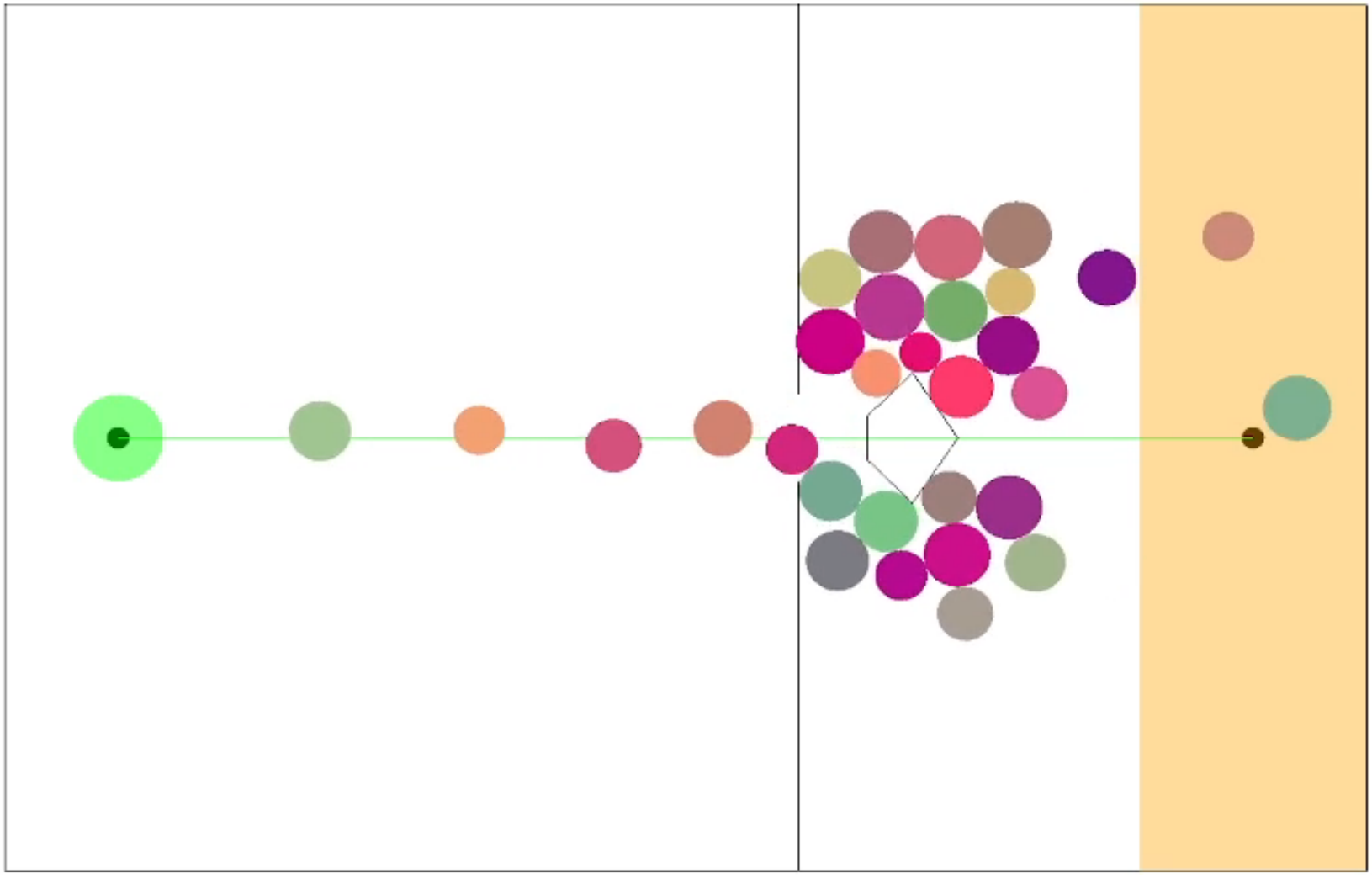


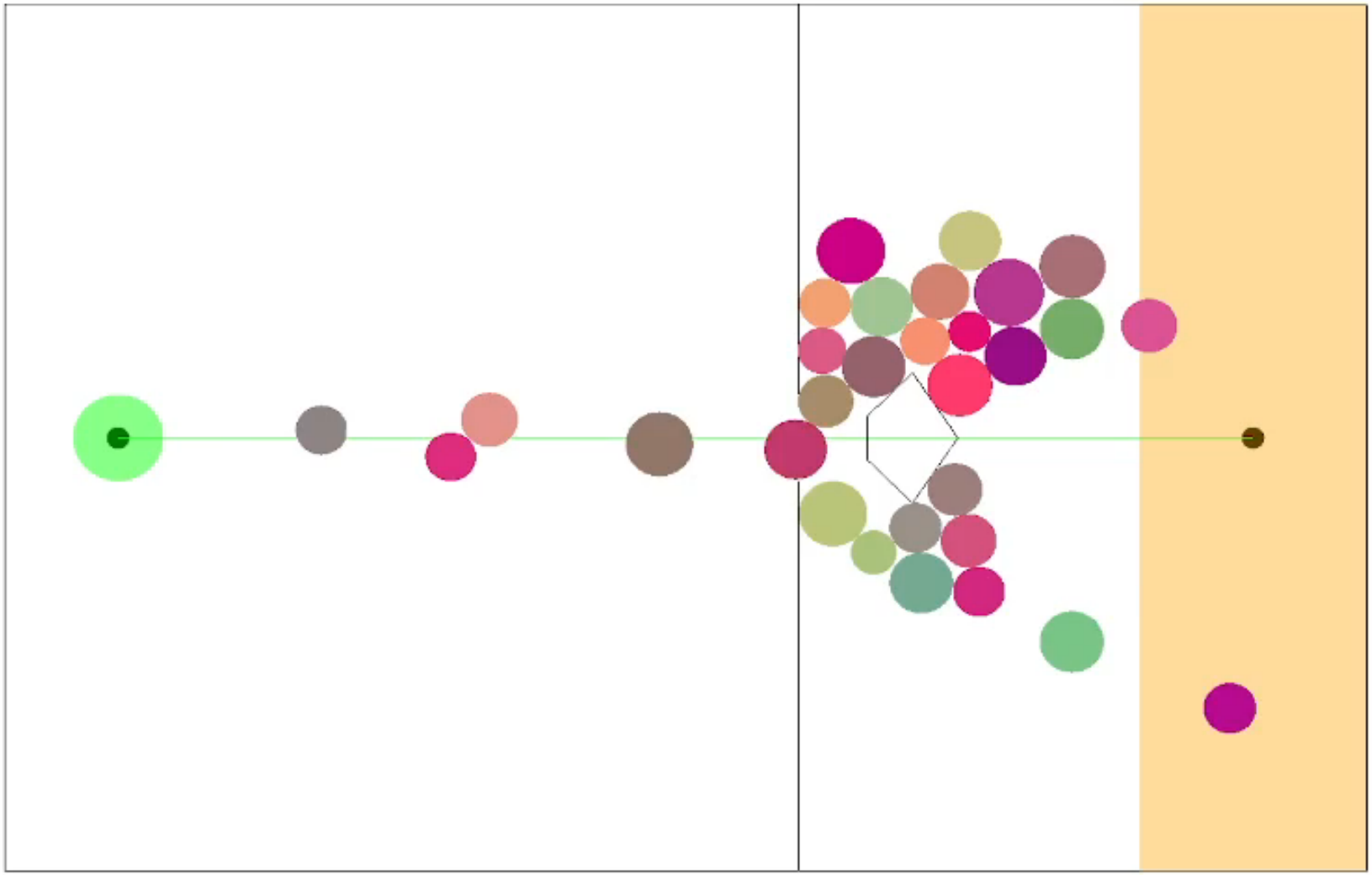












Simulation, résultats

Validation du modèle : obstacle en face d'une sortie

Experience

Mesurer le nombre de passages par seconde à travers une porte

- Sans obstacle devant cette porte
- Avec obstacle

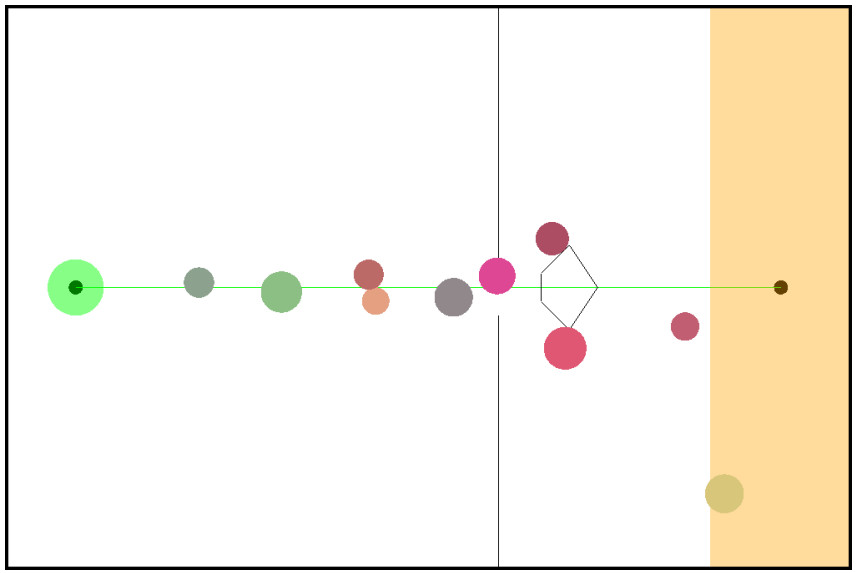


FIGURE – Avec obstacle

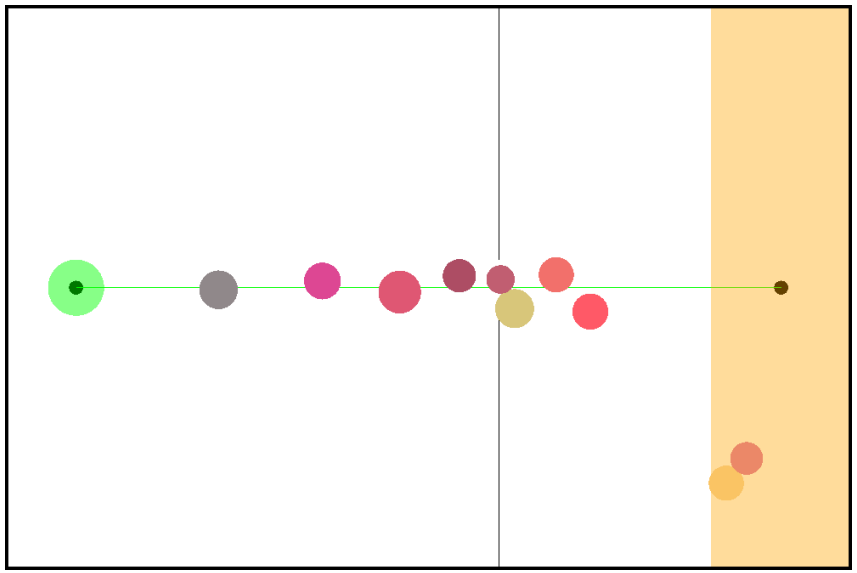


FIGURE – Sans obstacle

Simulation, résultats

Résultats

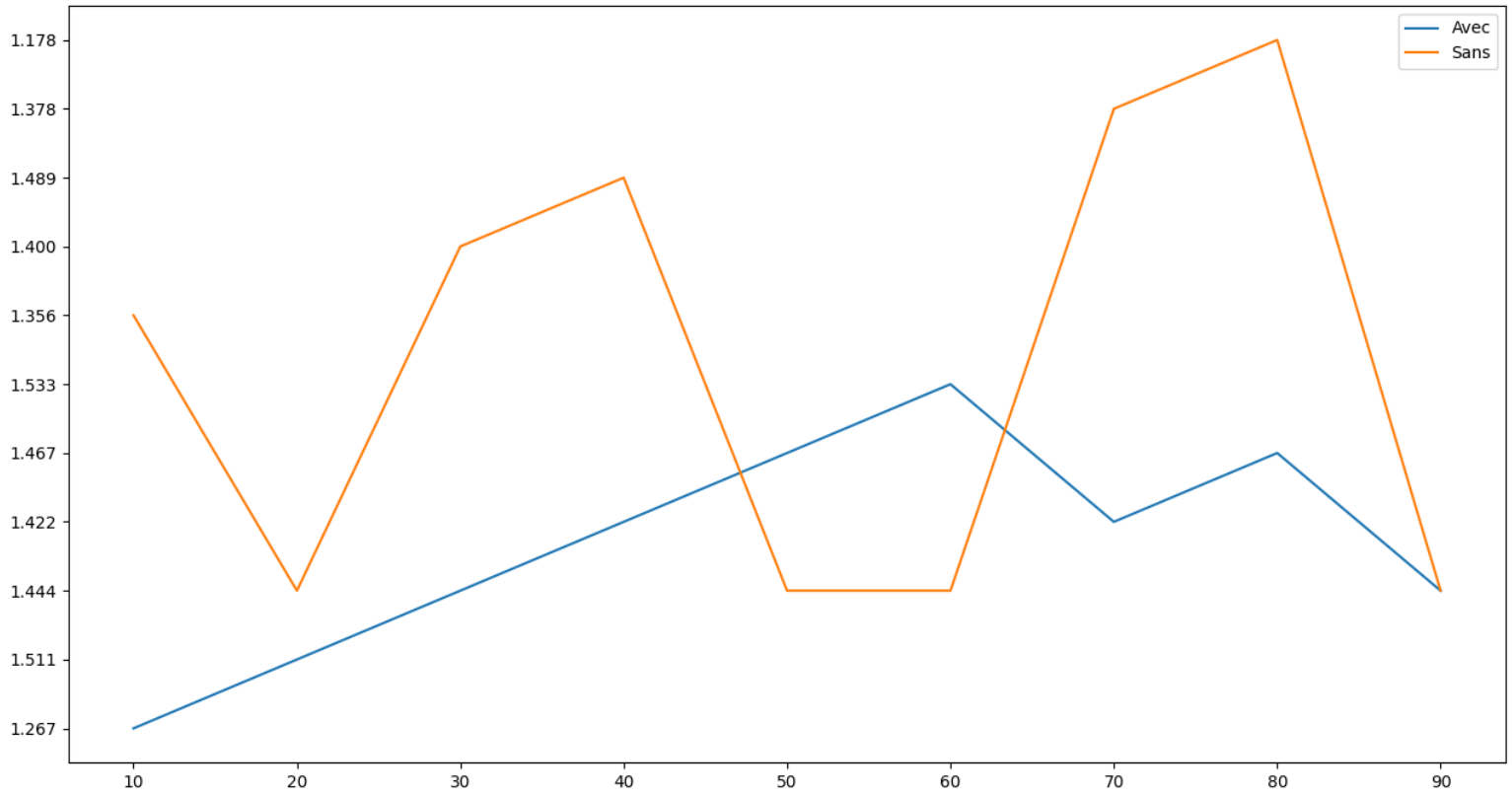
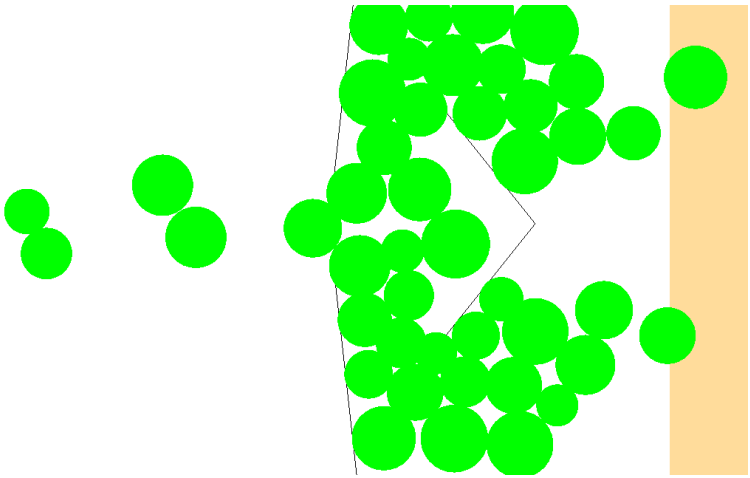
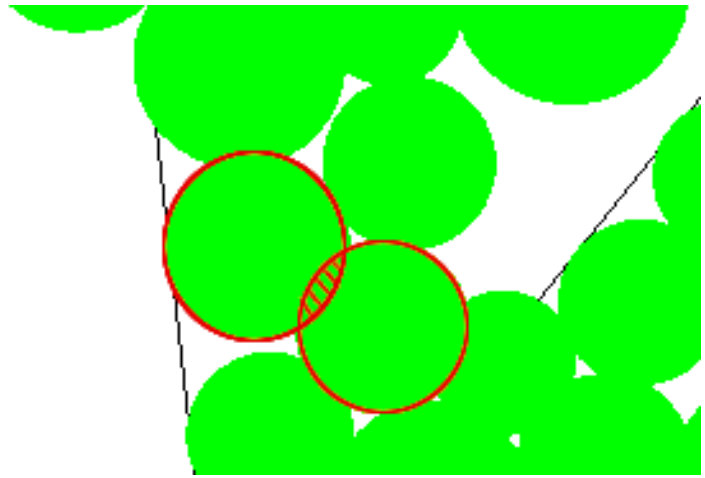


FIGURE – Résultats

Simulation, résultats

Résultats





Simulation, résultats

Amélioration des collisions

Avant :

- Les collisions piéton-mur et piéton-piéton sont représentées par des forces
- Ces forces sont ensuite intégrées : $\overrightarrow{F_{murs}} + \overrightarrow{F_{piétons}} + \overrightarrow{F_{trajectoire}} = m\overrightarrow{a_i}$

Après :

- Les collisions sont résolues directement
- Si deux piétons se traversent, leurs positions sont immédiatement rectifiées, sans passer par $\sum \overrightarrow{F_i} = m\overrightarrow{a}$

Simulation, résultats

Résultats

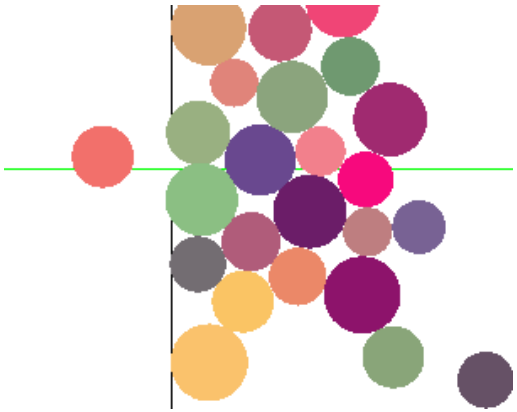


FIGURE – Les problèmes de collisions sont réglés

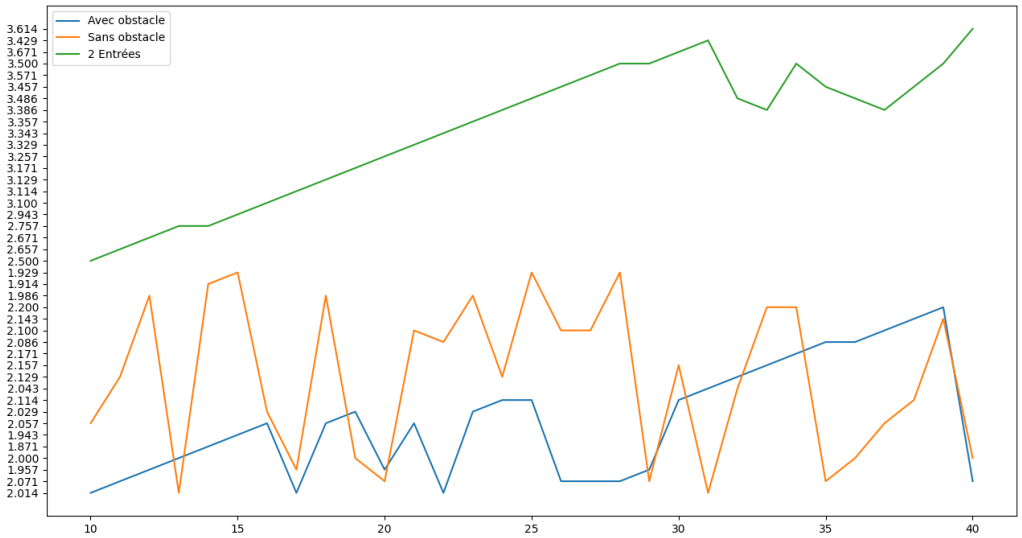
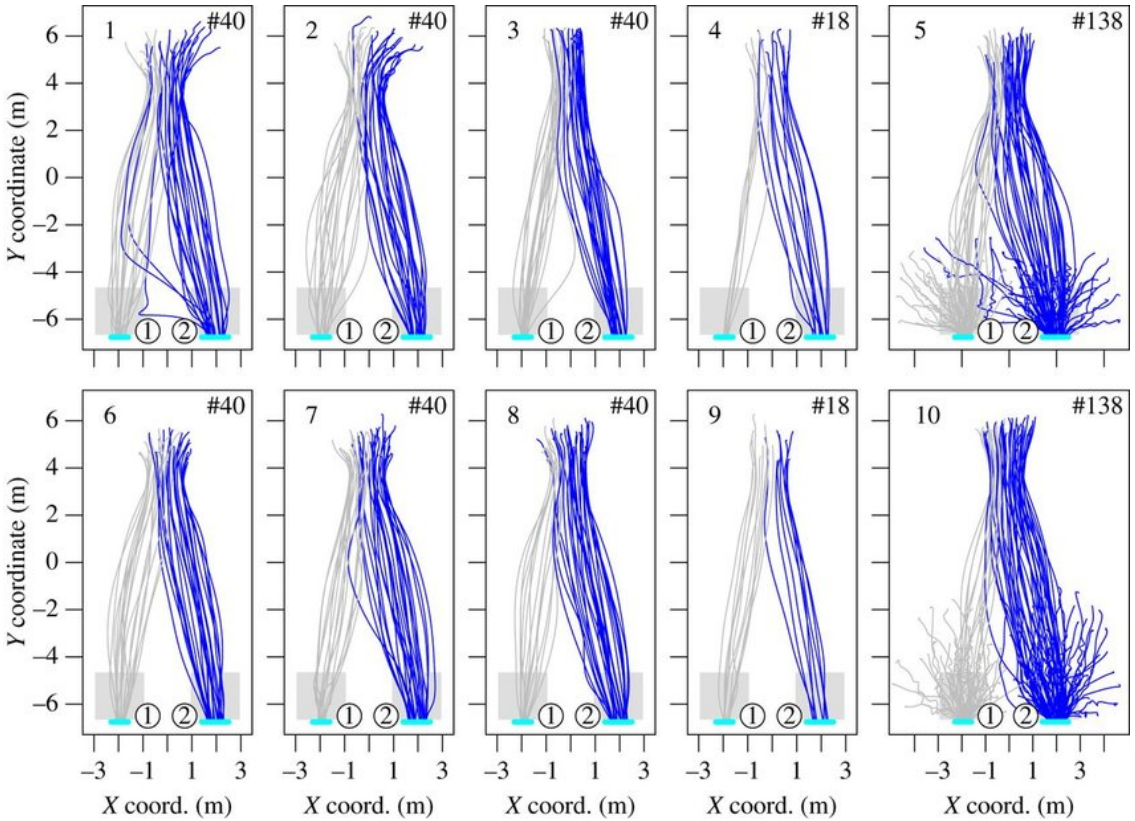


FIGURE – résultats après

Deuxième validation : trajectoires

Expériences empiriques

Understanding human queuing behaviour at exits : an empirical study



Deuxième validation : trajectoires

Prédictions du modèle

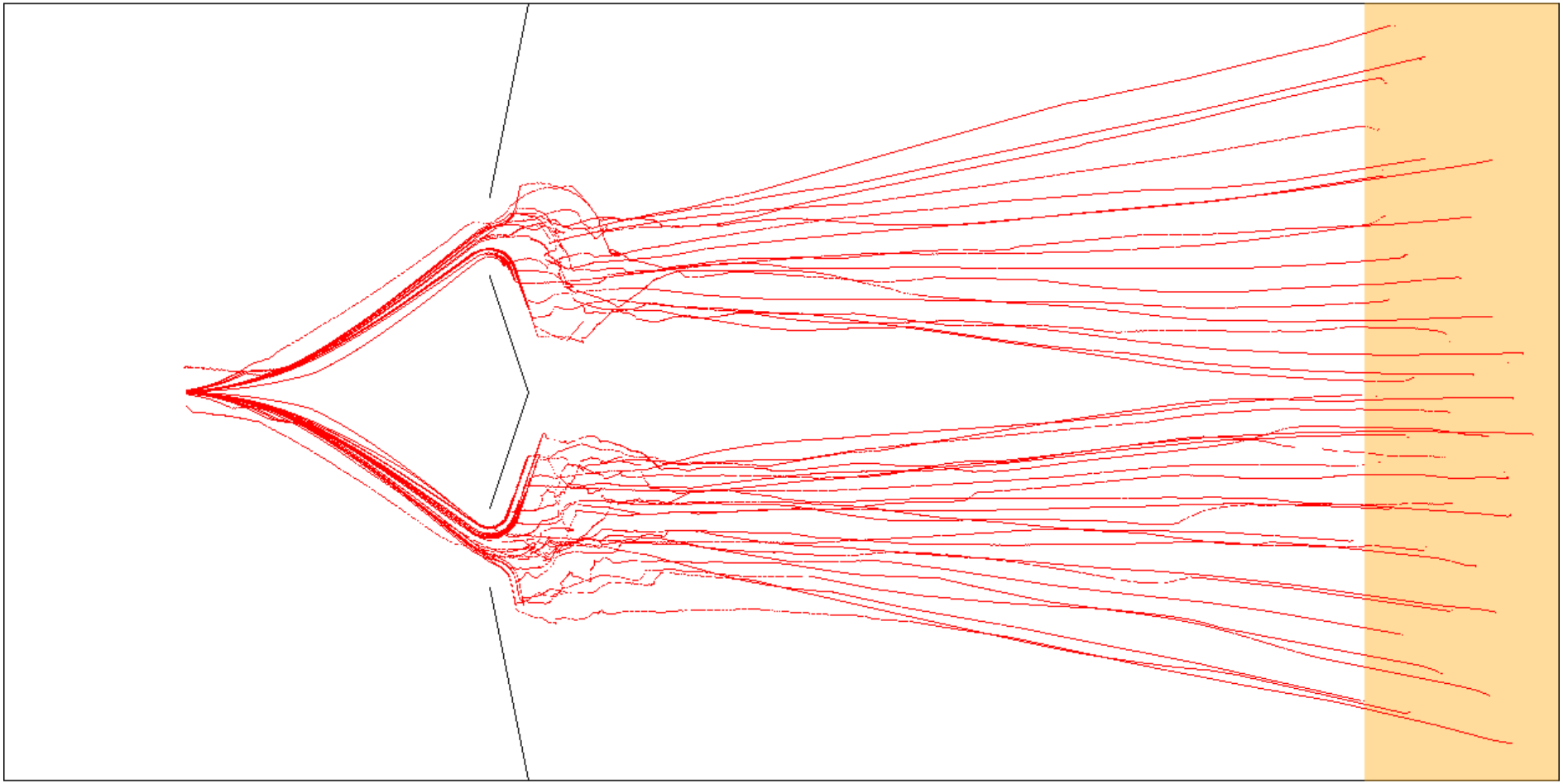


FIGURE – Beaucoup de piétons restent bloqués aux extrémités des murs

Deuxième validation : trajectoires

Prédictions du modèle

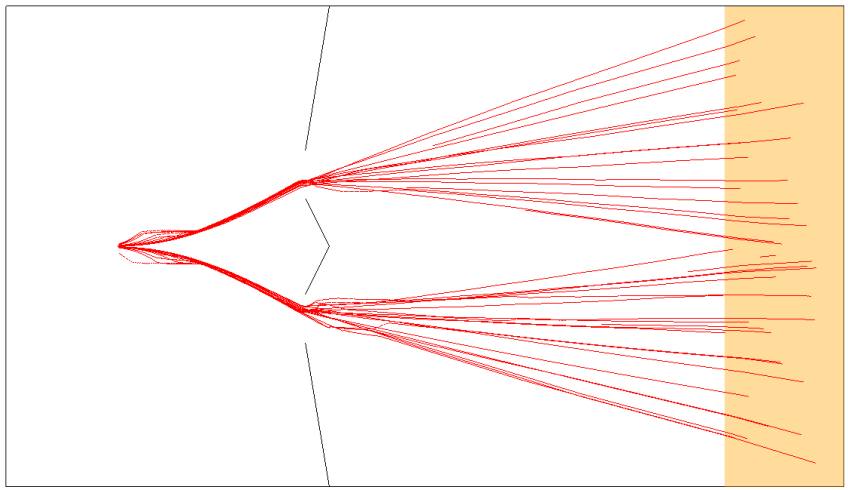


FIGURE – Après

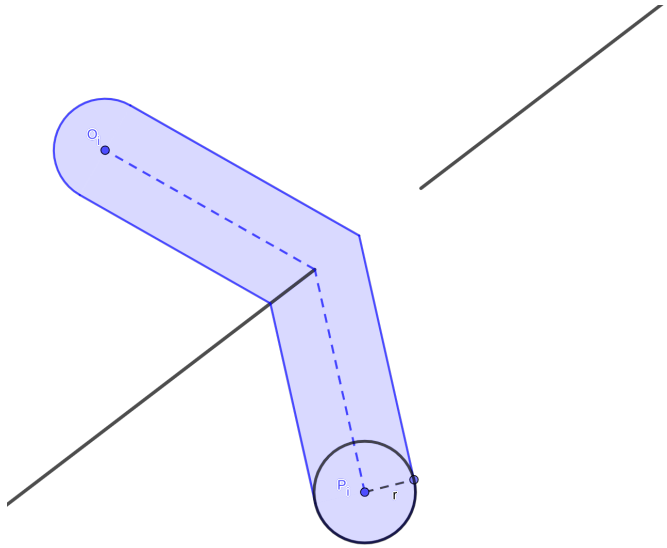


FIGURE – Le problème

Généralisation

Graphe

Comment appliquer ce modèle sur des bâtiments plus complexes ?

- Une ou plusieurs zones d'apparition
- Un graphe orienté
- Un noeud de sortie
- On calcul à chaque instant la densité de piétons autour d'un noeud donné
- Lorsque le piéton doit choisir entre deux noeuds, il choisit de se diriger vers le moins dense

```
 1 1
 2 5 0 1 4 1 0
 3 11
 4 0 0 6 0
 5 0 4 6 4
 6 0 0 0 4
 7 6 0 6 4
 8 3.5 0 3.5 1.8
 9 3.5 2.2 3.5 4
10 4 1.7 4.2 2.0
11 4.2 2.0 4 2.3
12 3.8 1.9 3.8 2.1
13 3.8 1.9 4 1.7
14 3.8 2.1 4 2.3
15 2 1
16 5.5 2 1 1
17 0.5 2 1 1
```

FIGURE – Représentation informatique du niveau

Généralisation

Graphe

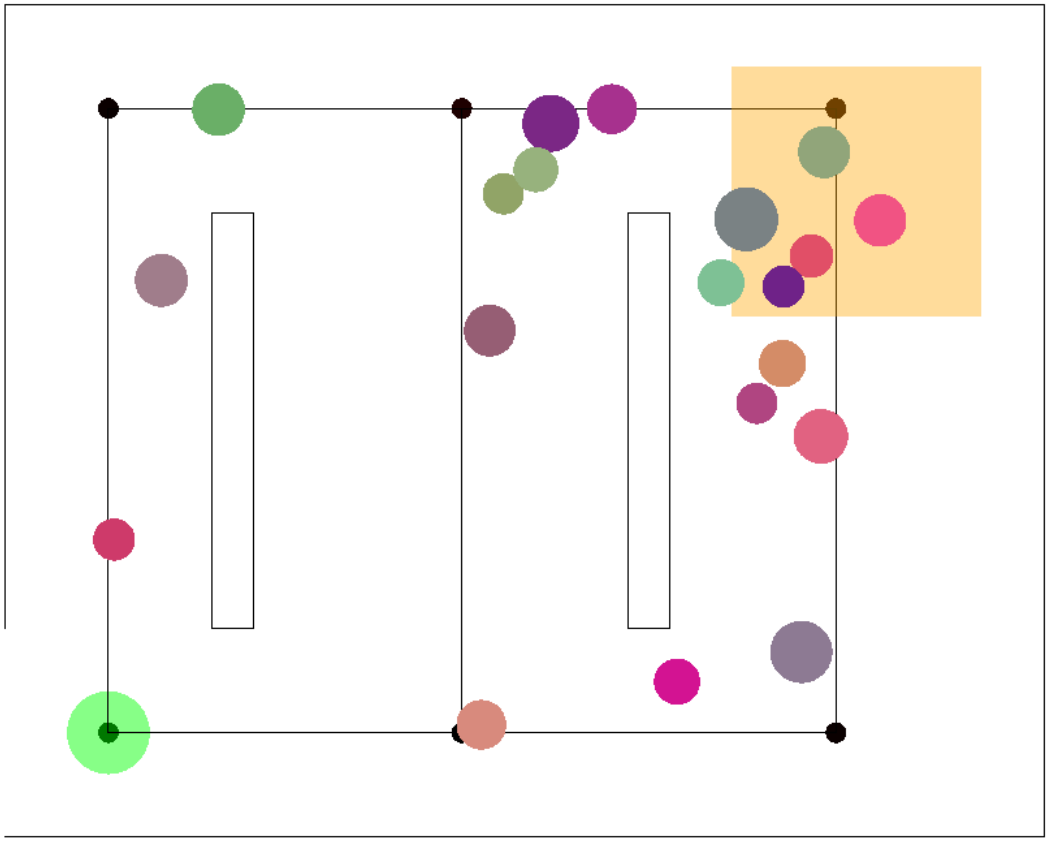


FIGURE – Simulation plus complexe avec un graphe

Conclusion

La suite :

- Calculer les forces de pression sur chaque piéton
- Calculer une carte de pression pour connaître les zones «à risques»
- Ajouter un éditeur de niveau graphique pour une utilisation plus simple du «logiciel»