

# Compte-rendu du projet pluridisciplinaire

Étude du comportement collectif et d'évitement d'obstacles  
dans un essaim

Adrien PANGUEL, Ahlam KHAZZAR, Alain MOUAWAD,  
Kareem ABI KAEDBEY, Karim HECHEIME, Roger DACCACHE

12/19/2024



FIGURE 1 – Wildebeest swarming through rough terrain



FIGURE 2 – Different pedestrian swarms crossing intersection in different directions

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Objectif du premier modèle . . . . .	3
1.2	Objectif du second modèle . . . . .	3
<b>2</b>	<b>Cadrage du sujet</b>	<b>3</b>
2.1	Contexte général, développements historiques . . . . .	3
2.1.1	Éléments d'histoire des sciences et des techniques . . . . .	4
2.1.2	Contexte et enjeux sociétaux . . . . .	5
2.2	Objectifs du travail et étude dans ce projet . . . . .	5
<b>3</b>	<b>Démarche suivie et analyse du code</b>	<b>6</b>
3.1	Création des agents et initialisation de l'environnement . . . . .	6
3.2	Définition des règles comportementales . . . . .	7
3.3	Simulation dynamique . . . . .	8
<b>4</b>	<b>Objectifs des modélisations</b>	<b>9</b>
4.1	Comprendre les dynamiques des groupes et des essaims . . . . .	9
4.2	Analyser les interactions avec l'environnement . . . . .	9
4.3	Appliquer les résultats à des cas concrets . . . . .	10
<b>5</b>	<b>Résultats originaux</b>	<b>10</b>
5.1	Observations et interprétations . . . . .	10
5.2	Analyse et extensions potentielles . . . . .	12
5.3	Retour critique sur notre travail . . . . .	12
5.4	Aspects sécurité informatique . . . . .	13
5.5	Aspects environnementaux . . . . .	13
5.6	Aspects organisationnels . . . . .	14
<b>6</b>	<b>Conclusion</b>	<b>16</b>
<b>7</b>	<b>Crédits et références</b>	<b>16</b>
<b>8</b>	<b>Bibliographie</b>	<b>17</b>
<b>9</b>	<b>Annexe</b>	<b>18</b>

# **Étude du comportement collectif et d'évitement d'obstacles dans un essaim**

## **1 Introduction**

L'objectif de ce projet est d'analyser le comportement collectif d'un essaim d'agents en interaction à l'aide d'un outil de modélisation. Nous cherchons à comprendre comment les règles locales influencent la structure et les mouvements d'un groupe d'agents, et comment ces agents évitent les obstacles, tels que des structures simples immobiles, ou des agents mobiles comme des prédateurs. Nous avons développé avec l'outil NetLogo une série de comportements locaux pour les agents comme la cohésion, l'alignement, la séparation et l'évitement. Le but final est d'examiner comment ces comportements influencent l'ensemble du groupe dans des environnements complexes.

### **1.1 Objectif du premier modèle**

Dans un premier modèle, on souhaite d'étudier un cas particulier de mouvement de foule, l'évacuation d'une bibliothèque. Nous prendrons comme exemple la BU L1 (Figure 3), Bibliothèque Universitaire du L1 de notre université. On cherche à refléter le comportement optimal en vue de garantir la survie du plus grand nombre de personnes et d'assurer une évacuation la plus rapide possible, en fonction des paramètres initialisé à chaque simulation.

### **1.2 Objectif du second modèle**

Pour compléter notre étude du comportement collectif d'un essaim d'agents en interaction, nous avons exploré un second modèle. Nous nous sommes concentrés sur l'analyse de l'impact des règles individuelles sur la structure et des mouvements globaux. En particulier, tout comme pour la première modélisation, on souhaite étudier la capacité des essaims à maintenir leur cohésion tout en évitant les obstacles et les menaces. Ici, les agents sont des poissons qui ne doivent pas croiser le chemin des prédateurs.

## **2 Cadrage du sujet**

### **2.1 Contexte général, développements historiques**

La notion de mouvement collectif se réfère à la modification simultanée d'un groupe d'individus non seulement par rapport à l'extérieur, mais aussi

au sein même du groupe. Ce groupe peut être un essaim d'une certaine espèce d'animaux comme un banc de poissons dans l'océan, ou bien une foule de personnes sur un trottoir.

### 2.1.1 Éléments d'histoire des sciences et des techniques

L'idée de modéliser ces comportements a été inspiré par les travaux de chercheurs comme Craig W. Reynolds. Cet informaticien et infographiste américain pionnier des domaines de la vie artificielle du centre de recherche du groupe Sony Computer Entertainment a introduit le concept de "boids", qui vient de la contraction des mots "bird" et "oid", et qui signifie littéralement "forme d'oiseau". Les boids sont d'abord restreints à la simple analyse du comportement de groupe. En effet, les premiers travaux sur les comportements collectifs remontent dans les années 1950 à des études en biologie comportementale. Des chercheurs en éthologie comme Nikolaas Tinbergen et Konrad Lorenz ont posé les bases de la compréhension des comportements collectifs chez les animaux, en observant les bancs de poissons, les essaims d'insectes et les migrations d'oiseaux. Ils ont étudié l'importance de la communication interne entre individus d'un même groupe via des signaux sensoriels et des interactions locales.

C'est dans les années 1980 que ces concepts ont été appliqués à des systèmes informatiques et à la robotique. Craig W. Reynolds a révolutionné le domaine avec son modèle de "boids". Pour simuler le comportement collectif d'un essaim, il a introduit trois règles simples :

- **Cohésion** : Les boids se rapprochent les uns des autres pour former un groupe.
- **Alignement** : Les boids ajustent leur trajectoire afin de suivre un chemin commun tout en restant groupés.
- **Séparation** : Deux boids ne peuvent occuper la même position simultanément.

Ces règles, bien que locales, produisent des dynamiques globales complexes et émergentes, c'est-à-dire que le comportement global d'un groupe résulte de chacun des comportements individuels. C'est plus tard que les boids sont utilisés pour créer des images de synthèse comme des nombreux films et dessins animés pour modéliser les troupeaux de gnous dans Le Roi Lion (Figure 4), ou le troupeau de Gallimimus dans Jurrassic Park (Figure 5). Au XXIe siècle, l'étude des comportements collectifs s'est enrichie grâce aux avancées en intelligence artificielle et en calcul distribué. Les modèles

inspirés de la nature, tels que ceux des fourmis ou des abeilles, sont maintenant intégrés dans des systèmes modernes d'optimisation et de gestion. En parallèle, les technologies comme les drones autonomes et les robots collaboratifs s'appuient sur ces principes pour accomplir des tâches complexes de manière coordonnée. En Chine par exemple, le 5 septembre 2024, la ville de Shenzhen a été illuminée par plus de 8000 drones lors d'un spectacle organisé par l'entreprise chinoise HighGreat (Figure 6).

### 2.1.2 Contexte et enjeux sociétaux

Les applications pratiques de l'étude des comportements collectifs sont nombreuses :

- **Gestion des foules et la sécurité publique** : Les modèles de mouvements collectifs sont cruciaux pour concevoir des espaces publics comme les stades, les gares et les aéroports, et gérer des situations d'urgence. Récemment, lors des J.O, il a fallu simuler les comportements humains lors de potentielles évacuations pour optimiser les plans d'évacuation et réduire les risques de panique.
- **Optimisation des flux et des réseaux** : Dans les transports, les modèles d'essaim sont utilisés pour améliorer la gestion du trafic urbain et concevoir des systèmes de transport intelligents. En particulier, le monde de l'automobile connaît de grands changements, notamment avec l'arrivée des véhicules autonomes qui pourraient révolutionner les déplacements en réduisant les embouteillages et les accidents, par la simulation en direct du traffic et les prédictions sur les potentiels dangers des embouteillages et accidents.
- **Applications militaires et stratégiques** : Les comportements collectifs sont étudiés pour concevoir des systèmes de défense adaptatifs. Des essaims de drones sont déjà utilisés pour des missions de reconnaissance ou de neutralisation d'obstacles, imitant la coordination d'un essaim naturel.

## 2.2 Objectifs du travail et étude dans ce projet

L'objectif principal de ce projet est de comprendre comment les agents se déplacent dans un environnement complexe tout en conservant les 3 règles de cohésion de groupe, d'alignement des agents à la trajectoire du groupe, et de séparation au sein du groupe. Pour cela, nous avons décidé d'orienter notre analyse par l'étude de deux modèles distincts dont les codes sont en annexe (Figures 7 et 8), et qui se concentrent sur le comportement d'individus dans

un groupe au sein d'un environnement complexe.

Dans un premier modèle, nous étudierons le comportement de tout un groupe de personnes pour atteindre un point géographique donné dans un environnement donné. Tous les agents possèdent une connaissance préalable du trajet optimal pour atteindre la sortie. Cela permet de modéliser un mouvement de foule dit "intelligent", où chaque individu suit un chemin rationnel et préétabli. De plus, chaque agent est caractérisé par un comportement particulier : certains sont patients, et d'autres impatients. Les agents impatients ont donc la possibilité d'avancer et d'enfreindre la règle de séparation, phénomène qui peut arriver dans la vie de tous les jours lors d'une évacuation d'urgence d'une infrastructure quelconque. Notre étude du modèle 1 se divise en deux parties distinctes :

- **Mouvement de foule intelligent dans un environnement connu (Figure 9)** : Nous analysons le comportement des agents évoluant dans un environnement qu'ils connaissent bien, la Bibliothèque Universitaire du L1.
- **Simple évacuation vers une sortie (Figure 10)** : Cette deuxième partie se concentre sur une situation plus générale, une simple évacuation vers un point de l'espace qu'on identifie à une sortie. Cette simplification du modèle 1 permet une meilleure visualisation du phénomène de blocage des individus entre eux.

Dans un second modèle (Figure 11), nous modéliserons un banc de poissons, dont chaque entité suit les mêmes règles de comportement, et qui doit faire face à un événement imprévu, un prédateur, qui peut être déplacé par l'utilisateur à l'aide de la souris.

Nous souhaitons également tester la résistance de ces groupes face à des menaces, comme des prédateurs ou un simple incendie, et évaluer leur capacité à prendre compte ces menaces et à éviter les obstacles.

### 3 Démarche suivie et analyse du code

#### 3.1 Crédit des agents et initialisation de l'environnement

##### Modèle 1

Dans les deux parties de ce modèle, chaque agent est initialisé à une position aléatoire de l'espace, tout comme le départ d'incendie, et les obstacles sont placés à l'avance pour faire correspondre à notre modèle le plan de la BU

L1. Il est possible de choisir le nombre initial d'agents placés aléatoirement, et le nombre de sorties entre (1 et 2) et leur localisation dans la BU L1.

## Modèle 2

Dans ce modèle, l'environnement est initialisé avec un nombre configurable de poissons positionnés au hasard dans l'espace, et un prédateur contrôlé par l'utilisateur à l'aide de la souris.

### 3.2 Définition des règles comportementales

#### Modèle 1

Nos agents possèdent tous une même caractéristique qu'on qualifie de comportement intelligent. Ils peuvent tous prendre le meilleur chemin possible qui leur est donné en temps réel. De plus, les agents disposent de deux types de comportement qui est défini au préalable pour chaque individu :

- **Comportement patient** : Les agents dits 'patients' disposeront d'une jauge de patience, définie avant le lancement de la simulation. Contrairement à d'autres agents, ils ne tenteront pas de bousculer ou de traverser les obstacles, y compris les autres agents. Lorsqu'ils rencontrent un autre agent, ils patienteront pendant une durée spécifique, déterminée par leur jauge de patience, avant d'essayer de le contourner.
- **Comportement impatient** : Ces agents n'hésitent pas à bousculer les autres agents devant eux et à avancer sans attendre pour sortir, en écrasant les agents devant eux, causant leur mort.

#### Modèle 2

Chaque agent possède des propriétés spécifiques :

- Les poissons ont des voisins identifiés comme *flockmates* et un *nearest-neighbor*, et obéissent à des règles comportementales locales.
- Le prédateur influence directement par sa position les trajectoires des poissons.

En plus des règles de cohésion, d'alignement, et de séparation, nous avons implémenté un autre comportement pour les poissons : la fuite. Lorsqu'un prédateur est détecté, les poissons s'éloignent dans la direction opposée pour éviter une menace, tout en respectant les trois autres règles. Ces comportements reposent sur des calculs géométriques, tels que les distances et les angles, pour déterminer les interactions locales et guider les mouvements. Le

prédateur quant à lui ne suit pas de règles comportementales dans la mesure où c'est nous, utilisateurs, qui le déplaçons dans le cadre avec la souris.

### 3.3 Simulation dynamique

Pour modéliser ces deux modèles complexes, nous avons utilisé la plate-forme de simulation NetLogo. Cet environnement de programmation multi-agents fut développé par Uri Wilensky en 1999 au Center for Connected Learning and Computer-Based Modeling de l'Université Northwestern. Il utilise un langage de programmation simple accessible à tous et contient des bibliothèques de modèles permettant de modéliser un grand nombre de comportements dans tous les domaines de recherche scientifique.

#### Modèle 1

Dans le modèle 1, nous utilisons plusieurs procédures :

- **setup** : Initialise la position des obstacles, pose les agents aléatoirement dans le cadre prédéfini, et donne une position de départ au feu. Ce bouton modélise donc la BU des L1 remplie, juste avant le début de la propagation du feu.
- **go** : Met à jour le modèle qui fait avancer les agents selon les règles préétablies et provoque la propagation du feu à chaque tick.
- **population** : Contrôle le nombre initial de personnes dans la bibliothèque.
- **death-counter/free-counter** : Compte le nombre de personnes décédées dans la bibliothèque, et celles qui ont pu s'échapper.
- **nb-sorties** : Permet de choisir entre 1 ou 2 sorties.
- **patientes/impatientes** : Initialise le nombre de personnes au caractère patient ou impatient.
- **patience** : Permet de définir le taux de patience des agents patients, pour permettre à certains de changer de comportement patient à impatient dans certaines situations.

Ainsi, il est possible de simuler l'évacuation de la bibliothèque L1 tout en changeant les paramètres directement grâce à ces boutons d'initialisation.

#### Modèle 2

Le modèle 2 demande d'autres procédures en plus du "setup" et du "go" :

- **setup** : Initialise l'environnement en plaçant les poissons et le prédateur dans des positions aléatoires.

- **go** : Met à jour du modèle qui fait avancer les agents selon leur comportement à chaque tick.
- **population** : Contrôle le nombre initial de poissons dans l'espace
- **vision** : Détermine la distance maximale en-dessous de laquelle les capteurs sensoriels de chaque poisson détecte un congénère ou un obstacle.
- **maximum separation** : Fixe la distance maximale entre 2 poissons.
- **max-align-turn** : Fixe l'angle maximal d'alignement de chaque poisson avec sa trajectoire.
- **max-cohere-turn** : Fixe l'angle maximal de cohésion des poissons entre eux.
- **max-separate-turn** : Fixe l'angle maximal de séparation de chaque poisson à un obstacle.

Les poissons combinent les règles comportementales pour ajuster leur mouvement en douceur, tandis que le prédateur, contrôlé par la souris, sert à tester les capacités d'évitement de l'essaim.

## 4 Objectifs des modélisations

### 4.1 Comprendre les dynamiques des groupes et des essaims

- Étudier les interactions locales et leurs effets sur la formation d'ensembles organisés.
- Explorer les effets des rétroactions locales sur les dynamiques globales.
- Observer l'émergence de structures stables ou chaotiques nées de l'apparition d'évènements inattendus.

### 4.2 Analyser les interactions avec l'environnement

- Simuler des scénarios d'évitement d'obstacles, fixes ou mobiles, pour évaluer l'efficacité des règles comportementales.
- Explorer l'impact des paramètres individuels tels que le rayon de vision ou les seuils de séparation sur l'organisation globale.
- Comparer les stratégies basées sur des distances métriques et topologiques.

### **4.3 Appliquer les résultats à des cas concrets**

Les applications possibles incluent :

- Biologie et écologie : Simulation des migrations animales ou des réactions face au prédateur.
- Transports et gestion de flux : Optimisation des mouvements de foules ou des convois de véhicules autonomes.
- Robotique collaborative : Essaims de robots autonomes pour des missions de recherche ou de sauvetage.
- Entertainment : Développement d’algorithmes inspirés des comportements d’essaim pour des représentations lumineuses (Shenzhen).

## **5 Résultats originaux**

### **5.1 Observations et interprétations**

#### **Modèle 1**

Dans le premier modèle, les agents se dirigent vers les patch rouges qui symbolisent les sorties. Le modèle de la BU L1 est fiable et réaliste, et nous donne une idée globale du comportement d’une foule dans le cas d’un incendie. Toutefois, c’est le modèle simplifié de la simple évacuation sans obstacle matériel qui met l’accent sur l’impact des comportements individuels sur tout le groupe. En effet, on observe plus facilement l’adaptation des individus à leur environnement :

- Les agents évitent les obstacles matériels modélisés en jaune.
- Quand les agents sont patients et qu’un autre individu se trouve devant eux, ils attendent jusqu’à un certain seuil où leur comportement change et ils deviennent impatients.
- Dans le cas où toutes les sorties sont prises par les flammes, les agents arrêtent de bouger et attendent de mourir des flammes.
- Les agents qui sont soit morts des flammes, soit morts écrasés, arrêtent de bouger.

D’une part, nous constatons que le nombre de sorties joue un rôle crucial lorsque le nombre de personnes présentes dans la bibliothèque est élevé, notamment lorsque le feu s’est répandu jusqu’à une des deux sorties. Dans le cas d’une unique sortie, les agents s’arrêtent de bouger quand celle-ci est couverte par les flammes. Avec une deuxième sortie, non seulement il existe

toujours une autre option quand l'une d'elle est bloquée, mais elle facilite également une évacuation plus rapide des agents, dans la mesure où elles sont éloignées l'une de l'autre et peuvent diminuer le temps d'évacuation des individus. D'autre part, la modification des paramètres de base comme le nombre d'agents patients/impatients, et le degré de patience, a un impact significatif sur la dynamique globale de la foule. En effet, avec un taux de patience très faible, on se retrouve très rapidement dans une situation où tous les agents sont impatients, ce qui engendre un hausse des décès par écrasement. Les agent sont plus pressés en moyenne et n'hésitent pas à se marcher dessus.

## Modèle 2

Ce second modèle représente un banc de poisson qui se déplace dans l'espace tout en évitant le chemin du prédateur, qu'il soit immobile, ou dirigé par la souris. Sa position influence le comportement des poissons. Quand on place par exemple le prédateur sur la trajectoire d'un groupe de poissons, chaque individu va corriger sa trajectoire en tenant compte du "max-separate-turn" et va essayer d'éviter le prédateur tout en respectant les règles de cohésion et d'alignement paramétrées par le "max-cohere-turn" et le "max-align-turn". Nous observons aussi que les comportements locaux influencent directement la structure globale de l'essaim. En effet, quand une grande "distance-maximale" ou un faible champ de vision est imposée à chaque agent, cela entraîne des essaims dispersés, tandis qu'une faible "distance-maximale" et un large champ de vision favorise des groupes compacts.

Malgré le dérèglement de la structure des groupes de poissons causée par la présence d'un prédateur, on constate que les règles préétablies permettent de maintenir une structure stable et organisée. Cette modélisation appliquée à la biologie est fiable et peut être réutilisée dans un certain contexte, à savoir celui d'un groupe d'agents en présence d'un ou plusieurs obstacles fixes ou mobiles. Ce modèle peut en plus être transposé à d'autres situations comme à un essaim d'oiseau en vol, un troupeau de moutons attaqué par une meute de loups, ou une synchronisation de drones en volants.

En somme, les deux modèles que nous avons développés remplissent les objectifs que nous nous sommes posés en début de projet. Nos résultats montrent que des règles simples peuvent produire des comportements complexes, et que les interactions locales influencent les structures globales. En jouant sur les paramètres d'initialisation des agens et de l'environnement,

nous avons pu appliquer nos résultats à des cas concrets comme l'évacuation d'incendie et la survie d'espèce en groupe. Toutefois, il est nécessaire de procéder à quelques ajustements pour simuler des environnements encore plus complexes et ainsi étendre nos modèles à des applications plus variées.

## 5.2 Analyse et extensions potentielles

### Modèle 1

En outre, bien que le mouvement des personnes puisse être qualifié d'intelligent, dans la mesure où elles empruntent généralement un chemin quasi optimal, ce processus demeure perfectible. En effet, les comportements patient et impatient pourraient être davantage détaillés afin de mieux refléter les réactions réelles des individus lors d'une évacuation. Il est aussi important de souligner que le comportement humain ne se limite pas à ces deux catégories car d'autres comportements peuvent également être envisagés. De plus, les individus peuvent présenter des caractéristiques variées telles que l'âge, la force ou la vitesse, qui impacteraient leur manière d'agir dans une situation d'urgence.

### Modèle 2

Ce second modèle est plus complet que le premier, mais il pourrait être encore plus généralisable à toute situation, notamment avec l'ajout d'obstacles fixes pour simuler des environnements complexes. Nous pourrions aussi faire une analyse comparative entre des mesures métriques, les distances physiques, et des mesures topologiques comme le nombre de voisins considérés pour améliorer la fiabilité et le réalisme de ce modèle. Enfin, nous pourrions ne pas nous limiter à un unique prédateur déplacés avec la souris, mais à plusieurs prédateurs. Leur déplacement aléatoire serait régi par un certain comportement prenant en compte la distance avec un poisson, le nombre de poissons du groupe qu'ils poursuivent, voire même munis d'une intelligence stratégique consistant à entourer un banc de poissons pour avoir plus de chance d'en attraper.

## 5.3 Retour critique sur notre travail

Au cours de la réalisation de notre projet, nous sommes parvenus à rendre un travail complet, des modèles fiables et réutilisables, et des codes lisibles pour toute personne extérieure au groupe de travail. Nous avons su décomposer notre travail en plusieurs petites tâches qui nous ont permis d'atteindre nos objectifs initiaux à savoir analyser le comportement collectif d'un

essaim d'agents en interaction à l'aide d'un outil de modélisation, et examiner comment ces comportements influencent l'ensemble du groupe dans des environnements complexes.

Cependant, nous avons rencontré des difficultés. Nous avons eu du mal au début à nous organiser et à nous répartir les tâches pour avancer efficacement dans notre travail parce que nous n'avons pas l'habitude de travailler aussi nombreux en groupe de 6 et parce qu'il y avait de nombreux objectifs à atteindre. Nous avons aussi eu besoin d'un certain temps pour nous familiariser avec le logiciel Netlogo, bien qu'il soit accessible à tous. On a donc été confronté à un certain nombre de problèmes de modélisation, liés à des erreurs de logique dans notre raisonnement informatique, ce qui a ralenti notre progression, mais nous a aidé à perfectionner notre rapport au logiciel.

Enfin, nous nous sommes parfois appuyés sur les IA ChatGPT et Copilot pour comprendre des lignes de codes trouvées sur internet, ou même pour améliorer un de nos propres codes. Cependant, ce n'était pas évident de transmettre à l'IA l'idée précise de notre démarche. Par exemple, pour permettre aux agents d'éviter les obstacles, l'IA ne pouvait pas lancer la simulation des codes qu'elle nous proposait. Elle ne pouvait pas tenir compte à la fois de l'emplacement des sorties dans le cadre, des obstacles, et de la position du feu. C'est pourquoi nous ne l'avons utilisée uniquement pour trouver des idées de modélisation, par exemple comment tracer un losange, comment faire un feu qui se répand de manière non carré et plus aléatoire, ou encore comment ajouter des options de panique.

## 5.4 Aspects sécurité informatique

Notre travail respecte les principes généraux de précaution vis-à-vis du numérique :

- Aucune de nos données personnelles n'a été utilisée.
- Nos algorithmes, les commandes, et les fonctions que nous avons définies peuvent être compréhensibles, vérifiées, et réutilisées.
- Nous avons cherché à optimiser nos codes pour réduire l'impact environnemental.

## 5.5 Aspects environnementaux

Notre projet respecte les principes généraux relatifs aux enjeux écologiques et environnementaux en minimisant la consommation énergétique grâce à

l'optimisation de nos codes et à l'utilisation du logiciel Netlogo. Sa simplicité nous a permis de limiter le nombre d'erreurs que nous avons pu commettre et ainsi diminuer le nombre de simulations des codes. Bien que l'impact environnemental de ces simulations soit faible car il n'impacte pas directement le monde réel, nous avons essayé d'implémenter le moins de boucles possibles mais ce n'était pas évident. Toutefois, une évaluation précise de l'impact environnemental pourrait être envisagée à l'avenir, ainsi que l'exploration d'infrastructures alimentées par des énergies renouvelables.

## 5.6 Aspects organisationnels

Nous avons commencé chacun de notre côté à mener nos propres recherches sur le phénomène des mouvements de foule et le comportement des groupes face à l'inattendu. Nous avons aussi visionné des tutoriels sur l'utilisation de Netlogo. Ensuite, nous avons décidé de nous scinder en deux équipes de travail, chacune travaillant sur un modèle différent, les étudiants en BU L1 et les poissons dans l'eau, mais nous avons rapidement constaté que ce n'était pas la bonne manière de fonctionner. Finalement, nous avons convenu de tous nous concentrer sur le premier modèle de la BU L1 pour avancer plus vite tous ensemble. Nous avons commencé à modéliser juste des agents et une sortie au centre (Figure 12), puis nous nous sommes fixés plusieurs objectifs et nous avons avancé pas-à-pas :

- Avec en plus le losange central (Figure 13)
- Puis nous avons essayé diverses méthodes d'évitement des obstacles, soit avec un nouveau cap à suivre, soit avec une distance minimale à conserver avec les autres agents, qui finalement s'est traduit par coder les bords des obstacles pour diriger les agents qui les touchaient dans la bonne direction, restreignant davantage le code à cet unique modèle.
- Avec en plus le feu qui se propage (Figure 14)
- Puis nous avons été amené à nous demander comment modéliser la panique, nous l'avons fait en obligeant chaque agent effectuer à chaque agent un mouvement dans une direction aléatoire tous les 3 mouvements.
- Puis nous avons ajouté l'option des bousculement et de la distance minimale de sécurité entre les agents
- Puis nous avons essayé d'implémenter une jauge de panique qui deviendra enfin le comportement de patience/impatience (Figure 15)
- Puis nous avons fait en sorte que les agents qui sont proches du feu trop longtemps soient comme asphyxiés et meurent (Figure 16)

Après toutes ces complexifications en vue d'une amélioration du réalisme de notre modèle, nous nous sommes rendus compte que l'interaction des agents entre eux n'était pas assez claire, donc nous avons choisi de faire en parallèle un modèle simplifié de l'évacuation d'incendie pour ne se concentrer que sur l'impact des comportements individuels sur toute la structure. Nous avons donc repris les comportements essentiels : le cap vers les patch rouges (sorties), les comportements patient et impatient des agents, le feu qui tue, et les bousclements qui tuent.

Grâce à cette spécification de notre analyse, nous avons pu mieux cerner l'impact des comportements locaux sur les dynamiques globales. Toutefois, au cours de notre étude, l'ajout de règles supplémentaires venait modifier le comportement des agents en fonction des complexités précédentes. Par exemple, lorsque nous avons développé une méthode d'évitement des obstacles pour le losange central, ils se trouvent qu'elle ne fonctionnait pas pour tous les obstacle rectangulaires, ce qui nous a amenés à coder chaque contour d'obstacle. Ou encore, quand nous avons ajouté l'option de panique, certains agents traversaient les murs et les obstacles, et oubliaient les procédures qui devaient les guider jusqu'aux sorties. Nous avons donc été contraints de modifier ce mouvement aléatoire en ne ciblant que des cases vides différentes des obstacles. Tout au long de ces simulations, nous avons ajouté des compteurs "death-count" et "free-count" qui nous indiquent le nombre d'agents sauvés et tués, pour assurer ou non la validité de notre modèle.

Une fois le modèle 1 bien avancé, nous avons constaté que le temps passait, donc nous nous sommes divisés en deux groupes, pour non seulement finaliser l'analyse du modèle 1, mais aussi commencer l'étude d'un second modèle, celui du banc de poissons.

Nous avons été très efficace pour modéliser la situation d'un banc de poisson qui évite un prédateur. Après quelques recherches sur internet et quelques simulations, nous avons trouvé un code qui nous permettent de ne pas être pasif lors de l'exécution du programme en contrôlant la trajectoire du prédateur avec la souris. Nous avons ainsi pu observer et modéliser plus facilement les situations que nous souhaitions étudier. Nous avons par exemple placés le prédateur immobile au centre du cadre en définissant à 0 le champ de vision des poissons. Puis nous avons choisi de tester le modèle pour certains paramètres donnés afin d'observer le comportement des individus d'abord, et ensuite du groupe, et de constater que les comportements individuels ont effets très impactant sur l'allure du groupe.

## 6 Conclusion

Nous avons définit deux modèles où les agents suivent des règles qui guident leur comportement individuel et impactent les comportements globaux des groupes auxquels ils appartiennent. Le changement de direction d'un seul poisson ou étudiant pousse tous les autres autour de lui à recalculer sa trajectoire et à prévoir son prochain déplacement. Nous avons étendu nos deux modèles de manière à simuler l'évitement des obstacles par l'essaim, que cet obstacle soit immobile comme les tables de la BU L1, ou mobile comme le prédateur dans l'eau. Enfin, nous avons analysé comment différents modèles basés sur différentes règles conduisent à différents comportements d'évitement d'obstacles au niveau de l'essaim et modifier sa forme. Les poissons comme les étudiants ont tendance à s'aligner les uns derrière les autres quand c'est possible.

## 7 Crédits et références

Pour la partie prise de référence et familiarisation avec le sujet, nous nous sommes inspirés des travaux de Reynolds sur le modèle des "boids", et de M. Ballerini et al intitulé "Interaction ruling animal collective behavior depends on topological rather than metric distance : Evidence from a field study", PNAS, Vol 105, No 4, Jan 2008, pp 1232-1237.

L'étude menée par Ballerini et al. explore les règles d'interaction entre les étourneaux et l'impact sur le comportement collectif du groupe d'individus. L'analyse se base sur des reconstitutions 3D de la position des oiseaux, et arrive à la conclusion que les groupes dépendent de la distance topologique qu'est le nombre fixe de voisins interactifs, plutôt que de la distance métrique qu'est la distance physique. Cela nous a fait comprendre que notre étude se ramenait en réalité à étudier un unique comportement, celui d'un individu, et d'observer les changements de dynamique globale induits par l'adaptation de cet unique individu à son environnement. A partir de leurs résultats, Ballerini et al. en ont conclut que les modèles topologiques surpassent les modèles métriques car ils maintiennent la cohésion des groupes après des attaques simulées sur le groupe d'étourneaux ou l'apparition d'obstacles.

De là nous est venu l'idée d'adapter ce modèle à une évacuation d'étudiants dans un espace fermé. Nous avons ensuite été très inspirés par une modélisation semblable mais tout de même bien différente de la notre : "crowd-evacuation-simulation" de Shingkid sur Github (Figure 17). Nous avons compris comment le code était fait, et nous avons essayé à notre manière de procéder de la même façon pour le modèle de la BU L1.

Pour le modèle 2, nous nous sommes inspirés d'une librairie déjà existante sur Netlogo appelée "Flocking". Elle modélise des agents qui se déplacent dans l'espace et qui sont paramétrés par une population initiale, un champ de vision, une séparation minimale, et trois angles d'alignement, de cohésion, et de séparation. Pour complexifier le modèle et surtout avoir une chance d'observer le comportement individuel et du groupe face à un évènement inattendu, nous avons cherché sur internet une façon d'ajouter un prédateur et nous sommes tombés sur le modèle "fish-boids" de Paramjeet2810 sur Github. Il ajoute au modèle disponible sur Netlogo en librairie des prédateurs de 1 à 5, qui chassent les poissons et qui sont paramétrés par un coefficient de séparation entre eux, un rayon de séparation entre eux, et un rayon de vision.

## 8 Bibliographie

Voici les liens vers nos sources internet :

Le modèle d'évacuation incendie d'un bâtiment qui nous a inspiré pour le modèle 1, [crowd-evacuation-simulation](#), de shingkid, Github.

Un autre code qui nous a aidé, <https://github.com/armostafizi/EvacuationModel>, d'armostafizi, Github.

[Flocking Algorithm: Simulating Collective Behavior in Nature-inspired Systems](#), Pramodaya Jayalath.

Flock, D. (2015), [Flocking: A simple model of collective behavior](#), Journal of Computational Physics, 300, 1-12.

## 9 Annexe



FIGURE 3 – Plan BU L1



FIGURE 4 – Troupeau de gnoux du Roi Lion 1



FIGURE 5 – Troupeau de Gallimimus de Jurrasic Park

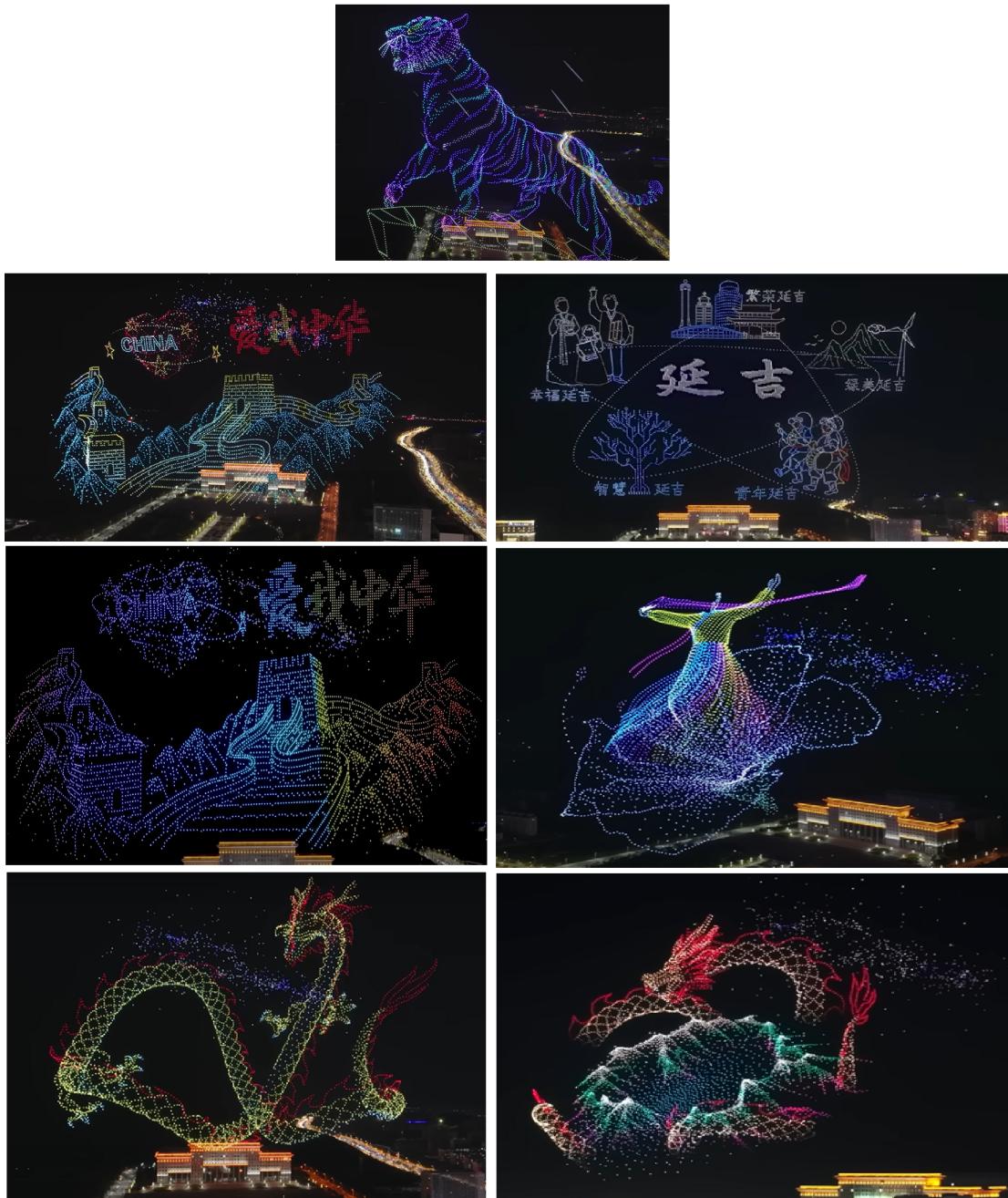


FIGURE 6 – Synchronisation de drones dans le ciel de Shenzhen, Chine pour un spectacle nocturne

```

globals [
  death-counter ; Counts dead turtles
  free-counter ; Counts free turtles
  walk-counter
]

turtles-own [
  life ; Life of each turtle
  impatient? ; True if person is impatient, and
  false if not
  waiting-time
  patience
]

to setup
  clear-all ; Resets space
  reset-ticks ; Resets time

  ; Defines exits as red patches
  ask patches with [pxcor = -20 and pycor <= 22
  and pycor >= 18] [
    set pcolor red
  ]
  if nb-sorties = 2 [ ; Defines second exit if set up
    ask patches with [pxcor = 39 and pycor >= -22
    and pycor <= -18] [
      set pcolor red
    ]]
  ; Draws obstacles
  ask patches with [ ; Middle diamond
    abs(pxcor) + abs(pycor) <= 10
  ][
    set pcolor yellow
  ]

  ; Draws tables and all obstacles
  draw-rectangle -19 -10 10 30
  draw-rectangle-exit
  draw-rectangle-verti-h -1 1 18 32
  draw-rectangle-verti-h 5 7 18 32
  draw-rectangle-verti-h 11 13 18 32

  draw-rectangle-hori-h 18 35 30 32
  draw-rectangle-hori-h 18 35 24 26
  draw-rectangle-hori-h 18 35 18 20
  draw-rectangle-hori-b-d 18 35 12 14
  draw-rectangle-hori-b-d 18 35 4 6

  draw-rectangle-hori-b-d-2 18 35 -8 -6
  draw-rectangle-hori-b-d-2 18 35 -16 -14
  draw-rectangle-hori-b-d-2 18 35 -22 -20
  draw-rectangle-hori-b-d-2 18 35 -28 -26
  draw-rectangle-hori-b-d-2 18 35 -34 -32

  draw-rectangle-verti-b 11 13 -34 -20
  draw-rectangle-verti-b 5 7 -34 -20
  draw-rectangle-verti-b -1 1 -34 -20
  draw-rectangle-verti-b -7 -5 -34 -20
  draw-rectangle-hori-b -20 -11 -34 -32
  draw-rectangle-hori-b -20 -11 -28 -26
  draw-rectangle-hori-b -20 -11 -22 -20

  draw-rectangle-hori-b -35 -24 -34 -32
  draw-rectangle-hori-b -35 -24 -28 -26
  draw-rectangle-hori-b -35 -24 -22 -20
  draw-rectangle-hori-b -35 -24 -15 -13
  draw-rectangle-hori-b -35 -24 -9 -7
  draw-rectangle-hori-b -35 -24 -3 -1
  draw-rectangle-hori-b -35 -24 3 5

  draw-rectangle -35 -25 16 34

  ask patches with [
    pxcor > -35 and pxcor <= -25 and pycor = 16] [
    set pcolor yellow + 2]
  ask patches with [
    pxcor = -35 and pycor >= 16 and pycor < 30 ][
    set pcolor yellow - 2]
  ask patches with [
    pxcor = -35 and pycor >= 30 and pycor < 34 ][
    set pcolor yellow - 1]

  ; External walls
  ask patches [
    if (pxcor = 40 or pxcor = -40 or pycor = 40 or
    pycor = -40) [
      set pcolor green
    ]
  ]

  ; Creates people
  create-turtles impatient [
    set color red ; People color
    set size 2 ; People size
    set life 100 ; People life
    set impatient? true
  ]

```

FIGURE 7 – Code modèle 1

```

move-to one-of patches with [
  not any? turtles-here and
  pcolor = black and
  pxcor != 40 and pxcor != -40 and
  pycor != 40 and pycor != -40
]
]
create-turtles patients [
  set color blue ; People color
  set size 2 ; People size
  set life 100 ; People life
  set patience turtle-patience
  set impatient? false
move-to one-of patches with [
  not any? turtles-here and
  pcolor = black and
  pxcor != 40 and pxcor != -40 and
  pycor != 40 and pycor != -40
]
]

set death-counter 0 ; Initialization of dead
turtles
set free-counter 0 ; Initialization of free turtles

; Fire
ask patch random-xcor-between -39 39
random-ycor-between -39 39 [
  set pcolor orange
]
end

; Defines random coordinates
to-report random-xcor-between [min-x max-x]
  report random (max-x - min-x + 1) + min-x
end

to-report random-ycor-between [min-y max-y]
  report random (max-y - min-y + 1) + min-y
end

; Launches action
to go
  spread-fire
  ask turtles [
    if color != brown and color != green and life >
    0[
      move-toward-exit]
  ]
]

] ; Movement
to move-toward-exit
  ; Find the nearest exit (red patch)
  let exit min-one-of patches with [pcolor = red]
  [distance myself]
  let check2 patch 39 -20 ; Adaptation to
  number of exits
  if [pcolor] of check2 != red [
    draw-rectangle-hori-b-d 18 35 -8 -6 ; 9
    draw-rectangle-hori-b-d 18 35 -16 -14 ; 11
    draw-rectangle-hori-b-d 18 35 -22 -20 ; 12
    draw-rectangle-hori-b-d 18 35 -28 -26 ; 13
    draw-rectangle-hori-b-d 18 35 -34 -32 ; 14
  ]
  if exit != nobody [
    face exit
    let ahead one-of patches in-radius 3 with
    [pcolor = yellow or pcolor = yellow + 1 or pcolor =
    yellow + 2 or pcolor = yellow + 3 or pcolor =
    yellow - 1 or pcolor = yellow - 2]
    ifelse ahead != nobody [
      if [pcolor] of patch-ahead 1 = black [
        forward 1 ]
      if [pcolor] of patch-ahead 1 = red [
        forward 1 ]
      if [pcolor] of patch-ahead 1 = yellow + 1 [
        move-left
      ]
      if [pcolor] of patch-ahead 1 = yellow - 1 [
        move-up
      ]
      if [pcolor] of patch-ahead 1 = yellow - 2 [
        move-down
      ]
      if [pcolor] of patch-ahead 1 = yellow [
        move-losange-g
      ]
      if [pcolor] of patch-ahead 1 = yellow + 3 [
        move-losange-d
      ]
      if [pcolor] of patch-ahead 1 = yellow + 2 [
        move-right
      ]
    ]
  ]

```

```

]
[
  ifelse patience != 0 [
    move-patiently] [move-forcefully]
]]
] [
  ifelse patience != 0 [
    move-patiently] [move-forcefully]
]
end

to move-losange-g
ifelse [pcolor] of patch-ahead 1 = yellow [
  left 120
  if [pcolor] of patch-ahead 1 = yellow [
    left 145
  ]
  if [pcolor] of patch-ahead 1 = yellow - 1 [
    set heading 0
  ]
  if [pcolor] of patch-ahead 1 = yellow - 2 [
    set heading 180
  ]
  if [pcolor] of patch-ahead 1 = yellow + 1 [
    set heading 270
  ]
  if [pcolor] of patch-ahead 1 = yellow + 2[
    set heading 90
  ]
  if [pcolor] of patch-ahead 1 = yellow + 3[
    right 120
  ]
  ifelse patience != 0 [
    move-patiently] [move-forcefully]
]
  ifelse patience != 0 [
    move-patiently ] [move-forcefully]
]
end

to move-losange-d
ifelse [pcolor] of patch-ahead 1 = yellow + 3 [
  right 120
  if [pcolor] of patch-ahead 1 = yellow [
    left 145
  ]
  if [pcolor] of patch-ahead 1 = yellow - 1 [
    set heading 0
  ]
  if [pcolor] of patch-ahead 1 = yellow - 2 [
    set heading 180
  ]
  if [pcolor] of patch-ahead 1 = yellow + 1 [

```

```

        set heading 270
    ]
    if [pcolor] of patch-ahead 1 = yellow + 2[
        set heading 90
    ]
    if [pcolor] of patch-ahead 1 = yellow + 3[
        right 120
    ]
    ifelse patience != 0 [
        move-patiently][move-forcefully]
    ]
    [ifelse patience != 0 [
        move-patiently][move-forcefully]
    ]
end

to move-right
ifelse [pcolor] of patch-ahead 1 = yellow + 2 [
    set heading 90
    if [pcolor] of patch-ahead 1 = yellow [
        left 145
    ]
    if [pcolor] of patch-ahead 1 = yellow - 1 [
        set heading 0
    ]
    if [pcolor] of patch-ahead 1 = yellow - 2 [
        set heading 180
    ]
    if [pcolor] of patch-ahead 1 = yellow + 1 [
        set heading 270
    ]
    if [pcolor] of patch-ahead 1 = yellow + 2[
        set heading 90
    ]
    if [pcolor] of patch-ahead 1 = yellow + 3[
        right 120
    ]
    ifelse patience != 0 [
        move-patiently][move-forcefully]
    ]
    [ifelse patience != 0 [
        move-patiently][move-forcefully]
    ]
end

to move-up
ifelse [pcolor] of patch-ahead 1 = yellow - 1 [
    set heading 0
    if [pcolor] of patch-ahead 1 = yellow [
        left 145
    ]
    if [pcolor] of patch-ahead 1 = yellow - 1 [
        set heading 0
    ]
    if [pcolor] of patch-ahead 1 = yellow - 2 [
        set heading 180
    ]
    if [pcolor] of patch-ahead 1 = yellow + 1 [
        set heading 270
    ]
    if [pcolor] of patch-ahead 1 = yellow + 2[
        set heading 90
    ]
    if [pcolor] of patch-ahead 1 = yellow + 3[
        right 120
    ]
]

```

```

ifelse patience != 0 [
  move-patiently ][move-forcefully]
] [
  ifelse patience != 0 [
    move-patiently][move-forcefully]
]
end

to move-forcefully
let target-patch patches in-cone 1 30
if target-patch != nobody[
  if any? other turtles-on target-patch [
    ask one-of other turtles-on target-patch [
      let suu one-of patches in-radius 1 with
[distance myself > 0]
      if suu != nobody [
        let numm count turtles-on suu
        if [pcolor] of suu != yellow and numm < 3 [
          set life life - 15
          move-to suu
          if life = 0 [
            set color green
            set walk-counter walk-counter + 1
          ]
        ]
      ]]
    if patch-ahead 1 != nobody [
      let turtles-number count other turtles-on
patch-ahead 1
      if turtles-number = 0 [
        fd 1
      ]
    ]
  ]
end

; Movement for patient turtles : wait for space
in way
to move-patiently
let target-patch patch-ahead 1
let others-in-vision other turtles in-cone 1 30
if any? others-in-vision [
  ; Increase waiting time if blocked ahead
  set patience patience - 25
]
if patience < 1 [
  set color red
  move-forcefully
]
] ]
fd 1

end

; Try to circumvent if blocked
to try-to-move-around
let random-patch one-of patches in-radius 1
with [distance myself > 0]; Find a random
patch around the turtle
let numbers count other turtles-on random-
patch
if random-patch != nobody and [pcolor] of
random-patch != yellow and numbers < 5 [
  face random-patch ; Make the turtle face the
randomly selected patch
  fd 1 ; Move forward by 1 step
]
end

to spread-fire
; Fire spreads
if ticks mod 2 = 0 [
  ask patches with [pcolor = orange] [
    ask neighbors with [pcolor != orange and
random-float 1.0 < 0.2] [
      set pcolor orange
    ]
  ]
]
; Turtles in fire die
ask turtles [
  if [pcolor] of patch-here = orange and color !=
green and color != brown[
    set death-counter death-counter + 1
    set color brown
  ]
]
end

; Draw sides of obsacles to customize
behaviour of turtles facing them
to draw-rectangle [xmin xmax ymin ymax]
ask patches with [
  ((pxcor >= xmin and pxcor <= xmax and pycor
>= ymin and pycor <= ymax))
]
set pcolor yellow

```

```

        ]
end

to draw-rectangle-hori-b [xmin xmax ymin ymax]
  ask patches with [
    ((pxcor >= xmin and pxcor <= xmax and pycor
    >= ymin and pycor <= ymax))
  ][
    set pcolor yellow
  ]
  ask patches with [pxcor >= xmin and pxcor <=
  (xmax + xmin) / 2 and pycor = ymin][
    set pcolor yellow + 1
  ]
  ask patches with [pxcor <= xmax and pxcor >
  (xmax + xmin) / 2 and pycor = ymin][
    set pcolor yellow + 2
  ]
  ask patches with [pxcor >= xmin and pxcor <=
  xmax and pycor = ymax][
    set pcolor yellow - 1
  ]
  ask patches with [pxcor = xmin and pycor >
  ymin and pycor <= ymax][
    set pcolor yellow
  ]
  ask patches with [pxcor = xmax and pycor >
  ymin and pycor <= ymax][
    set pcolor yellow
  ]
end

to draw-rectangle-hori-h [xmin xmax ymin ymax]
  ask patches with [
    ((pxcor >= xmin and pxcor <= xmax and pycor
    >= ymin and pycor <= ymax))
  ][
    set pcolor yellow
  ]
  ask patches with [pxcor >= xmin and pxcor <=
  xmax and pycor = ymin][
    set pcolor yellow + 1
  ]
  ask patches with [pxcor >= xmin and pxcor <=
  xmax and pycor = ymax][
    set pcolor yellow + 1
  ]
end

to draw-rectangle-verti-b [xmin xmax ymin ymax]
  ask patches with [pxcor = xmin and pycor >
  ymin and pycor <= ymax][
    set pcolor yellow
  ]
  ask patches with [pxcor = xmax and pycor >
  ymin and pycor <= ymax][
    set pcolor yellow
  ]
end

to draw-rectangle-verti-h [xmin xmax ymin ymax]
  ask patches with [
    ((pxcor >= xmin and pxcor <= xmax and pycor
    >= ymin and pycor <= ymax))
  ][
    set pcolor yellow
  ]

```

```

]
ask patches with [pxcor >= xmin and pxcor <
xmax and pycor = ymin] [
  set pc当地色 yellow + 1
]

ask patches with [pxcor >= xmin and pxc当地or <
xmax and pycor = ymax] [
  set pc当地色 yellow + 1
]

ask patches with [pxcor = xmin and pycor >
ymin and pycor < ymax] [
  set pc当地色 yellow - 2
]

ask patches with [pxcor = xmax and pycor >=
ymin and pycor < ymax] [
  set pc当地色 yellow - 2
]
end

to draw-rectangle-exit
ask patches with [
  pxc当地 = -19 and pycor > 10 and pycor < 20 ] [
  set pc当地色 yellow - 1 ]
ask patches with [
  pxc当地 = -10 and pycor >= 20 and pycor <= 30 ] [
  set pc当地色 yellow - 1 ]
ask patches with [
  pxc当地 = -19 and pycor < 30 and pycor > 20 ] [
  set pc当地色 yellow - 2 ]
ask patches with [
  pxc当地 = -10 and pycor < 20 and pycor >= 10 ] [
  set pc当地色 yellow - 2 ]
ask patches with [
  pxc当地 >= -19 and pxc当地 < -10 and pycor = 30 ] [
  set pc当地色 yellow + 1 ]
end

to draw-rectangle-hori-b-d [xmin xmax ymin
ymax]
ask patches with [
  ((pxcor >= xmin and pxc当地 <= xmax and pycor
>= ymin and pycor <= ymax))
][
  set pc当地色 yellow
]
ask patches with [pxcor >= xmin and pxc当地 <=
xmax and pycor = ymin] [
  set pc当地色 yellow + 1
]
ask patches with [pxcor >= xmin and pxc当地 <=
xmax and pycor = ymax] [
  set pc当地色 yellow + 1
]
ask patches with [pxcor = xmin and pycor >
ymin and pycor < ymax] [
  set pc当地色 yellow
]
ask patches with [pxcor = xmax and pycor >
ymin and pycor < ymax] [
  set pc当地色 yellow
]
end

```

```

breed [ fishes fish ]
breed [ predators predator ]

turtles-own [
  flockmates    ; agentset of nearby turtles
  nearest-neighbor ; closest one of our
  flockmates
  predators-in-view
]
end

to setup
  clear-all
  create-fishes population
  [ set color yellow - 2 + random 7 ; random
  shades look nice
    set size 1.5 ; easier to see
    setxy random-xcor random-ycor
    set flockmates no-turtles ]

  create-predators 1 [
    set shape "fish"
    set color red
    set size 4
    set heading 315
    hide-turtle
  ]
end

  reset-ticks
end

to go
  ask fishes [ flock
  flee
  ]
  ;; the following line is used to make the turtles
  ;; animate more smoothly.
  repeat 5 [ ask fishes [ fd 0.2 ] display ]
  ;; for greater efficiency, at the expense of
  smooth
  ;; animation, substitute the following line
  instead:
  ;; ask turtles [ fd 1 ]

  ask predators [
    move
  ]
end

  tick
end

to move
  setxy mouse-xcor mouse-ycor
  ;; only show the predator if the mouse pointer
  is
  ;; actually inside the view
  set hidden? not mouse-inside?
end

to flock ; turtle procedure
  find-flockmates
  if any? flockmates
    [ find-nearest-neighbor
      ifelse distance nearest-neighbor <
      maximum-separation
        [ separate ]
        [ align
        cohere ] ]
end

to flee
  set predators-in-view predators in-cone 10
  120
  if any? predators-in-view [
    set heading 90 + towards one-of predators-
    in-view
  ]
end

to find-flockmates ; turtle procedure
  set flockmates other fishes in-radius vision
end

to find-nearest-neighbor ; turtle procedure
  set nearest-neighbor min-one-of flockmates
  [distance myself]
end

  ;;; SEPARATE

to separate ; turtle procedure
  turn-away ([heading] of nearest-neighbor)
  max-separate-turn
end

```

FIGURE 8 – Code modèle 2

```

;;; ALIGN
to align ; turtle procedure
  turn-towards average-flockmate-heading
  max-align-turn
end

to-report average-flockmate-heading ; turtle
procedure
  ; We can't just average the heading variables
  here.
  ; For example, the average of 1 and 359
  should be 0,
  ; not 180. So we have to use trigonometry.
  let x-component sum [dx] of flockmates
  let y-component sum [dy] of flockmates
  ifelse x-component = 0 and y-component = 0
    [ report heading ]
    [ report atan x-component y-component ]
end

;;; COHERE

to cohere ; turtle procedure
  turn-towards average-heading-towards-
  flockmates max-cohere-turn
end

to-report average-heading-towards-
flockmates ; turtle procedure
  ; "towards myself" gives us the heading from
  the other turtle
  ; to me, but we want the heading from me to
  the other turtle,
  ; so we add 180
  let x-component mean [sin (towards myself +
  180)] of flockmates
  let y-component mean [cos (towards myself +
  180)] of flockmates
  ifelse x-component = 0 and y-component = 0
    [ report heading ]
    [ report atan x-component y-component ]
end

;;; HELPER PROCEDURES

to turn-towards [new-heading max-turn] ;

```

; Copyright 1998 Uri Wilensky.  
; See Info tab for full copyright and license.

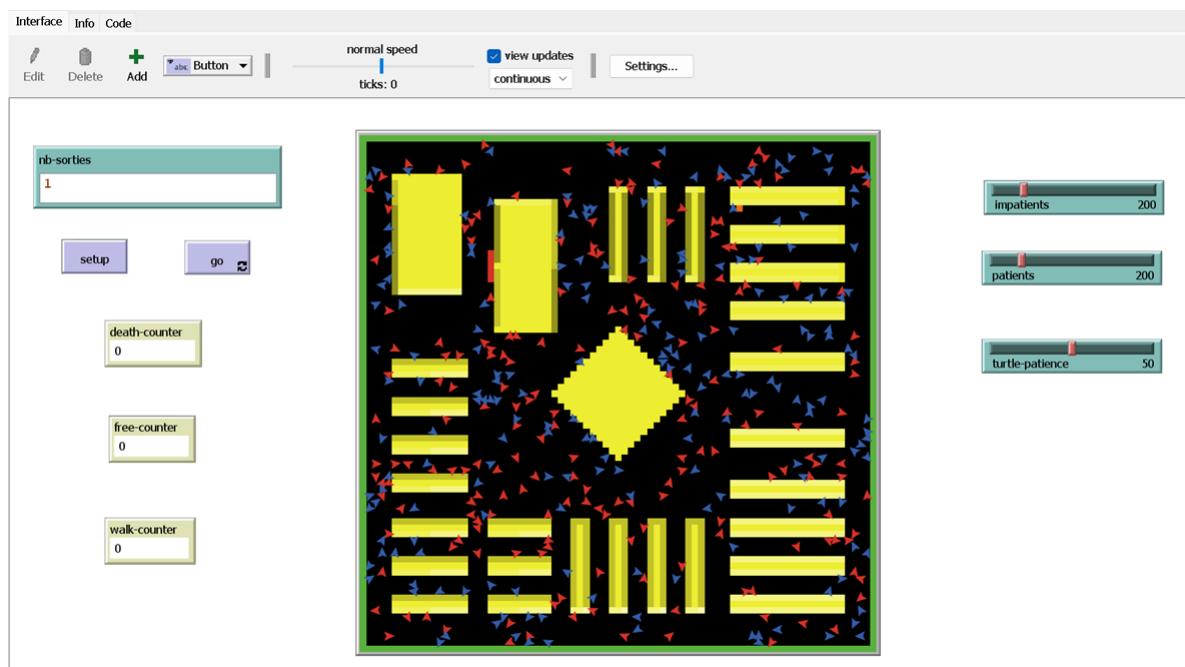


FIGURE 9 – Modèle 1 BU L1

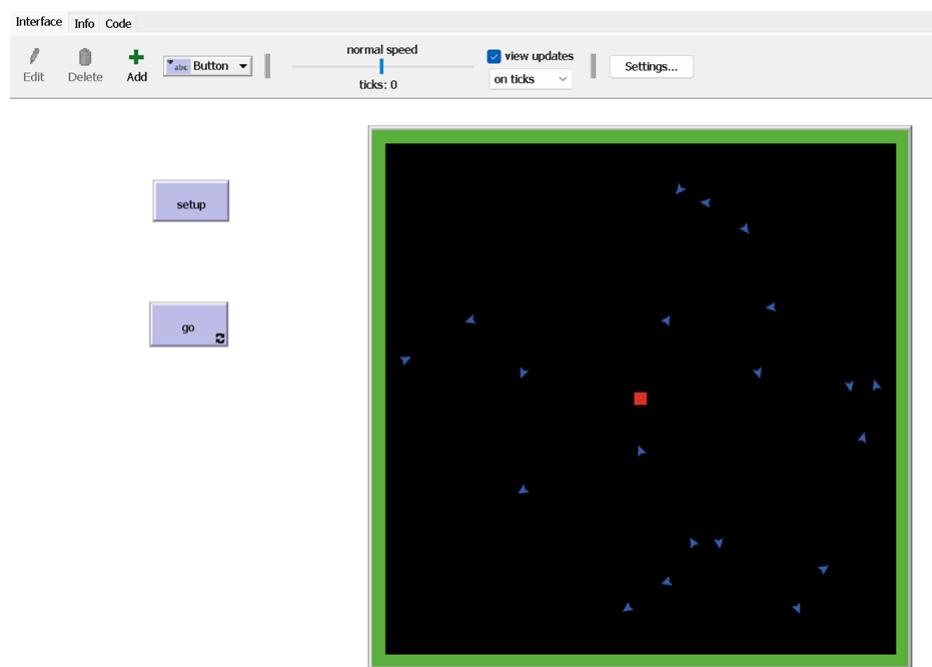


FIGURE 10 – Modèle 1 Simple sortie

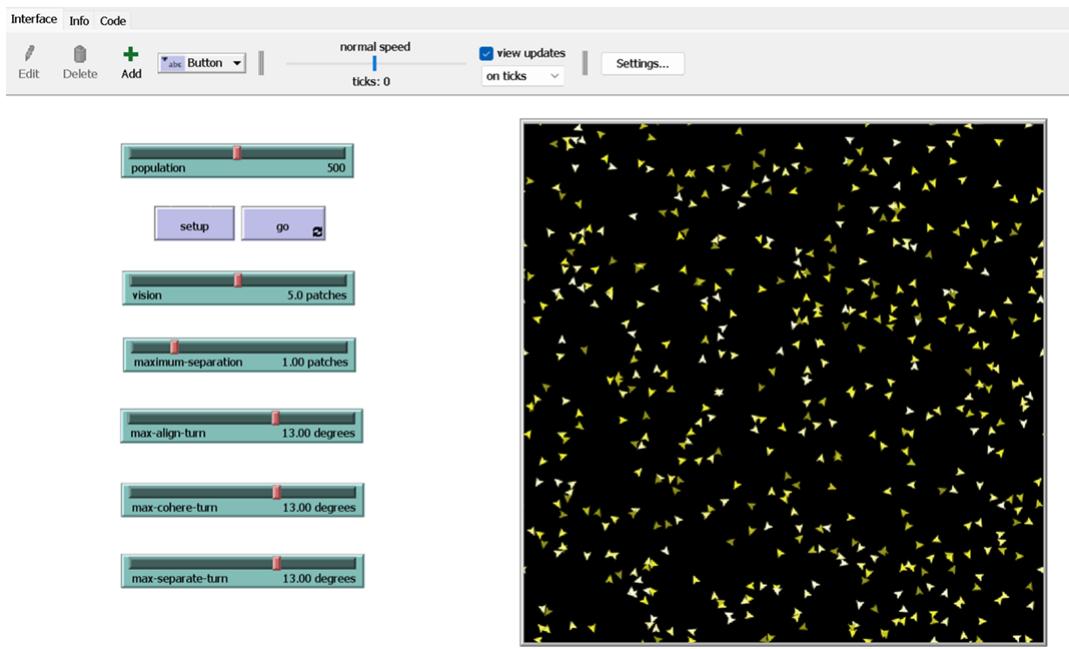


FIGURE 11 – Modèle 2 Banc de poissons

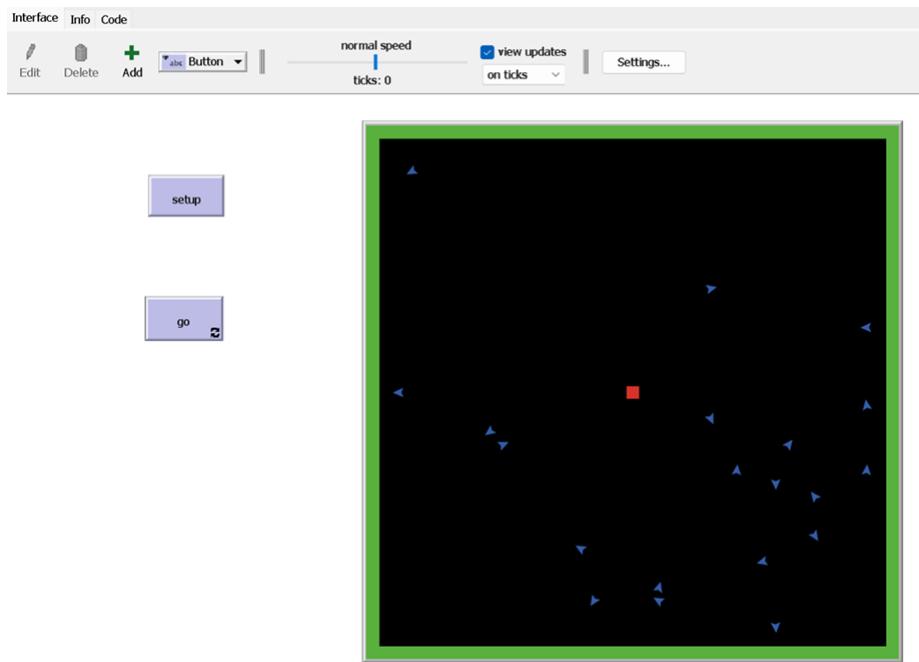


FIGURE 12 – Des agents et une sortie

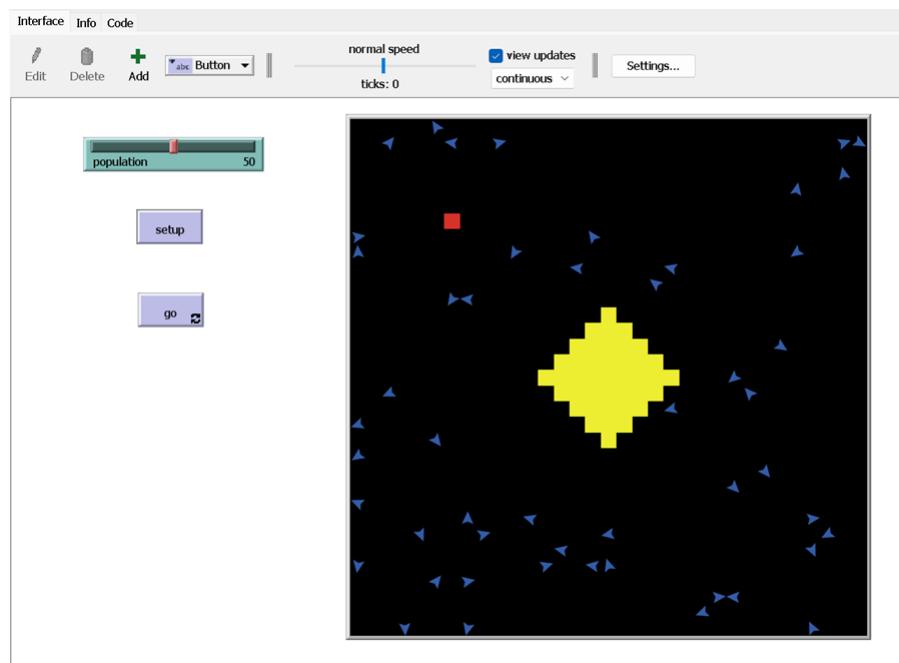


FIGURE 13 – Ajout d'un obstacle losange central

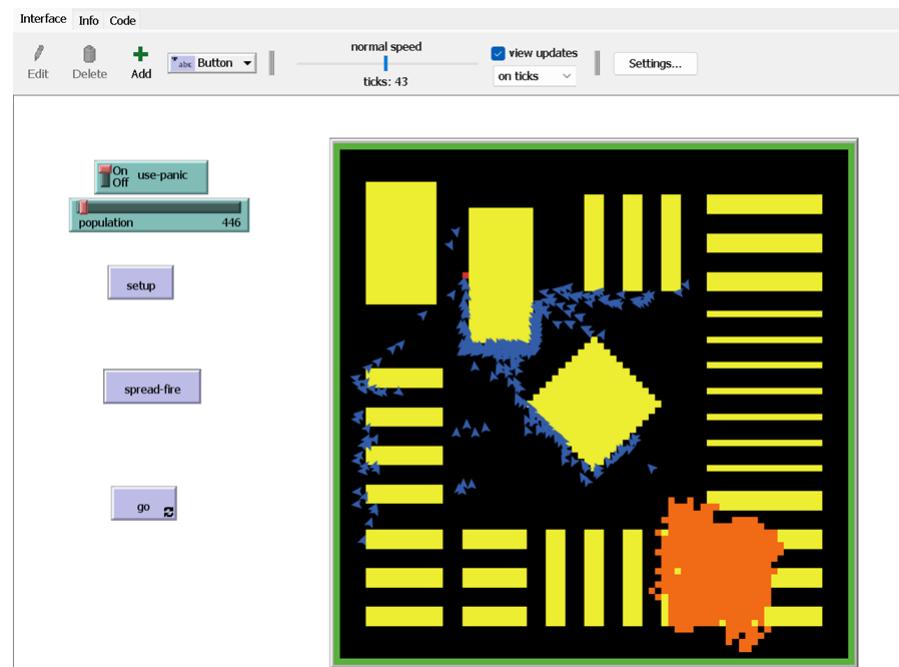


FIGURE 14 – Ajout du feu qui se propage

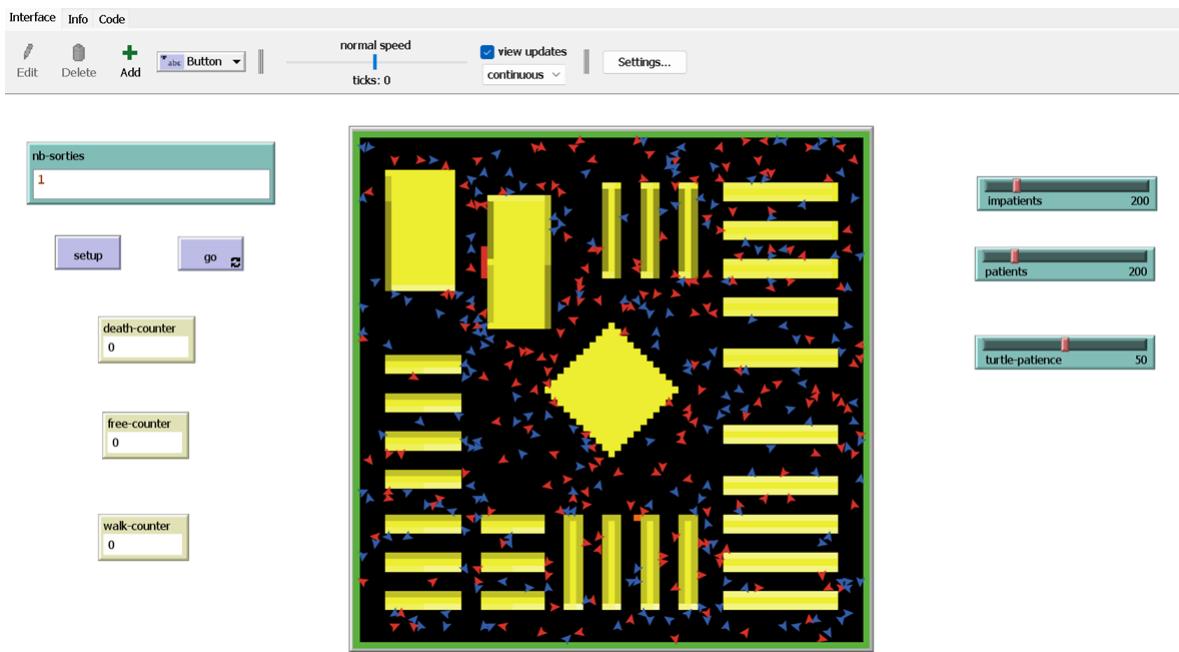


FIGURE 15 – Ajout du comportement patient/impatient

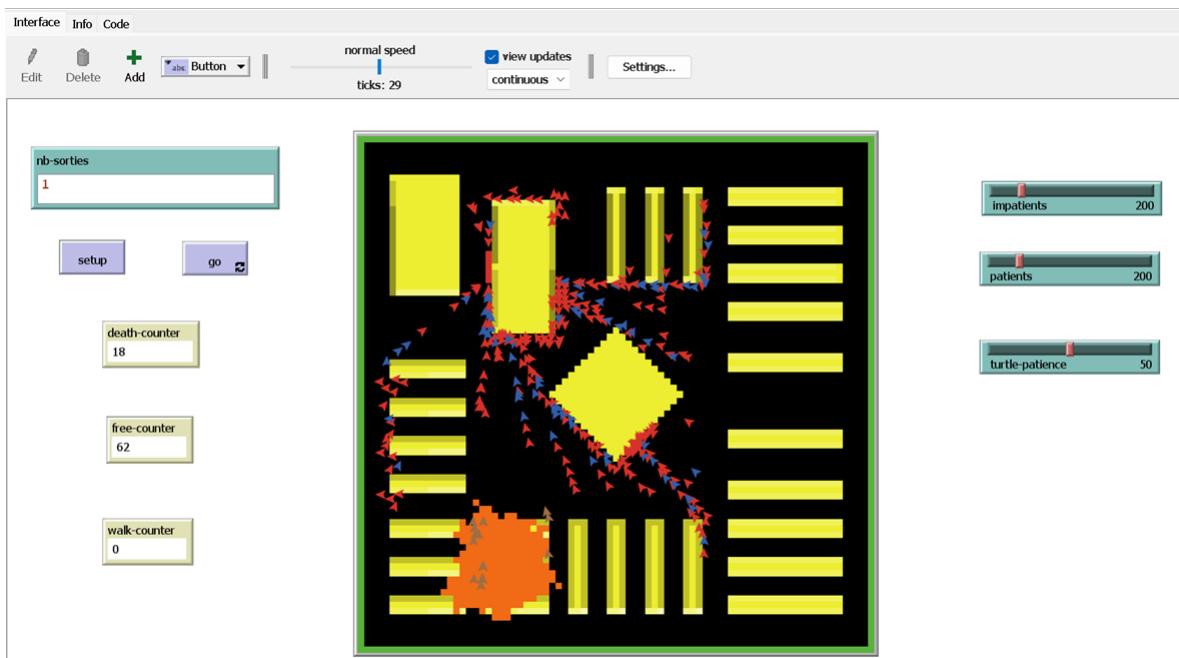


FIGURE 16 – Ajout de l'état de brûlé et d'asphyxié

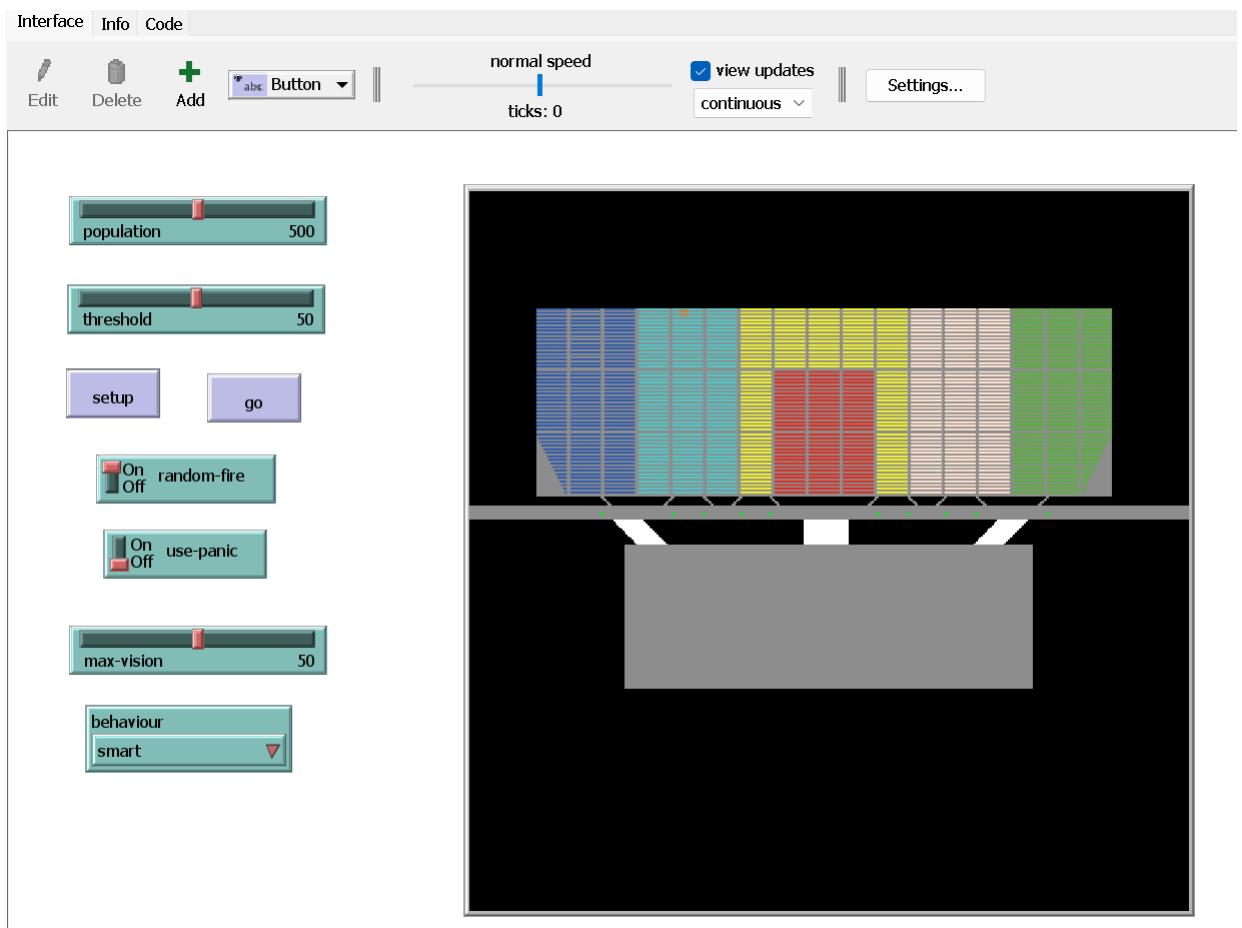


FIGURE 17 – Modèle d'évacuation d'incendie qui nous a inspiré