



Rapport Projet  
TIA  
Sujet 1  
**MONSTERS**

Groupe:  
**NESR Mohamad**  
**EL HIRACH Yassine**

1- Scenario/ Contexte:	2
2- Description des agents et leurs actions:	2
3- En quoi se distingue notre modèle?	4
4- Validation:	5
5- Tests:	5
6- Conclusion:	7

Enseignante: **KABACHI Nadia**

Année universitaire **2021/2022**

# **1- Scenario/ Contexte:**

Tout d'abord, nous avons choisi de commencer de 0 au lieu de reprendre un exemple disponible dans la librairie de modèles NetLogo. Nous nous sommes inspirés de la science-fiction afin de créer notre scénario qui est un combat entre les hommes et les monstres. Notre modèle consiste de 3 différents agents: les soldats, les étudiants et les monstres. Les monstres ont pour objectif de manger les humains, les soldats sont à la chasse des monstres alors que les étudiants sont paniqués et essayent de fuir les monstres. Tous les agents sont dotés d'une énergie différente de l'autre. Si cette énergie atteint la valeur de 0, l'agent meurt. L'objectif final est donc la survie des humains face aux monstres.

Bien sûr les monstres sont des espèces fictives donc impossible à étudier scientifiquement. Pour cela, nous avons imaginé notre propre idée de comment pourraient se comporter ces espèces.

# **2- Description des agents et leurs actions:**

## **Variables communes aux 3 agents:**

- energy : l'énergie est perdue lorsqu'on est attaqué. L'agent meurt quand elle atteint 0
- rank : 1 -> soldats, 2 -> étudiants, 3 -> monstres
- types : 1 -> humains, 2 -> monstres
- grouping-reach ; le rayon sur lequel les agents se regroupent

## **Les étudiants:**

- radius : distance à laquelle l'étudiant peut voir
- angle : angle du champ de vision de l'étudiant

Les étudiants n'ont aucune puissance offensive, leur but est de fuir les monstres quand ils sont dans leur champ de vision (changement de direction). Sinon, leur mouvement est aléatoire

## **Les soldats:**

- damage : la puissance de leur attaque vers les monstres

- reach : la distance qu'ils peuvent atteindre avec leur attaque
- targeting : la distance dans laquelle les soldats peuvent détecter les monstres
- oriented-monster : le monstre cible du soldat

Les soldats se déplacent sur la carte à la recherche des monstres, une fois trouvés, ils le poursuivent jusqu'à ce que l'un d'entre eux meurt.

### Les monstres:

- power: la puissance de leur attaque (inférieure à celle d'un soldat)
- monster-reach: la distance qu'ils peuvent atteindre avec leurs attaques
- hearing-range: la distance dont laquelle ils peuvent détecter les humains
- oriented-human: l'humain cible du monstre

Les monstres se déplacent sur la carte à la recherche des soldats, une fois trouvés, ils le poursuivent jusqu'à ce que l'un d'entre eux meurt.

### Setup:

On initialise nos agents avec les valeurs de leurs données. On fait donc appel à setup-soldiers, setup-students, setup-monsters.

```

to setup-students
  create-students number-students [
    setxy random-ycor random-xcor
    set shape "person student"
    set size 2
    set color blue
    set energy 600
    set angle 90
    set radius 1
    set types 1
    set rank 2
  ]
end

to setup-soldiers
  create-soldiers number-soldiers [
    setxy random-ycor random-xcor
    set shape "person soldier"
    set size 2
    set color green
    set energy 1000
    set targeting 5
    set damage 45
    set reach 2
    set types 1
    set rank 1
  ]
end

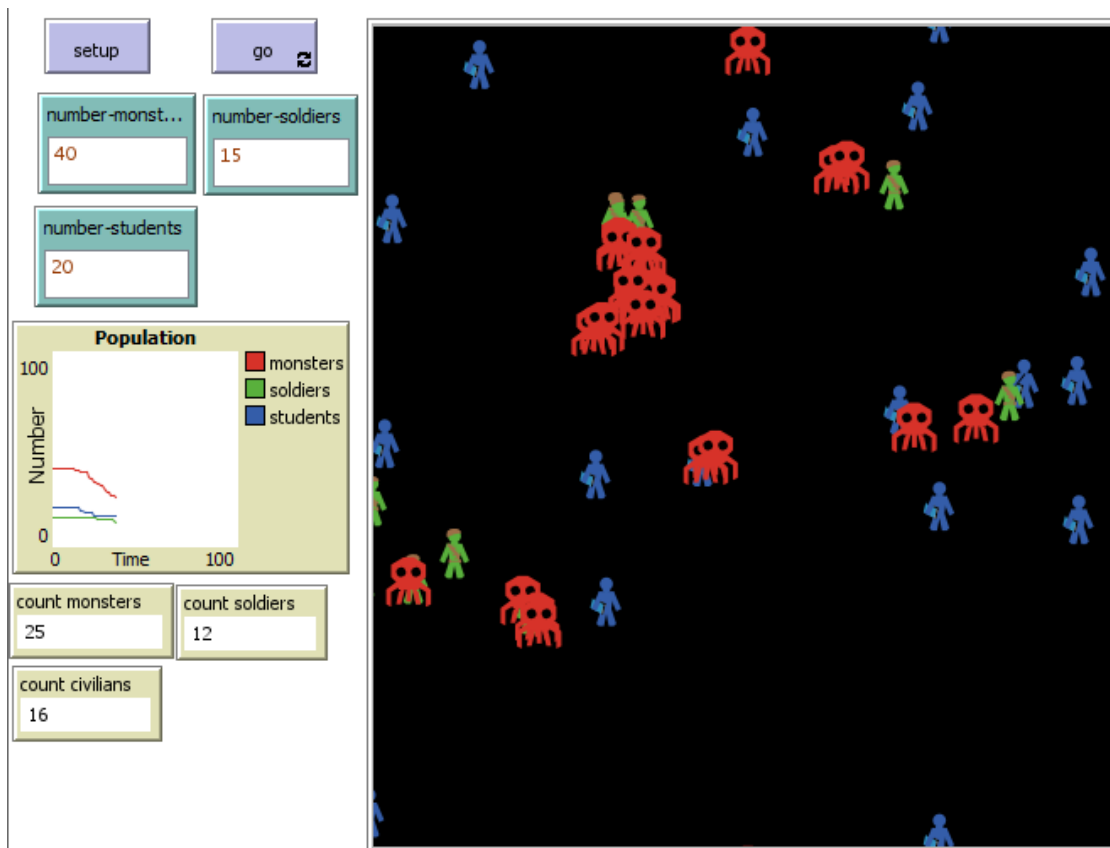
to setup-monsters
  create-monsters number-monsters [
    setxy random-ycor random-xcor
    set shape "monster"
    set size 2
    set color red
    set energy 800
    set power 20
    set monster-reach 1
    set hearing-range 2
    set types 2
    set rank 3
    set grouping-reach 5
  ]
end

```

### Go:

C'est là que se déroule tout le mouvement et les actions de notre simulation. Student-move, monster-move et soldier-move permettent le déplacement de nos agents de manière spécifique à lui. Un étudiant se déplace aléatoirement mais suivra un soldat si il est très proche afin d'être sûr d'être protégé. D'autre part, le soldat ne cherche pas à se regrouper, il se déplace aléatoirement jusqu'à ce qu'il rentre en contact avec un monstre. A ce moment, le soldat s'oriente vers le monstre et passe à l'offensive afin de

le battre. Finalement, les monstres ne se déplacent pas indépendamment des autres. Dans un certain rayon “grouping-reach”, les monstres se regroupent autour du monstre ayant le plus d'énergie et vont ensuite se déplacer ensemble afin d'attaquer les humains. Malgré le fait qu'ils soient moins puissant que les soldats, le fait qu'ils se regroupent les rendra plus féroce et dangereux. Notre condition d'arrêt est soit la mort de tous les monstres soit la mort de tous les humains



Comme on peut l'observer, on distingue 3 comportements différents de nos 3 agents. Les monstres se regroupent ensemble, les soldats attaquent les monstres individuellement. Alors que les étudiants fuient ou se rapprochent d'un soldat (milieu à droite)

### **3- En quoi se distingue notre modèle?**

Ce modèle est différent des autres car on a 3 agents dont chacun se comporte d'une manière complètement différente des autres. Le regroupement des monstres en “cluster” est la partie la plus pertinente due au fait qu'un monstre tout seul va subir une défaite inévitable face à un soldat. Pour cela, les monstres se regroupent afin de pouvoir résister face aux humains. On

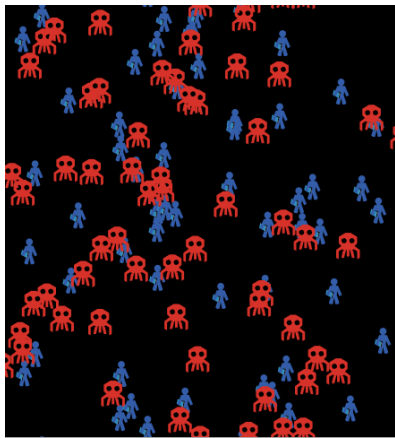
peut tester cela en désactivant le regroupement des monstres avec le switch "monsters-clustering".

## **4- Validation:**

Nous ne pouvons pas déterminer si notre modèle est juste ou faux vu que l'idée est imaginaire. Mais notre modèle décrit bien le problème posé et notre code se traduit bien dans nos simulations.

En réfléchissant logiquement, notre modèle retourne souvent les résultats auxquels on s'attend si on met des conditions extrêmes (pas de soldats par exemple).

On peut voir ici que les monstres gagnent comme prévu:



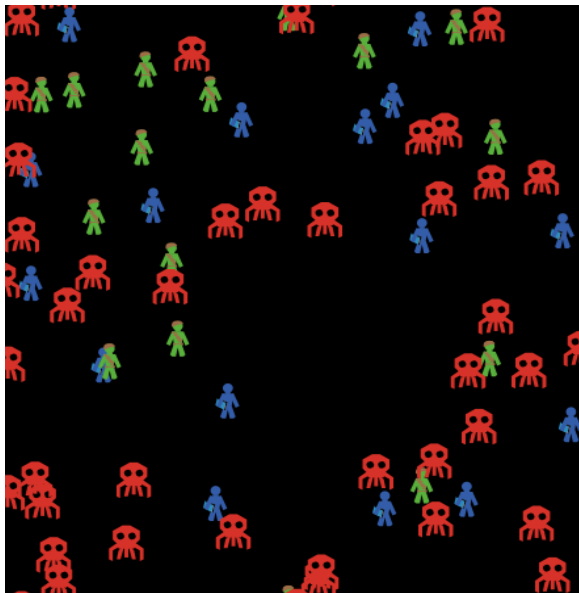
## **5- Tests:**

Vu que les soldats sont plus puissants que les monstres, une victoire est garantie si le nombre de soldats est équivalent à celui des monstres. Le soldat a le double de la force d'un monstre et 20% plus d'énergie au départ. Dans le cas où les étudiants sont exclus, si on suppose qu'on a 60 monstres, un combat équitable sera un combat avec 28 soldats (après les simulations, nous avons obtenu 30% de victoires de monstres contre 70% de victoires de soldats). D'autre part, 27 soldats permettent une victoire beaucoup plus fréquente pour les monstres (90%) alors que pour 30 soldats on a une victoire plus fréquentes des humains (90%).

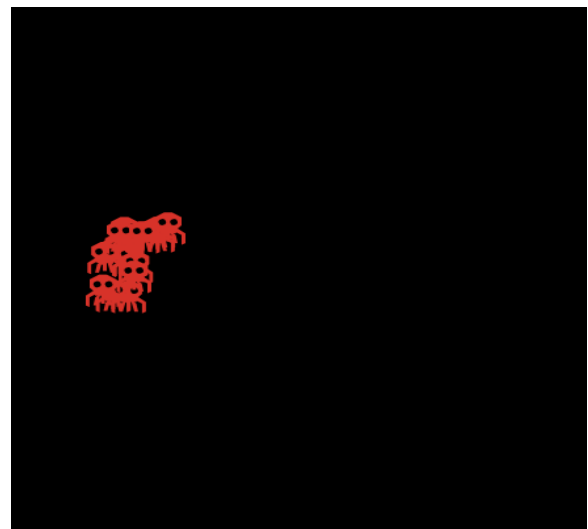
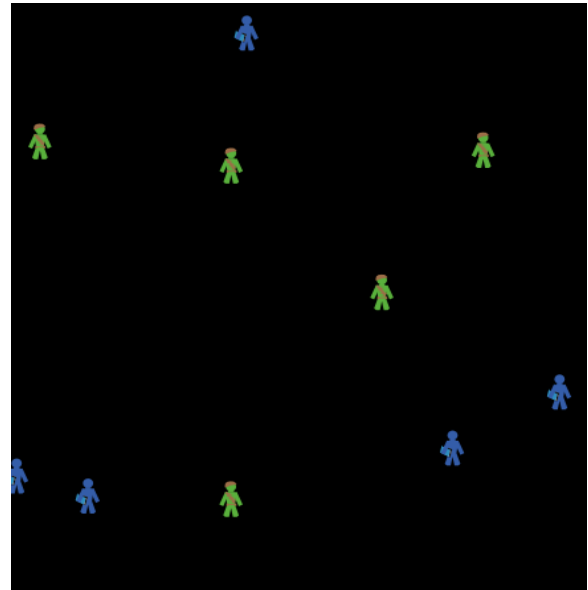
Nous ne pouvons bien sûr pas négliger l'influence des étudiants sur la victoire. Comme un monstre attaque un humain qu'il soit un soldat ou pas, il n'y aura pas de distinction lors de l'attaque et les étudiants vont donc distraire les monstres et les soldats pourront donc mieux les chasser.

Si on met 25 soldats, les monstres gagnent dans les 10 essais. Si on garde les mêmes paramètres mais on ajoute 25 étudiants sachant qu'ils n'ont aucun rôle en attaque, les monstres seront distraits et on verra que leur taux de victoire chute à 20%.

La meilleure combinaison pour rendre la simulation imprévisible sera de mettre 40 monstres, 15 soldats et 20 étudiants. Dans ce cas, on a 55% de victoires pour les humains.



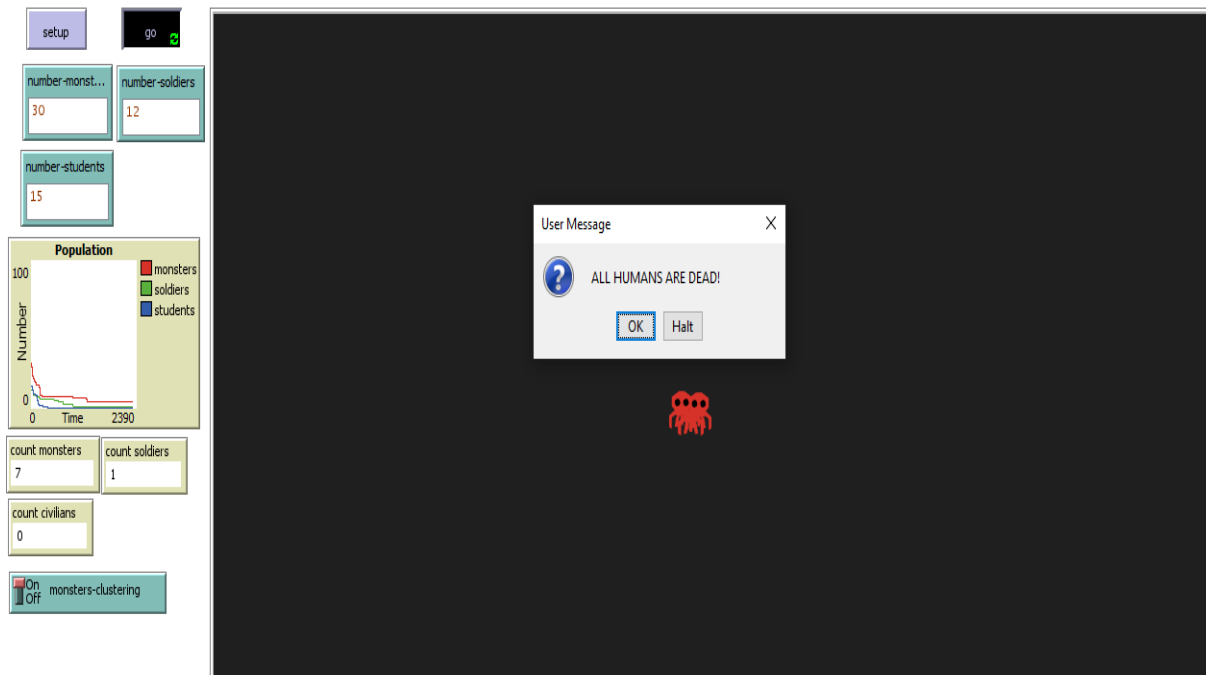
On voit bien qu'avec un démarrage identique nous avons obtenus des résultats différents



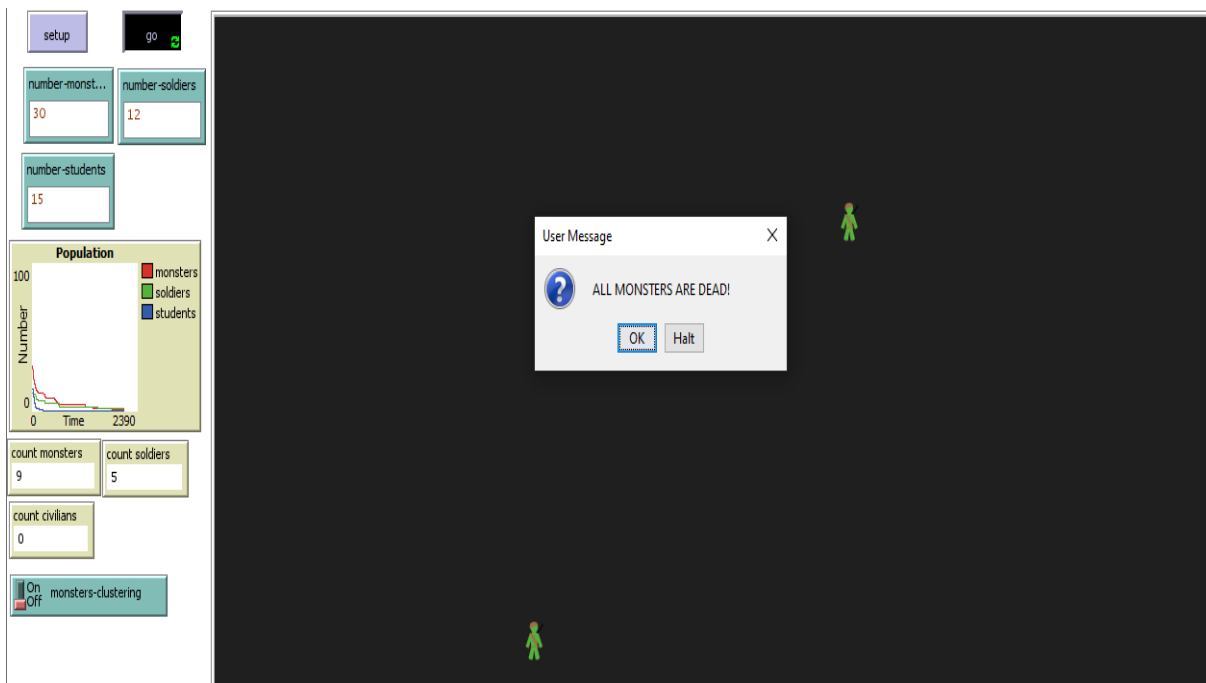
Le dernier test consiste à comparer le fonctionnement du modèle dans les 2 cas :avec le regroupement des monstres, et sans le regroupement.

On peut observer que dans cette exemple, quand le regroupement est activé, on a une victoire des monstres dans 55% des cas (11 sur 20 tests).De l'autre côté, si le regroupement est désactivé, ce taux chute jusqu'à 20% seulement (4 sur 20 tests).

### Avec clustering:



### Sans clustering:



## 6- Conclusion:

On peut en déduire que le nombre d'agents n'est pas le seul facteur qui affecte les simulations de modèles. Certains autres facteurs sont aussi importants que le nombre tel que la stratégie de combat (regroupement des monstres dans notre exemple).