

Rapport I62

Maxime HOAN, Van-Baptiste NGUYEN, Charlotte
MONTANARI

Mars 2023

Table des matières

1	Introduction	2
2	Exigences	2
2.1	IHM	2
2.2	Système	3
2.3	Déplacement	4
2.4	Nid	5
2.5	Ressources	5
3	Axe Fonctionnel	6
3.1	Diagramme statique	6
3.2	Diagramme des cas d'utilisations	7
3.3	Diagramme de séquence	7
3.3.1	Diagramme de séquence "QUIT"	8
3.4	Diagramme E1 : Interrupt the progress of the game	8
3.5	Diagramme A1 : Refuse the confirmation	9
3.5.1	Diagramme de séquence "SET UP"	10
3.5.2	Diagramme de séquence "CONTROL"	12
3.5.3	Diagramme de séquence "CREATE"	13
3.5.4	Diagramme de séquence "SAVE"	14
4	Axe Statique	14
4.1	Diagramme de classe	14
5	Maquettage de l'application des fourmis	18
6	Répartition des tâches	19
6.1	Les exigences	19
6.2	UML	19
6.3	Algo et tkinter	20

1 Introduction

2 Exigences

2.1 IHM

EX_IHM_0010 : Modifier la taille du monde

Le système permet à l'utilisateur de changer la taille du monde à l'aide d'un curseur.

EX_IHM_0020 : Modifier les paramètres d'espèce de fourmi

Le système permet à l'utilisateur de modifier les paramètres suivant de chaque espèce :

- sa race
- sa couleur
- sa vitesse
- sa force
- son radar
- son endurance
- sa population par nid

par la saisie d'une valeur pour chaque caractéristique Tant que la partie n'a pas commencé, on peut modifier les caractéristiques de la fourmi à partir de sa fiche de caractéristiques où l'utilisateur pourra appuyer sur un bouton "stylo" qui permettra de modifier.

EX_IHM_0030 : Modifier les paramètres d'espèces

Le système permet à l'utilisateur de modifier les paramètres d'une espèce suite à la sélection de l'espèce par l'utilisateur, puis de la modification de certaines valeurs.

EX_IHM_0031 : Sélectionner une espèce

Le système permet à l'utilisateur de sélectionner une espèce à l'aide d'un menu déroulant.

EX_IHM_0040 : Indiquer le temps/la période de la partie

Le système permet à l'utilisateur de voir le temps actuel de la partie (soleil ou nuit)

EX_IHM_0050 : Gérer la vitesse de la partie

Le système permet à l'utilisateur de gérer la vitesse de la partie (lent, normal, rapide)

EX_IHM_0060 : Apporter une aide

Le système permet à l'utilisateur de disposer d'une aide textuelle suite à l'activation d'un bouton aide qui sera écrit dans la langue choisie par l'utilisateur au début sur l'application.

EX_IHM_0070 : Choisir le déroulement de la partie

le système doit permettre à l'utilisateur de pouvoir choisir le déroulement de la partie :

- le déroulement pas à pas (voir les fourmis se déplacer un tour)
- le déroulement peut se faire avec un pas choisit par l'utilisateur (nombre de tour qu'il souhaite voir défiler)
- le déroulement en continu laisse la "partie " se faire sans arrêt.

EX_IHM_0080 : Pouvoir mettre le jeu en pause

Le système doit donner la possibilité à l'utilisateur de pouvoir mettre le jeu en pause.

EX_IHM_0090 : Afficher les traces de phéromones

Le système doit afficher des traces de phéromones sur l'écran en laissant le choix de la couleur des phéromones pour chaque espèce de fourmi et la trace s'épaissit en fonction du nombre de fourmis qui passe par ce chemin.

EX_IHM_0100 : Afficher le nombre de ressources Le système doit pouvoir afficher à l'utilisateur le nombre de ressources recueillies pour une espèce notamment pour chaque fourmi seule, que celle-ci grossit en fonction du nombre de ressources qu'elle a récupéré et que le nid s'agrandit en fonction des ressources qu'il dispose.

EX_IHM_0110 : Afficher les fourmis Le système doit prendre en compte les différentes caractéristiques de la fourmi donc notamment :

- le système doit ralentir une fourmi qui devient plus grosse
- le système doit permettre à la fourmi d'être plus grosse en fonction de sa grosseur
- le système doit forcer les fourmis à protéger leur reine

EX_IHM_0120 : Gérer les combats entre différentes espèces Le système doit permettre aux fourmis d'espèces différentes de se combattre.

2.2 Système

EX_SYSTEME_0010 : Charger un monde prédéfini

Le système permet à l'utilisateur de sélectionner un fichier avec pour extension ".mde" qui les coordonnées pour représenter un mur ou une partie d'un mur.

EX_SYSTEME_0020 : Créer un monde

Le système permet à l'utilisateur d'appuyer sur un bouton "créer un monde" qui lui permettra de dessiner des murs en restant appuyé sur le clic gauche et en déplaçant la souris à travers une carte vide.

EX_SYSTEME_0030 : Sauvegarder un monde en création

Le système permet à l'utilisateur de sauvegarder le monde en cours de création en appuyant sur un bouton "sauvegarder". Le système demandera un nom de fichier à l'utilisateur, puis enregistrera un fichier texte avec ce nom suivi de l'extension ".mde".

Ce fichier contiendra des coordonnées correspondant à l'emplacement d'un mur.

EX_SYSTEME_0040 : Création de maladie pour les fourmis le système doit laisser la possibilité de créer une maladie aux fourmis notamment par le biais de :

- le système doit pouvoir gérer les maladies en fonction des ressources qui sont récupérées par les fourmis
- le système doit pouvoir gérer les transmissions de maladie entre fourmi alliés et ennemies
- le système doit pouvoir gérer les maladies qui peuvent être présentes sur le chemin

EX_SYSTEME_0050 : Gestion des conditions météo le système doit gérer les différentes possibilités de météo comme par exemple :

- le système doit gérer l'abondance d'eau dû à une pluie et donc destruction de nid
- le système doit gérer un tremblement de terre et donc mort de fourmi et/ou nid
- le système doit gérer le problème de vent qui peuvent retourner des fourmis et casser un nid

2.3 Déplacement

EX_DEPLACEMENT_0010 : Déplacement des fourmis

Le système permet à la fourmi de connaître la position du nid si la partie est en mode soleil. Si la partie n'est pas en mode soleil, la fourmi cherche des traces de phéromones pouvant la ramener à un nid. La vitesse de la fourmi est réduite proportionnellement aux ressources qu'elle possède.

EX_DEPLACEMENT_0020 : Gérer l'endurance de la fourmi Le système doit diminuer l'endurance de la fourmi par rapport à l'itération de la partie. Pour chaque déplacement, la fourmi perd un montant fixe d'endurance.

EX_DEPLACEMENT_0030 : Gérer l'évaporation des phéromones

Le système doit pouvoir gérer le taux d'évaporation des phéromones à chaque itération

EX_DEPLACEMENT_0040 : Gérer la portée de vision Le système doit pouvoir gérer la portée de vision de la fourmi (son radar) qui sera défini avant que la partie commence et sera fixe tout le long de la partie. La vision permettra à la fourmi de détecter les éléments se trouvant dans son champs de vision (son radar).

EX_DEPLACEMENT_0050 : Gérer le retour au nid d'une fourmi Le système doit pouvoir gérer le retour au nid d'une fourmi fatiguée ou ayant découvert une ressource Dès que le seuil de fatigue est atteint, la fourmi retourne automatiquement au nid.

EX_DEPLACEMENT_0060 : Déplacer la fourmi Le système doit pouvoir gérer le suivi d'une trace de phéromones qui permet à la fourmi de privilégier ce chemin

EX_DEPLACEMENT_0070 : Déposer les phéromones Le système doit pouvoir gérer la quantité de phéromones déposées lors de la découverte d'une ressource (celle-ci sera multipliée par la richesse de la ressource). Plus on est proche d'une ressource, plus il y aura de phéromones (sur un obstacle, il y aura des phéromones de stress)

EX_DEPLACEMENT_0080 : Gérer le mouvement aléatoire le système doit laisser la possibilité de mouvement aléatoire aux fourmis si elles n'ont pas senti assez de phéromones.

2.4 Nid

EX_NID_0010 : Poser un nid Le système permet à l'utilisateur de poser un nid sur une zone vide de la carte suite à un clic gauche de la souris, après sélection de l'option "Poser nid".

EX_NID_0020 : Choisir la population d'un nid Le système permet à l'utilisateur de choisir le nombre de population d'un nid à l'aide d'une zone de saisie de texte (en vérifiant la valeur saisie) à choisir avant de poser un nid.

2.5 Ressources

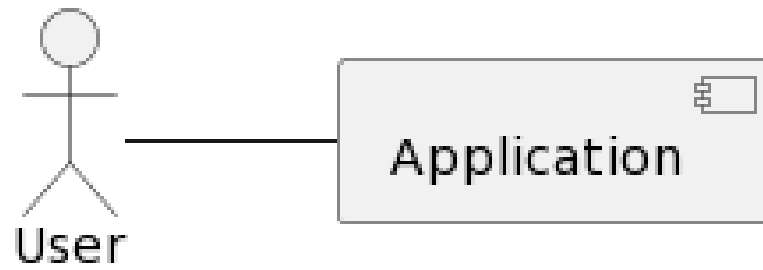
EX_RESSOURCE_0010 : Poser une ressource Le système permet à l'utilisateur de poser un nid sur une zone vide de la carte suite à un clic gauche de la souris, après sélection de l'option "Poser ressource".

EX_RESSOURCE_0020 : Choisir la richesse d'une ressource Le système permet à l'utilisateur de choisir la quantité d'une ressource à l'aide d'une zone de saisie de texte (en vérifiant la valeur saisie) à choisir avant de poser une ressource.

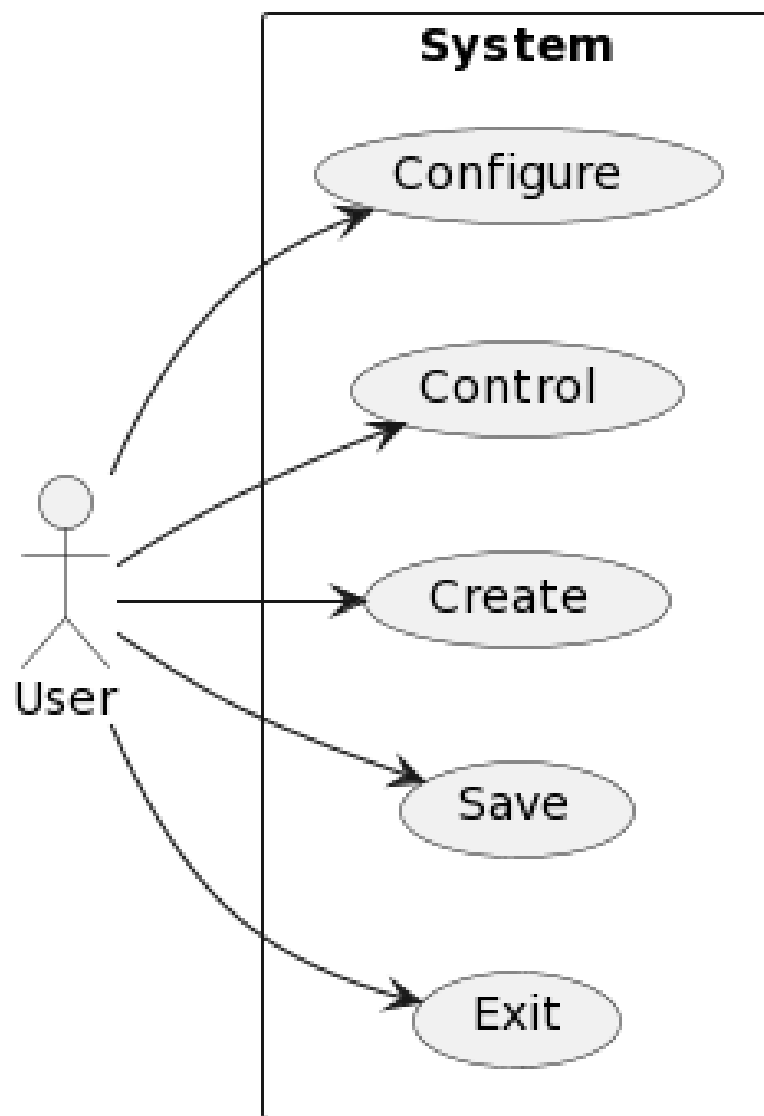
EX_RESSOURCE_0030 : Stocker les ressources par fourmi Le système permet à chaque fourmi de stocker sur elle un nombre max de ressources

3 Axe Fonctionnel

3.1 Diagramme statique



3.2 Diagramme des cas d'utilisations



3.3 Diagramme de séquence

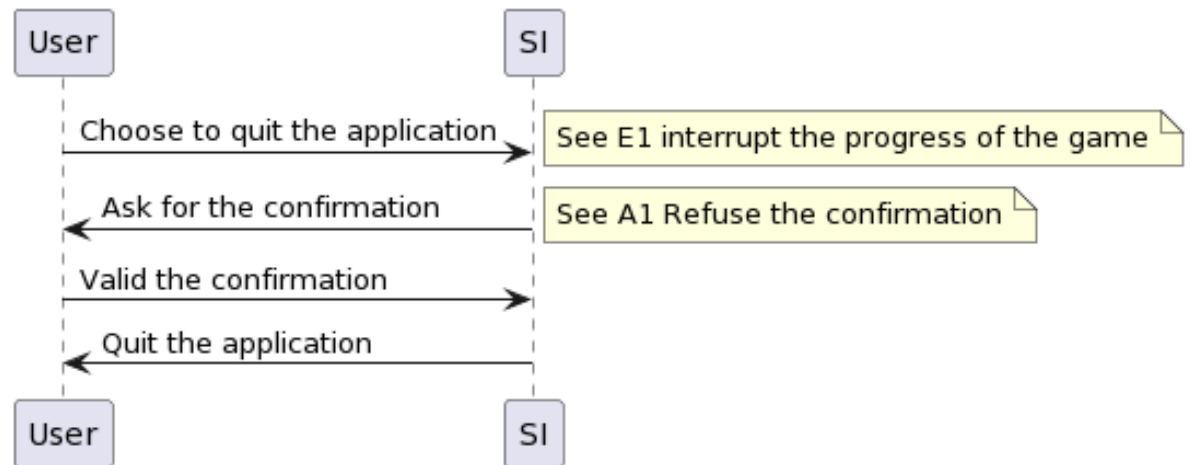
```
1
2 @startuml
3 User --> SI : Choose to quit the application
4 note right : See E1 interrupt the progress of the game
```

```

5 SI --> User : ask for the cnonfirmation
6 note right : See A1 Refuse the confirmation
7 User --> SI : Valid the confirmation
8 SI --> User : Quit the application
9 @enduml

```

3.3.1 Diagramme de séquence "QUIT"

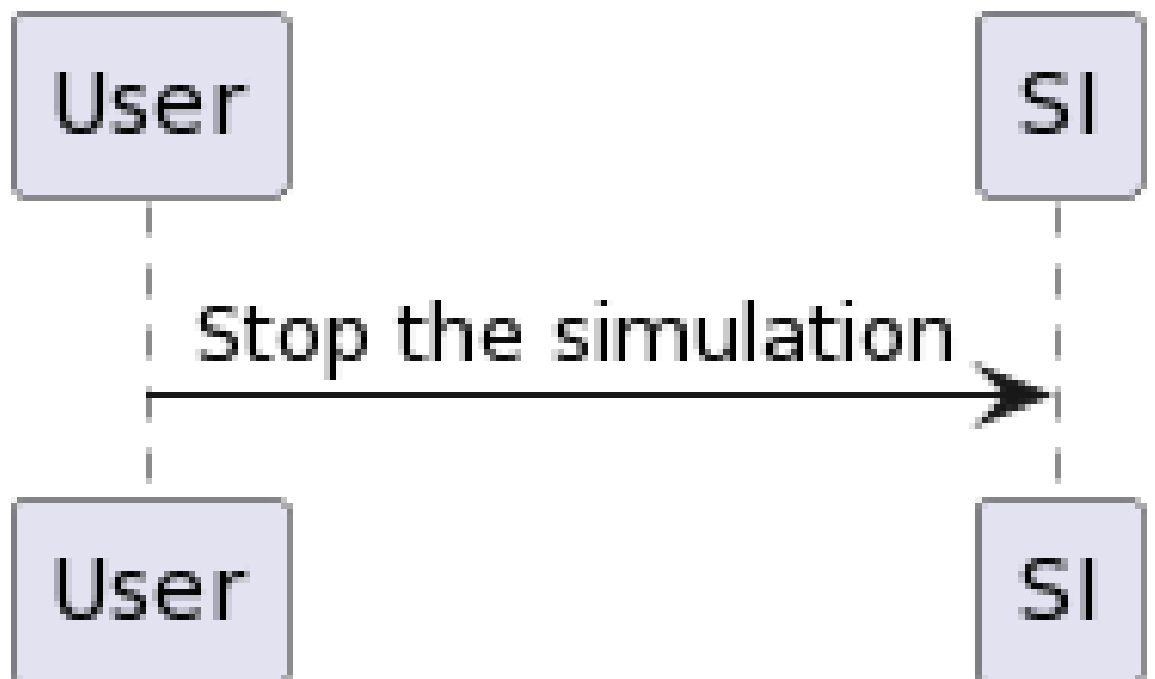


3.4 Diagramme E1 : Interrupt the progress of the game

```

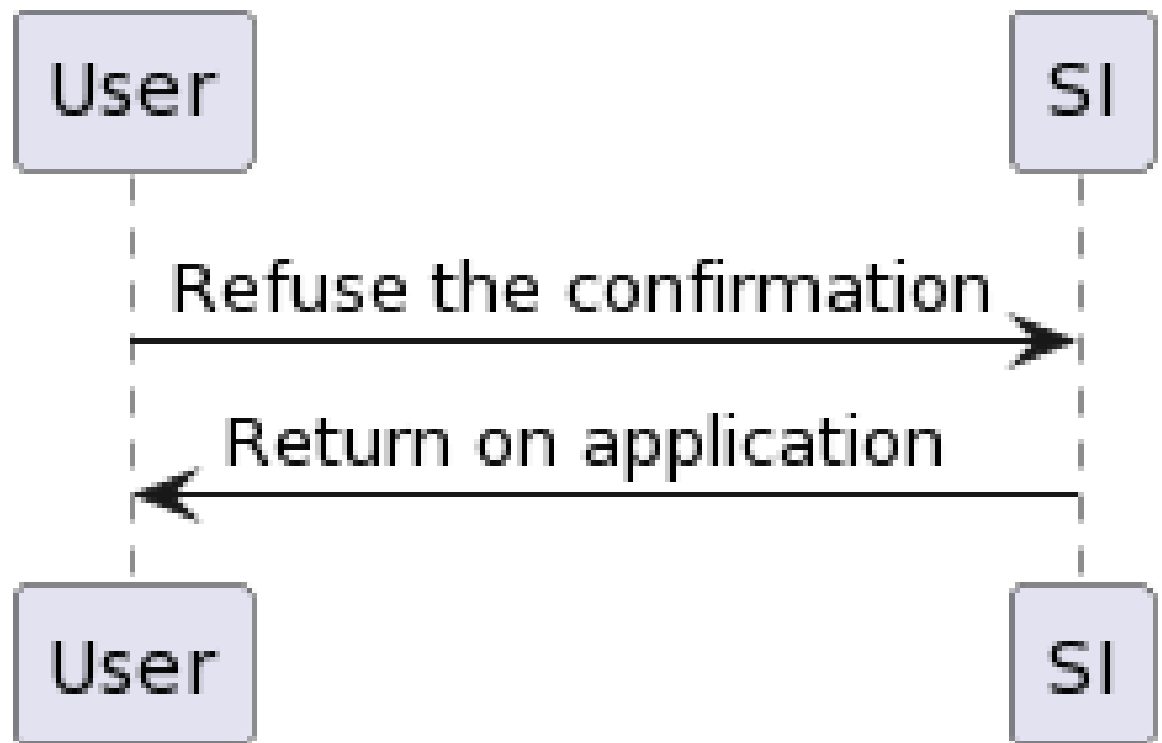
1
2 @startuml
3 User -> SI : Stop the simulation
4 @enduml

```

3.5 Diagramme A1 : Refuse the confirmation

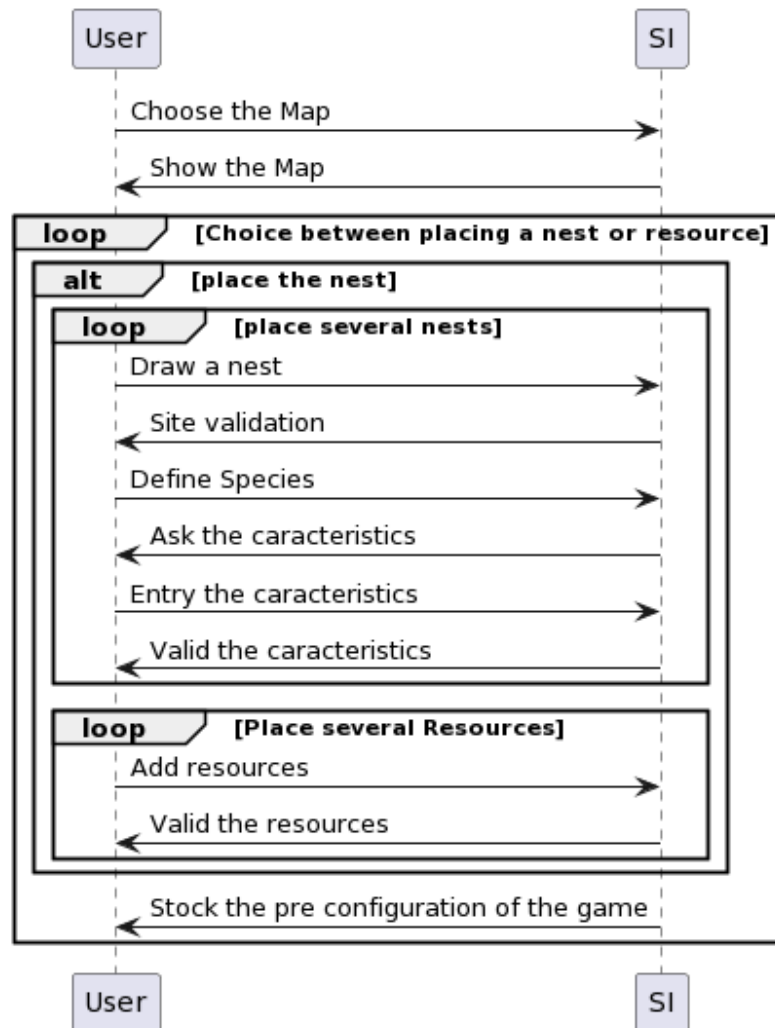
```
1
2 @startuml
3 User -> SI : Refuse the confirmation
4 SI -> User : Return on application
5 @enduml
```



3.5.1 Diagramme de séquence "SET UP"

```

1
2 @startuml
3 User -> SI : Choose the Map
4 SI -> User : Show the Map
5 loop Choice between placing a nest or resource
6 alt place the nest
7 loop place several nests
8 User -> SI : Draw a nest
9 SI -> User : Site validation
10 User -> SI : Define Species
11 SI -> User : Ask the characteristics
12 User -> SI : Entry the characteristics
13 SI -> User : Valid the characteristics
14 end
15 Loop Place several Resources
16 User -> SI : Add resources
17 SI -> User : Valid the resources
18 end
19 end
20 SI -> User : Stock the pre configuration of the game
21 end
  
```



```

1
2  @startuml
3
4  Alt Choice of the course of the game
5  Alt Pause
6  User -> SI : Pause the game
7  SI -> User : Paused the game
8  else Resume the game
9  User -> SI : Resume the game
10 SI -> User : The game is resumed
11 end

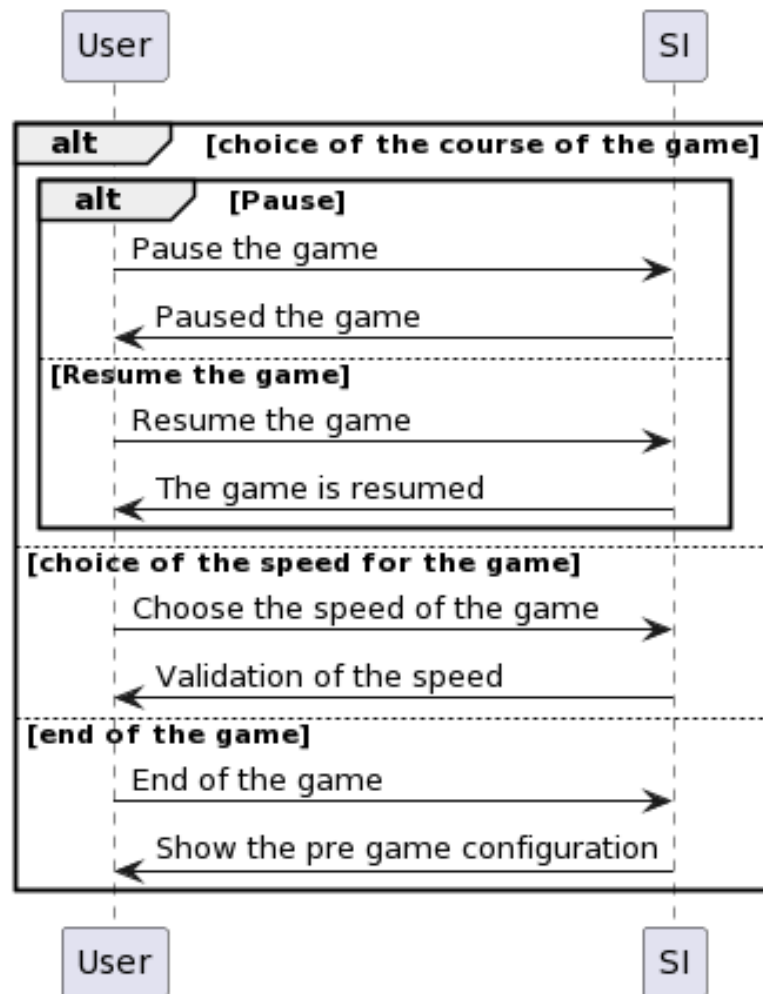
```

```

12 else Choice of speed for the game
13 User -> SI : Choose the speed of the game
14 SI -> User : Validation of the speed
15 else End of the game
16 User -> SI : End of the game
17 SI -> User : Show the pre-game configuration
18 end
19 @enduml

```

3.5.2 Diagramme de séquence "CONTROL"



```

1
2 @startuml

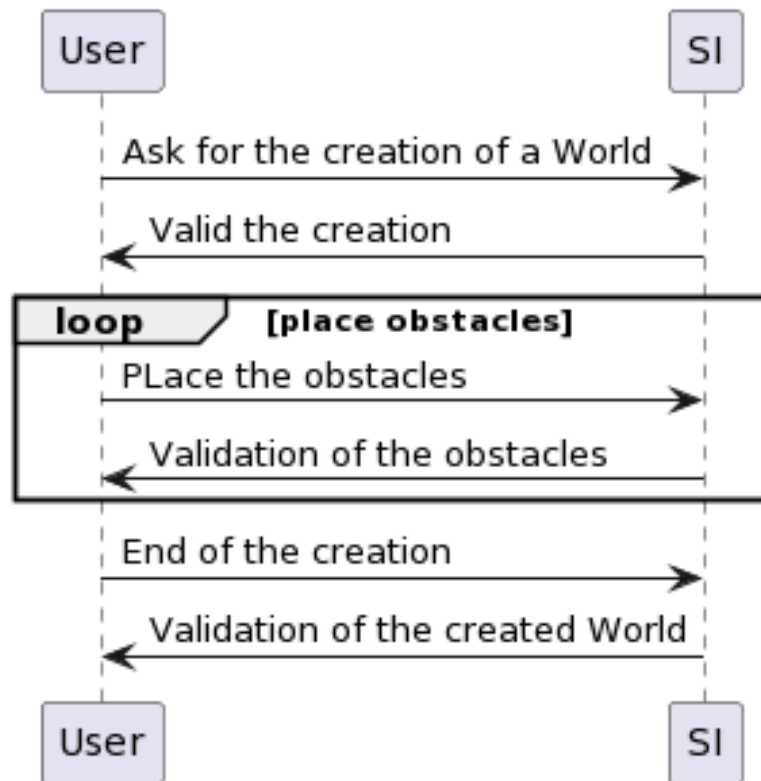
```

```

3
4 User -> SI : Ask for the creation of the Map
5 SI -> User : Valid the creation
6 loop Place obstacles
7 User -> SI : Add Obstacle
8 SI -> User : Validation of the obstacles
9 end
10 User -> SI : End of creation
11 SI -> User : Validation of the created of the Map
12 @enduml

```

3.5.3 Diagramme de séquence "CREATE"

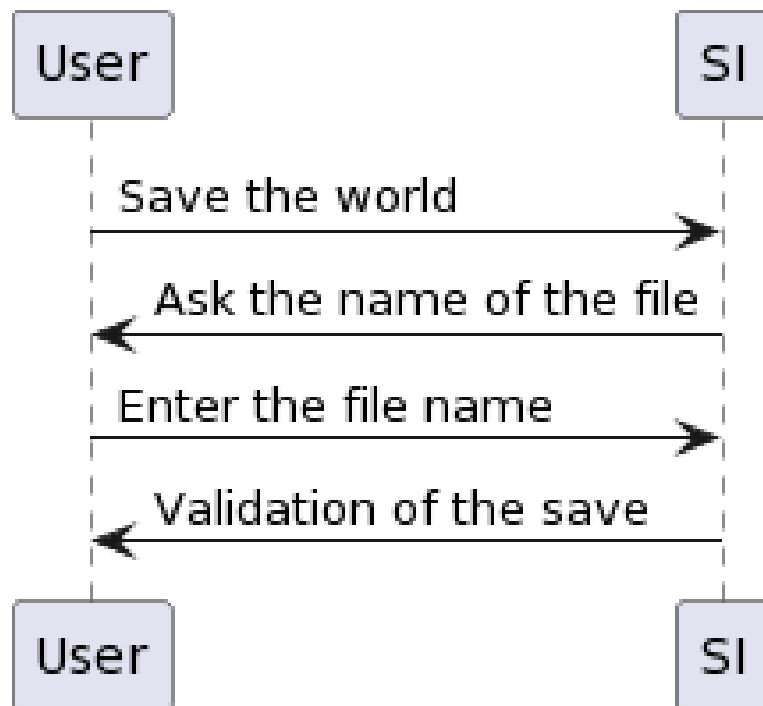


```

1
2 @startuml
3
4 User -> SI : Save the Map
5 SI -> User : Ask the name of the File
6 User -> SI : Enter the file name
7 SI -> User : Validation of the save
8 @enduml

```

3.5.4 Diagramme de séquence "SAVE"



4 Axe Statique

4.1 Diagramme de classe

```
1 @startuml
2 class Window {
3
4 my_window()
5 exit_game()
6 help_button()
7 language_define()
8 nest_define()
9 validate_worker()
10 validate_queen()
11 validate_scout()
12 pause()
13 sure_to_exit()
14 draw_nest()
15 draw_resources()
16 destroy_nest()
17 speed_value()
18 move()
19 }
```

```

19 start()
20 my_events()
21
22 }
23
24 class Map {
25     matrix : List
26     column : Integer
27     line : Integer
28     list_obstacle : List
29     list_meat : List
30     list_water : List
31     list_seed : List
32     list_nest : List
33     init_matrix()
34     print_matrix()
35     verification_location()
36     add_on_matrix()
37     write_matrix()
38     choose_the_map()
39 }
40
41 class Nest {
42     x : Integer
43     y : Integer
44     species : String
45     ant_worker ← Integer
46     list_worker ← List
47     ant_scout ← Integer
48     list_scout ← List
49     ant_queen ← Integer
50     list_queen ← List
51
52     create_worker()
53     create_scout()
54     create_queen()
55 }
56
57 class Ant{
58     x : Integer
59     y : Integer
60     species : String
61     id_nest : Integer
62
63     move()
64 }
65
66 class Resource{
67     x : Integer
68     y : Integer
69
70 }
71
72 class Obstacle {
73     x : Integer
74     y : Integer

```

```

75 }
76
77 class Meat{
78     value : Integer
79     quantity : Integer
80
81 }
82
83
84 class Seed{
85     value : Integer
86     quantity : Integer
87
88 }
89
90
91 class Water{
92     value : Integer
93     quantity : Integer
94
95 }
96
97
98 class Ant_Queen{
99     form : String
100     life : Integer
101     stack : Integer
102     size : Integer
103     stamina : Interger
104
105 }
106
107
108 class Ant_Scout{
109     form : String
110     life : Integer
111     stack : Integer
112     size : Integer
113     stamina : Interger
114
115 }
116
117 class Ant_Worker{
118     form : String
119     life : Integer
120     stack : Integer
121     size : Integer
122     stamina : Interger
123
124 }
125
126 class Pheromone{
127     x : Integer
128     y : Integer
129     quantity : Integer
130 }

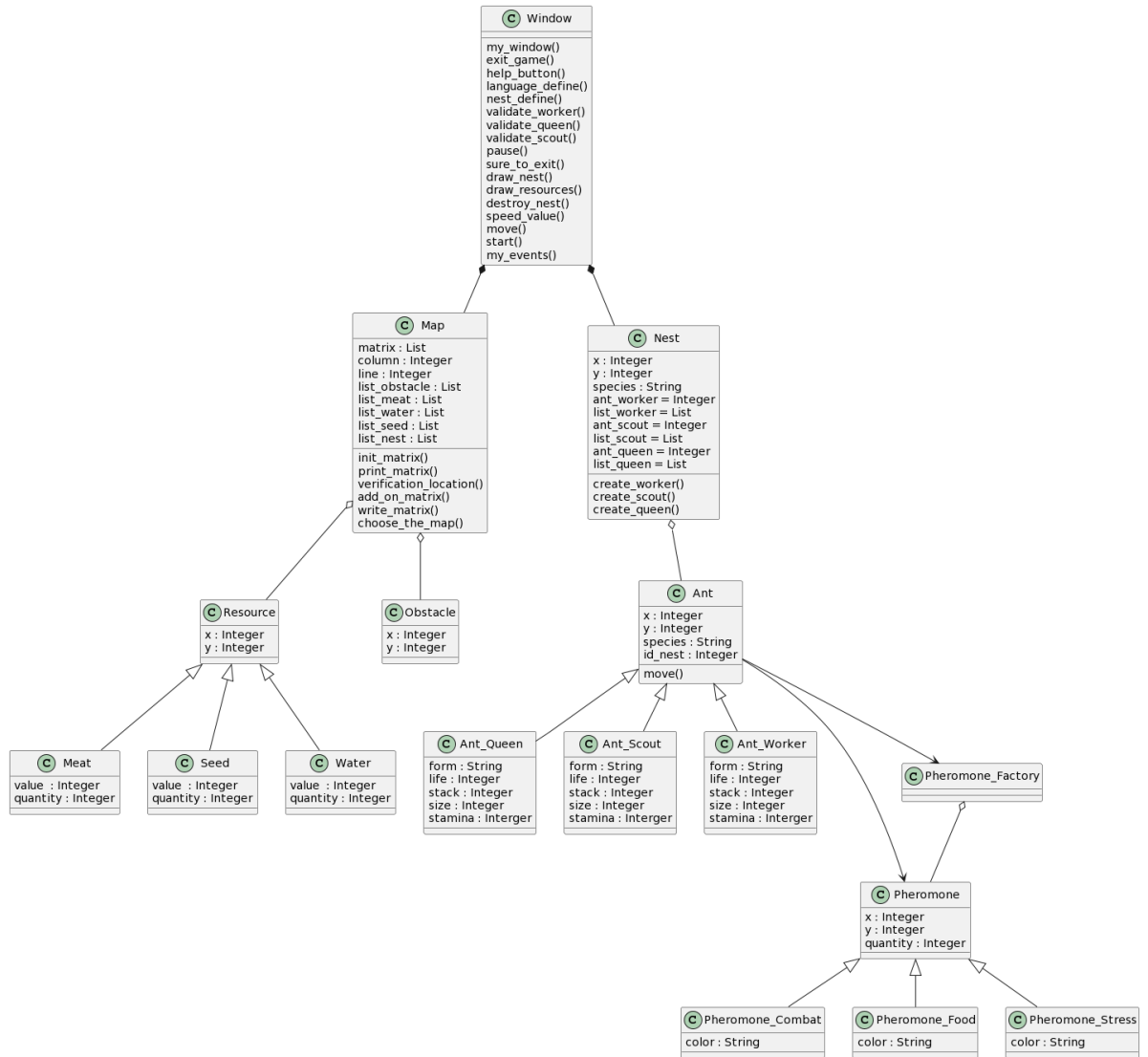
```



```

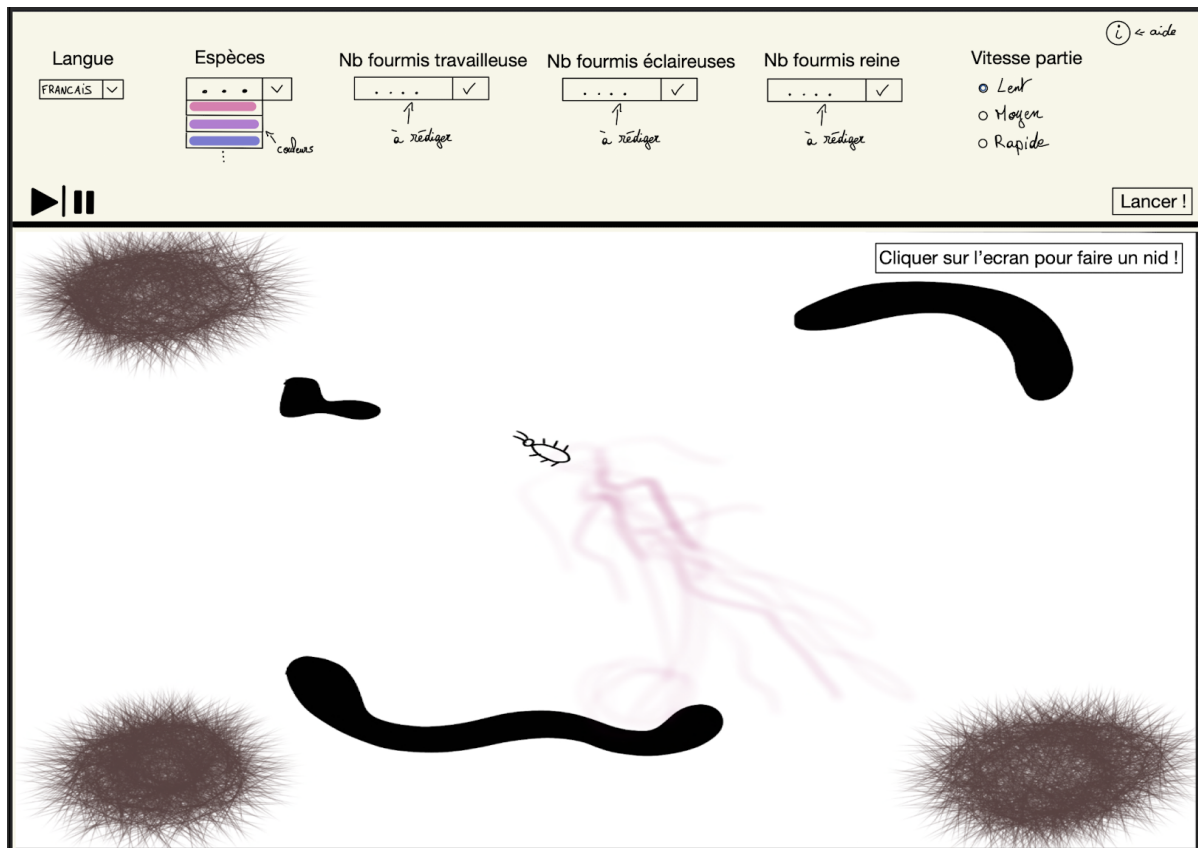
131
132 class Pheromone_Combat{
133     color : String
134 }
135
136 class Pheromone_Food{
137     color : String
138 }
139
140 class Pheromone_Stress{
141     color : String
142 }
143
144 class Pheromone_Factory{
145 }
146
147
148 Window *-- Map
149 Window *-- Nest
150 Map o-- Resource
151 Map o-- Obstacle
152 Nest o-- Ant
153 Resource <|-- Meat
154 Resource <|-- Water
155 Resource <|-- Seed
156 Ant --> Pheromone
157 Ant --> Pheromone_Factory
158 Pheromone_Factory o-- Pheromone
159 Ant <|-- Ant_Queen
160 Ant <|-- Ant_Scout
161 Ant <|-- Ant_Worker
162 Pheromone <|-- Pheromone_Food
163 Pheromone <|-- Pheromone_Combat
164 Pheromone <|-- Pheromone_Stress
165
166 @enduml

```



5 Maquettage de l'application des fourmis

Au cours de ce projet, nous avons du faire une maquette pour avoir un aperçu de notre application . Nous avons décidé de choisir une interface assez simple d'utilisation, avec les boutons mis en évidence afin que l'utilisateur ne soit pas décontenancé devant notre application, et que tout lui semble facile. Sur cette maquette on a essayé d'imaginer a quoi ressemblerai les obstacles ainsi que les nids dans le meilleur des cas, mais cela n'a pas été fait, dû à la complexité d'effectuer ces formes sur tkinter.



6 Répartition des tâches

6.1 Les exigences

Nous avons écrit chacun une dizaine d'exigence demandée par le cahier des charges, qu'on a ensuite mis en commun afin d'avoir une vision commune du projet. On a réparti ces exigences en 5 catégories différentes : IHM , Système , Déplacement , Nid et Ressources.

6.2 UML

Lors de la conception du projet, nous avons réfléchi et fait des diagrammes statiques et dynamiques afin de structurer nos idées. On a fait ces diagrammes sur le site de "plantUML" et on a, par la suite, mis ces diagrammes sur le rapport avec les scripts qui permet de les créer.

6.3 Algo et tkinter

Dans cette partie, nous avons implanté le code et les classes qui sont présentes dans le diagrammes de classes ainsi que ces méthodes afin de pouvoir créer l'application. Malheureusement nous n'avons pas pu implémenter toutes les fonctionnalités que l'on voulait par manque de temps et par les différents problèmes rencontrés (représentation des éléments sur le terrain, gestion de phéromones, la gestion des déplacements et interactions des fourmis).