

RG2A

Rapport de soutenance 1



RG2A

Redjeb Gabin

Gallardo Alexandre

Saadi Akram

Table des matières

1	Présentation du Groupe	3
1.1	Redjeb Gabin	3
1.2	Gallardo Alexandre	3
1.3	Saadi Akram	3
2	Présentation du Projet	3
2.1	Description	3
2.2	Inspirations	4
2.3	Intérêt algorithmique	4
3	Répartition des charges	7
4	Planning	8
5	Précision des tâches	9
5.1	Implémentation algo Boids	9
5.2	Implémentation de la recherche de la cible	9
5.3	Représentation graphique	9
5.4	Interface Graphique	9
5.5	Site Web	10
6	Outils de travail	10
7	Soutenance 1	11
7.1	Implémentation de l'algorithme de Boids	11
7.2	Implémentation Path finding	11
7.3	Représentation graphique	14
7.4	Interface graphique	15
7.5	Site Web	17
8	Conclusion	18

1 Présentation du Groupe

1.1 Redjeb Gabin

Je suis Gabin Redjeb chef de projet auto-proclamé et adoubé par tous. J'aime faire adhérer une équipe autour d'un projet et manager la motivation de chacun au cours du temps afin que chacun puisse donner le meilleur de lui même au service d'un projet commun RG2A.

1.2 Gallardo Alexandre

Je m'appelle Alexandre Gallardo et pour ma part j'entame mon premier s4, c'est donc pour moi une nouvelle étape à franchir avec de nouvelles difficultés avec toujours comme aboutissement l'amélioration de mes compétences notamment pour des projets en groupe comme celui ci.

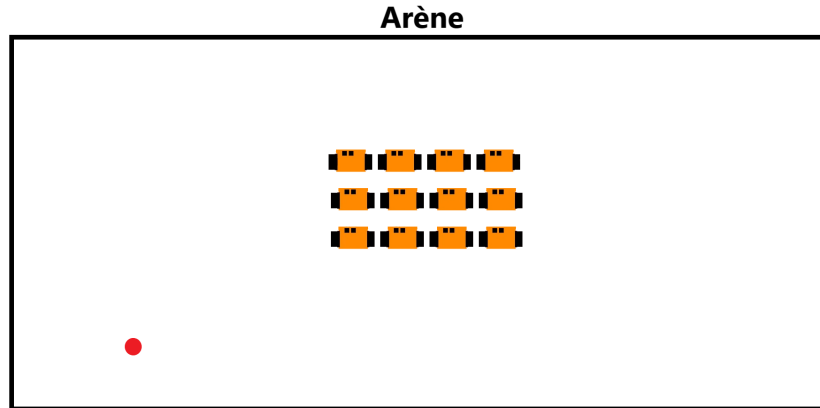
1.3 Saadi Akram

Quand j'ai rejoint epita, l'infographie était l'un de mes principaux intérêts en informatique, j'ai toujours voulu construire quelque chose à partir de rien, réinventer la roue, transformer le traitement informatique en images virtuelles pixel par pixel. J'étais aussi très intéressé par la modélisation et les simulations, c'est pourquoi j'ai choisi de faire ce projet car c'est un bon début pour mon parcours en infographie et modélisation. Je suis très investi dans ce projet et je suis sûr que je vais apprendre beaucoup.

2 Présentation du Projet

2.1 Description

Notre projet consiste à créer une simulation d'exploration où :
12 robots doivent être capables de trouver le plus rapidement possible, même dans les pires conditions, une cible positionnée dans une arène.
Les robots sont tous autonomes, il n'y a pas de hiérarchie et ne peuvent communiquer entre eux qu'à partir d'une certaine distance (pour pouvoir prévenir les autres robots de la position de la cible). La cible est placée dans l'arène de manière aléatoire. L'utilisateur pourra ajouter des obstacles dans l'arène permettant de complexifier la recherche (**arène modulable**). Pour mener à bien la recherche de la cible nous simulons de la vie artificielle, en utilisant le principe de robotique d'essaim (les comportements des essaims). Pour reproduire ces comportements, nous allons utiliser l'algorithme de Boids.



2.2 Inspirations

L'algorithme de Boids est utilisé dans divers secteurs :

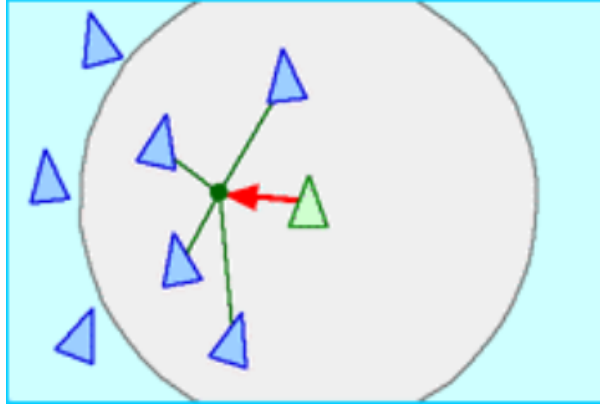
- l'infographie, fournissant des représentations réalistes de volées d'oiseaux et d'autres créatures, telles que des bancs de poissons ou des troupes d'animaux. :
- la cinématographie : permettant d'automatiser l'animation de créatures dans un décor comme des oiseaux, un banc de poissons ou une foule (Seigneurs des Anneaux)
- le contrôle et la stabilisation de véhicules sans pilote : permettant d'explorer des lieux inconnus qu'ils soient terrestres, aériens ou sous-marins.

2.3 Intérêt algorithmique

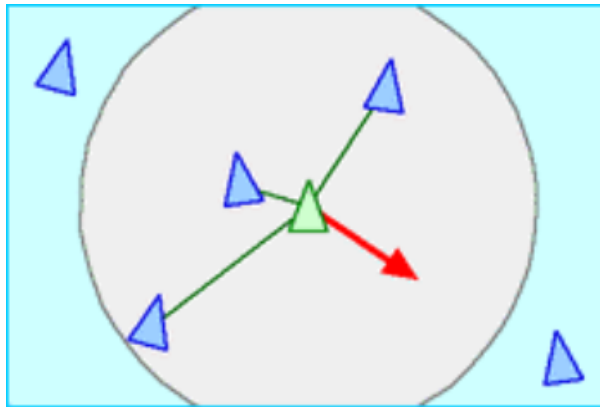
Boids est un programme de vie artificielle, qui simule le comportement de vol des oiseaux et qui permet de modéliser un comportement émergent, c'est-à-dire une complexité comportementale qui résulte de l'interaction d'agents individuels respectant un nombre limité de règles simples.

Règles :

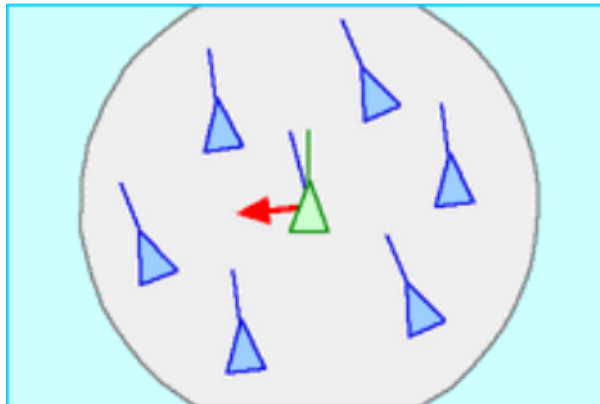
Cohésion : pour rester groupés, les robots essaient de suivre un même chemin.



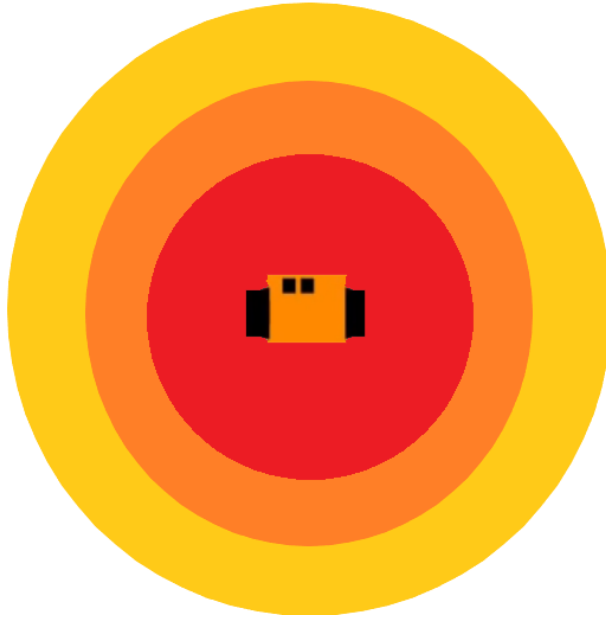
Séparation : deux robots ne peuvent pas se trouver au même endroit au même moment donc pour éviter de s'entasser chaque robot doit s'éviter.



Alignement : pour rester groupés les robots essaient de suivre le même chemin. 4



Dans les faits, cela se traduit par la réaction du robot à son voisinage. Ce voisinage se divise en trois zones, de la plus proche à la plus éloignée :



une **zone de répulsion**, une **zone d'orientation** et une **zone d'attraction**.

Lorsqu'un voisin entre dans sa zone de répulsion, le robot s'éloigne (séparation). Si le voisin est dans sa zone d'orientation, le boids* le suit (alignement). Et enfin, s'il est dans la zone d'attraction, le robot s'en rapproche (orientation).

boids : vient du mot bird-oid, humanoid pour un oiseau

3 Répartition des charges

<i>Tâches</i>	Gabin	Alexandre	Akram
Implémentation algo Boids :	-	-	X
Implémentation de la recherche de la cible :	X	X	X
Algorithme de colonies de fourmis	-	-	X
Algorithme path finding	X	X	-
Représentation graphique :	X	X	X
Représentation des boids	-	-	X
Représentation des vecteurs	-	-	X
Représentation des obstacles	X	-	-
Représentation de plusieurs équipes	-	X	-
Interface Graphique :	X	X	-
Paramétrage algorithme de boids	X	-	-
Paramétrage obstacles	X	-	-
Paramétrage path finding	-	X	-
Paramétrage équipes	X	-	-
Site Web :	X	-	-

TABLE 1 – Répartition des tâches

4 Planning

<i>Tâches</i>	Soutenance 1	Soutenance 2	Soutenance 3
Implémentation algo Boids :	90%	100%	100%
Implémentation de la recherche de la cible :	10%	50%	100%
Algorithme de colonies de fourmis	-	40%	100%
Algorithme path finding	20%	60%	100%
Représentation graphique :	70%	75%	100%
Représentation des boids	100%	100%	100%
Représentation des vecteurs	100%	100%	100%
Représentation des obstacles	80%	100%	100%
Représentation de plusieurs équipes	-	-	100%
Interface Graphique :	100%	100%	100%
Paramétrage algorithme de boids	100%	100%	100%
Paramétrage obstacles	100%	100%	100%
Paramétrage path finding	100%	100%	100%
Paramétrage équipes	100%	100%	100%
Site Web :	25%	60%	100%

TABLE 2 – Répartition des tâches

5 Précision des tâches

5.1 Implémentation algo Boids

On a déjà parlé de cet algorithme plus haut dans ce document, nous voulons l'utiliser afin d'établir les règles de déplacement des "robots".

5.2 Implémentation de la recherche de la cible

Pour implementer la recherche de la cible on va utiliser des algorithmes de recherche de chemin, cela consiste a trouver un chemin entre un point A a un point B en prenant en compte differentes contraintes. Il en existe plusieurs mais nous allons utiliser ces deux principaux : l'algorithme de Dijkstra qui sert a chercher le plus court chemin lorsque l'on connait tout les chemins possible. l'algorithme A* qui est très utile lorsque on ne connait pas les chemins possibles. l'implementation de ces deux algorithmes vont permettre la simulation de condition environnementaux connu ou non.

5.3 Représentation graphique

Représentation des boids :

Afin de rendre l'expérience utilisateur plus ergonomique, nous implémenterons une façon de visualiser de les "boids"

Représentation des vecteurs :

Même principe que pour les "boids", il nous faut un moyen de représenter les directions de chacun des éléments

Représentation des obstacles :

Pour ajouter un peu de difficulté à la tache de nos petits robots, nous allons ajouter des obstacles empechant leurs progression. Il faudra donc bien évidemment trouver un moyen d'implementer ceux-ci et de les représenter sur l'interface pour rendre la simulation plus esthetique.

Représentation de plusieurs équipes :

Afin de tester plusieurs l'algorithmes, nous mettrons plusieurs équipes en concurrence. Cela nous permettra de tirer des conclusions sur l'efficacité des différents algorithmes en fonction des configurations de l'arène (obstacles).

5.4 Interface Graphique

Paramétrage algorithme de Boids :

L'algorithme de Boids se base sur 3 règles : la cohésion, la séparation et l'alignement. Notre interface graphique permettra à l'utilisateur de varier ses paramètres pour pouvoir voir l'impact sur la simulation et donc sur le comportement des boids.

Paramétrage obstacles :

En accessionant cette fonctionnalité, l'utilisateur pourra dessiner des obstacles dans l'arène à l'aide de son curseur.

Paramétrage pathfinding :

Cette fonctionnalité permettra à l'utilisateur d'ordonner aux boids de trouver le chemin jusqu'à la position pointée par le curseur de sa souris.

Paramétrage équipes :

Cette fonctionnalité permettra de choisir le nombre d'équipes et les paramètres à associer à chaque équipe.

5.5 Site Web

Nous allons implémenter un site web afin de pouvoir y consigner nos avancées, partager le projet avec d'autres gens et donner de la visibilité à notre projet de manière plus générale.

6 Outils de travail

Nous utiliserons la bibliothèque SDL2 pour représenter la simulation et la bibliothèques GTK pour l'interface graphique.

7 Soutenance 1

7.1 Implémentation de l'algorithme de Boids

Comment les boids sont-ils représentés dans notre implémentation ?

Il n'existe pas de meilleure approche pour représenter un boid, mais dans notre cas, nous avons utilisé des triangles isocèles. Cela a été fait non seulement pour des raisons esthétiques mais aussi pour rendre l'algorithme plus simple à mettre en œuvre et à comprendre.

Pourquoi un triangle ?

Nous pouvons mettre en œuvre le comportement souhaité en déterminant si un autre point se trouve sur le côté gauche ou droit de notre boid en utilisant les deux points qui constituent la base du triangle. L'autre point de notre triangle définira la largeur du champ de vision du boid, et plus le champ de vision est large, plus la hauteur de notre boid doit être grande.

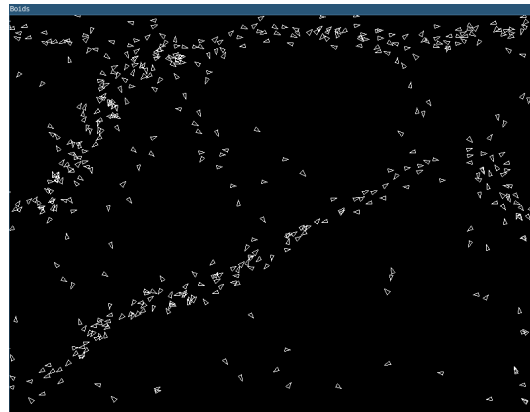
Quels sont les principaux comportements des boids ?

Il y a trois règles :

L'alignement : pour rester regroupés les boids essaient de se suivre sur le même chemin.

La séparation : les boids évitent autant que possible de se coller l'un à l'autre.

La cohésion : les boids ont tendance à se regrouper entre eux.

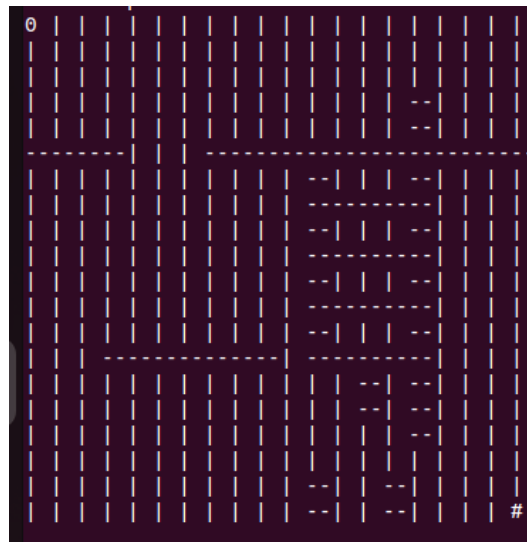


7.2 Implémentation Path finding

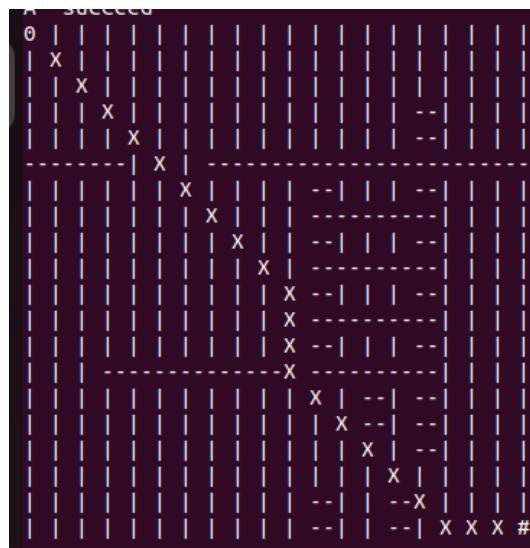
Pour l'implémentation de l'algorithme de pathfinding nous avons d'abord choisis d'implémenter celui de A* qui correspond au mieux à nos demande pour un algorithme de pathfinding le plus general possible. En effet celui-ci recherche le chemins le plus cours entre un point A et un point B en considerant des obstacles. Pour ce faire a cette premiere soutenance l'implémentation de cette algortihme et faite sur vecteur de int avec comme valeur possible 0 pour un emplacement libre et 1 pour un emplacement occupée. L'algorithme ce base sur une structure fifo avec ordre de prioritee basée sur une valeur heuristic. La valeur heuristic se calcule pour chaque noeud u par $heuristic = cout + distance(u, destination)$, le cout correspond au deplacement total depuis le point de depart.

```
struct pqueue{
    struct pqueue* next;
    int x;
    int y;
    int heure;
    int cout;
};
```

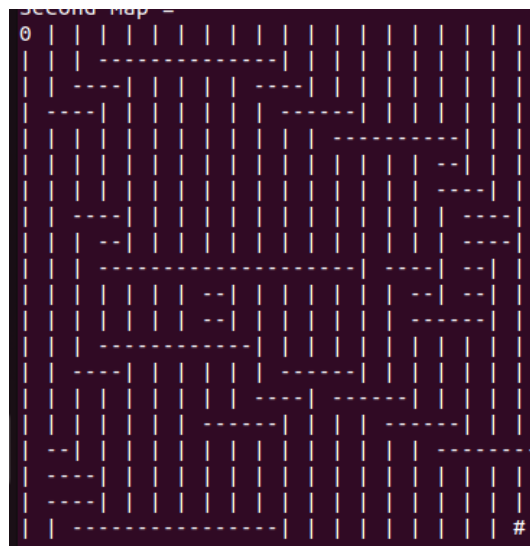
Ici la structure de donnée qui contiendra chaque pixel qui sera atteint par l'algorithme. Ci-dessous une map avec un point de départ (0) et un point d'arrivé (#) ainsi que les obstacles (-) et les postitions possible (—) :



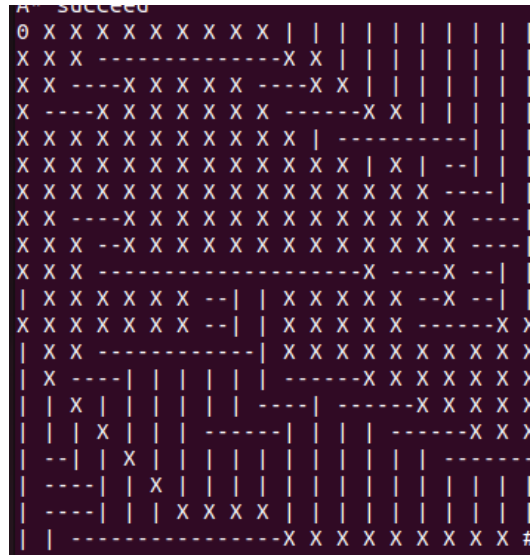
La meme map qui a été resolue par l'algorithme :



Maintenant le probleme de l'algorithme etant qu'il ne retrouve pas tout seul le chemin et sur de map plus complexe comme celle-ci :



Il va renvoyée le chemin mais aussi les cases qui ne sont pas supposée etre pris en compte :

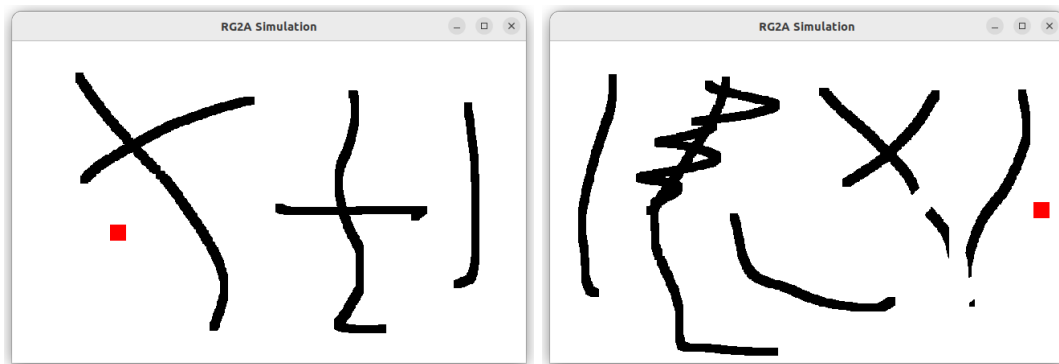


7.3 Représentation graphique

Représentation des obstacles - Paint :

Paint est la première version de l'implémentation des obstacles. Notre Paint fonctionne sur le même principe que le vrai Paint, dessiner sur un plan. A la différence que notre version dessine des rectangles noir en guise d'obstacles sauf sur la cible rouge positionnée de façon aléatoire lorsque l'on presse une touche du clavier.

Le Paint s'activera lorsque le checkbox obstacle(de l'interface graphique) est activé. Lorsque le click gauche est pressé, et que le curseur de la souris n'est pas en collision avec la cible rouge, un rectangle noir de dimension 10x10 est dessiner sur l'arène (en guise d'obstacle). Le click droit à la même fonctionnalité que le click gauche mais celui-ci efface seulement les obstacles.



7.4 Interface graphique

L'interface graphique est composé de 3 parties :

- Les paramètres généraux
- Les paramètres de l'équipe 1
- Les paramètres de l'équipe 2

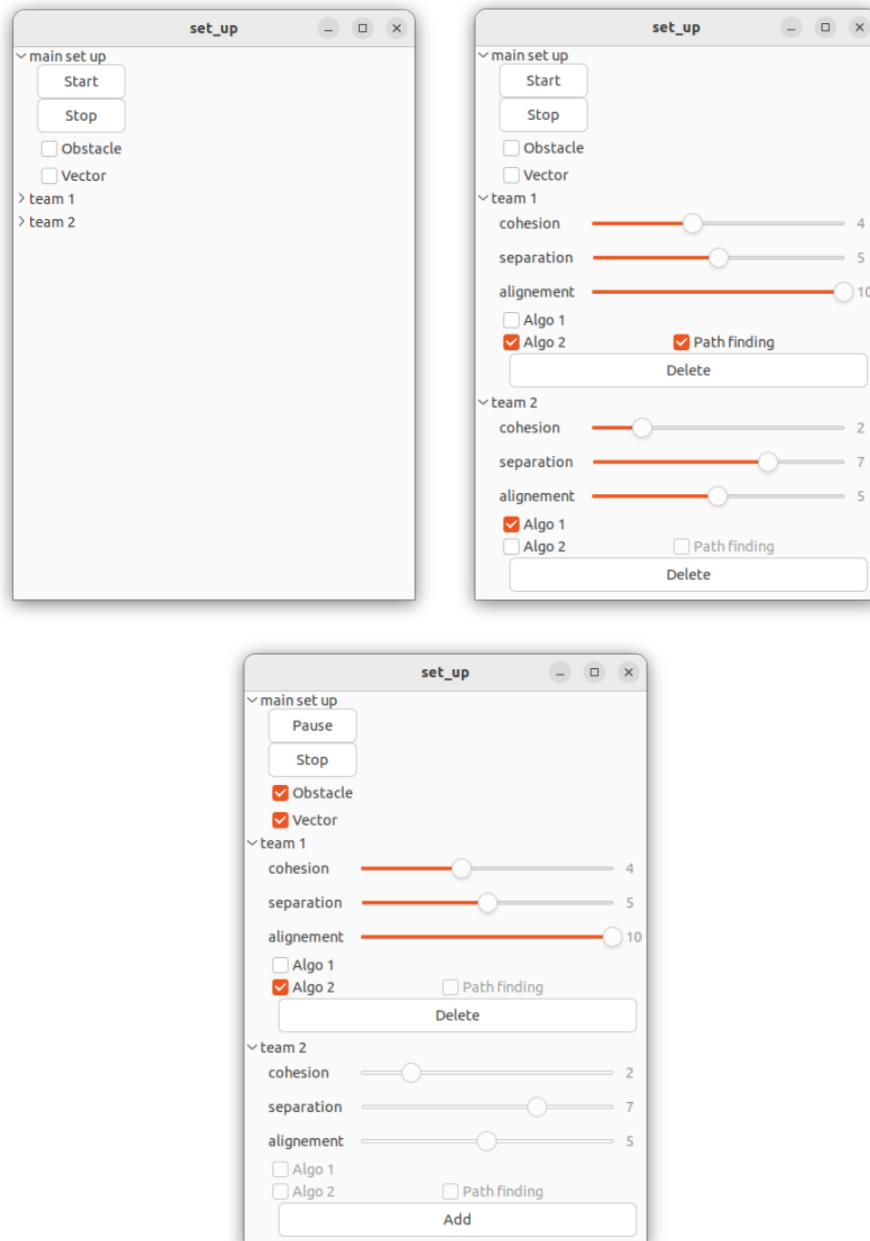
Le paramétrage général permet de paramétrer la simulation de manière générale, ces modifications s'appliquent sur toutes les équipes.

Il est composé de 2 boutons et de 2 checkbuttons :

- Le bouton Start/Pause : permet de lancer la simulation ou de la mettre en pause.
- Le bouton Stop : permet d'arrêter la simulation.
- Le checkbutton Obstacle : permet à l'utilisateur de dessiner des obstacles dans l'arène. Lorsque ce checkbutton est actionné, ceux du pathfinding de l'équipe 1 et 2 sont désactivés, pour que le curseur de la souris ne soit pas utilisé pour plusieurs actions.
- Le checkbutton Vector : permet d'afficher les vecteurs des robots.

Le paramétrage des équipes permet d'appliquer les différentes modifications (coefficient boids, différents algorithmes de recherche, pathfinding) sur chacune des équipes. Il est composé de 3 slider et de 3 checkbutton et d'un bouton.

- Le cohésion/séparation/alignement slider : permettent de modifier les différents coefficients de l'algorithme de boids pour chaque équipe.
- Le checkbutton algo1 : active l'algorithme numéro 1 celui qui se base sur l'algorithme de colonies de fourmis et désactive l'algo2 s'il a été activé.
- Le checkbutton algo2 : active l'algorithme numéro 2 qui s'appuie sur la recherche de chemin (pathfinding) et désactive l'algo1 s'il a été activé.
- Le checkbutton pathfinding : peut être activé que si le checkbutton obstacle et algo1 sont désactivés. Lorsqu'il est activé les robots associés au pathfinding de l'équipe de dirige vers le curseur de la souris.
- Le bouton Delete/Add : supprime/ajoute les robots associés à l'équipe.

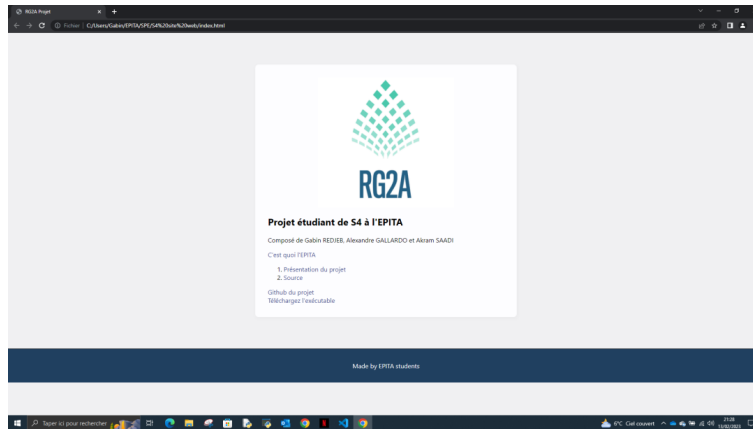


7.5 Site Web

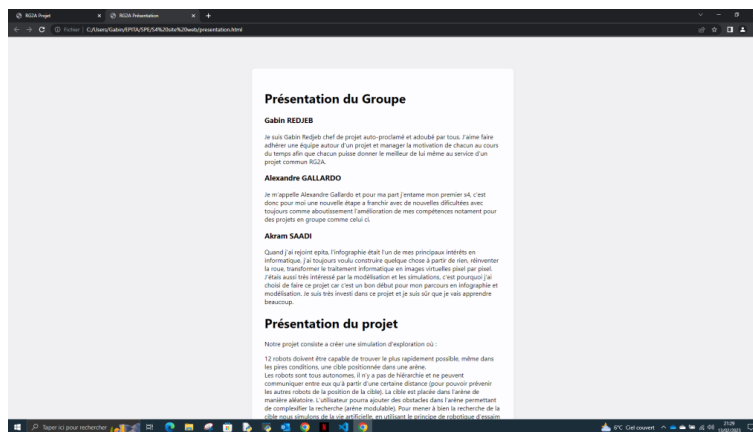
Le site web est composé d'une page d'accueil, d'une page de présentation du groupe et du projet et d'une page amenant aux sources utilisées pour la réalisation du projet.

La page d'accueil comporte 5 liens qui amène à :

- La page de l'EPITA
- La page de présentation du groupe et du projet
- La page source
- La page de projet GitHub
- L'installation de l'exécutable du projet

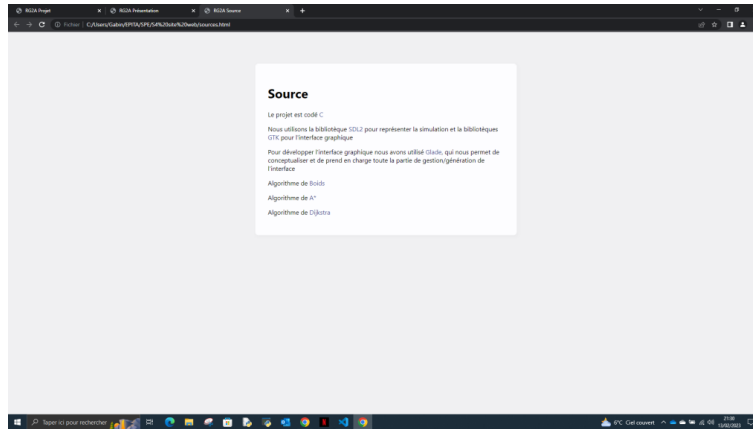


La page de présentation résume l'idée du projet et les objectifs personnels de chaque membre du projet.



La page source explicite les outils utilisés pour le projet, qui sont le langage C, SDL2 pour modéliser l'arène et les robots, GTK/Glade pour conceptualiser l'interface graphique qui va permettre de rendre la simulation modifiables et les algorithmes que nous utilisons tels que Boids, A*, Dijkstra et colonie

de fournis.



8 Conclusion

Nous sommes fier de notre travail vit a vit de cette premiere soutenance et de plus nous avons realise toutes les taches pour la premiere soutenance.