

Projet long de Technologie Objet

Fonctionnalités de l'application Sagittarius

Groupe IJ-1

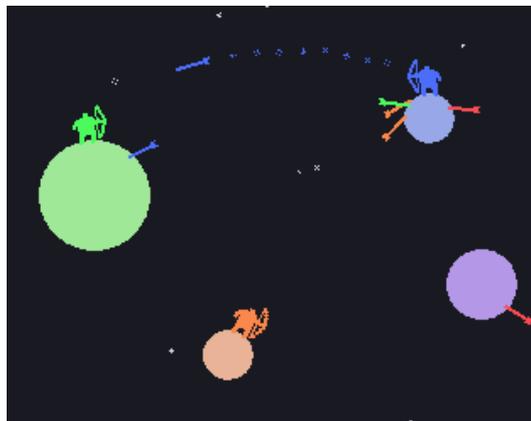
Fainsin Laurent
Guillemin Johan
Guillot Damien
Heurtebise Tom
Jourdan Pierre-eliot
Kirupanathan Nadesan

Département Sciences du Numérique
Première année
2020 — 2021

Table des matières

1	Objectif général du projet	4
2	Fonctionnalités	4
2.1	Génération du terrain	4
2.2	Tirer des flèches	4
2.3	Mourir	4
2.4	Se déplacer	5
2.5	Choisir le nombre de joueurs	5
2.6	Personnaliser son pseudo/couleur	5
2.7	Nombre de vies	5
2.8	(Visualiser la trajectoire des flèches)	5
2.9	(Tableau des scores)	5
2.10	(Densité de planètes)	5
2.11	(Taille de la carte)	5
2.12	(Planète/Lunes mobiles)	6
2.13	(Caméra mobile)	6
2.14	(Gadget)	6
2.15	(Mode par équipe)	6
2.16	(Changer de planète)	6
2.17	(Aimbot)	6
2.18	(IA)	6
2.19	(Mort subite)	6
2.20	(Sauter)	7
2.21	(Mode solo)	7
3	Interfaces utilisateur	7
3.1	Menu principal	7
3.2	Menu pause	8
3.3	Menu customisation archer	8
3.4	Menu option	9
3.5	Menu paramètres de la partie	10
3.6	Affichage de la map	10
4	Scénarios	10
4.1	Scénario n°1	11
4.2	Scénario n°2	11
4.3	Scénario n°3	11

4.4	Scénario n°4	11
4.5	Scénario n°5	11
5	Difficultés	12
5.1	Moteur Graphique	12
5.2	Moteur physique	12
5.3	Risque que rien ne marche	12
6	Références	13



Nous envisageons de créer un programme permettant de jouer à Sagittarius [1], un jeu tour par tour donc l'unique but est de tirer avec un arc et des flèches sur les autres joueurs et où le mécanisme principal est la gravité !

1 Objectif général du projet

L'objectif principal de ce programme est de pouvoir jouer à autant de parties de Sagittarius que l'on souhaite. Lors de la création d'une partie, il se génère aléatoirement un environnement céleste dans lequel des joueurs seront placés. Une partie pourra se jouer de 2 à 4 joueurs. Avant le début des hostilités, les utilisateurs auront le choix de leur couleur, mais aussi des paramètres de la partie. Lors de la partie, les joueurs peuvent se déplacer librement sur leur planète avant de tirer. La partie se déroule tour par tour, cependant ceux-ci seront limités à 30 secondes (par défaut) pour ne pas ralentir le jeu. Une partie se finit lorsqu'il ne reste qu'un seul joueur encore en vie.

2 Fonctionnalités

2.1 Génération du terrain

La génération du terrain se fera aléatoirement. Cependant, les planètes générées devront être suffisamment espacées et en nombre raisonnable (il faudra au moins autant de planètes que de joueurs). Les joueurs seront aléatoirement associés à une planète et avec une position aléatoire sur celle-ci.

2.2 Tirer des flèches

Grâce aux entrées (souris et/ou clavier) les joueurs pourront choisir la direction initiale de la flèche ainsi que sa vitesse initiale. La flèche décrira ensuite une trajectoire, calculée à partir des forces que les planètes exercent sur elle.

2.3 Mourir

Une fonctionnalité qui peut paraître incongrue et qui pourtant est la base même de notre jeu est la possibilité pour le joueur de mourir. Par mourir nous entendons que si jamais le joueur est frappé par une flèche adverse son avatar disparaît de la planète sur laquelle il se trouvait et il perd la partie. Il faudra donc implémenter la disparition des avatars et donc la mort des joueurs. A ceci

nous ajouterons la possibilité pour un joueur d'abandonner pour ne plus prendre part à la partie en cours. Cette option se modélisera par une flèche tirée au-dessus de lui qui fera disparaître le joueur.

2.4 Se déplacer

Les joueurs doivent pouvoir se déplacer sur la surface de leur planète, vers la gauche et la droite (contrôles avec les flèches). Ils doivent ainsi pouvoir faire le tour de leur planète.

2.5 Choisir le nombre de joueurs

Lors du lancement d'une partie, il est nécessaire de définir le nombre de joueurs. Le nombre de joueurs pour une même partie pourra aller de 1 (mode solo) à 4.

2.6 Personnaliser son pseudo/couleur

L'étape suivante (plus secondaire) du lancement d'une partie, sera d'affecter à chaque joueur un nom et une couleur. Chaque joueur pourra ainsi définir le nom qu'il souhaite et choisir sa couleur (qui sera aussi celle de sa planète).

2.7 Nombre de vies

Lors de la sélection des paramètres de la partie, les utilisateurs auront la possibilité de choisir le nombre de flèche nécessaire à l'élimination d'un joueur.

2.8 (Visualiser la trajectoire des flèches)

Pour faciliter la visée (et le debug), l'option d'affichage de la prévision de la trajectoire pourra être activée. Le tracé de la trajectoire de la flèche sera dessiné avant même que la flèche ne soit lancée.

2.9 (Tableau des scores)

Une autre fonctionnalité qui pourra être implanté si le temps le permet est l'affichage des scores et de certaines données. On pourra ainsi afficher le taux de précision de chaque joueur, le nombre d'autres joueurs touchés ainsi que la place obtenue.

2.10 (Densité de planètes)

Lors de l'initialisation de la partie, chaque joueur aura la possibilité de choisir la taille et la densité de sa planète de départ. Les autres planètes seront réparties de manière aléatoire (avec une densité aléatoirement associée) lors de la génération du terrain.

2.11 (Taille de la carte)

Lors de la création de la partie, un paramètre permettra de sélectionner la taille de la carte. Plus celle-ci sera grande, plus la distance séparant deux joueurs/planète sera élevée.

2.12 (Planète/Lunes mobiles)

Lors de la création de la partie, un paramètre permettra de sélectionner si des planètes/lunes mobiles (en orbite) seront présentes dans la partie. Celles-ci viendront ajouter un niveau de difficulté à la partie.

2.13 (Caméra mobile)

Déplacement du point de vue, soit si la carte est trop petite, soit pour avoir une meilleure vision pour le tir et ainsi pour viser plus facilement.

2.14 (Gadget)

Lorsque le joueur touche un objet spécial avec sa flèche, il obtient un bonus/malus.

2.15 (Mode par équipe)

Dans ce mode, la survie ne se fait plus individuellement mais par équipes. Ainsi les joueurs peuvent choisir leur couleur qui définit leur équipe. La dernière couleur en vie remporte la partie.

2.16 (Changer de planète)

Une fonctionnalité optionnelle est la possibilité pour un joueur de changer de planète. Évidemment ce dernier ne peut changer que pour une planète non occupée par un joueur. Cette possibilité lui sera offerte s'il parvient à se débarrasser d'un de ses adversaires et uniquement dans ce cas-là. Au tour qui suivra le « kill », le joueur survivant aura la possibilité de changer de planète pour celle de sa victime. Cette action vient se substituer à la capacité de tirer pour le tour en cours.

2.17 (Aimbot)

Le mode aimbot a pour but de déterminer le meilleur angle et la meilleure puissance de tir afin d'atteindre une cible. Il permettra d'implanter des ordinateurs pour jouer en solo, mais pourra aussi être utilisé par des joueurs mal intentionnés.

2.18 (IA)

Lors de la création des parties, les archers peuvent être pilotés par une intelligence artificielle dont le niveau de difficulté définit leur précision.

2.19 (Mort subite)

Pour éviter que la partie ne s'éternise, on définit un temps (ayant une valeur par défaut mais pouvant être changé au moment de l'initialisation d'une partie) à partir duquel des astéroïdes/météorites apparaissent, de manière régulière, avec des trajectoires aléatoires sur le terrain, susceptibles de tuer un joueur s'ils le touchent.

2.20 (Sauter)

Si nous avons suffisamment de temps pour le faire, nous souhaitons implanter une fonctionnalité qui permettrait au joueur, en plus de se déplacer sur la planète, d'effectuer des sauts pour éviter les flèches envoyées.

2.21 (Mode solo)

Dans ce mode, un unique joueur humain peut s'exercer au travers de niveau dans lesquels il y a différentes cibles qui sont des PNJ (Personnages Non Joueurs). Pour passer au niveau d'après le joueur doit atteindre toutes les cibles. Nous nous réservons le droit d'ajouter une fonctionnalité dans laquelle le joueur dispose d'un nombre limité de flèches avant de recommencer le niveau et une fonctionnalité dans laquelle ce nombre est illimité.

3 Interfaces utilisateur

Voici de *magnifiques* esquisses/exemples d'interfaces que le jeu utilisera.

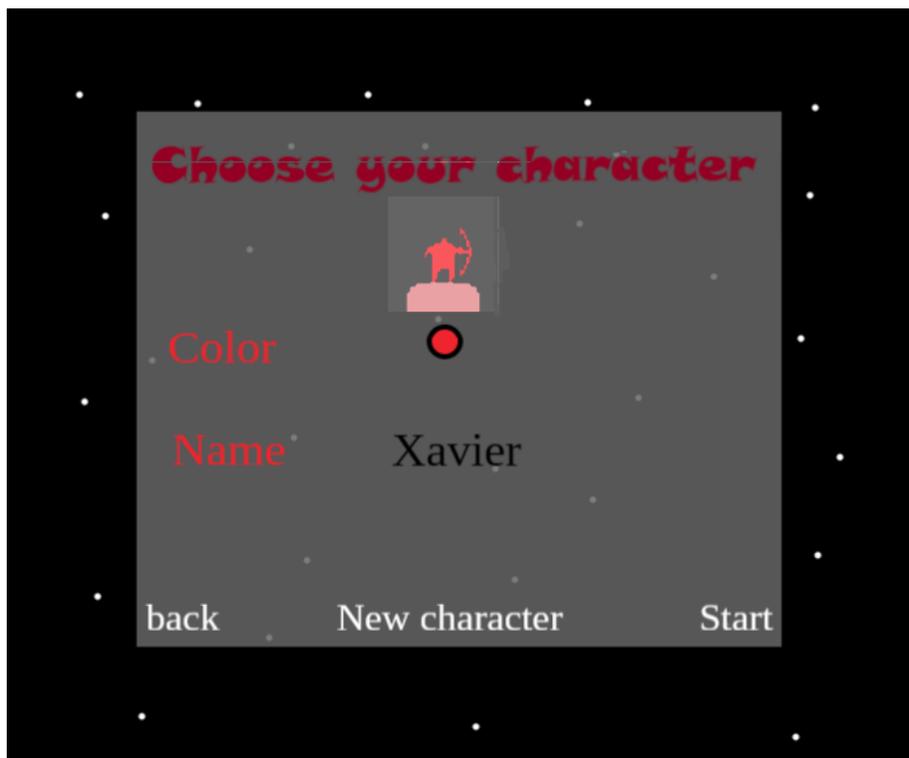
3.1 Menu principal



3.2 Menu pause



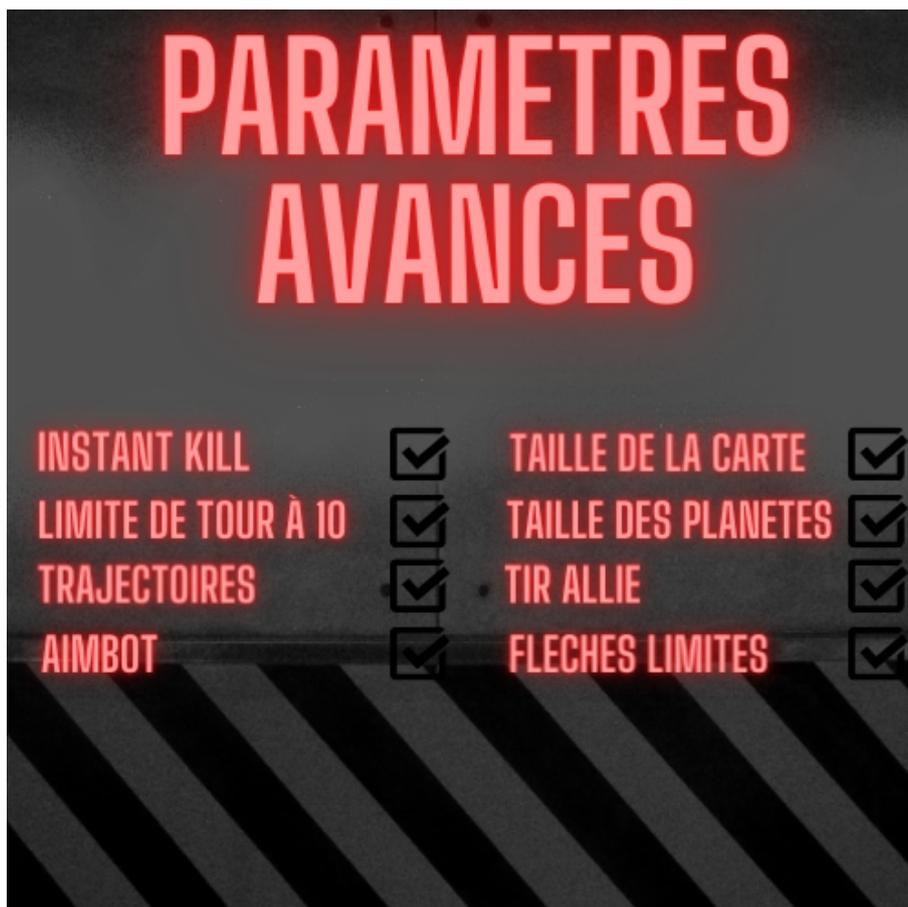
3.3 Menu customisation archer



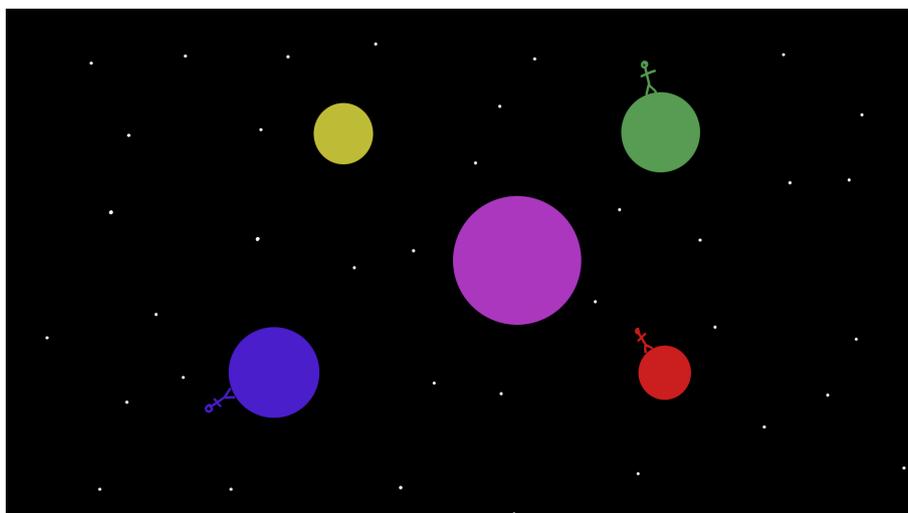
3.4 Menu option



3.5 Menu paramètres de la partie



3.6 Affichage de la map



4 Scénarios

Les scénarios pour tester le jeu sont multiples. Cette section sera sûrement à compléter par la suite mais en voici un premier jet.

4.1 Scénario n°1

Tout d’abord dans le premier scénario nous testerons si le jeu se lance bien et surtout si le mode classique (le mode de 2 à 4 joueurs humains) est fonctionnel. Aucun test automatique ne sera possible puisqu’il s’agira d’interagir avec le jeu directement. Dans l’exécution de ce scénario il faudra donc lancer le programme principal qui affiche l’interface graphique, (le lancement en ligne de commande ou par le biais d’un autre type d’IHM sera à notre convenance. Une fois lancé le menu des paramètres nous vérifierons que le menu principal nous amène bien dans chacun des modes (mode solo, mode à plusieurs, etc. . .). À chaque fois nous reviendrons au menu principal. Suite à cela nous entrerons dans le mode solo. Le test de ce mode se fera suivant un nombre quelconque de joueurs (4 par exemple) et avec des noms/couleurs aussi quelconques. Une fois la partie lancée nous constaterons que l’ensemble des fonctionnalités prévues pour de ce mode fonctionnent (avec une attention particulière sur les trajectoires de chaque flèche, les déplacements des personnages et leur décès). Nous concluons ce test par une victoire rapide d’un des joueurs pour vérifier la présence de l’écran de fin et la possibilité de rejouer/revenir au menu principal.

4.2 Scénario n°2

Le deuxième scénario envisagé concerne le mode solo (si implanté) dans lequel le joueur doit abattre des cibles fixes qui seront des PNJ (cf ce qui précède). Si les deux options de ce mode sont implantées il faudra lancer une partie avec le nombre de flèches limité et une deuxième avec le nombre de flèche illimitée. À ce moment il sera opportun de tester le menu de pause pour interrompre la partie qui peut durer indéfiniment.

4.3 Scénario n°3

Un troisième scénario qui pourra être testé est celui qui concerne le mode par équipes. Classiquement nous lancerons le jeu et ce mode en particulier en créant deux équipes de joueurs, (l’équipe rouge et l’équipe bleue par exemple). Pour ce scénario nous essaierons d’abord l’option “tir allié désactivé en faisant tirer le joueur bleu sur son coéquipier. Si tout se passe bien, ce dernier n’aura aucun dégât. Nous terminerons la partie par la victoire de l’équipe rouge suivie d’une deuxième dans laquelle le tir allié sera activé. Le bleu devrait cette fois mourir face au tir de son coéquipier.

4.4 Scénario n°4

Un quatrième scénario pourra nous permettre de tester des fonctionnalités plus avancées comme le mode “plusieurs vies” dans lequel il faut plusieurs flèches pour abattre un ennemi. Ce sera aussi l’occasion de tester l’utilisation d’objets ou encore la possibilité de changer de planètes. Toutes ces options devront être cochées au moment de la création de la partie.

4.5 Scénario n°5

Un dernier scénario devrait nous permettre de tester les fonctionnalités les plus avancées si elles ont pu être implémentées. Nous pourrions ainsi tenter de jouer sur une carte dont les paramètres comme le nombre de planètes et leur taille sont choisis par l’utilisateur. Ce sera aussi l’occasion de tester des modes expérimentaux comme le “aimbot” et éventuellement le jeu contre des Intelligence Artificielle.

5 Difficultés

5.1 Moteur Graphique

L’affichage graphique sera la partie la plus cruciale du projet car sans affichage pas de jeu, cependant elle va être difficile car nous allons devoir utiliser des bibliothèques annexes [5, 4] que nous ne maîtrisons pas forcément. Cela va donc nous demander un apprentissage il y a le risque de ne pas obtenir le rendu espéré.

5.2 Moteur physique

La trajectoire des flèches est un élément très important du projet. Toute la physique des flèches et donc du gameplay en dépendra. Il faudra trouver un moyen de calculer les trajectoires de manière efficace afin que le jeu reste fluide [3]. Il faudra également trouver un compromis entre réalisme des trajectoires et expérience de jeu.

5.3 Risque que rien ne marche

Comme dit précédemment, la création d’un “moteur physique” ainsi que l’utilisation d’un “moteur graphique” sont des tâches assez difficiles séparément, il y a donc un risque non négligeable que rien ne fonctionne si l’une des deux parties ne fonctionne pas.

6 Références

- [1] Itch.io, Sagittarius, George Prosser
<https://gprosser.itch.io/sagittarius>

- [2] Wikipedia, n-body problem
https://en.wikipedia.org/wiki/N-body_problem
https://en.wikipedia.org/wiki/Three-body_problem

- [3] Wikipedia, Verlet integration
https://en.wikipedia.org/wiki/Verlet_integration
<https://gamedev.stackexchange.com/a/41917>

- [4] Swing
[https://en.wikipedia.org/wiki/Swing_\(Java\)](https://en.wikipedia.org/wiki/Swing_(Java))
<https://docs.oracle.com/javase/8/docs/technotes/guides/swing/>

- [5] libGDX
<https://libgdx.com/>