

Rapport de soutenance - 2

Héphaïstos

*Groupe : G.O.
Avril 2023*



Lenny Chiadmi-Delage, Ryan Dahout, Clément Chang, Yvan Fedaouche

Table des matières

1	Introduction	2
1.1	Rappel du projet	2
1.2	Avancement lors de la dernière soutenance	2
2	Moteur Audio et Interfaces d'édition de carte	3
2.1	Moteur Audio	3
2.1.1	Descriptif du travail réalisé	3
2.1.2	Récapitulatif de l'avancement	3
2.1.3	Difficultés rencontrés	3
2.2	Conclusion	3
2.3	Interfaces d'édition de carte	4
2.3.1	Descriptif du travail réalisé	4
2.3.2	Récapitulatif de l'avancement	4
2.3.3	Difficultés surmontées	4
2.4	Conclusion	4
3	IA DES NPC	5
3.1	Ce qui était convenu de faire :	5
3.2	Ce qui a été effectué pour la première soutenance :	5
3.3	Déroulement du travail jusqu'à la première soutenance :	6
3.4	Les difficultés surmontées :	9
3.5	Ce qui est envisagé d'être fait pour la soutenance finale :	9
4	Physix	10
4.1	Objectif du travail	10
4.2	Rappels	10
4.3	Travail réalisé	10
4.4	Les difficultés rencontrées	10
4.5	Conclusion	10
5	Moteur de rendu et Cartes	11
5.1	Moteur de rendu	11
5.1.1	Des cases plus complexes.	12
5.1.2	Les entités.	14
5.1.3	Luminosité, précision des rayons et draw distance.	16
5.2	Cartes.	17
5.3	Limitations actuelles.	17
5.4	Récapitulatif de l'avancement.	17
5.5	Difficultés rencontrées.	17
5.6	Objectifs pour la prochaine soutenance.	17
5.7	Conclusion.	17
6	Website	18
6.1	Introduction	18
6.2	Avancée du travail	18
6.3	Ce qui est prévu pour la prochaine soutenance	20
6.4	Conclusion	20
7	Conclusion Globale	21
7.1	Planning originel	21
7.2	Avancement actuel	21
8	Annexes	22

1 Introduction

1.1 Rappel du projet

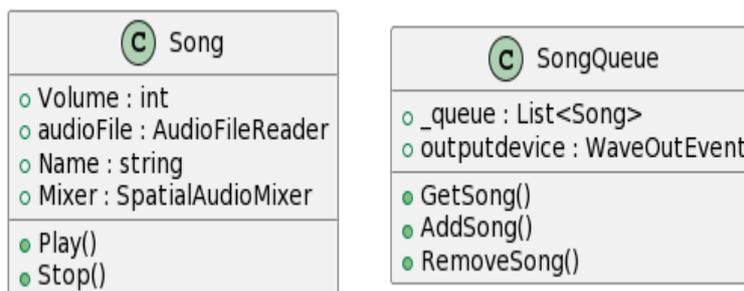
Pour rappel le but de ce projet est simple, créer un moteur de jeu fonctionnel. Celui-ci sera constitué de 2 grandes parties, la partie logiciel et bibliothèque. Tandis que la partie logiciel commencera à être implémentée et fonctionnelle à partir de la prochaine soutenance, la partie bibliothèque est déjà partiellement implémentée et fonctionnelle.

1.2 Avancement lors de la dernière soutenance

	Prévisions	Réalisé
Moteur de rendu	25%	50%
Moteur Physique	20%	40%
Moteur Audio	50%	50%
IA	25%	35%
Cartes	50%	50%
Interfaces	25%	25%

2 Moteur Audio et Interfaces d'édition de carte

2.1 Moteur Audio



2.1.1 Descriptif du travail réalisé

Pour la section moteur audio, le groupe devait pour la seconde soutenance, être capable de présenter 75% du travail. Cette fois-ci, l'équipe du secteur audio n'a pas rencontré d'impasse quant à l'implémentation des nouvelles méthodes.

2.1.2 Récapitulatif de l'avancement

Sur la précédente présentation, le groupe avait mis en avant, sa liste de bande son ainsi que ses méthodes "Play" permettant de jouer la piste, "Next" pour passer à la piste suivante et enfin "Stop" pour arrêter totalement la lecture. A cela s'ajoutait le convertisseur de fichier audio MP3 et WAV pour une meilleure homogénéité. Ce beau paquet était déjà un bon avancement pour le secteur audio, mais la lecture, l'arrêt et le saut de piste musicale en faisait un parfait lecteur audio et non un moteur audio. C'est pour cela que l'équipe a implémenté la spatialisation du son ainsi que la jouabilité des pistes en simultanée. La spatialisation favorisera l'immersion de l'utilisateur dans le jeu, il permet de jouir d'un son qui s'adaptera aux différentes positions d'émissions des bruits par les objets qui l'entoure. La simultanéité de lecture de pistes contribue lui aussi à l'immersion de l'utilisateur dans l'environnement qui l'entoure. Il est bien plus agréable de bénéficier de ses nouvelles fonctionnalités qui permettent de se plonger totalement dans le jeu.

2.1.3 Difficultés rencontrés

L'implémentation de la spatialisation, s'est vu ralenti. Malheureusement, le groupe a rencontré des problèmes pour la création de la spatiation, l'apprentissage de la nouvelle bibliothèque était bien plus compliqué et a demandé plus d'attention que prévu.

2.2 Conclusion

Malgré l'invalidité et la complexité qu'est le défi de la spatialisation, le groupe a su faire progresser son implémentation. Pour la prochaine soutenance, nous aurons fixé les derniers petits soucis et serons capable de vous présenter le moteur audio dans son intégralité.

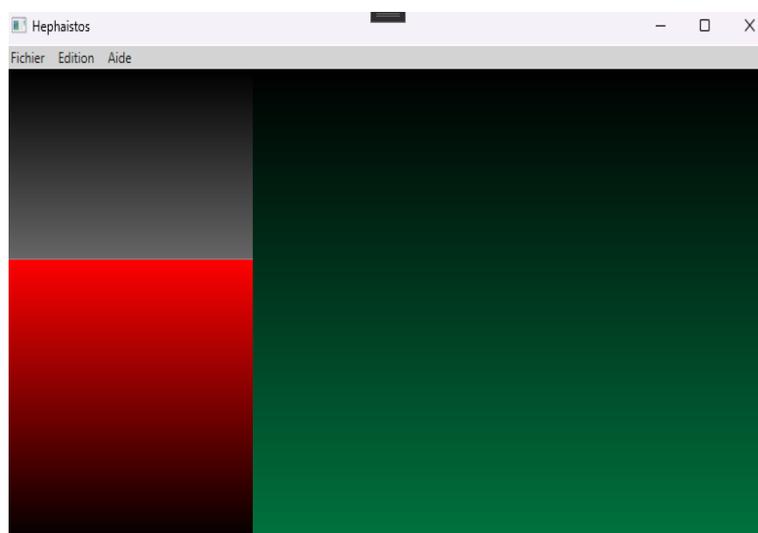
2.3 Interfaces d'édition de carte

2.3.1 Descriptif du travail réalisé

L'avancée de l'interface permet au groupe de présenter comme il se devait, 50% de l'interface finale. De nouvelles fenêtres apparaissent et l'interface prend enfin forme !

2.3.2 Récapitulatif de l'avancement

L'interface utilisateur s'est vu totalement changée. Afin de répondre au mieux aux attentes du groupe, l'interface a été totalement repensée. Cependant, la barre d'outil permettant à l'utilisateur de créer, d'ouvrir ou encore d'entregister son projet reste toujours d'actualité. Les changements les plus flagrants se situent au centre de la fenêtre d'affichage. Apparaissent maintenant 3 cases qui auront chacune un rôle important pour la conception du projet. L'un d'eux contiendra l'arborescence du projet de l'utilisateur, le suivant concentrera les différentes textures possibles et enfin la dernière étant la plus grande, aura pour but d'afficher l'édition de la carte du joueur.



2.3.3 Difficultés surmontées

La grande difficulté aura été de tout repenser. Le groupe avait créé une interface sous forme de Winform. Il s'avèrait que l'implémentation du projet était bien plus compliqué en Winform plutôt qu'en WPF (Windows Presentation Foundation). C'est alors que l'équipe s'est tournée vers une réalisation en WPF. La façon de développer n'était plus identique. Il a donc fallu apprendre à utiliser une nouvelle bibliothèque. Mais le secteur a réussi à relever le défi et vous propose pour cette seconde soutenance les 50% convenus.

2.4 Conclusion

L'interface commence à prendre forme ! Parti d'une simple page blanche, le groupe a su la décorer et lui donner forme, allant de la première barre d'outils à la représentation graphique du logiciel d'édition de carte. Il ne reste plus qu'à lier les tâches des autres secteurs afin de vous présenter le projet Héphaistos dans son entièreté !

3 IA DES NPC

Voici le récapitulatif de notre travail sur l'IA.

3.1 Ce qui était convenu de faire :

Concernant l'IA des NPC, nous devions en effectuer 50% pour la deuxième soutenance. Ce qui a été plus que respecté par le groupe . Jusqu'ici, l'objectif était de créer la structure d'un code cohérent, d'implémenter quelques types de NPC (par exemple : au comportement passif, actif, agressif) et d'implémenter les interactions joueur/NPC. Ces différents types de NPC sont des comportements passifs, actifs, agressifs, fuyatifs. Les interactions joueurs, quant à elles, désignent les attaques, les soins qu'effectuent les NPC aux joueurs. L'objectif était de rendre notre partie sur les IA plus variée et qualitative.

3.2 Ce qui a été effectué pour la première soutenance :

Lors de la première soutenance, le groupe a présenté le début de notre travail sur l'IA des NPC. Ce dernier se traduisait par une structure stable et cohérente afin de faciliter et d'optimiser le futur travail. Le groupe avait implémenté les premiers comportements de NPC : des comportements actifs avec des NPC patrouilleur sur l'axe X, l'axe Y et un axe diagonal. Le groupe a ensuite implémenté les comportements agressifs et fuyatifs avec deux types de NPC : le premier reposant sur l'algorithme A^* poursuit le joueur et le deuxième fuit le joueur en allant sur la position accessible la plus éloignée du joueur (en nous inspirant de l'algorithme A^*). Pour rappel, l'algorithme A^* est un algorithme heuristique qui permet de trouver très rapidement un plus court chemin entre deux points avec d'éventuels obstacles. Outre sa vitesse, cet algorithme est réputé pour garantir une solution en sortie. Néanmoins, cela n'était qu'une partie de notre travail car ces NPC n'avaient pas d'interactions avec le joueur. Ainsi, notre objectif était de les implémenter pour cette deuxième soutenance.

3.3 Déroulement du travail jusqu'à la première soutenance :

1. Le groupe a commencé par implémenter deux classes d'armes :
 - AssaultWeapon représentant les armes d'assauts
 - HealingWeapon représentant les armes de soins

2. NPC agressif

Nous avons implémenté un NPC agressif qui poursuit le joueur dès qu'il est entré dans son champ de vision.

Son champ de vision est une zone autour de lui.

Il est passif puis actif.

Dès que le joueur est à proximité du NPC, ce dernier l'attaque avec son arme AssaultWeapon.

Après cela, si le joueur est toujours en vie le jeu continue mais s'il n'a plus de vie le jeu s'arrête et la fenêtre se ferme.

Le NPC perd de la vie au fur et à mesure du temps et si sa vie atteint 0 il ne fait plus rien.



3. NPC fuyitif

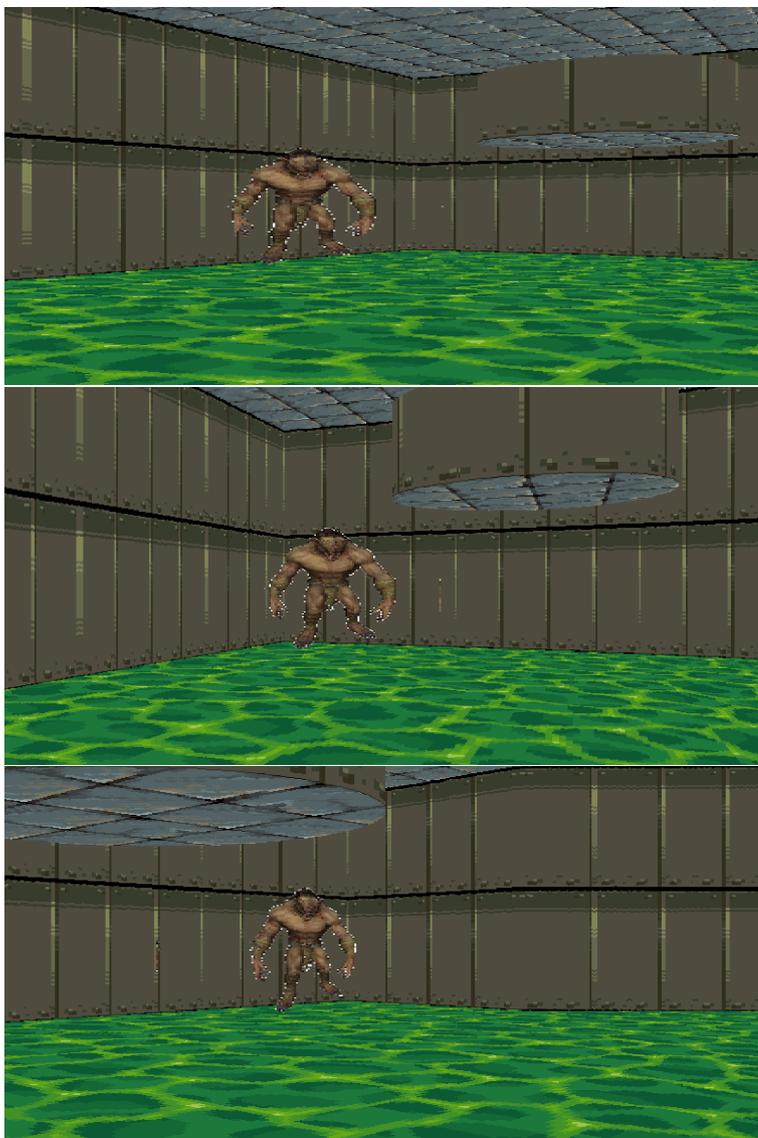
Nous avons implémenté un NPC fuyitif qui fuit le joueur dès que ce dernier est entré dans son champ de vision.

Son champ de vision est une zone autour de lui.

Il est passif puis actif.

Dès que le joueur est à proximité du NPC, ce dernier est obligé de le soigner avec son arme HealingWeapon.

Si le joueur a déjà sa vie au maximum, ce dernier n'en reçoit pas plus sinon il en gagne. Le NPC perd de la vie au fur et à mesure du temps et si sa vie atteint 0 il ne fait plus rien.



4. NPC patrouilleur X

Nous avons implémenté un NPC patrouillant sur l'axe X qui patrouille sur cet axe jusqu'à ce que le joueur soit dans son champ de vision.

Ce NPC peut voir sur l'axe X de sa position actuelle.

Il est actif.

Dès que le joueur est sur un de cet axe, le NPC inflige des dégâts au joueur .

Après cela, si le joueur est toujours est vie le jeu continue mais s'il il n'a plus de vie le jeu s'arrête et la fenêtre se ferme.

Le NPC perd de la vie au fur et à mesure du temps et si sa vie atteint 0 il ne fait plus rien.

5. NPC patrouilleur Z

Nous avons implémenté un NPC patrouillant sur l'axe Z qui patrouille sur cet axe jusqu'à ce que le joueur soit dans son champ de vision.

Ce NPC peut voir sur l'axe Z de sa position actuelle.

Il est actif.

Dès que le joueur est sur cet axe, le NPC inflige des dégâts au joueur.

Après cela, si le joueur est toujours est vie le jeu continue mais s'il il n'a plus de vie le jeu s'arrête et la fenêtre se ferme.

Le NPC perd de la vie au fur et à mesure du temps et si sa vie atteint 0 il ne fait plus rien.

6. NPC patrouilleur Diagonale

Nous avons implémenté un NPC patrouillant sur un axe diagonal qui patrouille sur cet axe jusqu'à ce que le joueur soit dans son champ de vision.

Son champ de vision est une zone autour de lui.

Il est actif.

Dès que le joueur est dans son champ de vision, le NPC inflige de lourds dégâts équivalent à sa vie, comme s'il explosait, au joueur (sa vie est mise à 0).

Après cela, si le joueur est toujours est vie le jeu continue mais s'il il n'a plus de vie le jeu s'arrête et la fenêtre se ferme.

Le NPC perd de la vie au fur et à mesure du temps et si sa vie atteint 0 il ne fait plus rien.

3.4 Les difficultés surmontées :

Le groupe s'est rendu compte d'un problème qui faisait que les NPC ne prenaient pas en compte un seul et même joueur. Chaque NPC prenait en compte un joueur différent. Ce qui compromettait le rendement de notre travail.

Nous nous sommes rendus compte que les NPC patrouilleurs ne faisaient seulement un aller sans retour puis bouclaient entre deux positions. Nous avons donc ajouté une variable afin qu'ils fassent des aller-retour sans arrêt.

3.5 Ce qui est envisagé d'être fait pour la soutenance finale :

Etant donné qu'à présent la structure de notre travail est cohérente et bien mise en place et que les comportements principaux et interactions importantes sont implémentés, nous allons approfondir et améliorer notre travail pour la soutenance finale. Ce travail reposera sur l'ajout d'interactions des NPC avec le joueur. Nous ajouterons aussi peut-être de nouveaux types de NPC. Notre travail sur l'IA des NPC gagnera en qualité, en richesse et en variété.

4 Physix

4.1 Objectif du travail

L'objectif de ce travail est de développer un moteur physique pour simuler les interactions physiques entre les objets dans un environnement virtuel afin d'élargir le champ des possibles. Le moteur physique doit être capable de gérer la dynamique des objets en prenant en compte les forces qui s'exercent sur eux.

Aujourd'hui plusieurs moteurs physiques sont disponibles sur le marché, tels que Bullet, PhysX ou même Box2D. Ces moteurs physiques sont utilisés pour simuler des jeux vidéos, des applications de réalité virtuelle, des simulations en tout genre et autres domaines nécessitant un déplacement de corps dans un monde virtuel. Chacun de ces moteurs a pour but d'être le plus fidèle à la réalité afin de proposer une meilleure immersion dans la simulation créée.

4.2 Rappels

Physix est le nom de notre moteur physique et tient son nom de celui de Nvidia (PhysX). Notre moteur physique suit la méthode d'Euler pour la simulation de la dynamique des objets.

Comment ça marche ?

La méthode d'Euler est une méthode numérique pour la simulation de la dynamique des objets. Nous avons décidé d'implémenter cette méthode plutôt que celle de Verlet qui est plus précise car elle est beaucoup moins coûteuse en ressources que celle de Verlet qui prend en compte la position actuelle et précédente pour calculer sa prochaine position. De plus, grâce à la méthode d'Euler nous sommes libres d'implémenter d'autres méthodes pour améliorer la précision et la stabilité de la simulation pour atteindre une précision de calcul pour ressembler à celle de Verlet. Même si la méthode de Verlet est plus stable, plus précise, symétrique dans le temps et permet aussi de conserver l'énergie des objets, Physix est un moteur physique simplifié et c'est pourquoi nous avons opté pour la méthode d'Euler. La méthode d'Euler calcule la nouvelle position de chaque objet à partir de sa position actuelle, son accélération et sa vitesse à chaque étape.

4.3 Travail réalisé

Lors de la dernière soutenance, le moteur Physix était capable de gérer les objets présents dans une simulation grâce à la formule d'Euler, en les faisant interagir entre eux ou en leur appliquant une certaine gravité.

Pour cette soutenance, quelques calculs ont été ajustés afin d'obtenir des meilleurs résultats.

4.4 Les difficultés rencontrées

En raison de conflits entre l'implémentation du moteur de rendu et le moteur physique, les bugs persistent tels que : les collisions qui ne sont pas entièrement prises en compte et le fait de convertir les sprites en objets physiques qui font planter le jeu lorsqu'il y a trop de sprites présents sur la carte.

4.5 Conclusion

Malgré quelques bugs, le moteur physique est bien avancé mais n'est pas encore parfait. Pour la dernière soutenance, nous prévoyons de corriger ces bugs et de permettre au joueur d'interagir avec les objets.



5 Moteur de rendu et Cartes

5.1 Moteur de rendu

Pour rappel le raycasting est une technique de rendu 3D principalement utilisée dans les jeux vidéo des années 90, qui à partir d'une simple matrice en 2D permettait de réaliser en temps réel un environnement qui a l'apparence d'une 3D rudimentaire à l'image de Wolfenstein (1992) et Doom 64 (1997). Pour la soutenance précédente la base d'un moteur raycasting à été réalisé et était accompagné de fonctionnalités supplémentaires telles que :

- Différents types de murs
- La possibilité de placer des murs en hauteur
- La possibilité de configurer la taille des murs
- Un axe Z (hauteur) sur lequel le joueur peut se déplacer librement (Mais ne peut orienter la caméra sur cet axe)
- Un système de sol, plafond et toits.

L'ensemble des fonctionnalités une fois réunies permettent d'obtenir un rendu beaucoup plus complexe que celui d'un moteur raycasting comme celui de Wolfenstein 3d.



Image 1 - Ancien Moteur de rendu.

Cependant l'ancien moteur de rendu possédait des limitations ou problèmes aujourd'hui outrepassés tels que : l'impossibilité de placer plusieurs murs au sein du même case, les sols, plafonds et toits devenaient transparents lorsque la caméra était trop proche de ceux-ci et il était impossible de représenter des objets en mouvement entre des cases de façon simple.

5.1.1 Des cases plus complexes.

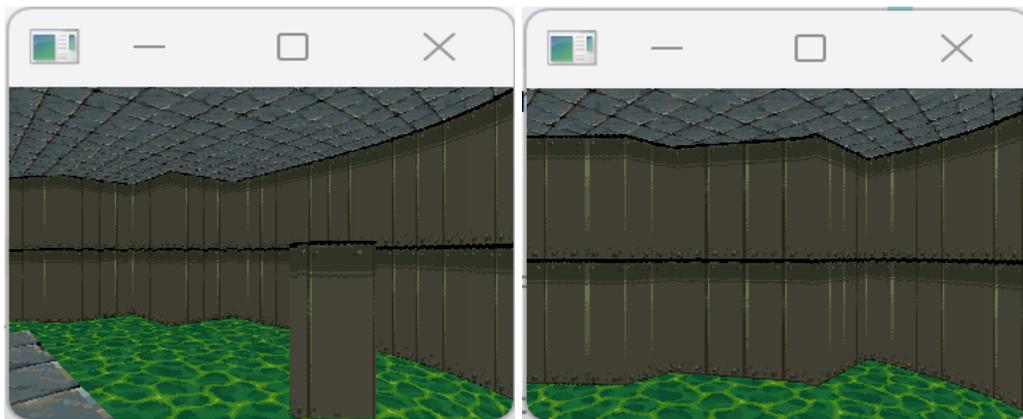
De nouvelles fonctionnalités ont été implémentées permettant au moteur de rendu d'afficher plusieurs murs (ainsi que leurs toits et dessous) au sein d'une même case. Cette fonctionnalité offre aux cartes de nombreuses nouvelles possibilités par exemple représenter un bâtiment à plusieurs étages.



Image 2 - Cases complexes.

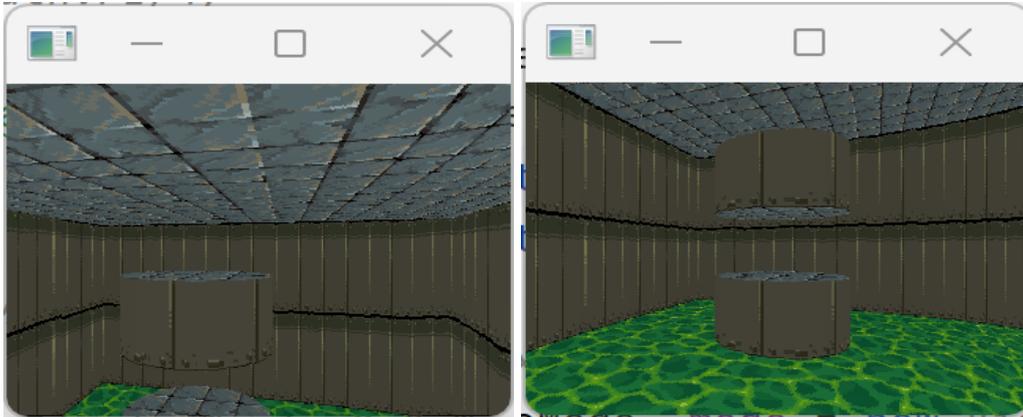
Cependant ce système possède une principale limitation, tous les murs au sein de la même case doivent posséder, ainsi que la même forme (leur hauteur peut être différente). Cette limitation est due à des raisons d'optimisation. Les formes de cases ont aussi reçu une mise à jour :

- Les cylindre possèdent un rayon configurable par l'utilisateur.
- Les triangles possèdent quant à eux un "angle" configurable.



Images 3 et 4 - Cases modulables.

Pour en finir avec les cases celles-ci ont désormais la possibilité de posséder un toit et un dessous en même temps :



Images 5 et 6 - Toit et dessous.

5.1.2 Les entités.

Le système d'entités est la nouveauté la plus importante de la soutenance. Pour faire simple, celles-ci permettent de tout faire, cependant elles sont à utiliser avec modération dû à un coup élevé en ressources. Pour résumer ce qu'elles sont, il s'agit de rectangles de taille totalement configurable, pivotables et pouvant se mouvoir entre les cases.

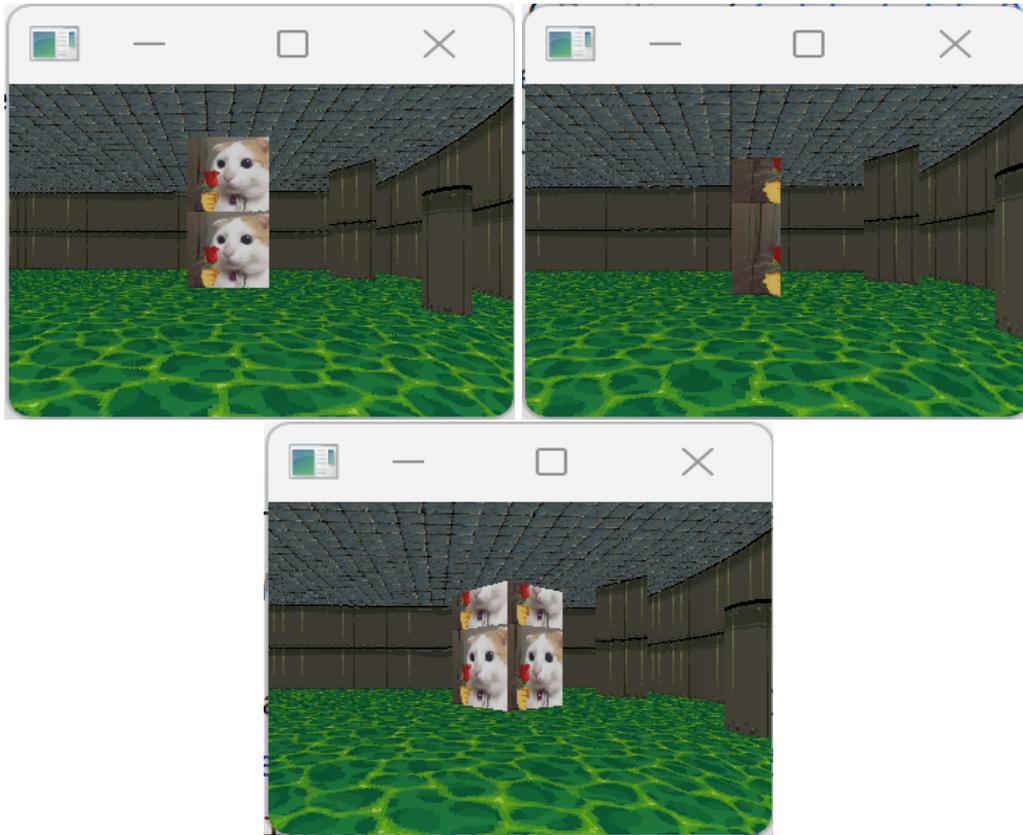


Image 7, 8 et 9 - Entité de formes différentes et pivotables.

De plus, les entités possèdent deux subtilités décuplant leur nombre de cas d'utilisation :

- Chaque face d'une entité est individuelle et peut en conséquence posséder une texture différente des autres.
- Les textures des entités ont la possibilité d'être partiellement ou totalement transparentes.

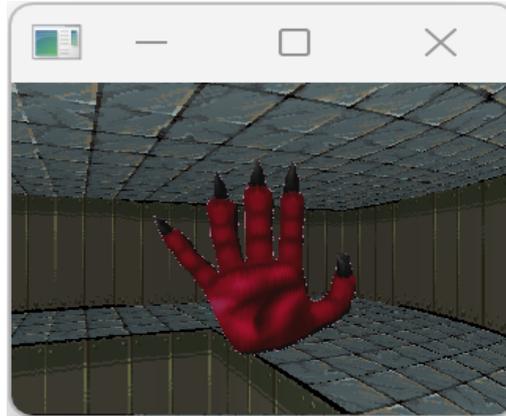


Image 10 - Monstre main.

Les entités peuvent donc être utilisées en tant qu'image constamment orientée vers le joueur ou être utilisées à d'autres fins comme pour faire des grilles, porte coulissante, etc...

5.1.3 Luminosité, précision des rayons et draw distance.

En infographie, la draw distance est la distance maximum d'affichage des objets dans une scène en 3 dimensions. En raycasting, cela se traduit par un arrêt des rayons après une certaine distance parcourue. De plus, ces rayons ont désormais la possibilité de devenir de moins en moins précis en fonction de la distance parcourue permettant d'économiser de la puissance de calcul et d'afficher des murs de plus en plus sombre en fonction de la distance parcourue pour obtenir un rendu plus naturel.

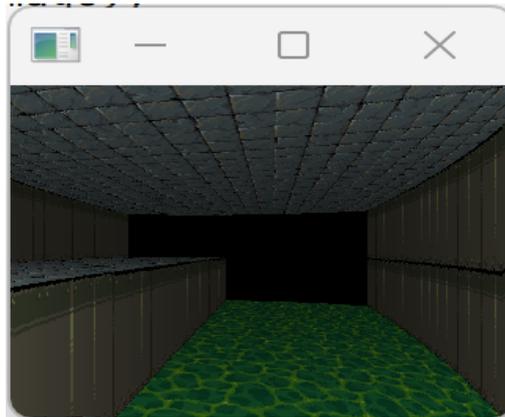


Image 11 - Draw distance et luminosité sombre.

Enfin, la possibilité de configurer la précision des rayons en fonction de la distance du joueur par rapport au point atteint permet de résoudre un problème présent lors de la première soutenance (Les sols, plafonds et toits devenaient transparents lorsque la caméra était trop proche de ceux-ci.).

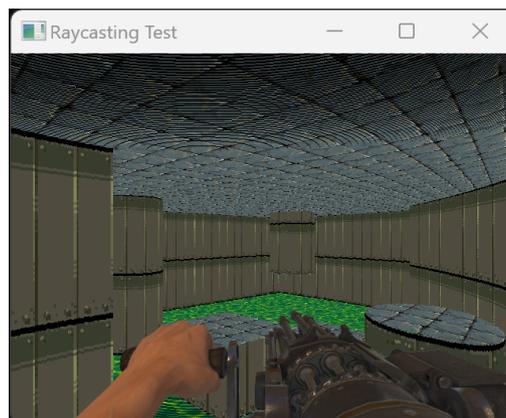


Image 12 - Ancien bug

5.2 Cartes.

Pour rappel, les cartes sont découpées en plusieurs tronçons d'une taille prédéfinie et uniforme et chargées dans une matrice à 3 dimensions pour en faciliter l'accès. Les cartes sont également automatiquement entourées de murs lors de leur création pour éviter tout problème lié à l'affichage. De plus, celles-ci possèdent des fonctions leur permettant d'être stockées et chargées depuis des fichiers en deux lignes de code.

```
73 Carte map3 = new Carte((14, 7, 2), ChunkSize: 7);
74 map3.Load(path: "Demo.hep");
```

Image 13 - Utilisation des fichiers contenant les cartes.

5.3 Limitations actuelles.

Le moteur de rendu possède encore plusieurs grosses limitations. En voici une liste exhaustive de celles-ci : Impossibilité de placer plusieurs murs de formes différentes au sein d'une même case, d'orienter la caméra verticalement et d'afficher un toit ou un dessous aux entités.

5.4 Récapitulatif de l'avancement.

Pour récapituler l'avancement le moteur de rendu permet de retranscrire à l'écran un espace constitué de multiples formes de tailles variables dans lesquelles on peut se mouvoir et orienter son regard horizontalement librement, dans lequel peut être représenté des objets en mouvement sur n'importe quel axe et qui adapte le niveau de détail des objets à afficher en fonction de la distance du joueur par rapport à ceux-ci.

5.5 Difficultés rencontrées.

La difficulté principale rencontrée lors de la réalisation du moteur de rendu est le manque de documentation pour l'implémentation des entités, aucune des techniques documentées fonctionnait correctement (Et n'étaient d'ailleurs pas totalement adaptée aux algorithmes utilisés), enfin l'autre difficulté principale a été l'optimisation.

5.6 Objectifs pour la prochaine soutenance.

Les principaux objectifs pour la prochaine soutenance sont l'implémentation d'un système de texture mapping permettant à l'utilisateur de configurer le placement et la méthode d'affichage des textures au sein de la carte, d'optimiser le moteur de rendu et de le rendre plus facile d'utilisation (à l'aide de configurations pré faite, de fonctions avec moins de paramètres, etc. . .).

5.7 Conclusion.

Pour conclure, le système de cartes / niveaux est fini comme annoncé, tandis que le moteur de rendu est en avance sur les prévisions données dans le cahier des charges.

6 Website

6.1 Introduction

Nous avons décidé de faire notre site internet en "Responsive" (c-à-d : Le site reste lisible peu importe l'appareil utilisé) afin qu'il soit plus agréable à regarder

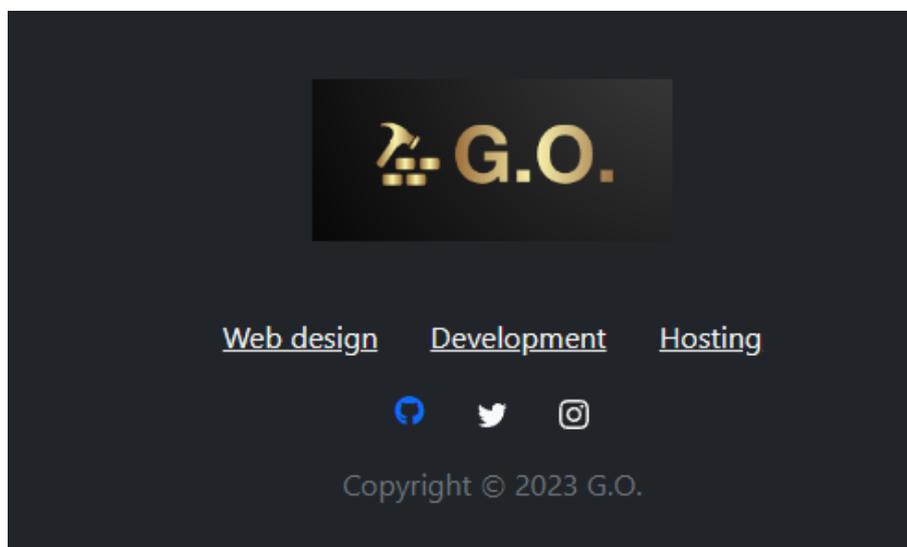
6.2 Avancée du travail

Sur chaque page du site, il y a un "Header" afin de faciliter la navigation du site.



Header du site

Il y a également un "Footer" qui redirigent vers des liens utiles.



Footer du site

La page "*index.html*" de notre site souhaite la bienvenue dans l'ancre d'Héphaïstos et propose des catégories pour en savoir davantage sur notre projet.



La page d'accueil de notre site

Le site contient une page À propos, qui présente les disciples d'Héphaïstos. Chaque membre a une phrase qui le caractérise et une photo. Certaines informations sur les membres ont aussi été transmises.



La présentation des disciples d'Héphaïstos

Il y a aussi une page pour consulter les documents du projet (Cahier des charges, Rapports de soutenance, Plan de soutenance etc...)



Les différents documents

6.3 Ce qui est prévu pour la prochaine soutenance

Pour la prochaine soutenance, il est prévu de mettre à jour les pages "Patch Note" et "Documents". Nous avons prévu de faire une page personnalisée pour l'erreur "404", une page "Lore" expliquant les raisons de notre haine envers Unity et un page "Tuto" qui sera le chemin à suivre pour maîtriser les outils d'Héphaïstos et devenir un de ses disciples. Enfin il est prévu de corriger d'ajouter tous les liens manquants tels que le lien de téléchargement qui ne fonctionne pas encore.

6.4 Conclusion

Le site est presque terminé, il ne manque plus qu'à le mettre à jour pour la prochaine soutenance et faire les ajouts prévus pour avoir le site final.

7 Conclusion Globale

Le groupe G.O a atteint ses objectifs. Le groupe reste malgré les quelques différents, bien soudés. La communication au sein du groupe permet à l'ensemble de l'équipe d'avancer dans la même et bonne direction. De plus, l'avance qu'ont pris certains dans leurs secteurs respectifs bonifie l'ambiance positive et la cohésion au sein du groupe de travail.

7.1 Planning originel

Tâches/Soutenances	1er	2ème	3ème
Moteur de rendu			
Algorithme de Raycasting	25%	50%	100%
Moteur Physique			
Moteur Physique	20%	60%	100%
Moteur Audio			
Moteur Audio	50%	75%	100%
IA			
IA des NPC	25%	50%	100%
Autre			
Système de cartes / niveaux	50%	100%	100%
Interfaces	25%	50%	100%
Logiciel d'édition de cartes	0%	50%	100%

7.2 Avancement actuel

	Prévisions	Réalisé
Moteur de rendu	50%	75%
Moteur Physique	60%	70%
Moteur Audio	75%	75%
IA	50%	80%
Cartes	100%	100%
Interfaces	50%	50%

8 Annexes

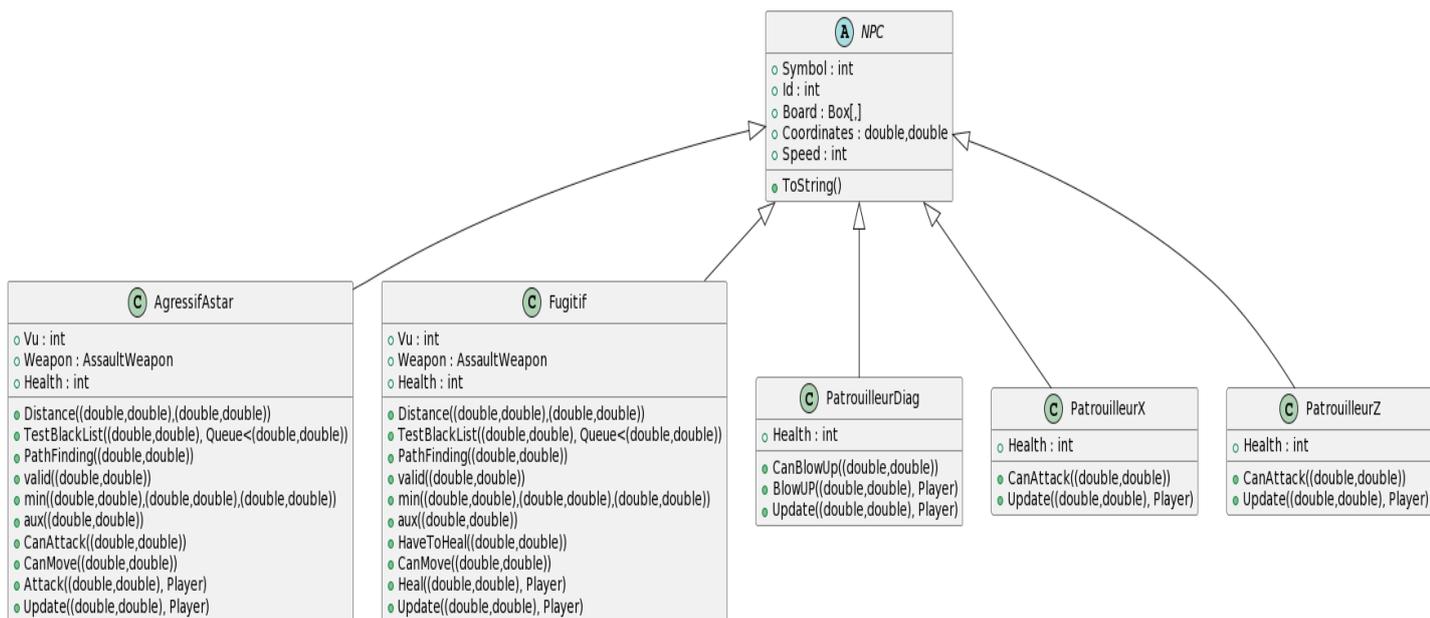


Diagramme de la classe NPC

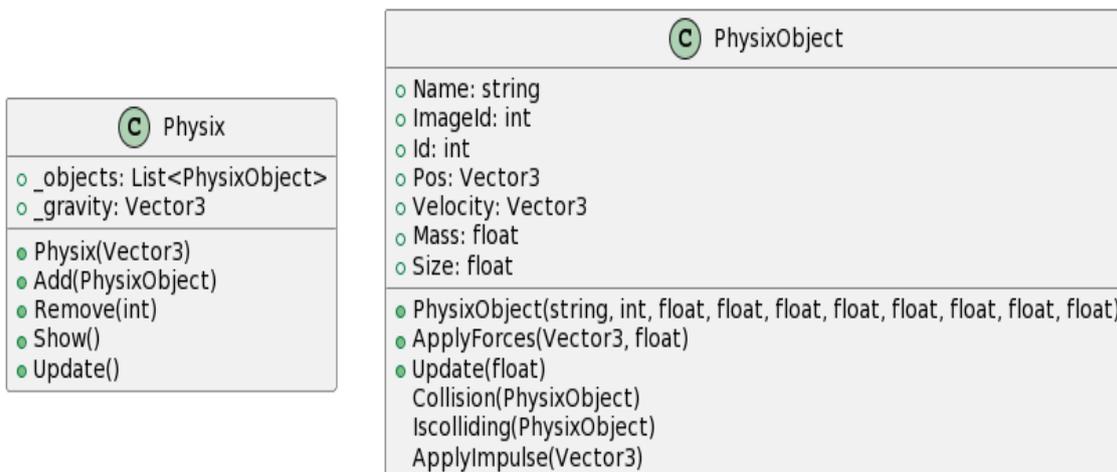


Diagramme de classe du moteur physique



SFML, www.sfml-dev.org.

