

</> LES TROPHÉES NSI

Édition 2023

DOSSIER DE CANDIDATURE
PRÉSENTATION DU PROJET



NOM DU PROJET :

PyGario

> PRÉSENTATION GÉNÉRALE :

- L'idée du projet est de créer une version du fameux jeu Agar.io en 3D. Il s'agit d'un jeu multijoueur où le joueur contrôle un cercle représentant une bactérie. Le but est de grossir en mangeant les joueurs plus petits. Notre version du jeu en 3D ne met pas en scène des bactéries mais des planètes qui, pour grandir, devront manger leurs congénères. Notre but ultime serait de réussir à implémenter le multijoueur et une version en ligne.

- L'origine du projet nous vient de notre camarade Nicolas qui, en nous faisant découvrir le jeu Agar.io, nous a proposé de le refaire en 3D. L'intérêt de ce projet est de créer, en groupe, un jeu en 3D codé en Python de A à Z. Cela permet également de voir les capacités du langage Python dans le domaine du jeu vidéo. Pour finir, nous avons utilisé la module Python Ursina, qui offre des interfaces utilisateur modernes ainsi que la possibilité d'écrire un code concis et performant.

> ORGANISATION DU TRAVAIL :

• Présentation de l'équipe :

Notre équipe est composée des 5 membres suivants :

Paul Ravoux : création des IA et des bots ;

Nicolas Nunney : développeur du menu ;

Emma Peyrard : développeuse des décors et des interactions entre planètes ;

Lucas Devanlay : développeur background, textures, pseudos ;

Dimitri Maréchal : développeur de la partie joueur et chargé de l'optimisation.

• Répartition des tâches :

Paul Ravoux s'est occupé de la création des IA des robots lorsqu'on joue en solo.

Nicolas Nunney s'est occupé du menu du jeu ainsi que des options, notamment de personnalisation.

Lucas Devanlay s'est chargé des recherches de texture pour les planètes et de leur mise en place dans le jeu, de l'éclairage et de l'affichage dans le jeu.

Emma Peyrard a mis en place le décor, le déroulement de la fin de partie ainsi que la création et les interactions des planètes.

Dimitri Maréchal a codé entièrement la classe Joueur et s'est chargé de l'optimisation de la détection de collisions et de la recherche du plus petit proche voisin.

Enfin, ils ont tous les cinq contribué à la détection des bugs dans les différentes classes et fonctions, à la création de la documentation et de la présentation écrite et vidéo, à la bonne humeur et à l'entente du groupe.

• Organisation du travail :

Le travail fut organisé de cette manière :

Chacun travaillait de son côté hors des cours, mais restait joignable sur le groupe discord créé pour l'occasion. De plus grâce à la plateforme GitHub nous pouvions voir en temps réel les modifications apportées par tous les membres du groupe. Puis, à raison d'une heure par semaine, lors des cours de NSI, nous faisons une réunion afin de déterminer les priorités pour les prochains jours et continuons notre travail ensemble.

LES ÉTAPES DU PROJET :

• Présenter les différentes étapes du projet :

- 1- Nous avons eu l'idée de refaire un jeu en 3D et nous avons choisi Agar.io.
- 2- Nous avons cherché un moyen de faire de la 3D sur Python et découvert Ursina.
- 3- Nous avons pris connaissance du module Ursina dans son intégralité, c'est-à-dire des différentes classes et fonctions qu'il regroupe.
- 4- Nous avons ensuite créé plusieurs fichiers Python afin de séparer les classes ou les constantes générales dont nous aurions besoin par la suite.
- 5- Puis nous avons réparti les tâches entre les 5 membres.
- 6- Nous avons initialisé un repository GitHub pour que tout le monde puisse accéder à tous les fichiers.
- 7- Puis chacun a codé la partie du programme qui lui avait été assignée.
- 8- Chacun devait par ailleurs prendre connaissance des nouveautés sur GitHub pour éviter certaines erreurs.
- 9- Enfin, nous avons réglé tous les petits ou grands bugs ensemble.
- 10- Nous avons invité certains de nos professeurs et amis à tester le jeu.

> FONCTIONNEMENT ET OPÉRATIONNALITÉ :

• Avancement du projet :

Les bases du projet sont au point, le jeu tourne sans bug. La quasi-totalité du jeu est finie, même si, comme vous le verrez ensuite, nous avons de nombreuses idées d'améliorations possibles. La seule chose qui reste encore à parfaire est le menu du jeu qui n'est pas optimisé au maximum.

• Approches mises en œuvre pour vérifier l'absence de bugs et s'assurer de la facilité d'utilisation du projet :

Pour s'assurer d'un jeu simple d'utilisation et pour dénicher les éventuels bugs visibles lors d'une partie, nous avons fait tester notre jeu à plusieurs camarades et professeurs de notre établissement. Puis pour les erreurs et bugs de code, nous implémentions des séries de tests, puis nous testions rigoureusement les fonctions implémentées dans le jeu. Si une erreur survenait, alors nous cherchions où était le problème avec des affichages, en relisant les différentes documentations et en apportant des modifications.

• Difficultés rencontrées et solutions apportées :

Les difficultés principales ont été l'optimisation du jeu et le manque de clarté sur la documentation sur Ursina. Pour y remédier nous avons relu plusieurs fois cette documentation, regardé des ressources sur Internet et parfois même le code source d'Ursina. De plus, après la découverte tardive de ce concours qui s'ajoutait au BAC et aux concours pour les grandes écoles, nous avons dû prendre énormément sur notre temps libre et organiser les cours de NSI d'une autre façon pour finir le programme.

> OUVERTURE :

• Idées d'améliorations (nouvelles fonctionnalités) :

Les prochaines mises à jour et les grandes améliorations seront : une version en ligne, un menu plus propre avec plus de personnalisation, des mouvements plus intuitifs, un meilleur rendu visuel, une meilleure optimisation et de meilleures performances, des bots beaucoup plus intelligents...

• Stratégie de diffusion pour toucher un large public :

Premièrement, nous le mettrons en ligne sur Google etc. Et nous comptons en faire la publicité sur les chaînes YouTube suivantes :

Flymeth: <https://youtube.com/flymeth> ;

Pitichicken : <https://youtube.com/@pitichicken3725> ;

Iceknith : <https://www.youtube.com/channel/UCL0TP0A9yj3cU115mrKPF8Q>

Ce sont les chaînes respectives de Johan Janin (un camarade de classe en dehors du projet), Paul Ravoux et de Dimitri Maréchal.

• Analyse critique du résultat

Si nous devons refaire le jeu nous changerions catégoriquement notre organisation pour éviter que certains travaillent plus que d'autres sur certains points et mieux travailler en groupe. De plus, si nous devons le refaire ce ne serait absolument pas codé en Python car ce langage n'est pas assez performant mais plutôt en C++ ou C#.

DOCUMENTATION

• Spécifications fonctionnelles

• Spécifications techniques

• Illustrations, captures d'écran, etc

Pour la documentation, il faut aller dans le dossier *doc* du dossier technique au lien suivant : https://drive.google.com/drive/folders/1HhdgiJ6lftRULw_WwHou7Tsc52NrdDOi?usp=share_link