



LES  
TROPHÉES NSI

Édition 2023

## DOSSIER DE CANDIDATURE PRÉSENTATION DU PROJET



Ce document est l'un des livrables à fournir lors du dépôt de votre projet : 4 pages maximum (hors documentation).

Pour accéder à la liste complète des éléments à fournir, consultez la page [Préparer votre participation](#).

Vous avez des questions sur le concours ? Vous souhaitez des informations complémentaires pour déposer un projet ? Contactez-nous à [info@trophees-nsi.fr](mailto:info@trophees-nsi.fr).

**NOM DU PROJET : Lapins Pas Crétins**

## > PRÉSENTATION GÉNÉRALE :

- *Idee et objectifs*
- *Origines et intérêts du projet*
- (...)

Le projet est une simulation d'écosystème à but éducatif sous la forme d'un site web. Il utilise le HTML, le CSS, le Javascript et la librairie Javascript d3 JS pour afficher des graphiques. Le projet consiste en une simulation paramétrable puis une partie explicative des phénomènes suivants que l'on pourra observer : la sélection naturelle, la mutation, la loi proie prédatrice, le pathfinding et la génération procédurale. Le projet a été fait à l'aide de github et de visual studio code principalement. Il s'adresse à des jeunes niveaux collège pour leur apprendre visuellement ces concepts de biologie et d'informatique. La simulation est une grille représentant un environnement généré procéduralement, des lapins se déplacent vers un terrier grâce au pathfinding, plus la couleur des lapins est proche de l'environnement moins ils seront mangés, donc de génération en génération les lapins se camouflent de mieux en mieux, c'est la sélection naturelle. On peut observer cela grâce à deux graphiques de d3 js, un sur la couleur moyenne des lapins et l'autre sur l'évolution des populations de lapins et d'aigles en temps réel. Le projet démontre aussi pourquoi il faut prendre soin de l'environnement, en effet si il y a un changement trop brusque sur l'environnement, les lapins seront mal camouflés et risquent de ne pas avoir le temps de s'adapter et donc l'espèce va s'éteindre. Le site possède un design moderne et attrayant pour les jeunes. Il peut aussi servir à des professeurs pour expliquer visuellement ces phénomènes. Le projet a été fait avec une équipe de 3 personnes, un codeur, un designer, un mathématicien mais tout le monde a touché aux différents aspects du projet. Celui-ci pourrait posséder plus de paramètres pour permettre de démontrer d'autres phénomènes, aussi pouvoir télécharger les résultats d'une simulation serait intéressant.

## > ORGANISATION DU TRAVAIL :

- *Présentation de l'équipe (prénom de chaque membre et rôle dans le projet)*
  - *Répartition des tâches*
  - *Organisation du travail (répartition par petits groupes, fréquence de réunions, travail en dehors de l'établissement scolaire, outils/logiciels utilisés pour la communication et le partage du code, etc.)*
- Thomas Chatelain : responsable web
  - Adam Senouci : Recherches mathématiques, développement des graphiques
  - Simon Josse : Développeur

Les tâches ont été réparties en fonction des affinités. Les rôles sont plus ceux de responsables que d'exécutants, puisqu'en réalité tout le monde a fait de tous les domaines du projet.

Pour la répartition du travail, les cours de NSI nous ont permis d'avoir du temps ensemble. Mais en dehors des cours nous avons utilisé des outils comme github, ou liveshare sur visual studio code pour travailler à distance, souvent en duo pour effectuer des tâches. Chaque week-end une réunion permettait de donner les objectifs pour la semaine d'après et de savoir les problèmes rencontrés. Très souvent, des problèmes ont persisté pendant plusieurs semaines et donc des réunions spécifiques aux problèmes ont aussi eu lieu, par exemple sur la manière de représenter les prédateurs qui a fait débat.

## LES ÉTAPES DU PROJET :

- *Présenter les différentes étapes du projet (de l'idée jusqu'à la finalisation du projet)*

**1. Idéation** : Nous avons commencé à discuter des idées de projets pour notre participation à un concours de projets éducatifs pour les jeunes de 15 ans. Nous avons décidé de créer une simulation d'écosystème à but éducatif sous forme de site web, qui utilise des graphiques interactifs pour enseigner visuellement les concepts de biologie et d'informatique.

**2. Planification** : Nous avons commencé à planifier les différentes étapes du projet et à attribuer des tâches à chaque membre de l'équipe. Nous avons également décidé des technologies à utiliser, telles que HTML, CSS, JavaScript et la bibliothèque D3.js pour les graphiques.

**3. Conception et développement initial** : Nous avons commencé à travailler sur le design du site web et la création de la simulation d'écosystème. Nous avons créé une maquette de la page d'accueil et de la simulation, ainsi que des graphiques de D3.js pour représenter l'évolution des populations de lapins et d'aigles.

**4. Tests et corrections** : Nous avons effectué des tests approfondis de la simulation et du site web pour détecter les éventuelles erreurs ou bugs. Nous avons également corrigé les problèmes identifiés et optimisé les performances de la simulation.

**5. Finalisation et présentation** : Nous avons finalisé le projet et préparé une présentation pour le concours.

**6. Réflexion et améliorations futures** : Nous avons réfléchi aux points forts et aux points faibles de notre projet et discuté des améliorations futures possibles, telles que l'ajout de paramètres pour démontrer d'autres phénomènes et la possibilité de télécharger les résultats de la simulation. Nous avons également envisagé d'autres utilisations possibles de la simulation en dehors du cadre éducatif.

## > FONCTIONNEMENT ET OPÉRATIONNALITÉ :

- *Avancement du projet (ce qui est terminé, en cours de réalisation, reste à faire)*
- *Approches mises en œuvre pour vérifier l'absence de bugs et s'assurer de la facilité d'utilisation du projet*
- *Difficultés rencontrées et solutions apportées*

### Avancement du projet :

- La simulation paramétrable de l'écosystème est terminée
- La partie explicative des phénomènes observables est en cours de finalisation
- Le design du site web est terminé
- Les graphiques avec la librairie D3 JS sont terminés
- Les paramètres de la simulation peuvent être modifiés facilement grâce à une interface utilisateur intuitive

### Approches pour vérifier l'absence de bugs et assurer la facilité d'utilisation :

- Les membres de l'équipe ont testé la simulation et signalé les éventuels bugs
- Des tests unitaires ont été réalisés sur le code
- Des utilisateurs externes ont testé le site web et ont donné des retours pour améliorer l'expérience utilisateur

### Difficultés rencontrées et solutions apportées :

La génération procédurale de l'environnement a été un défi pour l'équipe, mais elle a pu être réalisée grâce à une collaboration étroite entre le codeur et le mathématicien.

L'optimisation de la simulation pour un temps de chargement rapide a également été un défi, mais en utilisant les bonnes techniques d'optimisation, l'équipe a pu améliorer considérablement le temps de chargement.

La coordination entre les membres de l'équipe a été un défi, mais en mettant en place une communication régulière et une organisation efficace, l'équipe a pu surmonter cette difficulté.

#### Reste à faire :

- Finaliser la partie explicative des phénomènes observables
- Ajouter plus de paramètres pour permettre la démonstration d'autres phénomènes
- Ajouter la possibilité de télécharger les résultats de la simulation

### > OUVERTURE :

- *Idées d'améliorations (nouvelles fonctionnalités)*
- *Stratégie de diffusion pour toucher un large public (faites preuve d'originalité !)*
- *Analyse critique du résultat (si c'était à refaire, que changeriez-vous dans votre organisation, les fonctionnalités du projet et les choix techniques ?)*

Pour améliorer le projet on pourrait adapter l'interface aux téléphones, cela permettrait de le rendre encore plus accessible. De plus la possibilité de récupérer les graphiques sous forme d'image ou simplement les données de la simulation en texte seraient intéressants. Rajouter des paramètres serait aussi une manière de potentiellement expliquer de nouveaux phénomènes à partir de cette simulation .

Pour propager ce projet, le choix d'utiliser uniquement des technologies web le rend très facilement accessible sur internet, il suffit du lien ! De plus les parties d'explications sur le pathfinding et la génération procédurale intéresseront beaucoup les amateurs et créateurs de jeux vidéos, ce qui concerne une grande partie des jeunes actuellement, les parties sur la mutation et la sélection naturelle intéresseront les passionnés de nature et d'animaux, cela permet donc de cibler une très grande partie de la population, avec des profils très variés, garçons comme filles, jeunes et plus âgés. L'interaction entre les lapins et les aigles est totalement non violente et l'apparence mignonne des lapins peut séduire les jeunes enfants.

Le projet a selon nous le gros défaut de ne pas posséder une simulation très esthétique. Si nous pouvions le refaire, nous ferions la simulation en 3D avec la librairie three Js. De plus l'implémentation d'un tutoriel serait intéressante, pour l'organisation, documenter le code pendant la création aurait été d'une grande aide. Pour finir, le choix de faire la simulation sur une grille nous a permis d'implémenter le pathfinding et la génération procédurale mais nous avons aussi grandement limité sur la possibilité d'animer la simulation, nous aurions dû penser aux enjeux de ce choix technique plus tôt.

## DOCUMENTATION

- *Spécifications fonctionnelles (guide d'utilisation, déroulé des étapes d'exécution, description des fonctionnalités et des paramètres)*
- *Spécifications techniques (architecture, langages et bibliothèques utilisés, matériel, choix techniques, format de stockage des données, etc)*
- *Illustrations, captures d'écran, etc*

Pour utiliser le projet, il suffit de lancer le fichier index.html présent au chemin suivant : sources/front/index.html

#### Requiert :

Il faut être connecté à Internet pour que la librairie D3JS soit bien importée.

Le navigateur doit être récent (<2016) et ne pas être Internet Explorer.

L'écran doit être plus large que 1360 pixels.

Pour exécuter, l'utilisateur appuie sur le bouton JOUER, il n'y a pas besoin de modifier les paramètres présents sur la page. Il peut maintenant regarder la simulation et les graphiques en dessous se mettre à jours en temps réel, le paramètre vitesse permet d'obtenir des résultats intéressants plus vite. A la fin de la simulation l'utilisateur pourra regarder les différentes explications en dessous. S'il le souhaite, il peut relancer d'autres simulations paramétrées différemment pour revoir ou mieux observer certains phénomènes. Les possibilités sont le bouton SCORE CASE permettant de comprendre le pathfinding, le bouton « Appliquer un choc sur l'environnement » ainsi que le slider de mutation pour l'explication de la mutation. L'input « nombre de génération » pour avoir une simulation plus ou moins longue, pour celui ci il faut rentrer la valeur dans le champs associé puis appuyer sur « mettre a jours ». Tous les paramètres sont modifiables pendant l'exécution d'une simulation. Le bouton « RENTIALISER » permet d'initialiser une nouvelle simulation et de supprimer celle en cours d'exécution.

Nous avons utilisé la programmation orientée objet pour créer cette simulation avec le langage javascript, ainsi que le canva de HTML5 pour faire l'affichage, c'est un moyen d'afficher sans librairies externes et sans utiliser de moteurs graphiques trop complexes comme WebGL, bien que cela ait été envisagé.

Toute la logique est dans le dossier « scripts ». Ici « main.js » permet de gérer les interactions avec l'interface. Il importe aussi game.js, qui contient la classe Game qui est l'objet contenant toutes les méthodes et propriétés nécessaires au fonctionnement de la simulation. Cet objet importe lui même d'autres classes comme Map, Prey, Preda qui sont toutes dans le dossier « modules ». Cette architecture modulaire permet d'isoler les différents éléments et de travailler ensemble plus efficacement. Les fonctions utilitaires que l'on a utilisées plusieurs fois sont dans tools.js. Ce fichier contient la fonction colorSim, celle-ci permet de calculer le score de camouflage des lapins par rapport à leur environnement, la technique de comparaison des codes rgb a été pensée par [Vadim Gulyakin](#). Elle consiste à comparer la distance entre les 3 valeurs du code rgb, cet algorithme ne correspond pas exactement à la perception humaine des couleurs, mais est suffisant dans notre cas pour créer une sélection naturelle. Une solution plus précise aurait été d'utiliser une autre représentation de la couleur, comme le HSL, ou le HSV. Dans tout le programme, nous avons profité du très bon fonctionnement de JavaScript avec les objets pour stocker la plupart de nos données. Le plus complexe a été l'environnement, il fallait créer une structure qui contiendrait beaucoup d'éléments (chaque case) et que celle-ci soit très rapidement accessible, sans utiliser de recherche linéaire, ce qui ralentirait trop le programme. La carte de l'environnement est donc une liste, contenant une liste par ligne contenant toutes les colonnes, dans celle-ci se trouve un objet correspondant à la case et contenant ses données. On peut donc accéder à une case XY avec le code suivant `caseRecherche = map[X][Y]`. Cette idée s'inspire du fonctionnement des hashmaps. Aucun code extérieur n'a été utilisé mis à part pour colorSim(). La librairie D3js a été utilisée pour générer les graphiques, elle était très adaptée puisqu'elle permet de rendre les graphiques dynamiques et qu'elle est très légère.

De plus github et vsCode liveshare ont permis respectivement de garder les différentes versions du projet et d'héberger le site, et de coder à plusieurs en même temps sur un même fichier.