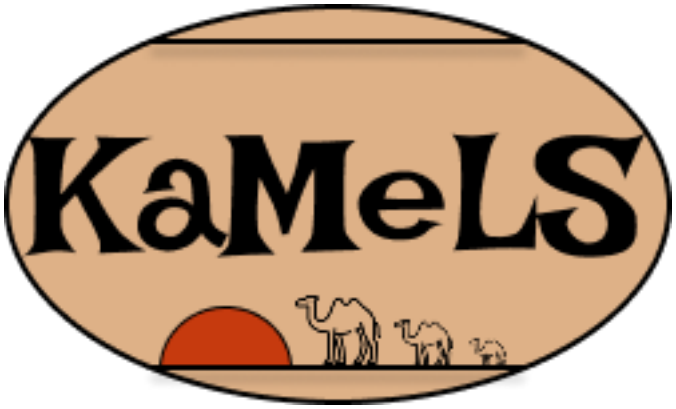
Rapport de Première Soutenance - Projet S4

Hayvolution

Groupe : KaMeLS



Kylian Djermoune Maxime Trimboli Lino Joninon Simon Campredon

Table des matières

1 Introduction

1.1 Rappel du projet

Hayvolution est un projet de simulation d'évolution d'écosystème qui possède ses propres règles, un jeu de la vie beaucoup plus poussé. Au sein de cet environnement représenté en caractères ASCII, différentes espèces animales évoluent entre elles. L'objectif est de réaliser une simulation que l'utilisateur pourrait entièrement personnaliser ; à savoir l'environnement et les animaux, pour voir comment ses choix influencent le déroulé de la simulation. Les animaux ayant tous des caractéristiques différentes, pouvoir observer leurs évolutions au fil des générations et essayer de les prédire aura un intérêt certain pour l'utilisateur. L'objectif de ce projet sera aussi de modéliser de basiquement mais fidèlement l'évolution ainsi que ses mécanismes, et ce peu importe les choix de départ de l'utilisateur, il s'agit donc de créer un système cohérent et équilibré. Dans ce projet se mélangeront des notions de génération aléatoire, que ce soit pour générer un environnement au début de la simulation mais aussi pour provoquer des mutations au sein des espèces d'une génération à l'autre. Un des objectifs du projet sera de simuler l'intelligence des animaux, pour cela nous utiliserons plusieurs algorithmes couplés, idéalement nous aimerions que les espèces s'adapte aussi de par leurs comportements, et que donc ces adaptations soient transmissibles aux générations suivantes. Enfin, nous souhaitons à terme avoir des interfaces graphiques agréables pour voir se dérouler la simulation, ainsi qu'une interface dédiée à des statistiques sur la simulation pour observer globalement l'évolution des

espèces génération par génération.

1.2 Objet d'étude

1.2.1 Aspect technique

Pour ce projet nous devons aborder des problématiques de codage qui nous permettent de mettre à l'épreuve nos connaissances mais également de rester réalistes dans nos capacités actuelles. Nous avons donc décidé d'approfondir ce que nous avons appris au semestre précédent et de s'essayer à de nouveaux aspects de la programmation. Pour l'interface nous réutiliserons ce que nous avons appris avec GTK ainsi que SDL. Cependant notre vision finale du projet nous demandera peut-être de s'essayer à d'autres moyens d'affichage. L'environnement affiché, lui, sera lu à partir d'une matrice. Nous allons pouvoir utiliser la manipulation de matrice que nous connaissons. Par la suite, il nous faudra également afficher les animaux, leurs interactions et l'environnement, ceci se fera à travers des structs et du code plus classique mais non moins important au projet. Enfin, le comportement des animaux doit-être défini. Dans un premier temps il sera fait à partir d'un algorithme intelligent mais ensuite nous souhaitons le remplacer par un réseau de neurones. Cette partie du projet est certes ambitieuse mais nous sommes confiants que l'expérience accumulée grâce au projet OCR nous sera d'une grande aide. Le site internet est une bonne occasion pour parfaire et maintenir nos compétences en HTML. Finalement, ce projet à la différence du projet OCR ne contient pas des parties clairement découpées mais un ensemble de tâches que nous nous sommes fixées. Il sera donc impératif de pouvoir communiquer

clairement et efficacement afin d'éviter les confusions ou les malentendus tout au long du projet. Ce point est encore plus crucial car les différentes parties, telles que celles des animaux et celle de l'environnement se chevauchent. L'organisation de notre projet est donc bien plus importante que pour les projets précédent, ceci est dû au fait que les différentes soutenances ne contiennent plus des objectifs prédéfinis mais des objectifs que nous nous sommes donnés nous-mêmes. L'esprit d'équipe est donc un aspect sur lequel il nous faudra travailler davantage.

1.2.2 Intérêt culturel

Cette simulation permettra de récréer, de manière bien plus simpliste, des phénomènes qui ont été constatés dans la vie réelle. L'évolution des prédateurs au fil des générations grâce à la sélection naturelle pour filtrer ceux dont les caractéristiques facilitent la chasse d'autres animaux. Et il en va de même pour les herbivores qui s'adapteront au fur et à mesure pour être capable de survivre à ces menaces. Les différents environnements présents pourront également, selon le déroulé des simulations, mener à la spéciation des espèces (divers groupes d'animaux possédant un ancêtre commun mais ayant des traits et caractéristiques bien différentes) ou encore observer une convergence évolutive de deux espèces non semblables originellement.

1.3 Présentation du groupe

Simon Campredon

Pour le projet OCR de S3, je me suis occupé du réseau de neurones, cela

a été bien difficile mais j'ai tout de même de bons souvenirs car il m'a permis de me découvrir une nouvelle passion pour l'intelligence artificielle. C'est pourquoi ce projet est pour moi important car il me permettra possiblement d'élargir mes connaissances déjà acquises ou bien d'en découvrir de nouvelles.

Kylian Djermoune

Je suis impatient de travailler sur ce projet de S4, l'idée d'un simulateur d'écosystème m'ayant directement emballé. La partie dont je m'occupe, l'environnement et notamment sa génération procédurale m'apporteront des compétences nouvelles en programmation. Je me suis occupé de la partie de création de l'environnement selon une taille variable et du début de la génération aléatoire des éléments au sein de cet environnement.

Lino Joninon

Lors de projets que nous avons effectués précédemment, j'ai senti un manque d'organisation dans les groupes dont je faisais partie. Nous obtenions un résultat fini, mais qui aurait pu être fait plus facilement et plus rapidement avec plus de communication et d'organisation. Donc, en plus de développer mes compétences de codage, je souhaite approfondir et affiner mes capacités à travailler et à interagir au sein d'une équipe. Ma contribution pour le projet est l'interface et lier les différentes parties avec l'interface. Comparer au projet OCR une difficulté supplémentaire dans l'interface est l'affichage constant de l'environnement.

Maxime Trimboli

Nous allons de nouveau réaliser un projet libre, et après le projet S2 et l'expérience que nous avons acquise en programmation, j'ai hâte de voir ce que nous allons produire. De plus, un des enjeux de ce projet est de pouvoir reproduire des concepts évolutifs réels et de les modéliser, et j'ai hâte de revenir à mes cours de SVT du lycée pour voir ce que nous pourrons rajouter à la simulation.

2 Tâches effectuées

2.1 Environnement (Kylian)

Je me suis occupé de la partie relative à la génération de l'environnement. Pour la création de l'environnement, j'ai utilisé une struct Environment. Un Environment possède trois attributs ; une matrice de caractères dont la longueur et la largeur sont également des attributs de la structure.

Différentes fonctions utilisables sur les environnements sont disponibles.

— -newEnvironment : La fonction « newEnvironment » qui permet de créer un Environment, s'assurant d'allouer l'espace nécessaire à celui-ci puis d'initialiser ses attributs length et width selon les deux arguments passés. En ce qui concerne l'initialisation de son attribut mat, elle fait appel à la fonction « createMat ».

```
char **createMat(int length, int width)
{
        char **mat = (char **) calloc(length, sizeof(char *));
        for (int i = 0; i < length; i++)
        {
                mat[i] = (char *) calloc(width, sizeof(char));
        }
        return mat;
}


Environment *newEnvironment(size_t length, size_t width)
{
        size_t temp = width;
        if(width > length)
        {
                width = length;
                length = temp;
        }

        Environment *env = (Environment *) malloc(sizeof(Environment));
        env->length = length;
        env->width = width;
        env->mat = createMat(length, width);

        return env;
}
```

Figure 1 – Fonction newEnvironment

— -createMat : Cette fonction crée simplement la matrice de caractères en allouant dynamiquement l'espace nécessaire à celle-ci selon les deux arguments passés à la fonction.

— -printEnvironment : Actuellement, cette fonction permet simplement d'afficher le contenu de la matrice. Cependant, elle est temporaire et se verra modifier dans le futur puisque cette affichage se fera depuis l'interface mais que nous n'avons pas encore lié les différentes parties du projet.

— -freeEnvironment : Une simple fonction libérant l'espace alloué à l'environnement et la matrice de caractères lui étant associée.

— -fillEnvironment : Cette fonction permet de remplir la matrice de caractères de l'environnement passé en argument. Au terme du projet, elle se complexifiera pour permettre de générer les points d'eau, les arbres et les autres éléments de décors selon le système de biomes qui sera mis en place et selon des règles plus poussées.

```
void *fillEnvironment(Environment *env)
{
        srand(time(NULL));
        for(size_t i = 0; i < (env->length); i++)
        {
                for(size_t j = 0; j < (env->width); j++)
                {
                        if (rand() % 2 == 0)
                        {
                                env->mat[i][j] = 'T';
                        }
                        else
                        {
                                env->mat[i][j] = ' ';
                        }
                }
        }

        return NULL;
}
```

Figure 2 – Fonction fillEnvironment

2.2 Animaux (Simon Maxime)

Pour la partie sur les animaux, nous nous sommes réparti les tâches :

2.2.1 Maxime

J'ai créé la struct d'animaux ainsi que des outils tels que des constructeurs pour une espèce ou une fonction pour free les animaux. J'ai aussi réfléchi à comment implémenter une structure d'animaux en arbre : Cette

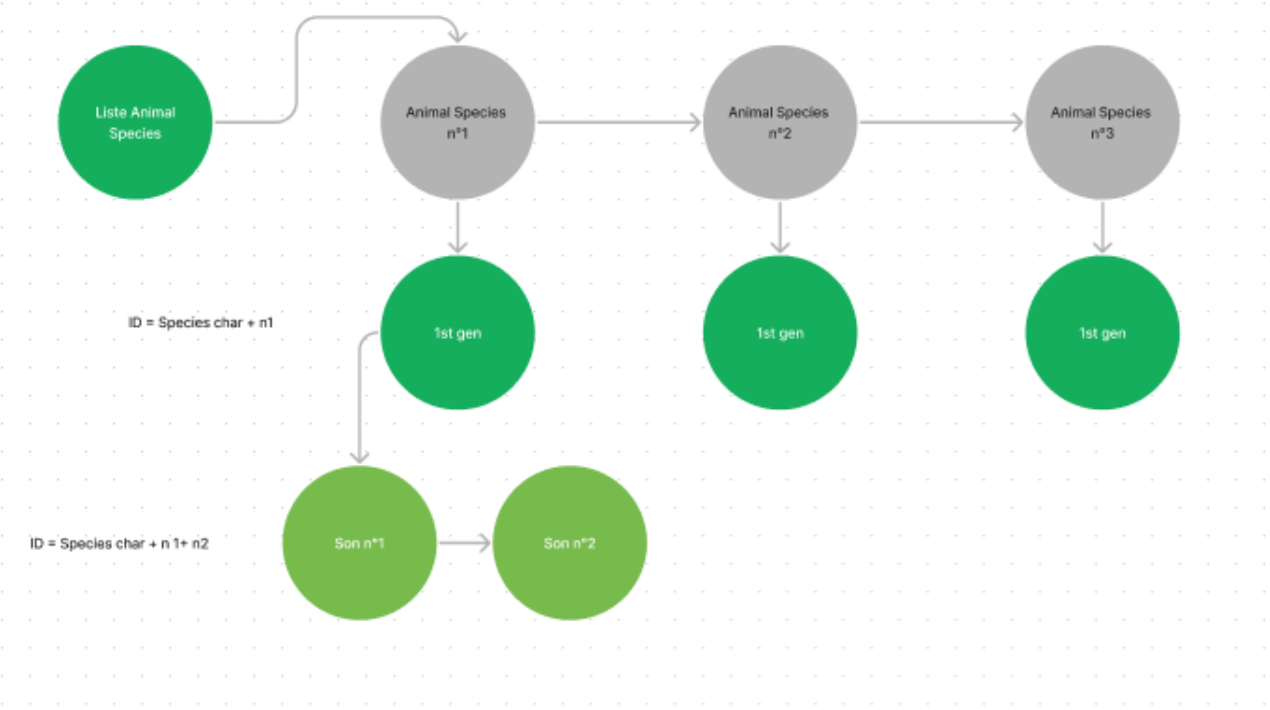


Figure 3 – Représentation en arbre des animaux

représentation a des avantages : elle permet d'avoir un représentant pour chaque espèce qui contient les statistiques de base d'une espèce, et permet d'appeler récursivement des fonctions sur la liste d'espèces. Cette implémentation permet aussi de représenter les liens de parenté entre les

animaux ce qui permet d'obtenir des informations cruciales pour après les représenter dans l'onglet statistiques. Cependant, cela a un désavantages, tout d'abord, si un animal meurt, on ne peut plus seulement l'enlever de la liste, il faut le garder en mémoire, et on peut lui donner un attribut booléen pour signifier qu'il est mort pour palier à ce problème.

2.2.2 Simon

Quant à moi, je me suis occupé des fonctions de logique et d'interaction entre les animaux, comme tout d'abord :

- getFoodPercent : Cette fonction va calculer le taux de nourriture de l'animal en fonction de la nourriture de base de son parent.
- kill : Cette fonction va « tuer » l'animal. De ce faite il sera simplement enlevé de la liste des animaux.
- reproduce : Cette fonction prend en paramètre un parent, et créer a partir de ces caractéristiques un enfant en modifiant légèrement les paramètres de perception, vitesse et force, puis l'ajoute a la liste des animaux.
- life-or-death : Cette fonction est la fonction principale de la gestion des condition de survie ou de reproduction. Le principe consiste à prendre chaque animal de la liste contenant tout les animaux, et en fonction de leurs espèces et de leur taux de nourriture, calculer grâce à la fonction « getFoodPercent », de décider si l'animal va ce reproduire, survivre, ou mourir.

```
void life_or_death(struct list *animal_list)
{
    struct list* p = animal_list;
    while(p)
    {
        struct animal *t = p->animal;
        switch(t->species)
        {
            case 'f':
                if (getFoodPercent(t) <= 10)
                    kill(animal_list, t->id);
                else if(getFoodPercent(t) >= 80)
                    reproduce(t, animal_list);
                break;
            case 's':
                if (getFoodPercent(t) <= 5)
                    kill(animal_list, t->id);
                else if(getFoodPercent(t) >= 70)
                    reproduce(t, animal_list);
                break;
            case 'q':
                if (getFoodPercent(t) <= 40)
                    kill(animal_list, t->id);
                else if(getFoodPercent(t) >= 50)
                    reproduce(t, animal_list);
                break;
            default:
                errx(1, "Unknown animal species\n");
        }
        p = p->next;
```

Figure 4 – Fonction de gestion reproduction/survie

Lors de la réalisation de ces fonctions, je me suis rendu compte que la gestion de l'espace de la liste des animaux allait être un problème. Avec l'aide de Maxime, nous avons implémenter la struct « list », qui est une structure de liste chaîné, réglant ainsi le problème de mémoire. Cette classe contient plusieurs méthode comme l'initialisation de la sentinelle, la recherche d'un élément, une fonction d'insertion et de suppression ainsi qu'une fonction de tri, qui nous sera utile plus tard dans le projet.

Pour la prochaine soutenance, en mettant ensemble nos parties, je pourrais commencer ma partie la plus importante, l'intelligence artificielle des animaux. Elle consistera a faire des calculs d'heuristie en jouant plusieurs tour d'affiler et choisissant la meilleur option possible. Tout ça cumuler avec quelques changement aléatoires, pour essayer de reproduire au mieux les interactions des animaux et de pouvoir suivre leurs évolution.

2.3 Interface (Lino)

L'interface que nous allons utiliser ressemble beaucoup à celui que l'on a fait pour le projet OCR mais en plus poussée et plus approfondie.

2.3.1 Bouton et fonctionnalité

La plus grande partie de l'écran est prit par une image qui affiche l'environnement dans lequel les animaux interagissent. Cette image devra être constamment mise à jour à chaque tour. Les animaux doivent se déplacer, disparaître ou se dupliquer au fil des tours. L'écran ou plutôt la simulation doit pouvoir s'arrêter lorsque l'on appuie sur les boutons Pause, Stop Simulation ou Play Step. Et pouvoir lancer ou relancer la simulation avec les boutons Start Simulation, Play Step et Play. Les boutons plus en détail sont les outils que l'utilisateur pourra utiliser afin de manipuler la temporalité de la simulation. Le bouton Start et Stop Simulation permet de lancer la simulation et donc la fonction main de notre projet. Le bouton Pause arrête la simulation à la fin de la génération actuel(soit environ 20 tours). Play relance une simulation qui a était arrêter mais ne démarre pas la simultion. Play Step lance la simultion pour une seule génération avant de se mettre en pause automatiquement. Statistiques doit plus tard afficher un écran qui affiche les statistiques des différents animaux, ainsi que des informations supplémentaires sur la simulation actuel. Le dernier bouton Quit permet de quitter le programme.

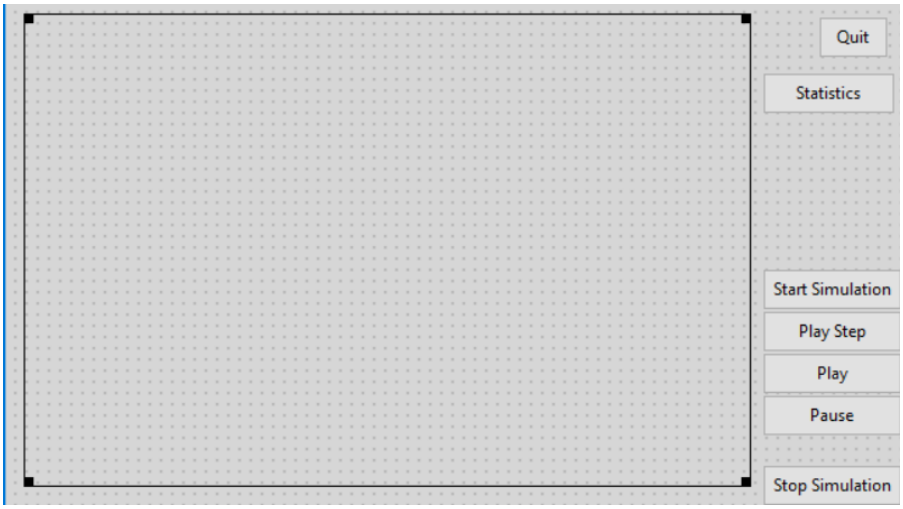


Figure 5 – état actuel de l'interface

2.3.2 Lecture de la matrice et bouton

La fonction de lecture de la matrice est plutôt simple à implémenter et le seul point important est de savoir ce que veulent dire les caractère que la fonction lit dans la matrice. Il est impératif que la matrice ou plutôt l'environnement obéit à des règles précises et inéchangeables. Cela est nécessaire pour la lisibilité et clarté de l'image afficher. Un deuxième travail est la représentation des symboles de la matrice avec des sprites. Il faut que le sprite de la montagne soit lisible qu'il soit une montagne au milieu du désert ou une chaîne de montagne. Cela relève d'un travail de graphiste plutôt que de programmeur. Ensuite les boutons devront tous être relier avec les fonctions correspondantes.

2.3.3 Problème rencontré et prévu

L'un des problèmes que l'on prévoit est la vitesse d'affichage qui variera en fonction du nombre d'animaux afficher. L'affichage et le code qui

gère les interactions des animaux risque d'être si rapide que l'utilisateur ne puisse plus voir le déplacements et le chemin que les animaux ont pris. Pour remédier à cela nous rajouterons un temps de latence pour laisser l'utilisateur profiter. Ce temps de latence pourra également être modifier plus tard pour accélérer les tours et les générations. Mais un autre problème est le cas inverse ou le nombre trop élevée d'animaux ralentit la simulation. Actuellement la seule solution est d'optimiser le travail le mieux possible.

2.4 Statistiques(Maxime)

Cette partie n'a pas encore été réellement programmée puisque nous n'avons pas encore de simulation fonctionelle. L'idée de cette partie est de pouvoir garantir un suivi global et dynamique des caractéristiques des animaux pendant la simulation, ainsi qu'un récapitulatif global à la fin. Les éléments importants à suivre sont évidemment le nombre d'animaux vivants dans la simulation et les valeurs moyennes de leurs caractéristiques (speed, sense, strenght) et leur food. Nous chercherons aussi à suivre l'évolution de la quantité de nourriture disponible dans l'environnement quand celui ci sera plus avancé. Enfin, si nous choisissons d'implémenter la structure d'animaux en arbre, nous pourront produire des statistiques sur l'évolution à partir d'un certain individu par exemple. Enfin, il sera important d'intégrer ces statistiques à l'interface, afin que l'utilisateur puisse visualiser clairement ces statistiques.

2.5 Site web (Maxime)

Le site web est déjà bien avancé dans sa structure. Nous pourrons faire des améliorations esthétiques mais le plus important sera de rajouter du contenu dedans. Il comporte une page de présentation du projet, une page pour télécharger le projet, le rapport de soutenance, ainsi qu'accéder à l'organisation GitHub de Kamels. Il y a aussi une page de présentation du groupe ainsi qu'une page de Documentation. La page de documentation permettra de donner toutes les spécifications pour utiliser le projet, ainsi que montrer des exemples de réglages pour obtenir des résultats intéressants. Documentation

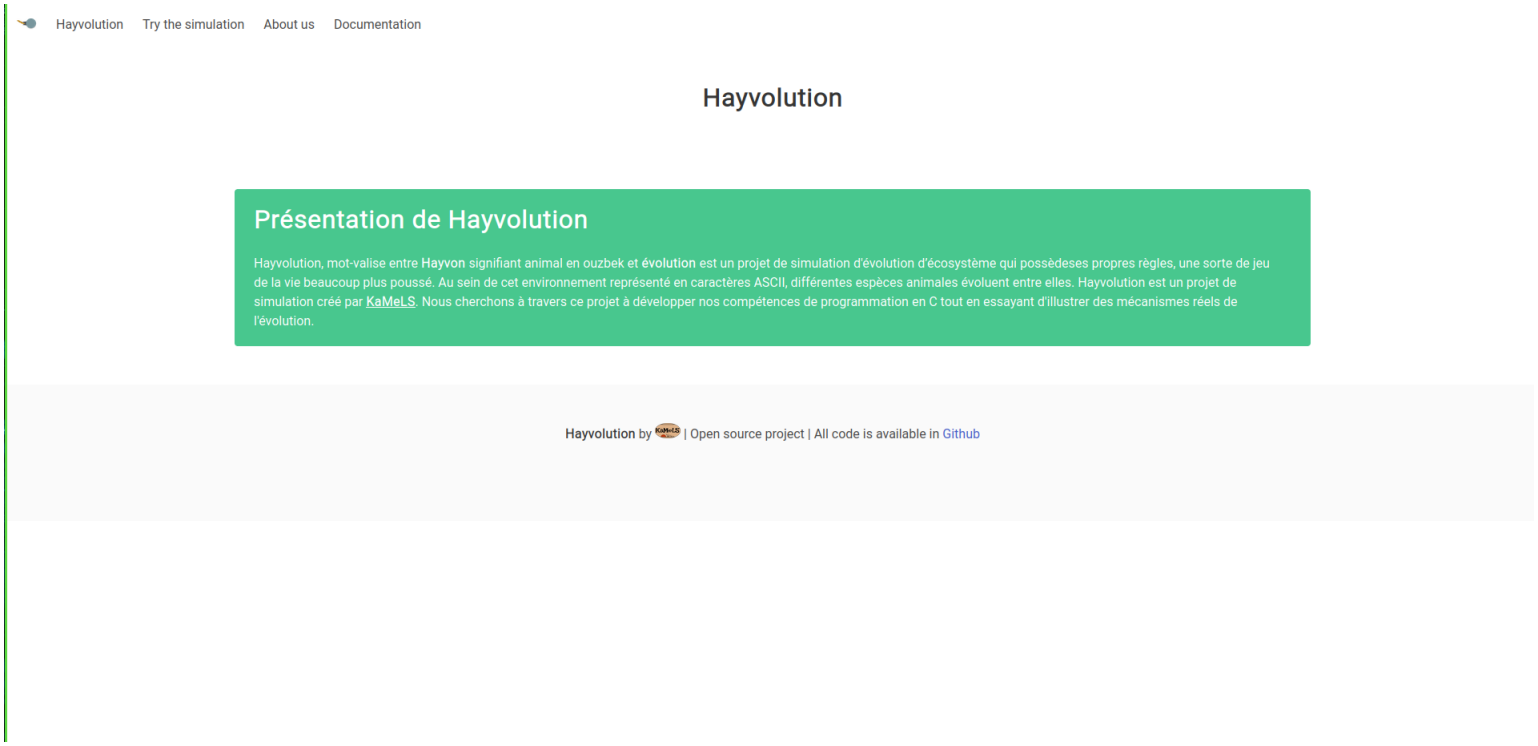


Figure 6 – la page de présentation du projet sur le site web

3 Conclusion

3.1 Répartition des tâches

Tâches	Simon	Kylian	Maxime	Lino
Interface				
Statistiques				
Environnement				
Création des animaux				
Caractéristiques des animaux				
Site Web				

3.2 Etat d'avancement du projet

Tâche - Date	prévisions 1re soutenance	Avancement réel
Interface	50%	40%
Statistiques	50%	0%
Animaux	50%	50%
Personnalisation de la simulation	0%	0%
Environnement	50%	30%
Interactions	75%	10%
Site web	50%	90%

3.3 Problèmes rencontrés

Nous n'avons malheureusement pas effectué l'ensemble des tâches que nous aurions voulu pour cette première soutenance. En effet, bien que

nous ayons avancé sur chacune des parties nous n'avons pas encore eu le temps de regrouper les différentes parties pour former une simulation fonctionelle.

3.4 Taches restantes à effectuer

Il nous faudra tout d'abord réunir les différentes parties du projet (environnement, animaux et interface) qui pour le moment sont toutes indépendantes.

Pour ce qui est de l'environnement il faudra implémenter le réel système de calcul de probabilités pour la génération de l'environnement. Cela, sans oublier l'implémentation du système le système de différents biomes qui chacun entraîneront des changements dans les probabilités d'apparitions des divers éléments (très peu d'arbres et d'eau dans un désert par exemple).

Quant aux animaux, il reste des fonctions à implémenter pour gérer leurs interactions, ainsi que les fonctions globales qui permettront de faire jouer tous les animaux.

Pour le site web, la plus part du travail est déjà faite, pour les prochaines soutenances nous chercherons à rajouter du contenu dessus.

Pour que l'interface soit entièrement fonctionnelle il manque l'affichage et la lecture de l'environnement. Le bouton doivent être lier au fonction correspondantes afin de lancer, arrêter, relancer et mettre en pause la simulation. Pour l'interface final que nous souhaitons obtenir, le bouton statistiques doit être fonctionnel, une option de zoom et dezoom sur l'image serait intéressant et apporter un peu de couleur à cet inter-

face monochrome. Si le temps nous le permet nous souhaitons rajouter l'option de créer ces propres animaux et cela requiert une toute nouvelle interface.

3.5 Conclusion

Bien que nous soyons en retard sur notre prévision initial, nous sommes confiant de pouvoir avancer rapidement et rattraper le retard. Le travail que nous avons effectuer pour cette première soutenance consiste de la base de ce projet et nous servira bien pour les soutenances suivantes. Notre équipe est motivé et nous avons l'envie de voir ce projet à son terme.