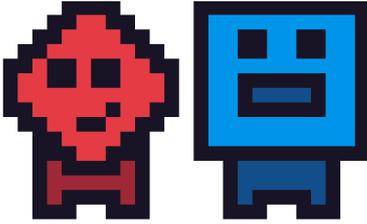


Le jeu de l'Humanité



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.

NOM DU PROJET : LE JEU DE L'HUMANITÉ

> PRÉSENTATION GÉNÉRALE :

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

.Idée:

Notre idée de départ vient de 2 sources d'inspirations :

-Le jeu de la vie (le nom de notre projet en est une référence) voir ce lien :

https://fr.wikipedia.org/wiki/Jeu_de_la_vie

-La simulation de sélection naturel du youtubeur « prime », consultable à ce lien : <https://youtu.be/0ZGbIKd0XrM>

Notre objectif est de simuler des choses complexes à partir de règles très simple.

Par exemple, dans « le jeu de la vie », des règles très simples permettent de créer de véritables systèmes complexes, un jeu de la vie à réussi à être recréer à l'intérieur du jeu de la vie.

C'est la même choses pour tous les êtres vivants :

Des molécules s'assemblent en suivant certaines règle primaire ce qui forme des molécules plus complexes jusqu'à obtenir des êtres vivants, système extrêmement complexe.

Nous avons donc opter pour un mix des 2 influences cités plus haut :

Nous avons utiliser des personnages à la place de carré noir et une pseudo sélection naturel avec des variables aléatoire pour chacun d'entre eux.

.Objectif:

Notre objectif est donc de simuler des systèmes complexes à partir de règles très simple.

Nous n'avons pas la prétention de dire que nous allons simuler la vie animal voir humaine, mais un très gros approchés dans un système simple avec peu de variable.

On veut observer une sélection des individus et une évolution de l'espèce qui s'adapte à son environnement.

> 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.)*

Romain Benslimane, chef du groupe, a eu l'idée de ce projet, il nous a encouragé et aidé même dans les parties de ses camarades car il possède des compétences plus élevées, il s'est donc naturellement dirigé vers les tâches plus ardues comme le déplacement, les statistiques etc. ..

Antoine Avelino-Ferrier, a permis de faire toutes les textures du projet car il a du talent en tant que graphiste, lui et Robin ont fait la page Web décrivant ce projet unique, il a aussi aidé sur le projet avec le menu et les fonctions utilitaires permettant de rendre le projet encore plus incroyable.

Robin Burckbuchler, quant à lui, a contribué à la page web avec Antoine, et pour le projet, il s'est occupé de la génération de la carte et des personnages, décor et animaux l'occupant. Et d'autres petites fonctions à droite à gauche.

Naturellement, l'entraide a été plus présente pendant cette longue période de travail et on ne peut compter le nombre d'heures passées à se parler ensemble sur discord pour parler de l'avancement et d'aide.

LES ÉTAPES DU PROJET :

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

Idée initiale a été atteinte et n'a pas changé.

Les étapes ont été les suivantes :

-Création pour tester d'une matrice affichée dans le terminal avec des points pour les cases et des 0 pour les personnages. Cela a permis de comprendre comment continuer pour la suite.

-Refonte du projet total et apprentissage de pygame. En effet, l'idée de matrice ne marchait pas avec pygame il a donc fallu tout refaire en fonction de cette dernière.

-Mise en place des graphismes, animations, collisions, variables, nourritures d'un seul bloc.

-Projet finit 1 jours avant la date limite avec de nombreux test permettant de déterminer quelles variables données pour résultat les meilleurs simulations.

> 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*

Fonctionnement :

Voici les lois de bases de notre simulation :

2 groupes de personnages : Les rouges et les bleus

1 groupes de nourritures : Les poulets

.Un personnage rouge attaque toujours un personnage bleu (ennemi) et inversement quand ils se rencontrent.

.Un personnage qui est a cotés d'un poulet essaye de manger le poulet

.Un personnage a cotés d'un personnage de sa même couleur se reproduit avec celui ci.

.Chaque personnage essaye de se déplacer à chaque tour

Chaque personnages à 4 variables aléatoires :

-La Vitesse (SPEED) ; Elle détermine la probabilité qu'un personnage se déplace d'une case à chaque tour.

-La Force (POWER) ; Elle détermine la probabilité que l'attaquant tue l'attaqué. Si l'attaque échoue, l'attaqué s'enfuit

-L'agilité (AGILITY) ; Elle détermine le pourcentage de chance qu'un personnage attrape un poulet. Si le personnage échoue, le poulet s'enfuit.

-La fertilité (FERTILITE) ; Elle détermine le pourcentage de chance qu'un personnage se reproduisent avec un autre personnage de la même couleur. En cas de reproduction, il est assuré d'avoir un enfant.

L'importance des variables est dans l'ordre donné, il attaquera en priorité un personnage puis mangera un poulet puis reproduira en cas de situation avec les conditions simultanée.

CHAQUE VARIABLE EST DÉTERMINÉ AU HASARD ENTRE 0 ET 100

Énergie :

Chaque personnages possèdent une énergie avant chaque début de jour de 100/100.

Les actions suivantes font perdre ou gagner de l'énergie :

- Être vivant (perd de l'énergie chaque tour)
- Manger un poulet
- Tuer un autre personnages
- Se reproduire
- Se déplacer

Noter que l'énergie gagné ou perdu après chaque action est déterminé à l'avance dans les paramètres.

Cycles et maison:

Un jour dure un nombre de tour définis à l'avance dans les paramètres.

À la fin du jour, chaque personnage rentre dans sa maison. La maison d'un personnage est créée sur la première case où le personnage est placé. La maison d'un personnage ne bouge jamais. Dans le cas où un personnage meurt, sa maison est détruite à la fin du jour.

Chaque personnage récupère son énergie la nuit en dormant chez lui. Dans le cas où 2 personnages se reproduisent, sa maison est placée à l'endroit où étaient placés le père ou la mère. Le personnage enfant apparaîtra dans sa maison le jour suivant.

Si le personnage n'attrape pas de poulet en rentrant chez lui, il meurt.

Au moment où l'énergie d'un personnage descend à inférieur ou égale 0, il meurt directement.

Avancement :

L'objectif de notre projet a été atteint et même plus.

Notre objectif quand à lui plus théorique n'a pas été atteint et est très loin d'être atteint car c'était de créer de véritables civilisations à partir de règles très simples en ajoutant des règles de

sociabilité, ainsi que la découverte du feu et l'invention des remparts, armes, bateau...

Des civilisations auraient donc été créées, avec des alliances des guerres, etc mais nous savions à l'avance que cela serait trop compliqué/trop long.

Nous n'avons pas vraiment d'idée d'amélioration si ce n'est rajouter plus de variables pour créer des systèmes plus complexes et réalistes. Par exemple la masse du personnage, des formules de physique réelle pour calculer l'énergie perdue pour chaque action.

Difficultés rencontrées :

Une difficulté majeure a été rencontrée :

-Le choix de l'architecture de et du fonctionnement initial de la carte avec les cases ;

Nous avons commencé notre projet avec une certaine architecture (matrice) mais cela c'est avéré impossible de continuer ainsi avec pygame. C'est donc en soi l'affichage de notre simulation qui a été un réel problème ;

2 autres difficultés mineures ont été rencontrées ::

-Nous aurions pu utiliser les classes pour grandement améliorer la performance et la facilité à créer le projet, mais cela est interdit pour participer au concours, nous avons donc utilisé une autre solution beaucoup plus complexe à mettre en place.

-L'affichage des animations : il a fallu calculer chaque image intermédiaire entre 2 positions ce qui a été très compliqué à faire.

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

-Pour vérifier qu'il n'y a pas de bug, des prints ont été ajoutés partout permettant d'avoir des logs de chaque action. Les bugs rencontrés étaient la plupart du temps des bugs graphiques (perso qui disparaît, se téléporte).

-La facilité d'utilisation du projet est assurée par un menu qui a été rajouté à la place de modifier à la main un fichier parametre.json.

Il n'y a rien d'autre à faire que choisir des paramètres et lancer la simulation donc le projet est très facile à utiliser.

> 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 ?)*

Nous n'avons pas vraiment d'idée d'amélioration si ce n'est rajouter plus de variables pour créer des systèmes plus complexe et réaliste. Par exemple la masse du personnage, des formules de physique réelle pour calculer l'énergie perdue pour chaque action.

Notre stratégie de diffusion est de faire des pubs en jouant sur le fait que c'est divertissant de voir des groupes se reproduire, évolués, mourir, s'entre-tué à très grande vitesse. C'est comme voir une vidéo en accélérés d'une endroit piétons avec les gens qui marchent, arrivent et partent à toutes vitesses. Donne un sentiment assez satisfaisant de voir tous cela évolué indépendamment d'une action humaine.

Très honnêtement, si c'était à refaire nous ne changerions pas grand choses si ce n'est l'architecture du projet car à cause d'une mauvaise architecture initiale, nous avons pratiquement dut refaire le projet en entier 2 fois.

Nous aurions aussi peut être commencés plus tôt pour avoir plus de temps et faire une simulation encore plus aboutit

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*

guide d'utilisation :

- Installer avec pip dans le cmd les bibliothèques se trouvant dans requirements.txt
- Lancer le fichier 'main.py'
- Un menu s'ouvre avec l'option 'quitter', 'jouer' et 'option'.

Liste des parametres :

"DAY": 50 :

Ce paramètre définit le nombre de tours qu'il y a dans 1 jours. Un nombre trop petit ne génère pas de simulation intéressante car les personnages n'ont pas le temps d'explorer la map.

"DURE_JEU": X :

Ce paramètre définit le nombre de jours avant que la simulation se finissent. Chacun qu'un jour égale à "DAY", ils faut donc multiplier $DURE_JEU * DAY$ pour avoir le nombre de tour totaux. Il est recommandé de ne pas mettre un nombre trop faible car cela ne donnerait pas des résultats très significatif.

"FPS": 10

Ce paramètre définit le nombre d'image entre chaque déplacement de case d'un personnage (FPS = 50 → 50 images entre 2 cases). Plus le FPS est grand, plus la simulation est fluide et « joli » à voir. Il est par défaut à 10 et il est recommandé de le baisser pour avoir une simulation qui tourne plus vite (moins de calcul).

"ZOOM": 13

Ce paramètre définit la taille de la map et donc le zoom. Il est différent pour chaque résolution d'écran car il est calculé à partir de celle-ci. Par exemple, un écran d'une grande qualité va pouvoir avoir plus de niveaux de zoom différents. Si le zoom est trop élevé pour la résolution de l'ordinateur, une erreur est déclarée et le programme cesse de fonctionner. (L'erreur a lieu dans la fonction "diviseur_commun" situé dans le fichier data_management. Attention ! Ne pas mettre le zoom à moins de 5 car vous ne verrez rien et votre ordinateur ne pourra supporter autant de calculs

"TIME": 5

Ce paramètre définit la vitesse de la simulation. Il faut diviser ce chiffre par 10 pour obtenir le temps entre chaque image affichée. Ne pas hésitez à le mettre à 0 pour faire une simulation très rapide

"NOMBRE NOURRITURE": 30

Ce paramètre définit le nombre de poulet qui seront générés.

"NOMBRE PERSO RED": 40

Ce paramètre définit le nombre de persos rouges qui seront générés.

"NOMBRE PERSO BLUE": 40,

Ce paramètre définit le nombre de persos bleus qui seront générés.

"CAILLOUX": 50

Ce paramètre définit le nombre de cailloux qui seront générés.

"ARBRE": 50

Ce paramètre définit le nombre d'arbres qui seront générés.

"ENERGY_MOVE": -2

"ENERGY_FIGHT": -3,

"ENERGY_CATCH": -2,

"ENERGY_REPRODUCTION": -4,

"ENERGY_FOOD": 5

Ne pas changer les paramètres suivant à moins que vous ayez une idée précise de comment ils marchent. Ils définissent le nombre d'énergie gagnés ou perdu en fonction de chaque action et ont été testés à l'avance pour générer la meilleure simulation possible

Spécifications techniques :

déroulé des étapes :

Le projet marche grâce à un dictionnaire commun pour tous les objets appelés « map ». Les clés de ce dictionnaire sont les différentes cases de la map (ex : clés (0,5) case $x = 0$ $y = 5$). Chaque clé est attribuée à un objet (herbe, perso, food, cailloux, ect...) Il ne peut donc y avoir qu'un objet par case car une clé est associée à un seul objet.

Tout au long de la partie, en suivant des calculs les valeurs des clés changent d'emplacement ou disparaissent.

Le langage de programmation est unique : python

Les bibliothèques utilisées sont pygame pour l'affichage et pyplot pour tracer des graphiques. Aucune donnée n'est véritablement stockée si ce n'est les paramètres.

Illustration :

Menu :

Le jeu de l'Humanité

Jouer

Options

Quitter

Lancement
simulation :



Attaque d'autre personnages et de poulets :

