

FOREST Giant Pits

PLAY

QUIT

NELLY DUPUYDENUS
LOONA GAILLARD
IMAC 2022
ANNÉE 2020

INTRODUCTION

Dans le cadre de nos études en école d'ingénieur IMAC (Image Multimédia Audiovisuel et Communication), nous avons été amenées à réaliser une version du célèbre jeu IMAC WARS.

Le but du projet est de mettre en pratique en application les compétences acquises en Programmation Algorithmique et Synthèse d'image, il s'agit donc de progresser dans 2 langages de programmation différents, le langage C et OpenGL.

Mais il s'agit aussi de nous entraîner à respecter un cahier des charge précis et également de nous apprendre à travailler en équipe sur un même projet.

Le projet s'est déroulé en plusieurs étapes.

Lire le sujet qui nous était proposé et s'assurer de comprendre tout ce qui était demandé.

La conception et la réalisation du jeu; quels ont été nos réflexions sur notre univers (partie Gamedesign), comment nous avons agencé notre code; plus précisément le développement du jeu. Quels résultats nous avons obtenus au fur et à mesure de l'avancement du projet.

L'organisation du projet afin de rendre un travail le plus qualitatif possible. De part la situation exceptionnelle du confinement nous avons dû revoir et apprendre à gérer un planning dans la précipitation.

Nous reviendrons également sur les difficultés que nous avons pu rencontrer lors de la réalisation de ce projet

PRESENTATION DU JEU

Retour au monde sauvage et aux animaux de la forêt.

Le Joueur incarne une troupe d'animaux sauvages unis pour défendre leur territoire de l'attaque d'une autre meute ou de l'humain venus conquérir un nouveau territoire et ainsi augmenter son pouvoir dans la région.

(Les 3 animaux parés au combat sont complémentaires ; le loup (vivace et fort), l'ours (fort et résistant) et le hibou (volant et vif).)

Alors paré au combat ?

CAHIER DES CHARGES

CE QU'ON A FAIT :

PARTIE COMMUNE :

- * Fonctionnalité du jeu
- * Architecture logiciel et propreté du code
- * Compilable par makefile

PARTIE SYNTHÈSE D'IMAGE :

- * Affichage de la carte
- * Affichage des unités
- * Gestion du tour par tour et IHM générale
- * Sélection et déplacement d'unité
- * Affichage des menus / Fin du jeu

PARTIE ALGORITHMIQUE :

- * Chargement de la carte (gestion des collisions)
- * Joueur artificiel

OPTIONNEL :

- * Case surprise
- * Spécificité de terrain
(le loup peut se cacher dans la forêt, l'ours dans la grotte et l'eau peut traverser l'eau et se poser sur les cailloux)

CE QUE L'ON N'A PAS FAIT :

- * Algorithme A-Star
- * Recherche de case avec arbre binaire



FONCTIONNEMENT DU JEU

DESCRIPTION DES FONCTIONNALITÉS

DÉMARRAGE DU JEU :

Pour démarrer le jeu le joueur doit avoir sa console ouverte. Il lui suffit ensuite de se rendre dans le bon dossier (exemple : **cd Bureau/ForestGuardians**). Une fois cela fait il devra compiler le jeu grâce à la commande «**make**». Pour lancer le jeu il faudra ensuite entrer «**./projet.out**». La fenêtre de jeu se sera alors lancée.

MENU DÉBUT :

Lorsque que le jeu se lance le joueur tombe alors sur ce menu d'introduction. Il a alors le choix de lancer la partie en cliquant sur «play» ou de quitter la partie en cliquant sur «quit».

S'il appuie sur **quit** la fenêtre se ferme.

S'il appuie sur **play** alors un nouvel affichage apparaît.

C'est celui du choix entre une partie 2 joueurs ou une partie contre l'ordinateur.



Affichage d'introduction



Affichage choix partie

INTRODUCTION AU JEU :

Après avoir lancé le jeu, la première unité du joueur lui est présentée. Il a alors le choix de jouer ce qui signifie qu'il démarre la partie avec une unité, ou d'acheter d'autres unités.

S'il décide d'acheter d'autres unités il tombe alors sur le menu achat du jeu.



Affichage menu achat

Chaque unité est présentée au joueur avec un petit descriptif et le prix qu'elle coûte. Encore une fois le joueur peut décider de faire marche arrière s'il ne veut rien acheter après réflexion.

Si la partie lancée est une partie 2 joueurs alors, il est demandé au deuxième joueur de choisir à son tour des unités supplémentaires, en respectant le même principe. Une fois que les deux joueurs ont fini de faire leurs achats, la map apparaît, contenant leur.s unité.s qui se positionnent aléatoirement sur la map.

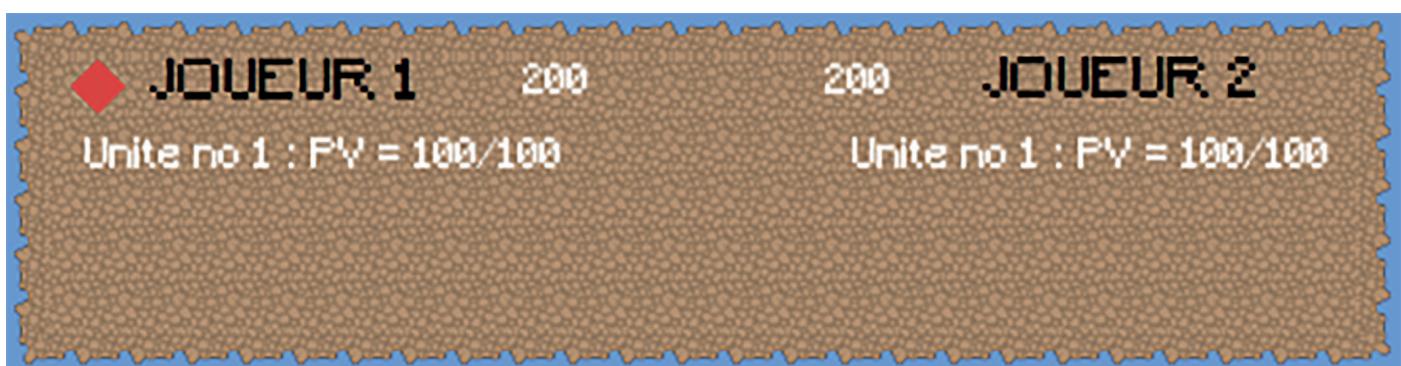
Si la partie lancée est une partie contre ordinateur, alors le joueur atterrit directement sur la map. Le joueur retrouve alors son ou ses unités ainsi que l'unique unité de l'ordinateur.

DANS UNE PARTIE :

Après avoir passer le mode introduction, les joueurs se retrouvent sur la map finale avec plusieurs éléments d'information à leur disposition.

INFORMATIONS BASIQUES :

Tout d'abord les informations relatives à l'état de leur.s unité.s et de leur cagnotte.



Affichage informations basiques

En fonction des tours, un marqueurs de couleurs vient s'accorder au nom du joueur qui doit jouer. Cela permet une meilleure compréhension des tours de jeu, cela améliore l'expérience utilisateur du jeu.

Dès le début de la partie les joueurs ont en leur possession une unité Loup, qui est l'unité la plus basique que nous ayons mise en place, dans le sens où elle dispose d'une force et d'une portée moyenne.

Le joueur peut dès le début de la partie décider d'acheter de nouvelles unités. Il a alors 3 choix qui s'offrent à lui :

acheter une autre unité Loup : force et portée moyennes

acheter une unité Ours : force augmenté mais portée et déplacement réduits

acheter une unité Hibou : force faible mais portée et déplacement augmentés (cette unité peut également voler au dessus des cases eau de la carte, un hibou ça vole...!)

Quand le choix des animaux s'est posé nous avons beaucoup réfléchis auxquels choisir. Nous voulions créer une équipe avec des facultés différentes mais complémentaires, d'où notre choix sur ces 3 animaux.

Pour renforcer la complémentarité de chaque unité, nous avons également pensé la map de telle sorte à ce que chaque unité ait un avantage sur les autres.

En effet sur la map on peut s'apercevoir qu'il y a des la forêt, une grotte et de l'eau. Seul le loup peut accéder à la forêt et la traverser.

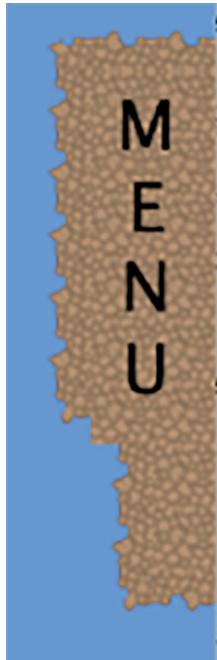
Concernant la grotte, seul l'ours peut y avoir accès et s'y cacher.

Et enfin seul la chouette peut voler au dessus de l'eau et peut ainsi traverser la rivière de la map alors que les autres types d'unités doivent la contourner ou emprunter le pont.

LE MENU

Les joueurs ont également accès au menu. Le seul moment où les joueurs ne peuvent pas appeler le menu c'est quand une unité va se déplacer ou jouer.

Quand un joueur clique sur le menu, une nouvelle fenêtre apparaît à l'écran.



Bouton menu



Affichage menu

Il a alors le choix entre les différents types d'unité que le jeu propose. Pour l'aider à faire son choix, toutes les informations relatives à l'unité (prix, force, mobilité...) sont renseignés.

S'il clique sur une des 3 unités, il retourne alors sur la map et sa nouvelle unité y est affichée.

Mais il a également le choix de revenir sur le jeu en cliquant sur le bouton retour.

En ce qui concerne le joueur ordinateur, il ne peut pas accéder au menu. Mais pour pallier à cet inconvénient, l'unité ordinateur est doté d'un plus grand nombre de pv, ce qui lui permet de résister et d'attaquer en conséquence.

TOUR DE JEU

Quand vient le tour d'un joueur, il lui est demandé de sélectionner une unité. Pour cela nous avons mis en place un curseur de sélection qui permet au joueur de se situer sur la map.

Pour sélectionner une unité il a juste à se placer sur l'unité choisie et appuyer sur entrée. Une fois sélectionnée, en fonction de la position de l'unité par rapport aux unités adverses, le joueur peut soit se déplacer soit attaquer. Si une unité adverse se trouve à la portée de l'unité sélectionnée alors le choix "attaquer" s'affiche.



Affichage choix action 1

Au contraire si l'unité est isolée, alors seul le choix "Se déplacer" est affiché.



Affichage choix action 2

Si le joueur choisit de se déplacer, il est alors libre dans son déplacement, qui s'effectue dans un périmètre correspondant à la mobilité du joueur, définie elle-même par son type. En effet le Loup, l'Ours et le Hibou n'ont pas les mêmes facultés de déplacement. Lors de son déplacement le joueur peut se heurter à des obstacles, tels que la forêt, l'eau ou encore la grotte, en fonction de son type il sera autorisé ou non à pouvoir franchir ses obstacles.

Pour déplacer son unité le joueur doit appuyer sur les différentes flèches du clavier en fonction de la direction qu'il souhaite.

Quand le joueur a atteint le nombre de case maximum que son unité peut parcourir, il lui suffit d'appuyer sur entrer pour terminer son déplacement.

Si le joueur choisit d'attaquer une unité adverse, un curseur attaque apparaît alors sur la map. Il ne lui reste alors qu'à sélectionner l'unité adverse qu'il veut attaquer en déplacement le curseur avec les flèches du clavier. Une fois le curseur sur l'unité adverse il lui suffit d'appuyer sur entrer pour déclencher l'attaque.

Une fois que l'action est terminée, déclenchée, c'est alors à l'autre joueur de jouer.

En ce qui concerne le joueur ordinateur, la stratégie que nous avons mis en place consiste à ce que la machine repère si une unité est dans le champs de portée du joueur ordinateur. Si c'est le cas alors le joueur attaque, si ce n'est pas le cas, l'unité se déplace simplement sur la carte.

Lors des tours de jeu, le joueur va également être confronté à des cases magiques, qui lui permettront de gagner des points de vie en plus en fonction de son état actuel. Pour pouvoir enclencher ces cases magiques il devra se rendre sur la case et valider sa position sur celle-ci.

FIN DE PARTIE

La partie se termine quand un joueur ne possède plus aucune unité encore en vie.



Affichage victoire Joueur 1



Affichage victoire Joueur 2



Affichage défaite Joueur vs Ordinateur



Affichage victoire Joueur vs Ordinateur

EXPÉRIENCE UTILISATEUR

L'expérience utilisateur mélange clics de la souris sur les différents boutons et également utilisation des flèches du clavier. L'utilisation des flèches du clavier est nécessaire pour la sélection d'une unité lorsque c'est à notre tour et lorsque l'on veut sélectionner une unité adverse à attaquer. La touche Entrée est également nécessaire pour valider son choix dans une des 2 situations énoncées précédemment.

Dans les autres situations, il est soit nécessaire de faire un clic gauche avec la souris sur les différents boutons.

EXEMPLES :

- * choix de l'action à réaliser entre Se déplacer et Attaquer,
- * choix entre les deux modes 2 PLAYERS ou VS COMPUTER.

Ou il est parfois possible d'interagir à la fois avec le clic gauche de la souris et à la fois avec la touche Entrée et la touche Echap.

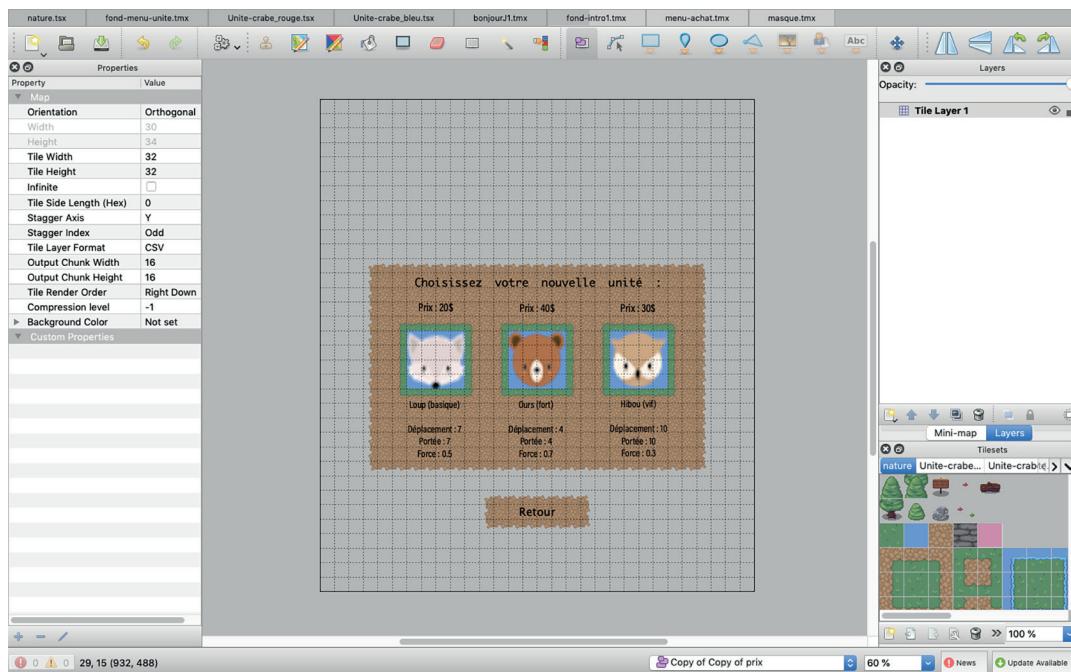
EXEMPLES :

- * Dans le menu Play/Quit on peut cliquer sur les 2 boutons mais en appuyant sur Entrée on entre aussi dans Play et en appuyant sur Echap on quitte le jeu;
- * Lorsque les premières unités des 2 joueurs sont présentées il est possible d'appuyer sur Entrée à la place de cliquer sur «Jouer» ;
- * Lorsqu'une erreur apparaît (problème de sélection d'unité) il est possible de cliquer sur OK ou d'appuyer sur Entrée pour revenir sur la map.

GAMEDESIGN

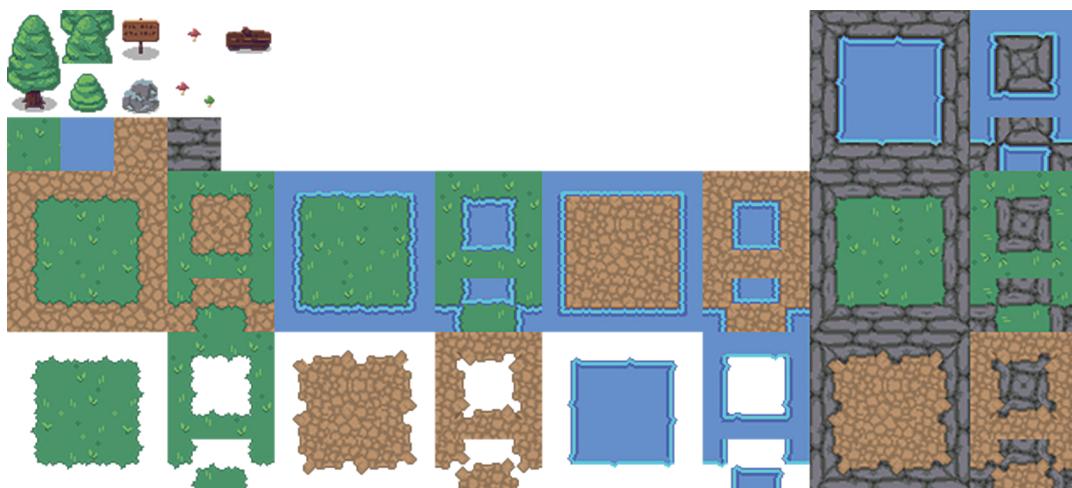
GRAPHISMES :

Pour promouvoir notre univers nous avons créé tous nos éléments graphiques grâce au logiciel Tiled; logiciel de création de map pour jeu 2D principalement.



Interface du logiciel Tiled

Nous avons utilisé ce tileset pour créer la map et les menus du jeu.

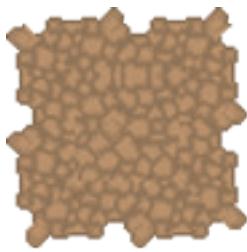


Tileset utilisé

Le sol de la carte peut être de 3 états différents : herbe, pierre et eau. Mais il peut également avoir des obstacles comme les arbres ou les pierres.



Herbe



Terre



Eau



Arbre



Rocher



Bûche

La mer, les arbres et les pierres représentent des cases «collisions» de la map non accessibles par tous les animaux. Le loup ne peut pas traverser l'eau ni aller dans la grotte en bas à droite mais peut aller dans la forêt. L'ours peut aller dans la grotte mais ne peut pas aller dans la forêt ni traverser l'eau. Enfin, le hibou, lui, ne peut pas aller dans la forêt et la grotte, cependant il peut traverser l'eau. Il ne peut pas s'y poser mais il possède des rochers dans le lac à droite sur lesquels il peut se poser.

On retrouve également un pont composé d'élément «bûches» qui traverse la rivière que tous les animaux peuvent emprunter.

Dans le mode VS COMPUTER, l'ordinateur prend la forme d'un chasseur humain qui peut se déplacer sur toute la map sauf dans l'eau.

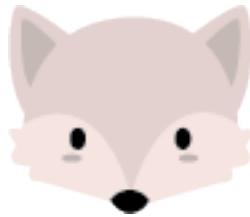


Chasseur

Dans le mode VS COMPUTER, l'ordinateur prend la forme d'un chasseur humain qui peut se déplacer sur toute la map sauf dans l'eau.



Hibou

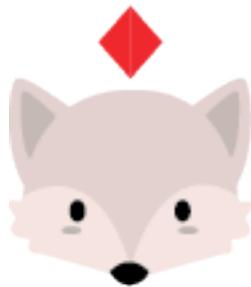


Loup



Ours

Concernant les animaux, nous les avons dessinés grâce au logiciel Adobe Illustrator. Pour permettre une différenciation entre les deux équipes, nous avons leur ajoutés une marque au dessus de la tête en fonction de la couleur du joueur auquel ils appartiennent.

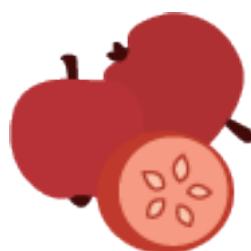


Unité J1



Unité J2

Le design de la case magique c'est également fait via Adobe illustrator. Nous avons décidé de dessiner les cases magiques sous forme de fruit, symboliquement manger est synonyme de reprendre des forces. C'est ce que propose cette case en faisant gagner des points de vie aux unités qui viennent jusqu'à elle.



Baies

TYPOGRAPHIES :

Pour les écritures, nous avons alternés entre utilisations de la bibliothèque SDL_TTF pour un affichage dynamique, notamment des informations relatives aux PVs des différentes unités des joueurs et également lors de l'affichage du texte de sélection et d'action d'une unité, et entre utilisations d'images avec des textes inscrits dessus lorsque ces derniers n'avaient pas besoin d'être dynamiques. (affichage des erreurs, des différents boutons etc..)

En typographie, nous en avons utilisé 3, dont une principale :

- * Minecraft (utilisée à la fois en affichage statique et en affichage dynamique avec la SDL_TTF) : Elle possède un aspect très «jeu vidéo» que l'on a voulu mettre en avant ici.

Minecraft Moyen

A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m
n o p q r s t u v w x y z
1 2 3 4 5 6 7 8 9 0

- * Apple Gothic : utilisée au début d'une partie 2 joueurs pour indiquer aux joueurs le positionnement de leurs premières unités.
- * 2 Tech : utilisée pour les différents boutons Play/Quit et 2 players/VS computer.

LE CODE :

ARCHITECTURE DU CODE :

Pour pouvoir faire fonctionner notre jeu correctement et jongler avec les différents affichages, nous avons décidé de créer des modes.

Ces modes sont appelés en fonction des évènements déclenchés par le joueur ou tout simplement en fonction des tours de jeu.

4 sont réservés aux introductions du jeu ;

1 pour l'affichage : Play & Quit

1 pour l'affichage du choix des types de parties : Players & Computer

1 pour l'introduction du joueur 1 :

«Bonjour Joueur 1»

1 pour l'introduction du joueur 2 :

«Bonjour Joueur 2»

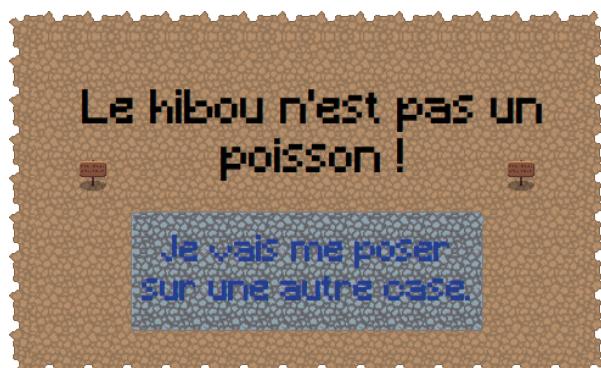
1 mode relatif à l'affichage de la map et des informations en fonction du joueur qui joue.

1 est réservé au menu achat ; affichage de la fenêtre achat.

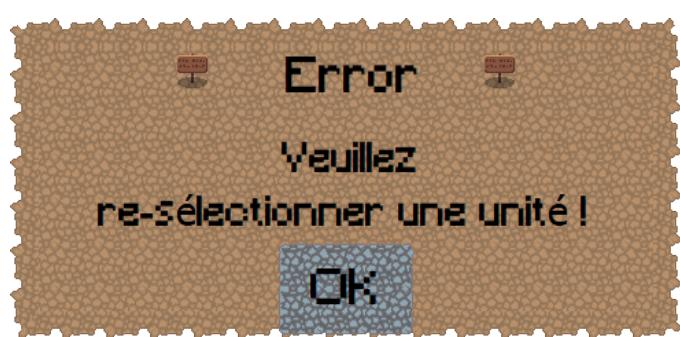
2 modes pour gérer les différentes actions ; se déplacer et attaquer.

2 sont destinés aux messages d'erreur auxquels peuvent être confrontés les joueurs :

Erreur chouette qui se pose sur l'eau.

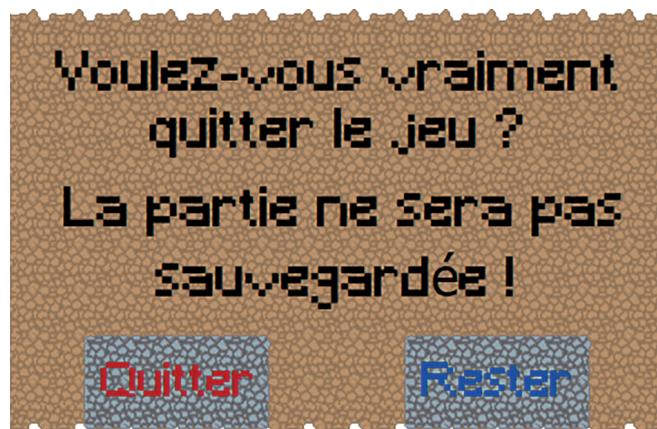


Erreur de sélection d'une unité.



1 mode est également présent pour afficher les messages de victoire en fin de partie.

Lorsque le joueur veut fermer la fenêtre cela enclenche le mode de sortie qui l'avertit que la partie ne pourra être restaurée.



Et un dernier mode qui permet d'afficher les commandes du jeu, au cas où le joueur n'aurait pas lu notre mode d'emploi avant de se lancer dans la partie.

A screenshot of a game help menu. On the left, there is a small icon of a question mark inside a brown square. The main text area has a brown textured background. It contains several sections of instructions:

Comment jouer ?

Sélectionner une de vos unité :
Utiliser les flèches du clavier pour déplacer le curseur
Appuyer sur « Entrée » pour sélectionner

Choix de l'action :
Cliquer sur « Se déplacer » ou « Attaquer » (quand cela vous est proposé)

Déplacer votre unité :
Utiliser les flèches du clavier pour vous déplacer
Appuyer sur « Entrée » pour valider l'action

Accès au menu :
Cliquer sur « Menu »

Menu :
Cliquer sur la nouvelle unité que vous voulez acheter
Cliquer sur « Retour » pour revenir sur la partie

Quitter le jeu :
Appuyer sur la touche « Echap » ou sur la touche « q » ou cliquer sur la croix de la fenêtre.
Puis appuyer sur « Quitter »



STRUCTURE DES DONNEES

Pour ce qui est de la structure des différentes données, nous avons eu du mal à visualiser la structure globale sachant que c'était la première fois que nous réalisions un jeu avec interface graphique. Avec le recul, nous nous rendons maintenant compte que nous n'avons pas une structure optimale et nous visualisons déjà des manières de procéder plus efficaces pour une prochaine fois.

Cela dit, nous avons plusieurs structures permettant de stocker nos données :

- * Joueur, permettant de stocker les différentes informations de chaque joueur (insérer le code de la structure ici)
- * Unité, permettant de stocker les informations de chacune des unités, notamment son type, sa position en coordonnées x et y, sa force, sa portée, sa mobilité, ses PVs et son image

Pour ce qui est de la map, c'est une image car on n'a pas réussi à réaliser un affichage grâce à une matrice notamment à cause des différents formats d'export du logiciel Tiled qu'on ne savait pas manipuler dans le code. Cependant, lorsque nous avons dû gérer les collisions nous avons mis en place un tableau de Case.

- * Case, structure permettant de savoir quel est le type de la case (si c'est de l'eau, de l'herbe, de la grotte ou de la forêt) et si la case est occupée (insérer code de la structure)

Ce tableau de 20 x 20 regroupe toutes les cases de la map et permet de tester si une unité peut aller sur une case selon son type et si la case est occupée ou non.

Nous voyons aujourd'hui que cette méthode n'est pas la plus efficace bien que fonctionnelle. Il aurait été possible de partir de la structure Case et de la modifier pour mettre en place un arbre regroupant toutes les cases.

Enfin nous avons également une structure :

- * Curseur, permettant de stocker la position en coordonnées x et y des curseurs ainsi que leur visuel (insérer code)

DIFFICULTES RENCONTREES

Lors de la réalisation nous nous sommes heurtées à plusieurs difficultés.

La première étant le matériel que nous avions à notre disposition. En effet, nous ne travaillions pas sur le mêmes systèmes d'exploitation (Mac et Windows), il a donc fallu au cours de nos envois de code rectifier quelques morceaux du codes afin que celui-ci soit exploitable par nos 2 systèmes. De plus nous avons toutes deux eu du mal à installer SDL sur nos machines. Mais l'un des problèmes est aussi que Nelly disposait d'un ordinateur capricieux qui s'éteind régulièrement et qui est peu puissant. Tout cela nous a fait perdre du temps sur l'avancement du projet.

L'autre difficulté rencontrée est que nous n'avons pas vu notre structure de code d'une autre façon, ce qui nous a porté préjudice sur les différentes fonctions demandées en programmation algorithmique. Quand nous nous en sommes aperçus, l'ordinateur de Nelly à malheureusement cessé de fonctionner et la date de rendu approchait...

Quoi qu'il en soit nous avons toutes les deux appris beaucoup durant ce projet et nous avons pris du plaisir à le réaliser.

REMERCIEMENTS

Nous tenions également à remercier le corps enseignant, qui a su s'adapter à la situation exceptionnelle que nous avons traverser avec la crise sanitaire.

Les professeurs ont su faire preuve de réactivité et ont toujours été à l'écoute et présent lors de nos demandes.