Ethnirybal

Jeu de Labyrinthe

< Sommaire >

- 1. Introduction
- 2. Manuel Utilisateur
- 3. Documentation Technique
- 4. Conclusion

< Introduction >

Dans le cadre du projet de fin du premier semestre de la Licence Mathématique-Informatique, nous avons réalisé un programme permettant de se déplacer dans les couloirs d'un labyrinthe en vue subjective. Le labyrinthe, généré aléatoirement est stocké dans un fichier texte permettant sa lecture par le programme. De plus, quelques améliorations ont été ajoutées au projet de base.

< Manuel d'Utilisation >

Comment Jouer?

Pour démarrer le jeu, lancer le fichier « Ethnirybal » et attendez quelques instants (Vous pouvez passer l'affichage « ProjectX » au démarrage via un clic ou l'appuie sur n'importe quelle touche).



Une fois sur le menu principal, utilisez la souris pour sélectionner l'option désirée :

Continuer : Reprendre votre partie sauvegardée (si une partie a été sauvegardée)

Reprendre

Sauvegarder

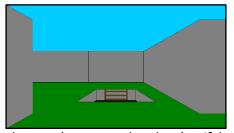
- Nouvelle Partie : Commencer une nouvelle partie depuis le début
- Quitter : Fermer le jeu

Vous pouvez utiliser le clic de votre souris pour interagir avec tous les menus du jeu comme le menu pause accessible en jeu depuis le bouton :

Lorsque vous êtes dans le labyrinthe, vous pouvez vous déplacer avec les boutons suivant :

- ¶ 5 d pour faire des déplacements dans les 4 directions suivantes : ↓ •
- a e pour faire une rotation de 90° à gauche (a) ou à droite (e)

Présentation du Jeu



Dans ce jeu, le joueur se retrouve dans un labyrinthe en vue subjective dans lequel il peut se déplacer afin d'atteindre la sortie afin de passer au labyrinthe suivant. Cependant, il y a un défi : le temps pour finir le labyrinthe est limité. En effet, quand le temps restant total arrive à 0, le joueur perd la partie. De plus, chaque labyrinthe est chronométré indiviuellement en fonction de la difficulté choisie au début (facile, normal, difficile, ...) et si ce

Temps Restant Total 01:08

Temps Restant 00:35

Score 000171

chronomètre passe dans le négatif, le temps restant total diminuera plus vite.

Mais pas de panique! Lorsque le joueur arrive à la sortie représentée par une échelle, il augmente son score et sont temps restant total est lui aussi augmenté afin de continer à jouer.

Pour se repérer plus facilement, il dispose également d'une boussole indiquant la direction dans laquelle il regarde ainsi qu'une mini carte qui se dessine avec la progression du joueur dans le labyrinthe.



Triangle Rouge: Joueur Rond Bleu: Point de Départ

Carré Vert : Sortie Carré Noir: Mur



Enfin, sur le menu pause, le joueur peut sauvegarder sa partie afin de la reprendre plus tard.

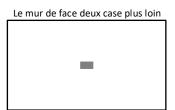
Documentation Technique >

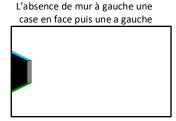
Afin de réaliser ce projet, nous avons développé diverses fonctionnalités : les déplacements, l'affichage, les menus, la carte, la sauvegarde, etc... Nous allons présenter ici les fonctionnalités les plus techniques.

Affichage du Labyrinthe

L'affichage d'un labyrinthe en vue subjective pose un principal problème : le grand nombre de possibilités de choses à afficher. Or nous ne pouvions pas dessiner chaque image pour chaque cas possible. Nous avons donc décidé de créer un système d'affichage avec des textures « modulables » permettant d'afficher indépendamment chaque partie du labyrinthe (un mur à droite, un sol loin devant, un mur au fond à gauche, etc...). Voici quelques exemples de ces textures :



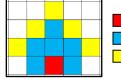






Pour choisir quelle image afficher, le programme va d'abord vérifier l'état de la case en face du joueur. Si c'est un mur, on affiche un mur, sinon le programme va vérifier les cases autour de celle-ci. Et le programme refait cela jusqu'à ce qu'il ait affiché tout le champs visuel du joueur. Cela permet de ne pas avoir à afficher les images de la case tout au fond si c'est en dehors du champs visuel à cause d'un mur se trouvant en face du joueur.

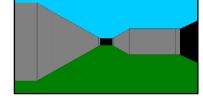
Le champs visuel du joueur est limité suivant ce schéma :



Joueur regardant vers le haut

Case où l'affichage du sol et des mur peut être fait

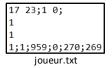
☐ Case où uniquement l'affichage du mur adjacent à ☐ peut être fait



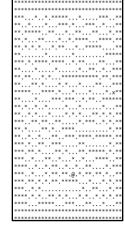
Sauvegarde

Etant donné la potentielle longueur d'une partie, une option de sauvegarde a été implémentée afin d'accorder des pauses au joueur.

Lorsque celui-ci appuie sur le bouton « Sauvegarder », sa position dans le labyrinthe, sa direction, mais aussi son score et les temps sont sauvegardés dans un fichier « joueur.txt ». Le labyrinthe et la carte, quant à eux, sont sauvegardé chacun dans un fichier unique pour faciliter leur lecture.



Les données ne sont pas sauvegardées dans le fichier sous forme de listes ou de nombres mais elles sont converties en chaines de caractères au moment de la sauvegarde puis reconverties dans leur type d'origine lors du chargement.



labyrinthe.txt

Carte

Dans un labyrinthe en vue subjective, il est facile de se perdre. Nous avons donc ajouté une carte qui se dessine au fur et à mesure de la progression du joueur afin de l'aider sans lui dévoiler tout le labyrinthe. Voici les différentes étapes que nous avons réalisé pour créer une carte :

- Création d'une carte vide: On créé d'abord une liste de listes de la même taille que le labyrinthe contenant que des . représentant les chemins de notre labyrinthe puis on les entoure avec deux lignes de * car tous les labyrinthes que l'on créé ont deux couches de murs afin de faciliter l'affichage. (module jeu fonction init carte)
- Remplissage de la carte : A chaque itération de la boucle principale du jeu, on actualise la carte en ajoutant dans la liste de listes les éléments du labyrinthe proches du joueur, c'est-à-dire ceux se trouvant aux emplacements des du schéma (le joueur regarde vers le haut). On permet de « voir à travers les murs » via la carte. (fonction actualise_carte)
- Affichage de la carte: Pour afficher la carte, nous avons choisis de donner un champs de vision de rayon 6 autour du joueur. On redimensionne donc la carte à une taille de 13x13 avec le joueur au centre. On fait attention quand le joueur est à moins de 6 cases du bord du labyrinthe. En effet, pour éviter les erreurs, on décide que toutes les cases de la carte en dehors du labyrinthe deviennent des murs. (fonction redimension_carte)
 Puis on affiche simplement la carte avec les fonctions de fltk.



Score / Temps

Pour ajouter une difficulté au jeu nous avons implémenté un système de temps. Cela permet de créer un enjeu lors de la partie pour donner plus d'intérêt au jeu.

Puisque la taille des labyrinthes est choisie aléatoirement, cela fait un bon élément de référence pour le score.

Pour le temps on pourra se fier à la surface du labyrinthe : plus la surface d'un labyrinthe est grande, plus le temps nécessaire pour le parcourir est grand et inversement, plus la surface d'un labyrinthe est petite, plus le temps nécessaire pour le parcourir est petit. Ces valeurs sont déterminées par une fonction issue de divers tests. Par exemple, un labyrinthe de 25x25 aura un temps de parcours de 3 minutes dans la difficulté facile.

Cependant pour éviter les frustrations liée à des génération de labyrinthe peu avantageux nous avons décidé d'ajouter un deuxième compteur : le temps total autorisé. Celui-ci augmente (en même temps que le score) à chaque fois que le joueur réussit à finir un labyrinthe en dessous du temps de parcours et c'est ce compteur qui détermine la fin d'une partie.

Temps de Jeu	Temps Restant Total	Temps Restant Du Labyrinthe En Cours	Score
00:05	03:40	01:06	000000

Génération du Labyrinthe

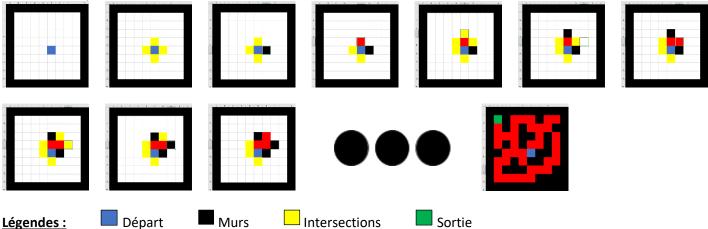
Génération:

Pour pouvoir générer un labyrinthe, nous deviens trouver un algorithme qui permettrait de produire des schémas variés de couloirs et de murs. Nous avons donc créé un algorithme simple basé sur l'aléatoire dont le fonctionnement est le suivant :

- Créer un labyrinthe vide d'une taille générée aléatoirement
- On ajoute les murs de 2 d'épaisseur autour
- On pose un curseur à une position aléatoire (Emplacement de départ du labyrinthe)
- Boucle While
 - o Si au moins deux cases adjacentes au curseur sont vide : On met un mur sur l'une d'entre elles
 - On avance dans une direction aléatoire
 - Si il reste des cases vides autour de la position précédente du curseur, on les enregistre en tant qu'intersection
 - O Si le curseur est bloqué (si il ne peut plus se déplacer) : Il se « téléporte » à une intersection aléatoire
 - Si il n'y a plus d'intersection : Fin de la génération (Le dernier emplacement du curseur représente la sortie du labyrinthe)

Cet algorithme nous permet de générer des labyrinthes variés avec paternes plutôt intéressants.

Voici un exemple d'exécution :



Affichage:

Pour afficher l'écran de chargement du labyrinthe, on prend les éléments autour du curseur à chaque répétition de l'algorithme et on regarde leur état (chemin, mur, intersection, etc...). On affiche alors à l'endroit correspondant un carré de couleur correspondant à cet état.

Cependant, avec cette méthode, on remarque de temps en temps une superposition du bord des carrés. Pour corriger ces bavures, on a décidé de réinitialiser le rendu de temps en temps en effaçant l'affichage pour le redessiner depuis les éléments connus du labyrinthe et ainsi lisser le résultat.

Dès la fin de l'affichage (qui peut être passé avec un clic de souris ou l'appuie sur une touche de clavier), le jeu reprend et le joueur peut se déplacer dans le labyrinthe qui vient d'être généré.

AP1 | Projet Fin S1: Labyrinthe | Bernier Valentin & Coserariu Alain



Pour conclure, nous avons trouvé ce projet intéressant car il a permis de développer nos compétences en algorithmes notamment grâce à certaines fonctionnalités complexes comme l'affichage en vue subjective ou la génération de labyrinthes aléatoire. Pour programmer ce projet, nous avons du avoir une excellente communication et une très bonne organisation. Pour cela nous avons écrit un document où nous répertorions chaque version du projet avec son descriptif.

Cependant, nous aurions aimé implémenter des dictionnaires pour limiter notamment le nombre de paramètres de certaines fonctions, ce qui aurait fait grandement gagner en lisibilité. Un autre point sur lequel nous aurions aimé plus insister est la rédaction du rapport. En effet, le contenu d'un rapport ne nous paraît pas encore assez clair et pose des difficultés.

١tı			oserariu Alain
ncies	Date Début	Date Fin	Contenu Ajouté / Modifié / Supprimé
0.1.0x	06/12/2020	06/12/2020	Au commencement, non, ensure vietre le tout
4.1334	08/12/2020	46122220	Squelette des fondamentaux du programme : Merss Boucle de jeu Sourrepartie Fonctions utilitaires
0.1.0	06/17/2020		Squelette des fondamentaux du programme : • Monte • Boucle de Jeu • Sauregarde • Fondions utilitaires
0.2.0v	07/12/2020	60/12/2020	Création des textures modulables (metils simples)
0.2.0a	07/12/2620	07/12/2020	
0.1.1	04/01/		Ajout de l'entite aux fichiers
0.2.0	04/01/2021		Menu + redirection vers le jeu eu quitter le jeu Création des textures modulables (motifs simples)
0.3.0v	04/01/2021	6401/2021	Gestion du fichier du labylinthe : importation dans une liste de listee • "pour un nur • paur une case vale • gli pour le juseur au déout de la partie • x pour la sarda
0.3.0a	04/01/2021	0401/2021	Gestion des déplacement théorique (rotations incluse)
0.3.0	04/01/		Gestion du fichier du labyrinthe : importation dans une liste de listes
0.4.0v		05/01/2021	Gestion de l'affidhage du labyriethe visites tectures modulables an fonction du labyriethe, de la position et de la rotation - Ajout Texture provisaire sortie
0.4.0	05/01/		Ajout déplacement du Joueur
0.5.0	05/01/2021		Ajcut de l'affichage du labyristhe MAJ type de l'orientation (liste)
0.5.1			M/J Détection événement
0.5.2	09/01/		MAJ Callisions mer MAJ Rotelion joueur MAJ Déplecement joueur
0.5.3		V2021	Ajoet Lago Démarrage (Fajout le plus Impertant évidement)
0.5.4		V2021	MAJ Documentation (Commentaires, doctests) Correction of éléments en fonction de gylesse
0.6.0	05/01/2021		Aject de l'emplacement et de la rotation de déparautementique en fonction du fichier lobyrinthe MAI du halynishte de démo Correction du hog l'impossible de quiter en jeu
0.5.1	09/01/2021		Correction doctasts
0.6.2	09/01/2021		Légère amélioration graphique (forme textures provisoires) : Changement de preparieres (mur de femil) Changement au whoau du "breuilland" Changement forme Scrite
0.7.0a		09/01/2021	
0.7.0v	89/81/2021	69/01/2021	Bossole Déplacement de l'écran de jeu Correction amplacement de l'artichage
0.7.0	09/01/2021		Monu Pause Doscole Dejohoment de l'écran de jeu Correction emplacement de l'éffichage Correction pylates
0.0.0v	10/01/2021	13010021	Agrandiscement du champs de vision : Nouvelles textures Modification de textures existantes Récenture presque totale des fonctions d'affichage
0.8.0a		10.01/2021	Sauvegarde de la partie
0.8.0		V2021	Agrandissement du champs de vision : Norroelles Sedunes : Medification de bestures additions : Réchture presque table des fonctions d'affichage : Savegarde de la parigne de
0.0.0v			Ajout de la map
0.9.04	13/01/2021	13010021	Générateur de labyrinthe eléatoire Modification des sauvegarde pour les labyrinthes
0.9.0	1301	12021	Map
0.10.0		V2021	Générateur de labyrinthe aléatoire Modification des sauvegardes pour les labyrinthes
0.11.8v	14/91/2021	14/01/2021	Ajout de la gestion d'étages Modif : Quand on appute sur la croix en jeu, cela envelusur le menu pause Réfonte système saune parde
0.11.fb	14012021		Ajout d'un écren de disrigement montrant le construction d labyrinthe
0.11.0	1,400	(2021	
	14/01/2021		Ajout de la gestion d'étages Modif : Quand on appuie sur la croix en jou, cela envoi sur le meru pause Rédute syntéme seure-parde
0.12.0v	14910021	14/01/2021	Chronomètra Sorre Ajust éléments à la seurregarda Miso en pièce système de difficultà (il manque le mon pour la theleir) On the comment of
0.12.0	1401/2021		Chronomètre Sorre Aparl Alements à le seurogande Illisse en place système de difficulté (il manque le mon pour la cheloiri)
1.0.0	15/01/2021		Todares: Mens Interface