

# Projet Informatique : Labyrinthe



BERGONT Christopher

LA TORRE Alexandre

ITI3 P16

# Table des matières

<b>TABLE DES MATIERES</b>	<b>1</b>
<b>1<sup>ERE</sup> PARTIE : PRESENTATION</b>	<b>2</b>
I) INTRODUCTION	2
II) CAHIER DES CHARGES	2
III) GANTT	3
IV) ROLES AU SEIN DE L'EQUIPE	3
<b>2<sup>EME</sup> PARTIE : NOTICE D'UTILISATION</b>	<b>4</b>
<b>3<sup>EME</sup> PARTIE : ETUDE LOGICIELLE</b>	<b>5</b>
I) PARTIE MENU	5
A) LA BARRE	5
B) LE MENU	6
C) LA NAVIGATION	7
II) PARTIE LABYRINTHE	8
A) LE DESSIN DU LABYRINTHE	8
B) LE DEPLACEMENT DU CURSEUR	9
C) LES FONCTIONS	10
<b>4<sup>EME</sup> PARTIE : PROPOSITION D'AMELIORATIONS ET CONCLUSION</b>	<b>11</b>
I) DIFFICULTES RENCONTREES	11
II) PROPOSITIONS D'AMELIORATION DU PROJET	11
III) CONCLUSION GENERALE	11

# 1<sup>ère</sup> Partie : Présentation

## I) Introduction

A première vue, le dessin du labyrinthe est un système de lignes plus ou moins décoratif, un motif géométrique plus ou moins symétrique, plus ou moins complexe. Et pourtant, l'essence du labyrinthe est dans son chemin, non pas dans son dessin. Un labyrinthe est fait pour être parcouru.

Un modèle réduit suffit au parcours : soit simple parcours visuel, soit à l'aide du doigt ou d'un instrument fin. Déjà le parcours visuel du chemin éveille des sensations kinesthésiques et permet de sentir le rythme du labyrinthe. L'utilisation du doigt est encore plus facile et plus efficace.

Le labyrinthe classique est à chemin unique, sans boucles ni impasses. Le plus connu est celui du sol de la cathédrale de Chartres en France ; il a été incorporé au dallage de sol de la cathédrale autour de l'an 1200.

## II) Cahier des charges



Le projet a pour but de créer un jeu de labyrinthe, il doit avoir au minimum une fonction de sauvegarde de la partie et une de chargement de la dernière partie sauvegardée.

Le chargement du labyrinthe à partir d'un fichier texte et la fonction quitter le programme doivent obligatoirement être implémentées.

Nous pourrions aussi choisir entre 4 schémas de labyrinthes différents et détecter l'arrivée à la fin de ceux-ci.

L'affichage d'un score, du temps et du chemin parcouru est optionnel.

Tout ceci est à réaliser en binôme.

FIGURE 1

### III) Gantt

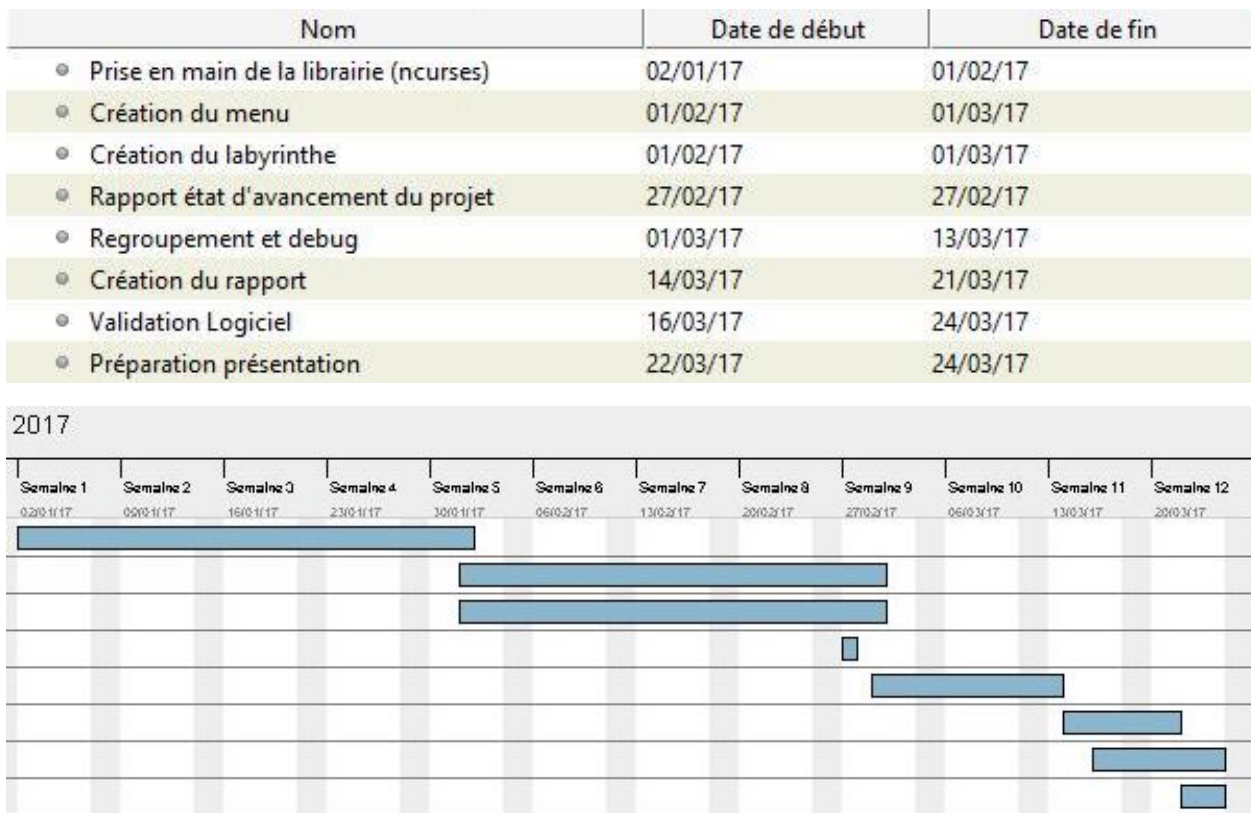


FIGURE 2

### IV) Rôles au sein de l'équipe

Christopher a donc désigné le menu graphique auquel Alexandre à incorporer l'algorithme du labyrinthe.

Christopher a organisé la planification du projet en mettant en place des délais pour les différentes fonctions à réaliser. Il a également réalisé l'architecture du projet. Alexandre a fait les choix techniques afin de réaliser un projet au plus proche du cahier des charges.

## 2<sup>ème</sup> Partie : Notice D'utilisation

Le menu est composé de 5 choix différents détaillé ci-dessous :

New game: Cette option nous permet de charger une nouvelle partie.

Load game: Grace à ce choix on peut charger une partie en récupérant la dernière position enregistrée. En effet nous pouvons sauvegarder celle-ci grâce à « save game ».

Save game: Cette option permet de sauvegarder la position et le fichier texte utilisé pour dessiner le labyrinthe afin de la charger ultérieurement.

Load labyrinth : Ce choix permet de changer de labyrinthe.

Quit : Permet de quitter le programme

### 3<sup>ème</sup> Partie : Etude Logicielle

#### I) Partie Menu

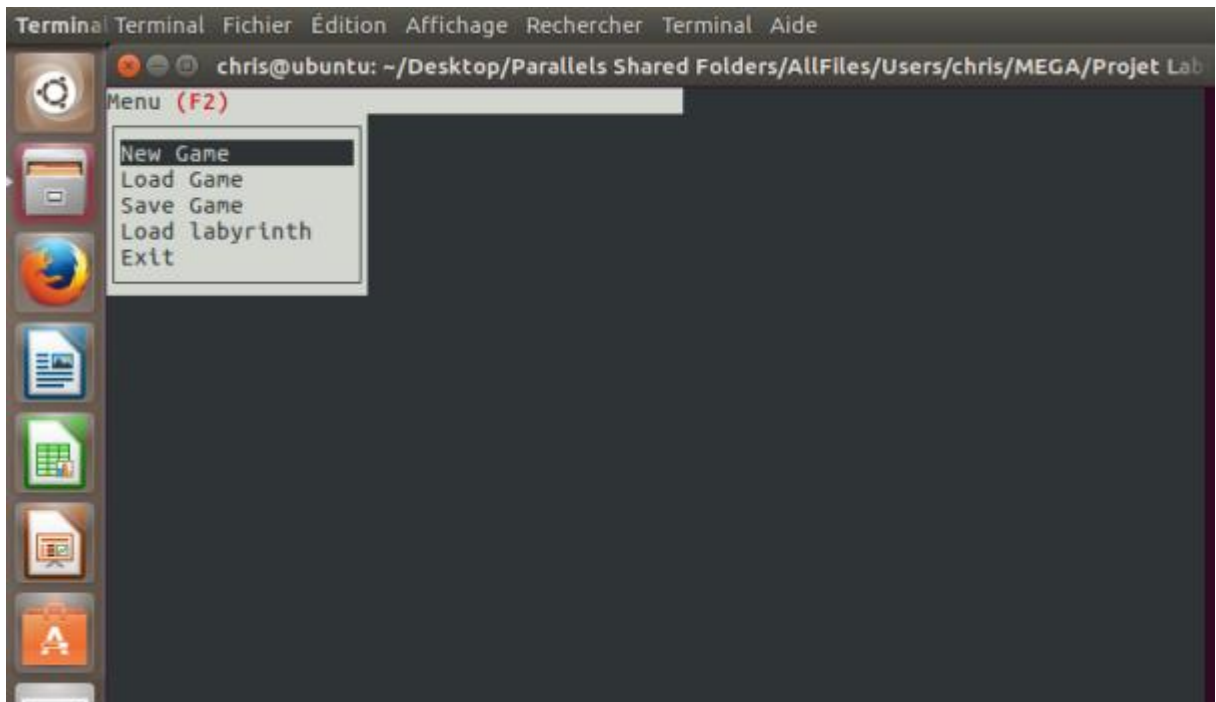


FIGURE 3

#### a) La barre

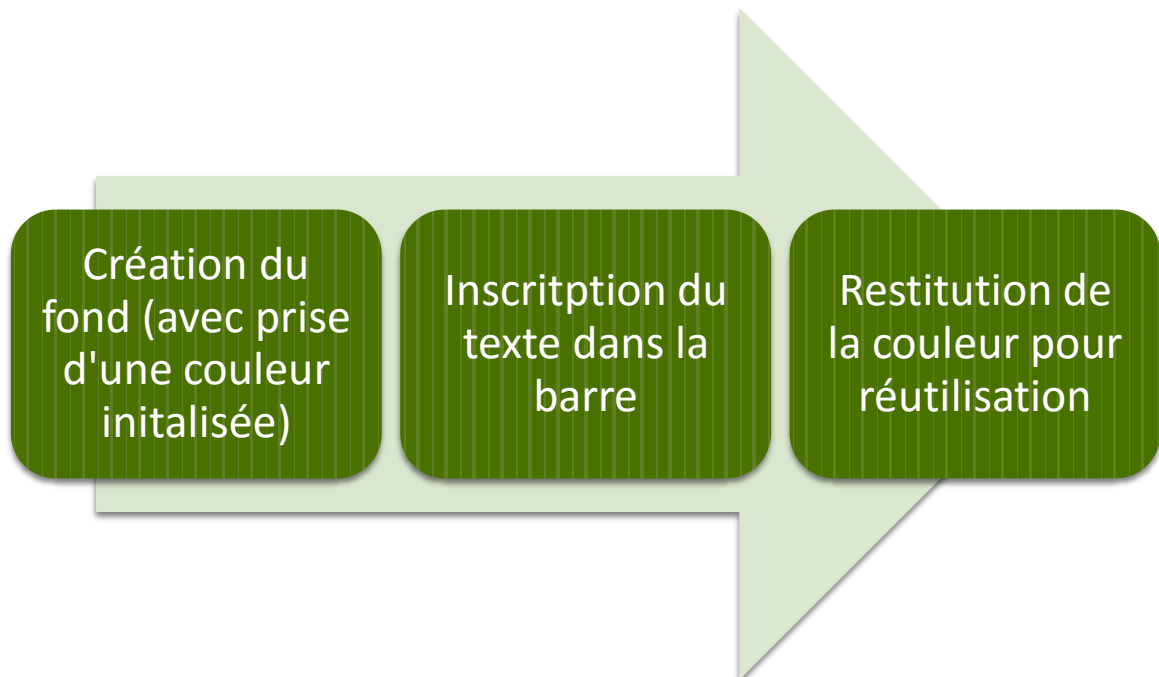


FIGURE 4

La création de la barre se fait en « coloriant » un fond, cette couleur est initialisée au départ, on inscrit ensuite le texte voulu dans ce fond.

## b) Le menu

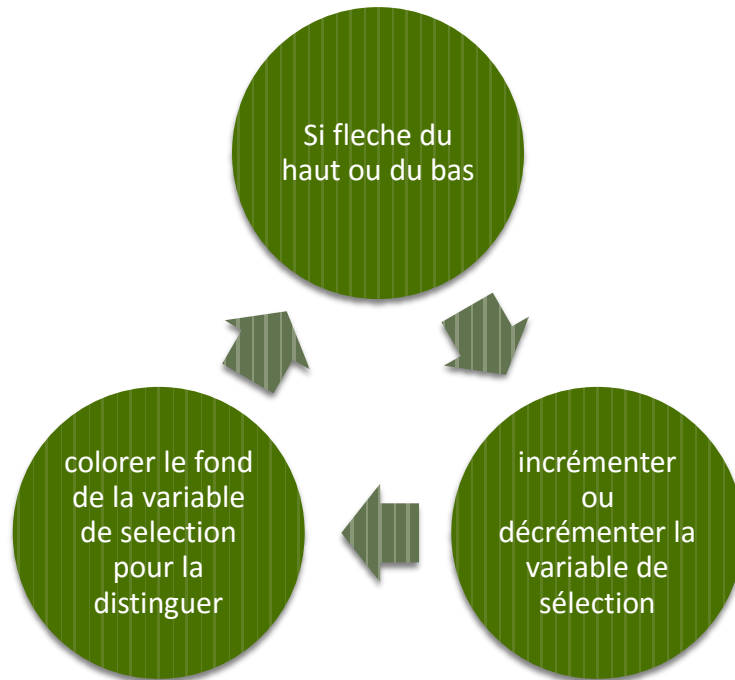


**FIGURE 5**

Le menu déroulant est le plus difficile à réaliser car il y a des problèmes de décalages de x ou y. Surtout lorsque l'on crée un contour, il faut prendre en compte le décalage de l'axe des x pour ne pas écrire sur le contour.

Pour créer ce menu on utilise la même méthode du fond en lui attribuant une taille en fonction du nombre de choix que l'on va mettre dedans. On crée le cadre « pour le design », les items les un au-dessous des autres.

### c) La navigation



**FIGURE 6**

La navigation est plutôt simple à coder. On utilise deux touches du clavier pour naviguer (ici flèche haut ou bas), lorsqu'une des touches est actionnée on incrémente ou décrémente une variable de sélection. Afin de visualiser ceci, on colore le fond de la variable de sélection (pour que l'utilisateur sache sur quel item il pointe)



## II) Partie Labyrinthe

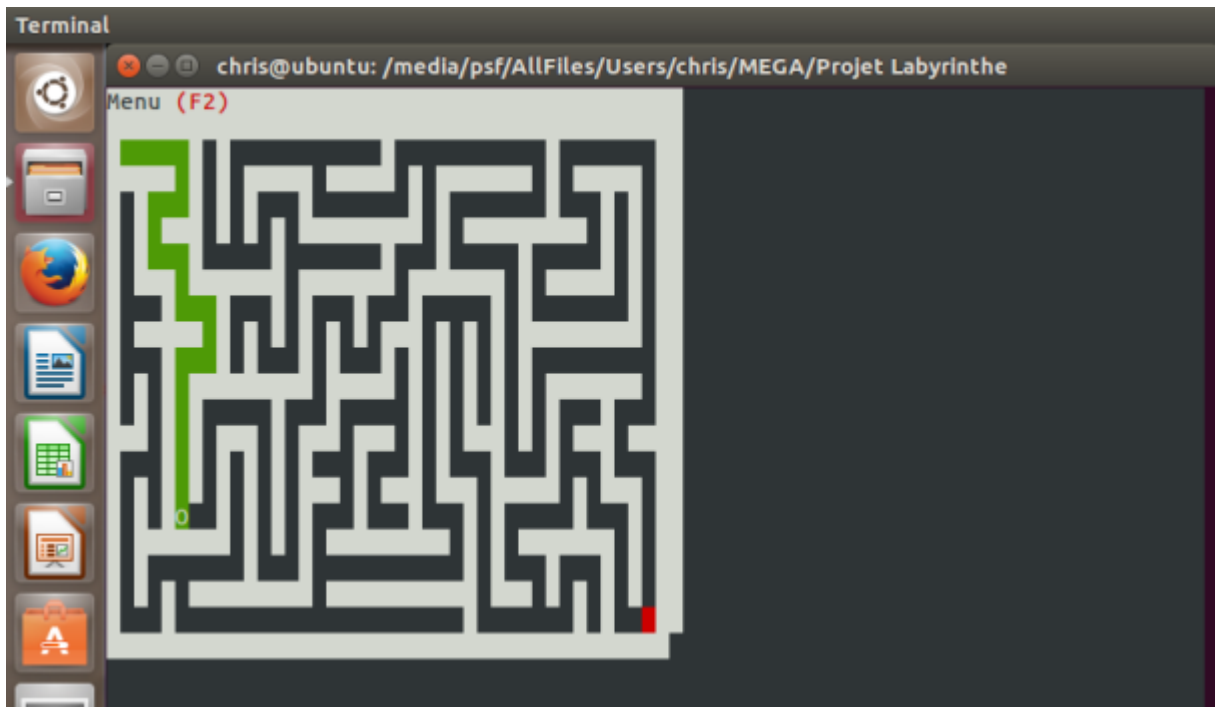


FIGURE 7

### a) Le dessin du labyrinthe

Le dessin du labyrinthe est effectué dans une fonction qui crée une matrice 2x2 de taille fixe et qui l'initialise à 1 (1 est interprété comme étant un mur). La fonction va ensuite lire un fichier texte et, selon la valeur de la case du tableau, attribuer à chaque case un rôle.

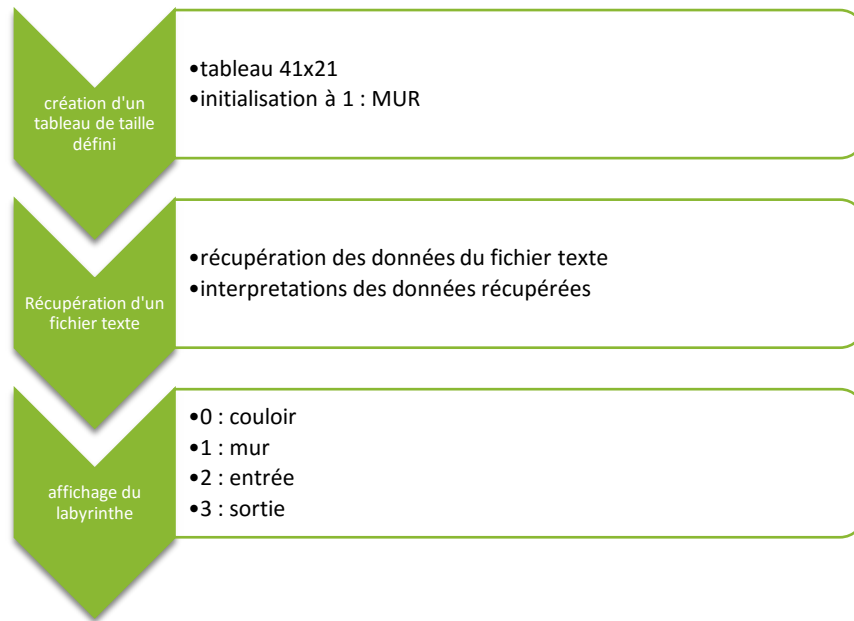


FIGURE 8

### b) Le déplacement du curseur

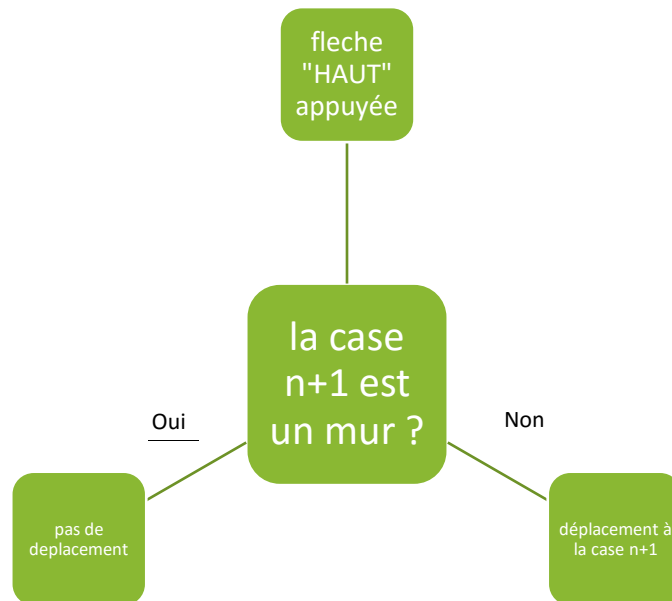


FIGURE 9

L'exemple ci-dessus explique le principe de fonctionnement du déplacement du curseur. Le principe est de scruter la case n+1 dans le sens de déplacement demandé par l'utilisateur à l'aide des flèches « HAUT », « BAS », « GAUCHE », « DROITE » et de vérifier si celle-ci est un mur ou pas. Si c'est un couloir le curseur est déplacé à la position n+1 sinon le curseur ne bouge pas. L'opération est répétée tant que le curseur n'a pas atteint la position finale.

### c) Les fonctions

New game : cette fonction dessine uniquement le labyrinthe comme expliqué ci-dessus.

Load game : cette fonction dessine le labyrinthe en prenant compte de la position de départ écrite dans un fichier texte si celui-ci existe. L'item correspondant à la position de départ par défaut du labyrinthe est donc ignorée.

Save game : cette fonction sauvegarde la position actuelle du curseur ainsi que le chemin du fichier texte utilisé pour dessiner le parcours.

## 4<sup>ème</sup> Partie : Proposition d'améliorations et conclusion

### I) Difficultés rencontrées

Les aspects graphiques de la librairie ont été difficiles à manier au départ, ensuite lorsque ceux-ci ont été maîtrisés le menu et le labyrinthe ont été assez faciles à mettre en œuvre.

Cependant le regroupement des deux programmes a été l'une des plus grandes difficultés rencontrées car de nombreux bugs d'affichages sont apparus à sa suite.

### II) Propositions d'amélioration du projet

L'affichage du score et du temps était un de nos objectifs que l'on n'a malheureusement pas pu réaliser faute de temps.

Ainsi que la génération aléatoire qui prend beaucoup de temps à mettre en place.

L'ajout de musique ainsi que d'une animation en fin de parcours semblerait aussi être une bonne amélioration.

### III) Conclusion générale

Ce projet nous a permis de consolider nos connaissances en C du dernier semestre, on a aussi pu utiliser une librairie inconnue, ce que nous n'avions pas réellement réalisé jusqu'ici. Tout ceci implique une recherche de documentation et un apprentissage de l'utilisation des commandes. Nous avons essayé de réaliser un code « professionnel », c'est-à-dire en créant des headers afin de rendre le code le plus propre possible.

En ce qui concerne le livrable, les fonctions obligatoires ont toutes été implémentées, en revanche en ce qui concerne les options seule le tracé du chemin a été réalisé. Le cahier des charges total a donc été réalisé à 60% (cahier des charges principal 100%, cahier des charges optionnel 20%)