PACMAN PROJECT



Le but de notre projet est de réaliser le célèbre jeu PacMan en C. Le but du PacMan est de manger tous les pixels disposé dans un labyrinthe fixe. En même temps des fantômes parcourent le labyrinthe pour essayer de tuer PacMan. Lorsque PacMan est touché par un fantôme, PacMan perd une vie. Cependant des pixels bonus (représentés par des carrés plus gros) sont disposés sur toute la carte permettant à PacMan de manger les fantômes. On pourra définir plusieurs niveaux qui joueront sur le nombre de vies que PacMan aura lors du commencement de la partie ou le nombre de fantômes.

Le jeu s’affichera dans la console. Il sera possible d’enregistrer son nom afin de pouvoir établir une liste des meilleurs scores. Le déplacement du personnage s’effectuera à l’aide des touches Z,Q,S et D. Un nombre défini de fantômes sortira d’un endroit à intervalle régulier.

Ce projet vous sera présenté par Paul TREHIOU et Victor SENE actuellement en TC02.

Sommaire

Table des matières

[1 Le Main 3](#_Toc389817060)

[1.1 Menu 3](#_Toc389817061)

[1.2 Rendu arène 3](#_Toc389817062)

[2 Les déplacements 3](#_Toc389817063)

[3 Les scores 3](#_Toc389817064)

# Le Main

Le programme appel en premier main.c qui déclare et initialise les éléments principaux du jeu à savoir le terrain et les coordonnées du PacMan et des fantômes. Le PacMan sera représenté par un C et les fantômes par des M. On initialise aussi les tableaux qui recueilleront les scores et pseudo.

Nous avons choisi d’utiliser une structure pour les coordonnés des entités du jeu, facilitant le codage et la compréhension du programme.

Nous utiliserons dans ce programme une nouvelle bibliothèque nous permettant d’utiliser des couleurs pour rendre le visuel plus attrayant.

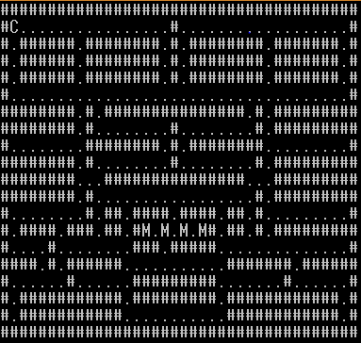
Notre but fût de coller le plus possible au jeu d’arcade mythique qu’est le PacMan, ainsi s’explique le choix des pseudos de 3 caractères notamment.

## Menu

Dans ce fichier nous afficherons un menu pour permettre à l’utilisateur de choisir l’action qu’il veut effectuer. Il pourra ainsi se renseigner sur les

## Initialisation

On génère tous les murs, les fantômes, PacMan et les bonus



## Rendu arène

Cette fonction remplace les bords du terrain symbolisés par des #, par des caractères ASCII rendant le parcours plus lisible. L’affectation des symboles est faite dans symboles.h

## Affichage

Le terrain est affiché une première fois

## Déplacement

On lance le jeu

## lectureScore

On lit tous les scores présents dans le fichier score.txt pui on récupère le nombre d’entrées

## triScore

On trie les scores et on les affiches

PROGRAMME PacMan

Variables tableau de caractère terrain [20][38]

S\_coordonees PacMan

S\_coordonees fantomeA

S\_coordonees fantomeB

S\_coordonees fantomeC

S\_coordonees fantomeD

entier mode, maxi

Tableau de 20 chaines de 38 caractères terrain

Tableau de 100 chaines de 4 caractères pseudo[100][4] //Les deux tableaux suivants servent à l'affichage des meilleurs scores

DEBUT

SetWindow(39,25) //forcer une taille de la fenêtre dans le CMD

initialisation du générateur de nombre aléatoire

FAIRE

menu(&mode)

CHOISIR PARMI(mode)

{

case 1 :

initialisation(terrain, &PacMan, &fantomeA, &fantomeB, &fantomeC, &fantomeD) //On génère l'arène

renduarene(terrain) //On effectue un rendu sur l'arène précédement générée afin d'avoir un résultat plus joli

affichage(terrain, 0, 0, 0) //On affiche une première fois le terrain

deplacements(terrain, &PacMan, &fantomeA, &fantomeB, &fantomeC, &fantomeD) //On lance la gestion des déplacements

break

case 2:

system("CLS")

maxi <-- lectureScore(pseuFAIRE, score)

triScore(pseudo, score, maxi)

break

default:

break

}

TANT QUE(mode>0 ET mode<3)

FIN

# Les déplacements

Les déplacements seront déclinés en 2 fichiers : l’un pour les déplacements généraux et l’autre pour les déplacements des fantômes qui doivent avancer sans manger les points du terrain, à l’inverse du Pacman.

# Les scores

La fonction fscore (contenu dans score.c) est appelée en fin de partie par le programme pour effectuer les tâches telles que la sauvegarde et l’affichage des meilleurs scores. On aura un donc une fonction pour écrire le score dans un fichier .txt (inscriptionScore.c) puis un algorithme effectuant la lecture des scores du fichier et l’inscription de ceux-ci dans un tableau (lectureScore.c) pour finir par un tri (triScore.c).

TriScore.c permet à la fois de classer les scores des différents joueurs, mais il appelle aussi la fonction affichageBestScore.c qui va afficher jusqu’au 10 meilleurs scores contenu dans le fichier. Nous avons prévu aussi un vidage du fichier pour éviter tous les problèmes de dépassement de capacité du tableau des scores.

4