

28/10/2021

Compte rendu n°2 :

Projet 4A – ILC : Développement Pac-Man en C++

Initialisation du niveau à partir d'un fichier texte :

On commence d'abord par déclarer les variables permettant la déclaration du niveau ; le nombre de colonnes et lignes qui composent le niveau, déclarer la matrice qui contiendra les informations et le fichier texte qui sera ouvert pour lire toutes ces informations.

```
int NbC, NbL;    // Taille du niveau
char** Matrix;
char level[] = "Stage0.txt";
```

Ensuite, on ouvre le fichier, on lit les informations et on les inscrit dans la matrice.

Dans le fichier teste, les 0 représentent les murs, et les 1 représentent les espaces vides. Par la suite, ces 1 seront remplacé par les points, les fruits et points d'apparition des personnages.

```

void OpenLvl(char* Stage1)
{
    // Instruction d'ouverture du niveau
    ifstream lvlFile; // lvlFile est un objet de type ifstream
    lvlFile.open(Stage1); // Ouverture du fichier en lecture seule
    if (lvlFile.good() == false) {
        cout << "Erreur lors de l'ouverture du fichier!" << endl;
        system("pause");
        exit(1);
    }

    lvlFile >> NbC;
    lvlFile >> Nbl;

    // Allocation du tableau du niveau
    Matrix = new char* [NbC];
    for (int i = 0; i < NbC; i++)
        Matrix[i] = new char[Nbl];

    // Initialisation des valeurs du tableau
    for (int i = 0; i < NbC; i++)
        for (int j = 0; j < Nbl; j++)
            Matrix[i][j] = '0';

    // Lecture du tableau
    for (int j = 0; j < Nbl; j++)
        for (int i = 0; i < NbC; i++)
        {
            lvlFile >> Matrix[i][j];
        }

    lvlFile.close();
}

```

Affichage du niveau :

Pour afficher le niveau, on parcourt la matrice. Par chaque 0 que l'on rencontre, on regarde dans chaque directions pour savoir se qui entoure notre point afin de savoir comment sera dessiné le mur ; si deux 0 sont côte à côte, pas la peine de faire une ligne de séparation entre ces deux points.

```
void DrawLvl() {
    cout << "1";
    for (int i = 0; i < NbC; i++) {
        for (int j = 0; j < NbL; j++) {
            if (Matrix[i][j] == '0') {
                cout << "2";
                glColor3d(0, 0, 1);
                glBegin(GL_LINES);
                if (Matrix[i][j - 1] != '0' && j != 0) { // Test vers le haut
                    cout << "3";
                    glVertex2f(i, j);
                    glVertex2f(i + 1, j);
                }
                if (Matrix[i][j + 1] != '0' && j < NbL - 1) { // Test vers le bas
                    glVertex2f(i + 1, j + 1);
                    glVertex2f(i, j + 1);
                }
                if (i >= 1) { // Test vers la gauche
                    if (Matrix[i - 1][j] != '0') {
                        glVertex2f(i, j);
                        glVertex2f(i, j + 1);
                    }
                }
                if (i < NbC - 1) { // Test vers la droite
                    if (Matrix[i + 1][j] != '0') {
                        glVertex2f(i + 1, j + 1);
                        glVertex2f(i + 1, j);
                    }
                }
            }
            glEnd();
        }
    }
}
```

Rendu lors de l'exécution :

