Clément
GRENOT
4A-IT-ILC

25/11/2021

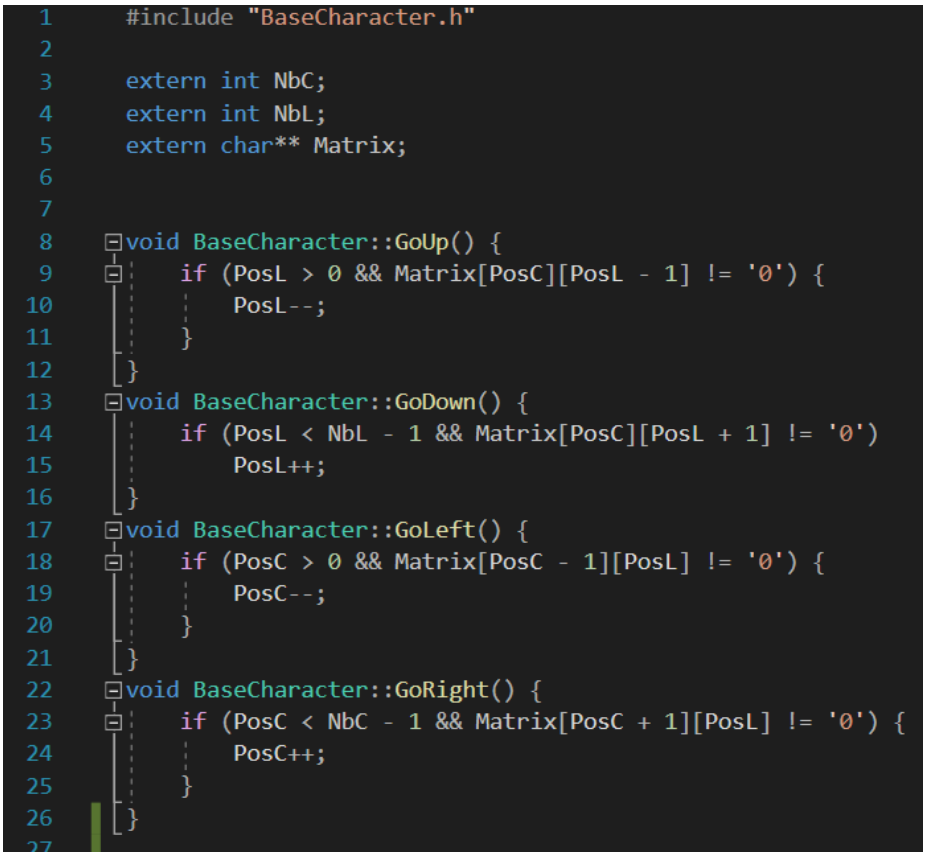
Compte rendu n°3 :

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

Déclaration de la classe 'BaseCharacter' qui aura pour attributs deux coordonnées C et L pour 'Column' et 'Line'. Ils serviront à localiser et déplacer nos personnages sur la grille. Celle-ci permet aussi de définir les conditions de déplacement dont hériterons les fantômes et le personnage PacMan (ne peux pas sortir de la grille et ne peux pas traverser les murs).

```cpp
#include <iostream>
#include "glut.h"

using namespace std;

class BaseCharacter
{
protected:
    // Position Colonne et Ligne sur la matrice
    int PosC, PosL;
public:
    // Accesseurs
    int GetPosC() const { return PosC; }
    int GetPosL() const { return PosL; }

    // Mutateurs
    void SetPosC(int C) { PosC = C; }
    void SetPosL(int L) { PosL = L; }

    // Constructeur
    BaseCharacter() { PosC = PosL = 0; }

    // Fonction de déplacement
    void GoUp();
    void GoDown();
    void GoLeft();
    void GoRight();
};
```

Déclaration BaseCharacter.h



```cpp
#include "BaseCharacter.h"

extern int NbC;
extern int NbL;
extern char** Matrix;


void BaseCharacter::GoUp() {
    if (PosL > 0 && Matrix[PosC][PosL - 1] != '0') {
        PosL--;
    }
}
void BaseCharacter::GoDown() {
    if (PosL < NbL - 1 && Matrix[PosC][PosL + 1] != '0')
        PosL++;
}
void BaseCharacter::GoLeft() {
    if (PosC > 0 && Matrix[PosC - 1][PosL] != '0') {
        PosC--;
    }
}
void BaseCharacter::GoRight() {
    if (PosC < NbC - 1 && Matrix[PosC + 1][PosL] != '0') {
        PosC++;
    }
}

```

Déclaration BaseCharacter.cpp

Déclaration de la classe ‘Hero’ qui hérite de la classe ‘BaseCharacter’. On intègre uniquement un constructeur qui reprend le constructeur de la classe mère et une méthode permettant de le dessiner, que l’on définit par la suite dans le fichier ‘.cpp’.

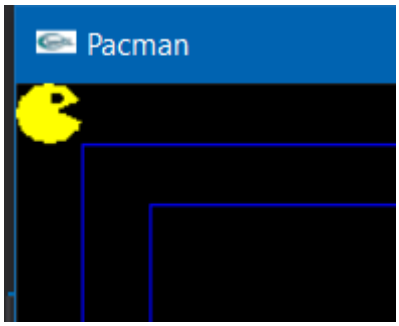
```cpp
#include "BaseCharacter.h"

class Hero : public BaseCharacter
{
public:
    // Appel du constructeur de la classe mère PersonnageBase
    Hero() : BaseCharacter() {}

    void Draw();
};
```

Déclaration Hero.h

```cpp
#include "Hero.h"

using namespace std;
extern char** Matrix;

void Hero::Draw()
{
    glPushMatrix();
    glTranslated(PosC + 0.5, PosL + 0.5, 0.0);
    glColor3d(1, 1, 0); //Couleur jaune
    glutSolidSphere(0.5, 12, 12); // Sphère de la tête

    glColor3d(0, 0, 0); //Couleur noire
    glTranslated(0.1, -0.25, 0.0);
    glutSolidSphere(0.1, 12, 12); // Oeil

    glBegin(GL_TRIANGLES);  //Création de la bouche
    glColor3f(0, 0, 0);
    glVertex3f(0, 0.2, 0);
    glVertex3f(0.4, 0, 0);
    glVertex3f(0.4, 0.4, 0);
    glEnd();

    glPopMatrix();
}
```

Déclaration Hero.cpp

On commence par définir le plan de l’image. Avec la bibliothèque ‘Glut’, nous devons définir la couleur de la forme générée avant d’indiquer quel seront les paramètres de la forme souhaités. C’est pour cela qu’avant chaque nouvelle forme, j’indique quelle couleur sera utilisée. De plus, les dessins se font les uns après les autres Nous commençons donc par la boule ronde et jaune caractéristique de pacman, puis par-dessus, le cercle noir de son œil et enfin, encore par-dessus le triangle noir de sa bouche. D’où l’importance de l’ordre d’exécution des instructions.



Résultat lors de l'exécution

Nous avons donc déclaré une classe permettant de définir les paramètres basiques d'un personnage et déclaré la définition de la classe de personnage principale ; PacMan, que nous arrivons a dessiner sur notre interface.