C 语言报告

@author Wangjikang 16029110064 (Deux premiers chiffres est 16)

ChenXiangru 17029110056

一、程序操作说明

1，在程序运行初始阶段，我们的程序将要求您选择游戏难度（四个），和游戏模式（鼠标模式和键盘模式），

2，选择不同难度时程的题，程序会读取相应的题目。

3，选择不同模式时会展开不同的操作界面，以及相应模式下的功能和按钮。

4、游戏中想要修改游戏难度或者模式，请选择 Selectionner la difficulte，

二，游戏模式说明

模式一：

1，此模式允许您使用键盘进行输入。

2，在此模式下，程序不会在您填写的过程中对所填写的数字是否是正确答案进行检验，直到您点击位于界面下方的 Verification Finale 按钮与 Verification Intermediaire 按钮进行检验。

3，此模式下的按钮及功能：Verification finale 最终检查，检查方格是否填写完毕且全部正确；Verification Intermediaire 过程检查，检查是否符合游戏规则(行，列以及九宫格无重复数字)；Quit 结束并退出程序；Selectionner la difficulte 重新选择难度与模式；Relever la solution 展示当前题目的答案。

模式二：

1、此模式允许您只利用鼠标来操作，点击您想要填入的网格，利用弹出的键盘选择要填入的数字。

2、在此模式下，您每填入一次，都会进行检查（行，列以及九宫格的判断），如果不满足游戏规则，则会弹出错误窗口并删除所填数字。所有网格填写完毕，则提示游戏结束。

3、此模式下的按钮：quit 结束并退出程序；Selectionner la difficulte 选择难度与模式；Relever la solution(展示答案)。

三、算法

1 思路：通过一个 9*9（81）的 ValeurSudoku 型数组进行数据传输，数组的每一个元素都代表数独游戏的一个网格，数组角标代表这个网格的坐标。创建 81 个文本显示区（例如模式一的 MakeTextWidget，模式二的 MakeDrawArea），用于显示文本。81 个 valeur 变量用来储存网格的值，而 callback 函数只能传递一个数据，所以我们命名一个结构体 ValeurSudoku 来

作为数据进行传输。

2.数据结构体说明 ：

```
typedef struct grille{
   Widget ZoneAffich; // zone d'affichage ，数据显示区
   char *valeur;          // le valeur de grille   此网格的数据
   enum{constant, variable}   nature;          // Si la valeur peut  être changée
   int x,y;                  // Coordonnées de la grille  网格的坐标
} ValeurSudoku;
```

每一个 ValeurSudoku 型变量都代表数独游戏中的其中一个网格，ZoneAffich 是这个网格的显示区，valeur 是这个网格中的数字，nature 是判断这个网格是否可以被用户操作，x,y 是这个网格的坐标（模式二中需要使用）。

4.文件读取 函数 initSudoku()：数独题目文件内为数独的题目，0 代表无数据。打开文件，使用 fscanf(fp,"%s",s)进行文件内数据的读取，每次读取一个数字（以字符串的形式），并用 strcpy（d[i*N+j].valeur, s）进行赋值。如果读取的数字是 0，则此数字可被改变（d[i*N+j].nature = variable;）否则不可被改变（d[i*N+j].nature = constant;），并对 x,y 赋值，

5.难度和模式的选择和运行：Selectionner la difficulte 按钮的 callback 函数 niveauDeDifficile(Widget w, void *data)会调用 difficulte()函数来创建选择难度的界面，选择难度的按钮的 callback 函数会调用 initSudoku()，将相应题目从文件读入程序，并调用 choosemode()创建选择模式的界面，两个模式的按钮 mode1 和 mode2 按钮的 callback 函数会调用相应的 makeZoneAffich1()，makeDisplay1()affichSudoku1()，makeZoneAffich2()，makeDisplay2()创建游戏界面。

6.程序刚运行时的选择难度窗口：程序运行后，直接在 init_display 函数中调用 difficulte()函数；

7.数据检查算法 函数 valeurValide()：分别检查相应网格所在的行，列，和大网格中是否有重复的数字，因为 valeur 是字符串型，所以我们先检查所读取的数据是否是单个字符，是否是一个数字，然后进行数独游戏规则判断。

8.答案计算 函数 solution()：用链表储存每个可填写的网格，每个网格的链表的值从 1 开始判断，如果符合游戏规则，则下一个网格从 1 开始判断。如果当前链表元素超过 9，则出栈，从上一个继续往下判断。最后一个网格计算完毕，就是最终答案。

9.答案显示 函数 reverler()：创建窗口，使用 text_edit 制作显示区，调用 solution()计算答案后，使用 affichSudoku1()将答案显示出来