

小黄和它的罐子(1)

2022秋程序设计 Project1

复旦大学2022级 傅全通

Outline

- 分析问题
- 流程图
- 伪代码
- 从伪代码到代码（示例）
- 细节与补充



这正是这张有名的图片所指示的思路

分析问题

- 一共要做哪些事？
- 初始化棋盘
- 处理用户输入，
并对其做出正确反应

P.S. 以上都离不开
对数据的合理存储及正确运算。

```
C:\Users\fqt15\Documents\程序设计Project\PJ1Handin_22307110118傅全通\PJ1.exe
# # # # # # # # # # # #
#          @          @          @ #
#          @ @ @          @ @          #
# @ @          @ @          @ @          #
# @          @ @          @ @          @ #
#          @ @ ! @          @          @ #
#          @          @ @          @ #
# @ @          @          @          @ #
#          @          @ @          @ #
# @ @ @ @ @          @          @ #
# @ @          @          @ @          #
# # # # # # # # # # # #

*****
*                                     *
*      score:          |      commmand:  pick      *
*              20      |                               *
*                                     *
*****
Type "help" for a helping document,
'ctrl+Z' or "exit" to exit.
>>>
```

流程图

- 当不知从何下手时，请回想这一原则：
- “自顶向下，逐步求精”
- 画流程图以帮助梳理思路。

例：手绘一个N-S流程图

初始化地图、小黄坐标

输入命令 ≠ “退出”

命令合法？

是

否

执行命令

更新分数

提示

非法

结算并给出最终分数

Talk is cheap,
show the code.

伪代码

```
1  int main(int argc, char **argv){
2      initialize();           // 伪函数, 在init.c
3
4      char cmd[512];          // 存储用户输入的指令
5      while (scanf("%s", cmd) != EOF && !strcmp(strlwr(cmd), "exit")){
6          if (!strcmp(cmd, "help")){
7              help();          // 显示帮助文档
8              continue;
9          }
10         if (!isdigit(*cmd) || *cmd > '6'){
11             invalid();        // 提示非法输入
12             continue;
13         }
14         move(*cmd ^ '0');     // 确定了*cmd是个数字, 将其从ASCII变为数码
15     }
16     puts("Final Score:");
17     puts("The game has ended! Now you can check your final score.");
18     system("pause");
19     return 0;
20 }
```

从伪代码到代码

(例:)

bool move(int cmd)

参数：一个整数，即命令

返回值：表示罐子捡完或仍有剩余

基本功能：

根据输入的命令，
操纵小黄移动（或捡罐子）
并实时更新分数

附加功能：回显（echo）输入的指令

```
14 bool move(int cmd) // return ture == cans has all been picked up.
15 {
16     if (cmd == 0)
17         cmd = rand() % 6 + 1;
18     MoveCursorTo(31, 16); // show commands.
19     switch (cmd)
20     {
21     case up:
22         puts("    up  ");
23         if (pos.y == 1)
24             score -= 5;
25         else
26         {
27             MoveCursorTo(pos.x * width, pos.y);
28             putchar(map[pos.x][pos.y]);
29             --pos.y;
30             MoveCursorTo(pos.x * width, pos.y);
31             putchar('!');
32         }
33         break;
34     case down:
35         puts("  down  ");
36         if (pos.y == 10)
37             score -= 5;
38         else
39         {
40             MoveCursorTo(pos.x * width, pos.y);
41             putchar(map[pos.x][pos.y]);
42             ++pos.y;
43             MoveCursorTo(pos.x * width, pos.y);
44             putchar('!');
45         }
46         break;
47     }
48     MoveCursorTo(11, 17);
49     printf("%4d", score);
50     return false;
51 }
```

细节与补充 (□述, 必要性递减)

- 必会!**
 - 多文件的组织形式 (*.h、*.c、**extern**、**static**、声明和实现分开)
 - 妙用: 库函数、enum、逗号表达式、短路运算符 (可举assert宏说明其被广泛采用)
- 素养**
 - 命名风格 (常量名、变量名、函数名、宏名; 驼峰命名等)
 - 良好代码习惯 (少“压行”、写好注释、尽可能缩小变量作用域)
 - C99以及更新的语言规范 (**inline**、**bool**、VLA、**for**内初始化)
- 美德**
 - 高性能 (不强求, 可能使可读性下降; 而且编译器也会帮助优化)
 - 可移植性 (仅做科普, 简述条件编译及其带来的同函数不同实现)

Review 分别对应Project实现的哪一部分？

数据与运算

常量、变量

输入、输出

表达式、运算

结构化程序设计

顺序结构

选择结构

循环结构

数组

一维数组

多维数组

字符数组

函数

声明和实现

库函数

函数调用

Review：数据与运算

常量	设置命令表（字符串字面量）来比对命令
----	--------------------

变量	合理存储罐子位置和数量、小黄位置和分数
----	---------------------

输入	正确从标准输入流得到用户输入的命令
----	-------------------

输出	对齐打印棋盘、指定宽度输出分数
----	-----------------

表达式	正确使用逗号、分号、括号
-----	--------------

运算	妙用位运算和逻辑运算（尤其短路运算）
----	--------------------

Review：数组和函数

一维数组	数组作为参数传递；数组名和首地址的对应
二维数组	理解其内存分布、首地址和每行首地址等等
字符数组	小心数据溢出；学会用库函数处理字符串
函数声明	在头文件内声明函数原型，必要时取有意义的参数名
函数实现	取得传入的参数或外部的变量；善用 return 跳转语句
库函数、标准库之外的API	

感谢倾听！

最后安利一下程序设计[校赛](#)（参与即有奖！）和个人[博客](#)