

# 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  
(有谁认出来吗?)

# 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

(有意“别扭”，  
凡困惑之处  
都可以问)

```
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")){           返回 char* 型
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;   #define EXIT_SUCCESS 0
20 }
```

main的右括号自动 return 0, 但还请显式写上

# 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) [0,6), 加一才是 1, 2, 3, 4, 5, 6}
17         cmd = rand() % 6 + 1; 简化剩余系
18     MoveCursorTo(31, 16); // show commands.
19     switch (cmd) 顾名思义, 它是 windows 下移动光标的
20     {
21     case up: 合法唷!
22         puts(" up ");
23         if (pos.y == 1)
24             score -= 5;
25         else
26         {
27             MoveCursorTo(pos.x * width, pos.y); 补上该有的
28             putchar(map[pos.x][pos.y]);
29             --pos.y; 这一行放进上一行就称“压行”
30             MoveCursorTo(pos.x * width, pos.y);
31             putchar('!'); 把小黄放到这里
32         }
33         break;
34     case down:
35         puts(" down ");
36         if (pos.y == 10)
37             score -= 5;
38         else
39             MoveCursorTo(11, 17);
40         printf("%4d", score);
41         return false; 非 void 函数切忌 return 缺省值
42     }
43 }
```

# 细节与补充 (□述, 必要性递减)

- 必会!**
  - 多文件的组织形式 (cmake 和 直接编译不同) (\*.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)  
赋予 signed int 以意义
- 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) static
  - C99以及更新的语言规范 (**inline**、**bool**、VLA、**for**内初始化)  
(auto/栈/尤其用于结构体最后一个元素)
- 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) -O2
  - 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)  
GNU倡议: 用 #if 代 #ifdef, 便于串联 #elif 等等好处



额外但实用的功能：(用 `strstr()`, `strtok()`, `strchr()` 等)

-help → 提供帮助文档

-wasd → 改由 W, A, S, D 控制, 且无需敲回车

-setw <num> 将 '#' 间横向距离设为  $(num-1)$ , num 也行, 看实现咯

甚至设计改变各动作得分的(接口) interface,

(Pj 1 中这看似小题大做, 但 Pj 2 里就有用了)

• `ch^'0'` 的解释:  $48 = 110000_6$ ,  $57 = 111001_6$ ,  $9 < 2^4 - 1$

↓ minecraft 中据此实现半加器

• java 整了个 Project 2, 想看可以稍演示.

# 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](#)

荐书，不具体荐，“开卷有益”；尤其读英文原版。