Lab4 实验报告

PB19151769 马宇骁

算法部分：

由实验 4 要求，考虑设计三个字符串，通过读取地址指针所在位置进行输出,设计 R4 作为 counter，每有一行在游戏过程中被取完就+1，并做一次判断，如果 R4 等于 3，则游戏结束，根据 R0 判断当前最一次操作由谁来完成，0 为 player2,1 为 player1，谁完成对方赢。由此思路编写算法如下。

编写部分：

根据书中学过的汇编语言，依据算法写出如下代码:

```
    .ORIG x3000
Newline     .FILL x000A
SAVE        .BLKW #1
SaveR0      .BLKW #1
SaveR00     .FILL #0
SAVE0       .FILL x0000
ASCIIA      .FILL #65
LastA       .BLKW #1
LastB       .BLKW #1
LastC       .BLKW #1

AND R4,R4,#0
AND R5,R5,#0
ADD R5,R5,#3
ST R5,LastA       ;ROW A 最初 3 个球
ADD R5,R5,#2
ST R5,LastB       ;ROW B 最初 5 个球
ADD R5,R5,#3
ST R5,LastC       ;ROW C 最初 8 个球

CHECKA  LEA R2,Prompt1
        LDR R3,R2,#6
        BRnp START
        ADD R4,R4,#1
        NOT R3,R4
        ADD R3,R3,#4
        BRz Judge
        BRnzp START

CHECKB  LEA R2,Prompt2
        LDR R3,R2,#6
        BRnp START
```

```
            ADD R4,R4,#1
            NOT R3,R4
            ADD R3,R3,#4 ;在这里判断 R4 是否为 3
            BRz Judge
            BRnzp START

CHECKC    LEA R2,Prompt3
            LDR R3,R2,#6
            BRnp START
            ADD R4,R4,#1
            NOT R3,R4
            ADD R3,R3,#4
            BRz Judge
            BRnzp START

START     LEA R0,Prompt1
            JSR Init
            LEA R0,Prompt2
            JSR Init
            LEA R0,Prompt3
            JSR Init

           LD R0,SaveR00   ;直接赋值
           NOT R0,R0
           ADD R0,R0,#1
           ADD R0,R0,#1
           ST R0,SaveR00
again    LD R0,SaveR00
           ADD R0,R0,#0
           BRp player1
           BRz player2
player1    LEA R0,Prompt4
            BRnzp L
player2     LEA R0,Prompt5
            BRnzp L
L          ST R0,SaveR0
           LDR R0,R0,#0
           BRz LOOP
           OUT
           LD R0,SaveR0
           ADD R0,R0,#1
           BRnP L
LOOP      GETC
           OUT
```

```
        AND R1,R1,#0
        ADD R1,R1,R0     ;R1 记录字母
        GETC
        ST R0,SAVE      ;SAVE 记录数字
        OUT
        LD R0,Newline
        OUT
        LD R0,SAVE
        LD R2,ASCIIA
        NOT R2,R2
        ADD R2,R2,#1
        ADD R2,R2,R1
        BRn fail     ;字母不符合要求
        BRz A        ;字母符合要求（下同）
        ADD R3,R2,#-1
        BRz B
        ADD R3,R2,#-2
        BRz C
        BRp fail

A       LEA R2,Prompt1
        LDR R5,R2,#6
        BRz fail
        ADD R1,R0,#-15       ;1->49 2->50 3->51
        ADD R1,R1,#-15
        ADD R1,R1,#-15
        ADD R1,R1,#-3
        BRnz fail
        ADD R3,R1,#0
        NOT R3,R3
        ADD R3,R3,#1
        LD R5,LastA
        ADD R3,R3,R5    ;R5-R3>=0
        BRn fail
        LEA R2,Prompt1
        NOT R1,R1
        ADD R1,R1,#1
        ADD R5,R5,R1    ;剩余球数
        ST    R5,LastA
        ADD R5,R5,#6
        ADD R2,R2,R5
        LD R0,SAVE0
        STR R0,R2,#0       ;打印 0
        BRnzp CHECKA
```

```
B       LEA R2,Prompt2
        LDR R5,R2,#6
        BRz fail
        ADD R1,R0,#-15
        ADD R1,R1,#-15
        ADD R1,R1,#-15
        ADD R1,R1,#-3
        BRnz fail
        ADD R3,R1,#0
        NOT R3,R3
        ADD R3,R3,#1
        LD R5,LastB
        ADD R3,R3,R5
        BRn fail
        LEA R2,Prompt2
        NOT R1,R1
        ADD R1,R1,#1
        ADD R5,R5,R1
        ST R5,LastB
        ADD R5,R5,#6
        ADD R2,R2,R5
        LD R0,SAVE0
        STR R0,R2,#0
        BRnzp CHECKB

C       LEA R2,Prompt3
        LDR R5,R2,#6
        BRz fail
        ADD R1,R0,#-15
        ADD R1,R1,#-15
        ADD R1,R1,#-15
        ADD R1,R1,#-3
        BRnz fail
        ADD R3,R1,#0
        NOT R3,R3
        ADD R3,R3,#1
        LD R5,LastC
        ADD R3,R3,R5
        BRn fail
        LEA R2,Prompt3
        NOT R1,R1
        ADD R1,R1,#1
        ADD R5,R5,R1
```

```
            ST R5,LastC
            ADD R5,R5,#6
            ADD R2,R2,R5
            LD R0,SAVE0
            STR R0,R2,#0
            BRnzp CHECKC

Judge    LD R0,SaveR00
            ADD R0,R0,#0
            BRp    Player_2_Wins
            BRz    Player_1_Wins
Player_2_Wins      LEA R0,Prompt8
                        JSR Init
                        HALT
Player_1_Wins      LEA R0,Prompt7
                        JSR Init
                        HALT

Init ST R0,SaveR0
        LDR R0,R0,#0
        BRz OK
        OUT
        LD R0,SaveR0
        ADD R0,R0,#1
        BRnp Init
OK      LD R0,Newline
        OUT
        RET

fail     LEA R0,Prompt6
            JSR Init
            BRnzp again

Prompt1 .STRINGZ "ROW A:ooo"
Prompt2 .STRINGZ "ROW B:ooooo"
Prompt3 .STRINGZ "ROW C:oooooooo"
Prompt4 .STRINGZ "Player1, choose a row and number of rocks:"
Prompt5 .STRINGZ "Player2, choose a row and number of rocks:"
Prompt6 .STRINGZ "Invalid move. Try again."
Prompt7 .STRINGZ "Player1 Wins."
Prompt8 .STRINGZ "Player2 Wins."

       .END
```

测试部分：
第一次：

```
ROW A:ooo
ROW B:ooooo
ROW C:oooooooo
Player1, choose a row and number of rocks:a1
Invalid move. Try again.
Player1, choose a row and number of rocks:A1
ROW A:oo
ROW B:ooooo
ROW C:oooooooo
Player2, choose a row and number of rocks:A2
ROW A:
ROW B:ooooo
ROW C:oooooooo
Player1, choose a row and number of rocks:A1
Invalid move. Try again.
Player1, choose a row and number of rocks:B3
ROW A:
ROW B:oo
ROW C:oooooooo
Player2, choose a row and number of rocks:B5
Invalid move. Try again.
Player2, choose a row and number of rocks:B2
ROW A:
ROW B:
ROW C:oooooooo
Player1, choose a row and number of rocks:C8
Player2 Wins.


--- Halting the LC-3 ---
```

第二次：

```
ROW A:ooo
ROW B:ooooo
ROW C:oooooooo
Player1, choose a row and number of rocks:A1
ROW A:oo
ROW B:ooooo
ROW C:oooooooo
Player2, choose a row and number of rocks:
A
Invalid move. Try again.
Player2, choose a row and number of rocks:2

Invalid move. Try again.
Player2, choose a row and number of rocks:A2
ROW A:
ROW B:ooooo
ROW C:oooooooo
Player1, choose a row and number of rocks:B5
ROW A:
ROW B:
ROW C:oooooooo
Player2, choose a row and number of rocks:C8
Player1 Wins.


--- Halting the LC-3 ---

□
```

测试无误，代码正确！