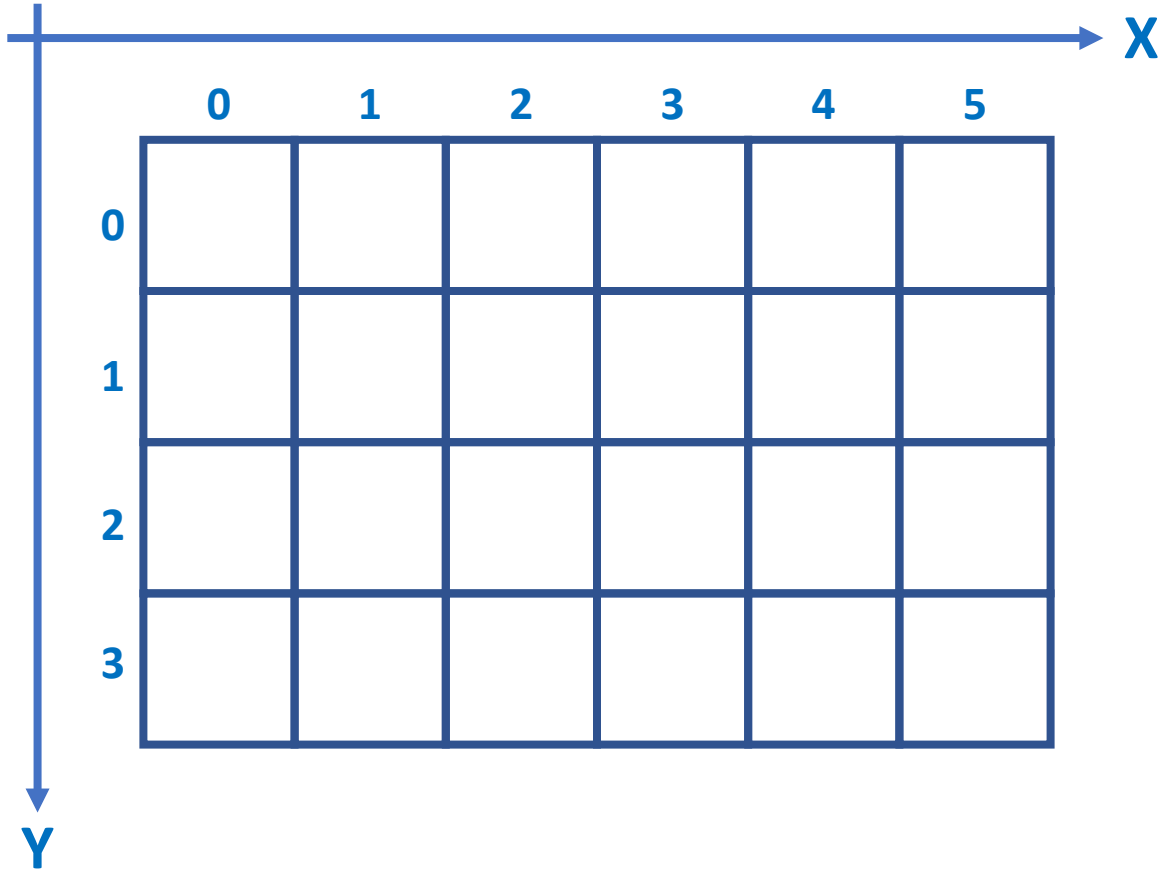
Trabalho com o Cesar 2019/2 Parte 1 (aplicativo)

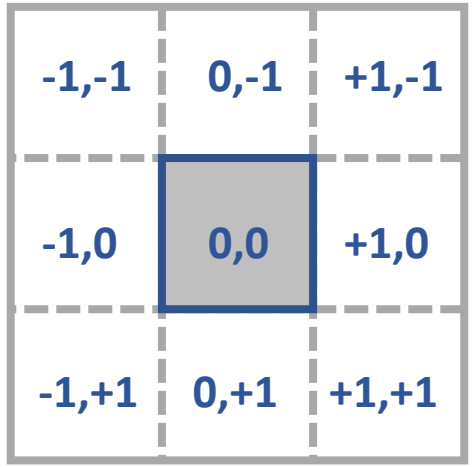
- Conceito de parede e coordenadas
- Câmera e movimentação
- Exemplos de uso das funções

Nota: leia o enunciado antes de olhar estes slides ;-)

Conceito de parede – exemplo: 6 x 4

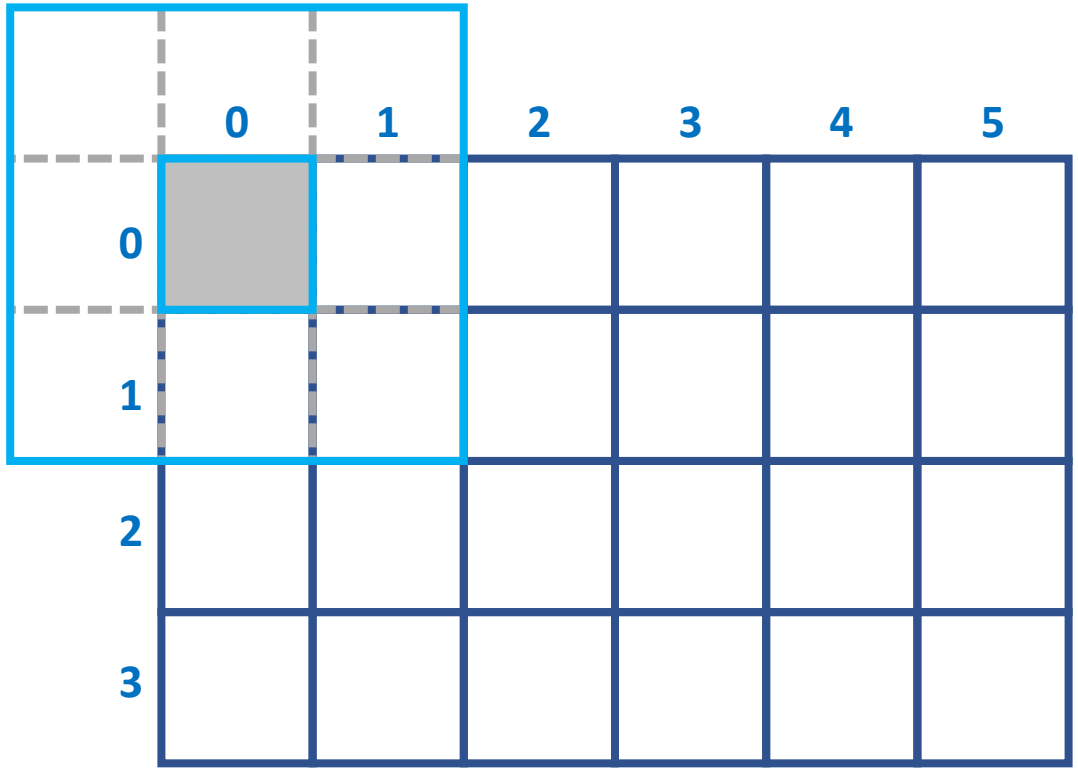


Conceito de câmera
(com as coordenadas relativas para **`read`**)



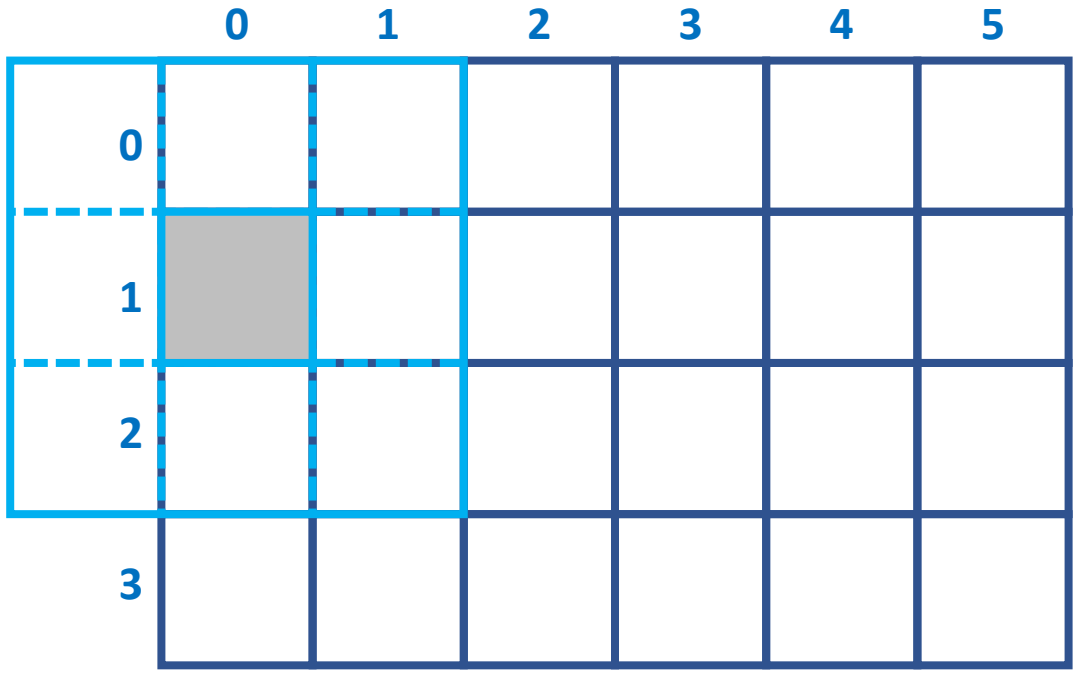
- a câmera sempre "vê" 9 ladrilhos, mas cada vez que a função **`read`** é chamada ela devolve apenas a informação sobre um ladrilho
- os valores devolvidos pela função **`read`** são: 0 (vazio), 1 (bom), 2 (danificado) ou 3 para coordenadas fora da parede – por exemplo: **`read`** (-1,0) quando a câmera estiver na coordenada (0,0) da parede
- as coordenadas da câmera sempre correspondem às coordenadas na parede do ladrilho central (cinza), mas as coordenadas passadas para a função **`read`** devem ser sempre relativas ao ladrilho central (as mostradas na figura acima)
- a câmera nunca se desloca para fora da parede; a função **`move`** "trava" a câmera se isto for tentado

Conceito de câmera – posição após **`move00`**



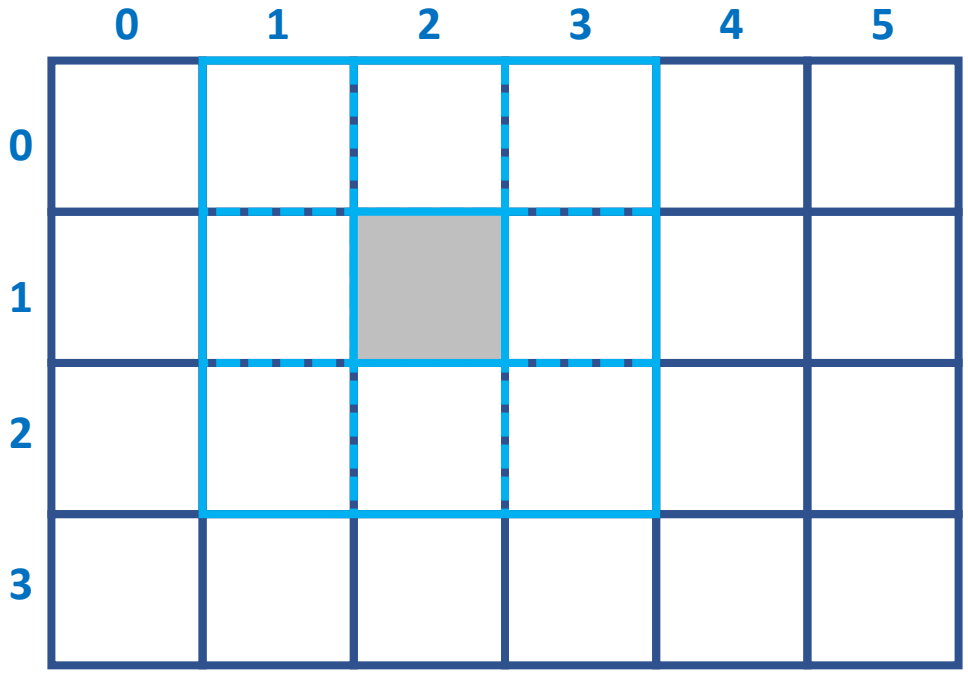
Nota: Uma chamada para a função `read` com parâmetros (-1,y) ou (x,-1) quando a câmera estiver nesta posição vai devolver o valor 3 (fora da parede)

Câmera – posição após **`move (0,+1)`**



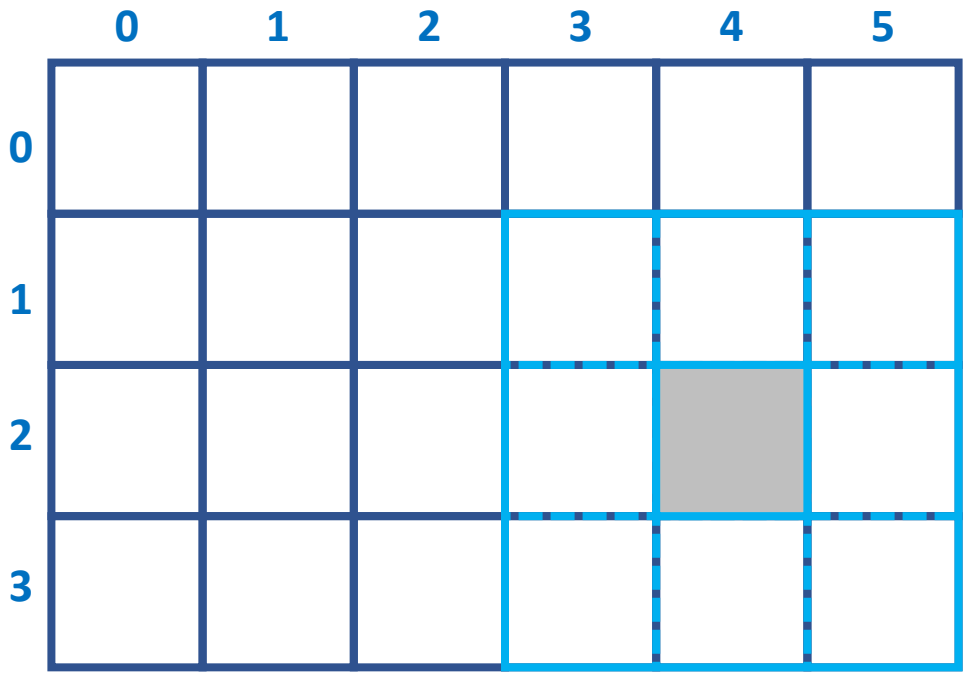
Tempo de movimentação: 150 ms (100 + 1x50)

Câmera – posição após **`move (+2,0)`**



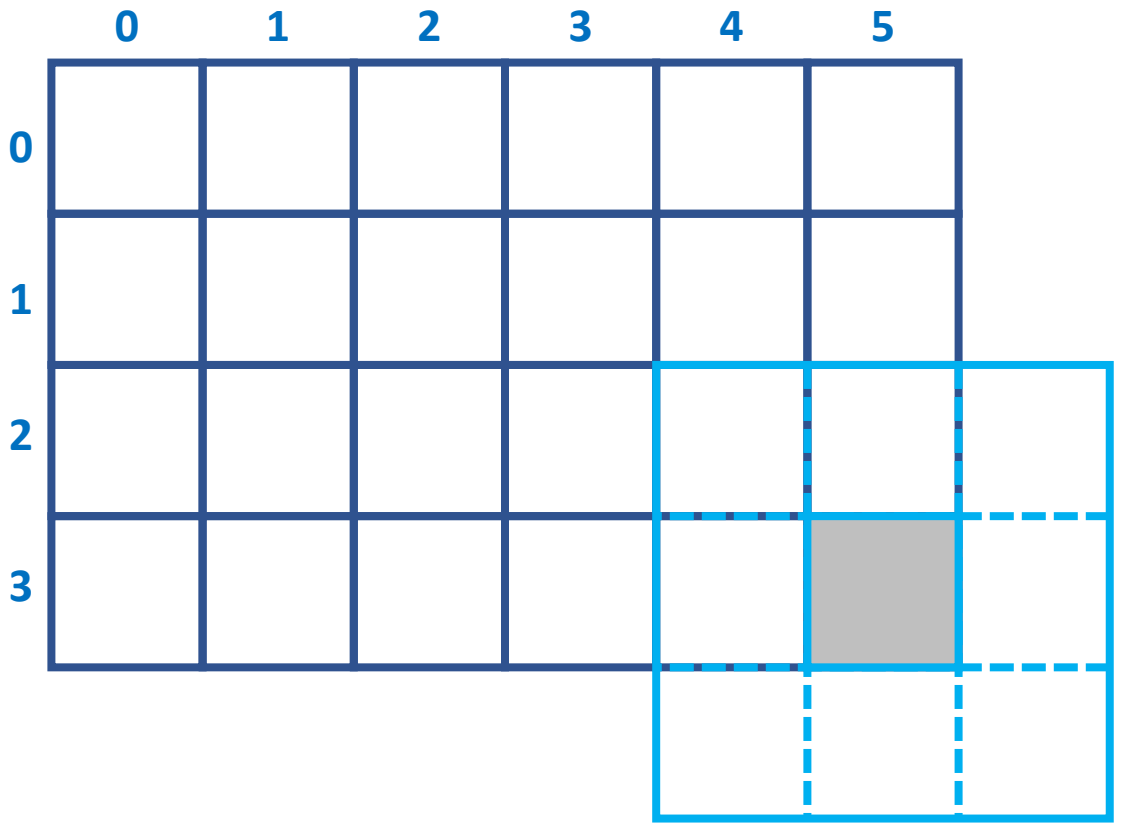
Tempo de movimentação: 200 ms (100 + 2x50)

Câmera – posição após **`move (+2,+1)`**



Tempo de movimentação: 250 ms (100 + 3x50)

Câmera – posição após **`move (+1,+2)`**



Nota: `move` “travou” a câmera para não sair fora da parede (como `move (+1,+1)`)

Funções de entrada e saída

ENTRADA (TECLADO)

- `kbhit` – informa se foi pressionada alguma tecla, sem ler o caractere; não espera que alguma tecla seja pressionada
- `getchar` – obtém o código ASCII do caractere digitado no teclado; se nada foi digitado, continua esperando sem voltar ao programa chamador

SAÍDA (VISOR)

- `setcursor` – move o cursor para a posição do visor especificada (0 a 35)
- `putchar` – escreve um caractere no visor, na posição onde está o cursor e avança o cursor para a posição seguinte; se escrever no último LED, o cursor desaparece
- `putmsg` – escreve uma mensagem no visor, a partir da posição onde está o cursor, e avança o cursor para a primeira posição após a mensagem escrita; se a mensagem for mais longa do que o espaço disponível até o fim do visor, ela é truncada e o cursor desaparece; a mensagem deve ser uma sequência de bytes com códigos ASCII, terminada por um byte com o valor 0

Funções específicas para a aplicação

`Informações sobre a parede e ladrilhos`

- **`setwall`** – informa a largura e a altura da parede passada como parâmetro
- **`read`** – devolve o estado do ladrilho cujas coordenadas (relativas ao centro da câmera) foram passadas como parâmetro

Controle do movimento da câmera

- **`move00`** – move a câmera para as coordenadas (0,0) da parede (aponta o centro da câmera para o ladrilho no canto superior esquerdo da parede)
- **`move`** – movimenta a câmera conforme os parâmetros recebidos; pode mover por 1 ou dois ladrilhos, nas 4 direções; se o centro da câmera chegar num limite da parede, a movimentação é interrompida; só pode ser chamada se a câmera estiver parada
- **`moving`** – informa se a câmera está parada ou em movimento – deve ser usada antes de chamar as funções **`move00`**, **`move`**, **`read`** ou **`getcamera`**
- **`getcamera`** – informa as coordenadas atuais da câmera

Exemplo de chamada: kbhit

```
; r0 = _SISTEMA(kbhit)
; if (r0!=0) {
;    // tecla pressionada
; }

    mov  #kbhit,r5      ;código da função no R5
    jsr  r7,_SISTEMA

    tst  r0
    beq  nenhuma_tecla
    ; tecla pressionada
nenhuma_tecla:
```

Nota: kbhit e _SISTEMA definidos por EQU no fonte fornecido

Exemplo de chamada: getchar

```
; r0 = _SISTEMA(getchar)

        mov  #getchar,r5
        jsr  r7,_SISTEMA
```

Exemplo de chamada: setcursor

```
; _SISTEMA(setcursor, 35)
; _SISTEMA(setcursor, 0)

    mov  #35,r0    ;coordenada para posicionar o cursor
    mov  #setcursor,r5
    jsr  r7,_SISTEMA

    clr  R0        ;coordenada para posicionar o cursor
    mov  #setcursor,r5
    jsr  r7,_SISTEMA
```

Exemplo de chamada: putchar

```
; _SISTEMA(putchar, ‘A’)

    mov  #’A’,r0
    mov  #putchar,r5
    jsr       r7,_SISTEMA

; Escreve a letra ‘A’ na posição do cursor
; O cursor avança para a próxima posição
; Se escreveu na posição 35, o cursor desaparece
```

Exemplo de chamada: putmsg

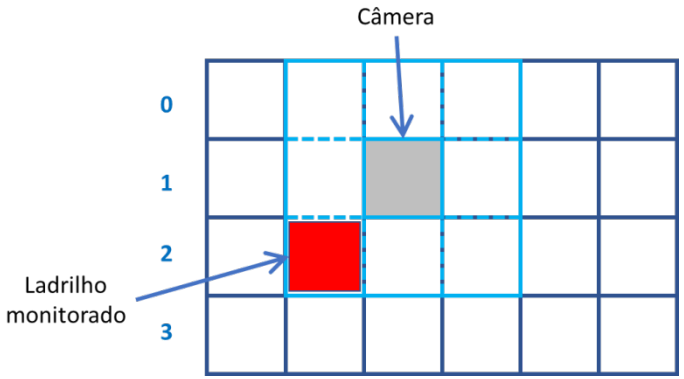
```
; _SISTEMA(putmsg, mensagem)

    mov  #mensagem,r0
    mov  #putmsg,r5
    jsr  r7,_SISTEMA

; Mensagem é escrita no visor, a partir da posição do cursor
; O cursor avança até o final da mensagem
; Se escreveu na posição 35, trunca a mensagem
; Se escreveu na posição 35, o cursor desaparece
; Exemplo de mensagem
mensagem: dab ‘Texto da mensagem’,0
```

Exemplo de chamada: setwall

```
; R0,R1 = _SISTEMA(setwall, 0)

    mov  #0,r0      ; número da parede escolhida
    mov  #setwall,r5
    jsr  r7,_SISTEMA

; devolve LARGURA da parede em R0 (coordenadas X: 0 a L-1)
; devolve ALTURA da parede em R1 (coordenadas Y: 0 a A-1)
; se parede solicitada não existe, devolve R0=R1=-1
```

Exemplo de chamada: read

```
; R0 = _SISTEMA(read, 0, 0)   // Coordenadas ΔX=0, ΔY=0
    mov  #0,r0
    mov  #0,r1
    mov  #read,r5
    jsr  r7,_SISTEMA     ; Retorna 0, 1, 2 ou 3

; R0 = _SISTEMA(read, -1, 1)  // Coordenadas ΔX=-1, ΔY=1
    mov       #-1,r0
    mov       #1,r1
    mov       #read,r5
    jsr       r7,_SISTEMA
```



Exemplo de chamada: moving

```
; R0 = _SISTEMA(moving)  // A câmera está se movendo?

      mov  #moving,r5
      jsr  r7,_SISTEMA

; Exemplo de uso
; while (_SISTEMA(moving)!=0);
espera_parar:
            mov       #moving,r5
            jsr       r7,_SISTEMA
            tst       r0
            bne       espera_parar
```

Exemplo de chamada: move00

```
; _SISTEMA(move00)  // Coordenadas da câmera = (0,0)

    mov  #move00,r5
    jsr  r7,_SISTEMA


; Antes de chamar esta função, verificar se a câmera está
parada
```

Exemplo de chamada: move

```
; _SISTEMA(move, 2, -2)    // move 2 ladrilhos para direita e
para cima
      mov   #2,r0
      mov   #-2,r1
      mov   #move,r5
      jsr   r7,_SISTEMA
; _SISTEMA(move, -1, 0)    // move 1 ladrilho para esquerda
; para mover 1 ladrilho para a esquerda:
      mov   #-1,r0
      mov   #0,r1
      mov   #move,r5
      jsr   r7,_SISTEMA
```

Exemplo de chamada: getcamera

```
; R0,R1 = _SISTEMA(getcamera); // informa coordenadas da
câmera
            mov         #getcamera,r5
            jsr         r7,_SISTEMA

; devolve no R0 a coordenada X
; devolve no R1 a coordenada Y
; se a câmera ainda estiver em movimento, devolve R0=R1=-1
```

Trabalho com o Cesar 2019/2
Parte 1 (aplicativo)