Thực hành KTMT Lab 11

Lê Quốc Đảng

20225801

Assignment 3

* Code

.eqv HEADING 0xffff8010 # Integer: An angle between 0 and 359

# 0 : North (up)

# 90: East (right)

# 180: South (down)

# 270: West (left)

.eqv MOVING 0xffff8050 # Boolean: whether or not to move

.eqv LEAVETRACK 0xffff8020 # Boolean (0 or non-0):

# whether or not to leave a track

.eqv WHEREX 0xffff8030 # Integer: Current x-location of MarsBot

.eqv WHEREY 0xffff8040 # Integer: Current y-location of MarsBot

.text

main:

# Di chuyển ra khỏi mép của màn hình

jal UNTRACK

nop

addi $a0,$zero,90

jal ROTATE

nop

jal GO

nop

sleep: addi $v0,$zero,32 # Keep running by sleeping in 5000 ms

li $a0,5000

syscall

# Vẽ cạnh góc vuông thứ nhất

jal TRACK # and draw new track line

nop

addi $a0, $zero, 180 # Marsbot rotates 180\* and start running

jal ROTATE

nop

jal GO

nop

sleep1: addi $v0,$zero,32 # Keep running by sleeping in 5000 ms

li $a0,5000

syscall

jal UNTRACK # keep old track

nop

# Vẽ cạnh góc vuông thứ 2

jal TRACK

nop

addi $a0, $zero, 90 # Marsbot rotates 90\*

jal ROTATE

nop

sleep2: addi $v0,$zero,32 # Keep running by sleeping in 5000 ms

li $a0,5000

syscall

jal UNTRACK # keep old track

nop

# Vẽ cạnh huyền

jal TRACK # and draw new track line

nop

addi $a0, $zero, 315 # Marsbot rotates 315\*

jal ROTATE

nop

sleep3: addi $v0,$zero,32 # Keep running by sleeping in 7071 ms

li $a0,7071

syscall

jal UNTRACK # keep old track

nop

end\_main: li $v0, 10

syscall

#-----------------------------------------------------------

# GO procedure, to start running

# param[in] none

#-----------------------------------------------------------

GO: li $at, MOVING # change MOVING port

addi $k0, $zero,1 # to logic 1,

sb $k0, 0($at) # to start running

nop

jr $ra

nop

#-----------------------------------------------------------

# STOP procedure, to stop running

# param[in] none

#-----------------------------------------------------------

STOP: li $at, MOVING # change MOVING port to 0

sb $zero, 0($at) # to stop

nop

jr $ra

nop

#-----------------------------------------------------------

# TRACK procedure, to start drawing line

# param[in] none

#-----------------------------------------------------------

TRACK: li $at, LEAVETRACK # change LEAVETRACK port

addi $k0, $zero,1 # to logic 1,

sb $k0, 0($at) # to start tracking

nop

jr $ra

nop

#-----------------------------------------------------------

# UNTRACK procedure, to stop drawing line

# param[in] none

#-----------------------------------------------------------

UNTRACK:li $at, LEAVETRACK # change LEAVETRACK port to 0

sb $zero, 0($at) # to stop drawing tail

nop

jr $ra

nop

#-----------------------------------------------------------

# ROTATE procedure, to rotate the robot

# param[in] $a0, An angle between 0 and 359

# 0 : North (up)

# 90: East (right)

# 180: South (down)

# 270: West (left)

#-----------------------------------------------------------

ROTATE: li $at, HEADING # change HEADING port

sw $a0, 0($at) # to rotate robot

nop

jr $ra

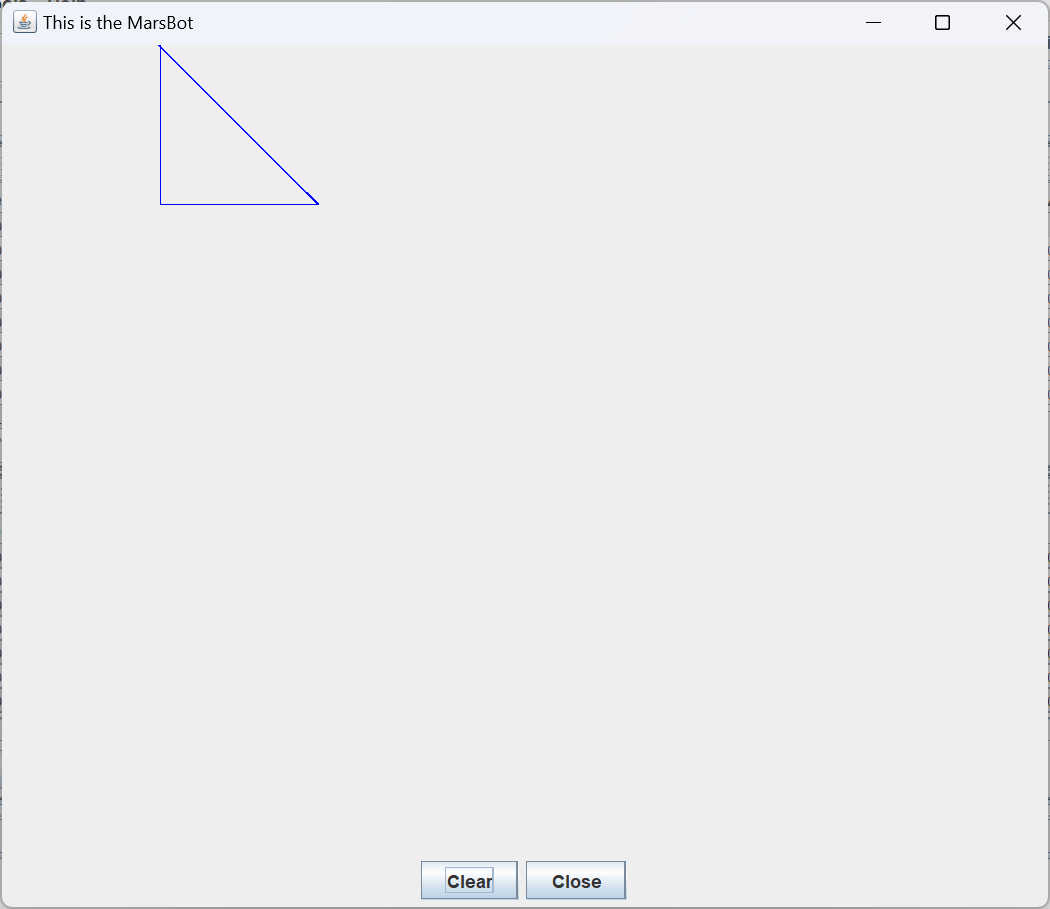
nop

* Thực thi

-Chương trình thực hiện vẽ tam giác vuông:

1. Đầu tiên di chuyển ra khỏi mép màn hình để vẽ hình.
2. Quay góc 180 độ vẽ cạnh góc vuông thứ nhất với sleep 5000ms tương đương 5.
3. Quay góc 0 độ vẽ cạnh góc vuông thứ 2 với sleep 5000ms tương đương 5.
4. Quay góc 315 vẽ cạnh huyền với sleep 7071ms tương đương 7.071 xấp xỉ .
5. Kết thúc chương trình

-Kết quả



Assignment 4

* Code

.eqv KEY\_CODE 0xFFFF0004 # ASCII code from keyboard, 1 byte

.eqv KEY\_READY 0xFFFF0000 # =1 if has a new keycode ?

# Auto clear after lw

.eqv DISPLAY\_CODE 0xFFFF000C # ASCII code to show, 1 byte

.eqv DISPLAY\_READY 0xFFFF0008 # =1 if the display has already to do

# Auto clear after sw

.data

exit\_string:

.asciiz "exit" # Chuỗi để so sánh

.text

li $k0, KEY\_CODE

li $k1, KEY\_READY

li $s0, DISPLAY\_CODE

li $s1, DISPLAY\_READY

la $t3, exit\_string # Lưu địa chỉ của chuỗi exit

li $t4, 0 # Lưu số kí tự trùng với exit tại thời điểm hiện tại

loop: nop

WaitForKey: lw $t1, 0($k1) # $t1 = [$k1] = KEY\_READY

nop

beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling

nop

#-----------------------------------------------------

ReadKey: lw $t0, 0($k0) # $t0 = [$k0] = KEY\_CODE

nop

#-----------------------------------------------------

WaitForDis: lw $t2, 0($s1) # $t2 = [$s1] = DISPLAY\_READY

nop

beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling

nop

#-----------------------------------------------------

Encrypt: addi $t0, $t0, 0 # change input key

#-----------------------------------------------------

ShowKey: sw $t0, 0($s0) # show key

nop

#-----------------------------------------------------

CheckExit:

lb $t5, 0($t3) # Nạp kí tự của chuỗi exit cần so sánh

bne $t0, $t5, NotExitChar # Nếu phím không khớp với ký tự lệnh thoát hiện tại, bỏ qua

addi $t3, $t3, 1 # Chuyển sang kí tự tiếp theo của chuỗi exit

addi $t4, $t4, 1 # Tăng kí tự hiện tại trùng với chuỗi exit lên 1

beq $t4,4, EndProgram # Nếu cả 4 kí tự mới nhất trùng với exit thì dừng

j loop

NotExitChar:

la $t3, exit\_string # Đặt lại bộ đếm lệnh thoát

li $t4, 0

j loop

# Nếu các ký tự bằng nhau thì kết thúc chương trình

EndProgram:

li $v0, 10

syscall # Kết thúc chương trình

* Thực thi

-Chương trình thực hiện so sánh số kí tự mới nhất được nhập vào nếu trùng với exit thì dừng. Nếu kí tự được nhập vào là ‘e’ thì tăng số kí tự trùng là $t4 = 1. Nếu kí tự tiếp theo là ‘x’ thì sẽ là $t4 = 2, không phải thì reset về $t4 = 0. Tương tự với ‘i’, ‘t’. Nếu $t4 = 4 thì dừng chương trình.

-Kết quả:

