Rogueima I MVP 1

- See: Part II Writing Games in Assembly Language (Game 2: Rogueima)

rogueima.sda

```
// Rogueima (C) 2026 Appledog Hu
// rogueima.sda
// rogueima 'main'

; Variables
.equ PX $00    ; player X
.equ PY $01    ; player Y
.equ RX $02    ; robot X
.equ RY $03    ; robot Y

.equ player_str    $04       ; 2 bytes
.equ player_hp     $06       ; 2 bytes
.equ player_name   $08       ; 16 bytes + 1 zero starting at $02
.equ pname_zero    $18       ; this must always be a zero even if there are
earlier zeroes.
.equ game_time     $19       ; 2 bytes
.equ player_score  $1b       ; 2 bytes

.equ VIDEO_MODE           $01EF00   ; Current video mode (1 byte)

    ; Address statement.
    .address $00C000

    ; BASIC stub: 10 SYS 269
    ;.bytes $FB, $0A, $00
    ;.bytes "SYS 269", $00
    ;.bytes $00, $00

; --- Entry point at $00010D (decimal 269) ---
start:
    ; Set mode 6.
    LDAH $40    ; Set video mode
    LDAL #6     ; to 6
    INT 0x10

    CALL @title_screen
    CALL @init_game

    CALL @get_player_name
    LDA #0
    STAL [@pname_zero]      ; ensure player name length
```

```
    JMP @main_loop    ; start game

init_game:
    ; Initialize variables
    LDAL #19
    STAL [@PX]
    LDAL #11
    STAL [@PY]
    LDAL #40
    STAL [@RX]
    LDAL #22
    STAL [@RY]

    LDA #10
    STA [@player_str]
    STA [@player_hp]

    LDA #0
    STA [@game_time]
    STA [@player_score]

    RET

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Main Loop
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
main_loop:
    CALL @draw_map
    CALL @get_input
    CALL @move_mobs
    CALL @robot_beep
    CALL @inc_game_time

    JMP @main_loop ; Player can press Q to quit.

inc_game_time:
    LDA [@game_time]
    INC A
    STA [@game_time]
    RET


get_player_name:
    LDELM @what_is_your_name
    LDAH $18      ; write string
    INT 0x10

    LDAH $68      ; IO_INPUT
    INT 0x05
    ; player name is now at ELM.
```

```
    ; Ensure player name is no longer than 16 characters.
    MOV FLD, ELM
    ADD FLD, #17
    LDAL #0
    STAL [FLD]

    ; Copy player name into variable space.
    LDFLD @player_name
name_copy_loop:
    LDAL [ELM, +]
    STAL [FLD, +]
    JNZ @name_copy_loop

    RET ;; return.

what_is_your_name:
    .bytes " What is your name? ", 0

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Get input from the player.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
get_input:
    ; Blocking GETKEY -> AL
    LDAH $02
    INT $10

    ; Uppercase: if AL >= 'a' and AL < 'z'+1, subtract 32
    CMP AL, #97
    JNC @gi_check_keys      ; AL < 'a', skip
    CMP AL, #123
    JC @gi_check_keys       ; AL >= '{', skip
    SUB AL, #32

gi_check_keys:
    CMP AL, #'H'
    JZ @move_left
    CMP AL, #'J'
    JZ @move_down
    CMP AL, #'K'
    JZ @move_up
    CMP AL, #'L'
    JZ @move_right
    CMP AL, #'Q'
    JZ @quit_game

    RET

move_left:
    LDAL [@PX]
    DEC AL
    CMP AL, #1
```

```
    JC @ml_ok               ; AL >= 1, keep
    LDAL #1
ml_ok:
    STAL [@PX]
    RET

move_down:
    LDAL [@PY]
    INC AL
    CMP AL, #23
    JNC @md_ok              ; AL < 23 (i.e. <= 22), keep
    LDAL #23
md_ok:
    STAL [@PY]
    RET

move_up:
    LDAL [@PY]
    DEC AL
    CMP AL, #1
    JC @mu_ok               ; AL >= 1, keep
    LDAL #1
mu_ok:
    STAL [@PY]
    RET

move_right:
    LDAL [@PX]
    INC AL
    CMP AL, #59
    JNC @mr_ok              ; AL < 59 (since we're in mode 6 for this game).
    LDAL #58
mr_ok:
    STAL [@PX]
    RET

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Move the robot.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
move_mobs:
    RET

;;;;;;;;;;;;;;;;;;;;;;;
;; Game Over.
;;;;;;;;;;;;;;;;;;;;;;;
game_over:
    ; CLS
    LDAH $10
    INT $10

    ; Print "OH NO! THE ROBOT CATCHES YOU."
```

```
    LDBLX @msg_game_over1
    LDAH $66
    INT $05
    LDAH $64                ; newline
    INT $05

    ; Print "GAME OVER."
    LDBLX @msg_game_over2
    LDAH $66
    INT $05
    LDAH $64
    INT $05

    RET

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Player quit game
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
quit_game:
    ; CLS
    LDAH $10
    INT $10

    ; Print "QUIT GAME"
    LDBLX @msg_quit
    LDAH $66
    INT $05

    POP ELM ; destroy return address of CALL from main loop
    RET     ; exit program.

robot_beep:
    LDELM @robot_sfx
    LDAH $52
    INT $11
    RET

;
================================================================================
; String data
;
================================================================================
msg_game_over1:
    .bytes "OH NO! THE ROBOT CATCHES YOU.", 0
msg_game_over2:
    .bytes "GAME OVER.", 0
msg_quit:
    .bytes "QUIT GAME", 10,13,0
robot_sfx:
    .bytes "T120 W1 V5 O2 L24 B", 0
```

```
; ============================================
; Title Screen
; ============================================
title_screen:
    ; Clear screen
    ; (This is done on mode switch, but we do it here anyways.)
    LDAH $10
    INT $10

    LDA $1500          ; Set cursor
    LDX #1
    LDY #1
    INT 0x10

    LDELM @str_title1
    LDA $1800
    INT 0x10
    RET

str_title1:
    .bytes "The Rogue's Tale I: Temple of Rogueima: The First Dungeon",10,13
    .bytes " Copyright 2026 by Appledog Hu.",10,10,13,0
```

draw.sda

```
// rogueima (C) 2026 Appledog Hu
// draw.asm
// Draw the display screen; borders, status, chat, wold map, etc.

draw_map:
    ; VSTOP
    LDAL $86               ; Mode 6 VSTOP
    STAL [@VIDEO_MODE]

    CALL @draw_borders
    CALL @draw_world
    CALL @draw_player
    CALL @draw_mobs
    CALL @draw_stats
    CALL @draw_msgs


    ; VSTART
    LDAL #6                ; Mode 6 VSTART
    STAL [@VIDEO_MODE]

    RET
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Draw borders.
;;;;;;;;;;;;;;;;;;;;;;;;;;
draw_borders:
    ; CLS
    LDAH $10        ; CLS
    INT $10

    LDX #0  ;; clear XH and YH.
    LDY #0

    ; Draw top/bottom border: '*' at (0..79, 0) and (0..79, 24)
    LDBL #'*'
    LDXL #0

dm_top_bottom:
    LDYL #0
    LDAH $11        ; write char AL at XL, YL
    INT $10

    LDYL #24
    LDAH $11        ; write char
    INT $10

    CMP XL, #59      ; if XL >= 10, then set carry.
    JNC @dm_skip_ms    ; So if it's not, then don't draw the middle
separator char.

    LDYL #10        ; status/text window separator
    LDAH $11        ; write char
    INT $10

dm_skip_ms:
    INC XL
    CMP XL, #80     ; Mode 6 is 80 chars wide.
    JNC @dm_top_bottom

    ; Draw left/right borders: '*' at (0, 0..24) and (59, 0..24) and (79,
0..24)
    LDBL #'*'
    LDYL #0
dm_left_right:
    LDXL #0         ; left border
    LDAH $11        ; write char
    INT $10

    LDXL #59        ; middle border
    LDAH $11        ; write char
    INT $10

    LDXL #79        ; right side border
```

```
    LDAH $11        ; write char
    INT $10

    INC YL
    CMP YL, #25
    JNC @dm_left_right
    RET


draw_player:
    ; Draw player '@'
    LDAH $11
    LDBL #'@'
    LDXL [@PX]
    LDYL [@PY]
    LDAH $11        ; write char
    INT $10
    RET


draw_mobs:
    ; stub for now
    RET


draw_stats:
    ; Set cursor
    LDA $1500
    LDX #61
    LDY #2
    INT 0x10

    LDELM @str_name  ; Draw 'Name: '
    LDA $1800        ; write string
    INT 0x10

    ; Draw player's name.
    ; This will draw right after the 'Name: ' above.
    ; That's why we added spaces to the strings (below).
    LDELM @player_name
    LDA $1800   ; write string
    INT 0x10

;;;;;;;;;;;;;;;;;;; Strength
    LDA $1500        ; set cursor
    LDX #61
    LDY #4
    INT 0x10

    LDELM @str_str     ; Draw 'Str: '
    LDA $1800          ; write string
```

```
    INT 0x10

    ; Draw player's strength score
    LDA $6300          ; print unsigned word (will print after string
above).
    LDB [@player_str]
    INT 0x05

;;;;;;;;;;;;;;;;;;; Health
    LDA $1500          ; set cursor
    LDX #61
    LDY #5
    INT 0x10

    LDELM @str_hp      ; Draw 'Health: '
    LDA $1800          ; Write string
    INT 0x10

    LDA $6300          ; Write number (unsigned)
    LDB [@player_hp]
    INT 0x05


;;;;;;;;;;;;;;;;;;; Score
    LDA $1500          ; Set cursor
    LDX #61
    LDY #7
    INT 0x10

    LDELM @str_score   ; Draw 'Score: '
    LDA $1800          ; write string
    INT 0x10

    LDA $6300          ; print number
    LDB [@player_score]
    INT 0x05


;;;;;;;;;;;;;;;;;;; Time
    LDA $1500          ; set cursor
    LDX #61
    LDY #8
    INT 0x10

    LDELM @str_time    ; Draw 'Time: '
    LDA $1800          ; write string
    INT 0x10

    LDA $6300          ; print number
    LDB [@game_time]
    INT 0x05
```

```
    RET

str_name:
    .bytes "Name: ", 0

str_str:
    .bytes "STR: ", 0

str_hp:
    .bytes "HP: ", 0

str_time:
    .bytes "T: ",0

str_score:
    .bytes "Score: ", 0

draw_msgs:
    ; We will work on this after.
    RET

draw_world:
    ; We will work on this last.
    RET
```