

Source code Megapole by Baudsurfer/RSI

BY: A GUEST ON SEP 13TH, 2015 | SYNTAX: ASM (NASM) | SIZE: 11.63 KB | VIEWS: 125 | EXPIRES: NEVER

DOWNLOAD | RAW | EMBED | REPORT ABUSE | PRINT | QR CODE | CLONE

0 Public P

1 Untit 2 sec

Untit 2 sec

Visua 14 se

Untit 16 se

Untit 17 se

Untit 25 se

Untit 26 se

Minog 27 se



Get a free year of SitePoint Premium and hosting from BlueHost for \$3.50/month



```
1. ; *****
2. ;
3. ;      /\_      /\_      /\_
4. ;      _/\_      _/\_      _/\_
5. ;      /  /      /  /      /  /
6. ;      /  /      /  /      /  /
7. ;      /  /      /  /      /  /
8. ;      /  /      /  /      /  /
9. ;      /  /      /  /      /  /
10. ;      /  /      /  /      /  /
11. ;      /  /      /  /      /  /
12. ;      /  /      /  /      /  /
13. ;
14. ;
15. ;      _/\_      _/\_      _/\_
16. ;      /  /      /  /      /  /
17. ;      /  /      /  /      /  /
18. ;      /  /      /  /      /  /
19. ;      /  /      /  /      /  /
20. ;      /  /      /  /      /  /
21. ;      /  /      /  /      /  /
22. ;      /  /      /  /      /  /
23. ;
24. ;      R E D   S E C T O R   I N C
25. ;
26. ; Metropole a 256 bytes intro by Baudsurfer/RSI 2015 aka olivier.poudade.free.fr
27. ; Presented first at the Function 2015 demoscene demo party in Budapest Hungaria
28. ; Greets Blabla Conscience Bon^2 BREWErS CODEX FLush Lineout Mandarin Onslaught
29. ; Paranoimia Quartex Rebels Razor1911 RiOT Titan and to all assembly programmers
30. ; rsi.untergrund.net twitter.com/red_sector_inc facebook.com/redsectorinc ircnet
31. ; RSI asciilogo by sEnsER/BRK vidcap youtube.com/watch?v=Z8Av7Sc7yGY by Fra/MDRN
32. ; *****
33. b equ byte           ; tested on xp, freedos, ms windows dos and its debug
34. w equ word          ; short form pretty-print helpers datatype specifiers
35. org 100h            ; entering ip=cs:256 just above .com psp 127-byte dta
36. mov fs,ax           ; ax=0? was pop bp before rewrite for non-zero fs seg
37. pop bp             ; bp=0 cs:[0ffffh]=ss:[sp]=0000 if not debug executed
38. mov al,13h         ; function switch to video mode 13h 320x200x256 & cls
39. int 10h            ; general video bios service for all mode 13h vga api
40. push w 0a000h      ; was les cx,[bx] es=9ffffh cx=20cdh &      ,[di-10h]
41. pop es             ; ms-dos v6.22 or freedos not "les rr,      ompatible
42. a:test bp,100h     ; script idx bounds reached? bp E [0;255] i.e aam 255
43. jz c               ; if hbyte OR rollover sign propagated to hbyte lsb
44. xor b[c],8h        ; xor mutex modify next opcode to keep idx normalized
45. c:dec bp           ; follow through and advance script idx dec bp/inc bp
46. e:mov cx,0ffh      ; cl=visibility fostrum, null ch implicit object mask
47. g:cwd             ; shorter xor dx,dx with ah<128 for div moved for agi
48. mov si,140h        ; vga vid mode 19 horizontal scanline width in pixels
49. mov bx,cx          ; bl=distance nullify bh raymarch object height limit
50. mov ax,di          ; di=beam spot absolute vga coord, no dos para fix-up
51. not bl             ; bl=distance/z axis orientation= -visibility fostrum
52. div si             ; main 3d projection returns with ax=y dx=x ; dh = x
53. call q             ; main 3d projection returns withah=(y-y0)*z ; bl = z
54. call q             ; main 3d projection returns withah=(x-x0)*z ; ah = y
55. mov si,46ch        ; 46ch=bda rtc off in zero seg plus ad hoc off buffer
56. add bl,[fs:si]     ; bl=z+=rtc word in bda advances camera, assumed fs=0
57. adc dh,[si]        ; dh+=beam spot camera x coordinates cs/ds:46ch & rtc
```

```

58. add ah,[si+1] ; ah+=beam spot camera y coordinates cs/ds:46dh & rtc
59. mov al,dh ; push/pop preserve texture x>>8 texel base for later
60. adc ch,al ; this object's implicit form xor /w building overlay
61. and dh,bl ; dh=x bl=x i.e x+=y bh and dl used as generic params
62. mov bh,30h ; bh=y height max of overpass, function generic param
63. mov dl,20h ; dl=y height min of overpass, function generic param
64. call r ; function returns if this object or building ray hit
65. jz h ; if objects volume intersect with ray texture former
66. push bx ; preserve prev rtc time to avoid costly seg override
67. mov dl,10h ; dl=y height max of spaceship function generic param
68. mov bh,14h ; bh=y height min of spaceship function generic param
69. sub bl,[fs:si] ; bl=z+rtc word in bda advances spaceship1 camera<--
70. sub bl,[fs:si] ; bl=z+rtc word in bda advances spaceship1 camera<--
71. xor ch,ch ; flag differenciates between spaceship* and overpass
72. call r ; function returns if this object or building ray hit
73. pop bx ; restore prev rtc time also implicit ch val returned
74. jz h ; if objects volume intersect with ray texture former
75. mov dl,18h ; dl=y height max of spaceship function generic param
76. mov bh,1ch ; bh=y height min of spaceship function generic param
77. add bl,[fs:si] ; bl=z+rtc word in bda advances spaceship2 camera-->
78. call r ; function returns if this object or building ray hit
79. jz h ; if objects volume intersect with ray texture former
80. loop g ; if no object volumes intersect then continue z rays
81. h:xchg ax,dx ; texture subroutine - clone ray collision height val
82. cmp dh,40h ; test if this ray collision height val is exactly 64
83. jz l ; process as scenery bottom floor, al=depth was saved
84. pushf ; ax disposed of in z-buffer order override data flow
85. pop ax ; subroutine marked eflags /w bit10 df=spaceship true
86. sahf ; convert this object's bit10 df to pf for conditions
87. jp k ; if z-ray collided with a spaceship object then exit
88. test cl,cl ; else test if ray collision exited on loop condition
89. jz i ; if distance=0=>no scenery intersection=open horizon
90. inc dh ; test if this ray collision height=top=255=sky limit
91. jnz j ; else ray hit other scenery building/overpass object
92. i:mov al,0ffh ; is sky so apply old b/w film rear projection effect
93. jmp m ; with brightest standard vga palette grayscale color
94. j:and al,bl ; is building/overpass process texel window step #1/3
95. xor al,dh ; is building/overpass process texel window step #2/3
96. xor dl,bl ; is building/overpass process texel bricks step #1/2
97. and dl,dh ; is building/overpass process texel bricks step #2/2
98. shl al,02h ; is building/overpass process texel window step #3/3
99. and dl,01h ; test for building/overpass window or bricks texture
100. cmovnz ax,cx ; if window texel then color val=distance 586+ opcode
101. mov ah,0ffh ; colour for window or bricks of building or overpass
102. sub ah,al ; is dynamic for windows and static for all other obj
103. k:mov al,ah ; thunk for building/overpass/window/bricks/spaceship
104. jmp m ; proceed to last step of grayscale color normalizing
105. l:not ah ; floor grey bicolor flat-shaded for building shadows
106. and al,ah ; floor color multiplexes shadow depth=k*(255-height)
107. m:cld ; common thunk nullifies next spaceship=true obj flag
108. aam 12h ; normalize with dithering add overLap ah=color/18+00
109. mov al,16 ; normalize with dithering add overLap ah=color/18+16
110. aad 1 ; dithering normalized and prepare for next frame cwd
111. test di,di ; test for all pixels plotted overrunning vga segment
112. jp o ; preserve zf flag and test if absolute beam position
113. inc ax ; parity even augmenting lighting for odd meta-pixels
114. o:stosb ; write screen pixel & advance absolute beam position
115. jnz e ; if dst idx then continue automatic vga wrap-up fill
116. mov w[si],3711h ; post-assigns camera fixed value coordinates (17,55)
117. mov w[fs:si-1ch],1701h ; bda mem vid page 0 title curs position col=1 row=24
118. mov ah,9 ; dos 1+ write $ terminated string to standard output
119. mov dx,p ; hardcoded 24h terminated ascii string of demo title
120. int 21h ; general ms-dos api /w function 9 print ds:dx string
121. jmp a ; process next demo frame (sorry no escape sequence!)
122. q:xchg ax,dx ; 3-axis rotations require 2-axis ah=dh=x dh=(y-y0)*z
123. sub ax,bp ; translate to demo script idx arbitrary origin bp,bp
124. imul ax,bx ; project abscissas/ordinates ah=(x-x0)*z dh=(y-y0)*z
125. add ax,67fh ; translate back to ~center=k*sqr(2) arbitrary fix-up
126. ret ; ----->return to caller (0c3h)
127. r:push ax ; isosurface discrimination preserve building overLay
128. test bl,40h ; is it time~depth (i.e z+rtc) displaying an overpass
129. jnz v ; if not then process default buildings intersections
130. cmp ah,bh ; is y height>min of spaceship/overpass generic param
131. jnc v ; if not then process default buildings intersections

```

```

132.    cmp ah,d1          ; is y height<max of spaceship/overpass generic param
133.    jc v              ; if not then process default buildings intersections
134.    test bl,78h       ; is spaceship/overpass 120<z depth<128 static params
135.    jnz u             ; if not then process modified building intersections
136.    test ch,ch        ; flag differenciates between spaceship* and overpass
137.    jnz t            ; if overpass then proceed to translate it vertically
138.    test al,78h      ; is spaceship only objects 120<x width<128 in static
139.    jnz u           ; if not then process modified building intersections
140.    std                ; is spaceship and visible so set df flag accordingly
141.    t:add ah,18h     ; translate spaceship/overpass objects vertically +24
142.    u:xor ah,ch      ; this object's implicit form xor /w building overlay
143.    v:test dh,10h   ; alternate scene vertical irregularity every step 16
144.    jnz x           ; i.e : _||_|||_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
145.    add dh,a1         ; reduce scene horizontally = stretch scene vertically
146.    x:or ah,dh       ; induce scene horizontal "y-colinear" irreguliarities
147.    sahf             ; implicit isosurface volume x AND y AND z AND 64=64?
148.    pop ax           ; isosurface discrimination preserve building overlay
149.    ret               ; ----->return to caller (0c3h)
150. p db "megapole$"   ; hardcoded 24h terminated ascii string of demo title
```

## RAW Paste Data

clone this paste

[illegible]

## Pastebin.com Tools & Applications

iPhone/iPad	Windows	Firefox	Chrome	WebOS	Android	Mac	Opera	Click.to	UNIX	WinPhone
-------------	---------	---------	--------	-------	---------	-----	-------	----------	------	----------

[create new paste](#) | [api](#) | [trends](#) | [syntax languages](#) | [faq](#) | [tools](#) | [privacy](#) | [cookies](#) | [contact](#) | [dmca](#) | [advertise on pastebin](#) | [shop new!](#) | [scrapir](#)

Follow us: [pastebin on facebook](#) | [pastebin on twitter](#) | [pastebin in the news](#)

## Dedicated Server Hosting by Steadfast

Pastebin v3.11 rendered in: 0.004 seconds

Site design & logo © 2015 Pastebin; user contributions (pastes) licensed under cc by-sa 3.0