0001 0000 device zxspectrum48

0003 0000

0004 0000 org 24832

0005 6100

0006 6100

0007 6100 C3 D6 81 jp start

0008 6103

0009 6103

0010 6103

0011 6103 WINDOWTOP equ 1

0012 6103 WINDOWLFT equ 0

0013 6103 WINDOWHGT equ 23

0014 6103 WINDOWWID equ 32 ;

0015 6103 MAPWID equ 3

0016 6103 FF FF FF defb 255,255,255

0017 6106 mapdat equ $

0018 6106 FF 00 FF defb 255,0,255

0019 6109 FF FF FF defb 255,255,255

0020 610C 01 stmap defb 1

0021 610D evnt00 equ $

0022 610D 3E 45 ld a,69 SPRITEINK 69

0023 610F E6 07 and 7

0024 6111 4F ld c,a

0025 6112 CD 33 85 call cspr

0026 6115 3A DF 81 ld a,(joyval) IF KEY UP

0027 6118 E6 08 and 8

0028 611A CA 39 61 jp z,a00074 1)

0029 611D 3E 01 ld a,1 LET FRAME = 1

0030 611F DD 77 07 ld (ix+7),a

0031 6122 3E 5C ld a,92 IF Y > 92

0032 6124 DD BE 08 cp (ix+8)

0033 6127 D2 36 61 jp nc,a00070 2)

0034 612A DD 35 08 dec (ix+8) SPRITEUP

0035 612D DD 35 08 dec (ix+8)

0036 6130 DD 35 08 dec (ix+8) SPRITEUP

0037 6133 DD 35 08 dec (ix+8)

ENDIF 2)

0038 6136 C3 43 61 a00070 jp a00091 ELSE 1)

0039 6139 AF a00074 xor a LET FRAME = 0

0040 613A DD 77 07 ld (ix+7),a

0041 613D DD 34 08 inc (ix+8) SPRITEDOWN

0042 6140 DD 34 08 inc (ix+8)

ENDIF 1)

0043 6143 3A DF 81 a00091 ld a,(joyval) IF KEY LEFT

0044 6146 E6 02 and 2

0045 6148 CA 5D 61 jp z,a00132 3)

0046 614B CD 93 8F call cangl IF CANGOLEFT

0047 614E C2 5D 61 jp nz,a00132 4)

0048 6151 DD 35 09 dec (ix+9) SPRITELEFT

0049 6154 DD 35 09 dec (ix+9)

0050 6157 DD 35 09 dec (ix+9) SPRITELEFT

0051 615A DD 35 09 dec (ix+9)

ENDIF 4)

ENDIF 3)

0052 615D 3A DF 81 a00132 ld a,(joyval) IF KEY RIGHT

0053 6160 E6 01 and 1

0054 6162 CA 77 61 jp z,a00173 5)

0055 6165 CD 9E 8F call cangr IF CANGORIGHT

0056 6168 C2 77 61 jp nz,a00173 6)

0057 616B DD 34 09 inc (ix+9) SPRITERIGHT

0058 616E DD 34 09 inc (ix+9)

0059 6171 DD 34 09 inc (ix+9) SPRITERIGHT

0060 6174 DD 34 09 inc (ix+9)

ENDIF 6)

ENDIF 5)

0061 6177 3A DF 81 a00173 ld a,(joyval) IF KEY FIRE

0062 617A E6 10 and 16

0063 617C CA AD 61 jp z,a00261 7)

0064 617F AF xor a LET SETTINGA = 0

0065 6180 DD BE 0B cp (ix+11)

0066 6183 C2 AA 61 jp nz,a00257 8)

0067 6186 3E 0C ld a,12 IF S < 12

0068 6188 21 FE 81 ld hl,vars

0069 618B BE cp (hl)

0070 618C DA AA 61 jp c,a00257

0071 618F CA AA 61 jp z,a00257 9)

0072 6192 01 01 01 ld bc,257 SPAWN 1 1

0073 6195 CD 7F 8C call spawn

0074 6198 3E 01 ld a,1 LET SETTINGA = 1

0075 619A DD 77 0B ld (ix+11),a

0076 619D DD 2A CE 8C ld ix,(spptr) SPAWNED

0077 61A1 3E 01 ld a,1 LET DIRECTION = 1

0078 61A3 DD 77 0A ld (ix+10),a

0079 61A6 DD 2A E7 92 ld ix,(ogptr) ENDSPRITE

0080 61AA C3 B1 61 a00257 jp a00269 ENDIF 9)

ENDIF 8)

ELSE 7)

0081 61AD AF a00261 xor a LET SETTINGA = 0

0082 61AE DD 77 0B ld (ix+11),a

ENDIF 7)

0083 61B1 3A E8 81 a00269 ld a,(wnbotx) IF Y >= BOTTOMEDGE

0084 61B4 DD BE 08 cp (ix+8)

0085 61B7 28 03 jr z,$+5

0086 61B9 D2 C2 61 jp nc,a00298 10)

0087 61BC 3A E8 81 ld a,(wnbotx) LET Y = BOTTOMEDGE

0088 61BF DD 77 08 ld (ix+8),a

ENDIF 10)

0089 61C2 3E 63 a00298 ld a,99 IF DIRECTION = 99

0090 61C4 DD BE 0A cp (ix+10)

0091 61C7 C2 DD 61 jp nz,a00344 11)

0092 61CA 3E 06 ld a,6 LET TYPE = 6

0093 61CC DD 77 05 ld (ix+5),a

0094 61CF 3E 04 ld a,4 LET IMAGE = 4

0095 61D1 DD 77 06 ld (ix+6),a

0096 61D4 AF xor a LET FRAME = 0

0097 61D5 DD 77 07 ld (ix+7),a

0098 61D8 3E 01 ld a,1 LET DIRECTION 1

0099 61DA DD 77 0A ld (ix+10),a

ENDIF (11)

0100 61DD DD 7E 09 a00344 ld a,(ix+9) LET M = X

0101 61E0 32 F8 81 ld (varm),a

0102 61E3 DD 7E 08 ld a,(ix+8) LET N = Y

0103 61E6 32 F9 81 ld (varn),a

0104 61E9 C9 ret

0105 61EA evnt01 equ $

0106 61EA 3E 01 ld a,1

0107 61EC DD BE 0A cp (ix+10)

0108 61EF C2 2D 62 jp nz,b00119

0109 61F2 3E 01 ld a,1

0110 61F4 DD 77 06 ld (ix+6),a

0111 61F7 3E 06 ld a,6

0112 61F9 32 81 5C ld (loopa),a

0113 61FC DD 35 08 b00033 dec (ix+8)

0114 61FF DD 35 08 dec (ix+8)

0115 6202 0E 02 ld c,2

0116 6204 DD 7E 0C ld a,(ix+12)

0117 6207 81 add a,c

0118 6208 DD 77 0C ld (ix+12),a

0119 620B 3E 58 ld a,88

0120 620D DD BE 0C cp (ix+12)

0121 6210 C2 18 62 jp nz,b00081

0122 6213 3E 63 ld a,99

0123 6215 DD 77 0A ld (ix+10),a

0124 6218 3E 08 b00081 ld a,8

0125 621A DD BE 08 cp (ix+8)

0126 621D DA 26 62 jp c,b00105

0127 6220 3E 63 ld a,99

0128 6222 DD 77 0A ld (ix+10),a

0129 6225 C9 ret

0130 6226 21 81 5C b00105 ld hl,loopa

0131 6229 35 dec (hl)

0132 622A C2 FC 61 jp nz,b00033

0133 622D 3E 63 b00119 ld a,99

0134 622F DD BE 0A cp (ix+10)

0135 6232 C2 4F 62 jp nz,b00175

0136 6235 DD 7E 07 ld a,(ix+7)

0137 6238 3C inc a

0138 6239 DD 77 07 ld (ix+7),a

0139 623C 3E 05 ld a,5

0140 623E DD BE 07 cp (ix+7)

0141 6241 C2 4F 62 jp nz,b00175

0142 6244 DD 36 05 FF ld (ix+5),255

0143 6248 3A FE 81 ld a,(vars)

0144 624B 3D dec a

0145 624C 32 FE 81 ld (vars),a

0146 624F C9 b00175 ret

0147 6250 evnt02 equ $

0148 6250 C9 ret

0149 6251 evnt03 equ $

0150 6251 3E 47 ld a,71

0151 6253 E6 07 and 7

0152 6255 4F ld c,a

0153 6256 CD 33 85 call cspr

0154 6259 DD 34 08 inc (ix+8)

0155 625C DD 34 08 inc (ix+8)

0156 625F DD 34 08 inc (ix+8)

0157 6262 DD 34 08 inc (ix+8)

0158 6265 DD 34 08 inc (ix+8)

0159 6268 DD 34 08 inc (ix+8)

0160 626B AF xor a

0161 626C DD BE 0A cp (ix+10)

0162 626F C2 75 62 jp nz,d00062

0163 6272 C3 9A 62 jp d00119

0164 6275 3E 08 d00062 ld a,8

0165 6277 DD BE 0A cp (ix+10)

0166 627A DA 89 62 jp c,d00092

0167 627D CA 89 62 jp z,d00092

0168 6280 DD 34 09 inc (ix+9)

0169 6283 DD 34 09 inc (ix+9)

0170 6286 C3 9A 62 jp d00119

0171 6289 3E 08 d00092 ld a,8

0172 628B DD BE 0A cp (ix+10)

0173 628E C2 94 62 jp nz,d00110

0174 6291 C3 9A 62 jp d00119

0175 6294 DD 35 09 d00110 dec (ix+9)

0176 6297 DD 35 09 dec (ix+9)

0177 629A 06 00 d00119 ld b,0

0178 629C CD B7 94 call sktyp

0179 629F D2 2C 63 jp nc,d00367

0180 62A2 3A F8 81 ld a,(varm)

0181 62A5 DD BE 09 cp (ix+9)

0182 62A8 D2 DF 62 jp nc,d00238

0183 62AB 3A F8 81 ld a,(varm)

0184 62AE 4F ld c,a

0185 62AF DD 7E 09 ld a,(ix+9)

0186 62B2 91 sub c

0187 62B3 DD 77 09 ld (ix+9),a

0188 62B6 3E 0A ld a,10

0189 62B8 21 08 82 ld hl,varrnd

0190 62BB BE cp (hl)

0191 62BC DA D1 62 jp c,d00212

0192 62BF DD 2A D8 94 ld ix,(skptr)

0193 62C3 3E 63 ld a,99

0194 62C5 DD 77 0A ld (ix+10),a

0195 62C8 DD 2A E7 92 ld ix,(ogptr)

0196 62CC 3E 63 ld a,99

0197 62CE DD 77 0A ld (ix+10),a

0198 62D1 3A F8 81 d00212 ld a,(varm)

0199 62D4 4F ld c,a

0200 62D5 DD 7E 09 ld a,(ix+9)

0201 62D8 81 add a,c

0202 62D9 DD 77 09 ld (ix+9),a

0203 62DC C3 2C 63 jp d00367

0204 62DF 3A F8 81 d00238 ld a,(varm)

0205 62E2 DD BE 09 cp (ix+9)

0206 62E5 DA 1F 63 jp c,d00347

0207 62E8 CA 1F 63 jp z,d00347

0208 62EB DD 7E 09 ld a,(ix+9)

0209 62EE 4F ld c,a

0210 62EF 3A F8 81 ld a,(varm)

0211 62F2 91 sub c

0212 62F3 32 F8 81 ld (varm),a

0213 62F6 3E 08 ld a,8

0214 62F8 21 F8 81 ld hl,varm

0215 62FB BE cp (hl)

0216 62FC DA 11 63 jp c,d00322

0217 62FF DD 2A D8 94 ld ix,(skptr)

0218 6303 3E 63 ld a,99

0219 6305 DD 77 0A ld (ix+10),a

0220 6308 DD 2A E7 92 ld ix,(ogptr)

0221 630C 3E 63 ld a,99

0222 630E DD 77 0A ld (ix+10),a

0223 6311 DD 7E 09 d00322 ld a,(ix+9)

0224 6314 4F ld c,a

0225 6315 3A F8 81 ld a,(varm)

0226 6318 81 add a,c

0227 6319 32 F8 81 ld (varm),a

0228 631C C3 2C 63 jp d00367

0229 631F DD 2A D8 94 d00347 ld ix,(skptr)

0230 6323 3E 63 ld a,99

0231 6325 DD 77 0A ld (ix+10),a

0232 6328 DD 2A E7 92 ld ix,(ogptr)

0233 632C 3E 63 d00367 ld a,99

0234 632E DD BE 0A cp (ix+10)

0235 6331 C2 47 63 jp nz,d00413

0236 6334 3E 06 ld a,6

0237 6336 DD 77 05 ld (ix+5),a

0238 6339 3E 04 ld a,4

0239 633B DD 77 06 ld (ix+6),a

0240 633E AF xor a

0241 633F DD 77 07 ld (ix+7),a

0242 6342 3E 02 ld a,2

0243 6344 DD 77 0A ld (ix+10),a

0244 6347 3E BC d00413 ld a,188

0245 6349 DD BE 08 cp (ix+8)

0246 634C 28 03 jr z,$+5

0247 634E D2 5C 63 jp nc,d00448

0248 6351 DD 36 05 FF ld (ix+5),255

0249 6355 3A FE 81 ld a,(vars)

0250 6358 3D dec a

0251 6359 32 FE 81 ld (vars),a

0252 635C C9 d00448 ret

0253 635D evnt04 equ $

0254 635D C9 ret

0255 635E evnt05 equ $

0256 635E 3A EE 81 ld a,(varc)

0257 6361 E6 07 and 7

0258 6363 4F ld c,a

0259 6364 CD 33 85 call cspr

0260 6367 AF xor a

0261 6368 DD BE 0B cp (ix+11)

0262 636B C2 9E 63 jp nz,f00112

0263 636E DD 36 10 FF ld (ix+16),255

0264 6372 3A F7 81 ld a,(varl)

0265 6375 32 08 82 ld (varrnd),a

0266 6378 0E 32 ld c,50

0267 637A 3A 08 82 ld a,(varrnd)

0268 637D 81 add a,c

0269 637E 32 08 82 ld (varrnd),a

0270 6381 3A 08 82 f00065 ld a,(varrnd)

0271 6384 DD BE 0A cp (ix+10)

0272 6387 CA 99 63 jp z,f00103

0273 638A CD 43 68 call read05

0274 638D DD 77 0A ld (ix+10),a

0275 6390 CD 43 68 call read05

0276 6393 DD 77 0C ld (ix+12),a

0277 6396 C3 81 63 jp f00065

0278 6399 3E 01 f00103 ld a,1

0279 639B DD 77 0B ld (ix+11),a

0280 639E 3E 01 f00112 ld a,1

0281 63A0 DD BE 0B cp (ix+11)

0282 63A3 C2 C5 63 jp nz,f00175

0283 63A6 CD 43 68 call read05

0284 63A9 DD 77 0A ld (ix+10),a

0285 63AC CD 43 68 call read05

0286 63AF DD 77 0C ld (ix+12),a

0287 63B2 3E 02 ld a,2

0288 63B4 DD 77 0B ld (ix+11),a

0289 63B7 3E 02 ld a,2

0290 63B9 DD BE 06 cp (ix+6)

0291 63BC C2 C5 63 jp nz,f00175

0292 63BF DD 7E 0A ld a,(ix+10)

0293 63C2 DD 77 07 ld (ix+7),a

0294 63C5 06 01 f00175 ld b,1

0295 63C7 CD B7 94 call sktyp

0296 63CA D2 DF 63 jp nc,f00216

0297 63CD DD 2A D8 94 ld ix,(skptr)

0298 63D1 3E 63 ld a,99

0299 63D3 DD 77 0A ld (ix+10),a

0300 63D6 DD 2A E7 92 ld ix,(ogptr)

0301 63DA 3E FF ld a,255

0302 63DC DD 77 0A ld (ix+10),a

0303 63DF 06 00 f00216 ld b,0

0304 63E1 CD B7 94 call sktyp

0305 63E4 D2 F9 63 jp nc,f00258

0306 63E7 DD 2A D8 94 ld ix,(skptr)

0307 63EB 3E 63 ld a,99

0308 63ED DD 77 0A ld (ix+10),a

0309 63F0 DD 2A E7 92 ld ix,(ogptr)

0310 63F4 3E FF ld a,255

0311 63F6 DD 77 0A ld (ix+10),a

0312 63F9 3E FF f00258 ld a,255

0313 63FB DD BE 0A cp (ix+10)

0314 63FE C2 13 64 jp nz,f00304

0315 6401 3E 06 ld a,6

0316 6403 DD 77 05 ld (ix+5),a

0317 6406 3E 04 ld a,4

0318 6408 DD 77 06 ld (ix+6),a

0319 640B AF xor a

0320 640C DD 77 07 ld (ix+7),a

0321 640F AF xor a

0322 6410 DD 77 0A ld (ix+10),a

0323 6413 AF f00304 xor a

0324 6414 DD BE 0A cp (ix+10)

0325 6417 C2 26 64 jp nz,f00335

0326 641A DD 35 08 dec (ix+8)

0327 641D DD 35 08 dec (ix+8)

0328 6420 DD 35 08 dec (ix+8)

0329 6423 DD 35 08 dec (ix+8)

0330 6426 3E 01 f00335 ld a,1

0331 6428 DD BE 0A cp (ix+10)

0332 642B C2 40 64 jp nz,f00375

0333 642E DD 34 09 inc (ix+9)

0334 6431 DD 34 09 inc (ix+9)

0335 6434 DD 35 08 dec (ix+8)

0336 6437 DD 35 08 dec (ix+8)

0337 643A DD 35 08 dec (ix+8)

0338 643D DD 35 08 dec (ix+8)

0339 6440 3E 02 f00375 ld a,2

0340 6442 DD BE 0A cp (ix+10)

0341 6445 C2 60 64 jp nz,f00424

0342 6448 DD 34 09 inc (ix+9)

0343 644B DD 34 09 inc (ix+9)

0344 644E DD 34 09 inc (ix+9)

0345 6451 DD 34 09 inc (ix+9)

0346 6454 DD 35 08 dec (ix+8)

0347 6457 DD 35 08 dec (ix+8)

0348 645A DD 35 08 dec (ix+8)

0349 645D DD 35 08 dec (ix+8)

0350 6460 3E 03 f00424 ld a,3

0351 6462 DD BE 0A cp (ix+10)

0352 6465 C2 7A 64 jp nz,f00464

0353 6468 DD 34 09 inc (ix+9)

0354 646B DD 34 09 inc (ix+9)

0355 646E DD 34 09 inc (ix+9)

0356 6471 DD 34 09 inc (ix+9)

0357 6474 DD 35 08 dec (ix+8)

0358 6477 DD 35 08 dec (ix+8)

0359 647A 3E 04 f00464 ld a,4

0360 647C DD BE 0A cp (ix+10)

0361 647F C2 8E 64 jp nz,f00495

0362 6482 DD 34 09 inc (ix+9)

0363 6485 DD 34 09 inc (ix+9)

0364 6488 DD 34 09 inc (ix+9)

0365 648B DD 34 09 inc (ix+9)

0366 648E 3E 05 f00495 ld a,5

0367 6490 DD BE 0A cp (ix+10)

0368 6493 C2 A8 64 jp nz,f00535

0369 6496 DD 34 09 inc (ix+9)

0370 6499 DD 34 09 inc (ix+9)

0371 649C DD 34 09 inc (ix+9)

0372 649F DD 34 09 inc (ix+9)

0373 64A2 DD 34 08 inc (ix+8)

0374 64A5 DD 34 08 inc (ix+8)

0375 64A8 3E 06 f00535 ld a,6

0376 64AA DD BE 0A cp (ix+10)

0377 64AD C2 C8 64 jp nz,f00584

0378 64B0 DD 34 09 inc (ix+9)

0379 64B3 DD 34 09 inc (ix+9)

0380 64B6 DD 34 09 inc (ix+9)

0381 64B9 DD 34 09 inc (ix+9)

0382 64BC DD 34 08 inc (ix+8)

0383 64BF DD 34 08 inc (ix+8)

0384 64C2 DD 34 08 inc (ix+8)

0385 64C5 DD 34 08 inc (ix+8)

0386 64C8 3E 07 f00584 ld a,7

0387 64CA DD BE 0A cp (ix+10)

0388 64CD C2 E2 64 jp nz,f00624

0389 64D0 DD 34 08 inc (ix+8)

0390 64D3 DD 34 08 inc (ix+8)

0391 64D6 DD 34 08 inc (ix+8)

0392 64D9 DD 34 08 inc (ix+8)

0393 64DC DD 34 09 inc (ix+9)

0394 64DF DD 34 09 inc (ix+9)

0395 64E2 3E 08 f00624 ld a,8

0396 64E4 DD BE 0A cp (ix+10)

0397 64E7 C2 F6 64 jp nz,f00655

0398 64EA DD 34 08 inc (ix+8)

0399 64ED DD 34 08 inc (ix+8)

0400 64F0 DD 34 08 inc (ix+8)

0401 64F3 DD 34 08 inc (ix+8)

0402 64F6 3E 09 f00655 ld a,9

0403 64F8 DD BE 0A cp (ix+10)

0404 64FB C2 10 65 jp nz,f00695

0405 64FE DD 34 08 inc (ix+8)

0406 6501 DD 34 08 inc (ix+8)

0407 6504 DD 34 08 inc (ix+8)

0408 6507 DD 34 08 inc (ix+8)

0409 650A DD 35 09 dec (ix+9)

0410 650D DD 35 09 dec (ix+9)

0411 6510 3E 0A f00695 ld a,10

0412 6512 DD BE 0A cp (ix+10)

0413 6515 C2 30 65 jp nz,f00744

0414 6518 DD 34 08 inc (ix+8)

0415 651B DD 34 08 inc (ix+8)

0416 651E DD 34 08 inc (ix+8)

0417 6521 DD 34 08 inc (ix+8)

0418 6524 DD 35 09 dec (ix+9)

0419 6527 DD 35 09 dec (ix+9)

0420 652A DD 35 09 dec (ix+9)

0421 652D DD 35 09 dec (ix+9)

0422 6530 3E 0B f00744 ld a,11

0423 6532 DD BE 0A cp (ix+10)

0424 6535 C2 4A 65 jp nz,f00784

0425 6538 DD 34 08 inc (ix+8)

0426 653B DD 34 08 inc (ix+8)

0427 653E DD 35 09 dec (ix+9)

0428 6541 DD 35 09 dec (ix+9)

0429 6544 DD 35 09 dec (ix+9)

0430 6547 DD 35 09 dec (ix+9)

0431 654A 3E 0C f00784 ld a,12

0432 654C DD BE 0A cp (ix+10)

0433 654F C2 5E 65 jp nz,f00815

0434 6552 DD 35 09 dec (ix+9)

0435 6555 DD 35 09 dec (ix+9)

0436 6558 DD 35 09 dec (ix+9)

0437 655B DD 35 09 dec (ix+9)

0438 655E 3E 0D f00815 ld a,13

0439 6560 DD BE 0A cp (ix+10)

0440 6563 C2 78 65 jp nz,f00856

0441 6566 DD 35 09 dec (ix+9)

0442 6569 DD 35 09 dec (ix+9)

0443 656C DD 35 09 dec (ix+9)

0444 656F DD 35 09 dec (ix+9)

0445 6572 DD 35 08 dec (ix+8)

0446 6575 DD 35 08 dec (ix+8)

0447 6578 3E 0E f00856 ld a,14

0448 657A DD BE 0A cp (ix+10)

0449 657D C2 98 65 jp nz,f00905

0450 6580 DD 35 09 dec (ix+9)

0451 6583 DD 35 09 dec (ix+9)

0452 6586 DD 35 09 dec (ix+9)

0453 6589 DD 35 09 dec (ix+9)

0454 658C DD 35 08 dec (ix+8)

0455 658F DD 35 08 dec (ix+8)

0456 6592 DD 35 08 dec (ix+8)

0457 6595 DD 35 08 dec (ix+8)

0458 6598 3E 0F f00905 ld a,15

0459 659A DD BE 0A cp (ix+10)

0460 659D C2 B2 65 jp nz,f00945

0461 65A0 DD 35 09 dec (ix+9)

0462 65A3 DD 35 09 dec (ix+9)

0463 65A6 DD 35 08 dec (ix+8)

0464 65A9 DD 35 08 dec (ix+8)

0465 65AC DD 35 08 dec (ix+8)

0466 65AF DD 35 08 dec (ix+8)

0467 65B2 3A E7 81 f00945 ld a,(wnlftx)

0468 65B5 DD BE 09 cp (ix+9)

0469 65B8 DA C0 65 jp c,f00968

0470 65BB 3E FE ld a,254

0471 65BD DD 77 0A ld (ix+10),a

0472 65C0 3A E9 81 f00968 ld a,(wnrgtx)

0473 65C3 DD BE 09 cp (ix+9)

0474 65C6 28 03 jr z,$+5

0475 65C8 D2 D0 65 jp nc,f00996

0476 65CB 3E FE ld a,254

0477 65CD DD 77 0A ld (ix+10),a

0478 65D0 3A E8 81 f00996 ld a,(wnbotx)

0479 65D3 DD BE 08 cp (ix+8)

0480 65D6 28 03 jr z,$+5

0481 65D8 D2 E0 65 jp nc,f01023

0482 65DB 3E FE ld a,254

0483 65DD DD 77 0A ld (ix+10),a

0484 65E0 3E FE f01023 ld a,254

0485 65E2 DD BE 0A cp (ix+10)

0486 65E5 C2 FA 65 jp nz,f01068

0487 65E8 DD 36 05 FF ld (ix+5),255

0488 65EC 3A FE 81 ld a,(vars)

0489 65EF 3D dec a

0490 65F0 32 FE 81 ld (vars),a

0491 65F3 3A F1 81 ld a,(varf)

0492 65F6 3D dec a

0493 65F7 32 F1 81 ld (varf),a

0494 65FA 3A FC 81 f01068 ld a,(varq)

0495 65FD DD BE 08 cp (ix+8)

0496 6600 DA 7B 66 jp c,f01294

0497 6603 CA 7B 66 jp z,f01294

0498 6606 AF xor a

0499 6607 21 02 82 ld hl,varw

0500 660A BE cp (hl)

0501 660B D2 7B 66 jp nc,f01294

0502 660E 3A F8 81 ld a,(varm)

0503 6611 DD BE 09 cp (ix+9)

0504 6614 DA 42 66 jp c,f01194

0505 6617 CA 42 66 jp z,f01194

0506 661A DD 7E 09 ld a,(ix+9)

0507 661D 4F ld c,a

0508 661E 3A F8 81 ld a,(varm)

0509 6621 91 sub c

0510 6622 32 F8 81 ld (varm),a

0511 6625 3A F4 81 ld a,(vari)

0512 6628 21 F8 81 ld hl,varm

0513 662B BE cp (hl)

0514 662C DA 34 66 jp c,f01168

0515 662F 3E 01 ld a,1

0516 6631 DD 77 0D ld (ix+13),a

0517 6634 DD 7E 09 f01168 ld a,(ix+9)

0518 6637 4F ld c,a

0519 6638 3A F8 81 ld a,(varm)

0520 663B 81 add a,c

0521 663C 32 F8 81 ld (varm),a

0522 663F C3 7B 66 jp f01294

0523 6642 DD 7E 09 f01194 ld a,(ix+9)

0524 6645 21 F8 81 ld hl,varm

0525 6648 BE cp (hl)

0526 6649 DA 76 66 jp c,f01286

0527 664C CA 76 66 jp z,f01286

0528 664F 3A F8 81 ld a,(varm)

0529 6652 4F ld c,a

0530 6653 DD 7E 09 ld a,(ix+9)

0531 6656 91 sub c

0532 6657 DD 77 09 ld (ix+9),a

0533 665A 3A F4 81 ld a,(vari)

0534 665D DD BE 09 cp (ix+9)

0535 6660 C2 68 66 jp nz,f01260

0536 6663 3E 01 ld a,1

0537 6665 DD 77 0D ld (ix+13),a

0538 6668 3A F8 81 f01260 ld a,(varm)

0539 666B 4F ld c,a

0540 666C DD 7E 09 ld a,(ix+9)

0541 666F 81 add a,c

0542 6670 DD 77 09 ld (ix+9),a

0543 6673 C3 7B 66 jp f01294

0544 6676 3E 01 f01286 ld a,1

0545 6678 DD 77 0D ld (ix+13),a

0546 667B 3E 01 f01294 ld a,1

0547 667D DD BE 0D cp (ix+13)

0548 6680 C2 C5 66 jp nz,f01424

0549 6683 3E 09 ld a,9

0550 6685 21 FE 81 ld hl,vars

0551 6688 BE cp (hl)

0552 6689 DA C1 66 jp c,f01415

0553 668C CA C1 66 jp z,f01415

0554 668F 3A F3 81 ld a,(varh)

0555 6692 57 ld d,a

0556 6693 CD D6 90 call random

0557 6696 67 ld h,a

0558 6697 CD 04 8A call imul

0559 669A 7C ld a,h

0560 669B 32 08 82 ld (varrnd),a

0561 669E 3E 01 ld a,1

0562 66A0 21 08 82 ld hl,varrnd

0563 66A3 BE cp (hl)

0564 66A4 C2 C1 66 jp nz,f01415

0565 66A7 DD 7E 0A ld a,(ix+10)

0566 66AA 32 08 82 ld (varrnd),a

0567 66AD 01 03 06 ld bc,1539

0568 66B0 CD 7F 8C call spawn

0569 66B3 DD 2A CE 8C ld ix,(spptr)

0570 66B7 3A 08 82 ld a,(varrnd)

0571 66BA DD 77 0A ld (ix+10),a

0572 66BD DD 2A E7 92 ld ix,(ogptr)

0573 66C1 AF f01415 xor a

0574 66C2 DD 77 0D ld (ix+13),a

0575 66C5 DD 7E 0C f01424 ld a,(ix+12)

0576 66C8 3D dec a

0577 66C9 DD 77 0C ld (ix+12),a

0578 66CC AF xor a

0579 66CD DD BE 0C cp (ix+12)

0580 66D0 C2 D8 66 jp nz,f01458

0581 66D3 3E 01 ld a,1

0582 66D5 DD 77 0B ld (ix+11),a

0583 66D8 C9 f01458 ret

0584 66D9 DB 66 rptr05 defw rdat05

0585 66DB rdat05 defb 51,0,8,20,7,8,6,8,5,8,4

0585 66DB 3300081407080608050804

0586 66E6 defb 16,52,0,8,20,9,8,10,8,11,8

0586 66E6 103400081409080A080B08

0587 66F1 defb 12,16,53,0,12,48,11,2,10,2,9

0587 66F1 0C1035000C300B020A0209

0588 66FC defb 2,8,2,7,2,6,2,5,2,4,40

0588 66FC 0208020702060205020428

0589 6707 defb 5,2,6,2,7,2,8,2,9,2,10

0589 6707 050206020702080209020A

0590 6712 defb 40,54,0,4,48,5,2,6,2,7,2

0590 6712 2836000430050206020702

0591 671D defb 8,2,9,2,10,2,11,2,12,40,11

0591 671D 080209020A020B020C280B

0592 6728 defb 2,10,2,9,2,8,2,7,2,6,40

0592 6728 020A020902080207020628

0593 6733 defb 55,0,13,26,14,4,15,4,0,4,1

0593 6733 37000D1A0E040F04000401

0594 673E defb 4,2,4,3,4,4,4,5,4,6,4

0594 673E 0402040304040405040604

0595 6749 defb 7,4,8,4,9,4,10,40,56,0,3

0595 6749 0704080409040A28380003

0596 6754 defb 26,1,4,0,4,15,4,14,4,13,4

0596 6754 1A010400040F040E040D04

0597 675F defb 12,4,11,4,10,4,9,4,8,4,7

0597 675F 0C040B040A040904080407

0598 676A defb 4,6,40,57,0,8,16,7,4,6,4

0598 676A 0406283900081007040604

0599 6775 defb 5,4,4,4,3,4,2,4,1,4,0

0599 6775 0504040403040204010400

0600 6780 defb 4,15,4,14,4,13,4,12,4,11,4

0600 6780 040F040E040D040C040B04

0601 678B defb 10,4,9,4,8,4,7,4,6,4,5

0601 678B 0A04090408040704060405

0602 6796 defb 4,4,60,58,0,8,16,9,4,10,4

0602 6796 04043C3A00081009040A04

0603 67A1 defb 11,4,12,4,13,4,14,4,15,4,0

0603 67A1 0B040C040D040E040F0400

0604 67AC defb 4,1,4,2,4,3,4,4,4,5,4

0604 67AC 0401040204030404040504

0605 67B7 defb 6,4,7,4,8,4,9,4,10,4,11

0605 67B7 06040704080409040A040B

0606 67C2 defb 2,12,60,59,0,8,16,9,4,10,4

0606 67C2 020C3C3B00081009040A04

0607 67CD defb 11,4,12,4,13,4,14,4,15,4,0

0607 67CD 0B040C040D040E040F0400

0608 67D8 defb 4,1,4,2,4,3,4,4,4,5,4

0608 67D8 0401040204030404040504

0609 67E3 defb 6,4,7,4,8,4,9,4,10,4,11

0609 67E3 06040704080409040A040B

0610 67EE defb 2,12,60,60,0,4,20,13,4,14,4

0610 67EE 020C3C3C0004140D040E04

0611 67F9 defb 15,4,0,8,1,4,2,4,3,4,4

0611 67F9 0F04000801040204030404

0612 6804 defb 4,8,8,7,2,6,2,5,2,4,40

0612 6804 0408080702060205020428

0613 680F defb 61,0,5,2,6,2,7,2,8,2,9

0613 680F 3D00050206020702080209

0614 681A defb 2,10,2,11,2,12,2,11,3,10,3

0614 681A 020A020B020C020B030A03

0615 6825 defb 9,3,8,3,7,3,6,3,5,4,6

0615 6825 0903080307030603050406

0616 6830 defb 4,7,4,8,4,9,4,10,4,62,0

0616 6830 040704080409040A043E00

0617 683B defb 8,3,4,16,9,24,4,40

0617 683B 0803041009180428

0618 6843 DD 6E 0F read05 ld l,(ix+15)

0619 6846 DD 66 10 ld h,(ix+16)

0620 6849 11 43 68 ld de,rdat05+360

0621 684C 37 scf

0622 684D EB ex de,hl

0623 684E ED 52 sbc hl,de

0624 6850 EB ex de,hl

0625 6851 30 03 jr nc,$+5

0626 6853 21 DB 66 ld hl,rdat05

0627 6856 7E ld a,(hl)

0628 6857 23 inc hl

0629 6858 DD 75 0F ld (ix+15),l

0630 685B DD 74 10 ld (ix+16),h

0631 685E C9 ret

0632 685F C9 ret

0633 6860 evnt06 equ $

0634 6860 AF xor a

0635 6861 CD 89 94 call animsp

0636 6864 3E 04 ld a,4

0637 6866 DD BE 07 cp (ix+7)

0638 6869 C2 CD 68 jp nz,g00191

0639 686C AF xor a

0640 686D DD BE 0A cp (ix+10)

0641 6870 C2 A9 68 jp nz,g00130

0642 6873 DD 36 05 FF ld (ix+5),255

0643 6877 3A FE 81 ld a,(vars)

0644 687A 3D dec a

0645 687B 32 FE 81 ld (vars),a

0646 687E 3A EF 81 ld a,(vard)

0647 6881 3D dec a

0648 6882 32 EF 81 ld (vard),a

0649 6885 3A F1 81 ld a,(varf)

0650 6888 3D dec a

0651 6889 32 F1 81 ld (varf),a

0652 688C AF xor a

0653 688D 21 EF 81 ld hl,vard

0654 6890 BE cp (hl)

0655 6891 C2 9D 68 jp nz,g00109

0656 6894 21 F4 01 ld hl,500

0657 6897 CD 29 8D call addsc

0658 689A C3 A6 68 jp g00126

0659 689D 3A FB 81 g00109 ld a,(varp)

0660 68A0 6F ld l,a

0661 68A1 26 00 ld h,0

0662 68A3 CD 29 8D call addsc

0663 68A6 C3 B5 68 g00126 jp g00152

0664 68A9 3E 01 g00130 ld a,1

0665 68AB DD BE 0A cp (ix+10)

0666 68AE C2 B5 68 jp nz,g00152

0667 68B1 21 0E 82 ld hl,deadf

0668 68B4 74 ld (hl),h

0669 68B5 3E 02 g00152 ld a,2

0670 68B7 DD BE 0A cp (ix+10)

0671 68BA C2 C8 68 jp nz,g00183

0672 68BD DD 36 05 FF ld (ix+5),255

0673 68C1 3A FE 81 ld a,(vars)

0674 68C4 3D dec a

0675 68C5 32 FE 81 ld (vars),a

0676 68C8 3E 01 g00183 ld a,1

0677 68CA 32 FF 81 ld (vart),a

0678 68CD C9 g00191 ret

0679 68CE evnt07 equ $

0680 68CE AF xor a

0681 68CF DD BE 0B cp (ix+11)

0682 68D2 C2 4F 69 jp nz,h00217

0683 68D5 AF xor a

0684 68D6 DD 77 0E ld (ix+14),a

0685 68D9 AF xor a

0686 68DA DD 77 0C ld (ix+12),a

0687 68DD DD 36 10 FF ld (ix+16),255

0688 68E1 3A F7 81 ld a,(varl)

0689 68E4 32 81 5C ld (loopa),a

0690 68E7 CD 5C 6A h00047 call read07

0691 68EA DD 77 0D ld (ix+13),a

0692 68ED CD 5C 6A call read07

0693 68F0 DD 77 09 ld (ix+9),a

0694 68F3 CD 5C 6A call read07

0695 68F6 DD 77 08 ld (ix+8),a

0696 68F9 CD 5C 6A call read07

0697 68FC 32 F5 81 ld (varj),a

0698 68FF CD 5C 6A call read07

0699 6902 32 F6 81 ld (vark),a

0700 6905 CD 5C 6A call read07

0701 6908 32 F0 81 ld (vare),a

0702 690B CD 5C 6A call read07

0703 690E 32 F2 81 ld (varg),a

0704 6911 CD 5C 6A call read07

0705 6914 32 FB 81 ld (varp),a

0706 6917 CD 5C 6A call read07

0707 691A 32 EE 81 ld (varc),a

0708 691D CD 5C 6A call read07

0709 6920 32 02 82 ld (varw),a

0710 6923 CD 5C 6A call read07

0711 6926 32 F3 81 ld (varh),a

0712 6929 CD 5C 6A call read07

0713 692C 32 FC 81 ld (varq),a

0714 692F CD 5C 6A call read07

0715 6932 32 F4 81 ld (vari),a

0716 6935 21 81 5C ld hl,loopa

0717 6938 35 dec (hl)

0718 6939 C2 E7 68 jp nz,h00047

0719 693C 3A F0 81 ld a,(vare)

0720 693F 32 EF 81 ld (vard),a

0721 6942 3A F0 81 ld a,(vare)

0722 6945 32 F1 81 ld (varf),a

0723 6948 DD 7E 0B ld a,(ix+11)

0724 694B 3C inc a

0725 694C DD 77 0B ld (ix+11),a

0726 694F 3A F2 81 h00217 ld a,(varg)

0727 6952 DD BE 0C cp (ix+12)

0728 6955 C2 B6 69 jp nz,h00400

0729 6958 3A F0 81 ld a,(vare)

0730 695B DD BE 0E cp (ix+14)

0731 695E DA 91 69 jp c,h00329

0732 6961 CA 91 69 jp z,h00329

0733 6964 3E 09 ld a,9

0734 6966 21 FE 81 ld hl,vars

0735 6969 BE cp (hl)

0736 696A DA 8E 69 jp c,h00324

0737 696D CA 8E 69 jp z,h00324

0738 6970 3A F5 81 ld a,(varj)

0739 6973 4F ld c,a

0740 6974 3A F6 81 ld a,(vark)

0741 6977 47 ld b,a

0742 6978 CD 7F 8C call spawn

0743 697B DD 7E 0E ld a,(ix+14)

0744 697E 3C inc a

0745 697F DD 77 0E ld (ix+14),a

0746 6982 DD 2A CE 8C ld ix,(spptr)

0747 6986 AF xor a

0748 6987 DD 77 0A ld (ix+10),a

0749 698A DD 2A E7 92 ld ix,(ogptr)

0750 698E C3 B2 69 h00324 jp h00391

0751 6991 AF h00329 xor a

0752 6992 21 F1 81 ld hl,varf

0753 6995 BE cp (hl)

0754 6996 C2 B2 69 jp nz,h00391

0755 6999 3A F7 81 ld a,(varl)

0756 699C 3C inc a

0757 699D 32 F7 81 ld (varl),a

0758 69A0 3E 0D ld a,13

0759 69A2 21 F7 81 ld hl,varl

0760 69A5 BE cp (hl)

0761 69A6 C2 AE 69 jp nz,h00383

0762 69A9 3E 01 ld a,1

0763 69AB 32 F7 81 ld (varl),a

0764 69AE AF h00383 xor a

0765 69AF DD 77 0B ld (ix+11),a

0766 69B2 AF h00391 xor a

0767 69B3 DD 77 0C ld (ix+12),a

0768 69B6 DD 7E 0C h00400 ld a,(ix+12)

0769 69B9 3C inc a

0770 69BA DD 77 0C ld (ix+12),a

0771 69BD C9 ret

0772 69BE C0 69 rptr07 defw rdat07

0773 69C0 rdat07 defb 0,132,8,5,2,8,8,5,67,1,10

0773 69C0 008408050208080543010A

0774 69CB defb 100,8,0,100,8,5,2,10,8,10,68

0774 69CB 640800640805020A080A44

0775 69D6 defb 1,10,100,32,0,232,16,5,2,12,6

0775 69D6 010A642000E81005020C06

0776 69E1 defb 15,69,1,8,100,100,0,8,32,5,2

0776 69E1 0F45010864640008200502

0777 69EC defb 6,6,15,70,1,8,100,8,0,232,146

0777 69EC 06060F460108640800E892

0778 69F7 defb 5,2,6,6,20,71,1,4,100,8,0

0778 69F7 0502060614470104640800

0779 6A02 defb 8,146,5,2,6,6,20,66,1,4,100

0779 6A02 0892050206061442010464

0780 6A0D defb 8,0,40,8,5,2,6,8,5,69,1

0780 6A0D 0800280805020608054501

0781 6A18 defb 2,140,8,0,200,8,5,2,6,8,5

0781 6A18 028C0800C8080502060805

0782 6A23 defb 69,1,2,140,8,0,200,8,5,2,6

0782 6A23 4501028C0800C808050206

0783 6A2E defb 8,5,69,1,2,140,8,0,8,128,5

0783 6A2E 08054501028C0800088005

0784 6A39 defb 2,6,8,5,67,1,10,100,8,0,80

0784 6A39 0206080543010A64080050

0785 6A44 defb 8,5,2,6,8,5,67,1,10,100,8

0785 6A44 08050206080543010A6408

0786 6A4F defb 0,80,8,5,2,6,8,5,67,1,10

0786 6A4F 005008050206080543010A

0787 6A5A 64 08 defb 100,8

0788 6A5C DD 6E 0F read07 ld l,(ix+15)

0789 6A5F DD 66 10 ld h,(ix+16)

0790 6A62 11 5C 6A ld de,rdat07+156

0791 6A65 37 scf

0792 6A66 EB ex de,hl

0793 6A67 ED 52 sbc hl,de

0794 6A69 EB ex de,hl

0795 6A6A 30 03 jr nc,$+5

0796 6A6C 21 C0 69 ld hl,rdat07

0797 6A6F 7E ld a,(hl)

0798 6A70 23 inc hl

0799 6A71 DD 75 0F ld (ix+15),l

0800 6A74 DD 74 10 ld (ix+16),h

0801 6A77 C9 ret

0802 6A78 C9 ret

0803 6A79 evnt08 equ $

0804 6A79 C9 ret

0805 6A7A evnt09 equ $

0806 6A7A 3A FE 81 ld a,(vars)

0807 6A7D 3C inc a

0808 6A7E 32 FE 81 ld (vars),a

0809 6A81 3E 01 ld a,1

0810 6A83 DD BE 05 cp (ix+5)

0811 6A86 C2 8D 6A jp nz,j00037

0812 6A89 AF xor a

0813 6A8A DD 77 0C ld (ix+12),a

0814 6A8D 3E 05 j00037 ld a,5

0815 6A8F DD BE 05 cp (ix+5)

0816 6A92 C2 99 6A jp nz,j00058

0817 6A95 AF xor a

0818 6A96 DD 77 0B ld (ix+11),a

0819 6A99 3E 07 j00058 ld a,7

0820 6A9B DD BE 05 cp (ix+5)

0821 6A9E C2 A5 6A jp nz,j00079

0822 6AA1 AF xor a

0823 6AA2 DD 77 0B ld (ix+11),a

0824 6AA5 C9 j00079 ret

0825 6AA6 evnt10 equ $

0826 6AA6 3E 03 ld a,3 STAR DOWN

0827 6AA8 4F ld c,a

0828 6AA9 CD 38 88 call qrand

0829 6AAC E6 03 and 3

0830 6AAE CC 8D 87 call z,star

0831 6AB1 3E 01 ld a,1 IF T = 1

0832 6AB3 21 FF 81 ld hl,vart

0833 6AB6 BE cp (hl)

0834 6AB7 C2 CC 6A jp nz,k00069

0835 6ABA 21 00 01 ld hl,256 AT 0 1

0836 6ABD 22 05 82 ld (charx),hl

0837 6AC0 21 D1 8C ld hl,score SHOWSCORE

0838 6AC3 06 06 ld b,6

0839 6AC5 CD EB 8C call dscor

0840 6AC8 AF xor a LET T 0

0841 6AC9 32 FF 81 ld (vart),a

0842 6ACC 3A DF 81 k00069 ld a,(joyval) IF KEY FIRE3

0843 6ACF E6 40 and 64

0844 6AD1 CA E5 6A jp z,k00110

0845 6AD4 3A DF 81 k00083 ld a,(joyval) WHILE KEY FIRE3

0846 6AD7 E6 40 and 64

0847 6AD9 CA E2 6A jp z,k00105

0848 6ADC CD 04 91 call joykey

0849 6ADF C3 D4 6A jp k00083

0850 6AE2 CD DA 83 k00105 call prskey WAITKEY

0851 6AE5 C9 k00110 ret

0852 6AE6 evnt11 equ $

0853 6AE6 C9 ret

0854 6AE7 evnt12 equ $

0855 6AE7 CD DA 83 call prskey

0856 6AEA AF xor a

0857 6AEB CD 9B 22 call 8859

0858 6AEE 3E 44 ld a,68

0859 6AF0 32 8D 5C ld (23693),a

0860 6AF3 CD CC 84 call cls

0861 6AF6 3E 63 ld a,99

0862 6AF8 32 04 82 ld (contrl),a

0863 6AFB 3E 63 m00037 ld a,99

0864 6AFD 21 04 82 ld hl,contrl

0865 6B00 BE cp (hl)

0866 6B01 28 03 jr z,$+5

0867 6B03 D2 35 6C jp nc,m00584

0868 6B06 3E 01 ld a,1

0869 6B08 32 B7 91 ld (prtmod),a

0870 6B0B 21 04 06 ld hl,1540

0871 6B0E 22 05 82 ld (charx),hl

0872 6B11 AF xor a

0873 6B12 CD 71 91 call dmsg

0874 6B15 21 12 05 ld hl,1298

0875 6B18 22 05 82 ld (charx),hl

0876 6B1B 3E 01 ld a,1

0877 6B1D CD 71 91 call dmsg

0878 6B20 AF xor a

0879 6B21 32 B7 91 ld (prtmod),a

0880 6B24 21 08 0A ld hl,2568

0881 6B27 22 05 82 ld (charx),hl

0882 6B2A 3E 02 ld a,2

0883 6B2C CD 71 91 call dmsg

0884 6B2F 21 0A 0A ld hl,2570

0885 6B32 22 05 82 ld (charx),hl

0886 6B35 3E 03 ld a,3

0887 6B37 CD 71 91 call dmsg

0888 6B3A 21 0C 0A ld hl,2572

0889 6B3D 22 05 82 ld (charx),hl

0890 6B40 3E 04 ld a,4

0891 6B42 CD 71 91 call dmsg

0892 6B45 21 0E 0A ld hl,2574

0893 6B48 22 05 82 ld (charx),hl

0894 6B4B 3E 05 ld a,5

0895 6B4D CD 71 91 call dmsg

0896 6B50 3E 63 ld a,99

0897 6B52 32 04 82 ld (contrl),a

0898 6B55 3E 63 m00189 ld a,99

0899 6B57 21 04 82 ld hl,contrl

0900 6B5A BE cp (hl)

0901 6B5B C2 32 6C jp nz,m00580

0902 6B5E 3A D2 81 ld a,(keys+7)

0903 6B61 CD E8 90 call ktest

0904 6B64 DA 6B 6B jp c,m00230

0905 6B67 AF xor a

0906 6B68 32 04 82 ld (contrl),a

0907 6B6B 3A D3 81 m00230 ld a,(keys+8)

0908 6B6E CD E8 90 call ktest

0909 6B71 DA 79 6B jp c,m00253

0910 6B74 3E 01 ld a,1

0911 6B76 32 04 82 ld (contrl),a

0912 6B79 3A D4 81 m00253 ld a,(keys+9)

0913 6B7C CD E8 90 call ktest

0914 6B7F DA 87 6B jp c,m00276

0915 6B82 3E 02 ld a,2

0916 6B84 32 04 82 ld (contrl),a

0917 6B87 3A D5 81 m00276 ld a,(keys+10)

0918 6B8A CD E8 90 call ktest

0919 6B8D DA 2F 6C jp c,m00575

0920 6B90 CD CC 84 call cls

0921 6B93 3A D5 81 m00295 ld a,(keys+10) WHILE KEY OPTION4

0922 6B96 CD E8 90 call ktest

0923 6B99 DA 9F 6B jp c,m00314

0924 6B9C C3 93 6B jp m00295

ENDWHILE

0925 6B9F 21 07 0C m00314 ld hl,3079 AT 7,12

0926 6BA2 22 05 82 ld (charx),hl

0927 6BA5 3E 06 ld a,6

0928 6BA7 CD 71 91 call dmsg

0929 6BAA 21 09 0C ld hl,3081

0930 6BAD 22 05 82 ld (charx),hl

0931 6BB0 3E 07 ld a,7 PRINT 7

0932 6BB2 CD 71 91 call dmsg

0933 6BB5 CD 8E 02 call 654 DEFINEKEY LEFT

0934 6BB8 1C inc e

0935 6BB9 28 FA jr z,$-4

0936 6BBB 21 CC 81 ld hl,keys+1

0937 6BBE 1D dec e

0938 6BBF 73 ld (hl),e

0939 6BC0 CD E6 83 call debkey

0940 6BC3 21 0B 0C ld hl,3083 AT 11 12

0941 6BC6 22 05 82 ld (charx),hl

0942 6BC9 3E 08 ld a,8 PRINT 8

0943 6BCB CD 71 91 call dmsg

0944 6BCE CD 8E 02 call 654 DEFINEKEY RIGHT

0945 6BD1 1C inc e

0946 6BD2 28 FA jr z,$-4

0947 6BD4 21 CB 81 ld hl,keys+0

0948 6BD7 1D dec e

0949 6BD8 73 ld (hl),e

0950 6BD9 CD E6 83 call debkey

0951 6BDC 21 0D 0C ld hl,3085

0952 6BDF 22 05 82 ld (charx),hl

0953 6BE2 3E 09 ld a,9

0954 6BE4 CD 71 91 call dmsg

0955 6BE7 CD 8E 02 call 654

0956 6BEA 1C inc e

0957 6BEB 28 FA jr z,$-4

0958 6BED 21 CE 81 ld hl,keys+3

0959 6BF0 1D dec e

0960 6BF1 73 ld (hl),e

0961 6BF2 CD E6 83 call debkey

0962 6BF5 21 0F 0C ld hl,3087

0963 6BF8 22 05 82 ld (charx),hl

0964 6BFB 3E 0A ld a,10

0965 6BFD CD 71 91 call dmsg

0966 6C00 CD 8E 02 call 654

0967 6C03 1C inc e

0968 6C04 28 FA jr z,$-4

0969 6C06 21 CF 81 ld hl,keys+4

0970 6C09 1D dec e

0971 6C0A 73 ld (hl),e

0972 6C0B CD E6 83 call debkey

0973 6C0E 21 11 0C ld hl,3089

0974 6C11 22 05 82 ld (charx),hl

0975 6C14 3E 0B ld a,11

0976 6C16 CD 71 91 call dmsg

0977 6C19 CD 8E 02 call 654

0978 6C1C 1C inc e

0979 6C1D 28 FA jr z,$-4

0980 6C1F 21 D1 81 ld hl,keys+6

0981 6C22 1D dec e

0982 6C23 73 ld (hl),e

0983 6C24 CD E6 83 call debkey

0984 6C27 3E 64 ld a,100

0985 6C29 32 04 82 ld (contrl),a

0986 6C2C CD CC 84 call cls

0987 6C2F C3 55 6B m00575 jp m00189

0988 6C32 C3 FB 6A m00580 jp m00037

0989 6C35 CD CC 84 m00584 call cls

0990 6C38 C9 ret

0991 6C39 evnt13 equ $

0992 6C39 3E 01 ld a,1

0993 6C3B 32 F7 81 ld (varl),a

0994 6C3E 3E 03 ld a,3

0995 6C40 32 EB 81 ld (numlif),a

0996 6C43 21 00 10 ld hl,4096

0997 6C46 22 05 82 ld (charx),hl

0998 6C49 3A EB 81 ld a,(numlif)

0999 6C4C CD 13 95 call disply

1000 6C4F 21 00 01 ld hl,256

1001 6C52 22 05 82 ld (charx),hl

1002 6C55 21 D1 8C ld hl,score

1003 6C58 06 06 ld b,6

1004 6C5A CD EB 8C call dscor

1005 6C5D 21 00 19 ld hl,6400

1006 6C60 22 05 82 ld (charx),hl

1007 6C63 21 D7 8C ld hl,hiscor

1008 6C66 06 06 ld b,6

1009 6C68 CD EB 8C call dscor

1010 6C6B C9 ret

1011 6C6C evnt14 equ $

1012 6C6C AF xor a

1013 6C6D 32 FE 81 ld (vars),a

1014 6C70 21 00 01 ld hl,256

1015 6C73 22 05 82 ld (charx),hl

1016 6C76 21 D1 8C ld hl,score

1017 6C79 06 06 ld b,6

1018 6C7B CD EB 8C call dscor

1019 6C7E 21 0A 0C ld hl,3082

1020 6C81 22 05 82 ld (charx),hl

1021 6C84 3E 0A ld a,10

1022 6C86 32 81 5C ld (loopa),a

1023 6C89 CD BB 6C o00052 call read14

1024 6C8C 32 EC 81 ld (vara),a

1025 6C8F 3A EC 81 ld a,(vara)

1026 6C92 CD 19 92 call achar

1027 6C95 DD E5 push ix

1028 6C97 06 05 ld b,5

1029 6C99 CD F0 83 call delay

1030 6C9C DD E1 pop ix

1031 6C9E 21 81 5C ld hl,loopa

1032 6CA1 35 dec (hl)

1033 6CA2 C2 89 6C jp nz,o00052

1034 6CA5 DD E5 push ix

1035 6CA7 06 64 ld b,100

1036 6CA9 CD F0 83 call delay

1037 6CAC DD E1 pop ix

1038 6CAE C9 ret

1039 6CAF B1 6C rptr14 defw rdat14

1040 6CB1 rdat14 defb 71,69,84,32,82,69,65,68,89,33

1040 6CB1 47455420524541445921

1041 6CBB 2A AF 6C read14 ld hl,(rptr14)

1042 6CBE 11 BB 6C ld de,rdat14+10

1043 6CC1 37 scf

1044 6CC2 EB ex de,hl

1045 6CC3 ED 52 sbc hl,de

1046 6CC5 EB ex de,hl

1047 6CC6 30 03 jr nc,$+5

1048 6CC8 21 B1 6C ld hl,rdat14

1049 6CCB 7E ld a,(hl)

1050 6CCC 23 inc hl

1051 6CCD 22 AF 6C ld (rptr14),hl

1052 6CD0 C9 ret

1053 6CD1 C9 ret

1054 6CD2 evnt15 equ $

1055 6CD2 C9 ret

1056 6CD3 evnt16 equ $

1057 6CD3 3A EB 81 ld a,(numlif)

1058 6CD6 3D dec a

1059 6CD7 32 EB 81 ld (numlif),a

1060 6CDA 21 00 10 ld hl,4096

1061 6CDD 22 05 82 ld (charx),hl

1062 6CE0 3A EB 81 ld a,(numlif)

1063 6CE3 CD 13 95 call disply

1064 6CE6 CD 55 97 call clw

1065 6CE9 CD 56 88 call inishr

1066 6CEC C9 ret

1067 6CED evnt17 equ $

1068 6CED 21 0A 0C ld hl,3082 AT 10,12

1069 6CF0 22 05 82 ld (charx),hl

1070 6CF3 3E 0A ld a,10 REPEAT 10

1071 6CF5 32 81 5C ld (loopa),a

1072 6CF8 CD 2A 6D r00021 call read17 READ A

1073 6CFB 32 EC 81 ld (vara),a

1074 6CFE 3A EC 81 ld a,(vara) CHR A

1075 6D01 CD 19 92 call achar

1076 6D04 DD E5 push ix DELAY 5

1077 6D06 06 05 ld b,5

1078 6D08 CD F0 83 call delay

1079 6D0B DD E1 pop ix

1080 6D0D 21 81 5C ld hl,loopa ENDREPEAT

1081 6D10 35 dec (hl)

1082 6D11 C2 F8 6C jp nz,r00021

1083 6D14 DD E5 push ix DELAY 100

1084 6D16 06 64 ld b,100

1085 6D18 CD F0 83 call delay

1086 6D1B DD E1 pop ix

1087 6D1D C9 ret

1088 6D1E 20 6D rptr17 defw rdat17

1089 6D20 rdat17 defb 71,65,77,69,32,79,86,69,82,33

1089 6D20 47414D45204F56455221

1090 6D2A 2A 1E 6D read17 ld hl,(rptr17)

1091 6D2D 11 2A 6D ld de,rdat17+10

1092 6D30 37 scf

1093 6D31 EB ex de,hl

1094 6D32 ED 52 sbc hl,de

1095 6D34 EB ex de,hl

1096 6D35 30 03 jr nc,$+5

1097 6D37 21 20 6D ld hl,rdat17

1098 6D3A 7E ld a,(hl)

1099 6D3B 23 inc hl

1100 6D3C 22 1E 6D ld (rptr17),hl

1101 6D3F C9 ret

1102 6D40 C9 ret

1103 6D41 evnt18 equ $

1104 6D41 C9 ret

1105 6D42 evnt19 equ $

1106 6D42 C9 ret

1107 6D43 evnt20 equ $

1108 6D43 C9 ret

1109 6D44 C9 ptcusr ret

1110 6D45 msgdat equ $

1111 6D45 defb 'MPAGD SHMUPKIT DEM',207

1111 6D45 4D504147442053484D55504B49542044454DCF

1112 6D58 defb 'minilopretro.com 202',178

1112 6D58 6D696E696C6F70726574726F2E636F6D20323032B2

1113 6D6D defb '1. KEYBOAR',196

1113 6D6D 312E204B4559424F4152C4

1114 6D78 defb '2. KEMPSTO',206

1114 6D78 322E204B454D5053544FCE

1115 6D83 defb '3. SINCLAI',210

1115 6D83 332E2053494E434C4149D2

1116 6D8E defb '4. REDEFIN',197

1116 6D8E 342E205245444546494EC5

1117 6D99 defb 'PRESS KEY FOR',186

1117 6D99 5052455353204B455920464F52BA

1118 6DA7 4C 45 46 D4 defb 'LEF',212

1119 6DAB 52494748D4 defb 'RIGH',212

1120 6DB0 defb 'THRUS',212

1120 6DB0 5448525553D4

1121 6DB6 46 49 52 C5 defb 'FIR',197

1122 6DBA 50415553C5 defb 'PAUS',197

1123 6DBF 0C nummsg defb 12

1124 6DC0 scdat equ $

1125 6DC0 73 00 defw 115

1126 6DC2 defb 1,255,0,30,1,1,255,0,30,1,1,255,0,30,1,1,255,0,30,1,1,255,0,30,1,1,255,0,30,1,1,255,0,30

1126 6DC2 01FF001E0101FF001E0101FF001E0101FF001E0101FF001E0101FF001E0101FF

1126 6DE2 001E

1127 6DE4 defb 1,1,255,0,30,1,1,255,0,30,1,1,255,0,30,1,1,255,0,30,1,1,255,0,30,1,1,255,0,30,1,1,255,0,30

1127 6DE4 0101FF001E0101FF001E0101FF001E0101FF001E0101FF001E0101FF001E0101

1127 6E04 FF001E

1128 6E07 defb 1,1,255,0,30,1,1,255,0,30,1,1,255,0,30,1,1,255,0,30,1,1,255,0,30,1,1,255,0,30,1,1,255,0,30

1128 6E07 0101FF001E0101FF001E0101FF001E0101FF001E0101FF001E0101FF001E0101

1128 6E27 FF001E

1129 6E2A defb 1,1,255,0,30,1,1,255,0,30,1

1129 6E2A 0101FF001E0101FF001E01

1130 6E35 01 numsc defb 1

1131 6E36 chgfx equ $

1132 6E36 defb 0,0,0,0,0,0,0,0

1132 6E36 0000000000000000

1133 6E3E defb 255,128,255,1,255,128,255,1

1133 6E3E FF80FF01FF80FF01

1134 6E46 bcol equ $

1135 6E46 47 defb 71

1136 6E47 47 defb 71

1137 6E48 bprop equ $

1138 6E48 00 defb 0

1139 6E49 02 defb 2

1140 6E4A sprgfx equ $

1141 6E4A defb 1,0,3,0,2,128,2,0,7,64,10,32,139,34,7,64,135,66,139,34,147,18,199,34,207,22,216,34,208,22,128,2

1141 6E4A 010003000280020007400A208B22074087428B229312C722CF16D822D0168002

1142 6E6A defb 0,64,0,192,0,160,0,128,1,208,2,136,162,200,1,208,161,208,162,200,164,196,177,200,179,197,182,8,180,5,160,0

1142 6E6A 004000C000A0008001D00288A2C801D0A1D0A2C8A4C4B1C8B3C5B608B405A000

1143 6E8A defb 0,16,0,48,0,40,0,32,0,116,0,162,40,178,0,116,40,116,40,178,41,49,44,114,108,241,45,130,109,1,40,0

1143 6E8A 0010003000280020007400A228B20074287428B229312C726CF12D826D012800

1144 6EAA defb 0,4,0,12,0,10,0,8,0,29,128,40,138,44,0,29,10,29,138,44,74,76,139,28,91,60,139,96,91,64,10,0

1144 6EAA 0004000C000A0008001D80288A2C001D0A1D8A2C4A4C8B1C5B3C8B605B400A00

1145 6ECA defb 1,0,3,0,2,128,2,0,7,64,10,32,139,34,7,64,135,66,139,34,147,18,199,34,207,22,216,34,210,150,129,2

1145 6ECA 010003000280020007400A208B22074087428B229312C722CF16D822D2968102

1146 6EEA defb 0,64,0,192,0,160,0,128,1,208,2,136,162,200,1,208,161,208,162,200,164,196,177,200,179,197,182,8,180,165,160,64

1146 6EEA 004000C000A0008001D00288A2C801D0A1D0A2C8A4C4B1C8B3C5B608B4A5A040

1147 6F0A defb 0,16,0,48,0,40,0,32,0,116,0,162,40,178,0,116,40,116,40,178,41,49,44,114,108,241,45,130,109,41,40,16

1147 6F0A 0010003000280020007400A228B20074287428B229312C726CF12D826D292810

1148 6F2A defb 0,4,0,12,0,10,0,8,0,29,128,40,138,44,0,29,10,29,138,44,74,76,139,28,91,60,139,96,91,74,10,4

1148 6F2A 0004000C000A0008001D80288A2C001D0A1D8A2C4A4C8B1C5B3C8B605B4A0A04

1149 6F4A defb 64,0,64,0,64,2,64,2,160,2,64,2,0,5,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

1149 6F4A 4000400040024002A00240020005000200000000000000000000000000000000

1150 6F6A defb 16,0,16,0,144,0,144,0,168,0,144,0,64,1,128,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

1150 6F6A 1000100090009000A80090004001800000000000000000000000000000000000

1151 6F8A defb 4,0,4,0,36,0,36,0,42,0,36,0,80,0,32,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

1151 6F8A 04000400240024002A0024005000200000000000000000000000000000000000

1152 6FAA defb 1,0,1,0,9,0,9,0,10,128,9,0,20,0,8,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

1152 6FAA 01000100090009000A8009001400080000000000000000000000000000000000

1153 6FCA defb 0,0,0,0,0,0,0,0,32,0,112,0,32,0,0,4,0,14,0,4,0,0,0,0,0,0,0,0,0,0,0,0

1153 6FCA 00000000000000002000700020000004000E0004000000000000000000000000

1154 6FEA defb 0,0,0,0,0,0,0,0,8,0,28,0,8,0,0,1,128,3,0,1,0,0,0,0,0,0,0,0,0,0,0,0

1154 6FEA 000000000000000008001C000800000180030001000000000000000000000000

1155 700A defb 0,0,0,0,0,0,0,0,2,0,7,0,2,0,64,0,224,0,64,0,0,0,0,0,0,0,0,0,0,0,0,0

1155 700A 00000000000000000200070002004000E0004000000000000000000000000000

1156 702A defb 0,0,0,0,0,0,0,0,0,128,1,192,0,128,16,0,56,0,16,0,0,0,0,0,0,0,0,0,0,0,0,0

1156 702A 0000000000000000008001C00080100038001000000000000000000000000000

1157 704A defb 0,0,0,0,0,0,32,0,80,0,136,0,80,4,32,10,0,17,0,10,0,4,0,0,0,0,0,0,0,0,0,0

1157 704A 0000000000002000500088005004200A0011000A000400000000000000000000

1158 706A defb 0,0,0,0,0,0,8,0,20,0,34,0,20,1,136,2,64,4,128,2,0,1,0,0,0,0,0,0,0,0,0,0

1158 706A 0000000000000800140022001401880240048002000100000000000000000000

1159 708A defb 0,0,0,0,0,0,2,0,5,0,8,128,69,0,162,0,16,1,160,0,64,0,0,0,0,0,0,0,0,0,0,0

1159 708A 0000000000000200050008804500A2001001A000400000000000000000000000

1160 70AA defb 0,0,0,0,0,0,0,128,1,64,2,32,17,64,40,128,68,0,40,0,16,0,0,0,0,0,0,0,0,0,0,0

1160 70AA 0000000000000080014002201140288044002800100000000000000000000000

1161 70CA defb 0,0,32,0,80,0,136,0,4,0,132,4,136,10,80,17,32,33,0,17,0,10,0,4,0,0,0,0,0,0,0,0

1161 70CA 000020005000880004008404880A501120210011000A00040000000000000000

1162 70EA defb 0,0,8,0,20,0,34,0,1,0,33,1,162,2,84,4,72,8,64,4,128,2,0,1,0,0,0,0,0,0,0,0

1162 70EA 000008001400220001002101A202540448084004800200010000000000000000

1163 710A defb 0,0,2,0,5,0,8,128,0,64,72,64,168,128,21,1,18,2,16,1,160,0,64,0,0,0,0,0,0,0,0,0

1163 710A 000002000500088000404840A880150112021001A00040000000000000000000

1164 712A defb 0,0,0,128,1,64,2,32,0,16,18,16,42,32,69,64,132,128,68,0,40,0,16,0,0,0,0,0,0,0,0,0

1164 712A 0000008001400220001012102A20454084804400280010000000000000000000

1165 714A defb 32,0,136,0,2,0,128,4,1,18,128,0,66,32,0,1,4,0,32,65,8,0,0,32,0,2,0,8,0,0,0,0

1165 714A 2000880002008004011280004220000104002041080000200002000800000000

1166 716A defb 8,0,34,0,0,128,32,1,128,68,32,0,16,136,64,0,1,0,72,16,2,0,0,8,128,0,0,2,0,0,0,0

1166 716A 0800220000802001804420001088400001004810020000088000000200000000

1167 718A defb 2,0,8,128,0,32,72,0,32,17,8,0,4,34,16,0,0,64,18,4,0,128,0,2,32,0,128,0,0,0,0,0

1167 718A 0200088000204800201108000422100000401204008000022000800000000000

1168 71AA defb 0,128,2,32,0,8,18,0,72,4,2,0,129,8,4,0,0,16,4,129,0,32,128,0,8,0,32,0,0,0,0,0

1168 71AA 0080022000081200480402008108040000100481002080000800200000000000

1169 71CA defb 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

1169 71CA 0000000000000000000000000000000000000000000000000000000000000000

1170 71EA defb 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

1170 71EA 0000000000000000000000000000000000000000000000000000000000000000

1171 720A defb 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

1171 720A 0000000000000000000000000000000000000000000000000000000000000000

1172 722A defb 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

1172 722A 0000000000000000000000000000000000000000000000000000000000000000

1173 724A defb 0,128,0,128,1,192,1,192,9,200,1,192,11,232,75,233,11,104,70,49,78,185,95,253,123,239,117,215,106,171,64,129

1173 724A 0080008001C001C009C801C00BE84BE90B6846314EB95FFD7BEF75D76AAB4081

1174 726A defb 0,32,0,32,0,112,0,112,2,114,0,112,2,250,82,250,2,218,81,140,83,174,87,255,222,251,221,117,218,170,80,32

1174 726A 00200020007000700272007002FA52FA02DA518C53AE57FFDEFBDD75DAAA5020

1175 728A defb 0,8,0,8,0,28,0,28,128,156,0,28,128,190,148,190,128,182,20,99,148,235,213,255,247,190,119,93,182,170,20,8

1175 728A 00080008001C001C809C001C80BE94BE80B6146394EBD5FFF7BE775DB6AA1408

1176 72AA defb 0,2,0,2,0,7,0,7,32,39,0,7,160,47,165,47,160,45,197,24,229,58,245,127,189,239,93,215,173,170,5,2

1176 72AA 000200020007000720270007A02FA52FA02DC518E53AF57FBDEF5DD7ADAA0502

1177 72CA defb 0,0,0,8,0,24,2,48,4,240,3,240,21,224,46,40,127,48,255,160,135,224,23,244,7,120,4,176,0,48,0,32

1177 72CA 000000080018023004F003F015E02E287F30FFA087E017F4077804B000300020

1178 72EA defb 0,0,0,2,0,6,0,140,1,60,0,252,5,120,11,138,31,204,63,232,33,248,5,253,1,222,1,44,0,12,0,8

1178 72EA 000000020006008C013C00FC05780B8A1FCC3FE821F805FD01DE012C000C0008

1179 730A defb 0,0,128,0,128,1,0,35,0,79,0,63,1,94,130,226,7,243,15,250,8,126,65,127,128,119,0,75,0,3,0,2

1179 730A 0000800080010023004F003F015E82E207F30FFA087E417F8077004B00030002

1180 732A defb 0,0,32,0,96,0,192,8,192,19,192,15,128,87,160,184,193,252,131,254,130,31,208,95,224,29,192,18,192,0,128,0

1180 732A 000020006000C008C013C00F8057A0B8C1FC83FE821FD05FE01DC012C0008000

1181 734A defb 0,0,0,0,0,4,9,8,18,48,36,112,127,224,254,64,15,72,39,208,23,224,9,200,20,208,2,224,0,192,0,128

1181 734A 0000000000040908123024707FE0FE400F4827D017E009C814D002E000C00080

1182 736A defb 0,0,0,0,0,1,2,66,4,140,9,28,31,248,63,144,3,210,9,244,5,248,2,114,5,52,0,184,0,48,0,32

1182 736A 0000000000010242048C091C1FF83F9003D209F405F80272053400B800300020

1183 738A defb 0,0,0,0,64,0,128,144,1,35,2,71,7,254,15,228,128,244,2,125,1,126,128,156,1,77,0,46,0,12,0,8

1183 738A 00000000400080900123024707FE0FE480F4027D017E809C014D002E000C0008

1184 73AA defb 0,0,0,0,16,0,32,36,192,72,192,145,129,255,3,249,32,61,64,159,128,95,32,39,64,83,128,11,0,3,0,2

1184 73AA 0000000010002024C048C09181FF03F9203D409F805F20274053800B00030002

1185 73CA defb 0,0,0,0,8,0,16,134,121,60,255,248,14,112,7,112,15,224,15,208,29,224,1,192,1,160,1,192,1,128,3,0

1185 73CA 0000000008001086793CFFF80E7007700FE00FD01DE001C001A001C001800300

1186 73EA defb 0,0,0,0,2,0,132,33,30,79,63,254,3,156,1,220,3,248,3,244,7,120,0,112,0,104,0,112,0,96,0,192

1186 73EA 00000000020084211E4F3FFE039C01DC03F803F40778007000680070006000C0

1187 740A defb 0,0,0,0,0,128,97,8,199,147,143,255,0,231,0,119,0,254,0,253,1,222,0,28,0,26,0,28,0,24,0,48

1187 740A 0000000000806108C7938FFF00E7007700FE00FD01DE001C001A001C00180030

1188 742A defb 0,0,0,0,0,32,24,66,241,228,227,255,192,57,192,29,128,63,64,63,128,119,0,7,128,6,0,7,0,6,0,12

1188 742A 0000000000201842F1E4E3FFC039C01D803F403F80770007800600070006000C

1189 744A defb 0,0,254,128,112,0,56,0,93,208,46,0,95,192,57,252,252,255,57,252,95,192,46,0,93,208,56,0,112,0,254,128

1189 744A 0000FE80700038005DD02E005FC039FCFCFF39FC5FC02E005DD038007000FE80

1190 746A defb 0,0,63,160,28,0,14,0,23,116,11,128,23,240,14,127,255,63,14,127,23,240,11,128,23,116,14,0,28,0,63,160

1190 746A 00003FA01C000E0017740B8017F00E7FFF3F0E7F17F00B8017740E001C003FA0

1191 748A defb 0,0,15,232,7,0,3,128,5,221,2,224,5,252,195,159,255,207,195,159,5,252,2,224,5,221,3,128,7,0,15,232

1191 748A 00000FE80700038005DD02E005FCC39FFFCFC39F05FC02E005DD038007000FE8

1192 74AA defb 0,0,3,250,1,192,0,224,65,119,0,184,1,127,240,231,255,243,240,231,1,127,0,184,65,119,0,224,1,192,3,250

1192 74AA 000003FA01C000E0417700B8017FF0E7FFF3F0E7017F00B8417700E001C003FA

1193 74CA defb 3,0,1,128,1,192,1,160,1,192,29,224,15,208,15,224,7,112,14,112,255,248,121,60,16,134,8,0,0,0,0,0

1193 74CA 0300018001C001A001C01DE00FD00FE007700E70FFF8793C1086080000000000

1194 74EA defb 0,192,0,96,0,112,0,104,0,112,7,120,3,244,3,248,1,220,3,156,63,254,30,79,132,33,2,0,0,0,0,0

1194 74EA 00C00060007000680070077803F403F801DC039C3FFE1E4F8421020000000000

1195 750A defb 0,48,0,24,0,28,0,26,0,28,1,222,0,253,0,254,0,119,0,231,143,255,199,147,97,8,0,128,0,0,0,0

1195 750A 00300018001C001A001C01DE00FD00FE007700E78FFFC7936108008000000000

1196 752A defb 0,12,0,6,0,7,128,6,0,7,128,119,64,63,128,63,192,29,192,57,227,255,241,228,24,66,0,32,0,0,0,0

1196 752A 000C00060007800600078077403F803FC01DC039E3FFF1E41842002000000000

1197 754A defb 0,128,0,192,2,224,20,208,9,200,23,224,39,208,15,72,254,64,127,224,36,112,18,48,9,8,0,4,0,0,0,0

1197 754A 008000C002E014D009C817E027D00F48FE407FE0247012300908000400000000

1198 756A defb 0,32,0,48,0,184,5,52,2,114,5,248,9,244,3,210,63,144,31,248,9,28,4,140,2,66,0,1,0,0,0,0

1198 756A 0020003000B80534027205F809F403D23F901FF8091C048C0242000100000000

1199 758A defb 0,8,0,12,0,46,1,77,128,156,1,126,2,125,128,244,15,228,7,254,2,71,1,35,128,144,64,0,0,0,0,0

1199 758A 0008000C002E014D809C017E027D80F40FE407FE024701238090400000000000

1200 75AA defb 0,2,0,3,128,11,64,83,32,39,128,95,64,159,32,61,3,249,129,255,192,145,192,72,32,36,16,0,0,0,0,0

1200 75AA 00020003800B40532027805F409F203D03F981FFC091C0482024100000000000

1201 75CA defb 0,32,0,48,4,176,7,120,23,244,135,224,255,160,127,48,46,40,21,224,3,240,4,240,2,48,0,24,0,8,0,0

1201 75CA 0020003004B0077817F487E0FFA07F302E2815E003F004F00230001800080000

1202 75EA defb 0,8,0,12,1,44,1,222,5,253,33,248,63,232,31,204,11,138,5,120,0,252,1,60,0,140,0,6,0,2,0,0

1202 75EA 0008000C012C01DE05FD21F83FE81FCC0B8A057800FC013C008C000600020000

1203 760A defb 0,2,0,3,0,75,128,119,65,127,8,126,15,250,7,243,130,226,1,94,0,63,0,79,0,35,128,1,128,0,0,0

1203 760A 00020003004B8077417F087E0FFA07F382E2015E003F004F0023800180000000

1204 762A defb 128,0,192,0,192,18,224,29,208,95,130,31,131,254,193,252,160,184,128,87,192,15,192,19,192,8,96,0,32,0,0,0

1204 762A 8000C000C012E01DD05F821F83FEC1FCA0B88057C00FC013C008600020000000

1205 764A defb 64,129,106,171,117,215,123,239,95,253,78,185,70,49,11,104,75,233,11,232,1,192,9,200,1,192,1,192,0,128,0,128

1205 764A 40816AAB75D77BEF5FFD4EB946310B684BE90BE801C009C801C001C000800080

1206 766A defb 80,32,218,170,221,117,222,251,87,255,83,174,81,140,2,218,82,250,2,250,0,112,2,114,0,112,0,112,0,32,0,32

1206 766A 5020DAAADD75DEFB57FF53AE518C02DA52FA02FA007002720070007000200020

1207 768A defb 20,8,182,170,119,93,247,190,213,255,148,235,20,99,128,182,148,190,128,190,0,28,128,156,0,28,0,28,0,8,0,8

1207 768A 1408B6AA775DF7BED5FF94EB146380B694BE80BE001C809C001C001C00080008

1208 76AA defb 5,2,173,170,93,215,189,239,245,127,229,58,197,24,160,45,165,47,160,47,0,7,32,39,0,7,0,7,0,2,0,2

1208 76AA 0502ADAA5DD7BDEFF57FE53AC518A02DA52FA02F000720270007000700020002

1209 76CA defb 4,0,12,0,13,32,30,224,47,232,7,225,5,255,12,254,20,116,7,168,15,192,15,32,12,64,24,0,16,0,0,0

1209 76CA 04000C000D201EE02FE807E105FF0CFE147407A80FC00F200C40180010000000

1210 76EA defb 1,0,3,0,3,72,7,184,11,250,65,248,193,127,131,63,5,29,1,234,3,240,3,200,3,16,6,0,4,0,0,0

1210 76EA 01000300034807B80BFA41F8C17F833F051D01EA03F003C80310060004000000

1211 770A defb 0,64,0,192,0,210,1,238,130,254,16,126,240,95,224,207,65,71,128,122,0,252,0,242,0,196,1,128,1,0,0,0

1211 770A 004000C000D201EE82FE107EF05FE0CF4147807A00FC00F200C4018001000000

1212 772A defb 0,16,0,48,128,52,128,123,160,191,132,31,252,23,248,51,208,81,160,30,0,63,128,60,0,49,0,96,0,64,0,0

1212 772A 001000308034807BA0BF841FFC17F833D051A01E003F803C0031006000400000

1213 774A defb 1,0,3,0,7,64,11,40,19,144,7,232,11,228,18,240,2,127,7,254,14,36,12,72,16,144,32,0,0,0,0,0

1213 774A 0100030007400B28139007E80BE412F0027F07FE0E240C481090200000000000

1214 776A defb 0,64,0,192,1,208,2,202,4,228,1,250,2,249,4,188,192,159,129,255,3,137,3,18,4,36,8,0,0,0,0,0

1214 776A 004000C001D002CA04E401FA02F904BCC09F81FF038903120424080000000000

1215 778A defb 0,16,0,48,0,116,128,178,1,57,128,126,64,190,1,47,240,39,224,127,64,226,128,196,1,9,2,0,0,0,0,0

1215 778A 00100030007480B20139807E40BE012FF027E07F40E280C40109020000000000

1216 77AA defb 0,4,0,12,0,29,160,44,64,78,160,31,144,47,192,75,252,9,248,31,144,56,32,49,64,66,0,128,0,0,0,0

1216 77AA 0004000C001DA02C404EA01F902FC04BFC09F81F903820314042008000000000

1217 77CA defb 0,192,1,128,3,128,5,128,3,128,7,184,11,240,7,240,14,224,14,112,31,255,60,158,97,8,0,16,0,0,0,0

1217 77CA 00C0018003800580038007B80BF007F00EE00E701FFF3C9E6108001000000000

1218 77EA defb 0,48,0,96,0,224,1,96,0,224,1,238,2,252,1,252,3,184,3,156,199,255,143,39,24,66,0,4,0,0,0,0

1218 77EA 0030006000E0016000E001EE02FC01FC03B8039CC7FF8F271842000400000000

1219 780A defb 0,12,0,24,0,56,0,88,0,56,128,123,0,191,0,127,0,238,0,231,241,255,227,201,134,16,0,1,0,0,0,0

1219 780A 000C0018003800580038807B00BF007F00EE00E7F1FFE3C98610000100000000

1220 782A defb 0,3,0,6,0,14,0,22,0,14,224,30,192,47,192,31,128,59,192,57,252,127,120,242,33,132,64,0,0,0,0,0

1220 782A 00030006000E0016000EE01EC02FC01F803BC039FC7F78F22184400000000000

1221 784A defb 0,0,1,127,0,14,0,28,11,186,0,116,3,250,63,156,255,63,63,156,3,250,0,116,11,186,0,28,0,14,1,127

1221 784A 0000017F000E001C0BBA007403FA3F9CFF3F3F9C03FA00740BBA001C000E017F

1222 786A defb 0,0,192,95,128,3,0,7,130,238,0,29,128,254,15,231,255,207,15,231,128,254,0,29,130,238,0,7,128,3,192,95

1222 786A 0000C05F8003000782EE001D80FE0FE7FFCF0FE780FE001D82EE00078003C05F

1223 788A defb 0,0,240,23,224,0,192,1,160,187,64,7,160,63,195,249,255,243,195,249,160,63,64,7,160,187,192,1,224,0,240,23

1223 788A 0000F017E000C001A0BB4007A03FC3F9FFF3C3F9A03F4007A0BBC001E000F017

1224 78AA defb 0,0,252,5,56,0,112,0,232,46,208,1,232,15,112,254,255,252,112,254,232,15,208,1,232,46,112,0,56,0,252,5

1224 78AA 0000FC0538007000E82ED001E80F70FEFFFC70FEE80FD001E82E70003800FC05

1225 78CA defb 0,0,0,0,0,16,97,8,60,158,31,255,14,112,14,224,7,240,11,240,7,184,3,128,5,128,3,128,1,128,0,192

1225 78CA 00000000001061083C9E1FFF0E700EE007F00BF007B8038005800380018000C0

1226 78EA defb 0,0,0,0,0,4,24,66,143,39,199,255,3,156,3,184,1,252,2,252,1,238,0,224,1,96,0,224,0,96,0,48

1226 78EA 00000000000418428F27C7FF039C03B801FC02FC01EE00E0016000E000600030

1227 790A defb 0,0,0,0,0,1,134,16,227,201,241,255,0,231,0,238,0,127,0,191,128,123,0,56,0,88,0,56,0,24,0,12

1227 790A 0000000000018610E3C9F1FF00E700EE007F00BF807B0038005800380018000C

1228 792A defb 0,0,0,0,64,0,33,132,120,242,252,127,192,57,128,59,192,31,192,47,224,30,0,14,0,22,0,14,0,6,0,3

1228 792A 000000004000218478F2FC7FC039803BC01FC02FE01E000E0016000E00060003

1229 794A defb 0,0,0,0,32,0,16,144,12,72,14,36,7,254,2,127,18,240,11,228,7,232,19,144,11,40,7,64,3,0,1,0

1229 794A 00000000200010900C480E2407FE027F12F00BE407E813900B28074003000100

1230 796A defb 0,0,0,0,8,0,4,36,3,18,3,137,129,255,192,159,4,188,2,249,1,250,4,228,2,202,1,208,0,192,0,64

1230 796A 00000000080004240312038981FFC09F04BC02F901FA04E402CA01D000C00040

1231 798A defb 0,0,0,0,2,0,1,9,128,196,64,226,224,127,240,39,1,47,64,190,128,126,1,57,128,178,0,116,0,48,0,16

1231 798A 000000000200010980C440E2E07FF027012F40BE807E013980B2007400300010

1232 79AA defb 0,0,0,0,0,128,64,66,32,49,144,56,248,31,252,9,192,75,144,47,160,31,64,78,160,44,0,29,0,12,0,4

1232 79AA 000000000080404220319038F81FFC09C04B902FA01F404EA02C001D000C0004

1233 79CA defb 0,0,16,0,24,0,12,64,15,32,15,192,7,168,20,116,12,254,5,255,7,225,47,232,30,224,13,32,12,0,4,0

1233 79CA 0000100018000C400F200FC007A814740CFE05FF07E12FE81EE00D200C000400

1234 79EA defb 0,0,4,0,6,0,3,16,3,200,3,240,1,234,5,29,131,63,193,127,65,248,11,250,7,184,3,72,3,0,1,0

1234 79EA 000004000600031003C803F001EA051D833FC17F41F80BFA07B8034803000100

1235 7A0A defb 0,0,1,0,1,128,0,196,0,242,0,252,128,122,65,71,224,207,240,95,16,126,130,254,1,238,0,210,0,192,0,64

1235 7A0A 00000100018000C400F200FC807A4147E0CFF05F107E82FE01EE00D200C00040

1236 7A2A defb 0,0,0,64,0,96,0,49,128,60,0,63,160,30,208,81,248,51,252,23,132,31,160,191,128,123,128,52,0,48,0,16

1236 7A2A 0000004000600031803C003FA01ED051F833FC17841FA0BF807B803400300010

1237 7A4A defb 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

1237 7A4A 0000000000000000000000000000000000000000000000000000000000000000

1238 7A6A defb 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

1238 7A6A 0000000000000000000000000000000000000000000000000000000000000000

1239 7A8A defb 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

1239 7A8A 0000000000000000000000000000000000000000000000000000000000000000

1240 7AAA defb 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

1240 7AAA 0000000000000000000000000000000000000000000000000000000000000000

1241 7ACA defb 0,0,0,0,0,0,0,144,11,224,7,248,14,112,30,56,14,60,7,240,7,240,9,200,0,64,0,0,0,0,0,0

1241 7ACA 00000000000000900BE007F80E701E380E3C07F007F009C80040000000000000

1242 7AEA defb 0,0,0,0,0,0,0,36,2,248,1,254,3,156,7,142,3,143,1,252,1,252,2,114,0,16,0,0,0,0,0,0

1242 7AEA 000000000000002402F801FE039C078E038F01FC01FC02720010000000000000

1243 7B0A defb 0,0,0,0,0,0,0,9,0,190,128,127,0,231,129,227,192,227,0,127,0,127,128,156,0,4,0,0,0,0,0,0

1243 7B0A 000000000000000900BE807F00E781E3C0E3007F007F809C0004000000000000

1244 7B2A defb 0,0,0,0,0,0,64,2,128,47,224,31,192,57,224,120,240,56,192,31,192,31,32,39,0,1,0,0,0,0,0,0

1244 7B2A 0000000000004002802FE01FC039E078F038C01FC01F20270001000000000000

1245 7B4A defb 128,8,1,128,102,240,27,154,30,234,148,60,26,84,56,30,106,22,48,156,46,48,31,252,13,200,39,116,128,192,0,1

1245 7B4A 8008018066F01B9A1EEA943C1A54381E6A16309C2E301FFC0DC8277480C00001

1246 7B6A defb 32,2,0,96,25,188,134,230,135,186,37,15,6,149,142,7,154,133,12,39,11,140,7,255,3,114,9,221,32,48,64,0

1246 7B6A 2002006019BC86E687BA250F06958E079A850C270B8C07FF037209DD20304000

1247 7B8A defb 136,0,0,24,6,111,161,185,161,238,201,67,65,165,227,129,102,161,195,9,2,227,193,255,128,220,66,119,8,12,16,0

1247 7B8A 88000018066FA1B9A1EEC94341A5E38166A1C30902E3C1FF80DC4277080C1000

1248 7BAA defb 34,0,0,6,193,155,104,110,168,123,242,80,80,105,120,224,89,168,112,194,192,184,240,127,32,55,208,157,2,3,4,0

1248 7BAA 22000006C19B686EA87BF250506978E059A870C2C0B8F07F2037D09D02030400

1249 7BCA defb 32,8,65,128,164,49,3,136,20,2,144,32,0,8,40,17,74,4,32,18,2,32,72,212,5,9,32,84,144,128,4,9

1249 7BCA 20084180A431038814029020000828114A042012022048D40509205490800409

1250 7BEA defb 8,2,16,96,105,12,0,226,133,0,36,8,0,2,74,4,18,129,136,4,0,136,18,53,65,66,8,21,36,32,65,2

1250 7BEA 08021060690C00E28500240800024A0412818804008812354142081524204102

1251 7C0A defb 130,0,4,24,26,67,128,56,33,64,9,2,128,0,18,129,68,160,34,1,0,34,68,141,144,80,66,5,9,8,144,64

1251 7C0A 820004181A438038214009028000128144A022010022448D9050420509089040

1252 7C2A defb 32,128,1,6,198,144,32,14,8,80,130,64,32,0,68,160,17,40,72,128,128,8,81,35,36,20,80,129,2,66,36,16

1252 7C2A 20800106C690200E08508240200044A011284880800851232414508102422410

1253 7C4A defb 130,8,72,33,1,0,0,4,32,33,0,0,0,2,128,0,16,36,0,1,128,0,8,0,64,136,33,1,8,18,64,128

1253 7C4A 8208482101000004202100000002800010240001800008004088210108124080

1254 7C6A defb 32,130,82,8,0,64,0,1,72,8,0,0,128,0,32,0,4,9,64,0,32,0,2,0,16,34,72,64,130,4,16,32

1254 7C6A 2082520800400001480800008000200004094000200002001022484082041020

1255 7C8A defb 136,32,20,130,0,16,64,0,18,2,0,0,32,0,8,0,65,2,16,0,8,0,0,128,132,8,18,16,32,129,4,8

1255 7C8A 8820148200104000120200002000080041021000080000808408121020810408

1256 7CAA defb 34,8,133,32,0,4,16,0,132,128,0,0,8,0,2,0,144,64,4,0,2,0,0,32,33,2,4,132,72,32,1,2

1256 7CAA 2208852000041000848000000800020090400400020000202102048448200102

1257 7CCA defb 0,64,65,4,128,0,0,0,0,2,16,0,0,0,0,0,0,2,0,0,128,0,0,0,4,0,32,68,128,0,4,9

1257 7CCA 0040410480000000000210000000000000020000800000000400204480000409

1258 7CEA defb 0,16,16,65,32,0,0,0,128,0,4,0,0,0,0,0,128,0,0,0,32,0,0,0,1,0,8,17,32,0,65,2

1258 7CEA 0010104120000000800004000000000080000000200000000100081120004102

1259 7D0A defb 0,4,68,16,8,0,0,0,32,0,1,0,0,0,0,0,32,0,0,0,8,0,0,0,0,64,66,4,8,0,144,64

1259 7D0A 0004441008000000200001000000000020000000080000000040420408009040

1260 7D2A defb 0,1,17,4,2,0,0,0,8,0,0,64,0,0,0,0,8,0,0,0,2,0,0,0,0,16,16,129,2,0,36,16

1260 7D2A 0001110402000000080000400000000008000000020000000010108102002410

1261 7D4A defb 65,4,163,138,6,128,103,72,182,146,175,74,166,130,180,82,175,138,70,132,236,14,169,42,68,68,3,128,1,0,1,0

1261 7D4A 4104A38A06806748B692AF4AA682B452AF8A4684EC0EA92A4444038001000100

1262 7D6A defb 16,65,168,226,1,160,25,210,173,164,171,210,169,160,173,20,171,226,17,161,187,3,170,74,17,17,0,224,0,64,0,64

1262 7D6A 1041A8E201A019D2ADA4ABD2A9A0AD14ABE211A1BB03AA4A111100E000400040

1263 7D8A defb 68,16,170,56,0,104,134,116,43,105,170,244,42,104,43,69,170,248,68,104,238,192,170,146,68,68,0,56,0,16,0,16

1263 7D8A 4410AA38006886742B69AAF42A682B45AAF84468EEC0AA924444003800100010

1264 7DAA defb 17,4,42,142,0,26,33,157,74,218,42,189,10,154,74,209,42,190,17,26,59,176,170,164,17,17,0,14,0,4,0,4

1264 7DAA 11042A8E001A219D4ADA2ABD0A9A4AD12ABE111A3BB0AAA41111000E00040004

1265 7DCA defb 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,64,0,128,1,64,0,128,1,192,1,192,1,192,0,128,0,0

1265 7DCA 0000000000000000000000000000014000800140008001C001C001C000800000

1266 7DEA defb 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,80,0,32,0,80,0,32,0,112,0,112,0,112,0,32,0,0

1266 7DEA 0000000000000000000000000000005000200050002000700070007000200000

1267 7E0A defb 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,20,0,8,0,20,0,8,0,28,0,28,0,28,0,8,0,0

1267 7E0A 00000000000000000000000000000014000800140008001C001C001C00080000

1268 7E2A defb 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5,0,2,0,5,0,2,0,7,0,7,0,7,0,2,0,0

1268 7E2A 0000000000000000000000000000000500020005000200070007000700020000

1269 7E4A frmlst equ $

1270 7E4A 00 02 defb 0,2

1271 7E4C 02 06 defb 2,6

1272 7E4E 08 10 defb 8,16

1273 7E50 18 01 defb 24,1

1274 7E52 19 05 defb 25,5

1275 7E54 1E 01 defb 30,1

1276 7E56 1F 01 20 00 defb 31,1,32,0

1277 7E5A nmedat defb 0,0,144,120,7,3,8,0,255

1277 7E5A 0000907807030800FF

1278 7E63 NUMOBJ equ 1

1279 7E63 objdta equ $

1280 7E63 defb 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,71,255,8,8,255,8,8

1280 7E63 0000000000000000000000000000000000000000000000000000000000000000

1280 7E83 47FF0808FF0808

1281 7E8A palett equ $

1282 7E8A defb 0,3,28,31,224,227,252,255,0,44,156,15,248,131,190,253

1282 7E8A 00031C1FE0E3FCFF002C9C0FF883BEFD

1283 7E9A defb 0,3,28,31,224,227,252,255,0,44,156,15,248,131,190,253

1283 7E9A 00031C1FE0E3FCFF002C9C0FF883BEFD

1284 7EAA defb 0,3,28,31,224,227,252,255,0,44,156,15,248,131,190,253

1284 7EAA 00031C1FE0E3FCFF002C9C0FF883BEFD

1285 7EBA defb 0,3,28,31,224,227,252,255,0,44,156,15,248,131,190,253

1285 7EBA 00031C1FE0E3FCFF002C9C0FF883BEFD

1286 7ECA font equ $

1287 7ECA defb 0,0,0,0,0,0,0,0

1287 7ECA 0000000000000000

1288 7ED2 defb 16,16,24,24,24,0,24,0

1288 7ED2 1010181818001800

1289 7EDA defb 108,108,36,0,0,0,0,0

1289 7EDA 6C6C240000000000

1290 7EE2 defb 0,52,126,52,44,126,44,0

1290 7EE2 00347E342C7E2C00

1291 7EEA defb 0,16,124,64,124,12,124,16

1291 7EEA 00107C407C0C7C10

1292 7EF2 defb 226,174,232,120,238,206,206,0

1292 7EF2 E2AEE878EECECE00

1293 7EFA defb 124,68,108,56,110,100,126,0

1293 7EFA 7C446C386E647E00

1294 7F02 defb 24,24,8,0,0,0,0,0

1294 7F02 1818080000000000

1295 7F0A defb 8,16,32,32,32,16,8,0

1295 7F0A 0810202020100800

1296 7F12 defb 32,16,8,8,8,16,32,0

1296 7F12 2010080808102000

1297 7F1A defb 0,16,124,56,124,16,0,0

1297 7F1A 00107C387C100000

1298 7F22 defb 0,16,16,126,24,24,0,0

1298 7F22 0010107E18180000

1299 7F2A defb 0,0,0,0,0,24,24,8

1299 7F2A 0000000000181808

1300 7F32 defb 0,0,0,126,0,0,0,0

1300 7F32 0000007E00000000

1301 7F3A defb 0,0,0,0,0,24,24,0

1301 7F3A 0000000000181800

1302 7F42 defb 2,6,12,24,48,96,96,0

1302 7F42 02060C1830606000

1303 7F4A defb 126,66,66,66,66,66,126,0

1303 7F4A 7E42424242427E00

1304 7F52 defb 48,16,16,16,60,60,60,0

1304 7F52 301010103C3C3C00

1305 7F5A defb 60,4,4,60,32,32,60,0

1305 7F5A 3C04043C20203C00

1306 7F62 defb 120,8,8,60,12,12,124,0

1306 7F62 7808083C0C0C7C00

1307 7F6A defb 96,96,96,108,124,12,12,0

1307 7F6A 6060606C7C0C0C00

1308 7F72 defb 124,64,64,124,4,4,124,0

1308 7F72 7C40407C04047C00

1309 7F7A defb 112,80,64,64,124,68,124,0

1309 7F7A 705040407C447C00

1310 7F82 defb 124,68,4,8,16,16,16,0

1310 7F82 7C44040810101000

1311 7F8A defb 60,36,36,126,102,102,126,0

1311 7F8A 3C24247E66667E00

1312 7F92 defb 124,68,68,124,12,12,12,0

1312 7F92 7C44447C0C0C0C00

1313 7F9A defb 0,24,24,0,0,24,24,0

1313 7F9A 0018180000181800

1314 7FA2 defb 0,24,24,0,0,24,24,8

1314 7FA2 0018180000181808

1315 7FAA defb 0,8,16,96,48,24,24,0

1315 7FAA 0008106030181800

1316 7FB2 defb 0,0,126,0,126,0,0,0

1316 7FB2 00007E007E000000

1317 7FBA defb 0,16,8,6,12,24,24,0

1317 7FBA 001008060C181800

1318 7FC2 defb 126,66,2,30,24,0,24,0

1318 7FC2 7E42021E18001800

1319 7FCA defb 254,130,158,210,222,192,254,0

1319 7FCA FE829ED2DEC0FE00

1320 7FD2 defb 60,36,36,126,102,102,102,0

1320 7FD2 3C24247E66666600

1321 7FDA defb 124,68,68,126,102,102,126,0

1321 7FDA 7C44447E66667E00

1322 7FE2 defb 120,72,64,96,100,100,124,0

1322 7FE2 7848406064647C00

1323 7FEA defb 120,68,68,100,100,100,120,0

1323 7FEA 7844446464647800

1324 7FF2 defb 60,32,32,120,96,96,124,0

1324 7FF2 3C20207860607C00

1325 7FFA defb 60,32,32,120,96,96,96,0

1325 7FFA 3C20207860606000

1326 8002 defb 120,72,64,78,70,70,126,0

1326 8002 7848404E46467E00

1327 800A defb 68,68,68,124,100,100,100,0

1327 800A 4444447C64646400

1328 8012 defb 16,16,16,24,24,24,24,0

1328 8012 1010101818181800

1329 801A defb 4,4,4,12,12,76,124,0

1329 801A 0404040C0C4C7C00

1330 8022 defb 76,76,76,126,66,66,66,0

1330 8022 4C4C4C7E42424200

1331 802A defb 64,64,64,96,96,96,124,0

1331 802A 4040406060607C00

1332 8032 defb 126,82,82,198,198,198,198,0

1332 8032 7E5252C6C6C6C600

1333 803A defb 124,68,68,100,100,100,100,0

1333 803A 7C44446464646400

1334 8042 defb 126,98,98,98,66,66,126,0

1334 8042 7E62626242427E00

1335 804A defb 124,68,68,124,96,96,96,0

1335 804A 7C44447C60606000

1336 8052 defb 126,66,66,70,70,70,126,0

1336 8052 7E42424646467E00

1337 805A defb 124,68,68,124,70,70,70,0

1337 805A 7C44447C46464600

1338 8062 defb 126,96,96,126,6,6,126,0

1338 8062 7E60607E06067E00

1339 806A defb 126,16,16,24,24,24,24,0

1339 806A 7E10101818181800

1340 8072 defb 68,68,68,100,100,100,120,0

1340 8072 4444446464647800

1341 807A defb 66,66,66,102,44,44,60,0

1341 807A 424242662C2C3C00

1342 8082 defb 198,198,198,214,84,84,108,0

1342 8082 C6C6C6D654546C00

1343 808A defb 204,204,204,120,196,196,196,0

1343 808A CCCCCC78C4C4C400

1344 8092 defb 68,68,68,124,12,12,124,0

1344 8092 4444447C0C0C7C00

1345 809A defb 124,4,4,124,96,96,124,0

1345 809A 7C04047C60607C00

1346 80A2 defb 60,32,32,48,48,48,60,0

1346 80A2 3C20203030303C00

1347 80AA defb 64,96,48,24,12,12,12,0

1347 80AA 406030180C0C0C00

1348 80B2 defb 60,4,4,12,12,12,60,0

1348 80B2 3C04040C0C0C3C00

1349 80BA defb 16,16,124,100,100,0,0,0

1349 80BA 10107C6464000000

1350 80C2 defb 0,0,0,0,0,0,0,255

1350 80C2 00000000000000FF

1351 80CA defb 28,20,16,126,48,48,126,0

1351 80CA 1C14107E30307E00

1352 80D2 defb 0,0,62,2,126,98,126,0

1352 80D2 00003E027E627E00

1353 80DA defb 64,64,64,126,70,70,126,0

1353 80DA 4040407E46467E00

1354 80E2 defb 0,0,126,64,96,96,126,0

1354 80E2 00007E4060607E00

1355 80EA defb 2,2,2,126,98,98,126,0

1355 80EA 0202027E62627E00

1356 80F2 defb 0,0,126,70,126,64,126,0

1356 80F2 00007E467E407E00

1357 80FA defb 30,18,16,126,24,24,24,0

1357 80FA 1E12107E18181800

1358 8102 defb 0,0,126,98,98,126,2,126

1358 8102 00007E62627E027E

1359 810A defb 64,64,64,126,70,70,70,0

1359 810A 4040407E46464600

1360 8112 defb 8,0,8,8,24,24,24,0

1360 8112 0800080818181800

1361 811A defb 4,0,4,4,12,12,76,124

1361 811A 040004040C0C4C7C

1362 8122 defb 64,72,72,126,70,70,70,0

1362 8122 4048487E46464600

1363 812A defb 24,8,8,8,24,24,24,0

1363 812A 1808080818181800

1364 8132 defb 0,0,126,82,86,86,86,0

1364 8132 00007E5256565600

1365 813A defb 0,0,126,66,70,70,70,0

1365 813A 00007E4246464600

1366 8142 defb 0,0,126,98,98,66,126,0

1366 8142 00007E6262427E00

1367 814A defb 0,0,126,66,66,126,96,96

1367 814A 00007E42427E6060

1368 8152 defb 0,0,126,66,66,126,6,6

1368 8152 00007E42427E0606

1369 815A defb 0,0,126,66,96,96,96,0

1369 815A 00007E4260606000

1370 8162 defb 0,0,126,64,126,6,126,0

1370 8162 00007E407E067E00

1371 816A defb 32,32,120,32,48,50,62,0

1371 816A 2020782030323E00

1372 8172 defb 0,0,66,66,70,70,126,0

1372 8172 0000424246467E00

1373 817A defb 0,0,66,66,118,52,60,0

1373 817A 0000424276343C00

1374 8182 defb 0,0,66,66,106,106,126,0

1374 8182 000042426A6A7E00

1375 818A defb 0,0,102,38,60,100,102,0

1375 818A 000066263C646600

1376 8192 defb 0,0,66,66,66,126,6,126

1376 8192 00004242427E067E

1377 819A defb 0,0,126,4,24,32,126,0

1377 819A 00007E0418207E00

1378 81A2 defb 30,16,16,120,24,24,30,0

1378 81A2 1E10107818181E00

1379 81AA defb 128,64,32,16,8,4,2,0

1379 81AA 8040201008040200

1380 81B2 defb 120,8,8,30,24,24,120,0

1380 81B2 7808081E18187800

1381 81BA defb 116,92,92,0,0,0,0,0

1381 81BA 745C5C0000000000

1382 81C2 defb 255,129,189,161,177,189,129,255

1382 81C2 FF81BDA1B1BD81FF

1383 81CA jtab equ $

1384 81CA 63 defb 99

1385 81CB keys defb 34,26,38,37,32,31,1,36,28,20,12

1385 81CB 221A2625201F01241C140C

1386 81D6

1387 81D6

1388 81D6 ; Game engine code --------------------------------------------------------------

1389 81D6

1390 81D6 ; Arcade Game Designer.

1391 81D6 ; (C) 2008 - 2020 Jonathan Cauldwell.

1392 81D6 ; ZX Spectrum Engine v0.7.10

1393 81D6

1394 81D6 ; Global definitions.

1395 81D6

1396 81D6 SIMASK equ 248 ; SPRITEINK mask - allows users to set BRIGHT/FLASH/CLUT as well.

1397 81D6 SHRAPN equ 63926 ; shrapnel table, just below screen address table.

1398 81D6 SCADTB equ 64256 ; screen address table, just below map.

1399 81D6 MAP equ 64768 ; properties map buffer.

1400 81D6 loopa equ 23681 ; loop counter system variable.

1401 81D6 loopb equ 23728 ; loop counter system variable.

1402 81D6 loopc equ 23729 ; loop counter system variable.

1403 81D6

1404 81D6 ; Block characteristics.

1405 81D6

1406 81D6 PLATFM equ 1 ; platform.

1407 81D6 WALL equ PLATFM + 1 ; solid wall.

1408 81D6 LADDER equ WALL + 1 ; ladder.

1409 81D6 FODDER equ LADDER + 1 ; fodder block.

1410 81D6 DEADLY equ FODDER + 1 ; deadly block.

1411 81D6 CUSTOM equ DEADLY + 1 ; custom block.

1412 81D6 WATER equ CUSTOM + 1 ; water block.

1413 81D6 COLECT equ WATER + 1 ; collectable block.

1414 81D6 NUMTYP equ COLECT + 1 ; number of types.

1415 81D6

1416 81D6 ; Sprites.

1417 81D6

1418 81D6 NUMSPR equ 12 ; number of sprites.

1419 81D6 TABSIZ equ 17 ; size of each entry.

1420 81D6 SPRBUF equ NUMSPR \* TABSIZ ; size of entire table.

1421 81D6 NMESIZ equ 4 ; bytes stored in nmetab for each sprite.

1422 81D6 X equ 8 ; new x coordinate of sprite.

1423 81D6 Y equ X + 1 ; new y coordinate of sprite.

1424 81D6 PAM1ST equ 5 ; first sprite parameter, old x (ix+5).

1425 81D6

1426 81D6 ; Particle engine.

1427 81D6

1428 81D6 NUMSHR equ 55 ; pieces of shrapnel.

1429 81D6 SHRSIZ equ 6 ; bytes per particle.

1430 81D6

1431 81D6

1432 81D6 ; Game starts here.

1433 81D6 ; Musicizer relies on distance between start and varm remaining constant at 34 bytes.

1434 81D6 ; Do not mess with this section or users will be unable to control music with the M variable.

1435 81D6

1436 81D6 start equ $

1437 81D6

1438 81D6 ; Set up the font.

1439 81D6

1440 81D6 21 CA 7D ld hl,font-256 ; address of font.

1441 81D9 22 36 5C ld (23606),hl ; set up game font.

1442 81DC

1443 81DC ; if you wish to exit to BASIC at the end of each game, change "jp gamelp" to "jp game".

1444 81DC C3 8F 88 jp gamelp ; start the game.

1445 81DF

1446 81DF 00 joyval defb 0 ; joystick reading.

1447 81E0 00 frmno defb 0 ; selected frame.

1448 81E1

1449 81E1 ; Don't change the order of these four. Menu routine relies on winlft following wintop.

1450 81E1

1451 81E1 01 wintop defb WINDOWTOP ; top of window.

1452 81E2 00 winlft defb WINDOWLFT ; left edge.

1453 81E3 17 winhgt defb WINDOWHGT ; window height.

1454 81E4 20 winwid defb WINDOWWID ; window width.

1455 81E5

1456 81E5 01 numob defb NUMOBJ ; number of objects in game.

1457 81E6

1458 81E6 ; Variables start here.

1459 81E6 ; Pixel versions of wintop, winlft, winhgt, winwid.

1460 81E6

1461 81E6 08 wntopx defb (8 \* WINDOWTOP)

1462 81E7 00 wnlftx defb (8 \* WINDOWLFT)

1463 81E8 B0 wnbotx defb ((WINDOWTOP \* 8) + (WINDOWHGT \* 8) - 16)

1464 81E9 F0 wnrgtx defb ((WINDOWLFT \* 8) + (WINDOWWID \* 8) - 16)

1465 81EA 00 scno defb 0 ; present screen number.

1466 81EB 03 numlif defb 3 ; number of lives.

1467 81EC 00 vara defb 0 ; general-purpose variable.

1468 81ED 00 varb defb 0 ; general-purpose variable.

1469 81EE 00 varc defb 0 ; general-purpose variable.

1470 81EF 00 vard defb 0 ; general-purpose variable.

1471 81F0 00 vare defb 0 ; general-purpose variable.

1472 81F1 00 varf defb 0 ; general-purpose variable.

1473 81F2 00 varg defb 0 ; general-purpose variable.

1474 81F3 00 varh defb 0 ; general-purpose variable.

1475 81F4 00 vari defb 0 ; general-purpose variable.

1476 81F5 00 varj defb 0 ; general-purpose variable.

1477 81F6 00 vark defb 0 ; general-purpose variable.

1478 81F7 00 varl defb 0 ; general-purpose variable.

1479 81F8 00 varm defb 0 ; general-purpose variable.

1480 81F9 00 varn defb 0 ; general-purpose variable.

1481 81FA 00 varo defb 0 ; general-purpose variable.

1482 81FB 00 varp defb 0 ; general-purpose variable.

1483 81FC 00 varq defb 0 ; general-purpose variable.

1484 81FD 00 varr defb 0 ; general-purpose variable.

1485 81FE 00 vars defb 0 ; general-purpose variable.

1486 81FF 00 vart defb 0 ; general-purpose variable.

1487 8200 00 varu defb 0 ; general-purpose variable.

1488 8201 00 varv defb 0 ; general-purpose variable.

1489 8202 00 varw defb 0 ; general-purpose variable.

1490 8203 00 varz defb 0 ; general-purpose variable.

1491 8204 00 contrl defb 0 ; control, 0 = keyboard, 1 = Kempston, 2 = Sinclair, 3 = Mouse.

1492 8205 00 charx defb 0 ; cursor x position.

1493 8206 00 chary defb 0 ; cursor y position.

1494 8207 00 clock defb 0 ; last clock reading.

1495 8208 FF varrnd defb 255 ; last random number.

1496 8209 FE varobj defb 254 ; last object number.

1497 820A FF varopt defb 255 ; last option chosen from menu.

1498 820B FF varblk defb 255 ; block type.

1499 820C 00 nexlev defb 0 ; next level flag.

1500 820D 00 restfl defb 0 ; restart screen flag.

1501 820E 00 deadf defb 0 ; dead flag.

1502 820F 00 gamwon defb 0 ; game won flag.

1503 8210 00 dispx defb 0 ; cursor x position.

1504 8211 00 dispy defb 0 ; cursor y position.

1505 8212

1506 8212 ; Make sure pointers are arranged in the same order as the data itself.

1507 8212

1508 8212 4A 7E frmptr defw frmlst ; sprite frames.

1509 8214 36 6E blkptr defw chgfx ; block graphics.

1510 8216 46 6E colptr defw bcol ; address of char colours.

1511 8218 48 6E proptr defw bprop ; address of char properties.

1512 821A C0 6D scrptr defw scdat ; address of screens.

1513 821C 5A 7E nmeptr defw nmedat ; enemy start positions.

1514 821E

1515 821E ; Assorted game routines which can go in contended memory.

1516 821E

1517 821E ; Modify for inventory.

1518 821E

1519 821E 21 9E 83 minve ld hl,invdis ; routine address.

1520 8221 22 A9 82 ld (mod0+1),hl ; set up menu routine.

1521 8224 22 4F 82 ld (mod2+1),hl ; set up count routine.

1522 8227 21 AB 83 ld hl,fopt ; find option from available objects.

1523 822A 22 31 83 ld (mod1+1),hl ; set up routine.

1524 822D 18 0F jr dbox ; do menu routine.

1525 822F

1526 822F ; Modify for menu.

1527 822F

1528 822F 21 C9 8F mmenu ld hl,always ; routine address.

1529 8232 22 A9 82 ld (mod0+1),hl ; set up routine.

1530 8235 22 4F 82 ld (mod2+1),hl ; set up count routine.

1531 8238 21 60 83 ld hl,fstd ; standard option selection.

1532 823B 22 31 83 ld (mod1+1),hl ; set up routine.

1533 823E

1534 823E ; Drop through into box routine.

1535 823E

1536 823E ; Work out size of box for message or menu.

1537 823E

1538 823E ;dbox ld hl,nummsg ; total messages.

1539 823E ; cp (hl) ; does this one exist?

1540 823E ; ret nc ; no, nothing to display.

1541 823E 21 45 6D dbox ld hl,msgdat ; pointer to messages.

1542 8241 CD 5B 92 call getwrd ; get message number.

1543 8244 E5 push hl ; store pointer to message.

1544 8245 16 01 ld d,1 ; height.

1545 8247 AF xor a ; start at object zero.

1546 8248 32 13 8F ld (combyt),a ; store number of object in combyt.

1547 824B 5F ld e,a ; maximum width.

1548 824C 06 00 dbox5 ld b,0 ; this line's width.

1549 824E CD C9 8F mod2 call always ; item in player's possession?

1550 8251 20 01 jr nz,dbox6 ; not in inventory, skip this line.

1551 8253 14 inc d ; add to tally.

1552 8254 7E dbox6 ld a,(hl) ; get character.

1553 8255 23 inc hl ; next character.

1554 8256 FE 2C cp ',' ; reached end of line?

1555 8258 28 0B jr z,dbox3 ; yes.

1556 825A FE 0D cp 13 ; reached end of line?

1557 825C 28 07 jr z,dbox3 ; yes.

1558 825E 04 inc b ; add to this line's width.

1559 825F A7 and a ; end of message?

1560 8260 FA 6C 82 jp m,dbox4 ; yes, end count.

1561 8263 18 EF jr dbox6 ; repeat until we find the end.

1562 8265 7B dbox3 ld a,e ; maximum line width.

1563 8266 B8 cp b ; have we exceeded longest so far?

1564 8267 30 E3 jr nc,dbox5 ; no, carry on looking.

1565 8269 58 ld e,b ; make this the widest so far.

1566 826A 18 E0 jr dbox5 ; keep looking.

1567 826C 7B dbox4 ld a,e ; maximum line width.

1568 826D B8 cp b ; have we exceeded longest so far?

1569 826E 30 01 jr nc,dbox8 ; no, carry on looking.

1570 8270 58 ld e,b ; final line is the longest so far.

1571 8271 15 dbox8 dec d ; decrement items found.

1572 8272 CA 7F 83 jp z,dbox15 ; total was zero.

1573 8275 7B ld a,e ; longest line.

1574 8276 A7 and a ; was it zero?

1575 8277 CA 7F 83 jp z,dbox15 ; total was zero.

1576 827A ED 53 D6 83 ld (bwid),de ; set up size.

1577 827E

1578 827E ; That's set up our box size.

1579 827E

1580 827E 3A E3 81 ld a,(winhgt) ; window height in characters.

1581 8281 92 sub d ; subtract height of box.

1582 8282 1F rra ; divide by 2.

1583 8283 21 E1 81 ld hl,wintop ; top edge of window.

1584 8286 86 add a,(hl) ; add displacement.

1585 8287 32 D8 83 ld (btop),a ; set up box top.

1586 828A 3A E4 81 ld a,(winwid) ; window width in characters.

1587 828D 93 sub e ; subtract box width.

1588 828E 1F rra ; divide by 2.

1589 828F 23 inc hl ; left edge of window.

1590 8290 86 add a,(hl) ; add displacement.

1591 8291 32 D9 83 ld (blft),a ; box left.

1592 8294 2A 36 5C ld hl,(23606) ; font.

1593 8297 22 E3 8C ld (grbase),hl ; set up for text display.

1594 829A E1 pop hl ; restore message pointer.

1595 829B 3A D8 83 ld a,(btop) ; box top.

1596 829E 32 10 82 ld (dispx),a ; set display coordinate.

1597 82A1 AF xor a ; start at object zero.

1598 82A2 32 13 8F ld (combyt),a ; store number of object in combyt.

1599 82A5 3A 13 8F dbox2 ld a,(combyt) ; get object number.

1600 82A8 CD C9 8F mod0 call always ; check inventory for display.

1601 82AB C2 6D 83 jp nz,dbox13 ; not in inventory, skip this line.

1602 82AE

1603 82AE 3A D9 83 ld a,(blft) ; box left.

1604 82B1 32 11 82 ld (dispy),a ; set left display position.

1605 82B4 3A D6 83 ld a,(bwid) ; box width.

1606 82B7 47 ld b,a ; store width.

1607 82B8 7E dbox0 ld a,(hl) ; get character.

1608 82B9 FE 2C cp ',' ; end of line?

1609 82BB 28 45 jr z,dbox1 ; yes, next one.

1610 82BD FE 0D cp 13 ; end of option?

1611 82BF 28 41 jr z,dbox1 ; yes, on to next.

1612 82C1 05 dec b ; one less to display.

1613 82C2 E6 7F and 127 ; remove terminator.

1614 82C4 C5 push bc ; store characters remaining.

1615 82C5 E5 push hl ; store address on stack.

1616 82C6 F5 push af ; store character.

1617 82C7 CD C3 8D call gaadd ; get attribute address.

1618 82CA 3A 8D 5C ld a,(23693) ; current colour.

1619 82CD 77 ld (hl),a ; set attribute.

1620 82CE F1 pop af ; restore character.

1621 82CF CD 32 8E call pchr ; display on screen.

1622 82D2 E1 pop hl ; retrieve address of next character.

1623 82D3 C1 pop bc ; chars left for this line.

1624 82D4 7E ld a,(hl) ; get character.

1625 82D5 23 inc hl ; next character.

1626 82D6 FE 80 cp 128 ; end of message?

1627 82D8 D2 10 83 jp nc,dbox7 ; yes, job done.

1628 82DB 78 ld a,b ; chars remaining.

1629 82DC A7 and a ; are any left?

1630 82DD 20 D9 jr nz,dbox0 ; yes, continue.

1631 82DF

1632 82DF ; Reached limit of characters per line.

1633 82DF

1634 82DF 7E dbox9 ld a,(hl) ; get character.

1635 82E0 23 inc hl ; next one.

1636 82E1 FE 2C cp ',' ; another line?

1637 82E3 28 21 jr z,dbox10 ; yes, do next line.

1638 82E5 FE 0D cp 13 ; another line?

1639 82E7 28 1D jr z,dbox10 ; yes, on to next.

1640 82E9 FE 80 cp 128 ; end of message?

1641 82EB 30 2A jr nc,dbox11 ; yes, finish message.

1642 82ED 18 F0 jr dbox9

1643 82EF

1644 82EF ; Fill box to end of line.

1645 82EF

1646 82EF E5 dboxf push hl ; store address on stack.

1647 82F0 C5 push bc ; store characters remaining.

1648 82F1 CD C3 8D call gaadd ; get attribute address.

1649 82F4 3A 8D 5C ld a,(23693) ; current colour.

1650 82F7 77 ld (hl),a ; set attribute.

1651 82F8 3E 20 ld a,32 ; space character.

1652 82FA CD 32 8E call pchr ; display character.

1653 82FD C1 pop bc ; retrieve character count.

1654 82FE E1 pop hl ; retrieve address of next character.

1655 82FF 10 EE djnz dboxf ; repeat for remaining chars on line.

1656 8301 C9 ret

1657 8302 23 dbox1 inc hl ; skip character.

1658 8303 CD EF 82 call dboxf ; fill box out to right side.

1659 8306 3A 10 82 dbox10 ld a,(dispx) ; x coordinate.

1660 8309 3C inc a ; down a line.

1661 830A 32 10 82 ld (dispx),a ; next position.

1662 830D C3 A5 82 jp dbox2 ; next line.

1663 8310 78 dbox7 ld a,b ; chars remaining.

1664 8311 A7 and a ; are any left?

1665 8312 28 03 jr z,dbox11 ; no, nothing to draw.

1666 8314 CD EF 82 call dboxf ; fill message to line.

1667 8317

1668 8317 ; Drawn the box menu, now select option.

1669 8317

1670 8317 3A D8 83 dbox11 ld a,(btop) ; box top.

1671 831A 32 10 82 ld (dispx),a ; set bar position.

1672 831D CD 04 91 dbox14 call joykey ; get controls.

1673 8320 E6 1F and 31 ; anything pressed?

1674 8322 20 F9 jr nz,dbox14 ; yes, debounce it.

1675 8324 CD 81 83 call dbar ; draw bar.

1676 8327 CD 04 91 dbox12 call joykey ; get controls.

1677 832A E6 1C and 28 ; anything pressed?

1678 832C 28 F9 jr z,dbox12 ; no, nothing.

1679 832E E6 10 and 16 ; fire button pressed?

1680 8330 C2 60 83 mod1 jp nz,fstd ; yes, job done.

1681 8333 CD 81 83 call dbar ; delete bar.

1682 8336 3A DF 81 ld a,(joyval) ; joystick reading.

1683 8339 E6 08 and 8 ; going up?

1684 833B 20 13 jr nz,dboxu ; yes, go up.

1685 833D 3A 10 82 ld a,(dispx) ; vertical position of bar.

1686 8340 3C inc a ; look down.

1687 8341 21 D8 83 ld hl,btop ; top of box.

1688 8344 96 sub (hl) ; find distance from top.

1689 8345 2B dec hl ; point to height.

1690 8346 BE cp (hl) ; are we at end?

1691 8347 CA 1D 83 jp z,dbox14 ; yes, go no further.

1692 834A 21 10 82 ld hl,dispx ; coordinate.

1693 834D 34 inc (hl) ; move bar.

1694 834E 18 CD jr dbox14 ; continue.

1695 8350 3A 10 82 dboxu ld a,(dispx) ; vertical position of bar.

1696 8353 21 D8 83 ld hl,btop ; top of box.

1697 8356 BE cp (hl) ; are we at the top?

1698 8357 CA 1D 83 jp z,dbox14 ; yes, go no further.

1699 835A 21 10 82 ld hl,dispx ; coordinate.

1700 835D 35 dec (hl) ; move bar.

1701 835E 18 BD jr dbox14 ; continue.

1702 8360 3A 10 82 fstd ld a,(dispx) ; bar position.

1703 8363 21 D8 83 ld hl,btop ; top of menu.

1704 8366 96 sub (hl) ; find selected option.

1705 8367 32 0A 82 ld (varopt),a ; store the option.

1706 836A C3 9C 84 jp redraw ; redraw the screen.

1707 836D

1708 836D ; Option not available. Skip this line.

1709 836D

1710 836D 7E dbox13 ld a,(hl) ; get character.

1711 836E 23 inc hl ; next one.

1712 836F FE 2C cp ',' ; another line?

1713 8371 CA A5 82 jp z,dbox2 ; yes, do next line.

1714 8374 FE 0D cp 13 ; another line?

1715 8376 CA A5 82 jp z,dbox2 ; yes, on to next line.

1716 8379 A7 and a ; end of message?

1717 837A FA 17 83 jp m,dbox11 ; yes, finish message.

1718 837D 18 EE jr dbox13

1719 837F E1 dbox15 pop hl ; pop message pointer from the stack.

1720 8380 C9 ret

1721 8381

1722 8381 3A D9 83 dbar ld a,(blft) ; box left.

1723 8384 32 11 82 ld (dispy),a ; set display coordinate.

1724 8387 CD 95 8D call gprad ; get printing address.

1725 838A EB ex de,hl ; flip into hl register pair.

1726 838B 3A D6 83 ld a,(bwid) ; box width.

1727 838E 4F ld c,a ; loop counter in c.

1728 838F 54 ld d,h ; store screen address high byte.

1729 8390 06 08 dbar1 ld b,8 ; pixel height in b.

1730 8392 7E dbar0 ld a,(hl) ; get screen byte.

1731 8393 2F cpl ; reverse all bits.

1732 8394 77 ld (hl),a ; write back to screen.

1733 8395 24 inc h ; next line down.

1734 8396 10 FA djnz dbar0 ; draw rest of character.

1735 8398 62 ld h,d ; rsetore screen address.

1736 8399 2C inc l ; one char right.

1737 839A 0D dec c ; decrement character counter.

1738 839B 20 F3 jr nz,dbar1 ; repeat for whole line.

1739 839D C9 ret

1740 839E

1741 839E E5 invdis push hl ; store message text pointer.

1742 839F D5 push de ; store de pair for line count.

1743 83A0 21 13 8F ld hl,combyt ; object number.

1744 83A3 7E ld a,(hl) ; get object number.

1745 83A4 34 inc (hl) ; ready for next one.

1746 83A5 CD FE 8B call gotob ; check if we have object.

1747 83A8 D1 pop de ; retrieve de pair from stack.

1748 83A9 E1 pop hl ; retrieve text pointer.

1749 83AA C9 ret

1750 83AB ;always xor a ; set zero flag.

1751 83AB ; ret

1752 83AB

1753 83AB ; Find option selected.

1754 83AB

1755 83AB 3A 10 82 fopt ld a,(dispx)

1756 83AE 21 D8 83 ld hl,btop ; top of menu.

1757 83B1 96 sub (hl) ; find selected option.

1758 83B2 3C inc a ; object 0 needs one iteration, 1 needs 2 and so on.

1759 83B3 47 ld b,a ; option selected in b register.

1760 83B4 21 13 8F ld hl,combyt ; object number.

1761 83B7 36 00 ld (hl),0 ; set to first item.

1762 83B9 C5 fopt0 push bc ; store option counter in b register.

1763 83BA CD CA 83 call fobj ; find next object in inventory.

1764 83BD C1 pop bc ; restore option counter.

1765 83BE 10 F9 djnz fopt0 ; repeat for relevant steps down the list.

1766 83C0 3A 13 8F ld a,(combyt) ; get option.

1767 83C3 3D dec a ; one less, due to where we increment combyt.

1768 83C4 32 0A 82 ld (varopt),a ; store the option.

1769 83C7 C3 9C 84 jp redraw ; redraw the screen.

1770 83CA

1771 83CA 21 13 8F fobj ld hl,combyt ; object number.

1772 83CD 7E ld a,(hl) ; get object number.

1773 83CE 34 inc (hl) ; ready for next item.

1774 83CF C8 ret z ; in case we loop back to zero.

1775 83D0 CD FE 8B call gotob ; do we have this item?

1776 83D3 C8 ret z ; yes, it's on the list.

1777 83D4 18 F4 jr fobj ; repeat until we find next item in pockets.

1778 83D6

1779 83D6 00 bwid defb 0 ; box/menu width.

1780 83D7 00 blen defb 0 ; box/menu height.

1781 83D8 00 btop defb 0 ; box coordinates.

1782 83D9 00 blft defb 0

1783 83DA

1784 83DA ; Wait for keypress.

1785 83DA

1786 83DA CD E6 83 prskey call debkey ; debounce key.

1787 83DD CD 31 84 prsky0 call vsync ; vertical synch.

1788 83E0 CD 8E 02 call 654 ; return keyboard state in e.

1789 83E3 1C inc e ; is it 255?

1790 83E4 28 F7 jr z,prsky0

1791 83E6

1792 83E6 ; Debounce keypress.

1793 83E6

1794 83E6 CD 31 84 debkey call vsync ; update scrolling, sounds etc.

1795 83E9 CD 8E 02 call 654 ; d=shift, e=key.

1796 83EC 1C inc e ; is it 255?

1797 83ED 20 F7 jr nz,debkey ; no - loop until key is released.

1798 83EF C9 ret

1799 83F0

1800 83F0 ; Delay routine.

1801 83F0

1802 83F0 C5 delay push bc ; store loop counter.

1803 83F1 CD 31 84 call vsync ; wait for interrupt.

1804 83F4 C1 pop bc ; restore counter.

1805 83F5 10 F9 djnz delay ; repeat.

1806 83F7 C9 ret

1807 83F8

1808 83F8 ; Clear sprite table.

1809 83F8

1810 83F8 21 1C 98 xspr ld hl,sprtab ; sprite table.

1811 83FB 06 CC ld b,SPRBUF ; length of table.

1812 83FD 36 FF xspr0 ld (hl),255 ; clear one byte.

1813 83FF 23 inc hl ; move to next byte.

1814 8400 10 FB djnz xspr0 ; repeat for rest of table.

1815 8402 C9 ret

1816 8403

1817 8403 CD EB 8A silenc call silen1 ; silence channel 1.

1818 8406 CD F8 8A call silen2 ; silence channel 2.

1819 8409 CD 05 8B call silen3 ; silence channel 3.

1820 840C C3 77 99 jp plsnd ; play all channels to switch them off.

1821 840F

1822 840F ; Initialise all objects.

1823 840F

1824 840F DD 21 63 7E iniob ld ix,objdta ; objects table.

1825 8413 3A E5 81 ld a,(numob) ; number of objects in the game.

1826 8416 47 ld b,a ; loop counter.

1827 8417 11 27 00 ld de,39 ; distance between objects.

1828 841A DD 7E 24 iniob0 ld a,(ix+36) ; start screen.

1829 841D DD 77 21 ld (ix+33),a ; set start screen.

1830 8420 DD 7E 25 ld a,(ix+37) ; find start x.

1831 8423 DD 77 22 ld (ix+34),a ; set start x.

1832 8426 DD 7E 26 ld a,(ix+38) ; get initial y.

1833 8429 DD 77 23 ld (ix+35),a ; set y coord.

1834 842C DD 19 add ix,de ; point to next object.

1835 842E 10 EA djnz iniob0 ; repeat.

1836 8430 C9 ret

1837 8431

1838 8431 ; Screen synchronisation.

1839 8431

1840 8431 CD 04 91 vsync call joykey ; read joystick/keyboard.

1841 8434 3A 9B 84 ld a,(sndtyp) ; sound to play.

1842 8437 A7 and a ; any sound?

1843 8438 CA 62 84 jp z,vsync1 ; no.

1844 843B 47 ld b,a ; outer loop.

1845 843C 3A 48 5C ld a,(23624) ; border colour.

1846 843F 1F rra ; put border bits into d0, d1 and d2.

1847 8440 1F rra

1848 8441 1F rra

1849 8442 4F ld c,a ; first value to write to speaker.

1850 8443 78 ld a,b ; sound.

1851 8444 A7 and a ; test it.

1852 8445 FA 7A 84 jp m,vsync6 ; play white noise.

1853 8448 79 vsync2 ld a,c ; get speaker value.

1854 8449 D3 FE out (254),a ; write to speaker.

1855 844B EE F8 xor 248 ; toggle all except the border bits.

1856 844D 4F ld c,a ; store value for next time.

1857 844E 50 ld d,b ; store loop counter.

1858 844F 21 07 82 vsync3 ld hl,clock ; previous clock setting.

1859 8452 3A 78 5C ld a,(23672) ; current clock setting.

1860 8455 BE cp (hl) ; subtract last reading.

1861 8456 C2 5E 84 jp nz,vsync4 ; yes, no more processing please.

1862 8459 10 F4 djnz vsync3 ; loop.

1863 845B 42 ld b,d ; restore loop counter.

1864 845C 10 EA djnz vsync2 ; continue noise.

1865 845E 7A vsync4 ld a,d ; where we got to.

1866 845F 32 9B 84 vsynca ld (sndtyp),a ; remember for next time.

1867 8462 3A 78 5C vsync1 ld a,(23672) ; clock low.

1868 8465 1F rra ; rotate bit into carry.

1869 8466 DC 74 84 call c,vsync5 ; time to play sound and do shrapnel/ticker stuff.

1870 8469 21 07 82 ld hl,clock ; last clock reading.

1871 846C 3A 78 5C vsync0 ld a,(23672) ; current clock reading.

1872 846F BE cp (hl) ; are they the same?

1873 8470 28 FA jr z,vsync0 ; yes, wait until clock changes.

1874 8472 77 ld (hl),a ; set new clock reading.

1875 8473 C9 ret

1876 8474 CD 77 99 vsync5 call plsnd ; play sound.

1877 8477 C3 A4 85 jp proshr ; shrapnel and stuff.

1878 847A

1879 847A ; Play white noise.

1880 847A

1881 847A 78 vsync6 ld a,b ; 128 - 255.

1882 847B D6 7F sub 127

1883 847D 47 ld b,a

1884 847E 21 07 82 ld hl,clock ; previous clock setting.

1885 8481 ED 5F vsync7 ld a,r ; get random speaker value.

1886 8483 E6 F8 and 248 ; only retain the speaker/earphone bits.

1887 8485 B1 or c ; merge with border colour.

1888 8486 D3 FE out (254),a ; write to speaker.

1889 8488 3A 78 5C ld a,(23672) ; current clock setting.

1890 848B BE cp (hl) ; subtract last reading.

1891 848C C2 98 84 jp nz,vsync8 ; yes, no more processing please.

1892 848F 78 ld a,b

1893 8490 E6 7F and 127

1894 8492 3C inc a

1895 8493 3D vsync9 dec a

1896 8494 20 FD jr nz,vsync9 ; loop.

1897 8496 10 E9 djnz vsync7 ; continue noise.

1898 8498 AF vsync8 xor a

1899 8499 18 C4 jr vsynca

1900 849B 00 sndtyp defb 0

1901 849C ;clock defb 0 ; last clock reading.

1902 849C

1903 849C ; Redraw the screen.

1904 849C

1905 849C ; Remove old copy of all sprites for redraw.

1906 849C

1907 849C DD E5 redraw push ix ; place sprite pointer on stack.

1908 849E CD C0 8E call droom ; show screen layout.

1909 84A1 CD 4C 8B call shwob ; draw objects.

1910 84A4 06 0C numsp0 ld b,NUMSPR ; sprites to draw.

1911 84A6 DD 21 1C 98 ld ix,sprtab ; sprite table.

1912 84AA DD 7E 00 redrw0 ld a,(ix+0) ; old sprite type.

1913 84AD 3C inc a ; is it enabled?

1914 84AE 28 0C jr z,redrw1 ; no, find next one.

1915 84B0 DD 7E 03 ld a,(ix+3) ; sprite x.

1916 84B3 FE B1 cp 177 ; beyond maximum?

1917 84B5 30 05 jr nc,redrw1 ; yes, nothing to draw.

1918 84B7 C5 push bc ; store sprite counter.

1919 84B8 CD C8 93 call sspria ; show single sprite.

1920 84BB C1 pop bc ; retrieve sprite counter.

1921 84BC 11 11 00 redrw1 ld de,TABSIZ ; distance to next odd/even entry.

1922 84BF DD 19 add ix,de ; next sprite.

1923 84C1 10 E7 djnz redrw0 ; repeat for remaining sprites.

1924 84C3 CD 7B 99 call rbloc ; redraw blocks if in adventure mode.

1925 84C6 CD 46 88 call dshrp ; redraw shrapnel.

1926 84C9 DD E1 pop ix ; retrieve sprite pointer.

1927 84CB C9 ret

1928 84CC

1929 84CC ; Clear screen routine.

1930 84CC

1931 84CC 21 00 40 cls ld hl,16384 ; screen address.

1932 84CF 75 ld (hl),l ; blank first byte.

1933 84D0 11 01 40 ld de,16385 ; second byte.

1934 84D3 01 00 18 ld bc,6144 ; bytes to copy.

1935 84D6 ED B0 ldir ; blank them all.

1936 84D8 3A 8D 5C ld a,(23693) ; fetch attributes.

1937 84DB 77 ld (hl),a ; set first attribute cell.

1938 84DC 01 FF 02 ld bc,767 ; number of attributes.

1939 84DF ED B0 ldir ; set all attributes.

1940 84E1 21 00 00 ld hl,0 ; set hl to origin (0, 0).

1941 84E4 22 05 82 ld (charx),hl ; reset coordinates.

1942 84E7 C9 ret

1943 84E8

1944 84E8 ; Set palette routine and data.

1945 84E8 ; Palette.

1946 84E8 ; 48955 = register select port.

1947 84E8 ; 65339 = data read/write port.

1948 84E8

1949 84E8 ; 48955 = write to register:

1950 84E8 ; d0-d5 : select register sub-group.

1951 84E8 ; d6-d7 : select register group.

1952 84E8 ; 00 = sub-group determines entry in palette table.

1953 84E8 ; 64 = mode select, write d0 to 65339 to toggle mode on/off.

1954 84E8

1955 84E8 ; 65339 = data read/write:

1956 84E8 ; d0-d1 : blue intensity (last bit duplicated so Bb is Bbb)

1957 84E8 ; d2-d4 : red intensity

1958 84E8 ; d5-d7 : green intensity

1959 84E8

1960 84E8 01 3B BF setpal ld bc,48955 ; register select.

1961 84EB 3E 40 ld a,64 ; mode select.

1962 84ED ED 79 out (c),a ; set ULAplus mode.

1963 84EF 06 FF ld b,255 ; data write.

1964 84F1 3E 01 ld a,1 ; mode on.

1965 84F3 ED 79 out (c),a ; switch on ULAplus.

1966 84F5

1967 84F5 06 40 ld b,64 ; number of palette table entries to write.

1968 84F7 21 8A 7E setpa1 ld hl,palett ; palette we want.

1969 84FA 1E 00 ld e,0 ; register number.

1970 84FC C5 setpa0 push bc ; store counter.

1971 84FD 06 BF ld b,191 ; register select.

1972 84FF 7B ld a,e ; register number to write.

1973 8500 ED 79 out (c),a ; write to port.

1974 8502 06 FF ld b,255 ; data select.

1975 8504 7E ld a,(hl) ; get colour data from table.

1976 8505 ED 79 out (c),a ; write to port.

1977 8507 1C inc e ; next clut entry.

1978 8508 23 inc hl ; next table entry.

1979 8509 C1 pop bc ; restore counter from stack.

1980 850A 10 F0 djnz setpa0 ; set rest of palette.

1981 850C C9 ret

1982 850D

1983 850D endpal equ $

1984 850D

1985 850D 7E fdchk ld a,(hl) ; fetch cell.

1986 850E FE 04 cp FODDER ; is it fodder?

1987 8510 C0 ret nz ; no.

1988 8511 36 00 ld (hl),0 ; rewrite block type.

1989 8513 E5 push hl ; store pointer to block.

1990 8514 11 00 FD ld de,MAP ; address of map.

1991 8517 A7 and a ; clear carry flag for subtraction.

1992 8518 ED 52 sbc hl,de ; find simple displacement for block.

1993 851A 7D ld a,l ; low byte is y coordinate.

1994 851B E6 1F and 31 ; column position 0 - 31.

1995 851D 32 11 82 ld (dispy),a ; set up y position.

1996 8520 29 add hl,hl ; multiply displacement by 8.

1997 8521 29 add hl,hl

1998 8522 29 add hl,hl

1999 8523 7C ld a,h ; x coordinate now in h.

2000 8524 32 10 82 ld (dispx),a ; set the display coordinate.

2001 8527 2A 14 82 ld hl,(blkptr) ; blocks.

2002 852A 22 E3 8C ld (grbase),hl ; set graphics base.

2003 852D AF xor a ; block to write.

2004 852E ; call wbloc ; Only for Fodderblocks in Adventuremode

2005 852E CD 0D 8E call pattr ; write block.

2006 8531 E1 pop hl ; restore block pointer.

2007 8532 C9 ret

2008 8533

2009 8533 ; Colour a sprite.

2010 8533

2011 8533 cspr equ $

2012 8533 DD 7E 08 ld a,(ix+8) ; look at the vertical first.

2013 8536 FE B1 cp 177 ; is it out-of-range?

2014 8538 D0 ret nc ; yes, can't colour it.

2015 8539 07 rlca ; divide by 64.

2016 853A 07 rlca ; quicker than 6 rrca operations.

2017 853B 6F ld l,a ; store in e register for now.

2018 853C E6 03 and 3 ; mask to find segment.

2019 853E C6 58 add a,88 ; attributes start at 88\*256=22528.

2020 8540 67 ld h,a ; that's our high byte sorted.

2021 8541 7D ld a,l ; vertical/64 - same as vertical\*4.

2022 8542 E6 E0 and 224 ; want a multiple of 32.

2023 8544 6F ld l,a ; vertical element calculated.

2024 8545 DD 7E 09 ld a,(ix+9) ; get horizontal position.

2025 8548 1F rra ; divide by 8.

2026 8549 1F rra

2027 854A 1F rra

2028 854B E6 1F and 31 ; want result in range 0-31.

2029 854D 85 add a,l ; add to existing low byte.

2030 854E 6F ld l,a ; that's the low byte done.

2031 854F 11 1E 00 ld de,30 ; distance to next line down.

2032 8552 E5 push hl ; store address of attributes on stack.

2033 8553 D9 exx ; put attributes in alternate bank.

2034 8554 E1 pop hl ; retrieve attributes address.

2035 8555 11 00 A5 ld de,MAP-22528 ; distance to properties.

2036 8558 19 add hl,de ; address of first block property.

2037 8559

2038 8559 11 1E 00 ld de,30 ; distance to next line down.

2039 855C DD 7E 08 ld a,(ix+8) ; x coordinate.

2040 855F 06 03 cspr2 ld b,3 ; default rows to write.

2041 8561 E6 07 and 7 ; does x straddle cells?

2042 8563 20 01 jr nz,cspr0 ; yes, loop counter is good.

2043 8565 05 dec b ; one fewer rows to write.

2044 8566 7E cspr0 ld a,(hl) ; get property.

2045 8567 A7 and a ; is it empty space?

2046 8568 20 07 jr nz,cspr6 ; yes, don't paint it.

2047 856A D9 exx ; switch to attributes.

2048 856B 7E ld a,(hl) ; get attribute.

2049 856C E6 F8 and SIMASK ; remove ink.

2050 856E B1 or c ; put in the new ink.

2051 856F 77 ld (hl),a ; write back again.

2052 8570 D9 exx ; switch to properties.

2053 8571 2C cspr6 inc l ; next cell.

2054 8572 D9 exx ; switch to attributes.

2055 8573 2C inc l ; next cell.

2056 8574 D9 exx ; switch to properties.

2057 8575 7E ld a,(hl) ; fetch block property.

2058 8576 A7 and a ; is it empty space?

2059 8577 20 07 jr nz,cspr7 ; yes, don't paint it.

2060 8579 D9 exx ; switch to attributes.

2061 857A 7E ld a,(hl) ; get attributes.

2062 857B E6 F8 and SIMASK ; remove ink.

2063 857D B1 or c ; put in the new ink.

2064 857E 77 ld (hl),a ; write back again.

2065 857F D9 exx ; switch to properties.

2066 8580 2C cspr7 inc l ; next cell.

2067 8581 D9 exx ; switch to attributes.

2068 8582 2C inc l ; next cell.

2069 8583 D9 exx ; switch to properties.

2070 8584 DD 7E 09 ld a,(ix+9) ; y coordinate.

2071 8587 E6 07 and 7 ; straddling cells?

2072 8589 28 0A jr z,cspr1 ; no, only 2 wide.

2073 858B 7E ld a,(hl) ; get attributes.

2074 858C A7 and a ; is it empty space?

2075 858D 20 06 jr nz,cspr1 ; yes, don't paint it.

2076 858F D9 exx ; switch to attributes.

2077 8590 E6 F8 and SIMASK ; remove ink.

2078 8592 B1 or c ; put in the new ink.

2079 8593 77 ld (hl),a ; write back again.

2080 8594 D9 exx ; switch to properties.

2081 8595 19 cspr1 add hl,de ; next row.

2082 8596 D9 exx ; switch to attributes.

2083 8597 19 add hl,de ; next row.

2084 8598 D9 exx ; switch to properties.

2085 8599 10 CB djnz cspr0

2086 859B C9 ret

2087 859C

2088 859C ; Scrolly text and puzzle variables.

2089 859C

2090 859C 80 txtbit defb 128 ; bit to write.

2091 859D 10 txtwid defb 16 ; width of ticker message.

2092 859E 45 6D txtpos defw msgdat

2093 85A0 45 6D txtini defw msgdat

2094 85A2 16 40 txtscr defw 16406

2095 85A4

2096 85A4 ; Specialist routines.

2097 85A4 ; Process shrapnel.

2098 85A4

2099 85A4 DD 21 B6 F9 proshr ld ix,SHRAPN ; table.

2100 85A8 06 37 ld b,NUMSHR ; shrapnel pieces to process.

2101 85AA 11 06 00 ld de,SHRSIZ ; distance to next.

2102 85AD DD 7E 00 prosh0 ld a,(ix+0) ; on/off marker.

2103 85B0 17 rla ; check its status.

2104 85B1 D4 BB 85 proshx call nc,prosh1 ; on, so process it.

2105 85B4 DD 19 add ix,de ; point there.

2106 85B6 10 F5 djnz prosh0 ; round again.

2107 85B8 C3 91 97 jp scrly

2108 85BB C5 prosh1 push bc ; store counter.

2109 85BC CD 45 86 call plot ; delete the pixel.

2110 85BF DD 7E 00 ld a,(ix+0) ; restore shrapnel type.

2111 85C2 21 D8 85 ld hl,shrptr ; shrapnel routine pointers.

2112 85C5 CD D0 85 call prosh2 ; run the routine.

2113 85C8 CD 23 86 call chkxy ; check x and y are good before we redisplay.

2114 85CB C1 pop bc ; restore counter.

2115 85CC 11 06 00 ld de,SHRSIZ ; distance to next.

2116 85CF C9 ret

2117 85D0 07 prosh2 rlca ; 2 bytes per address.

2118 85D1 5F ld e,a ; copy to de.

2119 85D2 19 add hl,de ; point to address of routine.

2120 85D3 7E ld a,(hl) ; get address low.

2121 85D4 23 inc hl ; point to second byte.

2122 85D5 66 ld h,(hl) ; fetch high byte from table.

2123 85D6 6F ld l,a ; put low byte in l.

2124 85D7 E9 jp (hl) ; jump to routine.

2125 85D8

2126 85D8 D9 86 shrptr defw laser ; laser.

2127 85DA B2 86 defw trail ; vapour trail.

2128 85DC E8 85 defw shrap ; shrapnel from explosion.

2129 85DE 13 86 defw dotl ; horizontal starfield left.

2130 85E0 17 86 defw dotr ; horizontal starfield right.

2131 85E2 1B 86 defw dotu ; vertical starfield up.

2132 85E4 1F 86 defw dotd ; vertical starfield down.

2133 85E6 44 6D defw ptcusr ; user particle.

2134 85E8

2135 85E8 ; Explosion shrapnel.

2136 85E8

2137 85E8 DD 5E 01 shrap ld e,(ix+1) ; get the angle.

2138 85EB 16 00 ld d,0 ; no high byte.

2139 85ED 21 72 86 ld hl,shrsin ; shrapnel sine table.

2140 85F0 19 add hl,de ; point to sine.

2141 85F1

2142 85F1 5E ld e,(hl) ; fetch value from table.

2143 85F2 23 inc hl ; next byte of table.

2144 85F3 56 ld d,(hl) ; fetch value from table.

2145 85F4 23 inc hl ; next byte of table.

2146 85F5 4E ld c,(hl) ; fetch value from table.

2147 85F6 23 inc hl ; next byte of table.

2148 85F7 46 ld b,(hl) ; fetch value from table.

2149 85F8 DD 6E 02 ld l,(ix+2) ; x coordinate in hl.

2150 85FB DD 66 03 ld h,(ix+3)

2151 85FE 19 add hl,de ; add sine.

2152 85FF DD 75 02 ld (ix+2),l ; store new coordinate.

2153 8602 DD 74 03 ld (ix+3),h

2154 8605 DD 6E 04 ld l,(ix+4) ; y coordinate in hl.

2155 8608 DD 66 05 ld h,(ix+5)

2156 860B 09 add hl,bc ; add cosine.

2157 860C DD 75 04 ld (ix+4),l ; store new coordinate.

2158 860F DD 74 05 ld (ix+5),h

2159 8612 C9 ret

2160 8613

2161 8613 DD 35 05 dotl dec (ix+5) ; move left.

2162 8616 C9 ret

2163 8617 DD 34 05 dotr inc (ix+5) ; move left.

2164 861A C9 ret

2165 861B DD 35 03 dotu dec (ix+3) ; move up.

2166 861E C9 ret

2167 861F DD 34 03 dotd inc (ix+3) ; move down.

2168 8622 C9 ret

2169 8623

2170 8623 ; Check coordinates are good before redrawing at new position.

2171 8623

2172 8623 21 E6 81 chkxy ld hl,wntopx ; window top.

2173 8626 DD 7E 03 ld a,(ix+3) ; fetch shrapnel coordinate.

2174 8629 BE cp (hl) ; compare with top window limit.

2175 862A 38 41 jr c,kilshr ; out of window, kill shrapnel.

2176 862C 23 inc hl ; left edge.

2177 862D DD 7E 05 ld a,(ix+5) ; fetch shrapnel coordinate.

2178 8630 BE cp (hl) ; compare with left window limit.

2179 8631 38 3A jr c,kilshr ; out of window, kill shrapnel.

2180 8633

2181 8633 23 inc hl ; point to bottom.

2182 8634 7E ld a,(hl) ; fetch window limit.

2183 8635 C6 0F add a,15 ; add height of sprite.

2184 8637 DD BE 03 cp (ix+3) ; compare with shrapnel x coordinate.

2185 863A 38 31 jr c,kilshr ; off screen, kill shrapnel.

2186 863C 23 inc hl ; point to right edge.

2187 863D 7E ld a,(hl) ; fetch shrapnel y coordinate.

2188 863E C6 0F add a,15 ; add width of sprite.

2189 8640 DD BE 05 cp (ix+5) ; compare with window limit.

2190 8643 38 28 jr c,kilshr ; off screen, kill shrapnel.

2191 8645

2192 8645 ; Drop through.

2193 8645 ; Display shrapnel.

2194 8645

2195 8645 DD 6E 03 plot ld l,(ix+3) ; x integer.

2196 8648 DD 66 05 ld h,(ix+5) ; y integer.

2197 864B 22 10 82 ld (dispx),hl ; workspace coordinates.

2198 864E DD 7E 00 ld a,(ix+0) ; type.

2199 8651 A7 and a ; is it a laser?

2200 8652 28 12 jr z,plot1 ; yes, draw laser instead.

2201 8654 7C plot0 ld a,h ; which pixel within byte do we

2202 8655 E6 07 and 7 ; want to set first?

2203 8657 16 00 ld d,0 ; no high byte.

2204 8659 5F ld e,a ; copy to de.

2205 865A 21 04 87 ld hl,dots ; table of small pixel positions.

2206 865D 19 add hl,de ; hl points to values we want to POKE to screen.

2207 865E 5E ld e,(hl) ; get value.

2208 865F CD AB 93 call scadd ; screen address.

2209 8662 7E ld a,(hl) ; see what's already there.

2210 8663 AB xor e ; merge with pixels.

2211 8664 77 ld (hl),a ; put back on screen.

2212 8665 C9 ret

2213 8666 CD AB 93 plot1 call scadd ; screen address.

2214 8669 7E ld a,(hl) ; fetch byte there.

2215 866A 2F cpl ; toggle all bits.

2216 866B 77 ld (hl),a ; new byte.

2217 866C C9 ret

2218 866D

2219 866D DD 36 00 80 kilshr ld (ix+0),128 ; switch off shrapnel.

2220 8671 C9 ret

2221 8672

2222 8672 ;explc defb 0 ; explosion counter.

2223 8672

2224 8672 shrsin defw 0,1024,391,946,724,724,946,391

2224 8672 000000048701B203D402D402B2038701

2225 8682 defw 1024,0,946,65144,724,64811,391,64589

2225 8682 00040000B20378FED4022BFD87014DFC

2226 8692 defw 0,64512,65144,64589,64811,64811,64589,65144

2226 8692 000000FC78FE4DFC2BFD2BFD4DFC78FE

2227 86A2 defw 64512,0,64589,391,64811,724,65144,946

2227 86A2 00FC00004DFC87012BFDD40278FEB203

2228 86B2

2229 86B2 DD 35 01 trail dec (ix+1) ; time remaining.

2230 86B5 CA D4 86 jp z,trailk ; time to switch it off.

2231 86B8 CD 38 88 call qrand ; get a random number.

2232 86BB 1F rra ; x or y axis?

2233 86BC 38 0B jr c,trailv ; use x.

2234 86BE 1F rra ; which direction?

2235 86BF 38 04 jr c,traill ; go left.

2236 86C1 DD 34 05 inc (ix+5) ; go right.

2237 86C4 C9 ret

2238 86C5 DD 35 05 traill dec (ix+5) ; go left.

2239 86C8 C9 ret

2240 86C9 1F trailv rra ; which direction?

2241 86CA 38 04 jr c,trailu ; go up.

2242 86CC DD 34 03 inc (ix+3) ; go down.

2243 86CF C9 ret

2244 86D0 DD 35 03 trailu dec (ix+3) ; go up.

2245 86D3 C9 ret

2246 86D4 DD 36 03 C8 trailk ld (ix+3),200 ; set off-screen to kill vapour trail.

2247 86D8 C9 ret

2248 86D9

2249 86D9 DD 7E 01 laser ld a,(ix+1) ; direction.

2250 86DC 1F rra ; left or right?

2251 86DD 30 04 jr nc,laserl ; move left.

2252 86DF 06 08 ld b,8 ; distance to travel.

2253 86E1 18 02 jr laserm ; move laser.

2254 86E3 06 F8 laserl ld b,248 ; distance to travel.

2255 86E5 DD 7E 05 laserm ld a,(ix+5) ; y position.

2256 86E8 80 add a,b ; add distance.

2257 86E9 DD 77 05 ld (ix+5),a ; set new y coordinate.

2258 86EC

2259 86EC ; Test new block.

2260 86EC

2261 86EC 32 11 82 ld (dispy),a ; set y for block collision detection purposes.

2262 86EF DD 7E 03 ld a,(ix+3) ; get x.

2263 86F2 32 10 82 ld (dispx),a ; set coordinate for collision test.

2264 86F5 CD 9C 90 call tstbl ; get block type there.

2265 86F8 FE 02 cp WALL ; is it solid?

2266 86FA 28 D8 jr z,trailk ; yes, it cannot pass.

2267 86FC FE 04 cp FODDER ; is it fodder?

2268 86FE C0 ret nz ; no, ignore it.

2269 86FF CD 0D 85 call fdchk ; remove fodder block.

2270 8702 18 D0 jr trailk ; destroy laser.

2271 8704

2272 8704 dots defb 128,64,32,16,8,4,2,1

2272 8704 8040201008040201

2273 870C

2274 870C ; Plot, preserving de.

2275 870C

2276 870C D5 plotde push de ; put de on stack.

2277 870D CD 45 86 call plot ; plot pixel.

2278 8710 D1 pop de ; restore de from stack.

2279 8711 C9 ret

2280 8712

2281 8712 ; Shoot a laser.

2282 8712

2283 8712 4F shoot ld c,a ; store direction in c register.

2284 8713 DD 7E 08 ld a,(ix+8) ; x coordinate.

2285 8716 C6 07 shoot1 add a,7 ; down 7 pixels.

2286 8718 6F ld l,a ; puty x coordinate in l.

2287 8719 DD 66 09 ld h,(ix+9) ; y coordinate in h.

2288 871C DD E5 push ix ; store pointer to sprite.

2289 871E CD E0 87 call fpslot ; find particle slot.

2290 8721 30 2C jr nc,vapou2 ; failed, restore ix.

2291 8723 DD 36 00 00 ld (ix+0),0 ; set up a laser.

2292 8727 DD 71 01 ld (ix+1),c ; set the direction.

2293 872A DD 75 03 ld (ix+3),l ; set x coordinate.

2294 872D CB 19 rr c ; check direction we want.

2295 872F 38 08 jr c,shootr ; shoot right.

2296 8731 7C ld a,h ; y position.

2297 8732 ; dec a ; left a pixel.

2298 8732 E6 F8 shoot0 and 248 ; align on character boundary.

2299 8734 DD 77 05 ld (ix+5),a ; set y coordinate.

2300 8737 18 2D jr vapou0 ; draw first image.

2301 8739 7C shootr ld a,h ; y position.

2302 873A C6 0F add a,15 ; look right.

2303 873C 18 F4 jr shoot0 ; align and continue.

2304 873E

2305 873E ; Create a bit of vapour trail.

2306 873E

2307 873E DD E5 vapour push ix ; store pointer to sprite.

2308 8740 DD 6E 08 ld l,(ix+8) ; x coordinate.

2309 8743 DD 66 09 ld h,(ix+9) ; y coordinate.

2310 8746 11 07 07 vapou3 ld de,7\*256+7 ; mid-point of sprite.

2311 8749 19 add hl,de ; point to centre of sprite.

2312 874A CD E0 87 call fpslot ; find particle slot.

2313 874D 38 03 jr c,vapou1 ; no, we can use it.

2314 874F DD E1 vapou2 pop ix ; restore sprite pointer.

2315 8751 C9 ret ; out of slots, can't generate anything.

2316 8752

2317 8752 DD 75 03 vapou1 ld (ix+3),l ; set up x.

2318 8755 DD 74 05 ld (ix+5),h ; set up y coordinate.

2319 8758 CD 38 88 call qrand ; get quick random number.

2320 875B E6 0F and 15 ; random time.

2321 875D C6 0F add a,15 ; minimum time on screen.

2322 875F DD 77 01 ld (ix+1),a ; set time on screen.

2323 8762 DD 36 00 01 ld (ix+0),1 ; define particle as vapour trail.

2324 8766 CD 23 86 vapou0 call chkxy ; plot first position.

2325 8769 18 E4 jr vapou2

2326 876B

2327 876B ; Create a user particle.

2328 876B

2329 876B 08 ptusr ex af,af' ; store timer.

2330 876C DD 6E 08 ld l,(ix+8) ; x coordinate.

2331 876F DD 66 09 ld h,(ix+9) ; y coordinate.

2332 8772 11 07 07 ld de,7\*256+7 ; mid-point of sprite.

2333 8775 19 add hl,de ; point to centre of sprite.

2334 8776 CD E0 87 call fpslot ; find particle slot.

2335 8779 38 01 jr c,ptusr1 ; no, we can use it.

2336 877B C9 ret ; out of slots, can't generate anything.

2337 877C

2338 877C DD 75 03 ptusr1 ld (ix+3),l ; set up x.

2339 877F DD 74 05 ld (ix+5),h ; set up y coordinate.

2340 8782 08 ex af,af' ; restore timer.

2341 8783 DD 77 01 ld (ix+1),a ; set time on screen.

2342 8786 DD 36 00 07 ld (ix+0),7 ; define particle as user particle.

2343 878A C3 23 86 jp chkxy ; plot first position.

2344 878D

2345 878D

2346 878D ; Create a vertical or horizontal star.

2347 878D

2348 878D DD E5 star push ix ; store pointer to sprite.

2349 878F CD E0 87 call fpslot ; find particle slot.

2350 8792 DA 98 87 jp c,star7 ; found one we can use.

2351 8795 DD E1 star0 pop ix ; restore sprite pointer.

2352 8797 C9 ret ; out of slots, can't generate anything.

2353 8798

2354 8798 79 star7 ld a,c ; direction.

2355 8799 E6 03 and 3 ; is it left?

2356 879B 28 21 jr z,star1 ; yes, it's horizontal.

2357 879D 3D dec a ; is it right?

2358 879E 28 2C jr z,star2 ; yes, it's horizontal.

2359 87A0 3D dec a ; is it up?

2360 87A1 28 35 jr z,star3 ; yes, it's vertical.

2361 87A3

2362 87A3 3A E6 81 ld a,(wntopx) ; get edge of screen.

2363 87A6 3C inc a ; down one pixel.

2364 87A7 DD 77 03 star8 ld (ix+3),a ; set x coord.

2365 87AA CD 38 88 call qrand ; get quick random number.

2366 87AD DD 77 05 star9 ld (ix+5),a ; set y position.

2367 87B0 79 ld a,c ; direction.

2368 87B1 E6 03 and 3 ; zero to three.

2369 87B3 C6 03 add a,3 ; 3 to 6 for starfield.

2370 87B5 DD 77 00 ld (ix+0),a ; define particle as star.

2371 87B8 CD 23 86 call chkxy ; plot first position.

2372 87BB C3 95 87 jp star0

2373 87BE CD 38 88 star1 call qrand ; get quick random number.

2374 87C1 DD 77 03 ld (ix+3),a ; set x coord.

2375 87C4 3A E9 81 ld a,(wnrgtx) ; get edge of screen.

2376 87C7 C6 0F add a,15 ; add width of sprite minus 1.

2377 87C9 C3 AD 87 jp star9

2378 87CC CD 38 88 star2 call qrand ; get quick random number.

2379 87CF DD 77 03 ld (ix+3),a ; set x coord.

2380 87D2 3A E7 81 ld a,(wnlftx) ; get edge of screen.

2381 87D5 C3 AD 87 jp star9

2382 87D8 3A E8 81 star3 ld a,(wnbotx) ; get edge of screen.

2383 87DB C6 0F add a,15 ; height of sprite minus one pixel.

2384 87DD C3 A7 87 jp star8

2385 87E0

2386 87E0

2387 87E0 ; Find particle slot for lasers or vapour trail.

2388 87E0 ; Can't use alternate accumulator.

2389 87E0

2390 87E0 DD 21 B6 F9 fpslot ld ix,SHRAPN ; shrapnel table.

2391 87E4 11 06 00 ld de,SHRSIZ ; size of each particle.

2392 87E7 06 37 ld b,NUMSHR ; number of pieces in table.

2393 87E9 DD 7E 00 fpslt0 ld a,(ix+0) ; get type.

2394 87EC 17 rla ; is this slot in use?

2395 87ED D8 ret c ; no, we can use it.

2396 87EE DD 19 add ix,de ; point to more shrapnel.

2397 87F0 10 F7 djnz fpslt0 ; repeat for all shrapnel.

2398 87F2 C9 ret ; out of slots, can't generate anything.

2399 87F3

2400 87F3 ; Create an explosion at sprite position.

2401 87F3

2402 87F3 4F explod ld c,a ; particles to create.

2403 87F4 DD E5 push ix ; store pointer to sprite.

2404 87F6 DD 6E 08 ld l,(ix+8) ; x coordinate.

2405 87F9 DD 66 09 ld h,(ix+9) ; y coordinate.

2406 87FC DD 21 B6 F9 ld ix,SHRAPN ; shrapnel table.

2407 8800 11 06 00 ld de,SHRSIZ ; size of each particle.

2408 8803 06 37 ld b,NUMSHR ; number of pieces in table.

2409 8805 DD 7E 00 expld0 ld a,(ix+0) ; get type.

2410 8808 17 rla ; is this slot in use?

2411 8809 38 07 jr c,expld1 ; no, we can use it.

2412 880B DD 19 expld2 add ix,de ; point to more shrapnel.

2413 880D 10 F6 djnz expld0 ; repeat for all shrapnel.

2414 880F DD E1 expld3 pop ix ; restore sprite pointer.

2415 8811 C9 ret ; out of slots, can't generate any more.

2416 8812 79 expld1 ld a,c ; shrapnel counter.

2417 8813 E6 0F and 15 ; 0 to 15.

2418 8815 85 add a,l ; add to x.

2419 8816 DD 77 03 ld (ix+3),a ; x coord.

2420 8819 3A 45 88 ld a,(seed3) ; crap random number.

2421 881C E6 0F and 15 ; 0 to 15.

2422 881E 84 add a,h ; add to y.

2423 881F DD 77 05 ld (ix+5),a ; y coord.

2424 8822 DD 36 00 02 ld (ix+0),2 ; switch it on.

2425 8826 D9 exx ; store coordinates.

2426 8827 CD 23 86 call chkxy ; plot first position.

2427 882A CD 38 88 call qrand ; quick random angle.

2428 882D E6 3C and 60 ; keep within range.

2429 882F DD 77 01 ld (ix+1),a ; angle.

2430 8832 D9 exx ; restore coordinates.

2431 8833 0D dec c ; one less piece of shrapnel to generate.

2432 8834 20 D5 jr nz,expld2 ; back to main explosion loop.

2433 8836 18 D7 jr expld3 ; restore sprite pointer and exit.

2434 8838 3A 45 88 qrand ld a,(seed3) ; random seed.

2435 883B 6F ld l,a ; low byte.

2436 883C 26 00 ld h,0 ; no high byte.

2437 883E ED 5F ld a,r ; r register.

2438 8840 AE xor (hl) ; combine with seed.

2439 8841 32 45 88 ld (seed3),a ; new seed.

2440 8844 C9 ret

2441 8845 00 seed3 defb 0

2442 8846

2443 8846 ; Display all shrapnel.

2444 8846

2445 8846 21 0C 87 dshrp ld hl,plotde ; display routine.

2446 8849 22 B2 85 ld (proshx+1),hl ; modify routine.

2447 884C CD A4 85 call proshr ; process shrapnel.

2448 884F 21 BB 85 ld hl,prosh1 ; processing routine.

2449 8852 22 B2 85 ld (proshx+1),hl ; modify the call.

2450 8855 C9 ret

2451 8856

2452 8856 ; Particle engine.

2453 8856

2454 8856 21 B6 F9 inishr ld hl,SHRAPN ; table.

2455 8859 06 37 ld b,NUMSHR ; shrapnel pieces to process.

2456 885B 11 06 00 ld de,SHRSIZ ; distance to next.

2457 885E 36 FF inish0 ld (hl),255 ; kill the shrapnel.

2458 8860 19 add hl,de ; point there.

2459 8861 10 FB djnz inish0 ; round again.

2460 8863 C9 ret

2461 8864

2462 8864 ; Check for collision between laser and sprite.

2463 8864

2464 8864 21 B6 F9 lcol ld hl,SHRAPN ; shrapnel table.

2465 8867 11 06 00 ld de,SHRSIZ ; size of each particle.

2466 886A 06 37 ld b,NUMSHR ; number of pieces in table.

2467 886C 7E lcol0 ld a,(hl) ; get type.

2468 886D A7 and a ; is this slot a laser?

2469 886E 28 04 jr z,lcol1 ; yes, check collision.

2470 8870 19 lcol3 add hl,de ; point to more shrapnel.

2471 8871 10 F9 djnz lcol0 ; repeat for all shrapnel.

2472 8873 C9 ret ; no collision, carry not set.

2473 8874 E5 lcol1 push hl ; store pointer to laser.

2474 8875 23 inc hl ; direction.

2475 8876 23 inc hl ; not used.

2476 8877 23 inc hl ; x position.

2477 8878 7E ld a,(hl) ; get x.

2478 8879 DD 96 08 sub (ix+X) ; subtract sprite x.

2479 887C FE 10 lcolh cp 16 ; within range?

2480 887E 30 0A jr nc,lcol2 ; no, missed.

2481 8880 23 inc hl ; not used.

2482 8881 23 inc hl ; y position.

2483 8882 7E ld a,(hl) ; get y.

2484 8883 DD 96 09 sub (ix+Y) ; subtract sprite y.

2485 8886 FE 10 cp 16 ; within range?

2486 8888 38 03 jr c,lcol4 ; yes, collision occurred.

2487 888A E1 lcol2 pop hl ; restore laser pointer from stack.

2488 888B 18 E3 jr lcol3

2489 888D E1 lcol4 pop hl ; restore laser pointer.

2490 888E C9 ret ; return with carry set for collision.

2491 888F

2492 888F ; Main game engine code starts here.

2493 888F

2494 888F CD 94 88 gamelp call game

2495 8892 18 FB jr gamelp

2496 8894

2497 8894 game equ $

2498 8894

2499 8894 ; Set up screen address table.

2500 8894

2501 8894 21 00 40 setsat ld hl,16384 ; start of screen.

2502 8897 11 00 FB ld de,SCADTB ; screen address table.

2503 889A 06 00 ld b,0 ; vertical lines on screen.

2504 889C EB setsa0 ex de,hl ; flip table and screen address.

2505 889D 72 ld (hl),d ; write high byte.

2506 889E 24 inc h ; second table.

2507 889F 73 ld (hl),e ; write low byte.

2508 88A0 25 dec h ; back to first table.

2509 88A1 2C inc l ; next position in table.

2510 88A2 EB ex de,hl ; flip table and screen address back again.

2511 88A3 CD 71 94 call nline ; next line down.

2512 88A6 10 F4 djnz setsa0 ; repeat for all lines.

2513 88A8

2514 88A8 CD E8 84 call setpal ; set up ULAplus palette.

2515 88AB CD 56 88 rpblc2 call inishr ; initialise particle engine.

2516 88AE CD E7 6A evintr call evnt12 ; call intro/menu event.

2517 88B1

2518 88B1 21 00 FD ld hl,MAP ; block properties.

2519 88B4 11 01 FD ld de,MAP+1 ; next byte.

2520 88B7 01 FF 02 ld bc,767 ; size of property map.

2521 88BA 36 02 ld (hl),WALL ; write default property.

2522 88BC ED B0 ldir

2523 88BE CD 0F 84 call iniob ; initialise objects.

2524 88C1 AF xor a ; put zero in accumulator.

2525 88C2 32 0F 82 ld (gamwon),a ; reset game won flag.

2526 88C5

2527 88C5 21 D1 8C ld hl,score ; score.

2528 88C8 CD FC 89 call inisc ; init the score.

2529 88CB 3A 0C 61 mapst ld a,(stmap) ; start position on map.

2530 88CE 32 F9 98 ld (roomtb),a ; set up position in table, if there is one.

2531 88D1 CD 78 99 inipbl call ibloc ; set up first screen.

2532 88D4 DD 21 E8 98 ld ix,ssprit ; default to spare sprite in table.

2533 88D8 CD 39 6C evini call evnt13 ; initialisation.

2534 88DB

2535 88DB ; Two restarts.

2536 88DB ; First restart - clear all sprites and initialise everything.

2537 88DB

2538 88DB CD B7 89 rstrt call rsevt ; restart events.

2539 88DE CD F8 83 call xspr ; clear sprite table.

2540 88E1 CD 7A 96 call sprlst ; fetch pointer to screen sprites.

2541 88E4 CD C3 96 call ispr ; initialise sprite table.

2542 88E7 18 0C jr rstrt0

2543 88E9

2544 88E9 ; Second restart - clear all but player, and don't initialise him.

2545 88E9

2546 88E9 CD B7 89 rstrtn call rsevt ; restart events.

2547 88EC CD 91 96 call nspr ; clear all non-player sprites.

2548 88EF CD 7A 96 call sprlst ; fetch pointer to screen sprites.

2549 88F2 CD E9 96 call kspr ; initialise sprite table, no more players.

2550 88F5

2551 88F5

2552 88F5 ; Set up the player and/or enemy sprites.

2553 88F5

2554 88F5 AF rstrt0 xor a ; zero in accumulator.

2555 88F6 32 0C 82 ld (nexlev),a ; reset next level flag.

2556 88F9 32 0D 82 ld (restfl),a ; reset restart flag.

2557 88FC 32 0E 82 ld (deadf),a ; reset dead flag.

2558 88FF CD C0 8E call droom ; show screen layout.

2559 8902 CD 7B 99 call rbloc ; redraw blocks if in adventure mode.

2560 8905 CD 56 88 call inishr ; initialise particle engine.

2561 8908 CD 4C 8B call shwob ; draw objects.

2562 890B DD 21 1C 98 ld ix,sprtab ; address of sprite table, even sprites.

2563 890F CD FB 92 call dspr ; display sprites.

2564 8912 DD 21 2D 98 ld ix,sprtab+TABSIZ ; address of first odd sprite.

2565 8916 CD FB 92 call dspr ; display sprites.

2566 8919

2567 8919 CD 31 84 mloop call vsync ; synchronise with display.

2568 891C

2569 891C DD 21 1C 98 ld ix,sprtab ; address of sprite table, even sprites.

2570 8920 CD FB 92 call dspr ; display even sprites.

2571 8923

2572 8923 CD 77 99 call plsnd ; play sounds.

2573 8926 CD 31 84 call vsync ; synchronise with display.

2574 8929 DD 21 2D 98 ld ix,sprtab+TABSIZ ; address of first odd sprite.

2575 892D CD FB 92 call dspr ; display odd sprites.

2576 8930 DD 21 E8 98 ld ix,ssprit ; point to spare sprite for spawning purposes.

2577 8934 CD A6 6A evlp1 call evnt10 ; called once per main loop.

2578 8937 CD B6 92 call pspr ; process sprites.

2579 893A

2580 893A ; Main loop events.

2581 893A

2582 893A DD 21 E8 98 ld ix,ssprit ; point to spare sprite for spawning purposes.

2583 893E CD E6 6A evlp2 call evnt11 ; called once per main loop.

2584 8941

2585 8941 CD 67 92 bsortx call bsort ; sort sprites.

2586 8944 3A 0C 82 ld a,(nexlev) ; finished level flag.

2587 8947 A7 and a ; has it been set?

2588 8948 20 1C jr nz,newlev ; yes, go to next level.

2589 894A 3A 0F 82 ld a,(gamwon) ; finished game flag.

2590 894D A7 and a ; has it been set?

2591 894E 20 26 jr nz,evwon ; yes, finish the game.

2592 8950 3A 0D 82 ld a,(restfl) ; finished level flag.

2593 8953 3D dec a ; has it been set?

2594 8954 28 85 jr z,rstrt ; yes, go to next level.

2595 8956 3D dec a ; has it been set?

2596 8957 28 90 jr z,rstrtn ; yes, go to next level.

2597 8959

2598 8959 3A 0E 82 ld a,(deadf) ; dead flag.

2599 895C A7 and a ; is it non-zero?

2600 895D 20 1D jr nz,pdead ; yes, player dead.

2601 895F

2602 895F 21 E0 81 ld hl,frmno ; game frame.

2603 8962 34 inc (hl) ; advance the frame.

2604 8963

2605 8963 ; Back to start of main loop.

2606 8963

2607 8963 C3 19 89 jp DEBUG ; debug or back to mloop.

2608 8966

2609 8966 3A EA 81 newlev ld a,(scno) ; current screen.

2610 8969 21 35 6E ld hl,numsc ; total number of screens.

2611 896C 3C inc a ; next screen.

2612 896D BE cp (hl) ; reached the limit?

2613 896E 30 06 jr nc,evwon ; yes, game finished.

2614 8970 32 EA 81 ld (scno),a ; set new level number.

2615 8973 C3 DB 88 jp rstrt ; restart, clearing all aliens.

2616 8976 CD 41 6D evwon call evnt18 ; game completed.

2617 8979 C3 8D 89 jp tidyup ; tidy up and return to BASIC/calling routine.

2618 897C

2619 897C ; Player dead.

2620 897C

2621 897C AF pdead xor a ; zeroise accumulator.

2622 897D 32 0E 82 ld (deadf),a ; reset dead flag.

2623 8980 CD D3 6C evdie call evnt16 ; death subroutine.

2624 8983 3A EB 81 ld a,(numlif) ; number of lives.

2625 8986 A7 and a ; reached zero yet?

2626 8987 C2 DB 88 jp nz,rstrt ; restart game.

2627 898A CD ED 6C evfail call evnt17 ; failure event.

2628 898D 21 D7 8C tidyup ld hl,hiscor ; high score.

2629 8990 11 D1 8C ld de,score ; player's score.

2630 8993 06 06 ld b,6 ; digits to check.

2631 8995 1A tidyu2 ld a,(de) ; get score digit.

2632 8996 BE cp (hl) ; are we larger than high score digit?

2633 8997 38 06 jr c,tidyu0 ; high score is bigger.

2634 8999 20 0C jr nz,tidyu1 ; score is greater, record new high score.

2635 899B 23 inc hl ; next digit of high score.

2636 899C 13 inc de ; next digit of score.

2637 899D 10 F6 djnz tidyu2 ; repeat for all digits.

2638 899F 21 58 27 tidyu0 ld hl,10072 ; BASIC likes this in alternate hl.

2639 89A2 D9 exx ; flip hl into alternate registers.

2640 89A3 01 D1 8C ld bc,score ; return pointing to score.

2641 89A6 C9 ret

2642 89A7 21 D1 8C tidyu1 ld hl,score ; score.

2643 89AA 11 D7 8C ld de,hiscor ; high score.

2644 89AD 01 06 00 ld bc,6 ; digits to copy.

2645 89B0 ED B0 ldir ; copy score to high score.

2646 89B2 CD 42 6D evnewh call evnt19 ; new high score event.

2647 89B5 18 E8 jr tidyu0 ; tidy up.

2648 89B7

2649 89B7 ; Restart event.

2650 89B7

2651 89B7 DD 21 E8 98 rsevt ld ix,ssprit ; default to spare element in table.

2652 89BB C3 6C 6C evrs jp evnt14 ; call restart event.

2653 89BE

2654 89BE ; Copy number passed in a to string position bc, right-justified.

2655 89BE

2656 89BE 6F num2ch ld l,a ; put accumulator in l.

2657 89BF 26 00 ld h,0 ; blank high byte of hl.

2658 89C1 3E 20 ld a,32 ; leading spaces.

2659 89C3 11 64 00 numdg3 ld de,100 ; hundreds column.

2660 89C6 CD D4 89 call numdg ; show digit.

2661 89C9 11 0A 00 numdg2 ld de,10 ; tens column.

2662 89CC CD D4 89 call numdg ; show digit.

2663 89CF F6 10 or 16 ; last digit is always shown.

2664 89D1 11 01 00 ld de,1 ; units column.

2665 89D4 E6 30 numdg and 48 ; clear carry, clear digit.

2666 89D6 ED 52 numdg1 sbc hl,de ; subtract from column.

2667 89D8 38 05 jr c,numdg0 ; nothing to show.

2668 89DA F6 10 or 16 ; something to show, make it a digit.

2669 89DC 3C inc a ; increment digit.

2670 89DD 18 F7 jr numdg1 ; repeat until column is zero.

2671 89DF 19 numdg0 add hl,de ; restore total.

2672 89E0 FE 20 cp 32 ; leading space?

2673 89E2 C8 ret z ; yes, don't write that.

2674 89E3 02 ld (bc),a ; write digit to buffer.

2675 89E4 03 inc bc ; next buffer position.

2676 89E5 C9 ret

2677 89E6 6F num2dd ld l,a ; put accumulator in l.

2678 89E7 26 00 ld h,0 ; blank high byte of hl.

2679 89E9 3E 20 ld a,32 ; leading spaces.

2680 89EB 11 64 00 ld de,100 ; hundreds column.

2681 89EE CD D4 89 call numdg ; show digit.

2682 89F1 F6 10 or 16 ; second digit is always shown.

2683 89F3 18 D4 jr numdg2

2684 89F5 6F num2td ld l,a ; put accumulator in l.

2685 89F6 26 00 ld h,0 ; blank high byte of hl.

2686 89F8 3E 30 ld a,48 ; leading spaces.

2687 89FA 18 C7 jr numdg3

2688 89FC

2689 89FC 06 06 inisc ld b,6 ; digits to initialise.

2690 89FE 36 30 inisc0 ld (hl),'0' ; write zero digit.

2691 8A00 23 inc hl ; next column.

2692 8A01 10 FB djnz inisc0 ; repeat for all digits.

2693 8A03 C9 ret

2694 8A04

2695 8A04

2696 8A04 ; Multiply h by d and return in hl.

2697 8A04

2698 8A04 5A imul ld e,d ; HL = H \* D

2699 8A05 4C ld c,h ; make c first multiplier.

2700 8A06 21 00 00 imul0 ld hl,0 ; zeroise total.

2701 8A09 54 ld d,h ; zeroise high byte.

2702 8A0A 06 08 ld b,8 ; repeat 8 times.

2703 8A0C CB 19 imul1 rr c ; rotate rightmost bit into carry.

2704 8A0E 30 02 jr nc,imul2 ; wasn't set.

2705 8A10 19 add hl,de ; bit was set, so add de.

2706 8A11 A7 and a ; reset carry.

2707 8A12 CB 13 imul2 rl e ; shift de 1 bit left.

2708 8A14 CB 12 rl d

2709 8A16 10 F4 djnz imul1 ; repeat 8 times.

2710 8A18 C9 ret

2711 8A19

2712 8A19 ; Divide d by e and return in d, remainder in a.

2713 8A19

2714 8A19 AF idiv xor a

2715 8A1A 06 08 ld b,8 ; bits to shift.

2716 8A1C CB 22 idiv0 sla d ; multiply d by 2.

2717 8A1E 17 rla ; shift carry into remainder.

2718 8A1F BB cp e ; test if e is smaller.

2719 8A20 38 02 jr c,idiv1 ; e is greater, no division this time.

2720 8A22 93 sub e ; subtract it.

2721 8A23 14 inc d ; rotate into d.

2722 8A24 10 F6 idiv1 djnz idiv0

2723 8A26 C9 ret

2724 8A27

2725 8A27 ; Initialise a sound.

2726 8A27

2727 8A27 ED 5B 4C 8A isnd ld de,(ch1ptr) ; first pointer.

2728 8A2B 1A ld a,(de) ; get first byte.

2729 8A2C 3C inc a ; reached the end?

2730 8A2D 28 11 jr z,isnd1 ; that'll do.

2731 8A2F ED 5B 4E 8A ld de,(ch2ptr) ; second pointer.

2732 8A33 1A ld a,(de) ; get first byte.

2733 8A34 3C inc a ; reached the end?

2734 8A35 28 0D jr z,isnd2 ; that'll do.

2735 8A37 ED 5B 50 8A ld de,(ch3ptr) ; final pointer.

2736 8A3B 1A ld a,(de) ; get first byte.

2737 8A3C 3C inc a ; reached the end?

2738 8A3D 28 09 jr z,isnd3 ; that'll do.

2739 8A3F C9 ret

2740 8A40 22 4C 8A isnd1 ld (ch1ptr),hl ; set up the sound.

2741 8A43 C9 ret

2742 8A44 22 4E 8A isnd2 ld (ch2ptr),hl ; set up the sound.

2743 8A47 C9 ret

2744 8A48 22 50 8A isnd3 ld (ch3ptr),hl ; set up the sound.

2745 8A4B C9 ret

2746 8A4C

2747 8A4C

2748 8A4C C0 93 ch1ptr defw spmask

2749 8A4E C0 93 ch2ptr defw spmask

2750 8A50 C0 93 ch3ptr defw spmask

2751 8A52

2752 8A52 ;plsnd call plsnd1 ; first channel.

2753 8A52 ; call plsnd2 ; second one.

2754 8A52 ; call plsnd3 ; final channel.

2755 8A52

2756 8A52 ; Write the contents of our AY buffer to the AY registers.

2757 8A52

2758 8A52 21 6A 8A w8912 ld hl,snddat ; start of AY-3-8912 register data.

2759 8A55 11 00 0E ld de,14\*256 ; start with register 0, 14 to write.

2760 8A58 0E FD ld c,253 ; low byte of port to write.

2761 8A5A 06 FF w8912a ld b,255 ; port 65533=select soundchip register.

2762 8A5C ED 59 out (c),e ; tell chip which register we're writing.

2763 8A5E 7E ld a,(hl) ; value to write.

2764 8A5F 06 BF ld b,191 ; port 49149=write value to register.

2765 8A61 ED 79 out (c),a ; this is what we're putting there.

2766 8A63 1C inc e ; next sound chip register.

2767 8A64 23 inc hl ; next byte to write.

2768 8A65 15 dec d ; decrement loop counter.

2769 8A66 C2 5A 8A jp nz,w8912a ; repeat until done.

2770 8A69 C9 ret

2771 8A6A

2772 8A6A 00 00 snddat defw 0 ; tone registers, channel A.

2773 8A6C 00 00 defw 0 ; channel B tone registers.

2774 8A6E 00 00 defw 0 ; as above, channel C.

2775 8A70 00 sndwnp defb 0 ; white noise period.

2776 8A71 3C sndmix defb 60 ; tone/noise mixer control.

2777 8A72 00 sndv1 defb 0 ; channel A amplitude/envelope generator.

2778 8A73 00 sndv2 defb 0 ; channel B amplitude/envelope.

2779 8A74 00 sndv3 defb 0 ; channel C amplitude/envelope.

2780 8A75 00 00 defw 0 ; duration of each note.

2781 8A77 00 defb 0

2782 8A78

2783 8A78 23 plwn inc hl ; next byte of sound.

2784 8A79 E6 38 and 56 ; check if we're bothering with white noise.

2785 8A7B C0 ret nz ; we're not.

2786 8A7C 7E ld a,(hl) ; fetch byte.

2787 8A7D 32 70 8A ld (sndwnp),a ; set white noise period.

2788 8A80 C9 ret

2789 8A81

2790 8A81

2791 8A81 CD 17 8B plsnd2 call cksnd2 ; check sound for first channel.

2792 8A84 FE FF cp 255 ; reached end?

2793 8A86 28 70 jr z,silen2 ; silence this channel.

2794 8A88 E6 0F and 15 ; sound bits.

2795 8A8A 32 73 8A ld (sndv2),a ; set volume for channel.

2796 8A8D 3A 71 8A ld a,(sndmix) ; mixer byte.

2797 8A90 E6 ED and 237 ; remove bits for this channel.

2798 8A92 47 ld b,a ; store in b register.

2799 8A93 CD D7 8A call plmix ; fetch mixer details.

2800 8A96 E6 12 and 18 ; mixer bits we want.

2801 8A98 B0 or b ; combine with mixer bits.

2802 8A99 32 71 8A ld (sndmix),a ; new mixer value.

2803 8A9C CD 78 8A call plwn ; white noise check.

2804 8A9F 23 inc hl ; tone low.

2805 8AA0 5E ld e,(hl) ; fetch value.

2806 8AA1 23 inc hl ; tone high.

2807 8AA2 56 ld d,(hl) ; fetch value.

2808 8AA3 ED 53 6C 8A ld (snddat+2),de ; set tone.

2809 8AA7 23 inc hl ; next bit of sound.

2810 8AA8 22 4E 8A ld (ch2ptr),hl ; set pointer.

2811 8AAB C9 ret

2812 8AAC

2813 8AAC CD 1C 8B plsnd3 call cksnd3 ; check sound for first channel.

2814 8AAF FE FF cp 255 ; reached end?

2815 8AB1 28 52 jr z,silen3 ; silence last channel.

2816 8AB3 E6 0F and 15 ; sound bits.

2817 8AB5 32 74 8A ld (sndv3),a ; set volume for channel.

2818 8AB8 3A 71 8A ld a,(sndmix) ; mixer byte.

2819 8ABB E6 DB and 219 ; remove bits for this channel.

2820 8ABD 47 ld b,a ; store in b register.

2821 8ABE CD D7 8A call plmix ; fetch mixer details.

2822 8AC1 E6 24 and 36 ; mixer bits we want.

2823 8AC3 B0 or b ; combine with mixer bits.

2824 8AC4 32 71 8A ld (sndmix),a ; new mixer value.

2825 8AC7 CD 78 8A call plwn ; white noise check.

2826 8ACA 23 inc hl ; tone low.

2827 8ACB 5E ld e,(hl) ; fetch value.

2828 8ACC 23 inc hl ; tone high.

2829 8ACD 56 ld d,(hl) ; fetch value.

2830 8ACE ED 53 6E 8A ld (snddat+4),de ; set tone.

2831 8AD2 23 inc hl ; next bit of sound.

2832 8AD3 22 50 8A ld (ch3ptr),hl ; set pointer.

2833 8AD6 C9 ret

2834 8AD7

2835 8AD7 7E plmix ld a,(hl) ; fetch mixer byte.

2836 8AD8 E6 C0 and 192 ; mix bits are d6 and d7.

2837 8ADA 07 rlca ; rotate into d0 and d1.

2838 8ADB 07 rlca

2839 8ADC 5F ld e,a ; displacement in de.

2840 8ADD 16 00 ld d,0

2841 8ADF E5 push hl ; store pointer on stack.

2842 8AE0 21 E7 8A ld hl,mixtab ; mixer table.

2843 8AE3 19 add hl,de ; point to mixer byte.

2844 8AE4 7E ld a,(hl) ; fetch mixer value.

2845 8AE5 E1 pop hl ; restore pointer.

2846 8AE6 C9 ret

2847 8AE7 3F 38 07 00 mixtab defb 63,56,7,0 ; mixer byte settings.

2848 8AEB

2849 8AEB AF silen1 xor a ; zero.

2850 8AEC 32 72 8A ld (sndv1),a ; sound off.

2851 8AEF 3A 71 8A ld a,(sndmix) ; mixer byte.

2852 8AF2 F6 09 or 9 ; mix bits off.

2853 8AF4 32 71 8A ld (sndmix),a ; mixer setting for channel.

2854 8AF7 C9 ret

2855 8AF8 AF silen2 xor a ; zero.

2856 8AF9 32 73 8A ld (sndv2),a ; sound off.

2857 8AFC 3A 71 8A ld a,(sndmix) ; mixer byte.

2858 8AFF F6 12 or 18 ; mix bits off.

2859 8B01 32 71 8A ld (sndmix),a ; mixer setting for channel.

2860 8B04 C9 ret

2861 8B05 AF silen3 xor a ; zero.

2862 8B06 32 74 8A ld (sndv3),a ; sound off.

2863 8B09 3A 71 8A ld a,(sndmix) ; mixer byte.

2864 8B0C F6 24 or 36 ; mix bits off.

2865 8B0E 32 71 8A ld (sndmix),a ; mixer setting for channel.

2866 8B11 C9 ret

2867 8B12 2A 4C 8A cksnd1 ld hl,(ch1ptr) ; pointer to sound.

2868 8B15 7E ld a,(hl) ; fetch mixer/flag.

2869 8B16 C9 ret

2870 8B17 2A 4E 8A cksnd2 ld hl,(ch2ptr) ; pointer to sound.

2871 8B1A 7E ld a,(hl) ; fetch mixer/flag.

2872 8B1B C9 ret

2873 8B1C 2A 50 8A cksnd3 ld hl,(ch3ptr) ; pointer to sound.

2874 8B1F 7E ld a,(hl) ; fetch mixer/flag.

2875 8B20 C9 ret

2876 8B21

2877 8B21 CD 12 8B plsnd1 call cksnd1 ; check sound for first channel.

2878 8B24 FE FF cp 255 ; reached end?

2879 8B26 28 C3 jr z,silen1 ; silence first channel.

2880 8B28 E6 0F and 15 ; sound bits.

2881 8B2A 32 72 8A ld (sndv1),a ; set volume for channel.

2882 8B2D 3A 71 8A ld a,(sndmix) ; mixer byte.

2883 8B30 E6 F6 and 246 ; remove bits for this channel.

2884 8B32 47 ld b,a ; store in b register.

2885 8B33 CD D7 8A call plmix ; fetch mixer details.

2886 8B36 E6 09 and 9 ; mixer bits we want.

2887 8B38 B0 or b ; combine with mixer bits.

2888 8B39 32 71 8A ld (sndmix),a ; new mixer value.

2889 8B3C CD 78 8A call plwn ; white noise check.

2890 8B3F 23 inc hl ; tone low.

2891 8B40 5E ld e,(hl) ; fetch value.

2892 8B41 23 inc hl ; tone high.

2893 8B42 56 ld d,(hl) ; fetch value.

2894 8B43 ED 53 6A 8A ld (snddat),de ; set tone.

2895 8B47 23 inc hl ; next bit of sound.

2896 8B48 22 4C 8A ld (ch1ptr),hl ; set pointer.

2897 8B4B C9 ret

2898 8B4C

2899 8B4C

2900 8B4C ; Objects handling.

2901 8B4C ; 32 bytes for image

2902 8B4C ; 1 for colour

2903 8B4C ; 3 for room, x and y

2904 8B4C ; 3 for starting room, x and y.

2905 8B4C ; 254 = disabled.

2906 8B4C ; 255 = object in player's pockets.

2907 8B4C

2908 8B4C ; Show items present.

2909 8B4C

2910 8B4C 21 63 7E shwob ld hl,objdta ; objects table.

2911 8B4F 11 21 00 ld de,33 ; distance to room number.

2912 8B52 19 add hl,de ; point to room data.

2913 8B53 3A E5 81 ld a,(numob) ; number of objects in the game.

2914 8B56 47 ld b,a ; loop counter.

2915 8B57 C5 shwob0 push bc ; store count.

2916 8B58 E5 push hl ; store item pointer.

2917 8B59 3A EA 81 ld a,(scno) ; current location.

2918 8B5C BE cp (hl) ; same as an item?

2919 8B5D CC 78 8B call z,dobjc ; yes, display object in colour.

2920 8B60 E1 pop hl ; restore pointer.

2921 8B61 C1 pop bc ; restore counter.

2922 8B62 11 27 00 ld de,39 ; distance to next item.

2923 8B65 19 add hl,de ; point to it.

2924 8B66 10 EF djnz shwob0 ; repeat for others.

2925 8B68 C9 ret

2926 8B69

2927 8B69 ; Display object.

2928 8B69 ; hl must point to object's room number.

2929 8B69

2930 8B69 23 dobj inc hl ; point to x.

2931 8B6A 11 10 82 dobj0 ld de,dispx ; coordinates.

2932 8B6D ED A0 ldi ; transfer x coord.

2933 8B6F ED A0 ldi ; transfer y too.

2934 8B71 11 DC FF ld de,65500 ; minus 36.

2935 8B74 19 add hl,de ; point to image.

2936 8B75 C3 4E 8E dobj1 jp sprite ; draw this sprite.

2937 8B78

2938 8B78 CD 69 8B dobjc call dobj ; display object.

2939 8B7B 4E ld c,(hl) ; put ink in c register.

2940 8B7C

2941 8B7C ; Need to write attribute routine here.

2942 8B7C ; set up colour in c register first.

2943 8B7C

2944 8B7C 7E cobj ld a,(hl) ; get colour byte.

2945 8B7D A7 and a ; test it.

2946 8B7E F8 ret m ; colour not set.

2947 8B7F 26 16 ld h,22 ; quarter of attrubte address.

2948 8B81 3A 10 82 ld a,(dispx) ; x coord.

2949 8B84 E6 F8 and 248 ; only want multiple of 8.

2950 8B86 17 rla ; multiply by 4.

2951 8B87 CB 14 rl h

2952 8B89 17 rla

2953 8B8A CB 14 rl h ; high byte now set up.

2954 8B8C 6F ld l,a

2955 8B8D 3A 11 82 ld a,(dispy) ; take y position.

2956 8B90 1F rra ; divide it by 8.

2957 8B91 1F rra

2958 8B92 1F rra

2959 8B93 E6 1F and 31 ; remove unwanted bits.

2960 8B95 85 add a,l ; add to low byte.

2961 8B96 6F ld l,a ; low byte of address.

2962 8B97

2963 8B97 11 1E 00 ld de,30 ; distance to next line down.

2964 8B9A 3A 10 82 ld a,(dispx) ; x coordinate.

2965 8B9D 06 03 ld b,3 ; default rows to write.

2966 8B9F E6 07 and 7 ; does x straddle cells?

2967 8BA1 20 01 jr nz,cobj0 ; yes, loop counter is good.

2968 8BA3 05 dec b ; one less row to write.

2969 8BA4 7E cobj0 ld a,(hl) ; get attributes.

2970 8BA5 E6 F8 and 248 ; remove ink.

2971 8BA7 B1 or c ; put in the new ink.

2972 8BA8 77 ld (hl),a ; write back again.

2973 8BA9 2C inc l ; next cell.

2974 8BAA 7E ld a,(hl) ; get attributes.

2975 8BAB E6 F8 and 248 ; remove ink.

2976 8BAD B1 or c ; put in the new ink.

2977 8BAE 77 ld (hl),a ; write back again.

2978 8BAF 2C inc l ; next cell.

2979 8BB0 3A 11 82 ld a,(dispy) ; y coordinate.

2980 8BB3 E6 07 and 7 ; straddling cells?

2981 8BB5 28 05 jr z,cobj1 ; no, only 2 wide.

2982 8BB7 7E ld a,(hl) ; get attributes.

2983 8BB8 E6 F8 and 248 ; remove ink.

2984 8BBA B1 or c ; put in the new ink.

2985 8BBB 77 ld (hl),a ; write back again.

2986 8BBC 19 cobj1 add hl,de ; next row.

2987 8BBD 10 E5 djnz cobj0

2988 8BBF C9 ret

2989 8BC0

2990 8BC0 ; Remove an object.

2991 8BC0

2992 8BC0 21 E5 81 remob ld hl,numob ; number of objects in game.

2993 8BC3 BE cp (hl) ; are we checking past the end?

2994 8BC4 D0 ret nc ; yes, can't get non-existent item.

2995 8BC5 F5 push af ; remember object.

2996 8BC6 CD D0 8B call getob ; pick it up if we haven't already got it.

2997 8BC9 F1 pop af ; retrieve object number.

2998 8BCA CD FE 8B call gotob ; get its address.

2999 8BCD 36 FE ld (hl),254 ; remove it.

3000 8BCF C9 ret

3001 8BD0

3002 8BD0 ; Pick up object number held in the accumulator.

3003 8BD0

3004 8BD0 21 E5 81 getob ld hl,numob ; number of objects in game.

3005 8BD3 BE cp (hl) ; are we checking past the end?

3006 8BD4 D0 ret nc ; yes, can't get non-existent item.

3007 8BD5 CD FE 8B call gotob ; check if we already have it.

3008 8BD8 C8 ret z ; we already do.

3009 8BD9 EB ex de,hl ; object address in de.

3010 8BDA 21 EA 81 ld hl,scno ; current screen.

3011 8BDD BE cp (hl) ; is it on this screen?

3012 8BDE EB ex de,hl ; object address back in hl.

3013 8BDF 20 1A jr nz,getob0 ; not on screen, so nothing to delete.

3014 8BE1 36 FF ld (hl),255 ; pick it up.

3015 8BE3 23 inc hl ; point to x coord.

3016 8BE4 5E getob1 ld e,(hl) ; x coord.

3017 8BE5 23 inc hl ; back to y coord.

3018 8BE6 56 ld d,(hl) ; y coord.

3019 8BE7 ED 53 10 82 ld (dispx),de ; set display coords.

3020 8BEB 11 DD FF ld de,65501 ; minus graphic size.

3021 8BEE 19 add hl,de ; point to graphics.

3022 8BEF CD 75 8B call dobj1 ; delete object sprite.

3023 8BF2 3A 46 6E ld a,(bcol) ; first block colour.

3024 8BF5 E6 07 and 7 ; only want ink attribute.

3025 8BF7 4F ld c,a ; set up colour.

3026 8BF8 C3 7C 8B jp cobj ; colour object's old position.

3027 8BFB 36 FF getob0 ld (hl),255 ; pick it up.

3028 8BFD C9 ret

3029 8BFE

3030 8BFE ; Got object check.

3031 8BFE ; Call with object in accumulator, returns zero set if in pockets.

3032 8BFE

3033 8BFE 21 E5 81 gotob ld hl,numob ; number of objects in game.

3034 8C01 BE cp (hl) ; are we checking past the end?

3035 8C02 30 06 jr nc,gotob0 ; yes, we can't have a non-existent object.

3036 8C04 CD 0E 8C call findob ; find the object.

3037 8C07 FE FF gotob1 cp 255 ; in pockets?

3038 8C09 C9 ret

3039 8C0A 3E FE gotob0 ld a,254 ; missing.

3040 8C0C 18 F9 jr gotob1

3041 8C0E

3042 8C0E 21 63 7E findob ld hl,objdta ; objects.

3043 8C11 11 27 00 ld de,39 ; size of each object.

3044 8C14 A7 and a ; is it zero?

3045 8C15 28 04 jr z,fndob1 ; yes, skip loop.

3046 8C17 47 ld b,a ; loop counter in b.

3047 8C18 19 fndob2 add hl,de ; point to next one.

3048 8C19 10 FD djnz fndob2 ; repeat until we find address.

3049 8C1B 1E 21 fndob1 ld e,33 ; distance to room it's in.

3050 8C1D 19 add hl,de ; point to room.

3051 8C1E 7E ld a,(hl) ; fetch status.

3052 8C1F C9 ret

3053 8C20

3054 8C20 ; Drop object number at (dispx, dispy).

3055 8C20

3056 8C20 21 E5 81 drpob ld hl,numob ; number of objects in game.

3057 8C23 BE cp (hl) ; are we checking past the end?

3058 8C24 D0 ret nc ; yes, can't drop non-existent item.

3059 8C25 CD FE 8B call gotob ; make sure object is in inventory.

3060 8C28 3A EA 81 ld a,(scno) ; screen number.

3061 8C2B BE cp (hl) ; already on this screen?

3062 8C2C C8 ret z ; yes, nothing to do.

3063 8C2D 77 ld (hl),a ; bring onto screen.

3064 8C2E 23 inc hl ; point to x coord.

3065 8C2F 3A 10 82 ld a,(dispx) ; sprite x coordinate.

3066 8C32 77 ld (hl),a ; set x coord.

3067 8C33 23 inc hl ; point to object y.

3068 8C34 3A 11 82 ld a,(dispy) ; sprite y coordinate.

3069 8C37 77 ld (hl),a ; set the y position.

3070 8C38 11 DD FF ld de,65501 ; minus graphic size.

3071 8C3B 19 add hl,de ; point to graphics.

3072 8C3C CD 75 8B call dobj1 ; delete object sprite.

3073 8C3F 4E ld c,(hl) ; put ink in c register.

3074 8C40 C3 7C 8B jp cobj ; colour the object.

3075 8C43

3076 8C43 ; Seek objects at sprite position.

3077 8C43

3078 8C43 21 63 7E skobj ld hl,objdta ; pointer to objects.

3079 8C46 11 21 00 ld de,33 ; distance to room number.

3080 8C49 19 add hl,de ; point to room data.

3081 8C4A 11 27 00 ld de,39 ; size of each object.

3082 8C4D 3A E5 81 ld a,(numob) ; number of objects in game.

3083 8C50 47 ld b,a ; set up the loop counter.

3084 8C51 3A EA 81 skobj0 ld a,(scno) ; current room number.

3085 8C54 BE cp (hl) ; is object in here?

3086 8C55 CC 5E 8C call z,skobj1 ; yes, check coordinates.

3087 8C58 19 add hl,de ; point to next object in table.

3088 8C59 10 F6 djnz skobj0 ; repeat for all objects.

3089 8C5B 3E FF ld a,255 ; end of list and nothing found, return 255.

3090 8C5D C9 ret

3091 8C5E 23 skobj1 inc hl ; point to x coordinate.

3092 8C5F 7E ld a,(hl) ; get coordinate.

3093 8C60 DD 96 08 sub (ix+8) ; subtract sprite x.

3094 8C63 C6 0F add a,15 ; add sprite height minus one.

3095 8C65 FE 1F cp 31 ; within range?

3096 8C67 D2 7D 8C jp nc,skobj2 ; no, ignore object.

3097 8C6A 23 inc hl ; point to y coordinate now.

3098 8C6B 7E ld a,(hl) ; get coordinate.

3099 8C6C DD 96 09 sub (ix+9) ; subtract the sprite y.

3100 8C6F C6 0F add a,15 ; add sprite width minus one.

3101 8C71 FE 1F cp 31 ; within range?

3102 8C73 D2 7C 8C jp nc,skobj3 ; no, ignore object.

3103 8C76 D1 pop de ; remove return address from stack.

3104 8C77 3A E5 81 ld a,(numob) ; objects in game.

3105 8C7A 90 sub b ; subtract loop counter.

3106 8C7B C9 ret ; accumulator now points to object.

3107 8C7C 2B skobj3 dec hl ; back to y position.

3108 8C7D 2B skobj2 dec hl ; back to room.

3109 8C7E C9 ret

3110 8C7F

3111 8C7F

3112 8C7F ; Spawn a new sprite.

3113 8C7F

3114 8C7F 21 1C 98 spawn ld hl,sprtab ; sprite table.

3115 8C82 3E 0C numsp1 ld a,NUMSPR ; number of sprites.

3116 8C84 11 11 00 ld de,TABSIZ ; size of each entry.

3117 8C87 08 spaw0 ex af,af' ; store loop counter.

3118 8C88 7E ld a,(hl) ; get sprite type.

3119 8C89 3C inc a ; is it an unused slot?

3120 8C8A 28 05 jr z,spaw1 ; yes, we can use this one.

3121 8C8C 19 add hl,de ; point to next sprite in table.

3122 8C8D 08 ex af,af' ; restore loop counter.

3123 8C8E 3D dec a ; one less iteration.

3124 8C8F 20 F6 jr nz,spaw0 ; keep going until we find a slot.

3125 8C91

3126 8C91 ; Didn't find one but drop through and set up a dummy sprite instead.

3127 8C91

3128 8C91 DD E5 spaw1 push ix ; existing sprite address on stack.

3129 8C93 22 CE 8C ld (spptr),hl ; store spawned sprite address.

3130 8C96 71 ld (hl),c ; set the type.

3131 8C97 23 inc hl ; point to image.

3132 8C98 70 ld (hl),b ; set the image.

3133 8C99 23 inc hl ; next byte.

3134 8C9A 36 00 ld (hl),0 ; frame zero.

3135 8C9C 23 inc hl ; next byte.

3136 8C9D DD 7E 08 ld a,(ix+X) ; x coordinate.

3137 8CA0 77 ld (hl),a ; set sprite coordinate.

3138 8CA1 23 inc hl ; next byte.

3139 8CA2 DD 7E 09 ld a,(ix+Y) ; y coordinate.

3140 8CA5 77 ld (hl),a ; set sprite coordinate.

3141 8CA6 23 inc hl ; next byte.

3142 8CA7 EB ex de,hl ; swap address into de.

3143 8CA8 2A CE 8C ld hl,(spptr) ; restore address of details.

3144 8CAB 01 05 00 ld bc,5 ; number of bytes to duplicate.

3145 8CAE ED B0 ldir ; copy first version to new version.

3146 8CB0 EB ex de,hl ; swap address into de.

3147 8CB1 DD 7E 0A ld a,(ix+10) ; direction of original.

3148 8CB4 77 ld (hl),a ; set the direction.

3149 8CB5 23 inc hl ; next byte.

3150 8CB6 70 ld (hl),b ; reset parameter.

3151 8CB7 23 inc hl ; next byte.

3152 8CB8 70 ld (hl),b ; reset parameter.

3153 8CB9 23 inc hl ; next byte.

3154 8CBA 70 ld (hl),b ; reset parameter.

3155 8CBB 23 inc hl ; next byte.

3156 8CBC 70 ld (hl),b ; reset parameter.

3157 8CBD DD 2A CE 8C rtssp ld ix,(spptr) ; address of new sprite.

3158 8CC1 CD 7A 6A evis1 call evnt09 ; call sprite initialisation event.

3159 8CC4 DD 2A CE 8C ld ix,(spptr) ; address of new sprite.

3160 8CC8 CD C8 93 call sspria ; display the new sprite.

3161 8CCB DD E1 pop ix ; address of original sprite.

3162 8CCD C9 ret

3163 8CCE

3164 8CCE 00 00 spptr defw 0 ; spawned sprite pointer.

3165 8CD0 00 seed defb 0 ; seed for random numbers.

3166 8CD1 score defb '000000' ; player's score.

3166 8CD1 303030303030

3167 8CD7 hiscor defb '000000' ; high score.

3167 8CD7 303030303030

3168 8CDD bonus defb '000000' ; bonus.

3168 8CDD 303030303030

3169 8CE3 00 3C grbase defw 15360 ; graphics base address.

3170 8CE5

3171 8CE5 7B checkx ld a,e ; x position.

3172 8CE6 FE 18 cp 24 ; off screen?

3173 8CE8 D8 ret c ; no, it's okay.

3174 8CE9 E1 pop hl ; remove return address from stack.

3175 8CEA C9 ret

3176 8CEB

3177 8CEB ; Displays the current score.

3178 8CEB

3179 8CEB CD 4A 92 dscor call preprt ; set up font and print position.

3180 8CEE CD E5 8C call checkx ; make sure we're in a printable range.

3181 8CF1 3A B7 91 ld a,(prtmod) ; get print mode.

3182 8CF4 A7 and a ; standard size text?

3183 8CF5 C2 1B 8D jp nz,bscor0 ; no, show double-height.

3184 8CF8 C5 dscor0 push bc ; place counter onto the stack.

3185 8CF9 E5 push hl

3186 8CFA 7E ld a,(hl) ; fetch character.

3187 8CFB CD DB 8D call pchar ; display character.

3188 8CFE CD C3 8D call gaadd ; get attribute address.

3189 8D01 3A 8D 5C ld a,(23693) ; current cell colours.

3190 8D04 77 ld (hl),a ; write to attribute cell.

3191 8D05 21 11 82 ld hl,dispy ; y coordinate.

3192 8D08 34 inc (hl) ; move along one.

3193 8D09 E1 pop hl

3194 8D0A 23 inc hl ; next score column.

3195 8D0B C1 pop bc ; retrieve character counter.

3196 8D0C 10 EA djnz dscor0 ; repeat for all digits.

3197 8D0E 2A 14 82 ld hl,(blkptr) ; blocks.

3198 8D11 22 E3 8C ld (grbase),hl ; set graphics base.

3199 8D14 2A 10 82 dscor2 ld hl,(dispx) ; general coordinates.

3200 8D17 22 05 82 ld (charx),hl ; set up display coordinates.

3201 8D1A C9 ret

3202 8D1B

3203 8D1B ; Displays the current score in double-height characters.

3204 8D1B

3205 8D1B C5 bscor0 push bc ; place counter onto the stack.

3206 8D1C E5 push hl

3207 8D1D 7E ld a,(hl) ; fetch character.

3208 8D1E CD DB 91 call bchar ; display big char.

3209 8D21 E1 pop hl

3210 8D22 23 inc hl ; next score column.

3211 8D23 C1 pop bc ; retrieve character counter.

3212 8D24 10 F5 djnz bscor0 ; repeat for all digits.

3213 8D26 C3 14 8D jp dscor2 ; tidy up line and column variables.

3214 8D29

3215 8D29 ; Adds number in the hl pair to the score.

3216 8D29

3217 8D29 11 D2 8C addsc ld de,score+1 ; ten thousands column.

3218 8D2C 01 10 27 ld bc,10000 ; amount to add each time.

3219 8D2F CD 4B 8D call incsc ; add to score.

3220 8D32 13 inc de ; thousands column.

3221 8D33 01 E8 03 ld bc,1000 ; amount to add each time.

3222 8D36 CD 4B 8D call incsc ; add to score.

3223 8D39 13 inc de ; hundreds column.

3224 8D3A 01 64 00 ld bc,100 ; amount to add each time.

3225 8D3D CD 4B 8D call incsc ; add to score.

3226 8D40 13 inc de ; tens column.

3227 8D41 01 0A 00 ld bc,10 ; amount to add each time.

3228 8D44 CD 4B 8D call incsc ; add to score.

3229 8D47 13 inc de ; units column.

3230 8D48 01 01 00 ld bc,1 ; units.

3231 8D4B E5 incsc push hl ; store amount to add.

3232 8D4C A7 and a ; clear the carry flag.

3233 8D4D ED 42 sbc hl,bc ; subtract from amount to add.

3234 8D4F 38 09 jr c,incsc0 ; too much, restore value.

3235 8D51 F1 pop af ; delete the previous amount from the stack.

3236 8D52 D5 push de ; store column position.

3237 8D53 CD 5C 8D call incsc2 ; do the increment.

3238 8D56 D1 pop de ; restore column.

3239 8D57 C3 4B 8D jp incsc ; repeat until all added.

3240 8D5A E1 incsc0 pop hl ; restore previous value.

3241 8D5B C9 ret

3242 8D5C 1A incsc2 ld a,(de) ; get amount.

3243 8D5D 3C inc a ; add one to column.

3244 8D5E 12 ld (de),a ; write new column total.

3245 8D5F FE 3A cp '9'+1 ; gone beyond range of digits?

3246 8D61 D8 ret c ; no, carry on.

3247 8D62 3E 30 ld a,'0' ; mae it zero.

3248 8D64 12 ld (de),a ; write new column total.

3249 8D65 1B dec de ; back one column.

3250 8D66 18 F4 jr incsc2

3251 8D68

3252 8D68 ; Add bonus to score.

3253 8D68

3254 8D68 11 D6 8C addbo ld de,score+5 ; last score digit.

3255 8D6B 21 E2 8C ld hl,bonus+5 ; last bonus digit.

3256 8D6E A7 and a ; clear carry.

3257 8D6F 01 30 06 ld bc,6\*256+48 ; 6 digits to add, ASCII '0' in c.

3258 8D72 1A addbo0 ld a,(de) ; get score.

3259 8D73 8E adc a,(hl) ; add bonus.

3260 8D74 91 sub c ; 0 to 18.

3261 8D75 71 ld (hl),c ; zeroise bonus.

3262 8D76 2B dec hl ; next bonus.

3263 8D77 FE 3A cp 58 ; carried?

3264 8D79 38 02 jr c,addbo1 ; no, do next one.

3265 8D7B D6 0A sub 10 ; subtract 10.

3266 8D7D 12 addbo1 ld (de),a ; write new score.

3267 8D7E 1B dec de ; next score digit.

3268 8D7F 3F ccf ; set carry for next digit.

3269 8D80 10 F0 djnz addbo0 ; repeat for all 6 digits.

3270 8D82 C9 ret

3271 8D83

3272 8D83 ; Swap score and bonus.

3273 8D83

3274 8D83 11 D1 8C swpsb ld de,score ; first score digit.

3275 8D86 21 DD 8C ld hl,bonus ; first bonus digit.

3276 8D89 06 06 ld b,6 ; digits to add.

3277 8D8B 1A swpsb0 ld a,(de) ; get score and bonus digits.

3278 8D8C 4E ld c,(hl)

3279 8D8D EB ex de,hl ; swap pointers.

3280 8D8E 71 ld (hl),c ; write bonus and score digits.

3281 8D8F 12 ld (de),a

3282 8D90 23 inc hl ; next score and bonus.

3283 8D91 13 inc de

3284 8D92 10 F7 djnz swpsb0 ; repeat for all 6 digits.

3285 8D94 C9 ret

3286 8D95

3287 8D95 ; Get print address.

3288 8D95

3289 8D95 3A 10 82 gprad ld a,(dispx) ; returns scr. add. in de.

3290 8D98 5F ld e,a ; place in e for now.

3291 8D99 E6 18 and 24 ; which of 3 segments do we need?

3292 8D9B C6 40 add a,64 ; add 64 for start address of screen.

3293 8D9D 57 ld d,a ; that's our high byte.

3294 8D9E 7B ld a,e ; restore x coordinate.

3295 8D9F 0F rrca ; multiply by 32.

3296 8DA0 0F rrca

3297 8DA1 0F rrca

3298 8DA2 E6 E0 and 224 ; lines within segment.

3299 8DA4 5F ld e,a ; set up low byte for x.

3300 8DA5 3A 11 82 ld a,(dispy) ; now get y coordinate.

3301 8DA8 83 add a,e ; add to low byte.

3302 8DA9 5F ld e,a ; final low byte.

3303 8DAA C9 ret

3304 8DAB

3305 8DAB ; Get property buffer address of char at (dispx, dispy) in hl.

3306 8DAB

3307 8DAB 3A 10 82 pradd ld a,(dispx) ; x coordinate.

3308 8DAE 0F rrca ; multiply by 32.

3309 8DAF 0F rrca

3310 8DB0 0F rrca

3311 8DB1 6F ld l,a ; store shift in l.

3312 8DB2 E6 03 and 3 ; high byte bits.

3313 8DB4 C6 FD add a,253 ; 88 \* 256 = 64768, start of properties map.

3314 8DB6 67 ld h,a ; that's our high byte.

3315 8DB7 7D ld a,l ; restore shift result.

3316 8DB8 E6 E0 and 224 ; only want low bits.

3317 8DBA 6F ld l,a ; put into low byte.

3318 8DBB 3A 11 82 ld a,(dispy) ; fetch y coordinate.

3319 8DBE E6 1F and 31 ; should be in range 0 - 31.

3320 8DC0 85 add a,l ; add to low byte.

3321 8DC1 6F ld l,a ; new low byte.

3322 8DC2 C9 ret

3323 8DC3

3324 8DC3 ; Get attribute address of char at (dispx, dispy) in hl.

3325 8DC3

3326 8DC3 3A 10 82 gaadd ld a,(dispx) ; x coordinate.

3327 8DC6 0F rrca ; multiply by 32.

3328 8DC7 0F rrca

3329 8DC8 0F rrca

3330 8DC9 6F ld l,a ; store shift in l.

3331 8DCA E6 03 and 3 ; high byte bits.

3332 8DCC C6 58 add a,88 ; 88 \* 256 = 22528, start of screen attributes.

3333 8DCE 67 ld h,a ; that's our high byte.

3334 8DCF 7D ld a,l ; restore shift result.

3335 8DD0 E6 E0 and 224 ; only want low bits.

3336 8DD2 6F ld l,a ; put into low byte.

3337 8DD3 3A 11 82 ld a,(dispy) ; fetch y coordinate.

3338 8DD6 E6 1F and 31 ; should be in range 0 - 31.

3339 8DD8 85 add a,l ; add to low byte.

3340 8DD9 6F ld l,a ; new low byte.

3341 8DDA C9 ret

3342 8DDB

3343 8DDB 07 pchar rlca ; multiply char by 8.

3344 8DDC 07 rlca

3345 8DDD 07 rlca

3346 8DDE 5F ld e,a ; store shift in e.

3347 8DDF E6 07 and 7 ; only want high byte bits.

3348 8DE1 57 ld d,a ; store in d.

3349 8DE2 7B ld a,e ; restore shifted value.

3350 8DE3 E6 F8 and 248 ; only want low byte bits.

3351 8DE5 5F ld e,a ; that's the low byte.

3352 8DE6 2A E3 8C ld hl,(grbase) ; address of graphics.

3353 8DE9 19 add hl,de ; add displacement.

3354 8DEA CD 95 8D pchark call gprad ; get screen address.

3355 8DED ; ldi ; transfer byte.

3356 8DED ; dec de ; back again.

3357 8DED ; inc d ; next screen row down.

3358 8DED ; ldi ; transfer byte.

3359 8DED ; dec de ; back again.

3360 8DED ; inc d ; next screen row down.

3361 8DED ; ldi ; transfer byte.

3362 8DED ; dec de ; back again.

3363 8DED ; inc d ; next screen row down.

3364 8DED ; ldi ; transfer byte.

3365 8DED ; dec de ; back again.

3366 8DED ; inc d ; next screen row down.

3367 8DED ; ldi ; transfer byte.

3368 8DED ; dec de ; back again.

3369 8DED ; inc d ; next screen row down.

3370 8DED ; ldi ; transfer byte.

3371 8DED ; dec de ; back again.

3372 8DED ; inc d ; next screen row down.

3373 8DED ; ldi ; transfer byte.

3374 8DED ; dec de ; back again.

3375 8DED ; inc d ; next screen row down.

3376 8DED ; ldi ; transfer byte.

3377 8DED 7E ld a,(hl) ; get image byte.

3378 8DEE 12 ld (de),a ; copy to screen.

3379 8DEF 23 inc hl ; next image byte.

3380 8DF0 14 inc d ; next screen row down.

3381 8DF1 7E ld a,(hl) ; get image byte.

3382 8DF2 12 ld (de),a ; copy to screen.

3383 8DF3 23 inc hl ; next image byte.

3384 8DF4 14 inc d ; next screen row down.

3385 8DF5 7E ld a,(hl) ; get image byte.

3386 8DF6 12 ld (de),a ; copy to screen.

3387 8DF7 23 inc hl ; next image byte.

3388 8DF8 14 inc d ; next screen row down.

3389 8DF9 7E ld a,(hl) ; get image byte.

3390 8DFA 12 ld (de),a ; copy to screen.

3391 8DFB 23 inc hl ; next image byte.

3392 8DFC 14 inc d ; next screen row down.

3393 8DFD 7E ld a,(hl) ; get image byte.

3394 8DFE 12 ld (de),a ; copy to screen.

3395 8DFF 23 inc hl ; next image byte.

3396 8E00 14 inc d ; next screen row down.

3397 8E01 7E ld a,(hl) ; get image byte.

3398 8E02 12 ld (de),a ; copy to screen.

3399 8E03 23 inc hl ; next image byte.

3400 8E04 14 inc d ; next screen row down.

3401 8E05 7E ld a,(hl) ; get image byte.

3402 8E06 12 ld (de),a ; copy to screen.

3403 8E07 23 inc hl ; next image byte.

3404 8E08 14 inc d ; next screen row down.

3405 8E09 7E ld a,(hl) ; get image byte.

3406 8E0A 12 ld (de),a ; copy to screen.

3407 8E0B C9 ret

3408 8E0C

3409 8E0C ; Print attributes, properties and pixels.

3410 8E0C

3411 8E0C 00 colpat defb 0

3412 8E0D

3413 8E0D 47 pattr ld b,a ; store cell in b register for now.

3414 8E0E 5F ld e,a ; displacement in e.

3415 8E0F 16 00 ld d,0 ; no high byte.

3416 8E11 2A 18 82 ld hl,(proptr) ; pointer to properties.

3417 8E14 19 add hl,de ; property cell address.

3418 8E15 4E ld c,(hl) ; fetch byte.

3419 8E16 79 ld a,c ; put into accumulator.

3420 8E17 FE 08 cp COLECT ; is it a collectable?

3421 8E19 C2 20 8E jp nz,pattr1 ; no, carry on as normal.

3422 8E1C 78 ld a,b ; restore cell.

3423 8E1D 32 0C 8E ld (colpat),a ; store collectable block.

3424 8E20 CD AB 8D pattr1 call pradd ; get property buffer address.

3425 8E23 71 ld (hl),c ; write property.

3426 8E24 78 ld a,b ; get block number.

3427 8E25

3428 8E25 ; Print attributes, no properties.

3429 8E25

3430 8E25 5F panp ld e,a ; displacement in e.

3431 8E26 16 00 ld d,0 ; no high byte.

3432 8E28 2A 16 82 ld hl,(colptr) ; pointer to colours.

3433 8E2B 19 add hl,de ; colour cell address.

3434 8E2C 4E ld c,(hl) ; fetch byte.

3435 8E2D CD C3 8D call gaadd ; get attribute address.

3436 8E30 71 ld (hl),c ; write colour.

3437 8E31 78 ld a,b ; restore cell.

3438 8E32

3439 8E32 ; Print character pixels, no more.

3440 8E32

3441 8E32 CD DB 8D pchr call pchar ; show character in accumulator.

3442 8E35 21 11 82 ld hl,dispy ; y coordinate.

3443 8E38 34 inc (hl) ; move along one.

3444 8E39 C9 ret

3445 8E3A

3446 8E3A ; Shifter sprite routine for objects.

3447 8E3A

3448 8E3A EE 07 sprit7 xor 7

3449 8E3C 3C inc a

3450 8E3D CB 15 sprit3 rl l ; shift into position.

3451 8E3F CB 11 rl c

3452 8E41 CB 14 rl h

3453 8E43 3D dec a ; one less iteration.

3454 8E44 C2 3D 8E jp nz,sprit3

3455 8E47 7D ld a,l

3456 8E48 69 ld l,c

3457 8E49 4C ld c,h

3458 8E4A 67 ld h,a

3459 8E4B C3 76 8E jp sprit0 ; now apply to screen.

3460 8E4E

3461 8E4E E5 sprite push hl ; store sprite graphic address.

3462 8E4F CD AB 93 call scadd ; get screen address in hl.

3463 8E52 EB ex de,hl ; switch to de.

3464 8E53 E1 pop hl ; restore graphic address.

3465 8E54 3A 11 82 ld a,(dispy) ; y position.

3466 8E57 E6 07 and 7 ; position straddling cells.

3467 8E59 47 ld b,a ; store in b register.

3468 8E5A 3E 10 ld a,16 ; pixel height.

3469 8E5C 08 sprit1 ex af,af'

3470 8E5D 4E ld c,(hl) ; fetch first byte.

3471 8E5E 23 inc hl ; next byte.

3472 8E5F E5 push hl ; store source address.

3473 8E60 6E ld l,(hl)

3474 8E61 26 00 ld h,0

3475 8E63 78 ld a,b ; position straddling cells.

3476 8E64 A7 and a ; is it zero?

3477 8E65 28 0F jr z,sprit0 ; yes, apply to screen.

3478 8E67 FE 05 cp 5

3479 8E69 30 CF jr nc,sprit7

3480 8E6B A7 and a ; clear carry.

3481 8E6C CB 19 sprit2 rr c

3482 8E6E CB 1D rr l

3483 8E70 CB 1C rr h

3484 8E72 3D dec a

3485 8E73 C2 6C 8E jp nz,sprit2

3486 8E76 1A sprit0 ld a,(de) ; fetch screen image.

3487 8E77 A9 xor c ; merge with graphic.

3488 8E78 12 ld (de),a ; write to screen.

3489 8E79 1C inc e ; next screen byte.

3490 8E7A 1A ld a,(de) ; fetch screen image.

3491 8E7B AD xor l ; combine with graphic.

3492 8E7C 12 ld (de),a ; write to screen.

3493 8E7D 13 inc de ; next screen address.

3494 8E7E 1A ld a,(de) ; fetch screen image.

3495 8E7F AC xor h ; combine with graphic.

3496 8E80 12 ld (de),a ; write to screen.

3497 8E81 1B dec de ; left to middle byte.

3498 8E82 1D dec e ; back to start byte.

3499 8E83 14 inc d ; increment line number.

3500 8E84 7A ld a,d ; segment address.

3501 8E85 E6 07 and 7 ; reached end of segment?

3502 8E87 C2 95 8E jp nz,sprit6 ; no, just do next line within cell.

3503 8E8A 7B ld a,e ; low byte.

3504 8E8B C6 20 add a,32 ; look down.

3505 8E8D 5F ld e,a ; new address.

3506 8E8E DA 95 8E jp c,sprit6 ; done.

3507 8E91 7A ld a,d ; high byte.

3508 8E92 D6 08 sub 8 ; start of segment.

3509 8E94 57 ld d,a ; new high byte.

3510 8E95 E1 sprit6 pop hl ; restore source address.

3511 8E96 23 inc hl ; next source byte.

3512 8E97 08 ex af,af'

3513 8E98 3D dec a

3514 8E99 C2 5C 8E jp nz,sprit1

3515 8E9C C9 ret

3516 8E9D

3517 8E9D ; Get room address.

3518 8E9D

3519 8E9D 3A EA 81 groom ld a,(scno) ; screen number.

3520 8EA0 11 00 00 groomx ld de,0 ; start at zero.

3521 8EA3 2A 1A 82 ld hl,(scrptr) ; pointer to screens.

3522 8EA6 A7 and a ; is it the first one?

3523 8EA7 28 0A groom1 jr z,groom0 ; no more screens to skip.

3524 8EA9 4E ld c,(hl) ; low byte of screen size.

3525 8EAA 23 inc hl ; point to high byte.

3526 8EAB 46 ld b,(hl) ; high byte of screen size.

3527 8EAC 23 inc hl ; next address.

3528 8EAD EB ex de,hl ; put total in hl, pointer in de.

3529 8EAE 09 add hl,bc ; skip a screen.

3530 8EAF EB ex de,hl ; put total in de, pointer in hl.

3531 8EB0 3D dec a ; one less iteration.

3532 8EB1 18 F4 jr groom1 ; loop until we reach the end.

3533 8EB3 2A 1A 82 groom0 ld hl,(scrptr) ; pointer to screens.

3534 8EB6 19 add hl,de ; add displacement.

3535 8EB7 3A 35 6E ld a,(numsc) ; number of screens.

3536 8EBA 16 00 ld d,0 ; zeroise high byte.

3537 8EBC 5F ld e,a ; displacement in de.

3538 8EBD 19 add hl,de ; add double displacement to address.

3539 8EBE 19 add hl,de

3540 8EBF C9 ret

3541 8EC0

3542 8EC0 ; Draw present room.

3543 8EC0

3544 8EC0 3A E1 81 droom ld a,(wintop) ; window top.

3545 8EC3 32 10 82 ld (dispx),a ; set x coordinate.

3546 8EC6 2A 14 82 droom2 ld hl,(blkptr) ; blocks.

3547 8EC9 22 E3 8C ld (grbase),hl ; set graphics base.

3548 8ECC CD 9D 8E call groom ; get address of current room.

3549 8ECF AF xor a ; zero in accumulator.

3550 8ED0 32 14 8F ld (comcnt),a ; reset compression counter.

3551 8ED3 3A E3 81 ld a,(winhgt) ; height of window.

3552 8ED6 F5 droom0 push af ; store row counter.

3553 8ED7 3A E2 81 ld a,(winlft) ; window left edge.

3554 8EDA 32 11 82 ld (dispy),a ; set cursor position.

3555 8EDD 3A E4 81 ld a,(winwid) ; width of window.

3556 8EE0 F5 droom1 push af ; store column counter.

3557 8EE1 CD F9 8E call flbyt ; decompress next byte on the fly.

3558 8EE4 E5 push hl ; store address of cell.

3559 8EE5 CD 0D 8E call pattr ; show attributes and block.

3560 8EE8 E1 pop hl ; restore cell address.

3561 8EE9 F1 pop af ; restore loop counter.

3562 8EEA 3D dec a ; one less column.

3563 8EEB 20 F3 jr nz,droom1 ; repeat for entire line.

3564 8EED 3A 10 82 ld a,(dispx) ; x coord.

3565 8EF0 3C inc a ; move down one line.

3566 8EF1 32 10 82 ld (dispx),a ; set new position.

3567 8EF4 F1 pop af ; restore row counter.

3568 8EF5 3D dec a ; one less row.

3569 8EF6 20 DE jr nz,droom0 ; repeat for all rows.

3570 8EF8 C9 ret

3571 8EF9

3572 8EF9 ; Decompress bytes on-the-fly.

3573 8EF9

3574 8EF9 3A 14 8F flbyt ld a,(comcnt) ; compression counter.

3575 8EFC A7 and a ; any more to decompress?

3576 8EFD 20 0C jr nz,flbyt1 ; yes.

3577 8EFF 7E ld a,(hl) ; fetch next byte.

3578 8F00 23 inc hl ; point to next cell.

3579 8F01 FE FF cp 255 ; is this byte a control code?

3580 8F03 C0 ret nz ; no, this byte is uncompressed.

3581 8F04 7E ld a,(hl) ; fetch byte type.

3582 8F05 32 13 8F ld (combyt),a ; set up the type.

3583 8F08 23 inc hl ; point to quantity.

3584 8F09 7E ld a,(hl) ; get quantity.

3585 8F0A 23 inc hl ; point to next byte.

3586 8F0B 3D flbyt1 dec a ; one less.

3587 8F0C 32 14 8F ld (comcnt),a ; store new quantity.

3588 8F0F 3A 13 8F ld a,(combyt) ; byte to expand.

3589 8F12 C9 ret

3590 8F13

3591 8F13

3592 8F13 00 combyt defb 0 ; byte type compressed.

3593 8F14 00 comcnt defb 0 ; compression counter.

3594 8F15

3595 8F15 ; Ladder down check.

3596 8F15

3597 8F15 DD 7E 08 laddd ld a,(ix+8) ; x coordinate.

3598 8F18 E6 FE and 254 ; make it even.

3599 8F1A DD 77 08 ld (ix+8),a ; reset it.

3600 8F1D DD 66 09 ld h,(ix+9) ; y coordinate.

3601 8F20 C6 10 numsp5 add a,16 ; look down 16 pixels.

3602 8F22 6F ld l,a ; coords in hl.

3603 8F23 18 0E jr laddv

3604 8F25

3605 8F25 ; Ladder up check.

3606 8F25

3607 8F25 DD 7E 08 laddu ld a,(ix+8) ; x coordinate.

3608 8F28 E6 FE and 254 ; make it even.

3609 8F2A DD 77 08 ld (ix+8),a ; reset it.

3610 8F2D DD 66 09 ld h,(ix+9) ; y coordinate.

3611 8F30 C6 0E numsp6 add a,14 ; look 2 pixels above feet.

3612 8F32 6F ld l,a ; coords in hl.

3613 8F33 22 10 82 laddv ld (dispx),hl ; set up test coordinates.

3614 8F36 CD 9C 90 call tstbl ; get map address.

3615 8F39 CD EA 8F call ldchk ; standard ladder check.

3616 8F3C C0 ret nz ; no way through.

3617 8F3D 23 inc hl ; look right one cell.

3618 8F3E CD EA 8F call ldchk ; do the check.

3619 8F41 C0 ret nz ; impassable.

3620 8F42 3A 11 82 ld a,(dispy) ; y coordinate.

3621 8F45 E6 07 and 7 ; position straddling block cells.

3622 8F47 C8 ret z ; no more checks needed.

3623 8F48 23 inc hl ; look to third cell.

3624 8F49 CD EA 8F call ldchk ; do the check.

3625 8F4C C9 ret ; return with zero flag set accordingly.

3626 8F4D

3627 8F4D ; Can go up check.

3628 8F4D

3629 8F4D DD 7E 08 cangu ld a,(ix+8) ; x coordinate.

3630 8F50 DD 66 09 ld h,(ix+9) ; y coordinate.

3631 8F53 D6 02 sub 2 ; look up 2 pixels.

3632 8F55 6F ld l,a ; coords in hl.

3633 8F56 22 10 82 ld (dispx),hl ; set up test coordinates.

3634 8F59 CD 9C 90 call tstbl ; get map address.

3635 8F5C CD C0 8F call lrchk ; standard left/right check.

3636 8F5F C0 ret nz ; no way through.

3637 8F60 23 inc hl ; look right one cell.

3638 8F61 CD C0 8F call lrchk ; do the check.

3639 8F64 C0 ret nz ; impassable.

3640 8F65 3A 11 82 ld a,(dispy) ; y coordinate.

3641 8F68 E6 07 and 7 ; position straddling block cells.

3642 8F6A C8 ret z ; no more checks needed.

3643 8F6B 23 inc hl ; look to third cell.

3644 8F6C CD C0 8F call lrchk ; do the check.

3645 8F6F C9 ret ; return with zero flag set accordingly.

3646 8F70

3647 8F70 ; Can go down check.

3648 8F70

3649 8F70 DD 7E 08 cangd ld a,(ix+8) ; x coordinate.

3650 8F73 DD 66 09 ld h,(ix+9) ; y coordinate.

3651 8F76 C6 10 numsp3 add a,16 ; look down 16 pixels.

3652 8F78 6F ld l,a ; coords in hl.

3653 8F79 22 10 82 ld (dispx),hl ; set up test coordinates.

3654 8F7C CD 9C 90 call tstbl ; get map address.

3655 8F7F CD CE 8F call plchk ; block, platform check.

3656 8F82 C0 ret nz ; no way through.

3657 8F83 23 inc hl ; look right one cell.

3658 8F84 CD CE 8F call plchk ; block, platform check.

3659 8F87 C0 ret nz ; impassable.

3660 8F88 3A 11 82 ld a,(dispy) ; y coordinate.

3661 8F8B E6 07 and 7 ; position straddling block cells.

3662 8F8D C8 ret z ; no more checks needed.

3663 8F8E 23 inc hl ; look to third cell.

3664 8F8F CD CE 8F call plchk ; block, platform check.

3665 8F92 C9 ret ; return with zero flag set accordingly.

3666 8F93

3667 8F93 ; Can go left check.

3668 8F93

3669 8F93 DD 6E 08 cangl ld l,(ix+8) ; x coordinate.

3670 8F96 DD 7E 09 ld a,(ix+9) ; y coordinate.

3671 8F99 D6 02 sub 2 ; look left 2 pixels.

3672 8F9B 67 ld h,a ; coords in hl.

3673 8F9C 18 09 jr cangh ; test if we can go there.

3674 8F9E

3675 8F9E ; Can go right check.

3676 8F9E

3677 8F9E DD 6E 08 cangr ld l,(ix+8) ; x coordinate.

3678 8FA1 DD 7E 09 ld a,(ix+9) ; y coordinate.

3679 8FA4 C6 10 add a,16 ; look right 16 pixels.

3680 8FA6 67 ld h,a ; coords in hl.

3681 8FA7

3682 8FA7 22 10 82 cangh ld (dispx),hl ; set up test coordinates.

3683 8FAA 06 03 cangh2 ld b,3 ; default rows to write.

3684 8FAC 7D ld a,l ; x position.

3685 8FAD E6 07 and 7 ; does x straddle cells?

3686 8FAF 20 01 jr nz,cangh0 ; yes, loop counter is good.

3687 8FB1 05 dec b ; one less row to write.

3688 8FB2 CD 9C 90 cangh0 call tstbl ; get map address.

3689 8FB5 11 20 00 ld de,32 ; distance to next cell.

3690 8FB8 CD C0 8F cangh1 call lrchk ; standard left/right check.

3691 8FBB C0 ret nz ; no way through.

3692 8FBC 19 add hl,de ; look down.

3693 8FBD 10 F9 djnz cangh1

3694 8FBF C9 ret

3695 8FC0

3696 8FC0 ; Check left/right movement is okay.

3697 8FC0

3698 8FC0 7E lrchk ld a,(hl) ; fetch map cell.

3699 8FC1 FE 02 cp WALL ; is it passable?

3700 8FC3 28 06 jr z,lrchkx ; no.

3701 8FC5 FE 04 cp FODDER ; fodder has to be dug.

3702 8FC7 28 02 jr z,lrchkx ; not passable.

3703 8FC9 AF always xor a ; report it as okay.

3704 8FCA C9 ret

3705 8FCB AF lrchkx xor a ; reset all bits.

3706 8FCC 3C inc a

3707 8FCD C9 ret

3708 8FCE

3709 8FCE ; Check platform or solid item is not in way.

3710 8FCE

3711 8FCE 7E plchk ld a,(hl) ; fetch map cell.

3712 8FCF FE 02 cp WALL ; is it passable?

3713 8FD1 28 F8 jr z,lrchkx ; no.

3714 8FD3 FE 04 cp FODDER ; fodder has to be dug.

3715 8FD5 28 F4 jr z,lrchkx ; not passable.

3716 8FD7 FE 01 cp PLATFM ; platform is solid.

3717 8FD9 28 06 jr z,plchkx ; not passable.

3718 8FDB FE 03 cp LADDER ; is it a ladder?

3719 8FDD 28 EC jr z,lrchkx ; on ladder, deny movement.

3720 8FDF AF plchk0 xor a ; report it as okay.

3721 8FE0 C9 ret

3722 8FE1 3A 10 82 plchkx ld a,(dispx) ; x coordinate.

3723 8FE4 E6 07 and 7 ; position straddling blocks.

3724 8FE6 28 E3 jr z,lrchkx ; on platform, deny movement.

3725 8FE8 18 F5 jr plchk0

3726 8FEA

3727 8FEA ; Check ladder is available.

3728 8FEA

3729 8FEA 7E ldchk ld a,(hl) ; fetch cell.

3730 8FEB FE 03 cp LADDER ; is it a ladder?

3731 8FED C9 ret ; return with zero flag set accordingly.

3732 8FEE

3733 8FEE ; Get collectables.

3734 8FEE

3735 8FEE 06 08 getcol ld b,COLECT ; collectable blocks.

3736 8FF0 CD 54 90 call tded ; test for collectable blocks.

3737 8FF3 B8 cp b ; did we find one?

3738 8FF4 C0 ret nz ; none were found, job done.

3739 8FF5 CD FD 8F call gtblk ; get block.

3740 8FF8 CD 43 6D call evnt20 ; collected block event.

3741 8FFB 18 F1 jr getcol ; repeat until none left.

3742 8FFD

3743 8FFD ; Get collectable block.

3744 8FFD

3745 8FFD 36 00 gtblk ld (hl),0 ; make it empty now.

3746 8FFF 11 00 FD ld de,MAP ; map address.

3747 9002 A7 and a ; clear carry.

3748 9003 ED 52 sbc hl,de ; find cell number.

3749 9005 7D ld a,l ; get low byte of cell number.

3750 9006 E6 1F and 31 ; 0 - 31 is column.

3751 9008 57 ld d,a ; store y in d register.

3752 9009 29 add hl,hl ; multiply by 8.

3753 900A 29 add hl,hl

3754 900B 29 add hl,hl ; x is now in h.

3755 900C 5C ld e,h ; put x in e.

3756 900D ED 53 10 82 ld (dispx),de ; set display coordinates.

3757 9011

3758 9011 3A 0C 8E ld a,(colpat) ; get collectable block used on this screen.

3759 9014 07 rlca ; multiply char by 8.

3760 9015 07 rlca

3761 9016 07 rlca

3762 9017 5F ld e,a ; store shift in e.

3763 9018 E6 07 and 7 ; only want high byte bits.

3764 901A 57 ld d,a ; store in d.

3765 901B 7B ld a,e ; restore shifted value.

3766 901C E6 F8 and 248 ; only want low byte bits.

3767 901E 5F ld e,a ; that's the low byte.

3768 901F 21 36 6E ld hl,chgfx ; address of graphics.

3769 9022 19 add hl,de ; add displacement.

3770 9023 CD 95 8D call gprad ; get screen address.

3771 9026

3772 9026 1A ld a,(de) ; get image byte.

3773 9027 AE xor (hl) ; merge with screen image.

3774 9028 12 ld (de),a ; copy to screen.

3775 9029 23 inc hl ; next image byte.

3776 902A 14 inc d ; next screen row down.

3777 902B 1A ld a,(de) ; get image byte.

3778 902C AE xor (hl) ; merge with screen image.

3779 902D 12 ld (de),a ; copy to screen.

3780 902E 23 inc hl ; next image byte.

3781 902F 14 inc d ; next screen row down.

3782 9030 1A ld a,(de) ; get image byte.

3783 9031 AE xor (hl) ; merge with screen image.

3784 9032 12 ld (de),a ; copy to screen.

3785 9033 23 inc hl ; next image byte.

3786 9034 14 inc d ; next screen row down.

3787 9035 1A ld a,(de) ; get image byte.

3788 9036 AE xor (hl) ; merge with screen image.

3789 9037 12 ld (de),a ; copy to screen.

3790 9038 23 inc hl ; next image byte.

3791 9039 14 inc d ; next screen row down.

3792 903A 1A ld a,(de) ; get image byte.

3793 903B AE xor (hl) ; merge with screen image.

3794 903C 12 ld (de),a ; copy to screen.

3795 903D 23 inc hl ; next image byte.

3796 903E 14 inc d ; next screen row down.

3797 903F 1A ld a,(de) ; get image byte.

3798 9040 AE xor (hl) ; merge with screen image.

3799 9041 12 ld (de),a ; copy to screen.

3800 9042 23 inc hl ; next image byte.

3801 9043 14 inc d ; next screen row down.

3802 9044 1A ld a,(de) ; get image byte.

3803 9045 AE xor (hl) ; merge with screen image.

3804 9046 12 ld (de),a ; copy to screen.

3805 9047 23 inc hl ; next image byte.

3806 9048 14 inc d ; next screen row down.

3807 9049 1A ld a,(de) ; get image byte.

3808 904A AE xor (hl) ; merge with screen image.

3809 904B 12 ld (de),a ; copy to screen.

3810 904C CD C3 8D call gaadd ; get attribute address for block.

3811 904F 3A 46 6E ld a,(bcol) ; first block colour.

3812 9052 77 ld (hl),a ; set block attribute.

3813 9053 C9 ret

3814 9054

3815 9054 ; Touched deadly block check.

3816 9054 ; Returns with DEADLY (must be non-zero) in accumulator if true.

3817 9054

3818 9054 DD 6E 08 tded ld l,(ix+8) ; x coordinate.

3819 9057 DD 66 09 ld h,(ix+9) ; y coordinate.

3820 905A 22 10 82 ld (dispx),hl ; set up test coordinates.

3821 905D CD 9C 90 call tstbl ; get map address.

3822 9060 11 1F 00 ld de,31 ; default distance to next line down.

3823 9063 B8 cp b ; is this the required block?

3824 9064 C8 ret z ; yes.

3825 9065 23 inc hl ; next cell.

3826 9066 7E ld a,(hl) ; fetch type.

3827 9067 B8 cp b ; is this deadly/custom?

3828 9068 C8 ret z ; yes.

3829 9069 3A 11 82 ld a,(dispy) ; horizontal position.

3830 906C 4F ld c,a ; store column in c register.

3831 906D E6 07 and 7 ; is it straddling cells?

3832 906F 28 05 jr z,tded0 ; no.

3833 9071 23 inc hl ; last cell.

3834 9072 7E ld a,(hl) ; fetch type.

3835 9073 B8 cp b ; is this the block?

3836 9074 C8 ret z ; yes.

3837 9075 1B dec de ; one less cell to next row down.

3838 9076 19 tded0 add hl,de ; point to next row.

3839 9077 7E ld a,(hl) ; fetch left cell block.

3840 9078 B8 cp b ; is this fatal?

3841 9079 C8 ret z ; yes.

3842 907A 23 inc hl ; next cell.

3843 907B 7E ld a,(hl) ; fetch type.

3844 907C B8 cp b ; is this fatal?

3845 907D C8 ret z ; yes.

3846 907E 79 ld a,c ; horizontal position.

3847 907F E6 07 and 7 ; is it straddling cells?

3848 9081 28 04 jr z,tded1 ; no.

3849 9083 23 inc hl ; last cell.

3850 9084 7E ld a,(hl) ; fetch type.

3851 9085 B8 cp b ; is this fatal?

3852 9086 C8 ret z ; yes.

3853 9087 3A 10 82 tded1 ld a,(dispx) ; vertical position.

3854 908A E6 07 and 7 ; is it straddling cells?

3855 908C C8 ret z ; no, job done.

3856 908D 19 add hl,de ; point to next row.

3857 908E 7E ld a,(hl) ; fetch left cell block.

3858 908F B8 cp b ; is this fatal?

3859 9090 C8 ret z ; yes.

3860 9091 23 inc hl ; next cell.

3861 9092 7E ld a,(hl) ; fetch type.

3862 9093 B8 cp b ; is this fatal?

3863 9094 C8 ret z ; yes.

3864 9095 79 ld a,c ; horizontal position.

3865 9096 E6 07 and 7 ; is it straddling cells?

3866 9098 C8 ret z ; no.

3867 9099 23 inc hl ; last cell.

3868 909A 7E ld a,(hl) ; fetch final type.

3869 909B C9 ret ; return with final type in accumulator.

3870 909C

3871 909C

3872 909C ; Fetch block type at (dispx, dispy).

3873 909C

3874 909C 3A 10 82 tstbl ld a,(dispx) ; fetch x coord.

3875 909F 07 rlca ; divide by 8,

3876 90A0 07 rlca ; and multiply by 32.

3877 90A1 57 ld d,a ; store in d.

3878 90A2 E6 E0 and 224 ; mask off high bits.

3879 90A4 5F ld e,a ; low byte.

3880 90A5 7A ld a,d ; restore shift result.

3881 90A6 E6 03 and 3 ; high bits.

3882 90A8 57 ld d,a ; got displacement in de.

3883 90A9 3A 11 82 ld a,(dispy) ; y coord.

3884 90AC 1F rra ; divide by 8.

3885 90AD 1F rra

3886 90AE 1F rra

3887 90AF E6 1F and 31 ; only want 0 - 31.

3888 90B1 83 add a,e ; add to displacement.

3889 90B2 5F ld e,a ; displacement in de.

3890 90B3 21 00 FD ld hl,MAP ; position of dummy screen.

3891 90B6 19 add hl,de ; point to address.

3892 90B7 7E ld a,(hl) ; fetch byte there.

3893 90B8 C9 ret

3894 90B9

3895 90B9 ; Jump - if we can.

3896 90B9 ; Requires initial speed to be set up in accumulator prior to call.

3897 90B9

3898 90B9 ED 44 jump neg ; switch sign so we jump up.

3899 90BB 4F ld c,a ; store in c register.

3900 90BC ; ld a,(ix+8) ; x coordinate.

3901 90BC ; ld h,(ix+9) ; y coordinate.

3902 90BC ;numsp4 add a,16 ; look down 16 pixels.

3903 90BC ; ld l,a ; coords in hl.

3904 90BC ; and 7 ; are we on platform boundary?

3905 90BC ; ret nz ; no, cannot jump.

3906 90BC ; ld (dispx),hl ; set up test coordinates.

3907 90BC ; ld b,a ; copy to b register.

3908 90BC ; call tstbl ; get map address.

3909 90BC ; call plchk ; block, platform check.

3910 90BC ; jr nz,jump0 ; it's solid, we can jump.

3911 90BC ; inc hl ; look right one cell.

3912 90BC ; call plchk ; block, platform check.

3913 90BC ; jr nz,jump0 ; it's solid, we can jump.

3914 90BC ; ld a,b ; y coordinate.

3915 90BC ; and 7 ; position straddling block cells.

3916 90BC ; ret z ; no more checks needed.

3917 90BC ; inc hl ; look to third cell.

3918 90BC ; call plchk ; block, platform check.

3919 90BC ; ret z ; not solid, don't jump.

3920 90BC DD 7E 0D jump0 ld a,(ix+13) ; jumping flag.

3921 90BF A7 and a ; is it set?

3922 90C0 C0 ret nz ; already in the air.

3923 90C1 DD 34 0D inc (ix+13) ; set it.

3924 90C4 DD 71 0E ld (ix+14),c ; set jump height.

3925 90C7 C9 ret

3926 90C8

3927 90C8 DD 7E 0D hop ld a,(ix+13) ; jumping flag.

3928 90CB A7 and a ; is it set?

3929 90CC C0 ret nz ; already in the air.

3930 90CD DD 36 0D FF ld (ix+13),255 ; set it.

3931 90D1 DD 36 0E 00 ld (ix+14),0 ; set jump table displacement.

3932 90D5 C9 ret

3933 90D6

3934 90D6 ; Random numbers code.

3935 90D6 ; Pseudo-random number generator, 8-bit.

3936 90D6

3937 90D6 21 D0 8C random ld hl,seed ; set up seed pointer.

3938 90D9 7E ld a,(hl) ; get last random number.

3939 90DA 47 ld b,a ; copy to b register.

3940 90DB 0F rrca ; multiply by 32.

3941 90DC 0F rrca

3942 90DD 0F rrca

3943 90DE EE 1F xor 31

3944 90E0 80 add a,b

3945 90E1 DE FF sbc a,255

3946 90E3 77 ld (hl),a ; store new seed.

3947 90E4 32 08 82 ld (varrnd),a ; return number in variable.

3948 90E7 C9 ret

3949 90E8

3950 90E8

3951 90E8 ; Keyboard test routine.

3952 90E8

3953 90E8 4F ktest ld c,a ; key to test in c.

3954 90E9 E6 07 and 7 ; mask bits d0-d2 for row.

3955 90EB 3C inc a ; in range 1-8.

3956 90EC 47 ld b,a ; place in b.

3957 90ED CB 39 srl c ; divide c by 8

3958 90EF CB 39 srl c ; to find position within row.

3959 90F1 CB 39 srl c

3960 90F3 3E 05 ld a,5 ; only 5 keys per row.

3961 90F5 91 sub c ; subtract position.

3962 90F6 4F ld c,a ; put in c.

3963 90F7 3E FE ld a,254 ; high byte of port to read.

3964 90F9 0F ktest0 rrca ; rotate into position.

3965 90FA 10 FD djnz ktest0 ; repeat until we've found relevant row.

3966 90FC DB FE in a,(254) ; read port (a=high, 254=low).

3967 90FE 1F ktest1 rra ; rotate bit out of result.

3968 90FF 0D dec c ; loop counter.

3969 9100 C2 FE 90 jp nz,ktest1 ; repeat until bit for position in carry.

3970 9103 C9 ret

3971 9104

3972 9104

3973 9104 ; Joystick and keyboard reading routines.

3974 9104

3975 9104 3A 04 82 joykey ld a,(contrl) ; control flag.

3976 9107 3D dec a ; is it the keyboard?

3977 9108 28 18 jr z,joyjoy ; no, it's Kempston joystick.

3978 910A 3D dec a ; Sinclair?

3979 910B 28 36 jr z,joysin ; read Sinclair joystick.

3980 910D

3981 910D ; Keyboard controls.

3982 910D

3983 910D 21 D1 81 ld hl,keys+6 ; address of last key.

3984 9110 1E 00 ld e,0 ; zero reading.

3985 9112 16 07 ld d,7 ; keys to read.

3986 9114 7E joyke0 ld a,(hl) ; get key from table.

3987 9115 CD E8 90 call ktest ; being pressed?

3988 9118 3F ccf ; complement the carry.

3989 9119 CB 13 rl e ; rotate into reading.

3990 911B 2B dec hl ; next key.

3991 911C 15 dec d ; one less to do.

3992 911D C2 14 91 jp nz,joyke0 ; repeat for all keys.

3993 9120 18 1C jr joyjo1 ; store the value.

3994 9122

3995 9122 ; Kempston joystick controls.

3996 9122

3997 9122 01 1F 00 joyjoy ld bc,31 ; port for Kempston interface.

3998 9125 ED 78 in a,(c) ; read it.

3999 9127 E6 1F and $1f ; filter 5 buttons

4000 9129 5F joyjo3 ld e,a ; copy to e register.

4001 912A 3A D0 81 ld a,(keys+5) ; key six.

4002 912D CD E8 90 call ktest ; being pressed?

4003 9130 38 02 jr c,joyjo0 ; not pressed.

4004 9132 CB EB set 5,e ; set bit d5.

4005 9134 3A D1 81 joyjo0 ld a,(keys+6) ; key seven.

4006 9137 CD E8 90 call ktest ; being pressed?

4007 913A 38 02 jr c,joyjo1 ; not pressed.

4008 913C CB F3 set 6,e ; set bit d6.

4009 913E 7B joyjo1 ld a,e ; copy e register to accumulator.

4010 913F 32 DF 81 joyjo2 ld (joyval),a ; remember value.

4011 9142 C9 ret

4012 9143

4013 9143 ; Sinclair joystick controls.

4014 9143

4015 9143 01 FE EF joysin ld bc,61438 ; port for Sinclair 2.

4016 9146 ED 78 in a,(c) ; read joystick.

4017 9148 57 ld d,a ; clear values.

4018 9149 AF xor a ; clear accumulator.

4019 914A 1E 10 ld e,16 ; Kempston fire bit value.

4020 914C CB 42 bit 0,d ; fire bit pressed?

4021 914E CC 6F 91 call z,joysi0 ; add bit.

4022 9151 1E 01 ld e,1 ; Kempston bit value.

4023 9153 CB 5A bit 3,d ; fire bit pressed?

4024 9155 CC 6F 91 call z,joysi0 ; add bit.

4025 9158 1E 02 ld e,2 ; Kempston bit value.

4026 915A CB 62 bit 4,d ; fire bit pressed?

4027 915C CC 6F 91 call z,joysi0 ; add bit.

4028 915F 1E 08 ld e,8 ; Kempston bit value.

4029 9161 CB 4A bit 1,d ; fire bit pressed?

4030 9163 CC 6F 91 call z,joysi0 ; add bit.

4031 9166 1E 04 ld e,4 ; Kempston bit value.

4032 9168 CB 52 bit 2,d ; fire bit pressed?

4033 916A CC 6F 91 call z,joysi0 ; add bit.

4034 916D 18 BA jr joyjo3 ; read last 2 keys a la Kempston.

4035 916F

4036 916F 83 joysi0 add a,e ; add bit value.

4037 9170 C9 ret

4038 9171

4039 9171 ; Display message.

4040 9171

4041 9171 ;dmsg ld hl,nummsg ; total messages.

4042 9171 ; cp (hl) ; does this one exist?

4043 9171 ; ret nc ; no, nothing to display.

4044 9171 21 45 6D dmsg ld hl,msgdat ; pointer to messages.

4045 9174 CD 5B 92 call getwrd ; get message number.

4046 9177 CD 4A 92 dmsg3 call preprt ; pre-printing stuff.

4047 917A CD E5 8C call checkx ; make sure we're in a printable range.

4048 917D 3A B7 91 ld a,(prtmod) ; print mode.

4049 9180 A7 and a ; standard size?

4050 9181 C2 B8 91 jp nz,bmsg1 ; no, double-height text.

4051 9184 E5 dmsg0 push hl ; store string pointer.

4052 9185 7E ld a,(hl) ; fetch byte to display.

4053 9186 E6 7F and 127 ; remove any end marker.

4054 9188 FE 0D cp 13 ; newline character?

4055 918A 28 1B jr z,dmsg1

4056 918C CD DB 8D call pchar ; display character.

4057 918F CD C3 8D call gaadd ; get attribute address.

4058 9192 3A 8D 5C ld a,(23693) ; current cell colours.

4059 9195 77 ld (hl),a ; write to attribute cell.

4060 9196 CD 38 92 call nexpos ; display position.

4061 9199 20 03 jr nz,dmsg2 ; not on a new line.

4062 919B CD 42 92 call nexlin ; next line down.

4063 919E E1 dmsg2 pop hl

4064 919F 7E ld a,(hl) ; fetch last character.

4065 91A0 17 rla ; was it the end?

4066 91A1 DA 14 8D jp c,dscor2 ; yes, job done.

4067 91A4 23 inc hl ; next character to display.

4068 91A5 18 DD jr dmsg0

4069 91A7 21 10 82 dmsg1 ld hl,dispx ; x coordinate.

4070 91AA 34 inc (hl) ; newline.

4071 91AB 7E ld a,(hl) ; fetch position.

4072 91AC FE 18 cp 24 ; past screen edge?

4073 91AE 38 02 jr c,dmsg4 ; no, it's okay.

4074 91B0 36 00 ld (hl),0 ; restart at top.

4075 91B2 23 dmsg4 inc hl ; y coordinate.

4076 91B3 36 00 ld (hl),0 ; carriage return.

4077 91B5 18 E7 jr dmsg2

4078 91B7 00 prtmod defb 0 ; print mode, 0 = standard, 1 = double-height.

4079 91B8

4080 91B8 ; Display message in big text.

4081 91B8

4082 91B8 7E bmsg1 ld a,(hl) ; get character to display.

4083 91B9 E5 push hl ; store pointer to message.

4084 91BA E6 7F and 127 ; only want 7 bits.

4085 91BC FE 0D cp 13 ; newline character?

4086 91BE 28 0A jr z,bmsg2

4087 91C0 CD DB 91 call bchar ; display big char.

4088 91C3 E1 bmsg3 pop hl ; retrieve message pointer.

4089 91C4 7E ld a,(hl) ; look at last character.

4090 91C5 23 inc hl ; next character in list.

4091 91C6 17 rla ; was terminator flag set?

4092 91C7 30 EF jr nc,bmsg1 ; no, keep going.

4093 91C9 C9 ret

4094 91CA 21 05 82 bmsg2 ld hl,charx ; x coordinate.

4095 91CD 34 inc (hl) ; newline.

4096 91CE 34 inc (hl) ; newline.

4097 91CF 7E ld a,(hl) ; fetch position.

4098 91D0 FE 17 cp 23 ; past screen edge?

4099 91D2 38 EF jr c,bmsg3 ; no, it's okay.

4100 91D4 36 00 ld (hl),0 ; restart at top.

4101 91D6 23 inc hl ; y coordinate.

4102 91D7 36 00 ld (hl),0 ; carriage return.

4103 91D9 18 E8 jr bmsg3

4104 91DB

4105 91DB

4106 91DB ; Big character display.

4107 91DB

4108 91DB 07 bchar rlca ; multiply char by 8.

4109 91DC 07 rlca

4110 91DD 07 rlca

4111 91DE 5F ld e,a ; store shift in e.

4112 91DF E6 07 and 7 ; only want high byte bits.

4113 91E1 57 ld d,a ; store in d.

4114 91E2 7B ld a,e ; restore shifted value.

4115 91E3 E6 F8 and 248 ; only want low byte bits.

4116 91E5 5F ld e,a ; that's the low byte.

4117 91E6 2A 36 5C ld hl,(23606) ; address of font.

4118 91E9 19 add hl,de ; add displacement.

4119 91EA CD 95 8D call gprad ; get screen address.

4120 91ED EB ex de,hl ; font in de, screen address in hl.

4121 91EE 06 08 ld b,8 ; height of character in font.

4122 91F0 1A bchar0 ld a,(de) ; get a bit of the font.

4123 91F1 13 inc de ; next line of font.

4124 91F2 77 ld (hl),a ; write to screen.

4125 91F3 24 inc h ; down a line.

4126 91F4 77 ld (hl),a ; write to screen.

4127 91F5 CD 71 94 call nline ; next line down.

4128 91F8 10 F6 djnz bchar0 ; repeat.

4129 91FA CD C3 8D call gaadd ; get attribute address.

4130 91FD 3A 8D 5C ld a,(23693) ; current colour.

4131 9200 77 ld (hl),a ; set attribute.

4132 9201 4F ld c,a ; copy colour to c.

4133 9202 11 20 00 ld de,32 ; distance to next line.

4134 9205 19 add hl,de ; point to second cell.

4135 9206 7C ld a,h ; high byte of address.

4136 9207 FE 5B cp 91 ; past edge of screen?

4137 9209 30 01 jr nc,bchar1 ; yes, don't write to printer buffer/sysvars.

4138 920B 71 ld (hl),c ; set second cell's attributes.

4139 920C CD 38 92 bchar1 call nexpos ; display position.

4140 920F C2 16 92 jp nz,bchar2 ; not on a new line.

4141 9212 34 bchar3 inc (hl) ; newline.

4142 9213 CD 42 92 call nexlin ; next line check.

4143 9216 C3 14 8D bchar2 jp dscor2 ; tidy up line and column variables.

4144 9219

4145 9219 ; Display a character.

4146 9219

4147 9219 47 achar ld b,a ; copy to b.

4148 921A CD 4A 92 call preprt ; get ready to print.

4149 921D 3A B7 91 ld a,(prtmod) ; print mode.

4150 9220 A7 and a ; standard size?

4151 9221 78 ld a,b ; character in accumulator.

4152 9222 C2 DB 91 jp nz,bchar ; no, double-height text.

4153 9225 CD DB 8D call pchar ; display character.

4154 9228 CD C3 8D call gaadd ; get attribute address.

4155 922B 3A 8D 5C ld a,(23693) ; current cell colours.

4156 922E 77 ld (hl),a ; write to attribute cell.

4157 922F CD 38 92 call nexpos ; display position.

4158 9232 CA 12 92 jp z,bchar3 ; next line down.

4159 9235 C3 16 92 jp bchar2 ; tidy up.

4160 9238

4161 9238 ; Get next print column position.

4162 9238

4163 9238 21 11 82 nexpos ld hl,dispy ; display position.

4164 923B 7E ld a,(hl) ; get coordinate.

4165 923C 3C inc a ; move along one position.

4166 923D E6 1F and 31 ; reached edge of screen?

4167 923F 77 ld (hl),a ; set new position.

4168 9240 2B dec hl ; point to x now.

4169 9241 C9 ret ; return with status in zero flag.

4170 9242

4171 9242 ; Get next print line position.

4172 9242

4173 9242 34 nexlin inc (hl) ; newline.

4174 9243 7E ld a,(hl) ; vertical position.

4175 9244 FE 18 cp 24 ; past screen edge?

4176 9246 D8 ret c ; no, still okay.

4177 9247 36 00 ld (hl),0 ; restart at top.

4178 9249 C9 ret

4179 924A

4180 924A ; Pre-print preliminaries.

4181 924A

4182 924A ED 5B 36 5C preprt ld de,(23606) ; font pointer.

4183 924E ED 53 E3 8C ld (grbase),de ; set up graphics base.

4184 9252 ED 5B 05 82 prescr ld de,(charx) ; display coordinates.

4185 9256 ED 53 10 82 ld (dispx),de ; set up general coordinates.

4186 925A C9 ret

4187 925B

4188 925B ; On entry: hl points to word list

4189 925B ; a contains word number.

4190 925B

4191 925B A7 getwrd and a ; first word in list?

4192 925C C8 ret z ; yep, don't search.

4193 925D 47 ld b,a

4194 925E 7E getwd0 ld a,(hl)

4195 925F 23 inc hl

4196 9260 FE 80 cp 128 ; found end?

4197 9262 38 FA jr c,getwd0 ; no, carry on.

4198 9264 10 F8 djnz getwd0 ; until we have right number.

4199 9266 C9 ret

4200 9267

4201 9267

4202 9267 ; Bubble sort.

4203 9267

4204 9267 06 0B bsort ld b,NUMSPR - 1 ; sprites to swap.

4205 9269 DD 21 1C 98 ld ix,sprtab ; sprite table.

4206 926D C5 bsort0 push bc ; store loop counter for now.

4207 926E

4208 926E DD 7E 00 ld a,(ix+0) ; first sprite type.

4209 9271 3C inc a ; is it switched off?

4210 9272 28 22 jr z,swemp ; yes, may need to switch another in here.

4211 9274

4212 9274 DD 7E 11 ld a,(ix+TABSIZ) ; check next slot exists.

4213 9277 3C inc a ; is it enabled?

4214 9278 28 08 jr z,bsort2 ; no, nothing to swap.

4215 927A

4216 927A DD 7E 14 ld a,(ix+(3+TABSIZ)); fetch next sprite's coordinate.

4217 927D DD BE 03 cp (ix+3) ; compare with this x coordinate.

4218 9280 38 09 jr c,bsort1 ; next sprite is higher - may need to switch.

4219 9282 11 11 00 bsort2 ld de,TABSIZ ; distance to next odd/even entry.

4220 9285 DD 19 add ix,de ; next sprite.

4221 9287 C1 pop bc ; retrieve loop counter.

4222 9288 10 E3 djnz bsort0 ; repeat for remaining sprites.

4223 928A C9 ret

4224 928B

4225 928B DD 7E 11 bsort1 ld a,(ix+TABSIZ) ; sprite on/off flag.

4226 928E 3C inc a ; is it enabled?

4227 928F 28 F1 jr z,bsort2 ; no, nothing to swap.

4228 9291 CD A1 92 call swspr ; swap positions.

4229 9294 18 EC jr bsort2

4230 9296

4231 9296 DD 7E 11 swemp ld a,(ix+TABSIZ) ; next table entry.

4232 9299 3C inc a ; is that one on?

4233 929A 28 E6 jr z,bsort2 ; no, nothing to swap.

4234 929C CD A1 92 call swspr ; swap positions.

4235 929F 18 E1 jr bsort2

4236 92A1

4237 92A1 ; Swap sprites.

4238 92A1

4239 92A1 DD E5 swspr push ix ; table address on stack.

4240 92A3 E1 pop hl ; pop into hl pair.

4241 92A4 54 ld d,h ; copy to de pair.

4242 92A5 5D ld e,l

4243 92A6 01 11 00 ld bc,TABSIZ ; distance to second entry.

4244 92A9 09 add hl,bc ; point to second sprite entry.

4245 92AA 06 11 ld b,TABSIZ ; bytes to swap.

4246 92AC 4E swspr0 ld c,(hl) ; fetch second byte.

4247 92AD 1A ld a,(de) ; fetch first byte.

4248 92AE 77 ld (hl),a ; copy to second.

4249 92AF 79 ld a,c ; second byte in accumulator.

4250 92B0 12 ld (de),a ; copy to first sprite entry.

4251 92B1 13 inc de ; next byte.

4252 92B2 23 inc hl ; next byte.

4253 92B3 10 F7 djnz swspr0 ; swap all bytes in table entry.

4254 92B5 C9 ret

4255 92B6

4256 92B6

4257 92B6 ; Process sprites.

4258 92B6

4259 92B6 06 0C pspr ld b,NUMSPR ; sprites to process.

4260 92B8 DD 21 1C 98 ld ix,sprtab ; sprite table.

4261 92BC C5 pspr1 push bc ; store loop counter for now.

4262 92BD DD 7E 00 ld a,(ix+0) ; fetch sprite type.

4263 92C0 FE 09 cp 9 ; within range of sprite types?

4264 92C2 DC CE 92 call c,pspr2 ; yes, process this one.

4265 92C5 11 11 00 ld de,TABSIZ ; distance to next odd/even entry.

4266 92C8 DD 19 add ix,de ; next sprite.

4267 92CA C1 pop bc ; retrieve loop counter.

4268 92CB 10 EF djnz pspr1 ; repeat for remaining sprites.

4269 92CD C9 ret

4270 92CE DD 22 E7 92 pspr2 ld (ogptr),ix ; store original sprite pointer.

4271 92D2 CD DA 92 call pspr3 ; do the routine.

4272 92D5 DD 2A E7 92 rtorg ld ix,(ogptr) ; restore original pointer to sprite.

4273 92D9 C9 rtorg0 ret

4274 92DA 21 E9 92 pspr3 ld hl,evtyp0 ; sprite type events list.

4275 92DD 87 pspr4 add a,a ; double accumulator.

4276 92DE 5F ld e,a ; copy to de.

4277 92DF 16 00 ld d,0 ; no high byte.

4278 92E1 19 add hl,de ; point to address of routine.

4279 92E2 5E ld e,(hl) ; address low.

4280 92E3 23 inc hl ; next byte of address.

4281 92E4 56 ld d,(hl) ; address high.

4282 92E5 EB ex de,hl ; swap address into hl.

4283 92E6 E9 jp (hl) ; go there.

4284 92E7 00 00 ogptr defw 0 ; original sprite pointer.

4285 92E9

4286 92E9 ; Address of each sprite type's routine.

4287 92E9

4288 92E9 0D 61 evtyp0 defw evnt00

4289 92EB EA 61 evtyp1 defw evnt01

4290 92ED 50 62 evtyp2 defw evnt02

4291 92EF 51 62 evtyp3 defw evnt03

4292 92F1 5D 63 evtyp4 defw evnt04

4293 92F3 5E 63 evtyp5 defw evnt05

4294 92F5 60 68 evtyp6 defw evnt06

4295 92F7 CE 68 evtyp7 defw evnt07

4296 92F9 79 6A evtyp8 defw evnt08

4297 92FB

4298 92FB

4299 92FB ; Display sprites.

4300 92FB

4301 92FB 06 06 dspr ld b,NUMSPR/2 ; number of sprites to display.

4302 92FD C5 dspr0 push bc ; store loop counter for now.

4303 92FE DD 7E 00 ld a,(ix+0) ; get sprite type.

4304 9301 3C inc a ; is it enabled?

4305 9302 20 21 jr nz,dspr1 ; yes, it needs deleting.

4306 9304 DD 7E 05 dspr5 ld a,(ix+5) ; new type.

4307 9307 3C inc a ; is it enabled?

4308 9308 20 4E jr nz,dspr3 ; yes, it needs drawing.

4309 930A

4310 930A DD E5 dspr2 push ix ; put ix on stack.

4311 930C E1 pop hl ; pop into hl.

4312 930D 5D ld e,l ; copy to de.

4313 930E 54 ld d,h

4314 930F

4315 930F ;dspr2 ld e,ixl ; copy ix to de.

4316 930F ; ld d,ixh

4317 930F ; ld l,e ; copy to hl.

4318 930F ; ld h,d

4319 930F 01 05 00 ld bc,5 ; distance to new type.

4320 9312 09 add hl,bc ; point to new properties.

4321 9313 ED A0 ldi ; copy to old positions.

4322 9315 ED A0 ldi

4323 9317 ED A0 ldi

4324 9319 ED A0 ldi

4325 931B ED A0 ldi

4326 931D 0E 22 ld c,TABSIZ\*2 ; distance to next odd/even entry.

4327 931F DD 09 add ix,bc ; next sprite.

4328 9321 C1 pop bc ; retrieve loop counter.

4329 9322 10 D9 djnz dspr0 ; repeat for remaining sprites.

4330 9324 C9 ret

4331 9325 ;dspr1 ld a,(ix+3) ; old x coord.

4332 9325 ; cp 177 ; beyond maximum?

4333 9325 ; jr nc,dspr5 ; yes, don't delete it.

4334 9325 DD 7E 05 dspr1 ld a,(ix+5) ; type of new sprite.

4335 9328 3C inc a ; is this enabled?

4336 9329 20 06 jr nz,dspr4 ; yes, display both.

4337 932B CD C8 93 dspr6 call sspria ; show single sprite.

4338 932E C3 0A 93 jp dspr2

4339 9331

4340 9331 ; Displaying two sprites. Don't bother redrawing if nothing has changed.

4341 9331

4342 9331 DD 7E 04 dspr4 ld a,(ix+4) ; old y.

4343 9334 DD BE 09 cp (ix+9) ; compare with new value.

4344 9337 20 19 jr nz,dspr7 ; they differ, need to redraw.

4345 9339 DD 7E 03 ld a,(ix+3) ; old x.

4346 933C DD BE 08 cp (ix+8) ; compare against new value.

4347 933F 20 11 jr nz,dspr7 ; they differ, need to redraw.

4348 9341 DD 7E 02 ld a,(ix+2) ; old frame.

4349 9344 DD BE 07 cp (ix+7) ; compare against new value.

4350 9347 20 09 jr nz,dspr7 ; they differ, need to redraw.

4351 9349 DD 7E 01 ld a,(ix+1) ; old image.

4352 934C DD BE 06 cp (ix+6) ; compare against new value.

4353 934F CA 0A 93 jp z,dspr2 ; everything is the same, don't redraw.

4354 9352 CD DD 93 dspr7 call sspric ; delete old sprite, draw new one simultaneously.

4355 9355 C3 0A 93 jp dspr2

4356 9358 CD D7 93 dspr3 call ssprib ; show single sprite.

4357 935B C3 0A 93 jp dspr2

4358 935E

4359 935E

4360 935E ; Get sprite address calculations.

4361 935E ; gspran = new sprite, gsprad = old sprite.

4362 935E

4363 935E DD 6E 08 gspran ld l,(ix+8) ; new x coordinate.

4364 9361 DD 66 09 ld h,(ix+9) ; new y coordinate.

4365 9364 22 10 82 ld (dispx),hl ; set display coordinates.

4366 9367 DD 7E 06 ld a,(ix+6) ; new sprite image.

4367 936A CD 71 96 call gfrm ; fetch start frame for this sprite.

4368 936D 7E ld a,(hl) ; frame in accumulator.

4369 936E DD 86 07 add a,(ix+7) ; new add frame number.

4370 9371 C3 87 93 jp gspra0

4371 9374

4372 9374 DD 6E 03 gsprad ld l,(ix+3) ; x coordinate.

4373 9377 DD 66 04 ld h,(ix+4) ; y coordinate.

4374 937A 22 10 82 ld (dispx),hl ; set display coordinates.

4375 937D DD 7E 01 ld a,(ix+1) ; sprite image.

4376 9380 CD 71 96 call gfrm ; fetch start frame for this sprite.

4377 9383 7E ld a,(hl) ; frame in accumulator.

4378 9384 DD 86 02 add a,(ix+2) ; add frame number.

4379 9387

4380 9387 0F gspra0 rrca ; multiply by 128.

4381 9388 57 ld d,a ; store in d.

4382 9389 E6 80 and 128 ; low byte bit.

4383 938B 5F ld e,a ; got low byte.

4384 938C 7A ld a,d ; restore result.

4385 938D E6 7F and 127 ; high byte bits.

4386 938F 57 ld d,a ; displacement high byte.

4387 9390 21 4A 6E ld hl,sprgfx ; address of play sprites.

4388 9393 19 add hl,de ; point to frame.

4389 9394

4390 9394 3A 11 82 ld a,(dispy) ; y coordinate.

4391 9397 E6 06 and 6 ; position within byte boundary.

4392 9399 4F ld c,a ; low byte of table displacement.

4393 939A 07 rlca ; multiply by 32.

4394 939B 07 rlca ; already a multiple

4395 939C 07 rlca ; of 2, so just 4

4396 939D 07 rlca ; shifts needed.

4397 939E 5F ld e,a ; put displacement in low byte of de.

4398 939F 16 00 ld d,0 ; zero the high byte.

4399 93A1 42 ld b,d ; no high byte for mask displacement either.

4400 93A2 19 add hl,de ; add to sprite address.

4401 93A3 EB ex de,hl ; need it in de for now.

4402 93A4 21 C0 93 ld hl,spmask ; pointer to mask table.

4403 93A7 09 add hl,bc ; add displacement to pointer.

4404 93A8 4E ld c,(hl) ; left mask.

4405 93A9 23 inc hl

4406 93AA 46 ld b,(hl) ; right mask.

4407 93AB

4408 93AB ; Drop into screen address routine.

4409 93AB ; This routine returns a screen address for (dispx, dispy) in hl.

4410 93AB

4411 93AB 3A 10 82 scadd ld a,(dispx) ; coordinate.

4412 93AE 6F ld l,a ; low byte of table.

4413 93AF 26 FB ld h,251 ; high byte of 64256 (SCADTB).

4414 93B1 7E ld a,(hl) ; fetch high byte.

4415 93B2 24 inc h ; point to low byte table.

4416 93B3 6E ld l,(hl) ; fetch low byte.

4417 93B4 67 ld h,a ; hl points to start of line.

4418 93B5 3A 11 82 ld a,(dispy) ; y pixel coordinate.

4419 93B8 0F rrca ; divide by 8.

4420 93B9 0F rrca

4421 93BA 0F rrca

4422 93BB E6 1F and 31 ; squares 0 - 31 across screen.

4423 93BD 85 add a,l ; add to address.

4424 93BE 6F ld l,a ; copy to hl = address of screen.

4425 93BF C9 ret

4426 93C0

4427 93C0 spmask defb 255,0,63,192,15,240,3,252

4427 93C0 FF003FC00FF003FC

4428 93C8

4429 93C8

4430 93C8 ; These are the sprite routines.

4431 93C8 ; sspria = single sprite, old (ix).

4432 93C8 ; ssprib = single sprite, new (ix+5).

4433 93C8 ; sspric = both sprites, old (ix) and new (ix+5).

4434 93C8

4435 93C8 CD 74 93 sspria call gsprad ; get old sprite address.

4436 93CB 3E 10 sspri2 ld a,16 ; vertical lines.

4437 93CD 08 sspri0 ex af,af' ; store line counter away in alternate registers.

4438 93CE CD 60 94 call dline ; draw a line.

4439 93D1 08 ex af,af' ; restore line counter.

4440 93D2 3D dec a ; one less to go.

4441 93D3 C2 CD 93 jp nz,sspri0

4442 93D6 C9 ret

4443 93D7

4444 93D7 CD 5E 93 ssprib call gspran ; get new sprite address.

4445 93DA C3 CB 93 jp sspri2

4446 93DD

4447 93DD CD 74 93 sspric call gsprad ; get old sprite address.

4448 93E0 D9 exx ; store addresses.

4449 93E1 CD 5E 93 call gspran ; get new sprite addresses.

4450 93E4 CD 60 94 call dline ; draw a line.

4451 93E7 D9 exx ; restore old addresses.

4452 93E8 CD 60 94 call dline ; delete a line.

4453 93EB D9 exx ; flip to new sprite addresses.

4454 93EC CD 60 94 call dline ; draw a line.

4455 93EF D9 exx ; restore old addresses.

4456 93F0 CD 60 94 call dline ; delete a line.

4457 93F3 D9 exx ; flip to new sprite addresses.

4458 93F4 CD 60 94 call dline ; draw a line.

4459 93F7 D9 exx ; restore old addresses.

4460 93F8 CD 60 94 call dline ; delete a line.

4461 93FB D9 exx ; flip to new sprite addresses.

4462 93FC CD 60 94 call dline ; draw a line.

4463 93FF D9 exx ; restore old addresses.

4464 9400 CD 60 94 call dline ; delete a line.

4465 9403 D9 exx ; flip to new sprite addresses.

4466 9404 CD 60 94 call dline ; draw a line.

4467 9407 D9 exx ; restore old addresses.

4468 9408 CD 60 94 call dline ; delete a line.

4469 940B D9 exx ; flip to new sprite addresses.

4470 940C CD 60 94 call dline ; draw a line.

4471 940F D9 exx ; restore old addresses.

4472 9410 CD 60 94 call dline ; delete a line.

4473 9413 D9 exx ; flip to new sprite addresses.

4474 9414 CD 60 94 call dline ; draw a line.

4475 9417 D9 exx ; restore old addresses.

4476 9418 CD 60 94 call dline ; delete a line.

4477 941B D9 exx ; flip to new sprite addresses.

4478 941C CD 60 94 call dline ; draw a line.

4479 941F D9 exx ; restore old addresses.

4480 9420 CD 60 94 call dline ; delete a line.

4481 9423 D9 exx ; flip to new sprite addresses.

4482 9424 CD 60 94 call dline ; draw a line.

4483 9427 D9 exx ; restore old addresses.

4484 9428 CD 60 94 call dline ; delete a line.

4485 942B D9 exx ; flip to new sprite addresses.

4486 942C CD 60 94 call dline ; draw a line.

4487 942F D9 exx ; restore old addresses.

4488 9430 CD 60 94 call dline ; delete a line.

4489 9433 D9 exx ; flip to new sprite addresses.

4490 9434 CD 60 94 call dline ; draw a line.

4491 9437 D9 exx ; restore old addresses.

4492 9438 CD 60 94 call dline ; delete a line.

4493 943B D9 exx ; flip to new sprite addresses.

4494 943C CD 60 94 call dline ; draw a line.

4495 943F D9 exx ; restore old addresses.

4496 9440 CD 60 94 call dline ; delete a line.

4497 9443 D9 exx ; flip to new sprite addresses.

4498 9444 CD 60 94 call dline ; draw a line.

4499 9447 D9 exx ; restore old addresses.

4500 9448 CD 60 94 call dline ; delete a line.

4501 944B D9 exx ; flip to new sprite addresses.

4502 944C CD 60 94 call dline ; draw a line.

4503 944F D9 exx ; restore old addresses.

4504 9450 CD 60 94 call dline ; delete a line.

4505 9453 D9 exx ; flip to new sprite addresses.

4506 9454 CD 60 94 call dline ; draw a line.

4507 9457 D9 exx ; restore old addresses.

4508 9458 CD 60 94 call dline ; delete a line.

4509 945B D9 exx ; flip to new sprite addresses.

4510 945C CD 60 94 call dline ; draw a line.

4511 945F D9 exx ; restore old addresses.

4512 9460

4513 9460 ; Drop through.

4514 9460 ; Line drawn, now work out next target address.

4515 9460

4516 9460 1A dline ld a,(de) ; graphic data.

4517 9461 A1 and c ; mask away what's not needed.

4518 9462 AE xor (hl) ; XOR with what's there.

4519 9463 77 ld (hl),a ; bung it in.

4520 9464 2C inc l ; next screen address.

4521 9465 2C inc l ; next screen address.

4522 9466 1A ld a,(de) ; fetch data.

4523 9467 A0 and b ; mask away unwanted bits.

4524 9468 AE xor (hl) ; XOR with what's there.

4525 9469 77 ld (hl),a ; bung it in.

4526 946A 13 inc de ; next graphic.

4527 946B 2D dec l ; one character cell to the left.

4528 946C 1A ld a,(de) ; second bit of data.

4529 946D AE xor (hl) ; XOR with what's there.

4530 946E 77 ld (hl),a ; bung it in.

4531 946F 13 inc de ; point to next line of data.

4532 9470 2D dec l ; another char left.

4533 9471

4534 9471 ; Line drawn, now work out next target address.

4535 9471

4536 9471 24 nline inc h ; increment pixel.

4537 9472 7C ld a,h ; get pixel address.

4538 9473 E6 07 and 7 ; straddling character position?

4539 9475 C0 ret nz ; no, we're on next line already.

4540 9476 7C ld a,h ; get pixel address.

4541 9477 D6 08 sub 8 ; subtract 8 for start of segment.

4542 9479 67 ld h,a ; new high byte of address.

4543 947A 7D ld a,l ; get low byte of address.

4544 947B C6 20 add a,32 ; one line down.

4545 947D 6F ld l,a ; new low byte.

4546 947E D0 ret nc ; not reached next segment yet.

4547 947F 7C ld a,h ; address high.

4548 9480 C6 08 add a,8 ; add 8 to next segment.

4549 9482 67 ld h,a ; new high byte.

4550 9483 FE 58 cp 88 ; reached end of screen?

4551 9485 D8 ret c ; not yet.

4552 9486 26 38 ld h,56 ; back to ROM.

4553 9488 C9 ret

4554 9489

4555 9489

4556 9489 ; Animates a sprite.

4557 9489

4558 9489 21 E0 81 animsp ld hl,frmno ; game frame.

4559 948C A6 and (hl) ; is it time to change the frame?

4560 948D C0 ret nz ; not this frame.

4561 948E DD 7E 06 ld a,(ix+6) ; sprite image.

4562 9491 CD 71 96 call gfrm ; get frame data.

4563 9494 23 inc hl ; point to frames.

4564 9495 DD 7E 07 ld a,(ix+7) ; sprite frame.

4565 9498 3C inc a ; next one along.

4566 9499 BE cp (hl) ; reached the last frame?

4567 949A 38 01 jr c,anims0 ; no, not yet.

4568 949C AF xor a ; start at first frame.

4569 949D DD 77 07 anims0 ld (ix+7),a ; new frame.

4570 94A0 C9 ret

4571 94A1 21 E0 81 animbk ld hl,frmno ; game frame.

4572 94A4 A6 and (hl) ; is it time to change the frame?

4573 94A5 C0 ret nz ; not this frame.

4574 94A6 DD 7E 06 ld a,(ix+6) ; sprite image.

4575 94A9 CD 71 96 call gfrm ; get frame data.

4576 94AC 23 inc hl ; point to frames.

4577 94AD DD 7E 07 ld a,(ix+7) ; sprite frame.

4578 94B0 A7 and a ; first one?

4579 94B1 20 01 jr nz,rtanb0 ; yes, start at end.

4580 94B3 7E ld a,(hl) ; last sprite.

4581 94B4 3D rtanb0 dec a ; next one along.

4582 94B5 18 E6 jr anims0 ; set new frame.

4583 94B7

4584 94B7 ; Check for collision with other sprite, strict enforcement.

4585 94B7

4586 94B7 21 1C 98 sktyp ld hl,sprtab ; sprite table.

4587 94BA 3E 0C numsp2 ld a,NUMSPR ; number of sprites.

4588 94BC 08 sktyp0 ex af,af' ; store loop counter.

4589 94BD 22 D8 94 ld (skptr),hl ; store pointer to sprite.

4590 94C0 7E ld a,(hl) ; get sprite type.

4591 94C1 B8 cp b ; is it the type we seek?

4592 94C2 28 16 jr z,coltyp ; yes, we can use this one.

4593 94C4 2A D8 94 sktyp1 ld hl,(skptr) ; retrieve sprite pointer.

4594 94C7 11 11 00 ld de,TABSIZ ; size of each entry.

4595 94CA 19 add hl,de ; point to next sprite in table.

4596 94CB 08 ex af,af' ; restore loop counter.

4597 94CC 3D dec a ; one less iteration.

4598 94CD C2 BC 94 jp nz,sktyp0 ; keep going until we find a slot.

4599 94D0 21 00 00 ld hl,0 ; default to ROM address - no sprite.

4600 94D3 22 D8 94 ld (skptr),hl ; store pointer to sprite.

4601 94D6 B4 or h ; don't return with zero flag set.

4602 94D7 C9 ret ; didn't find one.

4603 94D8 00 00 skptr defw 0 ; search pointer.

4604 94DA

4605 94DA DD 7E 00 coltyp ld a,(ix+0) ; current sprite type.

4606 94DD B8 cp b ; seeking sprite of same type?

4607 94DE 28 27 jr z,colty1 ; yes, need to check we're not detecting ourselves.

4608 94E0 11 08 00 colty0 ld de,X ; distance to x position in table.

4609 94E3 19 add hl,de ; point to coords.

4610 94E4 5E ld e,(hl) ; fetch x coordinate.

4611 94E5 23 inc hl ; now point to y.

4612 94E6 56 ld d,(hl) ; that's y coordinate.

4613 94E7

4614 94E7 ; Drop into collision detection.

4615 94E7

4616 94E7 DD 7E 08 colc16 ld a,(ix+X) ; x coord.

4617 94EA 93 sub e ; subtract x.

4618 94EB 30 02 jr nc,colc1a ; result is positive.

4619 94ED ED 44 neg ; make negative positive.

4620 94EF FE 10 colc1a cp 16 ; within x range?

4621 94F1 30 D1 jr nc,sktyp1 ; no - they've missed.

4622 94F3 4F ld c,a ; store difference.

4623 94F4 DD 7E 09 ld a,(ix+Y) ; y coord.

4624 94F7 92 sub d ; subtract y.

4625 94F8 30 02 jr nc,colc1b ; result is positive.

4626 94FA ED 44 neg ; make negative positive.

4627 94FC FE 10 colc1b cp 16 ; within y range?

4628 94FE 30 C4 jr nc,sktyp1 ; no - they've missed.

4629 9500 81 add a,c ; add x difference.

4630 9501 FE 1A cp 26 ; only 5 corner pixels touching?

4631 9503 D8 ret c ; carry set if there's a collision.

4632 9504 C3 C4 94 jp sktyp1 ; try next sprite in table.

4633 9507

4634 9507 DD E5 colty1 push ix ; base sprite address onto stack.

4635 9509 D1 pop de ; pop it into de.

4636 950A EB ex de,hl ; flip hl into de.

4637 950B ED 52 sbc hl,de ; compare the two.

4638 950D EB ex de,hl ; restore hl.

4639 950E 28 B4 jr z,sktyp1 ; addresses are identical.

4640 9510 C3 E0 94 jp colty0

4641 9513

4642 9513 ; Display number.

4643 9513

4644 9513 01 24 95 disply ld bc,displ0 ; display workspace.

4645 9516 CD BE 89 call num2ch ; convert accumulator to string.

4646 9519 0B displ1 dec bc ; back one character.

4647 951A 0A ld a,(bc) ; fetch digit.

4648 951B F6 80 or 128 ; insert end marker.

4649 951D 02 ld (bc),a ; new value.

4650 951E 21 24 95 ld hl,displ0 ; display space.

4651 9521 C3 77 91 jp dmsg3 ; display the string.

4652 9524 00 00 00 8D displ0 defb 0,0,0,13+128

4653 9528

4654 9528

4655 9528 ; Initialise screen.

4656 9528

4657 9528 3A F9 98 initsc ld a,(roomtb) ; whereabouts in the map are we?

4658 952B CD 35 95 call tstsc ; find displacement.

4659 952E FE FF cp 255 ; is it valid?

4660 9530 C8 ret z ; no, it's rubbish.

4661 9531 32 EA 81 ld (scno),a ; store new room number.

4662 9534 C9 ret

4663 9535

4664 9535 ; Test screen.

4665 9535

4666 9535 21 03 61 tstsc ld hl,mapdat-MAPWID ; start of map data, subtract width for negative.

4667 9538 47 ld b,a ; store room in b for now.

4668 9539 C6 03 add a,MAPWID ; add width in case we're negative.

4669 953B 5F ld e,a ; screen into e.

4670 953C 16 00 ld d,0 ; zeroise d.

4671 953E 19 add hl,de ; add displacement to map data.

4672 953F 7E ld a,(hl) ; find room number there.

4673 9540 C9 ret

4674 9541

4675 9541 ; Screen left.

4676 9541

4677 9541 3A F9 98 scrl ld a,(roomtb) ; present room table pointer.

4678 9544 3D dec a ; room left.

4679 9545 CD 35 95 scrl0 call tstsc ; test screen.

4680 9548 3C inc a ; is there a screen this way?

4681 9549 C8 ret z ; no, return to loop.

4682 954A 78 ld a,b ; restore room displacement.

4683 954B 32 F9 98 ld (roomtb),a ; new room table position.

4684 954E CD 28 95 scrl1 call initsc ; set new screen.

4685 9551 21 0D 82 ld hl,restfl ; restart screen flag.

4686 9554 36 02 ld (hl),2 ; set it.

4687 9556 C9 ret

4688 9557 3A F9 98 scrr ld a,(roomtb) ; room table pointer.

4689 955A 3C inc a ; room right.

4690 955B 18 E8 jr scrl0

4691 955D 3A F9 98 scru ld a,(roomtb) ; room table pointer.

4692 9560 D6 03 sub MAPWID ; room up.

4693 9562 18 E1 jr scrl0

4694 9564 3A F9 98 scrd ld a,(roomtb) ; room table pointer.

4695 9567 C6 03 add a,MAPWID ; room down.

4696 9569 18 DA jr scrl0

4697 956B

4698 956B ; Jump to new screen.

4699 956B

4700 956B 21 06 61 nwscr ld hl,mapdat ; start of map data.

4701 956E 01 00 50 ld bc,256\*80 ; zero room count, 80 to search.

4702 9571 BE nwscr0 cp (hl) ; have we found a match for screen?

4703 9572 28 05 jr z,nwscr1 ; yes, set new point in map.

4704 9574 23 inc hl ; next room.

4705 9575 0C inc c ; count rooms.

4706 9576 10 F9 djnz nwscr0 ; keep looking.

4707 9578 C9 ret

4708 9579 79 nwscr1 ld a,c ; room displacement.

4709 957A 32 F9 98 ld (roomtb),a ; set the map position.

4710 957D 18 CF jr scrl1 ; draw new room.

4711 957F

4712 957F

4713 957F ; Gravity processing.

4714 957F

4715 957F DD 7E 0D grav ld a,(ix+13) ; in-air flag.

4716 9582 A7 and a ; are we in the air?

4717 9583 C8 ret z ; no we are not.

4718 9584 3C inc a ; increment it.

4719 9585 CA CF 95 jp z,ogrv ; set to 255, use old gravity.

4720 9588 DD 77 0D ld (ix+13),a ; write new setting.

4721 958B 1F rra ; every other frame.

4722 958C 30 0A jr nc,grav0 ; don't apply gravity this time.

4723 958E DD 7E 0E ld a,(ix+14) ; pixels to move.

4724 9591 FE 10 cp 16 ; reached maximum?

4725 9593 28 03 jr z,grav0 ; yes, continue.

4726 9595 DD 34 0E inc (ix+14) ; slow down ascent/speed up fall.

4727 9598 DD 7E 0E grav0 ld a,(ix+14) ; get distance to move.

4728 959B CB 2F sra a ; divide by 2.

4729 959D A7 and a ; any movement required?

4730 959E C8 ret z ; no, not this time.

4731 959F FE 80 cp 128 ; is it up or down?

4732 95A1 30 0C jr nc,gravu ; it's up.

4733 95A3 47 gravd ld b,a ; set pixels to move.

4734 95A4 CD 70 8F gravd0 call cangd ; can we go down?

4735 95A7 20 15 jr nz,gravst ; can't move down, so stop.

4736 95A9 DD 34 08 inc (ix+8) ; adjust new x coord.

4737 95AC 10 F6 djnz gravd0

4738 95AE C9 ret

4739 95AF ED 44 gravu neg ; flip the sign so it's positive.

4740 95B1 47 ld b,a ; set pixels to move.

4741 95B2 CD 4D 8F gravu0 call cangu ; can we go up?

4742 95B5 C2 56 96 jp nz,ifalls ; can't move up, go down next.

4743 95B8 DD 35 08 dec (ix+8) ; adjust new x coord.

4744 95BB 10 F5 djnz gravu0

4745 95BD C9 ret

4746 95BE DD 7E 0E gravst ld a,(ix+14) ; jump pointer high.

4747 95C1 DD 36 0D 00 ld (ix+13),0 ; reset falling flag.

4748 95C5 DD 36 0E 00 ld (ix+14),0 ; store new speed.

4749 95C9 FE 08 cp 8 ; was speed the maximum?

4750 95CB CA D2 6C evftf jp z,evnt15 ; yes, fallen too far.

4751 95CE C9 ret

4752 95CF

4753 95CF ; Old gravity processing for compatibility with 4.6 and 4.7.

4754 95CF

4755 95CF DD 5E 0E ogrv ld e,(ix+14) ; get index to table.

4756 95D2 16 00 ld d,0 ; no high byte.

4757 95D4 21 CA 81 ld hl,jtab ; jump table.

4758 95D7 19 add hl,de ; hl points to jump value.

4759 95D8 7E ld a,(hl) ; pixels to move.

4760 95D9 FE 63 cp 99 ; reached the end?

4761 95DB 20 04 jr nz,ogrv0 ; no, continue.

4762 95DD 2B dec hl ; go back to previous value.

4763 95DE 7E ld a,(hl) ; fetch that from table.

4764 95DF 18 03 jr ogrv1

4765 95E1 DD 34 0E ogrv0 inc (ix+14) ; point to next table entry.

4766 95E4 A7 ogrv1 and a ; any movement required?

4767 95E5 C8 ret z ; no, not this time.

4768 95E6 FE 80 cp 128 ; is it up or down?

4769 95E8 30 0C jr nc,ogrvu ; it's up.

4770 95EA 47 ogrvd ld b,a ; set pixels to move.

4771 95EB CD 70 8F ogrvd0 call cangd ; can we go down?

4772 95EE 20 14 jr nz,ogrvst ; can't move down, so stop.

4773 95F0 DD 34 08 inc (ix+8) ; adjust new x coord.

4774 95F3 10 F6 djnz ogrvd0

4775 95F5 C9 ret

4776 95F6 ED 44 ogrvu neg ; flip the sign so it's positive.

4777 95F8 47 ld b,a ; set pixels to move.

4778 95F9 CD 4D 8F ogrvu0 call cangu ; can we go up?

4779 95FC 20 1D jr nz,ogrv2 ; can't move up, go down next.

4780 95FE DD 35 08 dec (ix+8) ; adjust new x coord.

4781 9601 10 F6 djnz ogrvu0

4782 9603 C9 ret

4783 9604 DD 5E 0E ogrvst ld e,(ix+14) ; get index to table.

4784 9607 16 00 ld d,0 ; no high byte.

4785 9609 21 CA 81 ld hl,jtab ; jump table.

4786 960C 19 add hl,de ; hl points to jump value.

4787 960D 7E ld a,(hl) ; fetch byte from table.

4788 960E FE 63 cp 99 ; is it the end marker?

4789 9610 DD 36 0D 00 ld (ix+13),0 ; reset jump flag.

4790 9614 DD 36 0E 00 ld (ix+14),0 ; reset pointer.

4791 9618 C3 CB 95 jp evftf

4792 961B 21 CA 81 ogrv2 ld hl,jtab ; jump table.

4793 961E 06 00 ld b,0 ; offset into table.

4794 9620 7E ogrv4 ld a,(hl) ; fetch table byte.

4795 9621 FE 64 cp 100 ; hit end or downward move?

4796 9623 38 04 jr c,ogrv3 ; yes.

4797 9625 23 inc hl ; next byte of table.

4798 9626 04 inc b ; next offset.

4799 9627 18 F7 jr ogrv4 ; keep going until we find crest/end of table.

4800 9629 DD 70 0E ogrv3 ld (ix+14),b ; set next table offset.

4801 962C C9 ret

4802 962D

4803 962D ; Initiate fall check.

4804 962D

4805 962D DD 7E 0D ifall ld a,(ix+13) ; jump pointer flag.

4806 9630 A7 and a ; are we in the air?

4807 9631 C0 ret nz ; if set, we're already in the air.

4808 9632 DD 66 09 ld h,(ix+9) ; y coordinate.

4809 9635 3E 10 ld a,16 ; look down 16 pixels.

4810 9637 DD 86 08 add a,(ix+8) ; add x coordinate.

4811 963A 6F ld l,a ; coords in hl.

4812 963B 22 10 82 ld (dispx),hl ; set up test coordinates.

4813 963E CD 9C 90 call tstbl ; get map address.

4814 9641 CD CE 8F call plchk ; block, platform check.

4815 9644 C0 ret nz ; it's solid, don't fall.

4816 9645 23 inc hl ; look right one cell.

4817 9646 CD CE 8F call plchk ; block, platform check.

4818 9649 C0 ret nz ; it's solid, don't fall.

4819 964A 3A 11 82 ld a,(dispy) ; y coordinate.

4820 964D E6 07 and 7 ; position straddling block cells.

4821 964F 28 05 jr z,ifalls ; no more checks needed.

4822 9651 23 inc hl ; look to third cell.

4823 9652 CD CE 8F call plchk ; block, platform check.

4824 9655 C0 ret nz ; it's solid, don't fall.

4825 9656 DD 34 0D ifalls inc (ix+13) ; set in air flag.

4826 9659 DD 36 0E 00 ld (ix+14),0 ; initial speed = 0.

4827 965D C9 ret

4828 965E

4829 965E DD 7E 0D tfall ld a,(ix+13) ; jump pointer flag.

4830 9661 A7 and a ; are we in the air?

4831 9662 C0 ret nz ; if set, we're already in the air.

4832 9663 CD 2D 96 call ifall ; do fall test.

4833 9666 DD 7E 0D ld a,(ix+13) ; get falling flag.

4834 9669 A7 and a ; is it set?

4835 966A C8 ret z ; no.

4836 966B DD 36 0D FF ld (ix+13),255 ; we're using the table.

4837 966F 18 AA jr ogrv2 ; find position in table.

4838 9671

4839 9671

4840 9671 ; Get frame data for a particular sprite.

4841 9671

4842 9671 07 gfrm rlca ; multiple of 2.

4843 9672 5F ld e,a ; copy to de.

4844 9673 16 00 ld d,0 ; no high byte as max sprite is 128.

4845 9675 2A 12 82 ld hl,(frmptr) ; frames used by game.

4846 9678 19 add hl,de ; point to frame start.

4847 9679 C9 ret

4848 967A

4849 967A ; Find sprite list for current room.

4850 967A

4851 967A 3A EA 81 sprlst ld a,(scno) ; screen number.

4852 967D 2A 1C 82 sprls2 ld hl,(nmeptr) ; pointer to enemies.

4853 9680 47 ld b,a ; loop counter in b register.

4854 9681 A7 and a ; is it the first screen?

4855 9682 C8 ret z ; yes, don't need to search data.

4856 9683 11 04 00 ld de,NMESIZ ; bytes to skip.

4857 9686 7E sprls1 ld a,(hl) ; fetch type of sprite.

4858 9687 3C inc a ; is it an end marker?

4859 9688 28 03 jr z,sprls0 ; yes, end of this room.

4860 968A 19 add hl,de ; point to next sprite in list.

4861 968B 18 F9 jr sprls1 ; continue until end of room.

4862 968D 23 sprls0 inc hl ; point to start of next screen.

4863 968E 10 F6 djnz sprls1 ; continue until room found.

4864 9690 C9 ret

4865 9691

4866 9691

4867 9691 ; Clear all but a single player sprite.

4868 9691

4869 9691 06 0C nspr ld b,NUMSPR ; sprite slots in table.

4870 9693 DD 21 1C 98 ld ix,sprtab ; sprite table.

4871 9697 11 11 00 ld de,TABSIZ ; distance to next odd/even entry.

4872 969A DD 7E 00 nspr0 ld a,(ix+0) ; fetch sprite type.

4873 969D A7 and a ; is it a player?

4874 969E 28 0D jr z,nspr1 ; yes, keep this one.

4875 96A0 DD 36 00 FF ld (ix+0),255 ; delete sprite.

4876 96A4 DD 36 05 FF ld (ix+5),255 ; remove next type.

4877 96A8 DD 19 add ix,de ; next sprite.

4878 96AA 10 EE djnz nspr0 ; one less space in the table.

4879 96AC C9 ret

4880 96AD DD 36 00 FF nspr1 ld (ix+0),255 ; delete sprite.

4881 96B1 DD 19 add ix,de ; point to next sprite.

4882 96B3 10 01 djnz nspr2 ; one less to do.

4883 96B5 C9 ret

4884 96B6 DD 36 00 FF nspr2 ld (ix+0),255 ; delete sprite.

4885 96BA DD 36 05 FF ld (ix+5),255 ; remove next type.

4886 96BE DD 19 add ix,de ; next sprite.

4887 96C0 10 F4 djnz nspr2 ; one less space in the table.

4888 96C2 C9 ret

4889 96C3

4890 96C3

4891 96C3 ; Two initialisation routines.

4892 96C3 ; Initialise sprites - copy everything from list to table.

4893 96C3

4894 96C3 06 0C ispr ld b,NUMSPR ; sprite slots in table.

4895 96C5 DD 21 1C 98 ld ix,sprtab ; sprite table.

4896 96C9 7E ispr2 ld a,(hl) ; fetch byte.

4897 96CA FE FF cp 255 ; is it an end marker?

4898 96CC C8 ret z ; yes, no more to do.

4899 96CD DD 7E 00 ispr1 ld a,(ix+0) ; fetch sprite type.

4900 96D0 FE FF cp 255 ; is it enabled yet?

4901 96D2 20 07 jr nz,ispr4 ; yes, try another slot.

4902 96D4 DD 7E 05 ld a,(ix+5) ; next type.

4903 96D7 FE FF cp 255 ; is it enabled yet?

4904 96D9 28 08 jr z,ispr3 ; no, process this one.

4905 96DB 11 11 00 ispr4 ld de,TABSIZ ; distance to next odd/even entry.

4906 96DE DD 19 add ix,de ; next sprite.

4907 96E0 10 EB djnz ispr1 ; repeat for remaining sprites.

4908 96E2 C9 ret ; no more room in table.

4909 96E3 CD 18 97 ispr3 call cpsp ; initialise a sprite.

4910 96E6 10 E1 djnz ispr2 ; one less space in the table.

4911 96E8 C9 ret

4912 96E9

4913 96E9 ; Initialise sprites - but not player, we're keeping the old one.

4914 96E9

4915 96E9 06 0C kspr ld b,NUMSPR ; sprite slots in table.

4916 96EB DD 21 1C 98 ld ix,sprtab ; sprite table.

4917 96EF 7E kspr2 ld a,(hl) ; fetch byte.

4918 96F0 FE FF cp 255 ; is it an end marker?

4919 96F2 C8 ret z ; yes, no more to do.

4920 96F3 A7 and a ; is it a player sprite?

4921 96F4 20 06 jr nz,kspr1 ; no, add to table as normal.

4922 96F6 11 04 00 ld de,NMESIZ ; distance to next item in list.

4923 96F9 19 add hl,de ; point to next one.

4924 96FA 18 F3 jr kspr2

4925 96FC DD 7E 00 kspr1 ld a,(ix+0) ; fetch sprite type.

4926 96FF FE FF cp 255 ; is it enabled yet?

4927 9701 20 07 jr nz,kspr4 ; yes, try another slot.

4928 9703 DD 7E 05 ld a,(ix+5) ; next type.

4929 9706 FE FF cp 255 ; is it enabled yet?

4930 9708 28 08 jr z,kspr3 ; no, process this one.

4931 970A 11 11 00 kspr4 ld de,TABSIZ ; distance to next odd/even entry.

4932 970D DD 19 add ix,de ; next sprite.

4933 970F 10 EB djnz kspr1 ; repeat for remaining sprites.

4934 9711 C9 ret ; no more room in table.

4935 9712 CD 18 97 kspr3 call cpsp ; copy sprite to table.

4936 9715 10 D8 djnz kspr2 ; one less space in the table.

4937 9717 C9 ret

4938 9718

4939 9718 ; Copy sprite from list to table.

4940 9718

4941 9718 7E cpsp ld a,(hl) ; fetch byte from table.

4942 9719 DD 77 00 ld (ix+0),a ; set up type.

4943 971C DD 77 05 ld (ix+PAM1ST),a ; set up type.

4944 971F 23 inc hl ; move to next byte.

4945 9720 7E ld a,(hl) ; fetch byte from table.

4946 9721 DD 77 06 ld (ix+6),a ; set up image.

4947 9724 23 inc hl ; move to next byte.

4948 9725 7E ld a,(hl) ; fetch byte from table.

4949 9726 DD 36 03 C8 ld (ix+3),200 ; set initial coordinate off screen.

4950 972A DD 77 08 ld (ix+8),a ; set up coordinate.

4951 972D 23 inc hl ; move to next byte.

4952 972E 7E ld a,(hl) ; fetch byte from table.

4953 972F DD 77 09 ld (ix+9),a ; set up coordinate.

4954 9732 23 inc hl ; move to next byte.

4955 9733 AF xor a ; zeroes in accumulator.

4956 9734 DD 77 07 ld (ix+7),a ; reset frame number.

4957 9737 DD 77 0A ld (ix+10),a ; reset direction.

4958 973A ; ld (ix+12),a ; reset parameter B.

4959 973A DD 77 0D ld (ix+13),a ; reset jump pointer low.

4960 973D DD 77 0E ld (ix+14),a ; reset jump pointer high.

4961 9740 DD 36 10 FF ld (ix+16),255 ; reset data pointer to auto-restore.

4962 9744 DD E5 push ix ; store ix pair.

4963 9746 E5 push hl ; store hl pair.

4964 9747 C5 push bc

4965 9748 CD 7A 6A evis0 call evnt09 ; perform event.

4966 974B C1 pop bc

4967 974C E1 pop hl ; restore hl.

4968 974D DD E1 pop ix ; restore ix.

4969 974F 11 11 00 ld de,TABSIZ ; distance to next odd/even entry.

4970 9752 DD 19 add ix,de ; next sprite.

4971 9754 C9 ret

4972 9755

4973 9755 ; Clear the play area window.

4974 9755

4975 9755 2A E1 81 clw ld hl,(wintop) ; get coordinates of window.

4976 9758 22 10 82 ld (dispx),hl ; put into dispx for calculation.

4977 975B 3A E3 81 ld a,(winhgt) ; height of window.

4978 975E 47 ld b,a ; copy to b register.

4979 975F C5 clw3 push bc ; store lines on stack.

4980 9760 3A E4 81 ld a,(winwid) ; width of window.

4981 9763 08 clw2 ex af,af' ; store column counter.

4982 9764 CD 95 8D call gprad ; get print address.

4983 9767 AF xor a ; zero byute to write.

4984 9768 06 08 ld b,8 ; pixel height of each cell.

4985 976A 12 clw1 ld (de),a ; copy to screen.

4986 976B 14 inc d ; next screen row down.

4987 976C 10 FC djnz clw1

4988 976E CD C3 8D call gaadd ; get attribute address.

4989 9771 3A 8D 5C ld a,(23693) ; get colour.

4990 9774 77 ld (hl),a ; write colour.

4991 9775 21 11 82 ld hl,dispy ; column position.

4992 9778 34 inc (hl) ; next column.

4993 9779 08 ex af,af' ; restore column counter.

4994 977A 3D dec a ; one less to do.

4995 977B 20 E6 jr nz,clw2 ; repeat for remaining columns.

4996 977D 3A E2 81 ld a,(winlft) ; get left edge.

4997 9780 32 11 82 ld (dispy),a ; reset y.

4998 9783 21 10 82 ld hl,dispx ; line.

4999 9786 34 inc (hl) ; next line down.

5000 9787 C1 pop bc ; restore line counter.

5001 9788 10 D5 djnz clw3 ; repeat down the screen.

5002 978A 2A E1 81 ld hl,(wintop) ; get coordinates of window.

5003 978D 22 05 82 ld (charx),hl ; put into display position.

5004 9790 C9 ret

5005 9791

5006 9791 ; Effects code.

5007 9791 ; Ticker routine is called 25 times per second.

5008 9791

5009 9791 C9 scrly ret

5010 9792 A2 85 defw txtscr ; get screen address.

5011 9794 06 08 ld b,8 ; 8 pixel rows.

5012 9796 E5 push hl ; store screen address.

5013 9797 C5 scrly1 push bc ; store rows on stack.

5014 9798 E5 push hl

5015 9799 3A 9D 85 ld a,(txtwid) ; characters wide.

5016 979C 47 ld b,a ; put into the loop counter.

5017 979D A7 and a ; reset carry flag.

5018 979E CB 16 scrly0 rl (hl) ; rotate left.

5019 97A0 2D dec l ; char left.

5020 97A1 10 FB djnz scrly0 ; repeat for width of ticker message.

5021 97A3 E1 pop hl

5022 97A4 24 inc h ; next row down.

5023 97A5 C1 pop bc ; retrieve row counter from stack.

5024 97A6 10 EF djnz scrly1 ; repeat for all rows.

5025 97A8 2A 9E 85 ld hl,(txtpos) ; get text pointer.

5026 97AB 7E ld a,(hl) ; find character we're displaying.

5027 97AC E6 7F and 127 ; remove end marker bit if applicable.

5028 97AE FE 0D cp 13 ; is it newline?

5029 97B0 20 02 jr nz,scrly5 ; no, it's okay.

5030 97B2 3E 20 ld a,32 ; convert to a space instead.

5031 97B4 07 scrly5 rlca

5032 97B5 07 rlca

5033 97B6 07 rlca ; multiply by 8 to find char.

5034 97B7 47 ld b,a ; store shift in b.

5035 97B8 E6 03 and 3 ; keep within 768-byte range of font.

5036 97BA 57 ld d,a ; that's our high byte.

5037 97BB 78 ld a,b ; restore the shift.

5038 97BC E6 F8 and 248

5039 97BE 5F ld e,a

5040 97BF 2A 36 5C ld hl,(23606) ; font address.

5041 97C2 19 add hl,de ; point to image of character.

5042 97C3 EB ex de,hl ; need the address in de.

5043 97C4 E1 pop hl

5044 97C5

5045 97C5 3A 9C 85 ld a,(txtbit)

5046 97C8 4F ld c,a

5047 97C9 06 08 ld b,8

5048 97CB 1A scrly3 ld a,(de) ; get image of char line.

5049 97CC A1 and c ; test relevant bit of char.

5050 97CD 28 01 jr z,scrly2 ; not set - skip.

5051 97CF 34 inc (hl) ; set bit.

5052 97D0 24 scrly2 inc h ; next line of window.

5053 97D1 13 inc de ; next line of char.

5054 97D2 10 F7 djnz scrly3

5055 97D4 21 9C 85 ld hl,txtbit ; bit of text to display.

5056 97D7 CB 0E rrc (hl) ; next bit of char to use.

5057 97D9 D0 ret nc ; not reached end of character yet.

5058 97DA 2A 9E 85 ld hl,(txtpos) ; text pointer.

5059 97DD 7E ld a,(hl) ; what was the character?

5060 97DE 23 inc hl ; next character in message.

5061 97DF 17 rla ; end of message?

5062 97E0 ; ret nc ; not yet, exit here.

5063 97E0 ; ld a,201 ; code for ret.

5064 97E0 ; ld (scrly),a ; disable scrolling routine.

5065 97E0 30 03 jr nc,scrly6 ; not yet - continue.

5066 97E2 2A A0 85 scrly4 ld hl,(txtini) ; start of scrolling message.

5067 97E5 22 9E 85 scrly6 ld (txtpos),hl ; new text pointer position.

5068 97E8 C9 ret

5069 97E9

5070 97E9 CD 52 92 iscrly call prescr ; set up display position.

5071 97EC 21 45 6D ld hl,msgdat ; text messages.

5072 97EF 78 ld a,b ; width.

5073 97F0 3D dec a ; subtract one.

5074 97F1 FE 20 cp 32 ; is it between 1 and 32?

5075 97F3 30 0B jr nc,iscrl0 ; no, disable messages.

5076 97F5 79 ld a,c ; message number.

5077 97F6 50 ld d,b ; copy width to d.

5078 97F7 CD 5B 92 call getwrd ; find message start.

5079 97FA 42 ld b,d ; restore width to b register.

5080 97FB 22 A0 85 ld (txtini),hl ; set initial text position.

5081 97FE 3E 2A ld a,42 ; code for ld hl,(nn).

5082 9800 32 91 97 iscrl0 ld (scrly),a ; enable/disable scrolling routine.

5083 9803 CD 52 92 call prescr ; set up display position.

5084 9806 CD 95 8D call gprad ; get print address.

5085 9809 68 ld l,b ; width in b so copy to hl.

5086 980A 26 00 ld h,0 ; no high byte.

5087 980C 2B dec hl ; width minus one.

5088 980D 19 add hl,de ; add width.

5089 980E 22 A2 85 ld (txtscr),hl ; set text screen address.

5090 9811 78 ld a,b ; width.

5091 9812 32 9D 85 ld (txtwid),a ; set width in working storage.

5092 9815 21 9C 85 ld hl,txtbit ; bit of text to display.

5093 9818 36 80 ld (hl),128 ; start with leftmost bit.

5094 981A 18 C6 jr scrly4

5095 981C

5096 981C ; Sprite table.

5097 981C ; ix+0 = type.

5098 981C ; ix+1 = sprite image number.

5099 981C ; ix+2 = frame.

5100 981C ; ix+3 = x coord.

5101 981C ; ix+4 = y coord.

5102 981C

5103 981C ; ix+5 = new type.

5104 981C ; ix+6 = new image number.

5105 981C ; ix+7 = new frame.

5106 981C ; ix+8 = new x coord.

5107 981C ; ix+9 = new y coord.

5108 981C

5109 981C ; ix+10 = direction.

5110 981C ; ix+11 = parameter 1.

5111 981C ; ix+12 = parameter 2.

5112 981C ; ix+13 = jump pointer low.

5113 981C ; ix+14 = jump pointer high.

5114 981C ; ix+15 = data pointer low.

5115 981C ; ix+16 = data pointer high.

5116 981C

5117 981C

5118 981C sprtab equ $

5119 981C

5120 981C ; block NUMSPR \* TABSIZ,255

5121 981C

5122 981C defb 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255

5122 981C FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

5123 982D defb 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255

5123 982D FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

5124 983E defb 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255

5124 983E FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

5125 984F defb 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255

5125 984F FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

5126 9860 defb 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255

5126 9860 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

5127 9871 defb 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255

5127 9871 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

5128 9882 defb 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255

5128 9882 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

5129 9893 defb 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255

5129 9893 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

5130 98A4 defb 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255

5130 98A4 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

5131 98B5 defb 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255

5131 98B5 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

5132 98C6 defb 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255

5132 98C6 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

5133 98D7 defb 255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255

5133 98D7 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

5134 98E8 ssprit defb 255,255,255,255,255,255,255,0,192,120,0,0,0,255,255,255,255

5134 98E8 FFFFFFFFFFFFFF00C078000000FFFFFFFF

5135 98F9

5136 98F9 22 roomtb defb 34 ; room number.

5137 98FA

5138 98FA ; Everything below here will be generated by the editors.

5139 98FA

5140 98FA ; Sounds.

5141 98FA

5142 98FA 8F fx1 defb 128+15 ; volume and mixer.

5143 98FB 1F defb 31 ; white noise.

5144 98FC E8 03 defw 1000 ; tone register.

5145 98FE 8F defb 128+15 ; volume and mixer.

5146 98FF 19 defb 25 ; white noise.

5147 9900 E8 03 defw 1000 ; tone register.

5148 9902 8E defb 128+14 ; volume and mixer.

5149 9903 13 defb 19 ; white noise.

5150 9904 E8 03 defw 1000 ; tone register.

5151 9906 8D defb 128+13 ; volume and mixer.

5152 9907 0D defb 13 ; white noise.

5153 9908 E8 03 defw 1000 ; tone register.

5154 990A 8C defb 128+12 ; volume and mixer.

5155 990B 07 defb 7 ; white noise.

5156 990C E8 03 defw 1000 ; tone register.

5157 990E 8B defb 128+11 ; volume and mixer.

5158 990F 00 defb 0 ; white noise.

5159 9910 E8 03 defw 1000 ; tone register.

5160 9912 8A defb 128+10 ; volume and mixer.

5161 9913 06 defb 6 ; white noise.

5162 9914 E8 03 defw 1000 ; tone register.

5163 9916 88 defb 128+8 ; volume and mixer.

5164 9917 0C defb 12 ; white noise.

5165 9918 E8 03 defw 1000 ; tone register.

5166 991A 86 defb 128+6 ; volume and mixer.

5167 991B 12 defb 18 ; white noise.

5168 991C E8 03 defw 1000 ; tone register.

5169 991E 83 defb 128+3 ; volume and mixer.

5170 991F 18 defb 24 ; white noise.

5171 9920 E8 03 defw 1000 ; tone register.

5172 9922 FF defb 255

5173 9923

5174 9923 4F fx2 defb 064+15 ; volume and mixer.

5175 9924 1B defb 27 ; white noise.

5176 9925 E8 03 defw 1000 ; tone register.

5177 9927 4E defb 064+14 ; volume and mixer.

5178 9928 1F defb 31 ; white noise.

5179 9929 D0 07 defw 2000 ; tone register.

5180 992B 4D defb 064+13 ; volume and mixer.

5181 992C 1C defb 28 ; white noise.

5182 992D B8 0B defw 3000 ; tone register.

5183 992F 4C defb 064+12 ; volume and mixer.

5184 9930 1F defb 31 ; white noise.

5185 9931 D0 07 defw 2000 ; tone register.

5186 9933 4B defb 064+11 ; volume and mixer.

5187 9934 1D defb 29 ; white noise.

5188 9935 E8 03 defw 1000 ; tone register.

5189 9937 4A defb 064+10 ; volume and mixer.

5190 9938 1F defb 31 ; white noise.

5191 9939 D0 07 defw 2000 ; tone register.

5192 993B 49 defb 064+9 ; volume and mixer.

5193 993C 1E defb 30 ; white noise.

5194 993D B8 0B defw 3000 ; tone register.

5195 993F 48 defb 064+8 ; volume and mixer.

5196 9940 1F defb 31 ; white noise.

5197 9941 D0 07 defw 2000 ; tone register.

5198 9943 47 defb 064+7 ; volume and mixer.

5199 9944 1F defb 31 ; white noise.

5200 9945 E8 03 defw 1000 ; tone register.

5201 9947 46 defb 064+6 ; volume and mixer.

5202 9948 1F defb 31 ; white noise.

5203 9949 D0 07 defw 2000 ; tone register.

5204 994B FF defb 255

5205 994C

5206 994C 4F fx3 defb 064+15 ; volume and mixer.

5207 994D 00 defb 0 ; white noise.

5208 994E A0 0F defw 4000 ; tone register.

5209 9950 4F defb 064+15 ; volume and mixer.

5210 9951 00 defb 0 ; white noise.

5211 9952 04 10 defw 4100 ; tone register.

5212 9954 4E defb 064+14 ; volume and mixer.

5213 9955 00 defb 0 ; white noise.

5214 9956 68 10 defw 4200 ; tone register.

5215 9958 4E defb 064+14 ; volume and mixer.

5216 9959 00 defb 0 ; white noise.

5217 995A CC 10 defw 4300 ; tone register.

5218 995C 4D defb 064+13 ; volume and mixer.

5219 995D 00 defb 0 ; white noise.

5220 995E 30 11 defw 4400 ; tone register.

5221 9960 4D defb 064+13 ; volume and mixer.

5222 9961 00 defb 0 ; white noise.

5223 9962 94 11 defw 4500 ; tone register.

5224 9964 4C defb 064+12 ; volume and mixer.

5225 9965 00 defb 0 ; white noise.

5226 9966 F8 11 defw 4600 ; tone register.

5227 9968 4C defb 064+12 ; volume and mixer.

5228 9969 00 defb 0 ; white noise.

5229 996A 5C 12 defw 4700 ; tone register.

5230 996C 4B defb 064+11 ; volume and mixer.

5231 996D 00 defb 0 ; white noise.

5232 996E C0 12 defw 4800 ; tone register.

5233 9970 4A defb 064+10 ; volume and mixer.

5234 9971 00 defb 0 ; white noise.

5235 9972 24 13 defw 4900 ; tone register.

5236 9974 FF defb 255

5237 9975

5238 9975 63 defb 99 ; temporary marker.

5239 9976

5240 9976 include "user.asm"

0001+ 9976 C9 user ret

0002+ 9977

5241 9977

5242 9977 C9 plsnd ret

5243 9978 DEBUG equ mloop

5244 9978 C3 28 95 ibloc jp initsc

5245 997B C9 rbloc ret

5246 997C eop equ $

5247 997C savetap "test.tap",start

5248 997C