```
;##############################################################################
;#
;#  EIERSTOK.EXE  -  leesbare reconstructie
;#
;#  Demo: "Neural Wave Final" door The Black Lotus (TBL), 1997
;#  Party: LowRes '97  |  64KB intro (origineel was gepakt tot ~63 KB)
;#
;#  ------------------------------------------------------------------
;#  WAT IS DIT BESTAND?
;#  ------------------------------------------------------------------
;#  Dit is GEEN originele broncode van de makers. De originele .ASM is
;#  verloren. Wat je hier ziet is terugvertaald uit pure machinecode
;#  (de bytes die de processor uitvoert).
;#
;#  Doel van deze versie:
;#    - Zoveel mogelijk begrijpelijk maken, ook voor wie geen
;#      assembler-expert is.
;#    - Wel byte-identiek blijven aan het originele uitgepakte programma,
;#      zodat het nog steeds correct draait.
;#
;#  ------------------------------------------------------------------
;#  HOE DRAAIT ZO'N OUDE DEMO?
;#  ------------------------------------------------------------------
;#  1. Je start NWFINAL.EXE (klein, ~63 KB, gepakt met WWPACK).
;#  2. Die pakt intern dit programma (EIERSTOK, ~231 KB) uit.
;#  3. EIERSTOK is een 32-bit DOS-programma dat via de DOS32-extender
;#     draait. Het zet het scherm in een hoge VESA-resolutie en toont
;#     realtime grafische effecten (de "Neural Wave"-animatie).
;#
;#  ------------------------------------------------------------------
;#  BELANGRIJKE BEGRIPPEN (voor beginners)
;#  ------------------------------------------------------------------
;#  • Assembler / machinecode
;#      Elke regel hieronder wordt omgezet naar 1 of meer bytes die de
;#      CPU direct begrijpt. "mov eax, 1" wordt bijvoorbeeld B8 01 00 00 00.
;#
;#  • Register (eax, ebx, ecx, edx, esi, edi, ebp, esp, …)
;#      Kleine, razendsnelle opslagplaatsen ín de processor.
;#      Vergelijk met post-its op je bureau: limited, maar supersnel.
;#
;#  • Geheugenadres / label
;#      Een naam voor een plek in het geheugen, bv. Keyboard_GetKey.
;#      De CPU springt daar naartoe bij een "call" of "jmp".
;#
;#  • Interrupt (int 10h, int 16h, int 21h, int 31h)
;#      Een "hulplijn" naar het besturingssysteem of de BIOS:
;#        int 16h  = toetsenbord
;#        int 10h  = beeldscherm / VESA
;#        int 21h  = DOS (bestanden, afsluiten, …)
;#        int 31h  = DPMI / DOS32 (protected mode, geheugen, …)
;#
;#  • VESA
;#      Standaard om hoge resoluties (bijv. 640×480) te gebruiken op
;#      oude VGA/SVGA-kaarten. De demo zoekt een geschikte modus en
;#      schrijft pixels rechtstreeks naar het videogeheugen.
;#
;#  • db / times / incbin
;#      Manieren om ruwe data (getallen, strings, plaatjes) in het
;#      programma te stoppen in plaats van instructies.
;#
;#  • "db 0x32, 0xe4  ; xor ah, ah"
;#      Soms staat een instructie als ruwe bytes. Reden: de originele
;#      assembler (waarschijnlijk TASM) koos een andere maar equivalente
;#      encoding dan NASM. Gedrag is identiek; alleen de bytes verschillen
;#      als je het als normale instructie zou schrijven.
;#
;#  ------------------------------------------------------------------
;#  STRUCTUUR VAN DIT PROGRAMMA (globaal)
;#  ------------------------------------------------------------------
```

```
;#  [1]  Leeg/BSS-gebied          - variabelen die op 0 beginnen
;#  [2]  Small helpers            - toetsenbord, DOS32-hulpjes
;#  [3]  VESA-initialisatie       - schermmodus zoeken en instellen
;#  [4]  Blit / pixel-copy        - beelden naar het scherm kopiëren
;#  [5]  Hoofdlogica & effecten   - start, scripts, processlist, timeline
;#  [5a] Script-interpreter       - bytecode voor scènes/effecten
;#  [5b] ProcessList-interpreter  - bytecode voor 3D/objecten
;#  [5c] Timeline                 - tijd-/muziek-gestuurde events
;#  [5d] Videomodus-helpers       - resolutie en buffers kiezen
;#  [5e] 3D-objecten & meshes     - alloc, grid, translate, rotate
;#  [5f] Matrix / vertex-transform
;#  [5g] Pixel-effecten           - o.a. plasma (sinus-patronen)
;#  [6]  Data & tabellen          - kleuren, coördinaten, graphics, …
;#
;#  Waar de rol duidelijk is uit de code (interrupts, lussen, FPU,
;#  jump-tabellen, …) staan Nederlandse namen + uitleg.
;#  Resterende L_0xxxx/D_0xxxx: adres bekend, functie nog niet hard
;#  genoeg te benoemen zonder te gissen (vaak unieke effectvarianten
;#  of data). Die zijn bewust generiek gelaten.
;#
;#  ---------------------------------------------------------------
;#  BOUWEN
;#  ---------------------------------------------------------------
;#    nasm -f bin eierstok_leesbaar.asm -o image.bin
;#    cat assets/dos32_stub.bin assets/dos32_header.bin image.bin \
;#        assets/reloc_tail.bin > EIERSTOK_REBUILT.EXE
;#
;#  Resultaat is byte-identiek aan het originele uitgepakte programma.
;#
;###############################################################################

bits 32
org 0

; ============================================================================
; [1] Leeg startgebied / variabelen (BSS)
;
; Bij het laden staat dit geheugen op nul. Hier bewaart het programma
; later status, schermbreedte, pointers naar buffers, enz.
; ============================================================================

    times 8192 db 0
D_02000:
    db 0x00
D_02001:
    db 0x00
D_02002:
    db 0x00
D_02003:
    db 0x00
D_02004:
    db 0x00
D_02005:
    db 0x00
D_02006:
    db 0x00
D_02007:
    db 0x00
D_02008:
    db 0x00
D_02009:
    db 0x00
D_0200a:
    db 0x00
D_0200b:
    db 0x00
D_0200c:
    db 0x00, 0x00, 0x00, 0x00
D_02010:
```

```
    db 0x00, 0x00, 0x00, 0x00
D_02014:
    db 0x00, 0x00, 0x00, 0x00
D_02018:
    times 8 db 0
    db 0xff, 0xff, 0xff, 0xff
dos32_base_offset:
    db 0x00, 0x00, 0x00, 0x00
D_02028:
    db 0x00, 0x00
vesa_mode_count:
    times 12 db 0
; ------------------------------------------------------------------------
; [2a] DOS32 / DPMI-hulpjes
;
; DOS32 is een "extender": hij laat 32-bit code draaien onder gewoon DOS.
; Via interrupt 31h praat het programma met die extender (geheugen,
;# selectors, enz.).
; ------------------------------------------------------------------------
Dos32_GetSelectorBase:
    ; Vraag aan DOS32: "wat is het basisadres van deze selector?"
    ; (functie AX = EE42h). Wordt o.a. gebruikt om lineaire adressen
    ; om te rekenen.
    mov eax, 0xee42
    int 0x31
    ret
    ; Onbereikbare rest (staat in de originele bytes; niet uitgevoerd)
    mov eax, 0xee40
    int 0x31
    ret

Dos_Terminate:
    ; Laadt een waarde in EAX en roept DOS-interrupt 21h aan.
    ; (In deze context geen klassieke "AH=4Ch exit", maar een
    ;  DOS32-gerelateerde call; labelnaam is historisch.)
    mov eax, Blit_AddYOffset
    int 0x21
D_0204d:
    db 0x00, 0x00, 0x00, 0x00
D_02051:
    db 0x00, 0x00, 0x00, 0x00
D_02055:
    db 0x00, 0x00, 0x00, 0x00
D_02059:
    db 0x00, 0x00, 0x00, 0x00
D_0205d:
    db 0x00, 0x00, 0x00, 0x00
D_02061:
    db 0x00, 0x00, 0x00, 0x00
D_02065:
    db 0x00
D_02066:
    db 0x00
D_02067:
    db 0x6b, 0x20, 0x00, 0x00
D_0206b:
    db 0x00, 0x00, 0x00, 0x00
D_0206f:
    db 0x00, 0x00, 0x00, 0x00
D_02073:
    db 0x00, 0x00, 0x00, 0x00
; ------------------------------------------------------------------------
; [2b] Toetsenbord (BIOS interrupt 16h)
;
; De BIOS biedt eenvoudige toetsenbordfuncties:
;#   AH=0  → wacht tot de gebruiker een toets indrukt, geef de code terug
;#   AH=1  → kijk of er al een toets is (zonder te wachten)
;
; Keyboard_ReadNormalized normaliseert de invoer:
```

```
;#   - letters worden hoofdletters
;#   - ENTER en ESC worden apart herkend
;#   - het laatste teken komt in kbd_last_char
; ------------------------------------------------------------------------
Keyboard_GetKey:
    ; Wacht op een toets (BIOS AH=0).
    ; "db 0x32,0xe4" = xor ah,ah in de encoding van de originele assembler.
    db 0x32, 0xe4     ; xor ah, ah
    int 0x16
    ret

Keyboard_CheckKey:
    ; Is er al een toets ingedrukt? (BIOS AH=1)
    ; Zero-flag = 1 betekent: nog geen toets.
    mov ah, 1
    int 0x16
    ret

Keyboard_ReadNormalized:
    ; Lees een toets, maak er een hoofdletter van indien nodig,
    ; en onthoud ENTER/ESC. Resultaat o.a. in kbd_last_char.
    mov byte [kbd_last_char], 0
    test byte [kbd_enter_flag], 1
    jne short Keyboard_WaitKey
    call Keyboard_CheckKey
    je short Keyboard_SetReady
    call Keyboard_GetKey
    cmp al, 0x7a
    ja short Keyboard_CheckEsc
    cmp al, 0x61
    jae short Keyboard_ToUpper
    cmp al, 0x5a
    ja short Keyboard_CheckEsc
    cmp al, 0x41
    jae short Keyboard_StoreChar
    cmp al, 0x39
    ja short Keyboard_CheckEsc
    cmp al, 0x30
    jae short Keyboard_StoreChar
Keyboard_CheckEsc:
    cmp al, 0x1b
    je short Keyboard_Ret
Keyboard_WaitKey:
    call Keyboard_GetKey
    cmp al, 0xd
    jne short Keyboard_ClearEnterFlag
    mov byte [kbd_enter_flag], 1
    or al, 1
    jmp short Keyboard_Ret
Keyboard_ClearEnterFlag:
    mov byte [kbd_enter_flag], 0
    cmp al, 0x1b
Keyboard_Ret:
    ret
Keyboard_ToUpper:
    sub al, 0x20
Keyboard_StoreChar:
    mov byte [kbd_last_char], al
Keyboard_SetReady:
    or al, 1
    ret

; --- Toetsenbordstatus ---
kbd_enter_flag:          ; 1 = ENTER is recent gebruikt
    db 0x00
kbd_last_char:           ; laatste genormaliseerde toets (ASCII)
    db 0x00, 0x90
; --- Video-/modusvlaggen ---
flag_020e4:              ; actieve buffer/modus-id (1/2/3)
```

```
277         db 0x00
278     flag_020e5:
279         db 0x00
280     flag_020e6:             ; parameter bij mode-set
281         db 0x00
282     vesa_mode_flags:        ; o.a. bit7 = banked vs linear pad
283         db 0x00
284     ; --- Framebuffer-pointers (bron voor blit) ---
285     ptr_framebuffer_src:    ; huidige render-bron
286         db 0x00, 0x00, 0x00, 0x00
287     ptr_020ec:              ; buffer bij modus 2
288         db 0x00, 0x00, 0x00, 0x00
289     ptr_020f0:              ; buffer bij modus 3
290         db 0x00, 0x00, 0x00, 0x00
291     ptr_020f4:              ; o.a. breedte-gerelateerde hulp
292         db 0x00, 0x00, 0x00, 0x00
293     D_020f8:
294         times 12 db 0
295     D_02104:
296         db 0x00, 0x00
297
298     ; --- Schermgeometrie (na Video_ActivateMode) ---
299     screen_width:
300         db 0x00, 0x00, 0x00, 0x00
301     screen_height_or_pitch:
302         db 0x00, 0x00, 0x00, 0x00
303     screen_height:
304         db 0x00, 0x00, 0x00, 0x00
305     screen_bytes_per_scan:  ; bytes per beeldlijn (pitch)
306         db 0x00, 0x00, 0x00, 0x00
307     D_02116:
308         db 0x00, 0x00
309     D_02118:
310         db 0x00
311     D_02119:
312         db 0x00
313     D_0211a:
314         db 0x00
315     D_0211b:
316         db 0x00
317     D_0211c:
318         db 0x00
319     D_0211d:
320         db 0x00
321     D_0211e:
322         db 0x00
323     D_0211f:
324         times 9 db 0
325     D_02128:
326         times 9 db 0
327     vesa_win_func_ptr:
328         db 0x00, 0x00, 0x00, 0x00
329     vesa_win_granularity:
330         db 0x00, 0x00, 0x00, 0x00
331     linear_framebuffer:
332         times 12 db 0
333     D_02145:
334         db 0x00, 0x00, 0x00, 0x00
335
336     ; ------------------------------------------------------------------------
337     ; VESA_FindModeBySize - zoek VESA-modus met gevraagde breedte/hoogte/bpp
338     ;
339     ; EAX=breedte, EBX=hoogte, DL=bits-per-pixel.
340     ; Loopt door de eerder opgebouwde modustabel; bij treffer: AX=mode-nummer.
341     ; ------------------------------------------------------------------------
342     VESA_FindModeBySize:
343         movzx ecx, word [vesa_mode_count]
344         inc ecx
345         mov esi, VESA_ModeTable
```

```
346 VESA_FindMode_Scan:
347     add esi, 8
348     dec ecx
349     je short VESA_FindMode_Fail
350     cmp ax, word [esi]
351     jne short VESA_FindMode_Scan
352     cmp bx, word [esi + 2]
353     jne short VESA_FindMode_Scan
354     cmp dl, byte [esi + 4]
355     jne short VESA_FindMode_Scan
356     mov ax, word [esi + 5]
357 VESA_ModeTable:
358     clc
359     ret
360 VESA_FindMode_Fail:
361     mov ax, 0xffff
362     stc
363     ret
364     times 8192 db 0
365     db "not enough free low-memory! blame DOS32!!"
366     db 0x07
367     db "$no vesa extensions installed, use univbe! (of koop een botte et6000!!!!)"
368     db 0x07
369     db "$unable to map physical memory, blame DOS32!!"
370     db 0x07
371     db "$someone messed with the VBEPMEXT buffer located in default.inc, this buffer"
372     db 0x0a, 0x0d
373     db "has GOT to be at offset 0...maybe dos32 v3.5+ ???? want die heeft lamme allocs"
374     db 0x07
375     db "$unable to use the panning capabilities of your VBE driver"
376     db 0x07
377     db "$unable to use the banking capabilities of your VBE driver"
378     db 0x07, 0x24
379 D_0432e:
380     db 0x00, 0x00, 0x00, 0x00
381 D_04332:
382     db 0x00, 0x00
383 D_04334:
384     db 0x00, 0x00, 0x00, 0x00
385 D_04338:
386     times 16 db 0
387 D_04348:
388     times 8 db 0
389 D_04350:
390     db 0x00, 0x00, 0x00, 0x00
391 D_04354:
392     db 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
393 D_0435a:
394     times 16 db 0
395 D_0436a:
396     db 0x00, 0x00, 0x00, 0x00
397
398 ; ============================================================================
399 ; [3] VESA / SuperVGA - het scherm klaarzetten
400 ;
401 ; Stappen in mensentaal:
402 ;   1. Vraag de videokaart: "ondersteun je VESA?"
403 ;   2. Loop door alle beschikbare grafische modi (resoluties).
404 ;   3. Kies een geschikte (bij voorkeur met lineair framebuffer).
405 ;   4. Map dat videogeheugen zodat we er pixels naartoe kunnen schrijven.
406 ;   5. Als het mislukt → toon foutmelding en stop.
407 ;
408 ; Zonder deze stap zou de demo geen hoge resolutie kunnen gebruiken.
409 ; ============================================================================
410 VESA_Init:
411     mov eax, 0xa0000
412     sub eax, dword [dos32_base_offset]
413     mov dword [linear_framebuffer], eax
414     mov ah, 0x48
```

```
415         mov bx, 0x20
416         int 0x21
417         mov edx, 0x4178
418         jb near Fatal_NotEnoughVram
419         movzx eax, ax
420         mov dword [D_0432e], eax
421         db 0x8b, 0xe8       ; mov ebp, eax
422         shl ebp, 4
423         sub ebp, dword [dos32_base_offset]
424         mov edi, D_04338
425         mov word [edi + 0x22], ax
426         mov word [edi + 0x1c], 0x4f00
427         mov word [edi], 0
428         mov ax, 0x300
429         mov bx, 0x10
430         db 0x66, 0x33, 0xc9     ; xor cx, cx
431         int 0x31
432         mov ax, word [edi + 0x1c]
433         db 0x66, 0x3d, 0x4f, 0x00     ; cmp ax, 0x4f
434         mov edx, 0x41a3
435         jne near Fatal_NotEnoughVram
436         cmp dword [ds:ebp], 0x41534556
437         jne near Fatal_NotEnoughVram
438         db 0x66, 0x3e, 0x81, 0x7d, 0x04, 0x02, 0x01     ; cmp word ptr ds:[ebp + 4], 0x102
439         jb near Fatal_NotEnoughVram
440         db 0x66, 0x3e, 0x8b, 0x45, 0x12     ; mov ax, word ptr ds:[ebp + 0x12]
441         shl ax, 6
442         cmp ax, 0x800
443         jbe short VESA_EnumModes
444         mov ax, 0x800
445     VESA_EnumModes:
446         mov word [D_04332], ax
447         cmp ax, 0x400
448         jb near Fatal_ErrorMsg
449         mov edi, D_04338
450         mov ecx, 0x32
451         db 0x32, 0xc0     ; xor al, al
452         rep stosb
453         movzx eax, word [ds:ebp + 0xe]
454         movzx esi, word [ds:ebp + 0x10]
455         shl esi, 4
456         db 0x03, 0xf0     ; add esi, eax
457         sub esi, dword [dos32_base_offset]
458         mov ebx, 0x2178
459         mov word [vesa_mode_count], 0
460         mov dword [D_04334], 0
461     VESA_NextMode:
462         movzx ecx, word [esi]
463         add esi, 2
464         cmp cx, -1
465         je near VESA_MapFramebuffer
466         push ebx
467         push ecx
468         mov edi, D_04338
469         mov eax, dword [D_0432e]
470         mov word [edi + 0x22], ax
471         mov word [edi + 0x1c], 0x4f01
472         mov word [edi + 0x18], cx
473         mov word [edi], 0
474         mov ax, 0x300
475         mov bx, 0x10
476         db 0x66, 0x33, 0xc9     ; xor cx, cx
477         int 0x31
478         pop ecx
479         pop ebx
480         mov ax, word [edi + 0x1c]
481         db 0x66, 0x3d, 0x4f, 0x00     ; cmp ax, 0x4f
482         jne short VESA_NextMode
483         mov al, byte [ds:ebp]
```

```
484         test al, 1
485         je short VESA_NextMode
486         test al, 8
487         je short VESA_NextMode
488         test al, 0x10
489         je short VESA_NextMode
490         mov byte [ebx + 7], al
491         mov eax, dword [ds:ebp + 0x28]
492         test eax, eax
493         je short VESA_ModeLoop
494         mov dword [D_04334], eax
495         jmp short VESA_StoreModeInfo
496     VESA_ModeLoop:
497         and byte [ebx + 7], 0x7f
498     VESA_StoreModeInfo:
499         mov eax, dword [ds:ebp + 0x12]
500         mov word [ebx], ax
501         db 0x66, 0x3e, 0x8b, 0x45, 0x14     ; mov ax, word ptr ds:[ebp + 0x14]
502         mov word [ebx + 2], ax
503         mov al, byte [ds:ebp + 0x19]
504         cmp al, 8
505         jbe short VESA_CheckModeAttrs
506         mov al, byte [ds:ebp + 0x1f]
507         add al, byte [ds:ebp + 0x21]
508         add al, byte [ds:ebp + 0x23]
509     VESA_CheckModeAttrs:
510         mov byte [ebx + 4], al
511         mov word [ebx + 5], cx
512         cmp word [ebx], 0x280
513         jne short VESA_ModeAdvance
514         sub al, 0xf
515         cmp al, 1
516         ja short VESA_ModeAdvance
517         test byte [ds:ebp], 0x80
518         jne short VESA_ModeAdvance
519         db 0x66, 0x3e, 0x8b, 0x45, 0x0a     ; mov ax, word ptr ds:[ebp + 0xa]
520         test ax, 3
521         jne near VESA_NextMode
522     VESA_ModeAdvance:
523         add ebx, 8
524         inc word [vesa_mode_count]
525         jmp near VESA_NextMode
526     VESA_MapFramebuffer:
527         cmp dword [D_04334], 0
528         je short VESA_GetWindowControl
529         mov ax, 0x800
530         mov ebx, dword [D_04334]
531         db 0x66, 0x8b, 0xcb     ; mov cx, bx
532         shr ebx, 0x10
533         movzx esi, word [D_04332]
534         shl esi, 0xa
535         db 0x66, 0x8b, 0xfe     ; mov di, si
536         shr esi, 0x10
537         int 0x31
538         mov edx, 0x41ed
539         jb near Fatal_NotEnoughVram
540         shl ebx, 0x10
541         db 0x66, 0x8b, 0xd9     ; mov bx, cx
542         sub ebx, dword [dos32_base_offset]
543         mov dword [D_0436a], ebx
544     VESA_GetWindowControl:
545         mov dword [vesa_win_func_ptr], 0
546         mov dword [vesa_win_granularity], 0
547         mov eax, 0
548         test eax, eax
549         jne short VESA_FallbackSetDisplayStart
550         mov edi, D_04338
551         mov word [D_04354], 0x4f0a
552         mov word [D_04348], 0
```

```
553         mov ax, 0x300
554         mov bx, 0x10
555         db 0x66, 0x33, 0xc9     ; xor cx, cx
556         int 0x31
557         mov ax, word [D_04354]
558         db 0x66, 0x3d, 0x4f, 0x00     ; cmp ax, 0x4f
559         jne short VESA_FallbackSetDisplayStart
560         movzx edx, word [D_0435a]
561         shl edx, 4
562         movzx eax, word [D_04338]
563         db 0x03, 0xd0     ; add edx, eax
564         je short VESA_FallbackSetDisplayStart
565         db 0x8b, 0xf2     ; mov esi, edx
566         sub esi, dword [dos32_base_offset]
567         mov edi, 0
568         movzx ecx, word [D_04350]
569         rep movsb
570         db 0x67, 0x0f, 0xb7, 0x06, 0x02, 0x00     ; movzx eax, word ptr [2]
571         mov dword [vesa_win_func_ptr], eax
572         db 0x67, 0x0f, 0xb7, 0x06, 0x00, 0x00     ; movzx eax, word ptr [0]
573         mov dword [vesa_win_granularity], eax
574         cmp dword [vesa_win_func_ptr], 0
575         jne short VESA_CheckWindow
576     VESA_FallbackSetDisplayStart:
577         mov ax, 0x4f07
578         mov bx, 1
579         int 0x10
580         cmp al, 0x4f
581         mov edx, 0x42b8
582         jne short Fatal_NotEnoughVram
583     VESA_CheckWindow:
584         cmp dword [vesa_win_granularity], 0
585         jne short VESA_InitOK
586         mov ax, 0x4f05
587         mov bx, 0x100
588         int 0x10
589         cmp al, 0x4f
590         mov edx, 0x42f3
591         jne short Fatal_NotEnoughVram
592     VESA_InitOK:
593         clc
594         ret
595     ; ---------------------------------------------------------------------------
596     ; Foutafhandeling - netjes stoppen met een melding
597     ;
598     ; int 21h AH=9  = print een $-afgesloten tekststring
599     ; int 21h AX=4C00h = programma beëindigen (exit code 0)
600     ; ---------------------------------------------------------------------------
601     Fatal_NotEnoughVram:
602         ; Niet genoeg videogeheugen (minimaal 512 KB verwacht).
603         ; DS:DX wijst al naar de fouttekst hieronder.
604         mov ah, 9
605         int 0x21
606         mov ax, 0x4c00
607         int 0x21
608         db "not enough memory on board of your gfx-adapter, should be at least 512kb!!"
609         db 0x07, 0x24
610
611     ; ---------------------------------------------------------------------------
612     ; Andere fatale fout (generieke melding + afsluiten)
613     ; ---------------------------------------------------------------------------
614     Fatal_ErrorMsg:
615         mov edx, 0x4636
616         mov ah, 9
617         int 0x21
618         mov ax, 0x4c00
619         int 0x21
620     VESA_ClearInfoBlock:
621         push eax
```

```
622         mov edi, D_04338
623         mov ecx, 0x32
624         db 0x32, 0xc0     ; xor al, al
625         rep stosb
626         mov edi, D_04338
627         mov eax, dword [D_0432e]
628         mov word [edi + 0x22], ax
629         pop eax
630         mov word [edi + 0x1c], 0x4f02
631         mov word [edi + 0x10], ax
632         mov word [edi], 0
633         mov ax, 0x300
634         mov bx, 0x10
635         db 0x66, 0x33, 0xc9     ; xor cx, cx
636         int 0x31
637         mov ax, word [edi + 0x1c]
638         jb short VESA_Call_Failed
639         db 0x66, 0x3d, 0x4f, 0x00     ; cmp ax, 0x4f
640         jne short VESA_Call_Failed
641         clc
642         ret
643     VESA_Call_Failed:
644         stc
645         ret
646         db 0x00, 0x00
647
648     ; ------------------------------------------------------------------------
649     ; Video_PickBestMode - kies de beste beschikbare resolutie
650     ;
651     ; Probeert in volgorde (met 16-bit en 15-bit kleur):
652     ;   320×200  →  640×400  →  640×480
653     ; Bij succes: Video_ActivateMode (VESA mode set + framebuffer).
654     ; Bij totaal falen: Video_Fatal_NoMode.
655     ; ------------------------------------------------------------------------
656     Video_PickBestMode:
657         mov dword [ptr_framebuffer_src], 0
658         mov dword [ptr_020f0], 0
659         mov dword [ptr_020ec], 0
660         mov byte [flag_020e5], 0
661         mov dword [D_02145], Video_Try_640x400
662         mov eax, 0x140
663         mov ebx, 0xc8
664         mov dl, 0x10
665         call VESA_FindModeBySize
666         db 0x66, 0x8b, 0xf8     ; mov di, ax
667         mov eax, 0x140
668         mov ebx, 0xc8
669         mov dl, 0x10
670         jae near Video_ActivateMode
671         mov eax, 0x140
672         mov ebx, 0xc8
673         mov dl, 0xf
674         call VESA_FindModeBySize
675         db 0x66, 0x8b, 0xf8     ; mov di, ax
676         mov eax, 0x140
677         mov ebx, 0xc8
678         mov dl, 0xf
679         jae near Video_ActivateMode
680
681     ; Volgende poging: 640×400 (0x280 × 0x190)
682     Video_Try_640x400:
683         mov dword [D_02145], Video_Try_640x480
684         mov eax, 0x280
685         mov ebx, 0x190
686         mov dl, 0x10
687         call VESA_FindModeBySize
688         db 0x66, 0x8b, 0xf8     ; mov di, ax
689         mov eax, 0x280
690         mov ebx, 0x190
```

```
    mov dl, 0x10
    jae short Video_ActivateMode
    mov eax, 0x280
    mov ebx, 0x190
    mov dl, 0xf
    call VESA_FindModeBySize
    db 0x66, 0x8b, 0xf8     ; mov di, ax
    mov eax, 0x280
    mov ebx, 0x190
    mov dl, 0xf
    jae short Video_ActivateMode

; Volgende poging: 640×480 (0x280 × 0x1e0)
Video_Try_640x480:
    mov dword [D_02145], Video_NoMode_Abort
    mov eax, 0x280
    mov ebx, 0x1e0
    mov dl, 0x10
    call VESA_FindModeBySize
    db 0x66, 0x8b, 0xf8     ; mov di, ax
    mov eax, 0x280
    mov ebx, 0x1e0
    mov dl, 0x10
    jae short Video_ActivateMode
    mov eax, 0x280
    mov ebx, 0x1e0
    mov dl, 0xf
    call VESA_FindModeBySize
    db 0x66, 0x8b, 0xf8     ; mov di, ax
    mov eax, 0x280
    mov ebx, 0x1e0
    mov dl, 0xf
    jae short Video_ActivateMode
Video_NoMode_Abort:
    jmp near Video_Fatal_NoMode

; ---------------------------------------------------------------------------
; Video_ActivateMode - gekozen VESA-modus echt aanzetten
;
; Slaat breedte/hoogte/bpp op, roept VESA 4F01h/4F02h-achtige calls
; (via DPMI int 31h → real-mode int 10h) en zet linear_framebuffer.
; ---------------------------------------------------------------------------
Video_ActivateMode:
    mov byte [D_02118], dl
    mov word [D_02116], di
    mov dword [screen_height_or_pitch], eax
    mov dword [screen_height], ebx
    db 0x66, 0x8b, 0xcf     ; mov cx, di
    mov edi, D_04338
    mov eax, dword [D_0432e]
    db 0x8b, 0xe8     ; mov ebp, eax
    shl ebp, 4
    sub ebp, dword [dos32_base_offset]
    push eax
    push ecx
    mov word [edi + 0x22], ax
    mov word [edi + 0x1c], 0x4f01
    mov word [edi + 0x18], cx
    mov word [edi], 0
    mov ax, 0x300
    mov bx, 0x10
    db 0x66, 0x33, 0xc9     ; xor cx, cx
    int 0x31
    pop ecx
    pop eax
    cmp word [edi + 0x1c], 0x4f
    je short Video_ModeSet_OK
    or cx, 0x4000
    mov word [edi + 0x22], ax
```

```
760         mov word [edi + 0x1c], 0x4f01
761         mov word [edi + 0x18], cx
762         mov word [edi], 0
763         mov ax, 0x300
764         mov bx, 0x10
765         db 0x66, 0x33, 0xc9     ; xor cx, cx
766         int 0x31
767         cmp word [edi + 0x1c], 0x4f
768         jne near Video_Fatal_NoMode
769     Video_ModeSet_OK:
770         cmp word [ebx], 0x140
771         jne short Video_AfterModeSet
772         cmp word [ebx + 2], 0xc8
773         jne short Video_AfterModeSet
774         cmp byte [ebx + 4], 0xf
775         jb short Video_AfterModeSet
776         db 0x66, 0x3e, 0x81, 0x7d, 0x10, 0x80, 0x02     ; cmp word ptr ds:[ebp + 0x10], 0x280
777         je short Video_AfterModeSet
778         and byte [ds:ebp], 0x7f
779     Video_AfterModeSet:
780         movzx eax, word [ds:ebp + 0x10]
781         mov dword [screen_width], eax
782         movzx eax, word [ds:ebp + 6]
783         shl eax, 0xa
784         mov dword [screen_bytes_per_scan], eax
785         mov al, byte [ds:ebp + 0x1f]
786         mov byte [D_0211c], al
787         mov al, byte [ds:ebp + 0x20]
788         mov byte [D_02119], al
789         mov al, byte [ds:ebp + 0x21]
790         mov byte [D_0211d], al
791         mov al, byte [ds:ebp + 0x22]
792         mov byte [D_0211a], al
793         mov al, byte [ds:ebp + 0x23]
794         mov byte [D_0211e], al
795         mov al, byte [ds:ebp + 0x24]
796         mov byte [D_0211b], al
797         mov al, byte [D_02118]
798         sub al, 0xf
799         xor al, 1
800         mov byte [D_0211f], al
801         cmp dword [screen_height_or_pitch], 0x140
802         setne al
803         shl al, 2
804         or byte [D_0211f], al
805         cmp dword [D_04334], 0
806         je short Video_Mode_Done
807         test byte [ds:ebp], 0x80
808         je short Video_Mode_Done
809         mov eax, dword [D_0436a]
810         mov dword [ptr_framebuffer_src], eax
811         or word [D_02116], 0x4000
812         jmp short Video_Mode_Fail
813     Video_Mode_Done:
814         or byte [D_0211f], 2
815     Video_Mode_Fail:
816         clc
817         ret
818     Video_Fatal_NoMode:
819         mov edx, 0x496b
820         mov ah, 9
821         int 0x21
822         mov ax, 0x4c00
823         int 0x21
824         db 0xc3, 0xba, 0x61, 0x49, 0x00, 0x00, 0xb4, 0x09, 0xcd, 0x21, 0xf9, 0xc3
825         db "We suck."
826         db 0x07
827         db "$We suck."
828         db 0x0a, 0x0d, 0x24
```

```
829
830 ; ------------------------------------------------------------------------
831 ; VESA_SetBank / VESA_SetDisplayStart - bank-switching (oudere kaarten)
832 ;
833 ; Niet alle kaarten hebben een lineair framebuffer. Dan schrijf je naar
834 ; een "raam" van 64 KB en wissel je banken (of display-start) via VESA.
835 ; Blit-routines roepen dit aan per scanline-blok.
836 ; ------------------------------------------------------------------------
837 VESA_SetBank:
838     cmp dword [vesa_win_func_ptr], 0
839     je short VESA_SetBank_Win
840     xchg ebx, eax
841     shr ebx, 2
842     shr ecx, 2
843     mul ecx
844     db 0x03, 0xc3      ; add eax, ebx
845     db 0x8b, 0xc8      ; mov ecx, eax
846     shr eax, 0x10
847     db 0x8b, 0xd0      ; mov edx, eax
848     mov ebx, 0x80
849     sub bl, byte [vesa_mode_flags]
850     call dword [vesa_win_func_ptr]
851     jmp short VESA_Bank_Ret
852 VESA_SetBank_Win:
853     mov edi, D_04338
854     mov word [edi + 0x1c], 0x4f07
855     mov word [edi + 0x18], ax
856     mov word [edi + 0x14], bx
857     mov word [edi + 0x10], 0x80
858     mov bl, byte [vesa_mode_flags]
859     sub byte [edi + 0x10], bl
860     mov ax, 0x300
861     mov bx, 0x10
862     db 0x66, 0x33, 0xc9      ; xor cx, cx
863     int 0x31
864 VESA_Bank_Ret:
865     cmp byte [vesa_mode_flags], 0
866     je short VESA_Bank_End
867     mov edx, 0x3da
868 VESA_Bank_Alt:
869     in al, dx
870     test al, 8
871     je short VESA_Bank_Alt
872 VESA_Bank_End:
873     ret
874 VESA_SetDisplayStart:
875     cmp dword [vesa_win_granularity], 0
876     jne short VESA_DispStart_Ret
877     push eax
878     push ebx
879     push edi
880     push edx
881     push edx
882     mov edi, D_04338
883     mov word [edi + 0x1c], 0x4f05
884     mov word [edi + 0x10], 0
885     mov word [edi + 0x14], dx
886     mov ax, 0x300
887     mov bx, 0x10
888     db 0x66, 0x33, 0xc9      ; xor cx, cx
889     int 0x31
890     pop edx
891     mov word [edi + 0x1c], 0x4f05
892     mov word [edi + 0x10], 1
893     mov word [edi + 0x14], dx
894     mov ax, 0x300
895     mov bx, 0x10
896     db 0x66, 0x33, 0xc9      ; xor cx, cx
897     int 0x31
```

```
898         pop edx
899         pop edi
900         pop ebx
901         pop eax
902         ret
903     VESA_DispStart_Ret:
904         push eax
905         push ebx
906         push edx
907         push ebp
908         push edx
909         mov bl, 0
910         call dword [vesa_win_granularity]
911         pop edx
912         mov bl, 1
913         call dword [vesa_win_granularity]
914         pop ebp
915         pop edx
916         pop ebx
917         pop eax
918         ret
919
920     ; Blit-pad voor banked (niet-lineaire) framebuffers
921     Blit_Banked_Path:
922         mov edx, 0x3da
923     Blit_Banked_Loop:
924         in al, dx
925         test al, 8
926         jne short Blit_Banked_Loop
927     L_04a5f:
928         in al, dx
929         test al, 8
930         je short L_04a5f
931         ret
932     L_04a65:
933         mov ah, byte [D_0211f]
934         shr ah, 1
935         test ah, ah
936         jne short L_04a86
937         movzx eax, al
938         mov ebx, 0x1f400
939         mul ebx
940         add eax, dword [D_0436a]
941         mov dword [ptr_framebuffer_src], eax
942     L_04a86:
943         ret
944         db 0x90
945     D_04a88:
946         db 0x00, 0x00, 0x00, 0x00
947     Blit_Or_Present:
948         mov ah, byte [D_0211f]
949         shr ah, 1
950         test ah, ah
951         je short L_04ab9
952         cmp ah, 1
953         je short L_04ad5
954         cmp ah, 2
955         je near L_04bca
956         cmp ah, 3
957         je near Blit_Dispatch
958         cmp ah, 4
959         je near Blit_FullCopy_Dispatch
960         ret
961     L_04ab9:
962         mov ecx, 0x280
963         movzx eax, al
964         mov ebx, 0xc8
965         mul ebx
966         db 0x8b, 0xd8     ; mov ebx, eax
```

```
967         mov eax, 0
968         call VESA_SetBank
969         ret
970     L_04ad5:
971         cmp byte [vesa_mode_flags], 0x80
972         je short L_04aee
973         mov eax, 0
974         db 0x8b, 0xc8      ; mov ecx, eax
975         db 0x8b, 0xd8      ; mov ebx, eax
976         call VESA_SetBank
977         jmp short L_04af3
978     L_04aee:
979         call Blit_Banked_Path
980     L_04af3:
981         mov esi, dword [ptr_framebuffer_src]
982         mov ebp, dword [screen_width]
983         sub ebp, 0x280
984         mov ebx, 0xc8
985         mov dword [D_04a88], 0xffffffff
986     L_04b14:
987         inc dword [D_04a88]
988         mov edx, dword [D_04a88]
989         call VESA_SetDisplayStart
990         mov edx, dword [screen_bytes_per_scan]
991         mov edi, dword [linear_framebuffer]
992         xchg ebx, ebx
993         nop
994     L_04b34:
995         mov ecx, 0xa0
996         rep movsd
997         dec ebx
998         je near L_04bc9
999         db 0x03, 0xfd      ; add edi, ebp
1000        sub edx, dword [screen_width]
1001        je short L_04b14
1002        cmp edx, dword [screen_width]
1003        jae short L_04b34
1004        cmp edx, 0x280
1005        jae short L_04b94
1006        db 0x8b, 0xca      ; mov ecx, edx
1007        rep movsb
1008        push edx
1009        inc dword [D_04a88]
1010        mov edx, dword [D_04a88]
1011        call VESA_SetDisplayStart
1012        mov edi, dword [linear_framebuffer]
1013        pop edx
1014        mov ecx, 0x280
1015        db 0x2b, 0xca      ; sub ecx, edx
1016        add edx, dword [screen_bytes_per_scan]
1017        sub edx, dword [screen_width]
1018        rep movsb
1019        db 0x03, 0xfd      ; add edi, ebp
1020        dec ebx
1021        jne short L_04b34
1022        ret
1023    L_04b94:
1024        mov ecx, 0xa0
1025        rep movsd
1026        sub edx, dword [screen_width]
1027        push edx
1028        inc dword [D_04a88]
1029        mov edx, dword [D_04a88]
1030        call VESA_SetDisplayStart
1031        mov edi, dword [linear_framebuffer]
1032        pop edx
1033        db 0x2b, 0xfa      ; sub edi, edx
1034        add edx, dword [screen_bytes_per_scan]
1035        dec ebx
```

```
1036        jne near L_04b34
1037    L_04bc9:
1038        ret
1039    L_04bca:
1040        cmp byte [vesa_mode_flags], 0x80
1041        je short L_04be3
1042        mov eax, 0
1043        db 0x8b, 0xc8     ; mov ecx, eax
1044        db 0x8b, 0xd8     ; mov ebx, eax
1045        call VESA_SetBank
1046        jmp short L_04be8
1047    L_04be3:
1048        call Blit_Banked_Path
1049    L_04be8:
1050        mov esi, dword [ptr_framebuffer_src]
1051        mov edi, dword [D_0436a]
1052        cmp dword [screen_height], 0x1e0
1053        jne short Blit_Setup
1054
1055    ; ================================================================================
1056    ; [4] Pixels naar het scherm kopiëren ("blit")
1057    ;
1058    ; De demo rekent effecten uit in een buffer in gewoon geheugen.
1059    ; Deze routines kopiëren (delen van) die buffer naar het echte
1060    ; videogeheugen, zodat je het resultaat ziet.
1061    ;
1062    ; Vaak gebeurt dat per scanline (beeldlijn) of in blokken van
1063    ; 8 bytes tegelijk voor snelheid.
1064    ; ================================================================================
1065    ; ================================================================================
1066    ; [4] Pixels naar het scherm kopiëren ("blit")
1067    ;
1068    ; De demo rekent effecten uit in een buffer in gewoon geheugen.
1069    ; Deze routines kopiëren (delen van) die buffer naar het echte
1070    ; videogeheugen, zodat je het resultaat ziet.
1071    ;
1072    ; Vaak gebeurt dat per scanline (beeldlijn) of in blokken van
1073    ; 8 bytes tegelijk voor snelheid.
1074    ; ================================================================================
1075    Blit_AddYOffset:
1076        mov eax, dword [screen_width]
1077        mov ebx, 0x28
1078        mul ebx
1079        db 0x03, 0xf8     ; add edi, eax
1080    Blit_Setup:
1081        mov ebx, 0xc8
1082        mov edx, dword [screen_width]
1083        db 0x8d, 0x2c, 0x55, 0x00, 0x00, 0x00, 0x00     ; lea ebp, [edx*2]
1084        sub ebp, 0x500
1085        xchg ebx, ebx
1086    Blit_RowLoop:
1087        mov ecx, 0xa0
1088        push ebx
1089        xchg ebx, ebx
1090    Blit_PixelPair:
1091        mov eax, dword [esi]
1092        db 0x8b, 0xd8     ; mov ebx, eax
1093        rol eax, 0x10
1094        xchg bx, ax
1095        mov dword [edi], eax
1096        mov dword [edi + 4], ebx
1097        mov dword [edi + edx], eax
1098        mov dword [edi + edx + 4], ebx
1099        add edi, 8
1100        add esi, 4
1101        dec ecx
1102        jne short Blit_PixelPair
1103        pop ebx
1104        mov ecx, 0xa0
```

```
1105        db 0x03, 0xfd     ; add edi, ebp
1106        dec ebx
1107        jne short Blit_RowLoop
1108        ret
1109    D_04c5a:
1110        db 0x01
1111    Blit_Dispatch:
1112        cmp byte [vesa_mode_flags], 0x80
1113        je short Blit_AltPath
1114        mov eax, 0
1115        db 0x8b, 0xc8     ; mov ecx, eax
1116        db 0x8b, 0xd8     ; mov ebx, eax
1117        call VESA_SetBank
1118        jmp short Blit_Main
1119    Blit_AltPath:
1120        call Blit_Banked_Path
1121    Blit_Main:
1122        mov esi, dword [ptr_framebuffer_src]
1123        mov ebp, dword [screen_width]
1124        sub ebp, 0x500
1125        mov ebx, dword [screen_height]
1126        mov dword [D_04a88], 0xffffffff
1127        mov byte [D_04c5a], 1
1128        mov ebx, 0x190
1129        mov dword [D_04a88], 0
1130        mov byte [D_04c5a], 1
1131        mov edx, dword [D_04a88]
1132        call VESA_SetDisplayStart
1133        mov edx, dword [screen_bytes_per_scan]
1134        mov edi, dword [linear_framebuffer]
1135        cmp dword [screen_height], 0x1e0
1136        jne short Blit_AfterAdjust
1137        push ebx
1138        push edx
1139        mov eax, dword [screen_width]
1140        mov ebx, 0x28
1141        mul ebx
1142        pop edx
1143        pop ebx
1144        db 0x03, 0xf8     ; add edi, eax
1145        db 0x2b, 0xd0     ; sub edx, eax
1146        jmp short Blit_AfterAdjust
1147    Blit_Cont:
1148        inc dword [D_04a88]
1149        mov edx, dword [D_04a88]
1150        call VESA_SetDisplayStart
1151        mov edx, dword [screen_bytes_per_scan]
1152        mov edi, dword [linear_framebuffer]
1153    Blit_AfterAdjust:
1154        cmp byte [D_04c5a], 0
1155        jne short L_04d4d
1156        mov ecx, 0xa0
1157        push ebx
1158        xor byte [D_04c5a], 1
1159        sub esi, 0x280
1160        xchg ebx, ebx
1161    L_04d2c:
1162        mov eax, dword [esi]
1163        db 0x8b, 0xd8     ; mov ebx, eax
1164        rol eax, 0x10
1165        xchg bx, ax
1166        mov dword [edi], eax
1167        mov dword [edi + 4], ebx
1168        add edi, 8
1169        add esi, 4
1170        dec ecx
1171        jne short L_04d2c
1172        pop ebx
1173        dec ebx
```

```
1174        db 0x03, 0xfd      ; add edi, ebp
1175        sub edx, dword [screen_width]
1176    L_04d4d:
1177        db 0x8b, 0xc2      ; mov eax, edx
1178        mov edx, 0
1179        mov ecx, dword [screen_width]
1180        db 0x03, 0xc9      ; add ecx, ecx
1181        div ecx
1182        db 0x03, 0xc0      ; add eax, eax
1183        db 0x3b, 0xd8      ; cmp ebx, eax
1184        jae short L_04d69
1185        db 0x8b, 0xc3      ; mov eax, ebx
1186        and eax, 0xfffffffe
1187    L_04d69:
1188        db 0x2b, 0xd8      ; sub ebx, eax
1189        shr eax, 1
1190        push ebx
1191        push edx
1192        db 0x8b, 0xd8      ; mov ebx, eax
1193        mov edx, dword [screen_width]
1194        mov ecx, 0xa0
1195    L_04d7c:
1196        push ebx
1197        xchg ebx, ebx
1198        nop
1199    L_04d80:
1200        mov eax, dword [esi]
1201        db 0x8b, 0xd8      ; mov ebx, eax
1202        rol eax, 0x10
1203        xchg bx, ax
1204        mov dword [edi], eax
1205        mov dword [edi + 4], ebx
1206        mov dword [edi + edx], eax
1207        mov dword [edi + edx + 4], ebx
1208        add edi, 8
1209        add esi, 4
1210        dec ecx
1211        jne short L_04d80
1212        mov ecx, 0xa0
1213        db 0x03, 0xfd      ; add edi, ebp
1214        db 0x03, 0xfa      ; add edi, edx
1215        pop ebx
1216        dec ebx
1217        jne short L_04d7c
1218        pop edx
1219        pop ebx
1220        test ebx, ebx
1221        je near Blit_FullCopy_Ret
1222        test edx, edx
1223        je near Blit_Cont
1224        cmp edx, dword [screen_width]
1225        jb short L_04e01
1226        mov ecx, 0xa0
1227        push ebx
1228        xor byte [D_04c5a], 1
1229        xchg ebx, ebx
1230    L_04dd4:
1231        mov eax, dword [esi]
1232        db 0x8b, 0xd8      ; mov ebx, eax
1233        rol eax, 0x10
1234        xchg bx, ax
1235        mov dword [edi], eax
1236        mov dword [edi + 4], ebx
1237        add edi, 8
1238        add esi, 4
1239        dec ecx
1240        jne short L_04dd4
1241        pop ebx
1242        dec ebx
```

```
1243        je near Blit_FullCopy_Ret
1244        db 0x03, 0xfd      ; add edi, ebp
1245        sub edx, dword [screen_width]
1246        je near Blit_Cont
1247    L_04e01:
1248        cmp edx, 0x500
1249        jae near L_04ebe
1250        db 0x8b, 0xca      ; mov ecx, edx
1251        shr ecx, 2
1252        push ebx
1253        xor byte [D_04c5a], 1
1254        je short L_04e22
1255        sub esi, 0x280
1256    L_04e22:
1257        shr ecx, 1
1258        jae short L_04e3c
1259        mov eax, dword [esi]
1260        mov ebx, dword [eax]
1261        rol eax, 0x10
1262        db 0x66, 0x8b, 0xc3      ; mov ax, bx
1263        mov dword [edi], eax
1264        add edi, 4
1265        add esi, 2
1266        jecxz L_04e53
1267        xchg ebx, ebx
1268    L_04e3c:
1269        mov eax, dword [esi]
1270        db 0x8b, 0xd8      ; mov ebx, eax
1271        rol eax, 0x10
1272        xchg bx, ax
1273        mov dword [edi], eax
1274        mov dword [edi + 4], ebx
1275        add edi, 8
1276        add esi, 4
1277        dec ecx
1278        jne short L_04e3c
1279    L_04e53:
1280        push edx
1281        inc dword [D_04a88]
1282        mov edx, dword [D_04a88]
1283        call VESA_SetDisplayStart
1284        mov edi, dword [linear_framebuffer]
1285        pop edx
1286        mov ecx, 0x500
1287        db 0x2b, 0xca      ; sub ecx, edx
1288        add edx, dword [screen_bytes_per_scan]
1289        sub edx, dword [screen_width]
1290        shr ecx, 2
1291        shr ecx, 1
1292        jae short L_04e9c
1293        mov eax, dword [esi]
1294        mov ebx, dword [eax]
1295        rol eax, 0x10
1296        db 0x66, 0x8b, 0xc3      ; mov ax, bx
1297        mov edi, dword [eax]
1298        add edi, 4
1299        add esi, 2
1300        jecxz L_04eb3
1301        xchg ebx, ebx
1302    L_04e9c:
1303        mov eax, dword [esi]
1304        db 0x8b, 0xd8      ; mov ebx, eax
1305        rol eax, 0x10
1306        xchg bx, ax
1307        mov dword [edi], eax
1308        mov dword [edi + 4], ebx
1309        add edi, 8
1310        add esi, 4
1311        dec ecx
```

```
    jne short L_04e9c
L_04eb3:
    db 0x03, 0xfd     ; add edi, ebp
    pop ebx
    dec ebx
    jne near Blit_AfterAdjust
    ret
L_04ebe:
    mov ecx, 0xa0
    push ebx
    xor byte [D_04c5a], 1
    je short L_04ed4
    sub esi, 0x280
    nop
L_04ed4:
    mov eax, dword [esi]
    db 0x8b, 0xd8     ; mov ebx, eax
    rol eax, 0x10
    xchg bx, ax
    mov dword [edi], eax
    mov dword [edi + 4], ebx
    add edi, 8
    add esi, 4
    dec ecx
    jne short L_04ed4
    sub edx, dword [screen_width]
    push edx
    inc dword [D_04a88]
    mov edx, dword [D_04a88]
    call VESA_SetDisplayStart
    mov edi, dword [linear_framebuffer]
    pop edx
    db 0x2b, 0xfa     ; sub edi, edx
    add edx, dword [screen_bytes_per_scan]
    pop ebx
    dec ebx
    jne near Blit_AfterAdjust
Blit_FullCopy_Ret:
    ret

; ---------------------------------------------------------------------------
; Blit_FullCopy - hele buffer in één keer naar het scherm
;
; Linear: snelle rep movsd.
; Banked: via Blit_Banked_Path / bank-wissel.
; ---------------------------------------------------------------------------
Blit_FullCopy_Dispatch:
    cmp byte [vesa_mode_flags], 0x80
    je short Blit_FullCopy_Banked
    mov eax, 0
    db 0x8b, 0xc8     ; mov ecx, eax
    db 0x8b, 0xd8     ; mov ebx, eax
    call VESA_SetBank
    jmp short Blit_FullCopy_Linear
Blit_FullCopy_Banked:
    call Blit_Banked_Path
Blit_FullCopy_Linear:
    mov edi, dword [D_0436a]
    mov esi, dword [ptr_framebuffer_src]
    mov ecx, L_07d00
    rep movsd
    ret

; ---------------------------------------------------------------------------
; [5d] Videomodus-helpers (resolutie + buffer)
;
; Zetten breedte/hoogte (bijv. 320×200 = 0x140 × 0xC8), koppelen een
; framebuffer-pointer, en vragen zo nodig extra DOS32-geheugen aan.
; flag_020e4 onthoudt welke modus actief is (1 / 2 / 3).
```

```
; ------------------------------------------------------------------------
Video_SetMode_320x200:
    mov dword [ptr_020f4], 0x140
    mov dword [D_020f8], 0xc8
    and byte [D_0211f], 7
    mov byte [flag_020e6], al
    mov ebx, dword [D_0436a]
    mov dword [ptr_framebuffer_src], ebx
    cmp al, 1
    je short Video_Alloc_320x200
    test byte [D_0211f], 6
    je short Video_SetMode_320_cont
Video_Alloc_320x200:
    mov edx, 0x26400
    call Dos32_GetSelectorBase
    jb near Dos_Terminate
    mov dword [ptr_framebuffer_src], edx
    test byte [D_0211f], 6
    jne short Video_SetMode_320_cont
    or byte [D_0211f], 8
Video_SetMode_320_cont:
    cmp ah, 1
    je short Video_Mode_320_done
    mov ax, word [D_02116]
    call VESA_ClearInfoBlock
Video_Mode_320_done:
    mov byte [flag_020e4], 1
    ret
; Video_SetMode variant 2 - bredere buffer (0x320 × 0x258)
    mov dword [ptr_020f4], 0x320
    mov dword [D_020f8], 0x258
    mov byte [flag_020e6], al
    mov ebx, dword [D_0436a]
    mov dword [ptr_020ec], ebx
    cmp al, 1
    je short L_04ff8
    test word [D_02128], 0x4000
    jne short L_0500e
L_04ff8:
    mov edx, 0xec540
    call Dos32_GetSelectorBase
    jb near Dos_Terminate
    mov dword [ptr_020ec], edx
L_0500e:
    cmp ah, 1
    je short L_0501e
    mov ax, word [D_02128]
    call VESA_ClearInfoBlock
L_0501e:
    mov byte [flag_020e4], 2
    ret
; Video_SetMode variant 3 - 640×480-achtig (0x280 × 0x1e0)
    mov dword [ptr_020f4], 0x280
    mov dword [D_020f8], 0x1e0
    mov byte [flag_020e6], al
    mov ebx, dword [D_0436a]
    mov dword [ptr_020f0], ebx
    cmp al, 1
    je short L_0505a
    test word [D_02104], 0x4000
    jne short L_05070
L_0505a:
    mov edx, 0x4bc80
    call Dos32_GetSelectorBase
    jb near Dos_Terminate
    mov dword [ptr_020f0], edx
L_05070:
    cmp ah, 1
    je short L_05080
```

```
1450        mov ax, word [D_02104]
1451        call VESA_ClearInfoBlock
1452    L_05080:
1453        mov byte [flag_020e4], 3
1454        ret
1455
1456    ; ---------------------------------------------------------------------------
1457    ; [5a] Script-interpreter (scène / effecten)
1458    ;
1459    ; Lees één opcode-byte uit [esi], kies bijbehorende routine, voer uit,
1460    ; ga door tot einde. Typische opdrachten: parameters laden, effect starten,
1461    ; floating-point constanten zetten, buffers vullen, …
1462    ; ---------------------------------------------------------------------------
1463    Script_PushEsi:
1464        push esi
1465    Script_Next:
1466        pop esi
1467    Script_Dispatch_Opcode:
1468        ; EBX = opcode (0..0x0E). Daarboven → Script_End.
1469        movzx ebx, byte [esi]
1470        inc esi
1471        cmp ebx, 0xe
1472        ja near Script_End
1473        mov edi, dword [ebx*4 + Script_JumpTable]
1474        movzx eax, byte [esi]
1475        jmp edi
1476    ; Opcode → handler (o.a. Mesh_BuildGrid_*, Object_Translate, rotaties, …)
1477    Script_JumpTable:
1478        db 0xdf, 0x50, 0x00, 0x00, 0xee, 0x51, 0x00, 0x00, 0xfc, 0x51, 0x00, 0x00, 0x53, 0x52
            , 0x00, 0x00
1479        db 0x19, 0x51, 0x00, 0x00, 0x7b, 0x51, 0x00, 0x00, 0x93, 0x51, 0x00, 0x00, 0xa4, 0x51
            , 0x00, 0x00
1480        db 0xbb, 0x51, 0x00, 0x00, 0xcc, 0x51, 0x00, 0x00, 0xdd, 0x51, 0x00, 0x00, 0xaf, 0x51
            , 0x00, 0x00
1481        db 0x2d, 0x52, 0x00, 0x00, 0x48, 0x52, 0x00, 0x00, 0x53, 0x51, 0x00, 0x00
1482        mov dword [D_05587], eax
1483        db 0x03, 0xc0       ; add eax, eax
1484        mov dword [D_05583], eax
1485        inc esi
1486        mov eax, dword [esi]
1487        mov dword [D_0553b], eax
1488        mov eax, dword [esi + 4]
1489        mov dword [D_0553f], eax
1490        mov eax, dword [esi + 8]
1491        mov dword [D_05543], eax
1492        mov eax, dword [esi + 0xc]
1493        mov dword [D_0556b], eax
1494        add esi, 0x10
1495        push esi
1496        call Mesh_BuildGrid_A
1497        jmp near Script_Next
1498        mov dword [D_05583], eax
1499        movzx eax, byte [esi + 1]
1500        mov dword [D_05587], eax
1501        add esi, 2
1502        mov edi, D_0553b
1503        mov ecx, 6
1504        rep movsd
1505        mov eax, dword [esi]
1506        mov dword [D_0556b], eax
1507        mov eax, dword [esi + 4]
1508        mov dword [D_0556f], eax
1509        add esi, 8
1510        push esi
1511        call Mesh_BuildGrid_B
1512        jmp near Script_Next
1513        mov dword [D_05583], eax
1514        movzx eax, byte [esi + 1]
1515        mov dword [D_05587], eax
```

```
1516        mov edi, D_0556b
1517        add esi, 2
1518        mov ecx, 6
1519        rep movsd
1520        push esi
1521        call Mesh_BuildGrid_C
1522        jmp near Script_Next
1523        inc esi
1524        mov edi, D_05547
1525        mov ecx, 3
1526        rep movsd
1527        push esi
1528        call Object_Translate
1529        jmp near Script_Next
1530        mov ebx, dword [esi + 1]
1531        add esi, 5
1532        push esi
1533        call Object_RotateAroundY
1534        jmp near Script_Next
1535        push esi
1536        call Object_MergeLastTwo
1537        jmp near Script_Next
1538        inc esi
1539        push esi
1540        call Object_Duplicate
1541        jmp near Script_Next
1542        mov ebx, dword [esi + 1]
1543        add esi, 5
1544        push esi
1545        call Matrix_RotateX
1546        jmp near Script_Next
1547        mov ebx, dword [esi + 1]
1548        add esi, 5
1549        push esi
1550        call Matrix_RotateY
1551        jmp near Script_Next
1552        mov ebx, dword [esi + 1]
1553        add esi, 5
1554        push esi
1555        call Matrix_RotateZ
1556        jmp near Script_Next
1557        movzx eax, byte [esi]
1558        inc esi
1559        mov dword [D_0558b], eax
1560        jmp near Script_Dispatch_Opcode
1561        movzx eax, byte [esi]
1562        mov dword [D_02000], eax
1563        fild dword [D_02000]
1564        fstp dword [D_05593]
1565        movzx eax, byte [esi + 1]
1566        mov dword [D_02000], eax
1567        fild dword [D_02000]
1568        fstp dword [D_05597]
1569        add esi, 2
1570        jmp near Script_Dispatch_Opcode
1571        push esi
1572        mov eax, 0
1573    L_05233:
1574        push eax
1575        call D_233d0
1576        pop eax
1577        inc eax
1578        cmp eax, dword [D_1bd0c]
1579        jb short L_05233
1580        jmp near Script_Next
1581        inc esi
1582        mov dword [D_0558f], eax
1583        jmp near Script_Dispatch_Opcode
1584    Script_End:
```

```
1585        ret
1586    Scene_ResetState:
1587        mov eax, dword [Heap_Base]
1588        mov dword [D_1bd00], eax
1589        add eax, 0xea600
1590        mov dword [D_1bcfc], eax
1591        add eax, 0x88b80
1592        mov dword [D_1bd04], eax
1593        add eax, 0xfa0
1594        mov dword [D_1bcf4], eax
1595        add eax, 0x27100
1596        mov dword [D_1bcf8], eax
1597        add eax, 0x27100
1598        mov dword [D_0551e], eax
1599        add eax, 0x8000
1600        mov edi, dword [Heap_Base]
1601        db 0x8b, 0xc8     ; mov ecx, eax
1602        db 0x2b, 0xcf     ; sub ecx, edi
1603        db 0x32, 0xc0     ; xor al, al
1604        rep stosb
1605        mov dword [D_1bd14], 0
1606        mov dword [D_1bd10], 0
1607        mov dword [D_1bd0c], 0
1608        mov eax, 0x40
1609        mov dword [D_0638e], eax
1610        mov dword [D_0558f], 3
1611        mov dword [D_0558b], 1
1612        fld1
1613        fst dword [D_05593]
1614        fstp dword [D_05597]
1615        ret
1616    Heap_Base:
1617        db 0x00, 0x00, 0x00, 0x00
1618
1619    ; ------------------------------------------------------------------------
1620    ; Object_Alloc - nieuw 3D-object registreren
1621    ;
1622    ; Zet een entry klaar met:
1623    ;   - pointer naar vertex-buffer
1624    ;   - pointer naar triangle-lijst
1625    ;   - aantallen (eax/ebx-parameters)
1626    ; Wordt aangeroepen door mesh-generators en duplicate.
1627    ; ------------------------------------------------------------------------
1628    Object_Alloc:
1629        push eax
1630        push ebx
1631        mov eax, dword [D_1bd0c]
1632        mov ebx, 0x50
1633        mul ebx
1634        db 0x8b, 0xf8     ; mov edi, eax
1635        add edi, dword [D_1bd04]
1636        mov dword [D_063e2], edi
1637        mov eax, dword [D_1bd10]
1638        mov dword [edi + 8], eax
1639        mov ebx, 0x40
1640        mul ebx
1641        add eax, dword [D_1bd00]
1642        mov dword [D_063da], eax
1643        mov eax, dword [D_1bd14]
1644        mov dword [edi], eax
1645        mov ebx, 0x1c
1646        mul ebx
1647        add eax, dword [D_1bcfc]
1648        mov dword [D_063de], eax
1649        pop ebx
1650        pop eax
1651        mov dword [edi + 0xc], eax
1652        mov dword [edi + 4], ebx
1653        add dword [D_1bd10], eax
```

```
    add dword [D_1bd14], ebx
    mov dword [edi + 0x4c], 0
    inc dword [D_1bd0c]
    ret

; ------------------------------------------------------------------------
; Triangle_FixUVWrap - UV-coördinaten "wrappen"
;
; Textuurcoördinaten kunnen buiten [0..1] vallen. Met fprem (drijvende
; modulo) worden ze teruggevouwen, zodat de texture naadloos herhaalt.
; Als een rand te groot is, wordt een correctie + vlag gezet.
; ------------------------------------------------------------------------
Triangle_FixUVWrap:
    pushad
    mov eax, dword [D_0551e]
    fld dword [D_05516]
    mov esi, dword [edi]
    fld dword [esi + 0x2c]
    fprem
    fstp dword [eax]
    fld dword [esi + 0x30]
    fprem
    fstp dword [eax + 4]
    mov esi, dword [edi + 4]
    fld dword [esi + 0x2c]
    fprem
    fstp dword [eax + 8]
    fld dword [esi + 0x30]
    fprem
    fstp dword [eax + 0xc]
    mov esi, dword [edi + 8]
    fld dword [esi + 0x2c]
    fprem
    fstp dword [eax + 0x10]
    fld dword [esi + 0x30]
    fprem
    fstp dword [eax + 0x14]
    faddp st0
    fld dword [eax]
    fsub dword [eax + 8]
    fst dword [D_05512]
    fabs
    fistp dword [D_063d6]
    cmp dword [D_063d6], 0x800000
    jb short Triangle_UV_Edge1
    mov esi, dword [D_05512]
    shr esi, 0x1f
    fld dword [eax + 8]
    fadd dword [esi*4 + D_05516]
    fstp dword [eax + 8]
    or byte [edi + 0xc], 0x80
Triangle_UV_Edge1:
    fld dword [eax + 8]
    fsub dword [eax + 0x10]
    fst dword [D_05512]
    fabs
    fistp dword [D_063d6]
    cmp dword [D_063d6], 0x800000
    jb short Triangle_UV_Edge2
    mov esi, dword [D_05512]
    shr esi, 0x1f
    fld dword [eax + 0x10]
    fadd dword [esi*4 + D_05516]
    fstp dword [eax + 0x10]
    or byte [edi + 0xc], 0x80
Triangle_UV_Edge2:
    fld dword [eax + 0x10]
    fsub dword [eax]
    fst dword [D_05512]
```

```
1723        fabs
1724        fistp dword [D_063d6]
1725        cmp dword [D_063d6], 0x800000
1726        jb short Triangle_UV_Edge3
1727        mov esi, dword [D_05512]
1728        shr esi, 0x1f
1729        fld dword [eax]
1730        fadd dword [esi*4 + D_05516]
1731        fstp dword [eax]
1732        or byte [edi + 0xc], 0x80
1733    Triangle_UV_Edge3:
1734        fld dword [eax + 4]
1735        fsub dword [eax + 0xc]
1736        fst dword [D_05512]
1737        fabs
1738        fistp dword [D_063d6]
1739        cmp dword [D_063d6], 0x800000
1740        jb short Triangle_UV_Edge4
1741        mov esi, dword [D_05512]
1742        shr esi, 0x1f
1743        fld dword [eax + 0xc]
1744        fadd dword [esi*4 + D_05516]
1745        fstp dword [eax + 0xc]
1746        or byte [edi + 0xc], 0x80
1747    Triangle_UV_Edge4:
1748        fld dword [eax + 0xc]
1749        fsub dword [eax + 0x14]
1750        fst dword [D_05512]
1751        fabs
1752        fistp dword [D_063d6]
1753        cmp dword [D_063d6], 0x800000
1754        jb short Triangle_UV_Edge5
1755        mov esi, dword [D_05512]
1756        shr esi, 0x1f
1757        fld dword [eax + 0x14]
1758        fadd dword [esi*4 + D_05516]
1759        fstp dword [eax + 0x14]
1760        or byte [edi + 0xc], 0x80
1761    Triangle_UV_Edge5:
1762        fld dword [eax + 0x14]
1763        fsub dword [eax + 4]
1764        fst dword [D_05512]
1765        fabs
1766        fistp dword [D_063d6]
1767        cmp dword [D_063d6], 0x800000
1768        jb short Triangle_UV_Edge6
1769        mov esi, dword [D_05512]
1770        shr esi, 0x1f
1771        fld dword [eax + 4]
1772        fadd dword [esi*4 + D_05516]
1773        fstp dword [eax + 4]
1774        or byte [edi + 0xc], 0x80
1775    Triangle_UV_Edge6:
1776        test byte [edi + 0xc], 0x80
1777        je short Triangle_UV_Done
1778        mov dword [edi + 0x10], eax
1779        add dword [D_0551e], 0x30
1780    Triangle_UV_Done:
1781        popad
1782        ret
1783    D_05512:
1784        db 0x00, 0x00, 0x00, 0x00
1785    D_05516:
1786        db 0x00, 0x00, 0x80, 0x4b, 0x00, 0x00, 0x80, 0xcb
1787    D_0551e:
1788        db 0x00, 0x00, 0x00, 0x00
1789
1790    ; ---------------------------------------------------------------------------
1791    ; Triangle_SetupEdges - randen van een driehoek voorbereiden
```

```
;
; Berekent hulpwaarden per edge voor later rasterizen / texturen.
; Wordt vaak aangeroepen vanuit de mesh-bouw-lussen.
; ------------------------------------------------------------------------
Triangle_SetupEdges:
    mov byte [edi + 0xc], 0
    mov eax, dword [D_0558f]
    mov byte [edi + 0xd], al
    mov eax, dword [D_0558b]
    mov byte [edi + 0xe], al
    mov byte [edi + 0xf], 0
    ret
D_0553b:
    db 0x00, 0x00, 0x00, 0x00
D_0553f:
    db 0x00, 0x00, 0x00, 0x00
D_05543:
    db 0x00, 0x00, 0x00, 0x00
D_05547:
    db 0x00, 0x00, 0x00, 0x00
D_0554b:
    db 0x00, 0x00, 0x00, 0x00
D_0554f:
    times 28 db 0
D_0556b:
    db 0x00, 0x00, 0x00, 0x00
D_0556f:
    db 0x00, 0x00, 0x00, 0x00
D_05573:
    db 0x00, 0x00, 0x00, 0x00
D_05577:
    db 0x00, 0x00, 0x00, 0x00
D_0557b:
    db 0x00, 0x00, 0x00, 0x00
D_0557f:
    db 0x00, 0x00, 0x00, 0x00
D_05583:
    db 0x00, 0x00, 0x00, 0x00
D_05587:
    db 0x00, 0x00, 0x00, 0x00
D_0558b:
    db 0x00, 0x00, 0x00, 0x00
D_0558f:
    db 0x00, 0x00, 0x00, 0x00
D_05593:
    db 0x00, 0x00, 0x00, 0x00
D_05597:
    db 0x00, 0x00, 0x00, 0x00

; ------------------------------------------------------------------------
; Mesh_BuildGrid_A/B/C - 3D-roosters (grids) opbouwen
;
; Script-opcodes roepen deze aan om een netwerk van vertices + triangles
; te maken (bijv. een vlak of cilinder-achtig grid).
; D_05583 / D_05587 = afmetingen van het rooster.
; Daarna volgen geneste lussen die driehoeken aan elkaar knopen en
; Triangle_SetupEdges / Triangle_FixUVWrap aanroepen.
; ------------------------------------------------------------------------
Mesh_BuildGrid_A:
    mov eax, dword [D_05587]
    dec eax
    mul dword [D_05583]
    db 0x03, 0xc0     ; add eax, eax
    db 0x8b, 0xd8     ; mov ebx, eax
    mov eax, dword [D_05587]
    inc eax
    mul dword [D_05583]
    call Object_Alloc
    mov edi, dword [D_063da]
```

```
1861        fldpi
1862        fadd st0
1863        fidiv dword [D_05583]
1864        fstp dword [D_063ee]
1865        fldpi
1866        fidiv dword [D_05587]
1867        fstp dword [D_063f2]
1868        mov eax, dword [D_05587]
1869        inc eax
1870        mov dword [D_063fa], eax
1871        mov dword [D_063ea], 0
1872    L_055f5:
1873        mov eax, dword [D_05583]
1874        mov dword [D_063f6], eax
1875        mov dword [D_063e6], 0
1876    L_05609:
1877        fld dword [D_063ea]
1878        fsin
1879        fld dword [D_063e6]
1880        fcos
1881        fmulp st1
1882        fmul dword [D_0556b]
1883        fadd dword [D_0553b]
1884        fstp dword [edi]
1885        fld dword [D_063ea]
1886        fcos
1887        fmul dword [D_0556b]
1888        fadd dword [D_0553f]
1889        fstp dword [edi + 4]
1890        fld dword [D_063ea]
1891        fsin
1892        fld dword [D_063e6]
1893        fsin
1894        fmulp st1
1895        fmul dword [D_0556b]
1896        fadd dword [D_05543]
1897        fstp dword [edi + 8]
1898        fld dword [D_063e6]
1899        fmul dword [D_0638a]
1900        fmul dword [D_0637e]
1901        fmul dword [D_05593]
1902        fstp dword [edi + 0x2c]
1903        fld dword [D_063ea]
1904        fmul dword [D_06386]
1905        fmul dword [D_0637e]
1906        fmul dword [D_05597]
1907        fstp dword [edi + 0x30]
1908        mov al, byte [D_0558f]
1909        shr al, 1
1910        mov byte [edi + 0x3c], al
1911        mov dword [edi + 0x20], 0
1912        mov dword [edi + 0x24], 0
1913        mov dword [edi + 0x28], 0
1914        add edi, 0x40
1915        fld dword [D_063e6]
1916        fadd dword [D_063ee]
1917        fstp dword [D_063e6]
1918        dec dword [D_063f6]
1919        jne near L_05609
1920        fld dword [D_063ea]
1921        fadd dword [D_063f2]
1922        fstp dword [D_063ea]
1923        dec dword [D_063fa]
1924        jne near L_055f5
1925        mov edi, dword [D_063de]
1926        mov eax, dword [D_05583]
1927        mov dword [D_063d2], eax
1928        mul dword [D_05587]
1929        mov dword [D_063fe], eax
```

```
1930        mov dword [D_063ce], 0
1931    L_0571a:
1932        mov dword [D_063c6], 0
1933        mov dword [D_063ca], 1
1934    L_0572e:
1935        mov eax, dword [D_063ce]
1936        mov ebx, dword [D_063d2]
1937        cmp ebx, dword [D_063fe]
1938        jae short L_0579f
1939        add eax, dword [D_063c6]
1940        mul dword [D_0638e]
1941        add eax, dword [D_063da]
1942        mov dword [edi], eax
1943        mov eax, dword [D_063d2]
1944        add eax, dword [D_063ca]
1945        mul dword [D_0638e]
1946        add eax, dword [D_063da]
1947        mov dword [edi + 4], eax
1948        mov eax, dword [D_063d2]
1949        add eax, dword [D_063c6]
1950        mul dword [D_0638e]
1951        add eax, dword [D_063da]
1952        mov dword [edi + 8], eax
1953        call Triangle_SetupEdges
1954        cmp dword [D_063ca], 0
1955        jne short L_0579c
1956        call Triangle_FixUVWrap
1957    L_0579c:
1958        add edi, 0x1c
1959    L_0579f:
1960        mov eax, dword [D_063ce]
1961        cmp eax, 0
1962        je short L_05807
1963        add eax, dword [D_063c6]
1964        mul dword [D_0638e]
1965        add eax, dword [D_063da]
1966        mov dword [edi], eax
1967        mov eax, dword [D_063ce]
1968        add eax, dword [D_063ca]
1969        mul dword [D_0638e]
1970        add eax, dword [D_063da]
1971        mov dword [edi + 4], eax
1972        mov eax, dword [D_063d2]
1973        add eax, dword [D_063ca]
1974        mul dword [D_0638e]
1975        add eax, dword [D_063da]
1976        mov dword [edi + 8], eax
1977        call Triangle_SetupEdges
1978        cmp dword [D_063ca], 0
1979        jne short L_05804
1980        call Triangle_FixUVWrap
1981    L_05804:
1982        add edi, 0x1c
1983    L_05807:
1984        inc dword [D_063c6]
1985        inc dword [D_063ca]
1986        mov eax, dword [D_063ca]
1987        cmp eax, dword [D_05583]
1988        jne short L_0582a
1989        mov dword [D_063ca], 0
1990    L_0582a:
1991        mov eax, dword [D_063c6]
1992        cmp eax, dword [D_05583]
1993        jb near L_0572e
1994        mov eax, dword [D_05583]
1995        add dword [D_063ce], eax
1996        add dword [D_063d2], eax
1997        mov eax, dword [D_063ce]
1998        cmp eax, dword [D_063fe]
```

```
1999        jb near L_0571a
2000        ret
2001    Mesh_BuildGrid_B:
2002        mov eax, dword [D_05587]
2003        mul dword [D_05583]
2004        db 0x03, 0xc0      ; add eax, eax
2005        db 0x8b, 0xd8      ; mov ebx, eax
2006        mov eax, dword [D_05587]
2007        inc eax
2008        mul dword [D_05583]
2009        call Object_Alloc
2010        mov ebx, D_05547
2011        mov esi, 0x6402
2012        mov edi, 0x5553
2013        call Matrix_SetupFromParams
2014        call Matrix_MulVec
2015        mov esi, D_05547
2016        mov ebx, 0x5553
2017        mov edi, 0x555f
2018        call Matrix_SetupFromParams
2019        call Matrix_MulVec
2020        mov edi, dword [D_063da]
2021        fldpi
2022        fadd st0
2023        fidiv dword [D_05583]
2024        fstp dword [D_063ee]
2025        fld1
2026        fidiv dword [D_05587]
2027        fstp dword [D_063f2]
2028        mov eax, dword [D_05587]
2029        inc eax
2030        mov dword [D_063fa], eax
2031        mov dword [D_063ea], 0
2032    L_058e9:
2033        mov eax, dword [D_05583]
2034        mov dword [D_063f6], eax
2035        mov dword [D_063e6], 0
2036        fld dword [D_0556f]
2037        fsub dword [D_0556b]
2038        fmul dword [D_063ea]
2039        fadd dword [D_0556b]
2040        fstp dword [D_05b4c]
2041    L_0591b:
2042        mov ebx, 8
2043    L_05920:
2044        fld dword [ebx + 0x553b]
2045        fld dword [D_063ea]
2046        fmul dword [ebx + 0x5547]
2047        faddp st1
2048        fld dword [D_063e6]
2049        fcos
2050        fmul dword [ebx + 0x5553]
2051        fld dword [D_063e6]
2052        fsin
2053        fmul dword [ebx + 0x555f]
2054        faddp st1
2055        fmul dword [D_05b4c]
2056        faddp st1
2057        fstp dword [edi + ebx]
2058        sub ebx, 4
2059        jns short L_05920
2060        fld dword [D_063e6]
2061        fmul dword [D_0638a]
2062        fmul dword [D_0637e]
2063        fmul dword [D_05593]
2064        fstp dword [edi + 0x2c]
2065        fld dword [D_063ea]
2066        fmul dword [D_06392]
2067        fmul dword [D_0637e]
```

```
2068        fmul dword [D_05597]
2069        fstp dword [edi + 0x30]
2070        mov al, byte [D_0558f]
2071        shr al, 1
2072        mov byte [edi + 0x3c], al
2073        mov dword [edi + 0x20], 0
2074        mov dword [edi + 0x24], 0
2075        mov dword [edi + 0x28], 0
2076        add edi, 0x40
2077        fld dword [D_063e6]
2078        fadd dword [D_063ee]
2079        fstp dword [D_063e6]
2080        dec dword [D_063f6]
2081        jne near L_0591b
2082        fld dword [D_063ea]
2083        fadd dword [D_063f2]
2084        fstp dword [D_063ea]
2085        dec dword [D_063fa]
2086        jne near L_058e9
2087        mov edi, dword [D_063de]
2088        mov eax, dword [D_05583]
2089        mov dword [D_063d2], eax
2090        mul dword [D_05587]
2091        mov dword [D_063fe], eax
2092        mov dword [D_063ce], 0
2093    L_05a1b:
2094        mov dword [D_063c6], 0
2095        mov dword [D_063ca], 1
2096    L_05a2f:
2097        mov eax, dword [D_063ce]
2098        add eax, dword [D_063c6]
2099        mul dword [D_0638e]
2100        add eax, dword [D_063da]
2101        mov dword [edi], eax
2102        mov eax, dword [D_063d2]
2103        add eax, dword [D_063ca]
2104        mul dword [D_0638e]
2105        add eax, dword [D_063da]
2106        mov dword [edi + 4], eax
2107        mov eax, dword [D_063ce]
2108        add eax, dword [D_063ca]
2109        mul dword [D_0638e]
2110        add eax, dword [D_063da]
2111        mov dword [edi + 8], eax
2112        call Triangle_SetupEdges
2113        cmp dword [D_063ca], 0
2114        jne short L_05a8f
2115        call Triangle_FixUVWrap
2116    L_05a8f:
2117        add edi, 0x1c
2118        mov eax, dword [D_063ce]
2119        add eax, dword [D_063c6]
2120        mul dword [D_0638e]
2121        add eax, dword [D_063da]
2122        mov dword [edi], eax
2123        mov eax, dword [D_063d2]
2124        add eax, dword [D_063c6]
2125        mul dword [D_0638e]
2126        add eax, dword [D_063da]
2127        mov dword [edi + 4], eax
2128        mov eax, dword [D_063d2]
2129        add eax, dword [D_063ca]
2130        mul dword [D_0638e]
2131        add eax, dword [D_063da]
2132        mov dword [edi + 8], eax
2133        call Triangle_SetupEdges
2134        cmp dword [D_063ca], 0
2135        jne short L_05af2
2136        call Triangle_FixUVWrap
```

```
L_05af2:
    add edi, 0x1c
    inc dword [D_063c6]
    inc dword [D_063ca]
    mov eax, dword [D_063ca]
    cmp eax, dword [D_05583]
    jne short L_05b18
    mov dword [D_063ca], 0
L_05b18:
    mov eax, dword [D_063c6]
    cmp eax, dword [D_05583]
    jb near L_05a2f
    mov eax, dword [D_05583]
    add dword [D_063ce], eax
    add dword [D_063d2], eax
    mov eax, dword [D_063ce]
    cmp eax, dword [D_063fe]
    jb near L_05a1b
    ret
D_05b4c:
    db 0x00, 0x00, 0x00, 0x00
Mesh_BuildGrid_C:
    mov eax, dword [D_05583]
    mul dword [D_05587]
    db 0x03, 0xc0     ; add eax, eax
    db 0x8b, 0xd8     ; mov ebx, eax
    db 0x03, 0xdb     ; add ebx, ebx
    call Object_Alloc
    mov edi, dword [D_063da]
    fldpi
    fadd st0
    fld st0
    fidiv dword [D_05583]
    fstp dword [D_063ee]
    fidiv dword [D_05587]
    fstp dword [D_063f2]
    mov eax, dword [D_05587]
    mov dword [D_063fa], eax
    mov dword [D_063ea], 0
L_05b9e:
    mov eax, dword [D_05583]
    mov dword [D_063f6], eax
    mov dword [D_063e6], 0
L_05bb2:
    fld dword [D_0556f]
    fld dword [D_063e6]
    fmul dword [D_05577]
    fsin
    fmul dword [D_05573]
    faddp st1
    fld dword [D_063ea]
    fmul dword [D_0557f]
    fsin
    fmul dword [D_0557b]
    faddp st1
    fst dword [D_063c6]
    fld dword [D_063e6]
    fcos
    fmulp st1
    fadd dword [D_0556b]
    fstp dword [D_063ca]
    fld dword [D_063ea]
    fcos
    fmul dword [D_063ca]
    fstp dword [edi]
    fld dword [D_063e6]
    fsin
    fmul dword [D_063c6]
    fstp dword [edi + 4]
```

```
2206        fld dword [D_063ea]
2207        fsin
2208        fmul dword [D_063ca]
2209        fstp dword [edi + 8]
2210        fld dword [D_063e6]
2211        fmul dword [D_0638a]
2212        fmul dword [D_0637e]
2213        fmul dword [D_05593]
2214        fstp dword [edi + 0x2c]
2215        fld dword [D_063ea]
2216        fmul dword [D_0638a]
2217        fmul dword [D_0637e]
2218        fmul dword [D_05597]
2219        fstp dword [edi + 0x30]
2220        mov al, byte [D_0558f]
2221        shr al, 1
2222        mov byte [edi + 0x3c], al
2223        mov dword [edi + 0x20], 0
2224        mov dword [edi + 0x24], 0
2225        mov dword [edi + 0x28], 0
2226        add edi, 0x40
2227        fld dword [D_063e6]
2228        fadd dword [D_063ee]
2229        fstp dword [D_063e6]
2230        dec dword [D_063f6]
2231        jne near L_05bb2
2232        fld dword [D_063ea]
2233        fadd dword [D_063f2]
2234        fstp dword [D_063ea]
2235        dec dword [D_063fa]
2236        jne near L_05b9e
2237        mov edi, dword [D_063de]
2238        mov eax, dword [D_05583]
2239        mov dword [D_063d2], eax
2240        mul dword [D_05587]
2241        mov dword [D_063fe], eax
2242        mov dword [D_063ce], 0
2243    L_05ceb:
2244        mov dword [D_063c6], 0
2245        mov dword [D_063ca], 1
2246    L_05cff:
2247        mov eax, dword [D_063ce]
2248        add eax, dword [D_063c6]
2249        mul dword [D_0638e]
2250        add eax, dword [D_063da]
2251        mov dword [edi], eax
2252        mov eax, dword [D_063ce]
2253        add eax, dword [D_063ca]
2254        mul dword [D_0638e]
2255        add eax, dword [D_063da]
2256        mov dword [edi + 4], eax
2257        mov eax, dword [D_063d2]
2258        add eax, dword [D_063ca]
2259        mul dword [D_0638e]
2260        add eax, dword [D_063da]
2261        mov dword [edi + 8], eax
2262        call Triangle_SetupEdges
2263        mov eax, dword [D_063ca]
2264        and eax, dword [D_063d2]
2265        jne short L_05d63
2266        call Triangle_FixUVWrap
2267    L_05d63:
2268        add edi, 0x1c
2269        mov eax, dword [D_063ce]
2270        add eax, dword [D_063c6]
2271        mul dword [D_0638e]
2272        add eax, dword [D_063da]
2273        mov dword [edi], eax
2274        mov eax, dword [D_063d2]
```

```
2275        add eax, dword [D_063ca]
2276        mul dword [D_0638e]
2277        add eax, dword [D_063da]
2278        mov dword [edi + 4], eax
2279        mov eax, dword [D_063d2]
2280        add eax, dword [D_063c6]
2281        mul dword [D_0638e]
2282        add eax, dword [D_063da]
2283        mov dword [edi + 8], eax
2284        call Triangle_SetupEdges
2285        mov eax, dword [D_063ca]
2286        and eax, dword [D_063d2]
2287        jne short L_05dca
2288        call Triangle_FixUVWrap
2289    L_05dca:
2290        add edi, 0x1c
2291        inc dword [D_063c6]
2292        inc dword [D_063ca]
2293        mov eax, dword [D_063ca]
2294        cmp eax, dword [D_05583]
2295        jne short L_05df0
2296        mov dword [D_063ca], 0
2297    L_05df0:
2298        mov eax, dword [D_063c6]
2299        cmp eax, dword [D_05583]
2300        jb near L_05cff
2301        mov eax, dword [D_05583]
2302        add dword [D_063ce], eax
2303        add dword [D_063d2], eax
2304        mov eax, dword [D_063d2]
2305        cmp eax, dword [D_063fe]
2306        jb short L_05e29
2307        mov dword [D_063d2], 0
2308    L_05e29:
2309        cmp eax, dword [D_05583]
2310        jne near L_05ceb
2311        ret
2312
2313    ; ------------------------------------------------------------------------
2314    ; Object_GetVerts - pointer naar vertices van object #EAX
2315    ;
2316    ; Elke object-slot is 0x50 bytes meta; vertices zijn 0x40 bytes per stuk.
2317    ; ESI = object-meta, EDI = start van vertex-array, ECX = aantal vertices.
2318    ; ------------------------------------------------------------------------
2319    Object_GetVerts:
2320        mov ecx, 0x50
2321        mul ecx
2322        add eax, dword [D_1bd04]
2323        db 0x8b, 0xf0     ; mov esi, eax
2324        mov eax, dword [esi + 8]
2325        mov ecx, 0x40
2326        mul ecx
2327        mov ecx, dword [esi + 0xc]
2328        add eax, dword [D_1bd00]
2329        db 0x8b, 0xf8     ; mov edi, eax
2330        ret
2331
2332    ; ------------------------------------------------------------------------
2333    ; Object_Translate - alle vertices verschuiven (optellen van een vector)
2334    ;
2335    ; D_05547/4B/4F = X/Y/Z-offset. Typisch voor "zet object hier neer".
2336    ; ------------------------------------------------------------------------
2337    Object_Translate:
2338        call Object_GetVerts
2339    Object_Translate_Loop:
2340        fld dword [edi]
2341        fadd dword [D_05547]
2342        fstp dword [edi]
2343        fld dword [edi + 4]
```

```
2344        fadd dword [D_0554b]
2345        fstp dword [edi + 4]
2346        fld dword [edi + 8]
2347        fadd dword [D_0554f]
2348        fstp dword [edi + 8]
2349        add edi, 0x40
2350        dec ecx
2351        jne short Object_Translate_Loop
2352        ret
2353
2354    ; ---------------------------------------------------------------------------
2355    ; Object_RotateAroundY - roteer object om de Y-as
2356    ;
2357    ; Hoek in EBX. Per vertex: sin/cos → 2D-rotatie in XZ-vlak, daarna
2358    ; Matrix_TransformVertex. Klassieke demoscene-rotatie.
2359    ; ---------------------------------------------------------------------------
2360    Object_RotateAroundY:
2361        mov dword [D_063e6], ebx
2362        call Object_GetVerts
2363        db 0x8b, 0xe9     ; mov ebp, ecx
2364        push edi
2365        db 0x33, 0xc0     ; xor eax, eax
2366        mov edi, D_063a2
2367        mov ecx, 8
2368        rep stosd
2369        fld1
2370        fstp dword [D_063c2]
2371        pop edi
2372    Object_RotateY_Loop:
2373        fld dword [edi + 4]
2374        fmul dword [D_063e6]
2375        fld st0
2376        fcos
2377        fst dword [D_063a2]
2378        fstp dword [D_063b2]
2379        fsin
2380        fst dword [D_063ae]
2381        fchs
2382        fstp dword [D_063a6]
2383        push edi
2384        add edi, 0
2385        call Matrix_TransformVertex
2386        pop edi
2387        add edi, 0x40
2388        dec ebp
2389        jne short Object_RotateY_Loop
2390        ret
2391
2392    ; ---------------------------------------------------------------------------
2393    ; Object_MergeLastTwo - laatste twee objecten samenvoegen
2394    ;
2395    ; Tel vertex-/triangle-aantallen bij elkaar op en gooi één slot weg.
2396    ; Handig na het bouwen van samengestelde meshes.
2397    ; ---------------------------------------------------------------------------
2398    Object_MergeLastTwo:
2399        mov eax, dword [D_1bd0c]
2400        cmp eax, 2
2401        jb short Object_Merge_Ret
2402        sub eax, 2
2403        call Object_GetVerts
2404        db 0x8b, 0xfe     ; mov edi, esi
2405        add edi, 0x50
2406        mov eax, dword [edi + 0xc]
2407        add dword [esi + 0xc], eax
2408        mov eax, dword [edi + 4]
2409        add dword [esi + 4], eax
2410        dec dword [D_1bd0c]
2411    Object_Merge_Ret:
2412        ret
```

```
2413
2414  ; ------------------------------------------------------------------------
2415  ; Object_Duplicate - kopie van een object (vertices + triangles)
2416  ;
2417  ; Kopieert raw vertex-bytes en past triangle-indices aan met een offset,
2418  ; zodat de kopie naar de nieuwe vertex-range wijst.
2419  ; ------------------------------------------------------------------------
2420  Object_Duplicate:
2421      call Object_GetVerts
2422      db 0x8b, 0xc1      ; mov eax, ecx
2423      mov ebx, dword [esi + 4]
2424      push ebx
2425      push esi
2426      push ecx
2427      push edi
2428      call Object_Alloc
2429      mov edi, dword [D_063da]
2430      pop esi
2431      pop eax
2432      mov ecx, 0x40
2433      mul ecx
2434      db 0x8b, 0xc8      ; mov ecx, eax
2435      rep movsb
2436      pop esi
2437      mov edi, dword [D_063e2]
2438      mov ebx, dword [edi + 8]
2439      sub ebx, dword [esi + 8]
2440      mov eax, 0x40
2441      mul ebx
2442      db 0x8b, 0xd8      ; mov ebx, eax
2443      mov edi, dword [D_063de]
2444      mov eax, dword [esi]
2445      mov edx, 0x1c
2446      mul edx
2447      add eax, dword [D_1bcfc]
2448      db 0x8b, 0xf0      ; mov esi, eax
2449      pop ebp
2450      mov edx, dword [D_0558b]
2451  Object_Dup_TriLoop:
2452      mov ecx, 0x1c
2453      rep movsb
2454      sub edi, 0x1c
2455      add dword [edi], ebx
2456      add dword [edi + 4], ebx
2457      add dword [edi + 8], ebx
2458      mov byte [edi + 0xe], dl
2459      add edi, 0x1c
2460      dec ebp
2461      jne short Object_Dup_TriLoop
2462      ret
2463
2464  ; ------------------------------------------------------------------------
2465  ; Matrix_RotateX / Y / Z - bouw een rotatiematrix
2466  ;
2467  ; Vult de 3×3 (plus extra) matrixcellen D_063a2… met sin/cos van de hoek.
2468  ; Matrix_ApplyToObject past die daarna toe op alle vertices van een object.
2469  ; ------------------------------------------------------------------------
2470  Matrix_RotateX:
2471      push eax
2472      db 0x33, 0xc0      ; xor eax, eax
2473      fld1
2474      fstp dword [D_063a2]
2475      mov edi, D_063a6
2476      mov ecx, 8
2477      rep stosd
2478      mov dword [D_063e6], ebx
2479      fld dword [D_063e6]
2480      fld st0
2481      fcos
```

```
2482        fst dword [D_063b2]
2483        fstp dword [D_063c2]
2484        fsin
2485        fst dword [D_063be]
2486        fchs
2487        fstp dword [D_063b6]
2488        pop eax
2489        jmp near Matrix_ApplyToObject
2490    Matrix_RotateY:
2491        push eax
2492        db 0x33, 0xc0      ; xor eax, eax
2493        mov edi, D_063a2
2494        mov ecx, 9
2495        rep stosd
2496        fld1
2497        fstp dword [D_063b2]
2498        mov dword [D_063e6], ebx
2499        fld dword [D_063e6]
2500        fld st0
2501        fcos
2502        fst dword [D_063a2]
2503        fstp dword [D_063c2]
2504        fsin
2505        fst dword [D_063aa]
2506        fchs
2507        fstp dword [D_063ba]
2508        pop eax
2509        jmp short Matrix_ApplyToObject
2510    Matrix_RotateZ:
2511        push eax
2512        db 0x33, 0xc0      ; xor eax, eax
2513        fld1
2514        fstp dword [D_063c2]
2515        mov edi, D_063a2
2516        mov ecx, 8
2517        rep stosd
2518        mov dword [D_063e6], ebx
2519        fld dword [D_063e6]
2520        fld st0
2521        fcos
2522        fst dword [D_063a2]
2523        fstp dword [D_063b2]
2524        fsin
2525        fst dword [D_063ae]
2526        fchs
2527        fstp dword [D_063a6]
2528        pop eax
2529        jmp short Matrix_ApplyToObject
2530    Matrix_ApplyToObject:
2531        call Object_GetVerts
2532    Matrix_Apply_Loop:
2533        push edi
2534        add edi, 0
2535        call Matrix_TransformVertex
2536        pop edi
2537        add edi, 0x40
2538        dec ecx
2539        jne short Matrix_Apply_Loop
2540        ret
2541    Matrix_MulVec:
2542        fld dword [edi]
2543        fmul st0
2544        fld dword [edi + 4]
2545        fmul st0
2546        fld dword [edi + 8]
2547        fmul st0
2548        faddp st1
2549        faddp st1
2550        fsqrt
```

```
2551        fld1
2552        fdivrp st1
2553        fld st0
2554        fld st0
2555        fmul dword [edi]
2556        fstp dword [edi]
2557        fmul dword [edi + 4]
2558        fstp dword [edi + 4]
2559        fmul dword [edi + 8]
2560        fstp dword [edi + 8]
2561        ret
2562    Matrix_SetupFromParams:
2563        fld dword [esi + 8]
2564        fmul dword [ebx + 4]
2565        fld dword [esi + 4]
2566        fmul dword [ebx + 8]
2567        fsubp st1
2568        fstp dword [edi]
2569        fld dword [esi]
2570        fmul dword [ebx + 8]
2571        fld dword [esi + 8]
2572        fmul dword [ebx]
2573        fsubp st1
2574        fstp dword [edi + 4]
2575        fld dword [esi + 4]
2576        fmul dword [ebx]
2577        fld dword [esi]
2578        fmul dword [ebx + 4]
2579        fsubp st1
2580        fstp dword [edi + 8]
2581        ret
2582
2583    ; ------------------------------------------------------------------------
2584    ; Matrix_TransformVertex - één vertex × matrix
2585    ;
2586    ; Standaard 3D-transform: nieuwe_x/y/z = dotproducts met matrixrijen.
2587    ; ------------------------------------------------------------------------
2588    Matrix_TransformVertex:
2589        mov esi, D_063a2
2590        mov ebx, 0x6396
2591        mov eax, dword [edi]
2592        mov dword [ebx], eax
2593        mov eax, dword [edi + 4]
2594        mov dword [ebx + 4], eax
2595        mov eax, dword [edi + 8]
2596        mov dword [ebx + 8], eax
2597    Matrix_Xform_Loop:
2598        fld dword [ebx]
2599        fmul dword [esi]
2600        fld dword [ebx + 4]
2601        fmul dword [esi + 4]
2602        fld dword [ebx + 8]
2603        fmul dword [esi + 8]
2604        faddp st1
2605        faddp st1
2606        fstp dword [edi]
2607        add esi, 0xc
2608        add edi, 4
2609        cmp esi, 0x63c6
2610        jb short Matrix_Xform_Loop
2611        ret
2612    D_0610e:
2613        db 0x00, 0x00, 0x00, 0x00
2614    D_06112:
2615        db 0x00, 0x00, 0x00, 0x00
2616    D_06116:
2617        db 0x00, 0x00, 0x00, 0x00
2618    D_0611a:
2619        db 0x00, 0x00, 0x00, 0x00
```

```
D_0611e:
    db 0x00, 0x00, 0x00, 0x00
D_06122:
    db 0x00, 0x00, 0x00, 0x00
D_06126:
    db 0x00, 0x00, 0xaf, 0x43
D_0612a:
    db 0x00, 0x00, 0x00, 0x00
L_0612e:
    mov esi, 8
L_06133:
    fld dword [esi + 0x611a]
    fsub dword [esi + 0x610e]
    fstp dword [esi + 0x611a]
    sub esi, 4
    jns short L_06133
    mov edi, D_0611a
    call Matrix_MulVec
    mov esi, 8
L_06159:
    fld dword [esi + 0x611a]
    fmul dword [D_06126]
    fadd dword [esi + 0x610e]
    fstp dword [esi + 0x611a]
    sub esi, 4
    jns short L_06159
    fld dword [D_0611a]
    fchs
    fst dword [D_0634e]
    fadd dword [D_0610e]
    fstp dword [D_06372]
    fld dword [D_0611e]
    fchs
    fst dword [D_06352]
    fadd dword [D_06112]
    fstp dword [D_06376]
    fld dword [D_06122]
    fchs
    fst dword [D_06356]
    fadd dword [D_06116]
    fstp dword [D_0637a]
    fldpi
    fld dword [D_06372]
    fld dword [D_0637a]
    fpatan
    fsubp st1
    fld st0
    fcos
    fstp dword [D_0632a]
    fsin
    fstp dword [D_06332]
    fld dword [D_06382]
    fld dword [D_06372]
    fmul st0
    fld dword [D_0637a]
    fmul st0
    faddp st1
    fsqrt
    fchs
    fld dword [D_06376]
    fpatan
    fsubp st1
    fld st0
    fcos
    fstp dword [D_0633a]
    fsin
    fst dword [D_06346]
    fmul dword [D_06332]
    fstp dword [D_06336]
```

```
2689        mov dword [D_0632e], 0
2690        fld dword [D_0632a]
2691        fmul dword [D_06346]
2692        fchs
2693        fstp dword [D_0633e]
2694        fld dword [D_06332]
2695        fmul dword [D_0633a]
2696        fchs
2697        fstp dword [D_06342]
2698        fld dword [D_0632a]
2699        fmul dword [D_0633a]
2700        fstp dword [D_0634a]
2701        fld dword [D_0612a]
2702        fld st0
2703        fsin
2704        fstp dword [D_063e6]
2705        fcos
2706        fstp dword [D_063ea]
2707        mov ebx, 0xc
2708    L_0628d:
2709        fld dword [ebx + 0x632a]
2710        fmul dword [D_063ea]
2711        fld dword [ebx + 0x6336]
2712        fmul dword [D_063e6]
2713        fsubp st1
2714        fstp dword [ebx + 0x635a]
2715        fld dword [ebx + 0x632a]
2716        fmul dword [D_063e6]
2717        fld dword [ebx + 0x6336]
2718        fmul dword [D_063ea]
2719        faddp st1
2720        fstp dword [ebx + 0x6366]
2721        sub ebx, 4
2722        jns short L_0628d
2723        mov esi, 0x635a
2724        mov edi, D_0632a
2725        mov ecx, 6
2726        rep movsd
2727        mov esi, D_0632a
2728        mov edi, D_063a2
2729        mov ecx, 9
2730        rep movsd
2731        mov edi, D_0634e
2732        call Matrix_TransformVertex
2733        mov edi, dword [D_1bd04]
2734        mov ebx, dword [D_1bd0c]
2735    L_0630a:
2736        push edi
2737        mov esi, D_0632a
2738        add edi, 0x10
2739        mov ecx, 0xc
2740        rep movsd
2741        pop edi
2742        mov eax, dword [D_06126]
2743        mov dword [edi + 0x40], eax
2744        add edi, 0x50
2745        dec ebx
2746        jne short L_0630a
2747        ret
2748    D_0632a:
2749        db 0x00, 0x00, 0x00, 0x00
2750    D_0632e:
2751        db 0x00, 0x00, 0x00, 0x00
2752    D_06332:
2753        db 0x00, 0x00, 0x00, 0x00
2754    D_06336:
2755        db 0x00, 0x00, 0x00, 0x00
2756    D_0633a:
2757        db 0x00, 0x00, 0x00, 0x00
```

```
D_0633e:
    db 0x00, 0x00, 0x00, 0x00
D_06342:
    db 0x00, 0x00, 0x00, 0x00
D_06346:
    db 0x00, 0x00, 0x00, 0x00
D_0634a:
    db 0x00, 0x00, 0x00, 0x00
D_0634e:
    db 0x00, 0x00, 0x00, 0x00
D_06352:
    db 0x00, 0x00, 0x00, 0x00
D_06356:
    times 28 db 0
D_06372:
    db 0x00, 0x00, 0x00, 0x00
D_06376:
    db 0x00, 0x00, 0x00, 0x00
D_0637a:
    db 0x00, 0x00, 0x00, 0x00
D_0637e:
    db 0x00, 0x00, 0x80, 0x47
D_06382:
    db 0xe4, 0xcb, 0x96, 0x40
D_06386:
    db 0x83, 0xf9, 0xa2, 0x42
D_0638a:
    db 0x83, 0xf9, 0x22, 0x42
D_0638e:
    db 0x00, 0x00, 0x00, 0x00
D_06392:
    db 0x00, 0x00, 0x80, 0x43
    times 12 db 0
D_063a2:
    db 0x00, 0x00, 0x00, 0x00
D_063a6:
    db 0x00, 0x00, 0x00, 0x00
D_063aa:
    db 0x00, 0x00, 0x00, 0x00
D_063ae:
    db 0x00, 0x00, 0x00, 0x00
D_063b2:
    db 0x00, 0x00, 0x00, 0x00
D_063b6:
    db 0x00, 0x00, 0x00, 0x00
D_063ba:
    db 0x00, 0x00, 0x00, 0x00
D_063be:
    db 0x00, 0x00, 0x00, 0x00
D_063c2:
    db 0x00, 0x00, 0x00, 0x00
D_063c6:
    db 0x00, 0x00, 0x00, 0x00
D_063ca:
    db 0x00, 0x00, 0x00, 0x00
D_063ce:
    db 0x00, 0x00, 0x00, 0x00
D_063d2:
    db 0x00, 0x00, 0x00, 0x00
D_063d6:
    db 0x00, 0x00, 0x00, 0x00
D_063da:
    db 0x00, 0x00, 0x00, 0x00
D_063de:
    db 0x00, 0x00, 0x00, 0x00
D_063e2:
    db 0x00, 0x00, 0x00, 0x00
D_063e6:
    db 0x00, 0x00, 0x00, 0x00
```

```
D_063ea:
    db 0x00, 0x00, 0x00, 0x00
D_063ee:
    db 0x00, 0x00, 0x00, 0x00
D_063f2:
    db 0x00, 0x00, 0x00, 0x00
D_063f6:
    db 0x00, 0x00, 0x00, 0x00
D_063fa:
    db 0x00, 0x00, 0x00, 0x00
D_063fe:
    db 0x00, 0x00, 0x00, 0x00, 0xa8, 0x1b, 0x71, 0x40, 0x54, 0x57, 0x0c, 0xc0
ProcessList_ObjectTable:
    db 0xa5, 0x30, 0x07, 0x3f
D_0640e:
    db 0xfa, 0x66, 0x00, 0x00, 0xfe, 0x66, 0x00, 0x00, 0x02, 0x67, 0x00, 0x00, 0x06, 0x67
    , 0x00, 0x00
ProcessList_CurrentObj:
    times 8 db 0

; ---------------------------------------------------------------------------
; [5b] ProcessList-interpreter (3D / objecten)
;
; Interne bytecode: eerste byte = opcode (0..0x13).
; Jump-tabel ProcessList_JumpTable kiest de handler.
; Objecten hangen aan ProcessList_ObjectTable / ProcessList_CurrentObj.
; Bij ongeldige opcode: foutmelding "error: verneukte processlist."
; ---------------------------------------------------------------------------
ProcessList_Start:
    mov dword [ProcessList_OutBuffer], edi
    mov edi, dword [D_0640e]
    mov edi, dword [edi]
    mov dword [ProcessList_CurrentObj], edi
ProcessList_Dispatch:
    push eax
    push edx
    mov ah, 9
    mov edx, 0x66ec
    int 0x21
    pop edx
    pop eax
    movzx eax, byte [esi]
    cmp eax, 0x13
    ja near ProcessList_Done
    inc esi
    mov edi, dword [ProcessList_CurrentObj]
    mov ebx, dword [eax*4 + ProcessList_JumpTable]
    jmp ebx

; Jump-tabel: index = opcode-byte → handler-adres (little-endian dwords).
; Opcode > 0x13 → foutpad ("verneukte processlist").
ProcessList_JumpTable:
    db 0xdb, 0x64, 0x00, 0x00, 0xd8, 0x66, 0x00, 0x00, 0xf4, 0x64, 0x00, 0x00, 0xff, 0x64
    , 0x00, 0x00
    db 0x0f, 0x65, 0x00, 0x00, 0x24, 0x65, 0x00, 0x00, 0xc5, 0x66, 0x00, 0x00, 0x47, 0x65
    , 0x00, 0x00
    db 0x52, 0x65, 0x00, 0x00, 0x7c, 0x65, 0x00, 0x00, 0xa6, 0x65, 0x00, 0x00, 0xd0, 0x65
    , 0x00, 0x00
    db 0xfe, 0x65, 0x00, 0x00, 0x28, 0x66, 0x00, 0x00, 0x43, 0x66, 0x00, 0x00, 0x5e, 0x66
    , 0x00, 0x00
    db 0x79, 0x66, 0x00, 0x00, 0xa2, 0x66, 0x00, 0x00, 0xb5, 0x66, 0x00, 0x00, 0xbd, 0x66
    , 0x00, 0x00
    db "error: verneukte processlist.$"
    db 0xba, 0xb3, 0x64, 0x00, 0x00, 0xb4, 0x09, 0xcd, 0x21, 0xc3
    db 0x33, 0xc0     ; xor eax, eax
    mov al, byte [esi]
    inc esi
    mov eax, dword [eax*4 + ProcessList_ObjectTable]
    mov eax, dword [eax]
```

```
2890        mov dword [ProcessList_CurrentObj], eax
2891        push esi
2892        jmp near ProcessList_Next
2893        push esi
2894        call L_0670a
2895        jmp near ProcessList_Next
2896        mov ecx, dword [esi]
2897        add esi, 4
2898        push esi
2899        call L_06719
2900        jmp near ProcessList_Next
2901        movzx eax, byte [esi]
2902        movzx ecx, byte [esi + 1]
2903        add esi, 2
2904        push esi
2905        db 0xe8, 0xf8, 0x04, 0x00, 0x00     ; call 0x6a17
2906        jmp near ProcessList_Next
2907        mov al, byte [esi]
2908        mov ah, byte [esi + 1]
2909        mov bl, byte [esi + 2]
2910        mov bh, byte [esi + 3]
2911        mov cl, byte [esi + 4]
2912        mov ch, byte [esi + 5]
2913        movzx edx, byte [esi + 6]
2914        add esi, 7
2915        push esi
2916        call L_067b8
2917        jmp near ProcessList_Next
2918        push esi
2919        call L_0721a+1
2920        jmp near ProcessList_Next
2921        movzx edi, byte [esi]
2922        mov edi, dword [edi*4 + ProcessList_ObjectTable]
2923        mov edi, dword [edi]
2924        movzx ecx, byte [esi + 1]
2925        add esi, 2
2926        push esi
2927        mov esi, dword [ProcessList_CurrentObj]
2928        mov dword [ProcessList_CurrentObj], edi
2929        call L_06a46
2930        jmp near ProcessList_Next
2931        movzx edi, byte [esi]
2932        mov edi, dword [edi*4 + ProcessList_ObjectTable]
2933        mov edi, dword [edi]
2934        movzx ecx, byte [esi + 1]
2935        add esi, 2
2936        push esi
2937        mov esi, dword [ProcessList_CurrentObj]
2938        mov dword [ProcessList_CurrentObj], edi
2939        call L_06a74
2940        jmp near ProcessList_Next
2941        movzx edi, byte [esi]
2942        mov edi, dword [edi*4 + ProcessList_ObjectTable]
2943        mov edi, dword [edi]
2944        movzx ecx, byte [esi + 1]
2945        add esi, 2
2946        push esi
2947        mov esi, dword [ProcessList_CurrentObj]
2948        mov dword [ProcessList_CurrentObj], edi
2949        call L_06b11
2950        jmp near ProcessList_Next
2951        movzx edi, byte [esi]
2952        mov edi, dword [edi*4 + ProcessList_ObjectTable]
2953        mov edi, dword [edi]
2954        movzx ebx, byte [esi + 1]
2955        movzx ecx, byte [esi + 2]
2956        add esi, 3
2957        push esi
2958        mov esi, dword [ProcessList_CurrentObj]
```

```
2959        mov dword [ProcessList_CurrentObj], edi
2960        call L_06a95
2961        jmp near ProcessList_Next
2962        movzx edi, byte [esi]
2963        mov edi, dword [edi*4 + ProcessList_ObjectTable]
2964        mov edi, dword [edi]
2965        movzx ecx, byte [esi + 1]
2966        add esi, 2
2967        push esi
2968        mov esi, dword [ProcessList_CurrentObj]
2969        mov dword [ProcessList_CurrentObj], edi
2970        call L_06ae2
2971        jmp near ProcessList_Next
2972        movzx ebx, byte [esi]
2973        movzx ecx, byte [esi + 1]
2974        add esi, 2
2975        push esi
2976        mov edi, dword [ProcessList_CurrentObj]
2977        call L_06b4a
2978        jmp near ProcessList_Next
2979        movzx ebx, byte [esi]
2980        mov ebx, dword [ebx*4 + ProcessList_ObjectTable]
2981        mov ebx, dword [ebx]
2982        movzx ecx, byte [esi + 1]
2983        add esi, 2
2984        push esi
2985        call L_06b73
2986        jmp short ProcessList_Next
2987        movzx ebx, byte [esi]
2988        mov ebx, dword [ebx*4 + ProcessList_ObjectTable]
2989        mov ebx, dword [ebx]
2990        movzx ecx, byte [esi + 1]
2991        add esi, 2
2992        push esi
2993        call L_06c6e
2994        jmp short ProcessList_Next
2995        movsx eax, word [esi]
2996        movzx ebx, word [esi + 2]
2997        movzx edi, byte [esi + 4]
2998        movzx ecx, byte [esi + 5]
2999        add esi, 6
3000        mov edi, dword [edi*4 + ProcessList_ObjectTable]
3001        mov edi, dword [edi]
3002        push esi
3003        mov esi, dword [ProcessList_CurrentObj]
3004        call L_06daf
3005        jmp short ProcessList_Next
3006        mov eax, dword [esi]
3007        mov ebx, dword [esi + 4]
3008        mov cl, byte [esi + 8]
3009        add esi, 9
3010        push esi
3011        call L_06ef1
3012        jmp short ProcessList_Next
3013        push esi
3014        call L_06f5b
3015        jmp short ProcessList_Next
3016        push esi
3017        call L_06f86
3018        jmp short ProcessList_Next
3019        mov ebx, dword [esi]
3020        add esi, 4
3021        push esi
3022        call L_06fb1
3023        jmp short ProcessList_Next
3024    ProcessList_Next:
3025        pop esi
3026        jmp near ProcessList_Dispatch
3027    ProcessList_Done:
```

```
3028        mov esi, dword [ProcessList_CurrentObj]
3029        mov edi, dword [ProcessList_OutBuffer]
3030        mov ecx, 0xc000
3031        rep movsd
3032        ret
3033        db 0x2e, 0x24
3034    ProcessList_OutBuffer:
3035        times 8 db 0
3036        db 0xc0, 0x03, 0x00, 0x00
3037    Buf_A:
3038        db 0x00, 0x00, 0x00, 0x00
3039    Buf_B:
3040        db 0x00, 0x00, 0x00, 0x00
3041    Buf_C:
3042        db 0x00, 0x00, 0x00, 0x00
3043    Buf_D:
3044        db 0x00, 0x00, 0x00, 0x00
3045    L_0670a:
3046        mov ebp, 0xc000
3047    L_0670f:
3048        call L_07235
3049        stosd
3050        dec ebp
3051        jne short L_0670f
3052        ret
3053    L_06719:
3054        mov dword [D_02008], ecx
3055    L_0671f:
3056        call L_07235
3057        db 0x8b, 0xf0      ; mov esi, eax
3058        shr esi, 9
3059        mov ecx, 3
3060        call L_067a9
3061        add esi, 0xfc
3062        mov ecx, 5
3063        call L_067a9
3064        add esi, 0xfa
3065        mov ecx, 7
3066        call L_067a9
3067        add esi, 0xf8
3068        mov edx, 5
3069    L_0675e:
3070        mov ecx, 9
3071        call L_067a9
3072        add esi, 0xf7
3073        dec edx
3074        jne short L_0675e
3075        inc esi
3076        mov ecx, 7
3077        call L_067a9
3078        add esi, 0xfa
3079        mov ecx, 5
3080        call L_067a9
3081        add esi, 0xfc
3082        mov ecx, 3
3083        call L_067a9
3084        dec dword [D_02008]
3085        jne near L_0671f
3086        ret
3087    L_067a9:
3088        and esi, 0xffff
3089        mov byte [edi + esi], al
3090        inc si
3091        dec ecx
3092        jne short L_067a9
3093        ret
3094    L_067b8:
3095        mov dword [D_0200c], edx
3096        mov byte [D_02000], al
```

```
3097        sar al, 7
3098        mov byte [D_02001], al
3099        mov byte [D_02002], ah
3100        sar ah, 7
3101        mov byte [D_02003], ah
3102        mov byte [D_02004], bl
3103        sar bl, 7
3104        mov byte [D_02005], bl
3105        mov byte [D_02006], bh
3106        sar bh, 7
3107        mov byte [D_02007], bh
3108        mov byte [D_02008], cl
3109        sar cl, 7
3110        mov byte [D_02009], cl
3111        mov byte [D_0200a], ch
3112        sar ch, 7
3113        mov byte [D_0200b], ch
3114        call L_07235
3115        mov dword [D_02010], eax
3116        mov dword [D_02014], 0x3b9aca00
3117        fild dword [D_02010]
3118        fidiv dword [D_02014]
3119        fldpi
3120        fmulp st1
3121        fstp dword [D_069e7]
3122        call L_07235
3123        mov dword [D_02010], eax
3124        fild dword [D_02010]
3125        fidiv dword [D_02014]
3126        fldpi
3127        fmulp st1
3128        fstp dword [D_069eb]
3129        call L_07235
3130        mov dword [D_02010], eax
3131        fild dword [D_02010]
3132        fidiv dword [D_02014]
3133        fldpi
3134        fmulp st1
3135        fstp dword [D_069ef]
3136        call L_07235
3137        mov dword [D_02010], eax
3138        fild dword [D_02010]
3139        fidiv dword [D_02014]
3140        fldpi
3141        fmulp st1
3142        fstp dword [D_069f3]
3143        fild word [D_02000]
3144        fmul dword [L_06a13]
3145        fstp dword [D_069fb]
3146        fild word [D_02002]
3147        fmul dword [L_06a13]
3148        fstp dword [D_069ff]
3149        fild word [D_02004]
3150        fmul dword [L_06a13]
3151        fstp dword [D_06a07]
3152        fild word [D_02006]
3153        fmul dword [L_06a13]
3154        fstp dword [D_06a0b]
3155        fild word [D_02008]
3156        fmul dword [L_06a13]
3157        fstp dword [D_06a03]
3158        fild word [D_0200a]
3159        fmul dword [L_06a13]
3160        fstp dword [D_06a0f]
3161        mov ebp, 0x100
3162
3163    ; ---------------------------------------------------------------------------
3164    ; Effect_Plasma - klassiek “plasma”-patroon
3165    ;
```

```
; Geneste lussen over het scherm; per pixel worden meerdere sinussen
; van x, y en de tijd opgeteld. Het resultaat (laag byte) is de kleur.
; Zeer typisch voor 90's demoscene-intros.
; ------------------------------------------------------------------------
Effect_Plasma_Row:
    fld dword [D_069ef]
    fadd dword [D_06a07]
    fst dword [D_069ef]
    fsin
    fldpi
    fmulp st1
    fld dword [D_069f3]
    fadd dword [D_06a0b]
    fst dword [D_069f3]
    faddp st1
    fadd dword [D_069e7]
    fsin
    fimul dword [D_0200c]
    fld dword [D_069e7]
    fadd dword [D_06a03]
    fst dword [D_069e7]
    fstp dword [D_069df]
    fld dword [D_069eb]
    fstp dword [D_069e3]
    fld dword [D_069ef]
    fstp dword [D_069f7]
    mov esi, 0x100
Effect_Plasma_Pixel:
    fld dword [D_069df]
    fadd dword [D_069fb]
    fst dword [D_069df]
    fsin
    fldpi
    fmulp st1
    fld dword [D_069e3]
    fadd dword [D_069ff]
    fst dword [D_069e3]
    faddp st1
    fld dword [D_069f7]
    fadd dword [D_06a0f]
    fst dword [D_069f7]
    faddp st1
    fsin
    fmul st1
    fistp dword [D_02010]
    mov al, byte [D_02010]
    mov byte [edi], al
    inc edi
    dec esi
    jne short Effect_Plasma_Pixel
    faddp st0
    dec ebp
    jne near Effect_Plasma_Row
    ret
D_069df:
    db 0x00, 0x00, 0x00, 0x00
D_069e3:
    db 0x00, 0x00, 0x00, 0x00
D_069e7:
    db 0x00, 0x00, 0x00, 0x00
D_069eb:
    db 0x00, 0x00, 0x00, 0x00
D_069ef:
    db 0x00, 0x00, 0x00, 0x00
D_069f3:
    db 0x00, 0x00, 0x00, 0x00
D_069f7:
    db 0x00, 0x00, 0x00, 0x00
D_069fb:
```

```
3235        db 0x00, 0x00, 0x00, 0x00
3236    D_069ff:
3237        db 0x00, 0x00, 0x00, 0x00
3238    D_06a03:
3239        db 0x00, 0x00, 0x00, 0x00
3240    D_06a07:
3241        db 0x00, 0x00, 0x00, 0x00
3242    D_06a0b:
3243        db 0x00, 0x00, 0x00, 0x00
3244    D_06a0f:
3245        db 0x00, 0x00, 0x00, 0x00
3246    L_06a13:
3247        fisttp dword [edi]
3248        leave
3249        cmp al, 0x83
3250        stc
3251        add esi, dword [eax + edx + 3]
3252        stc
3253        mov ecx, 0x10000
3254    L_06a23:
3255        add byte [edi], al
3256        add edi, 3
3257        dec ecx
3258        jne short L_06a23
3259        ret
3260        db 0x8a, 0xe0      ; mov ah, al
3261        shl eax, 8
3262        db 0x8a, 0xc4      ; mov al, ah
3263        shl eax, 8
3264        db 0x8a, 0xc4      ; mov al, ah
3265        mov ecx, 0xc000
3266    L_06a3d:
3267        add dword [edi], eax
3268        add edi, 4
3269        dec ecx
3270        jne short L_06a3d
3271        ret
3272    L_06a46:
3273        cmp ecx, 3
3274        jb short L_06a53
3275        mov ecx, 0xc000
3276        rep movsd
3277        ret
3278    L_06a53:
3279        db 0x03, 0xf9      ; add edi, ecx
3280        mov ecx, 0x4000
3281    L_06a5a:
3282        mov eax, dword [esi]
3283        mov byte [edi], al
3284        mov byte [edi + 3], ah
3285        shr eax, 0x10
3286        add esi, 4
3287        mov byte [edi + 6], al
3288        mov byte [edi + 9], ah
3289        add edi, 0xc
3290        dec ecx
3291        jne short L_06a5a
3292        ret
3293    L_06a74:
3294        db 0x03, 0xf1      ; add esi, ecx
3295        mov ecx, 0x4000
3296    L_06a7b:
3297        mov al, byte [esi + 6]
3298        mov ah, byte [esi + 9]
3299        shl eax, 0x10
3300        mov al, byte [esi]
3301        mov ah, byte [esi + 3]
3302        add esi, 0xc
3303        mov dword [edi], eax
```

```
3304        add edi, 4
3305        dec ecx
3306        jne short L_06a7b
3307        ret
3308    L_06a95:
3309        cmp ebx, 0x100
3310        je short L_06a46
3311        cmp ebx, 0
3312        jne short L_06aa3
3313        ret
3314    L_06aa3:
3315        db 0x8a, 0xfb      ; mov bh, bl
3316        neg bh
3317        cmp ecx, 3
3318        jb short L_06ac5
3319        mov ebp, 0x30000
3320    L_06ab1:
3321        mov al, byte [esi]
3322        mul bl
3323        db 0x8a, 0xcc      ; mov cl, ah
3324        mov al, byte [edi]
3325        mul bh
3326        db 0x02, 0xe1      ; add ah, cl
3327        mov byte [edi], ah
3328        inc esi
3329        inc edi
3330        dec ebp
3331        jne short L_06ab1
3332        ret
3333    L_06ac5:
3334        db 0x03, 0xf9      ; add edi, ecx
3335        mov ebp, 0x10000
3336    L_06acc:
3337        mov al, byte [esi]
3338        mul bl
3339        db 0x8b, 0xc8      ; mov ecx, eax
3340        mov al, byte [edi]
3341        mul bh
3342        db 0x03, 0xc1      ; add eax, ecx
3343        mov byte [edi], ah
3344        inc esi
3345        add edi, 3
3346        dec ebp
3347        jne short L_06acc
3348        ret
3349    L_06ae2:
3350        cmp ecx, 3
3351        jb short L_06afa
3352        mov ebp, 0x30000
3353    L_06aec:
3354        mov bl, byte [esi]
3355        mov al, byte [edi]
3356        mul bl
3357        mov byte [edi], ah
3358        inc esi
3359        inc edi
3360        dec ebp
3361        jne short L_06aec
3362        ret
3363    L_06afa:
3364        db 0x03, 0xf9      ; add edi, ecx
3365        mov ebp, 0x10000
3366    L_06b01:
3367        mov bl, byte [esi]
3368        mov al, byte [edi]
3369        mul bl
3370        mov byte [edi], ah
3371        inc esi
3372        add edi, 3
```

```
3373        dec ebp
3374        jne short L_06b01
3375        ret
3376    L_06b11:
3377        cmp ecx, 3
3378        jb short L_06b29
3379        mov ecx, 0xc000
3380    L_06b1b:
3381        mov eax, dword [esi]
3382        add dword [edi], eax
3383        add esi, 4
3384        add edi, 4
3385        dec ecx
3386        jne short L_06b1b
3387        ret
3388    L_06b29:
3389        db 0x03, 0xf9      ; add edi, ecx
3390        mov ecx, 0x4000
3391    L_06b30:
3392        mov eax, dword [esi]
3393        add byte [edi], al
3394        add byte [edi + 3], ah
3395        shr eax, 0x10
3396        add esi, 4
3397        add byte [edi + 6], al
3398        add byte [edi + 9], ah
3399        add edi, 0xc
3400        dec ecx
3401        jne short L_06b30
3402        ret
3403    L_06b4a:
3404        cmp ecx, 3
3405        jb short L_06b5f
3406        mov ecx, 0x30000
3407    L_06b54:
3408        mov al, byte [edi]
3409        mul bl
3410        mov byte [edi], ah
3411        inc edi
3412        dec ecx
3413        jne short L_06b54
3414        ret
3415    L_06b5f:
3416        db 0x03, 0xf9      ; add edi, ecx
3417        mov ecx, 0x10000
3418    L_06b66:
3419        mov al, byte [edi]
3420        mul bl
3421        mov byte [edi], ah
3422        add edi, 3
3423        dec ecx
3424        jne short L_06b66
3425        ret
3426    L_06b73:
3427        cmp ecx, 2
3428        ja short L_06bdb
3429        push ebx
3430        push ecx
3431        push edi
3432        db 0x8b, 0xf7      ; mov esi, edi
3433        db 0x8b, 0xfb      ; mov edi, ebx
3434        call L_06a74
3435        pop edi
3436        pop ecx
3437        pop esi
3438        db 0x03, 0xf9      ; add edi, ecx
3439        mov ecx, 0xff
3440        db 0x32, 0xff      ; xor bh, bh
3441    L_06b90:
```

```
3442        db 0x32, 0xdb     ; xor bl, bl
3443    L_06b92:
3444        db 0x33, 0xc0     ; xor eax, eax
3445        db 0x33, 0xd2     ; xor edx, edx
3446        call L_06bc3
3447        db 0x03, 0xc0     ; add eax, eax
3448        dec ch
3449        call L_06bc3
3450        add ch, 2
3451        call L_06bc3
3452        dec ch
3453        shr eax, 4
3454        mov byte [edi], al
3455        add edi, 3
3456        inc cl
3457        dec bl
3458        jne short L_06b92
3459        inc ch
3460        dec bh
3461        jne short L_06b90
3462        ret
3463    L_06bc3:
3464        mov dl, byte [esi + ecx]
3465        db 0x03, 0xc2     ; add eax, edx
3466        inc cl
3467        mov dl, byte [esi + ecx]
3468        lea eax, [eax + edx*2]
3469        inc cl
3470        mov dl, byte [esi + ecx]
3471        db 0x03, 0xc2     ; add eax, edx
3472        add cl, 0xfe
3473        ret
3474    L_06bdb:
3475        push ebx
3476        push ecx
3477        push edi
3478        db 0x8b, 0xf7     ; mov esi, edi
3479        db 0x8b, 0xfb     ; mov edi, ebx
3480        mov ecx, 3
3481        call L_06a46
3482        pop edi
3483        pop ecx
3484        pop ebx
3485        db 0x8b, 0xf3     ; mov esi, ebx
3486        mov ecx, 0xff
3487        mov dword [D_02018], 0x100
3488    L_06c00:
3489        mov dword [D_0200c], 0x100
3490    L_06c0a:
3491        mov ebp, 3
3492    L_06c0f:
3493        db 0x33, 0xc0     ; xor eax, eax
3494        db 0x33, 0xd2     ; xor edx, edx
3495        call L_06c4d
3496        dec ch
3497        db 0x03, 0xc0     ; add eax, eax
3498        call L_06c4d
3499        add ch, 2
3500        call L_06c4d
3501        dec ch
3502        shr eax, 4
3503        mov byte [edi], al
3504        inc esi
3505        inc edi
3506        dec ebp
3507        jne short L_06c0f
3508        sub esi, 3
3509        inc cl
3510        dec dword [D_0200c]
```

```
3511        jne short L_06c0a
3512        inc ch
3513        dec dword [D_02018]
3514        jne short L_06c00
3515        ret
3516    L_06c4d:
3517        lea ebx, [ecx + ecx*2]
3518        mov dl, byte [esi + ebx]
3519        db 0x03, 0xc2       ; add eax, edx
3520        inc cl
3521        lea ebx, [ecx + ecx*2]
3522        mov dl, byte [esi + ebx]
3523        lea eax, [eax + edx*2]
3524        inc cl
3525        lea ebx, [ecx + ecx*2]
3526        mov dl, byte [esi + ebx]
3527        db 0x03, 0xc2       ; add eax, edx
3528        add cl, 0xfe
3529        ret
3530    L_06c6e:
3531        cmp ecx, 2
3532        ja near L_06cf8
3533        push ebx
3534        push ecx
3535        push edi
3536        db 0x8b, 0xf7       ; mov esi, edi
3537        db 0x8b, 0xfb       ; mov edi, ebx
3538        call L_06a74
3539        pop edi
3540        pop ecx
3541        pop esi
3542        db 0x03, 0xf9       ; add edi, ecx
3543        mov ecx, 0xfffe
3544        db 0x32, 0xff       ; xor bh, bh
3545    L_06c8f:
3546        db 0x32, 0xdb       ; xor bl, bl
3547    L_06c91:
3548        db 0x33, 0xc0       ; xor eax, eax
3549        db 0x33, 0xd2       ; xor edx, edx
3550        call L_06cd0
3551        inc ch
3552        call L_06cd0
3553        inc ch
3554        call L_06cd0
3555        inc ch
3556        db 0x03, 0xc0       ; add eax, eax
3557        call L_06cd0
3558        sub ch, 4
3559        call L_06cd0
3560        inc ch
3561        shr eax, 6
3562        mov byte [edi], al
3563        add edi, 3
3564        inc cl
3565        dec bl
3566        jne short L_06c91
3567        inc ch
3568        dec bh
3569        jne short L_06c8f
3570        ret
3571    L_06cd0:
3572        mov dl, byte [esi + ecx]
3573        db 0x03, 0xc2       ; add eax, edx
3574        inc cl
3575        mov dl, byte [esi + ecx]
3576        lea eax, [eax + edx*2]
3577        inc cl
3578        mov dl, byte [esi + ecx]
3579        lea eax, [eax + edx*2]
```

```
3580        inc cl
3581        mov dl, byte [esi + ecx]
3582        lea eax, [eax + edx*2]
3583        inc cl
3584        mov dl, byte [esi + ecx]
3585        db 0x03, 0xc2       ; add eax, edx
3586        add cl, 0xfc
3587        ret
3588    L_06cf8:
3589        push ebx
3590        push ecx
3591        push edi
3592        db 0x8b, 0xf7       ; mov esi, edi
3593        db 0x8b, 0xfb       ; mov edi, ebx
3594        mov ecx, 3
3595        call L_06a46
3596        pop edi
3597        pop ecx
3598        pop ebx
3599        db 0x8b, 0xf3       ; mov esi, ebx
3600        mov ecx, 0xfffe
3601        mov dword [D_02018], 0x100
3602    L_06d1d:
3603        mov dword [D_0200c], 0x100
3604    L_06d27:
3605        mov ebp, 3
3606    L_06d2c:
3607        db 0x33, 0xc0       ; xor eax, eax
3608        db 0x33, 0xd2       ; xor edx, edx
3609        call L_06d78
3610        inc ch
3611        call L_06d78
3612        inc ch
3613        call L_06d78
3614        inc ch
3615        db 0x03, 0xc0       ; add eax, eax
3616        call L_06d78
3617        sub ch, 4
3618        call L_06d78
3619        inc ch
3620        shr eax, 6
3621        mov byte [edi], al
3622        inc esi
3623        inc edi
3624        dec ebp
3625        jne short L_06d2c
3626        sub esi, 3
3627        inc cl
3628        dec dword [D_0200c]
3629        jne short L_06d27
3630        inc ch
3631        dec dword [D_02018]
3632        jne short L_06d1d
3633        ret
3634    L_06d78:
3635        lea ebx, [ecx + ecx*2]
3636        mov dl, byte [esi + ebx]
3637        db 0x03, 0xc2       ; add eax, edx
3638        inc cl
3639        lea ebx, [ecx + ecx*2]
3640        mov dl, byte [esi + ebx]
3641        lea eax, [eax + edx*2]
3642        inc cl
3643        lea ebx, [ecx + ecx*2]
3644        mov dl, byte [esi + ebx]
3645        lea eax, [eax + edx*2]
3646        inc cl
3647        lea ebx, [ecx + ecx*2]
3648        mov dl, byte [esi + ebx]
```

```
3649        lea eax, [eax + edx*2]
3650        inc cl
3651        lea ebx, [ecx + ecx*2]
3652        mov dl, byte [esi + ebx]
3653        db 0x03, 0xc2       ; add eax, edx
3654        add cl, 0xfc
3655        ret
3656    L_06daf:
3657        mov dword [D_02000], eax
3658        mov dword [D_0200c], ebx
3659        fild dword [D_02000]
3660        fmul dword [D_06eed]
3661        fst dword [D_02000]
3662        fsin
3663        fchs
3664        fst dword [D_02004]
3665        fabs
3666        fld dword [D_02000]
3667        fcos
3668        fst dword [D_02008]
3669        fabs
3670        faddp st1
3671        fild dword [D_0200c]
3672        fdivrp st1
3673        fld dword [D_02004]
3674        fmul st1
3675        fistp dword [D_02004]
3676        fld dword [D_02008]
3677        fmulp st1
3678        fistp dword [D_02008]
3679        push esi
3680        push edi
3681        db 0x03, 0xf1       ; add esi, ecx
3682        add edi, 0x10000
3683        mov ecx, 0xfefe
3684    L_06e1d:
3685        inc cl
3686        lea edx, [ecx + ecx*2]
3687        db 0x33, 0xc0       ; xor eax, eax
3688        db 0x33, 0xdb       ; xor ebx, ebx
3689        mov al, byte [esi + edx]
3690        dec cl
3691        lea edx, [ecx + ecx*2]
3692        mov bl, byte [esi + edx]
3693        db 0x2b, 0xc3       ; sub eax, ebx
3694        imul dword [D_02008]
3695        mov dword [D_0200c], eax
3696        inc ch
3697        lea edx, [ecx + ecx*2]
3698        db 0x33, 0xc0       ; xor eax, eax
3699        dec ch
3700        mov al, byte [esi + edx]
3701        db 0x2b, 0xc3       ; sub eax, ebx
3702        imul dword [D_02004]
3703        add eax, dword [D_0200c]
3704        js short L_06e6c
3705        cmp eax, 0xffff
3706        ja short L_06e67
3707        not ah
3708        mov byte [edi], ah
3709        jmp short L_06e6f
3710    L_06e67:
3711        mov byte [edi], 0
3712        jmp short L_06e6f
3713    L_06e6c:
3714        mov byte [edi], 0xff
3715    L_06e6f:
3716        inc edi
3717        inc cx
```

```
3718        cmp ecx, 0xfefe
3719        jne short L_06e1d
3720        pop esi
3721        push esi
3722        db 0x8b, 0xfe       ; mov edi, esi
3723        add esi, 0x10000
3724        db 0x33, 0xc9       ; xor ecx, ecx
3725        db 0x33, 0xdb       ; xor ebx, ebx
3726    L_06e88:
3727        db 0x33, 0xc0       ; xor eax, eax
3728        inc ch
3729        call L_06eae
3730        db 0x03, 0xc0       ; add eax, eax
3731        inc ch
3732        call L_06eae
3733        sub ch, 2
3734        call L_06eae
3735        shr eax, 4
3736        mov byte [edi], al
3737        inc edi
3738        inc cx
3739        jne short L_06e88
3740        jmp short L_06ec7
3741    L_06eae:
3742        mov bl, byte [esi + ecx]
3743        db 0x03, 0xc3       ; add eax, ebx
3744        inc cl
3745        mov bl, byte [esi + ecx]
3746        db 0x03, 0xc3       ; add eax, ebx
3747        db 0x03, 0xc3       ; add eax, ebx
3748        inc cl
3749        mov bl, byte [esi + ecx]
3750        db 0x03, 0xc3       ; add eax, ebx
3751        sub cl, 2
3752        ret
3753    L_06ec7:
3754        pop esi
3755        pop edi
3756        mov ebp, 0x10000
3757    L_06ece:
3758        mov bl, byte [esi]
3759        mov bh, 3
3760    L_06ed2:
3761        mov al, byte [edi]
3762        db 0x8a, 0xe0       ; mov ah, al
3763        shr ah, 3
3764        db 0x2a, 0xc4       ; sub al, ah
3765        not al
3766        mul bl
3767        not ah
3768        mov byte [edi], ah
3769        inc edi
3770        dec bh
3771        jne short L_06ed2
3772        inc esi
3773        dec ebp
3774        jne short L_06ece
3775        ret
3776    D_06eed:
3777        db 0x35, 0xfa, 0x8e, 0x3c
3778    L_06ef1:
3779        mov dword [D_06f52], eax
3780        mov dword [D_06f56], eax
3781        mov byte [D_06f5a], cl
3782        mov dword [D_02018], 0x10000
3783    L_06f0b:
3784        mov ecx, dword [edi]
3785        mov esi, D_06f52
3786        mov bl, 3
```

```
3787  L_06f14:
3788      mov bh, 3
3789      db 0x33, 0xed     ; xor ebp, ebp
3790  L_06f18:
3791      movsx eax, byte [esi]
3792      db 0x33, 0xd2     ; xor edx, edx
3793      db 0x8a, 0xd1     ; mov dl, cl
3794      imul edx
3795      db 0x03, 0xe8     ; add ebp, eax
3796      inc esi
3797      ror ecx, 8
3798      dec bh
3799      jne short L_06f18
3800      ror ecx, 8
3801      db 0x8b, 0xc5     ; mov eax, ebp
3802      sar eax, 7
3803      cmp eax, 0xff
3804      jg short L_06f40
3805      jbe short L_06f42
3806      db 0x33, 0xc0     ; xor eax, eax
3807      jmp short L_06f42
3808  L_06f40:
3809      mov al, 0xff
3810  L_06f42:
3811      mov byte [edi], al
3812      inc edi
3813      dec bl
3814      jne short L_06f14
3815      dec dword [D_02018]
3816      jne short L_06f0b
3817      ret
3818  D_06f52:
3819      db 0x00, 0x00, 0x00, 0x00
3820  D_06f56:
3821      db 0x00, 0x00, 0x00, 0x00
3822  D_06f5a:
3823      db 0x00
3824  L_06f5b:
3825      mov dword [D_02018], 0x10000
3826  L_06f65:
3827      mov al, byte [edi + 2]
3828      mov bl, byte [edi + 1]
3829      mov cl, byte [edi]
3830      call L_070c0+1
3831      mov byte [edi], al
3832      mov byte [edi + 1], bl
3833      mov byte [edi + 2], cl
3834      add edi, 3
3835      dec dword [D_02018]
3836      jne short L_06f65
3837      ret
3838  L_06f86:
3839      mov dword [D_02018], 0x10000
3840  L_06f90:
3841      mov al, byte [edi]
3842      mov bl, byte [edi + 1]
3843      mov cl, byte [edi + 2]
3844      call L_06fd1
3845      mov byte [edi + 2], al
3846      mov byte [edi + 1], bl
3847      mov byte [edi], cl
3848      add edi, 3
3849      dec dword [D_02018]
3850      jne short L_06f90
3851      ret
3852  L_06fb1:
3853      movzx edx, word [ebx]
3854      add ebx, 2
3855      db 0x33, 0xc0     ; xor eax, eax
```

```
3856        dec edx
3857    L_06fba:
3858        movzx ecx, byte [ebx]
3859        inc ebx
3860        cmp ecx, 0
3861        jne short L_06fc9
3862        movzx ecx, word [ebx]
3863        add ebx, 2
3864    L_06fc9:
3865        rep stosb
3866        not al
3867        dec edx
3868        jns short L_06fba
3869        ret
3870    L_06fd1:
3871        and ecx, 0xff
3872        and eax, 0xff
3873        and ebx, 0xff
3874        mov byte [D_070be], cl
3875        inc ecx
3876        mov edx, 6
3877        mul edx
3878        db 0x8b, 0xe8       ; mov ebp, eax
3879        xor ah, 1
3880        ror ah, 1
3881        sar ah, 7
3882        db 0x32, 0xc4       ; xor al, ah
3883        and eax, 0xff
3884        mul ebx
3885        not ah
3886        db 0x8a, 0xc4       ; mov al, ah
3887        and eax, 0xff
3888        mul ecx
3889        mov byte [D_070bf], ah
3890        db 0x8a, 0xc3       ; mov al, bl
3891        not al
3892        and eax, 0xff
3893        mul ecx
3894        mov byte [L_070c0], ah
3895        db 0x8b, 0xc5       ; mov eax, ebp
3896        db 0x8a, 0xc4       ; mov al, ah
3897        and eax, 0xff
3898        lea ebx, [eax + eax*2]
3899        shl ebx, 2
3900        mov edx, dword [ebx + 0x7073]
3901        mov al, byte [D_070be]
3902        mov byte [edx], al
3903        mov edx, dword [ebx + 0x7077]
3904        mov al, byte [D_070bf]
3905        mov byte [edx], al
3906        mov edx, dword [ebx + 0x707b]
3907        mov al, byte [L_070c0]
3908        mov byte [edx], al
3909        db 0x33, 0xc0       ; xor eax, eax
3910        db 0x33, 0xdb       ; xor ebx, ebx
3911        db 0x33, 0xc9       ; xor ecx, ecx
3912        mov al, byte [D_070bb]
3913        mov bl, byte [D_070bc]
3914        mov cl, byte [D_070bd]
3915        ret
3916        db 0xbb, 0x70, 0x00, 0x00, 0xbc, 0x70, 0x00, 0x00, 0xbd, 0x70, 0x00, 0x00, 0xbc, 0x70
           , 0x00, 0x00
3917        db 0xbb, 0x70, 0x00, 0x00, 0xbd, 0x70, 0x00, 0x00, 0xbc, 0x70, 0x00, 0x00, 0xbd, 0x70
           , 0x00, 0x00
3918        db 0xbb, 0x70, 0x00, 0x00, 0xbd, 0x70, 0x00, 0x00, 0xbc, 0x70, 0x00, 0x00, 0xbb, 0x70
           , 0x00, 0x00
3919        db 0xbd, 0x70, 0x00, 0x00, 0xbb, 0x70, 0x00, 0x00, 0xbc, 0x70, 0x00, 0x00, 0xbb, 0x70
           , 0x00, 0x00
3920        db 0xbd, 0x70, 0x00, 0x00, 0xbc, 0x70, 0x00, 0x00
```

```
D_070bb:
    db 0x00
D_070bc:
    db 0x00
D_070bd:
    db 0x00
D_070be:
    db 0x00
D_070bf:
    db 0x00
L_070c0:
    add byte [0xff], ah
    and ebx, 0xff
    and ecx, 0xff
    db 0x33, 0xed     ; xor ebp, ebp
    db 0x3a, 0xc3     ; cmp al, bl
    jbe short L_070db
    or ebp, 0x10
L_070db:
    db 0x3a, 0xc1     ; cmp al, cl
    jbe short L_070e2
    or ebp, 0x20
L_070e2:
    db 0x3a, 0xd9     ; cmp bl, cl
    jbe short L_070e9
    or ebp, 0x40
L_070e9:
    lea edx, [ebp + 0x718f]
    mov edx, dword [edx]
    mov byte [edx], al
    lea edx, [ebp + 0x718f]
    mov edx, dword [edx + 4]
    mov byte [edx], bl
    lea edx, [ebp + 0x718f]
    mov edx, dword [edx + 8]
    mov byte [edx], cl
    db 0x33, 0xd2     ; xor edx, edx
    db 0x33, 0xdb     ; xor ebx, ebx
    mov eax, 0xff00
    mov bl, byte [D_07218]
    cmp bl, byte [L_0721a]
    je near L_0720f
    div ebx
    db 0x8b, 0xd8     ; mov ebx, eax
    db 0x33, 0xc0     ; xor eax, eax
    mov al, byte [D_07219]
    mul ebx
    not ah
    mov byte [D_07219], ah
    db 0x33, 0xc0     ; xor eax, eax
    mov al, byte [L_0721a]
    mul ebx
    not ah
    mov byte [D_07217], ah
    db 0x33, 0xdb     ; xor ebx, ebx
    db 0x8a, 0xdc     ; mov bl, ah
    mov eax, 0xff00
    db 0x33, 0xd2     ; xor edx, edx
    div ebx
    db 0x8b, 0xd8     ; mov ebx, eax
    db 0x33, 0xc0     ; xor eax, eax
    mov al, byte [D_07219]
    mul ebx
    db 0x8a, 0xdc     ; mov bl, ah
    not bl
    lea edx, [ebp + 0x719b]
    mov eax, dword [edx]
    db 0x8a, 0xc4     ; mov al, ah
    ror al, 1
```

```
3990        sar al, 7
3991        db 0x32, 0xc3     ; xor al, bl
3992        mov ebx, 6
3993        db 0x33, 0xd2     ; xor edx, edx
3994        db 0x33, 0xc9     ; xor ecx, ecx
3995        div ebx
3996        mov bl, byte [D_07217]
3997        mov cl, byte [D_07218]
3998        ret
3999        db 0x1a, 0x72, 0x00, 0x00, 0x19, 0x72, 0x00, 0x00, 0x18, 0x72, 0x00, 0x00, 0x00, 0x03
          , 0x00, 0x00
4000        db 0x19, 0x72, 0x00, 0x00, 0x1a, 0x72, 0x00, 0x00, 0x18, 0x72, 0x00, 0x00, 0x00, 0x04
          , 0x00, 0x00
4001        db 0x18, 0x72, 0x00, 0x00, 0x19, 0x72, 0x00, 0x00, 0x19, 0x72, 0x00, 0x00, 0x00, 0x00
          , 0x00, 0x00
4002        db 0x18, 0x72, 0x00, 0x00, 0x1a, 0x72, 0x00, 0x00, 0x19, 0x72, 0x00, 0x00, 0x00, 0x05
          , 0x00, 0x00
4003        db 0x1a, 0x72, 0x00, 0x00, 0x18, 0x72, 0x00, 0x00, 0x19, 0x72, 0x00, 0x00, 0x00, 0x02
          , 0x00, 0x00
4004        db 0x18, 0x72, 0x00, 0x00, 0x19, 0x72, 0x00, 0x00, 0x1a, 0x72, 0x00, 0x00, 0x00, 0x00
          , 0x00, 0x00
4005        db 0x19, 0x72, 0x00, 0x00, 0x18, 0x72, 0x00, 0x00, 0x1a, 0x72, 0x00, 0x00, 0x00, 0x01
          , 0x00, 0x00
4006        db 0x18, 0x72, 0x00, 0x00, 0x19, 0x72, 0x00, 0x00, 0x1a, 0x72, 0x00, 0x00, 0x00, 0x00
          , 0x00, 0x00
4007    L_0720f:
4008        db 0x33, 0xc0     ; xor eax, eax
4009        db 0x8b, 0xcb     ; mov ecx, ebx
4010        db 0x33, 0xdb     ; xor ebx, ebx
4011        ret
4012        db 0x00
4013    D_07217:
4014        db 0x00
4015    D_07218:
4016        db 0x00
4017    D_07219:
4018        db 0x00
4019    L_0721a:
4020        add byte [ecx + 0xc000], bh
4021    L_07220:
4022        not dword [edi]
4023        add edi, 4
4024        dec ecx
4025        jne short L_07220
4026        ret
4027    D_07229:
4028        db 0xce, 0xab, 0x34, 0x12
4029    D_0722d:
4030        db 0x91, 0x92, 0x04, 0x03
4031    D_07231:
4032        db 0xa8, 0xd9, 0xfc, 0x67
4033    L_07235:
4034        mov eax, dword [D_07229]
4035        mov ebx, dword [D_0722d]
4036        mov ecx, dword [D_07231]
4037        add eax, 0xe372471
4038        add ebx, 0xbc5ea9
4039        add ecx, 0xc8ade7
4040        db 0x33, 0xc3     ; xor eax, ebx
4041        imul ecx
4042        rol edx, 0xd
4043        db 0x03, 0xd0     ; add edx, eax
4044        rol ecx, 0xe
4045        db 0x2b, 0xd3     ; sub edx, ebx
4046        db 0x33, 0xc2     ; xor eax, edx
4047        db 0x03, 0xd9     ; add ebx, ecx
4048        db 0x33, 0xca     ; xor ecx, edx
4049        db 0x2b, 0xc3     ; sub eax, ebx
4050        rol ebx, 9
```

```
4051        mul ecx
4052        db 0x33, 0xda     ; xor ebx, edx
4053        add eax, 0x1e42ac3
4054        add ebx, 0x5edf04
4055        add ecx, 0x11a2ec1
4056        mov dword [D_07229], ebx
4057        mov dword [D_0722d], ecx
4058        mov dword [D_07231], eax
4059        ret
4060        db 0x02, 0x0d, 0x32, 0x00, 0x04, 0x32, 0x00, 0x0e, 0x02, 0x03, 0x0f, 0x02, 0x03, 0x0f
            , 0x02, 0x03
4061        db 0x0e, 0x02, 0x01, 0x00, 0x02, 0x05, 0x01, 0xff, 0x01, 0x01, 0x00, 0x00, 0x50, 0x04
            , 0x82, 0x03
4062        db 0x0c, 0x01, 0x02, 0x04, 0x28, 0x02, 0x00, 0x02, 0x05, 0x01, 0xff, 0x01, 0x01, 0x02
            , 0x00, 0x50
4063        db 0x04, 0xaa, 0x03, 0x0c, 0x01, 0x01, 0x10, 0x2c, 0x01, 0xc0, 0x12, 0x02, 0x01, 0x08
            , 0x03, 0x03
4064        db 0x12, 0x00, 0x02, 0x05, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x32, 0x0a, 0x03, 0x00
            , 0x13, 0x0b
4065        db 0x01, 0xe6, 0x03, 0x01, 0x00, 0x02, 0x05, 0x01, 0x00, 0x02, 0xff, 0xfe, 0xfe, 0x64
            , 0x04, 0x64
4066        db 0x03, 0x08, 0x03, 0x00, 0x00, 0x02, 0x05, 0xff, 0x01, 0x01, 0x01, 0x03, 0x02, 0x64
            , 0x04, 0x64
4067        db 0x03, 0x08, 0x03, 0x01, 0x00, 0x02, 0x05, 0x02, 0xfe, 0xff, 0x01, 0x01, 0x02, 0x64
            , 0x04, 0x64
4068        db 0x03, 0x08, 0x03, 0x02, 0x00, 0x01, 0x02, 0x00, 0x03, 0x0b, 0x01, 0x7a, 0x03, 0x0f
            , 0x02, 0x03
4069        db 0x0f, 0x02, 0x03, 0x12, 0x12, 0x00, 0x02, 0x09, 0x03, 0x00, 0x0a, 0x01, 0x00, 0x00
            , 0x02, 0x09
4070        db 0x03, 0x01, 0x0a, 0x01, 0x00, 0x00, 0x02, 0x09, 0x03, 0x02, 0x0a, 0x01, 0x00, 0x13
            , 0x0f, 0x02
4071        db 0x03, 0x10, 0x25, 0x00, 0x48, 0x0d, 0x03, 0x01, 0x12, 0x00, 0x02, 0x05, 0x01, 0xff
            , 0x01, 0x02
4072        db 0xff, 0x01, 0x28, 0x04, 0xc3, 0x03, 0x08, 0x03, 0x00, 0x00, 0x02, 0x05, 0xff, 0x01
            , 0xff, 0xff
4073        db 0x01, 0x02, 0x28, 0x0a, 0x03, 0x00, 0x09, 0x02, 0x00, 0x08, 0x01, 0x00, 0x13, 0x01
            , 0x00, 0x01
4074        db 0x02, 0x0e, 0x02, 0x03, 0x0f, 0x02, 0x03, 0x00, 0x02, 0x05, 0xff, 0x01, 0x00, 0xff
            , 0x01, 0x01
4075        db 0x32, 0x04, 0x78, 0x03, 0x0b, 0x01, 0x96, 0x00, 0x00, 0x02, 0x05, 0x02, 0xff, 0x02
            , 0xfe, 0x01
4076        db 0xff, 0x28, 0x04, 0x78, 0x03, 0x0b, 0x01, 0x9b, 0x01, 0x00, 0x02, 0x05, 0x01, 0x01
            , 0xfe, 0xfe
4077        db 0xff, 0xff, 0x28, 0x04, 0x78, 0x03, 0x0b, 0x01, 0x98, 0x02, 0x12, 0x0f, 0x02, 0x03
            , 0x10, 0x32
4078        db 0x00, 0x4c, 0x04, 0x02, 0x00, 0x12, 0x00, 0x02, 0x02, 0x0e, 0x03, 0x03, 0x0e, 0x03
            , 0x03, 0x0e
4079        db 0x03, 0x03, 0x0d, 0x64, 0x03, 0x04, 0xce, 0x03, 0x09, 0x03, 0x00, 0x08, 0x02, 0x00
            , 0x0a, 0x01
4080        db 0x00, 0x13, 0x10, 0xe2, 0xff, 0xe8, 0x03, 0x02, 0x00, 0x0e, 0x02, 0x03, 0x12, 0x04
            , 0xd8, 0x00
4081        db 0x13, 0x00, 0x02, 0x05, 0x02, 0x01, 0xff, 0x01, 0x03, 0x02, 0x41, 0x04, 0xa6, 0x03
            , 0x0c, 0x01
4082        db 0x00, 0x12, 0x00, 0x02, 0x0a, 0x01, 0x00, 0x13, 0x10, 0x2d, 0x00, 0xe8, 0x03, 0x03
            , 0x01, 0x01
4083        db 0x00, 0x01, 0x03, 0x10, 0x27, 0x00, 0x00, 0x08, 0x02, 0x00, 0x0f, 0x03, 0x03, 0x0f
            , 0x03, 0x03
4084        db 0x0f, 0x03, 0x03, 0x09, 0x01, 0x00, 0x01, 0x00, 0x01, 0x0d, 0x00, 0x03, 0x06, 0x00
            , 0x13, 0x03
4085        db 0x00, 0x08, 0x02, 0x02, 0x0e, 0x01, 0x03, 0x08, 0x01, 0x03, 0x0f, 0x03, 0x03, 0x0b
            , 0x02, 0x96
4086        db 0x03, 0x08, 0x04, 0x03, 0x00, 0x01, 0x02, 0x0e, 0x02, 0x03, 0x0f, 0x02, 0x03, 0x00
            , 0x02, 0x05
4087        db 0xff, 0x01, 0x00, 0xff, 0x01, 0x01, 0x32, 0x04, 0x78, 0x03, 0x0b, 0x01, 0x96, 0x00
            , 0x00, 0x02
4088        db 0x05, 0x02, 0xff, 0x02, 0xfe, 0x01, 0xff, 0x28, 0x04, 0x78, 0x03, 0x0b, 0x01, 0x9b
            , 0x01, 0x00
4089        db 0x02, 0x05, 0x01, 0x01, 0xfe, 0xfe, 0xff, 0xff, 0x28, 0x04, 0x78, 0x03, 0x0b, 0x01
            , 0x98, 0x02
```

```
4090        db 0x00, 0x04, 0x09, 0x03, 0x02, 0x0c, 0x01, 0x00, 0x00, 0x03, 0x0c, 0x01, 0x01, 0x00
            , 0x03, 0x0c
4091        db 0x01, 0x02, 0x10, 0x1b, 0x00, 0xdc, 0x05, 0x02, 0x02, 0x12, 0x0d, 0xe6, 0x01, 0x04
            , 0x19, 0x01
4092        db 0x13, 0x11, 0x7e, 0x0a, 0xfb, 0xfb, 0x7e, 0x0a, 0x0a, 0xfb, 0x7e, 0x01, 0x05, 0x01
            , 0x02, 0xff
4093        db 0x02, 0x03, 0x02, 0x64, 0x08, 0x02, 0x00, 0x04, 0x96, 0x00, 0x00, 0x01, 0x05, 0x02
            , 0x01, 0xff
4094        db 0xfe, 0x02, 0x03, 0x64, 0x08, 0x02, 0x01, 0x04, 0x96, 0x01, 0x00, 0x01, 0x05, 0x03
            , 0xff, 0x03
4095        db 0x00, 0x01, 0x04, 0x64, 0x08, 0x02, 0x02, 0x04, 0x96, 0x02, 0x12, 0x0f, 0x01, 0x02
            , 0x0f, 0x01
4096        db 0x02, 0x0f, 0x01, 0x01, 0x13, 0x0f, 0x01, 0x03, 0x00, 0x03, 0x03, 0x88, 0x13, 0x00
            , 0x00, 0x08
4097        db 0x04, 0x00, 0x0f, 0x03, 0x00, 0x0f, 0x03, 0x00, 0x0f, 0x03, 0x00, 0x0f, 0x03, 0x00
            , 0x09, 0x03
4098        db 0x00, 0x0d, 0x50, 0x03, 0x07, 0x0c, 0x01, 0x00, 0x00, 0x03, 0x0c, 0x01, 0x01, 0x00
            , 0x03, 0x0c
4099        db 0x01, 0x02, 0x11, 0x46, 0x1e
4100        db "2<Z2F(d"
4101        db 0x0d, 0xe6, 0x03, 0x0e, 0x02, 0x03, 0x00, 0x02, 0x05, 0x02, 0x01, 0xff, 0x03, 0x00
            , 0x01, 0x3c
4102        db 0x04, 0xbe, 0x03, 0x0c, 0x01, 0x00, 0x00, 0x02, 0x0c, 0x01, 0x01, 0x00, 0x02, 0x0c
            , 0x01, 0x02
4103        db 0x10, 0x04, 0x01, 0xa4, 0x06, 0x02, 0x02, 0x01, 0x05, 0x01, 0x02, 0xff, 0x02, 0x03
            , 0x02, 0x64
4104        db 0x08, 0x02, 0x00, 0x04, 0x96, 0x00, 0x00, 0x01, 0x05, 0x02, 0x01, 0xff, 0xfe, 0x02
            , 0x03, 0x64
4105        db 0x08, 0x02, 0x01, 0x04, 0x96, 0x01, 0x00, 0x01, 0x05, 0x03, 0xff, 0x03, 0x00, 0x01
            , 0x04, 0x64
4106        db 0x08, 0x02, 0x02, 0x04, 0x96, 0x02, 0x12, 0x0f, 0x01, 0x03, 0x0f, 0x01, 0x03, 0x00
            , 0x01, 0x05
4107        db 0x01, 0xff, 0xff, 0x01, 0x01, 0x02, 0x50, 0x04, 0xad, 0x03, 0x0c, 0x02, 0x00, 0x00
            , 0x01, 0x0c
4108        db 0x02, 0x01, 0x00, 0x01, 0x0c, 0x02, 0x02, 0x10, 0x6e, 0x00, 0x34, 0x08, 0x01, 0x02
            , 0x11, 0x50
4109        db 0x0a, 0x1e, 0x0a, 0x46, 0x14, 0x3c, 0x14, 0x5a, 0x01
4110    D_0759c:
4111        db 0x96, 0x43, 0x8b, 0x3c, 0x01, 0x07, 0x04, 0x0e, 0x07
4112        times 18 db 0
4113        db 0x6f, 0x43, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x5c, 0x42, 0x00, 0x00, 0x20, 0x41
            , 0x06, 0x00
4114        db 0x6f, 0x12, 0x83, 0x3a, 0x00, 0x07, 0x00, 0x00, 0x96, 0xc2, 0x00, 0x00, 0x7f, 0x43
            , 0x00, 0x00
4115        db 0x00, 0x00, 0x00, 0x00, 0x20, 0x42, 0x07, 0x05, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
            , 0x00, 0x25
4116        db 0xc4, 0x00, 0x00, 0x00, 0x00, 0x0b, 0x00, 0x0b, 0x00, 0x0b, 0x00, 0x0b, 0x00, 0x0b
            , 0x00, 0x0b
4117        db 0x00, 0x0b, 0x00, 0x09, 0x01, 0x00, 0x00, 0x80, 0x3f, 0x09, 0x02, 0x00, 0x00, 0x00
            , 0x40, 0x09
4118        db 0x03, 0x00, 0x00, 0x40, 0x40, 0x09, 0x04, 0x00, 0x00, 0x80, 0x40, 0x09, 0x05, 0x00
            , 0x00, 0xa0
4119        db 0x40, 0x0a, 0x00, 0x9a, 0x99, 0x99, 0x3e, 0x08, 0x00, 0xcd, 0xcc, 0x4c, 0x3e, 0x0a
            , 0x01, 0x00
4120        db 0x00, 0xc0, 0xbf, 0x08, 0x02, 0x66, 0x66, 0xa6, 0xbf, 0x0a, 0x03, 0x66, 0x66, 0x26
            , 0xc0, 0x08
4121        db 0x04, 0x00, 0x00, 0x40, 0xc0, 0x0a, 0x05, 0xcd, 0xcc, 0x8c, 0x3f, 0x08, 0x06, 0x66
            , 0x66, 0xe6
4122        db 0xbf, 0x0a, 0x07, 0xd8, 0x0f, 0x49, 0x40, 0x01, 0x02, 0x00, 0x09, 0x00, 0x00, 0x16
            , 0x43
4123        times 10 db 0
4124        db 0xc8, 0x42, 0x00, 0x09, 0x00, 0x00, 0x16, 0xc3
4125        times 10 db 0
4126        db 0xc8, 0x42, 0x00, 0x09, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x16, 0x43, 0x00, 0x00
            , 0x48, 0x42
4127        db 0x00, 0x00, 0xc8, 0x42, 0x00, 0x09, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x16, 0xc3
            , 0x00, 0x00
4128        db 0x00, 0x00, 0x00, 0x00, 0xc8, 0x42, 0x06, 0x08, 0x52, 0x49, 0x1d, 0x3a, 0x01, 0x01
            , 0x02, 0x03
```

```
4129        db 0x03, 0x00, 0x0b
4130        times 14 db 0
4131        db 0x2f, 0xc4, 0x07, 0x07, 0x07, 0x07, 0x07, 0x07, 0x07, 0x07, 0x07, 0x07, 0x07
            , 0x07, 0x06
4132        db 0x00, 0xac, 0xc5, 0xa7, 0x38, 0x09, 0x00, 0x66, 0x66, 0x66, 0x3f, 0x06, 0x00, 0xcd
            , 0xaf, 0xe6
4133        db 0x38, 0x0c, 0x03, 0x00, 0x80, 0x0e, 0x44, 0x00, 0x00, 0xfa, 0x43, 0x00, 0x00, 0x40
            , 0x3f, 0x00
4134        db 0x00, 0x60, 0xc1, 0xcd, 0xcc, 0xcc, 0x3e, 0x9a, 0x99, 0x99, 0x3e, 0x9a, 0x99, 0x99
            , 0x3e, 0xcd
4135        db 0xcc, 0x8c, 0x3f, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01, 0x02, 0x03, 0x02, 0x00, 0x0a
4136        times 14 db 0
4137        db 0x4d, 0xc4, 0x06, 0x00, 0x34, 0x80, 0x37, 0x3a, 0x08, 0x00, 0x00, 0x00, 0x80, 0x3f
            , 0x09, 0x00
4138        db 0x9a, 0x99, 0x99, 0x3e, 0x0a, 0x00, 0x33, 0x33, 0x33, 0x3f, 0x06, 0x00, 0x34, 0x80
            , 0x37, 0x3a
4139        db 0x01, 0x07, 0x02, 0x01, 0x01, 0x04, 0x0a, 0x0a, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80
            , 0x3b, 0xc4
4140        times 10 db 0
4141        db 0x25, 0x44, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x8c, 0x42, 0x00, 0x00, 0xf0, 0x41
            , 0x0b, 0x01
4142        db 0x0b, 0x01, 0x0b, 0x01, 0x0b, 0x01, 0x0b, 0x01, 0x06, 0x04, 0xac, 0xc5, 0xa7, 0x38
            , 0x06, 0x05
4143        db 0x62, 0xbe, 0xbc, 0x38, 0x06, 0x06, 0x17, 0xb7, 0x51, 0xb9, 0x09, 0x02, 0xcd, 0xcc
            , 0xcc, 0x3d
4144        db 0x09, 0x03, 0xcd, 0xcc, 0xcc, 0x3d, 0x09, 0x04, 0xcd, 0xcc, 0xcc, 0x3d, 0x09, 0x05
            , 0xcd, 0xcc
4145        db 0xcc, 0x3d, 0x09, 0x06, 0xcd, 0xcc, 0xcc, 0x3d, 0x0a, 0x02, 0x66, 0x66, 0x06, 0x40
            , 0x08, 0x03
4146        db 0x33, 0x33, 0x13, 0x40, 0x0a, 0x04, 0xc3, 0xf5, 0xc8, 0x3f, 0x08, 0x05, 0x66, 0x66
            , 0xa6, 0xbf
4147        db 0x0a, 0x06, 0xc3, 0xf5, 0xc8, 0xbf, 0x09, 0x03, 0xc3, 0xf5, 0xc8, 0x3f, 0x09, 0x04
            , 0xc3, 0xf5
4148        db 0xc8, 0x3f, 0x09, 0x05, 0xc3, 0xf5, 0xc8, 0x3f, 0x09, 0x06, 0xc3, 0xf5, 0xc8, 0x3f
            , 0x00, 0x09
4149        times 14 db 0
4150        db 0xc8, 0x42, 0x01, 0x02, 0x02, 0x02, 0x04, 0x0e, 0x0f, 0x18, 0x00, 0x00, 0x96, 0x43
            , 0x00, 0x00
4151        db 0xc8, 0x42, 0x00, 0x00, 0xa0, 0x41, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x20, 0x42
            , 0x00, 0x00
4152        db 0x40, 0x40, 0x0a, 0x08, 0xc3, 0xf5, 0xc8, 0x3f, 0x06, 0x08, 0x6f, 0x12, 0x03, 0x3b
            , 0x0a, 0x08
4153        db 0x85, 0xeb, 0x11, 0xbf, 0x08, 0x08, 0xa4, 0x70, 0x3d, 0x3f, 0x09, 0x08, 0x9a, 0x99
            , 0x19, 0x3f
4154        db 0x0c, 0x03, 0x00, 0x80, 0x04, 0x44, 0x00, 0x00, 0xff, 0x43, 0x00, 0x00, 0x00, 0x3f
            , 0x00, 0x00
4155        db 0x80, 0xc0, 0xdf, 0x4f, 0x8d, 0x3e, 0x64, 0x3b, 0x9f, 0x3e, 0xaa, 0xf1, 0xd2, 0x3e
            , 0x5e, 0xba
4156        db 0x99, 0x3f, 0x8f, 0xc2, 0xf5, 0x3c, 0x01, 0x06, 0x04, 0x08, 0x10
4157        times 18 db 0
4158        db 0xc8, 0x44, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x48, 0x43, 0x00, 0x00, 0x70, 0x42
            , 0x06, 0x00
4159        db 0x6f, 0x12, 0x03, 0x3a, 0x05, 0x00, 0x00, 0x00, 0xe1, 0x44, 0x00, 0x00, 0x80, 0xc4
            , 0x00, 0x00
4160        db 0x00, 0x00, 0x0b, 0x00, 0x0b, 0x00, 0x0b, 0x00, 0x0b, 0x00, 0x0b, 0x00, 0x0b, 0x00
            , 0x0b, 0x00
4161        db 0x09, 0x01, 0x14, 0xae, 0x47, 0x3f, 0x09, 0x02, 0xc3, 0xf5, 0xc8, 0x3f, 0x09, 0x03
            , 0x3d, 0x0a
4162        db 0x17, 0x40, 0x09, 0x04, 0xc3, 0xf5, 0x48, 0x40, 0x09, 0x05, 0x48, 0xe1, 0x7a, 0x40
            , 0x09, 0x06
4163        db 0x52, 0xb8, 0x96, 0x40, 0x09, 0x07, 0x14, 0xae, 0xaf, 0x40, 0x02, 0x02, 0x02, 0x01
            , 0x02, 0x00
4164        db 0x0a
4165        times 14 db 0
4166        db 0x16, 0xc5, 0x0c, 0x03, 0x00, 0x80, 0xa2, 0x45, 0x00, 0x00, 0xfa, 0x44, 0x00, 0x00
            , 0x40, 0x3f
4167        db 0x00, 0x00, 0x60, 0xc1, 0xcd, 0xcc, 0xcc, 0x3e, 0x9a, 0x99, 0x99, 0x3e, 0x9a, 0x99
            , 0x99, 0x3e
4168        db 0xcd, 0xcc, 0x8c, 0x3f, 0x00, 0x00, 0x00, 0x00, 0x01, 0x02, 0x02, 0x05, 0x07, 0x0e
```

```
        , 0x29, 0x1e
4169        db 0x00, 0x00, 0x2a, 0xc4, 0x00, 0x00, 0xe1, 0x43, 0x00, 0x00, 0x16, 0x43, 0x00, 0x00
        , 0x40, 0x40
4170        db 0x00, 0x00, 0x16, 0x43, 0x00, 0x00, 0x00, 0x40, 0x08, 0x00, 0xc3, 0xf5, 0xc8, 0x3f
        , 0x06, 0x00
4171        db 0x6f, 0x12, 0x83, 0x3a, 0x08, 0x00, 0xc3, 0xf5, 0xc8, 0xbf, 0x05, 0x00, 0x00, 0x00
        , 0x00, 0x00
4172        db 0x00, 0x00, 0xa0, 0x42, 0x00, 0x00, 0x00, 0x00, 0x0c, 0x03, 0x00, 0x00, 0x43, 0x44
        , 0x00, 0x80
4173        db 0x31, 0x44, 0xcd, 0xcc, 0x4c, 0x3f, 0x00, 0x00, 0x70, 0x41, 0x14, 0xae, 0x07, 0x3f
        , 0x1f, 0x85
4174        db 0xeb, 0x3e, 0x27, 0x31, 0x28, 0x3f, 0xb8, 0x1e, 0x85, 0x3e, 0x29, 0x5c, 0x8f, 0xbd
4175    D_07951:
4176        db 0x00, 0x00, 0x00, 0x00
4177    D_07955:
4178        db 0x00, 0x00, 0x00, 0x00
4179    L_07959:
4180        mov esi, dword [D_07955]
4181        mov edi, D_07a2b
4182        mov ecx, 9
4183        rep movsd
4184        fld dword [D_07951]
4185        fstp dword [D_02000]
4186        fld1
4187        fstp dword [D_02004]
4188        mov edi, D_0610e
4189        call L_079ac
4190        fld dword [D_07951]
4191        fadd dword [D_07a37]
4192        fstp dword [D_02000]
4193        fld dword [D_07a33]
4194        fstp dword [D_02004]
4195        mov edi, D_0611a
4196    L_079ac:
4197        fld dword [D_02000]
4198        fld st0
4199        fmul dword [D_07a3b]
4200        fld st0
4201        fsin
4202        fmul dword [D_07a43]
4203        faddp st1
4204        fld st0
4205        fcos
4206        fstp dword [edi]
4207        fsin
4208        fstp dword [edi + 8]
4209        fmul dword [D_07a3f]
4210        fsin
4211        fmul dword [D_07a47]
4212        fld st0
4213        fcos
4214        fld st0
4215        fmul dword [edi]
4216        fmul dword [D_07a2b]
4217        fmul dword [D_02004]
4218        fstp dword [edi]
4219        fmul dword [edi + 8]
4220        fmul dword [D_07a2b]
4221        fmul dword [D_02004]
4222        fstp dword [edi + 8]
4223        fsin
4224        fmul dword [D_07a2f]
4225        fmul dword [D_02004]
4226        fstp dword [edi + 4]
4227        fld dword [D_02000]
4228        fmul dword [D_07a4b]
4229        fstp dword [D_0612a]
4230        ret
4231    D_07a2b:
```

```
4232        db 0x00, 0x00, 0x00, 0x00
4233    D_07a2f:
4234        db 0x00, 0x00, 0x00, 0x00
4235    D_07a33:
4236        db 0x00, 0x00, 0x00, 0x00
4237    D_07a37:
4238        db 0x00, 0x00, 0x00, 0x00
4239    D_07a3b:
4240        db 0x00, 0x00, 0x00, 0x00
4241    D_07a3f:
4242        db 0x00, 0x00, 0x00, 0x00
4243    D_07a43:
4244        db 0x00, 0x00, 0x00, 0x00
4245    D_07a47:
4246        db 0x00, 0x00, 0x00, 0x00
4247    D_07a4b:
4248        db 0x00, 0x00, 0x00, 0x00
4249    L_07a4f:
4250        mov edi, dword [BufPtr_09a81]
4251        mov dword [D_09a31], edi
4252        mov ebp, 0x100
4253        fld dword [D_07ae2]
4254        fstp dword [D_07aea]
4255    L_07a6c:
4256        mov ecx, 0x100
4257        fld dword [D_07ae2]
4258        fstp dword [D_07ae6]
4259        fld dword [D_07aea]
4260        fmul st0
4261    L_07a85:
4262        fld1
4263        fadd dword [D_07ae6]
4264        fst dword [D_07ae6]
4265        fmul st0
4266        fadd st1
4267        fsqrt
4268        fchs
4269        fadd dword [D_07aee]
4270        fmul st0
4271        fmul dword [D_07af2]
4272        fld st0
4273        fsqrt
4274        fmulp st1
4275        fmul dword [D_07af6]
4276        fmul st0
4277        fmul dword [D_07af2]
4278        fistp dword [D_02000]
4279        mov al, byte [D_02000]
4280        mov byte [edi], al
4281        inc edi
4282        dec ecx
4283        jne short L_07a85
4284        faddp st0
4285        fld1
4286        fadd dword [D_07aea]
4287        fstp dword [D_07aea]
4288        dec ebp
4289        jne short L_07a6c
4290        ret
4291    D_07ae2:
4292        db 0x00, 0x00, 0x00, 0xc3
4293    D_07ae6:
4294        db 0x00, 0x00, 0x00, 0x00
4295    D_07aea:
4296        db 0x00, 0x00, 0x00, 0x00
4297    D_07aee:
4298        db 0x00, 0x00, 0x7f, 0x43
4299    D_07af2:
4300        db 0x00, 0x00, 0x80, 0x3b
```

```
D_07af6:
    db 0x00, 0x00, 0x80, 0x3d, 0x87, 0xdb
D_07afc:
    db 0x00, 0x00, 0x00, 0x00
D_07b00:
    db 0x00, 0x00, 0x00, 0x00
D_07b04:
    db 0x00, 0x00, 0x00, 0x00
D_07b08:
    db 0x00, 0x00, 0x00, 0x00
D_07b0c:
    db 0x00, 0x00, 0x00, 0x00
D_07b10:
    db 0x00, 0x00, 0x00, 0x00
D_07b14:
    db 0x00, 0x00, 0x00, 0x00
D_07b18:
    db 0x00, 0x00, 0x00, 0x00
D_07b1c:
    db 0x00, 0x00, 0x80, 0x42
D_07b20:
    db 0x00, 0x00, 0x00, 0x00
D_07b24:
    db 0x00, 0x00, 0x00, 0x00
D_07b28:
    db 0x01
D_07b29:
    db 0x35, 0x7e, 0x00, 0x00
D_07b2d:
    db 0x00, 0x00, 0x00, 0x00
D_07b31:
    db 0x00, 0x00, 0x00, 0x00
L_07b35:
    mov edi, dword [D_1bd04]
    imul eax, eax, 0x50
    db 0x03, 0xf8     ; add edi, eax
    mov dword [edi + 0x4c], 0
    mov eax, dword [D_07c40]
    mov dword [edi + 0x40], eax
    fldpi
    fld dword [D_07c78]
    fmul st1
    fdiv dword [D_07c68]
    fstp dword [D_07c44]
    fldpi
    fld dword [D_07c78]
    fadd dword [D_07c78]
    fadd dword [D_07c6c]
    fmul st1
    fdiv dword [D_07c68]
    fstp dword [D_07c48]
    fld dword [D_07c44]
    fsin
    fstp dword [D_07c4c]
    fld dword [D_07c44]
    fcos
    fstp dword [D_07c54]
    fld dword [D_07c48]
    fsin
    fstp dword [D_07c50]
    fld dword [D_07c48]
    fcos
    fstp dword [D_07c58]
    fld dword [D_07c54]
    fstp dword [edi + 0x10]
    fldz
    fstp dword [edi + 0x14]
    fld dword [D_07c4c]
    fchs
```

```
4370        fstp dword [edi + 0x18]
4371        fld dword [D_07c4c]
4372        fmul dword [D_07c50]
4373        fstp dword [edi + 0x1c]
4374        fld dword [D_07c58]
4375        fstp dword [edi + 0x20]
4376        fld dword [D_07c50]
4377        fmul dword [D_07c54]
4378        fstp dword [edi + 0x24]
4379        fld dword [D_07c58]
4380        fmul dword [D_07c4c]
4381        fstp dword [edi + 0x28]
4382        fld dword [D_07c50]
4383        fchs
4384        fstp dword [edi + 0x2c]
4385        fld dword [D_07c54]
4386        fmul dword [D_07c58]
4387        fstp dword [edi + 0x30]
4388        fldz
4389        fst dword [edi + 0x34]
4390        fild dword [D_07c74]
4391        fstp dword [edi + 0x38]
4392        fild dword [D_07c5c]
4393        fstp dword [edi + 0x3c]
4394        ret
4395    D_07c40:
4396        db 0x00, 0x00, 0xb4, 0x43
4397    D_07c44:
4398        db 0x00, 0x00, 0x00, 0x00
4399    D_07c48:
4400        db 0x00, 0x00, 0x00, 0x00
4401    D_07c4c:
4402        db 0x00, 0x00, 0x00, 0x00
4403    D_07c50:
4404        db 0x00, 0x00, 0x00, 0x00
4405    D_07c54:
4406        db 0x00, 0x00, 0x00, 0x00
4407    D_07c58:
4408        db 0x00, 0x00, 0x00, 0x00
4409    D_07c5c:
4410        db 0x00, 0x08, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x80, 0x00, 0x00, 0x00
4411    D_07c68:
4412        db 0x00, 0x00, 0x00, 0x47
4413    D_07c6c:
4414        db 0x00, 0x00, 0x80, 0x45
4415    D_07c70:
4416        db 0x00, 0x00, 0x00, 0x42
4417    D_07c74:
4418        db 0x00, 0x00, 0x00, 0x00
4419    D_07c78:
4420        db 0x00, 0x00, 0x00, 0x00
4421    L_07c7c:
4422        mov edi, 0xbcf0
4423        mov ecx, D_02000
4424        mov ebx, 0xfffff000
4425    L_07c8b:
4426        db 0x8b, 0xc3     ; mov eax, ebx
4427        imul eax
4428        mov dword [edi], eax
4429        inc ebx
4430        add edi, 4
4431        dec ecx
4432        jne short L_07c8b
4433        ret
4434    L_07c99:
4435        call L_07c7c
4436        call L_08e84
4437        mov esi, 0x23448
4438        mov edi, dword [BufPtr_09aad]
```

```
4439        and edi, 0xffffff00
4440        add edi, 0x100
4441        mov dword [BufPtr_09aad], edi
4442        mov ecx, 0x100
4443        mov byte [D_07dfd], 0
4444    L_07ccc:
4445        push esi
4446        push edi
4447        mov bl, byte [esi]
4448        and ebx, 7
4449        mov dword [edi], ebx
4450        add edi, 4
4451        and ebx, 7
4452        lea ebx, [ebx + ebx*2]
4453        db 0x0b, 0xdb       ; or ebx, ebx
4454        je near L_07db0
4455        push ecx
4456        mov cl, 4
4457    L_07ce9:
4458        movzx eax, byte [esi]
4459        shr eax, cl
4460        and al, 0xf
4461        add cl, 4
4462        and cl, 7
4463        jne short L_07cf9
4464        inc esi
4465    L_07cf9:
4466        mov eax, dword [eax*4 + D_07dcd]
4467    L_07d00:
4468        mov dword [edi], eax
4469        add edi, 4
4470        dec ebx
4471        jne short L_07ce9
4472        pop ecx
4473        db 0x8b, 0xd7       ; mov edx, edi
4474        mov dword [edi], 0
4475        add edi, 4
4476        mov al, byte [D_07dfd]
4477        and al, 0xf0
4478        cmp al, 0xf0
4479        je short L_07d2e
4480        db 0x0a, 0xc0       ; or al, al
4481        je short L_07d2e
4482        mov dword [edi], L_082f4
4483        inc dword [edx]
4484        add edi, 4
4485    L_07d2e:
4486        mov al, byte [D_07dfd]
4487        and al, 0xf
4488        cmp al, 0xf
4489        je short L_07d48
4490        db 0x0a, 0xc0       ; or al, al
4491        je short L_07d48
4492        mov dword [edi], L_08324
4493        inc dword [edx]
4494        add edi, 4
4495    L_07d48:
4496        mov al, byte [D_07dfd]
4497        and al, 0x33
4498        cmp al, 0x33
4499        je short L_07d62
4500        db 0x0a, 0xc0       ; or al, al
4501        je short L_07d62
4502        mov dword [edi], L_082bb
4503        inc dword [edx]
4504        add edi, 4
4505    L_07d62:
4506        mov al, byte [D_07dfd]
4507        and al, 0xcc
```

```
4508        cmp al, 0xcc
4509        je short L_07d7c
4510        db 0x0a, 0xc0     ; or al, al
4511        je short L_07d7c
4512        mov dword [edi], L_0828a
4513        inc dword [edx]
4514        add edi, 4
4515    L_07d7c:
4516        mov al, byte [D_07dfd]
4517        and al, 0x55
4518        cmp al, 0x55
4519        je short L_07d96
4520        db 0x0a, 0xc0     ; or al, al
4521        je short L_07d96
4522        mov dword [edi], L_0822e
4523        inc dword [edx]
4524        add edi, 4
4525    L_07d96:
4526        mov al, byte [D_07dfd]
4527        and al, 0xaa
4528        cmp al, 0xaa
4529        je short L_07db0
4530        db 0x0a, 0xc0     ; or al, al
4531        je short L_07db0
4532        mov dword [edi], L_0825c
4533        inc dword [edx]
4534        add edi, 4
4535    L_07db0:
4536        pop edi
4537        pop esi
4538        inc byte [D_07dfd]
4539        add esi, 8
4540        add edi, 0x80
4541        dec ecx
4542        jne near L_07ccc
4543        ret
4544        db 0x00, 0x00, 0x00, 0x00
4545    D_07dcd:
4546        db 0x55, 0x83, 0x00, 0x00, 0xb7, 0x83, 0x00, 0x00, 0x19, 0x84, 0x00, 0x00, 0x8e, 0x84
            , 0x00, 0x00
4547        db 0x03, 0x85, 0x00, 0x00, 0x62, 0x85, 0x00, 0x00, 0xd7, 0x85, 0x00, 0x00, 0x4c, 0x86
            , 0x00, 0x00
4548        db 0xc7, 0x86, 0x00, 0x00, 0x30, 0x87, 0x00, 0x00, 0xae, 0x87, 0x00, 0x00, 0x29, 0x88
            , 0x00, 0x00
4549    D_07dfd:
4550        db 0x00
4551    L_07dfe:
4552        push ebx
4553        push ecx
4554        push edx
4555        add ebx, 0x400
4556        add ecx, 0x400
4557        add edx, 0x400
4558        shr ebx, 6
4559        shr ecx, 6
4560        shr edx, 6
4561        db 0x8b, 0xf2     ; mov esi, edx
4562        shl ebx, 0xa
4563        shl ecx, 5
4564        db 0x03, 0xf3     ; add esi, ebx
4565        db 0x03, 0xf1     ; add esi, ecx
4566        shl esi, 4
4567        add esi, dword [BufPtr_09aa9]
4568        pop edx
4569        pop ecx
4570        pop ebx
4571        ret
4572    L_07e35:
4573        mov dword [D_07b2d], L_088b0
```

```
4574        mov dword [D_07b31], L_08a68
4575        mov ecx, 0xfffffd00
4576    L_07e4e:
4577        mov ebx, 0xfffffd00
4578        mov edx, 0xfffffd00
4579        call L_07dfe
4580    L_07e5d:
4581        mov edx, 0xfffffd00
4582        push esi
4583    L_07e63:
4584        call L_08085
4585        add esi, 0x20
4586        add edx, 0x80
4587        cmp edx, 0x300
4588        jne short L_07e63
4589        pop esi
4590        add esi, 0x8000
4591        add ebx, 0x80
4592        cmp ebx, 0x300
4593        jne short L_07e5d
4594        add ecx, 0x80
4595        cmp ecx, 0x300
4596        jne short L_07e4e
4597        ret
4598    L_07e9d:
4599        mov dword [D_07b2d], L_08c94
4600        mov dword [D_07b31], L_08d3a
4601        mov edi, 0x14
4602        xchg ebx, ebx
4603    L_07eb8:
4604        push edi
4605        call L_07ec5
4606        pop edi
4607        sub edi, 4
4608        jns short L_07eb8
4609        ret
4610    L_07ec5:
4611        mov ebx, dword [edi + 0x8c04]
4612        and ebx, 0xffffffc0
4613        mov ecx, dword [edi + 0x8c1c]
4614        and ecx, 0xffffffc0
4615        mov edx, dword [edi + 0x8c34]
4616        and edx, 0xffffffc0
4617        cmp ebx, 0xffffffc00
4618        jle near L_07f76
4619        cmp ecx, 0xffffffc00
4620        jle near L_07f76
4621        cmp edx, 0xffffffc00
4622        jle short L_07f76
4623        cmp ebx, 0x380
4624        jge short L_07f76
4625        cmp ecx, 0x380
4626        jge short L_07f76
4627        cmp edx, 0x380
4628        jge short L_07f76
4629        call L_07dfe
4630        push ebx
4631        push esi
4632    L_07f1f:
4633        add ebx, 0x40
4634        add esi, 0x4000
4635        cmp ebx, 0x380
4636        jge short L_07f3d
4637        call L_088a4
4638        mov byte [esi], al
4639        test al, 0x80
4640        je short L_07f66
4641        jmp short L_07f1f
4642    L_07f3d:
```

```
4643        pop esi
4644        pop ebx
4645        push ebx
4646        push esi
4647    L_07f41:
4648        sub ebx, 0x40
4649        sub esi, 0x4000
4650        cmp ebx, 0xfffffc00
4651        jle short L_07f74
4652        call L_088a4
4653        mov byte [esi], al
4654        test al, 0x80
4655        jne short L_07f41
4656        add ebx, 0x40
4657        add esi, 0x4000
4658    L_07f66:
4659        sub ebx, 0x40
4660        sub esi, 0x4000
4661        call L_08085
4662    L_07f74:
4663        pop esi
4664        pop ebx
4665    L_07f76:
4666        ret
4667    L_07f77:
4668        wait
4669        fninit
4670        wait
4671        fnstcw word [D_02000]
4672        wait
4673        and word [D_02000], 0xfcff
4674        fldcw word [D_02000]
4675        push eax
4676        mov eax, 0xffffffff
4677        mov ecx, 0x20000
4678        mov edi, dword [BufPtr_09aa9]
4679        rep stosd
4680        mov edi, dword [D_1bd00]
4681        mov eax, dword [D_1bd10]
4682        imul eax, eax, 0x40
4683        db 0x03, 0xf8      ; add edi, eax
4684        mov dword [D_07b04], edi
4685        mov dword [D_07b00], edi
4686        mov ebp, dword [D_1bcfc]
4687        mov eax, dword [D_1bd14]
4688        imul eax, eax, 0x1c
4689        db 0x03, 0xe8      ; add ebp, eax
4690        mov dword [D_07b0c], ebp
4691        mov dword [D_07b08], ebp
4692        call dword [D_07b29]
4693        pop eax
4694        mov edi, dword [D_1bd04]
4695        imul eax, eax, 0x50
4696        db 0x03, 0xf8      ; add edi, eax
4697        db 0x8b, 0xc5      ; mov eax, ebp
4698        sub eax, dword [D_07b08]
4699        cdq
4700        mov ebx, 0x1c
4701        idiv ebx
4702        mov dword [D_07b20], eax
4703        mov dword [edi + 4], eax
4704        mov eax, dword [D_07b04]
4705        sub eax, dword [D_07b00]
4706        cdq
4707        mov ebx, 0x40
4708        idiv ebx
4709        mov dword [D_07b24], eax
4710        mov dword [edi + 0xc], eax
4711        mov eax, dword [D_1bd10]
```

```
4712        mov dword [edi + 8], eax
4713        mov eax, dword [D_1bd14]
4714        mov dword [edi], eax
4715        mov eax, dword [D_08079]
4716        mov ebx, dword [D_08075]
4717        mov ecx, dword [D_0807d]
4718        mov dword [edi + 0x10], eax
4719        mov dword [edi + 0x14], ebx
4720        mov dword [edi + 0x18], ebx
4721        mov dword [edi + 0x1c], ebx
4722        mov dword [edi + 0x20], eax
4723        mov dword [edi + 0x24], ebx
4724        mov dword [edi + 0x28], ebx
4725        mov dword [edi + 0x2c], ebx
4726        mov dword [edi + 0x30], eax
4727        mov dword [edi + 0x34], ebx
4728        mov dword [edi + 0x38], ebx
4729        mov dword [edi + 0x3c], ecx
4730        mov dword [edi + 0x4c], 0
4731        mov eax, dword [D_08081]
4732        mov dword [edi + 0x40], eax
4733        ret
4734    D_08075:
4735        db 0x00, 0x00, 0x00, 0x00
4736    D_08079:
4737        db 0x00, 0x00, 0x80, 0x3f
4738    D_0807d:
4739        db 0x00, 0x00, 0x00, 0x45
4740    D_08081:
4741        db 0x00, 0x00, 0xb4, 0x43
4742    L_08085:
4743        cmp dword [esi], -1
4744        jne short L_08091
4745        call L_088a4
4746        mov byte [esi], al
4747    L_08091:
4748        add ebx, 0x40
4749        cmp dword [esi + 0x4000], -1
4750        jne short L_080a8
4751        call L_088a4
4752        mov byte [esi + 0x4000], al
4753    L_080a8:
4754        add ecx, 0x40
4755        cmp dword [esi + 0x4200], -1
4756        jne short L_080bf
4757        call L_088a4
4758        mov byte [esi + 0x4200], al
4759    L_080bf:
4760        add edx, 0x40
4761        cmp dword [esi + 0x4210], -1
4762        jne short L_080d6
4763        call L_088a4
4764        mov byte [esi + 0x4210], al
4765    L_080d6:
4766        sub ebx, 0x40
4767        cmp dword [esi + 0x210], -1
4768        jne short L_080ed
4769        call L_088a4
4770        mov byte [esi + 0x210], al
4771    L_080ed:
4772        sub edx, 0x40
4773        cmp dword [esi + 0x200], -1
4774        jne short L_08104
4775        call L_088a4
4776        mov byte [esi + 0x200], al
4777    L_08104:
4778        sub ecx, 0x40
4779        add ebx, 0x40
4780        add edx, 0x40
```

```
4781        cmp dword [esi + 0x4010], -1
4782        jne short L_08121
4783        call L_088a4
4784        mov byte [esi + 0x4010], al
4785    L_08121:
4786        sub ebx, 0x40
4787        cmp dword [esi + 0x10], -1
4788        jne short L_08132
4789        call L_088a4
4790        mov byte [esi + 0x10], al
4791    L_08132:
4792        sub edx, 0x40
4793        mov byte [esi + 3], 0
4794        mov dword [D_07b10], ebx
4795        mov dword [D_07b14], ecx
4796        mov dword [D_07b18], edx
4797        push ebx
4798        push ecx
4799        push edx
4800        mov al, byte [esi]
4801        mov ah, byte [esi + 0x4000]
4802        mov cl, byte [esi + 0x200]
4803        mov ch, byte [esi + 0x4200]
4804        and eax, 0x8080
4805        and ecx, 0x8080
4806        shr eax, 6
4807        shr ecx, 4
4808        shr al, 1
4809        shr cl, 1
4810        db 0x8a, 0xd9      ; mov bl, cl
4811        mov dl, byte [esi + 0x210]
4812        db 0x0a, 0xdd      ; or bl, ch
4813        mov dh, byte [esi + 0x4210]
4814        db 0x0a, 0xd8      ; or bl, al
4815        db 0x0a, 0xdc      ; or bl, ah
4816        mov al, byte [esi + 0x10]
4817        mov ah, byte [esi + 0x4010]
4818        and eax, 0x8080
4819        and edx, 0x8080
4820        shr eax, 2
4821        db 0x0a, 0xde      ; or bl, dh
4822        shr al, 1
4823        shr dl, 1
4824        db 0x0a, 0xd8      ; or bl, al
4825        db 0x0a, 0xdc      ; or bl, ah
4826        db 0x0a, 0xda      ; or bl, dl
4827        mov byte [L_08354], 0
4828        and ebx, 0xff
4829        mov edi, dword [BufPtr_09aad]
4830        shl ebx, 7
4831        db 0x03, 0xfb      ; add edi, ebx
4832        mov ecx, dword [edi]
4833        db 0x0b, 0xc9      ; or ecx, ecx
4834        je short L_0822a
4835        add edi, 4
4836        test ecx, 8
4837        je short L_081e3
4838        mov byte [L_08354], 0x40
4839        and ecx, 7
4840        je short L_0822a
4841    L_081e3:
4842        call dword [edi]
4843        mov dword [ds:ebp + 8], ebx
4844        call dword [edi + 4]
4845        mov dword [ds:ebp + 4], ebx
4846        call dword [edi + 8]
4847        mov dword [ds:ebp], ebx
4848        mov al, byte [D_07b28]
4849        mov byte [ds:ebp + 0xe], al
```

```
4850        mov byte [ds:ebp + 0xf], 0
4851        mov byte [ds:ebp + 0xd], 3
4852        mov al, byte [L_08354]
4853        mov byte [ds:ebp + 0xc], al
4854        add ebp, 0x1c
4855        add edi, 0xc
4856        dec ecx
4857        jne short L_081e3
4858        pop edx
4859        pop ecx
4860        pop ebx
4861        mov eax, dword [edi]
4862    L_08221:
4863        add edi, 4
4864        jmp dword [edi]
4865    L_08226:
4866        dec eax
4867        jne short L_08221
4868        ret
4869    L_0822a:
4870        pop edx
4871        pop ecx
4872        pop ebx
4873        ret
4874    L_0822e:
4875        cmp ebx, 0xfffffc00
4876        jle short L_08226
4877        cmp byte [esi - 0x3ffd], 0
4878        je short L_08226
4879        sub ebx, 0x40
4880        sub esi, 0x4000
4881        push eax
4882        push edi
4883        call L_08085
4884        pop edi
4885        pop eax
4886        add ebx, 0x40
4887        add esi, 0x4000
4888        jmp short L_08226
4889    L_0825c:
4890        cmp ebx, 0x380
4891        jge short L_08226
4892        cmp byte [esi + 0x4003], 0
4893        je short L_08226
4894        add ebx, 0x40
4895        add esi, 0x4000
4896        push eax
4897        push edi
4898        call L_08085
4899        pop edi
4900        pop eax
4901        sub ebx, 0x40
4902        sub esi, 0x4000
4903        jmp short L_08226
4904    L_0828a:
4905        cmp ecx, 0x380
4906        jge short L_08226
4907        cmp byte [esi + 0x203], 0
4908        je short L_08226
4909        add ecx, 0x40
4910        add esi, 0x200
4911        push eax
4912        push edi
4913        call L_08085
4914        pop edi
4915        pop eax
4916        sub ecx, 0x40
4917        sub esi, 0x200
4918        jmp near L_08226
```

```
4919  L_082bb:
4920      cmp ecx, 0xfffffc00
4921      jle near L_08226
4922      cmp byte [esi - 0x1fd], 0
4923      je near L_08226
4924      sub ecx, 0x40
4925      sub esi, 0x200
4926      push eax
4927      push edi
4928      call L_08085
4929      pop edi
4930      pop eax
4931      add ecx, 0x40
4932      add esi, 0x200
4933      jmp near L_08226
4934  L_082f4:
4935      cmp edx, 0x380
4936      jge near L_08226
4937      cmp byte [esi + 0x13], 0
4938      je near L_08226
4939      add edx, 0x40
4940      add esi, 0x10
4941      push eax
4942      push edi
4943      call L_08085
4944      pop edi
4945      pop eax
4946      sub edx, 0x40
4947      sub esi, 0x10
4948      jmp near L_08226
4949  L_08324:
4950      cmp edx, 0xfffffc00
4951      jle near L_08226
4952      cmp byte [esi - 0xd], 0
4953      je near L_08226
4954      sub edx, 0x40
4955      sub esi, 0x10
4956      push eax
4957      push edi
4958      call L_08085
4959      pop edi
4960      pop eax
4961      add edx, 0x40
4962      add esi, 0x10
4963      jmp near L_08226
4964  L_08354:
4965      add byte [ebx + 0x75ff047e], al
4966      pop eax
4967      db 0x33, 0xd2     ; xor edx, edx
4968      movzx ebx, byte [esi + 0x4000]
4969      mov dl, byte [esi]
4970      db 0x2b, 0xda     ; sub ebx, edx
4971      mov eax, 0x80
4972      db 0x2b, 0xc2     ; sub eax, edx
4973      shl eax, 6
4974      cdq
4975      idiv ebx
4976      mov ebx, dword [D_07b04]
4977      db 0x8b, 0xd0     ; mov edx, eax
4978      add edx, dword [D_07b10]
4979      mov dword [D_07afc], edx
4980      fild dword [D_07afc]
4981      fstp dword [ebx]
4982      fild dword [D_07b14]
4983      fstp dword [ebx + 4]
4984      fild dword [D_07b18]
4985      fstp dword [ebx + 8]
4986      call L_088aa
4987      mov dword [esi + 4], ebx
```

```
4988        add dword [D_07b04], 0x40
4989        ret
4990        mov ebx, dword [esi + 4]
4991        ret
4992        cmp dword [esi + 8], -1
4993        jne short L_08415
4994        db 0x33, 0xd2      ; xor edx, edx
4995        mov dl, byte [esi]
4996        movzx ebx, byte [esi + 0x200]
4997        db 0x2b, 0xda      ; sub ebx, edx
4998        mov eax, 0x80
4999        db 0x2b, 0xc2      ; sub eax, edx
5000        shl eax, 6
5001        cdq
5002        idiv ebx
5003        mov ebx, dword [D_07b04]
5004        db 0x8b, 0xd0      ; mov edx, eax
5005        add edx, dword [D_07b14]
5006        mov dword [D_07afc], edx
5007        fild dword [D_07afc]
5008        fstp dword [ebx + 4]
5009        fild dword [D_07b10]
5010        fstp dword [ebx]
5011        fild dword [D_07b18]
5012        fstp dword [ebx + 8]
5013        call L_088aa
5014        mov dword [esi + 8], ebx
5015        add dword [D_07b04], 0x40
5016        ret
5017    L_08415:
5018        mov ebx, dword [esi + 8]
5019        ret
5020        cmp dword [esi + 0x4008], -1
5021        jne short L_08487
5022        db 0x33, 0xd2      ; xor edx, edx
5023        mov dl, byte [esi + 0x4000]
5024        movzx ebx, byte [esi + 0x4200]
5025        db 0x2b, 0xda      ; sub ebx, edx
5026        mov eax, 0x80
5027        db 0x2b, 0xc2      ; sub eax, edx
5028        shl eax, 6
5029        cdq
5030        idiv ebx
5031        mov ebx, dword [D_07b04]
5032        db 0x8b, 0xd0      ; mov edx, eax
5033        add edx, dword [D_07b14]
5034        mov dword [D_07afc], edx
5035        fild dword [D_07afc]
5036        fstp dword [ebx + 4]
5037        fild dword [D_07b10]
5038        fadd dword [D_07b1c]
5039        fstp dword [ebx]
5040        fild dword [D_07b18]
5041        fstp dword [ebx + 8]
5042        mov dword [esi + 0x4008], ebx
5043        call L_088aa
5044        add dword [D_07b04], 0x40
5045        ret
5046    L_08487:
5047        mov ebx, dword [esi + 0x4008]
5048        ret
5049        cmp dword [esi + 0x204], -1
5050        jne short L_084fc
5051        db 0x33, 0xd2      ; xor edx, edx
5052        mov dl, byte [esi + 0x200]
5053        movzx ebx, byte [esi + 0x4200]
5054        db 0x2b, 0xda      ; sub ebx, edx
5055        mov eax, 0x80
5056        db 0x2b, 0xc2      ; sub eax, edx
```

```
5057        shl eax, 6
5058        cdq
5059        idiv ebx
5060        mov ebx, dword [D_07b04]
5061        db 0x8b, 0xd0      ; mov edx, eax
5062        add edx, dword [D_07b10]
5063        mov dword [D_07afc], edx
5064        fild dword [D_07afc]
5065        fstp dword [ebx]
5066        fild dword [D_07b14]
5067        fadd dword [D_07b1c]
5068        fstp dword [ebx + 4]
5069        fild dword [D_07b18]
5070        fstp dword [ebx + 8]
5071        mov dword [esi + 0x204], ebx
5072        call L_088aa
5073        add dword [D_07b04], 0x40
5074        ret
5075    L_084fc:
5076        mov ebx, dword [esi + 0x204]
5077        ret
5078        cmp dword [esi + 0xc], -1
5079        jne short L_0855e
5080        db 0x33, 0xd2      ; xor edx, edx
5081        mov dl, byte [esi]
5082        movzx ebx, byte [esi + 0x10]
5083        db 0x2b, 0xda      ; sub ebx, edx
5084        mov eax, 0x80
5085        db 0x2b, 0xc2      ; sub eax, edx
5086        shl eax, 6
5087        cdq
5088        idiv ebx
5089        mov ebx, dword [D_07b04]
5090        db 0x8b, 0xd0      ; mov edx, eax
5091        add edx, dword [D_07b18]
5092        mov dword [D_07afc], edx
5093        fild dword [D_07afc]
5094        fstp dword [ebx + 8]
5095        fild dword [D_07b10]
5096        fstp dword [ebx]
5097        fild dword [D_07b14]
5098        fstp dword [ebx + 4]
5099        mov dword [esi + 0xc], ebx
5100        call L_088aa
5101        add dword [D_07b04], 0x40
5102        ret
5103    L_0855e:
5104        mov ebx, dword [esi + 0xc]
5105        ret
5106        cmp dword [esi + 0x400c], -1
5107        jne short L_085d0
5108        db 0x33, 0xd2      ; xor edx, edx
5109        mov dl, byte [esi + 0x4000]
5110        movzx ebx, byte [esi + 0x4010]
5111        db 0x2b, 0xda      ; sub ebx, edx
5112        mov eax, 0x80
5113        db 0x2b, 0xc2      ; sub eax, edx
5114        shl eax, 6
5115        cdq
5116        idiv ebx
5117        mov ebx, dword [D_07b04]
5118        db 0x8b, 0xd0      ; mov edx, eax
5119        add edx, dword [D_07b18]
5120        mov dword [D_07afc], edx
5121        fild dword [D_07afc]
5122        fstp dword [ebx + 8]
5123        fild dword [D_07b10]
5124        fadd dword [D_07b1c]
5125        fstp dword [ebx]
```

```
5126        fild dword [D_07b14]
5127        fstp dword [ebx + 4]
5128        mov dword [esi + 0x400c], ebx
5129        call L_088aa
5130        add dword [D_07b04], 0x40
5131        ret
5132    L_085d0:
5133        mov ebx, dword [esi + 0x400c]
5134        ret
5135        cmp dword [esi + 0x20c], -1
5136        jne short L_08645
5137        db 0x33, 0xd2       ; xor edx, edx
5138        mov dl, byte [esi + 0x200]
5139        movzx ebx, byte [esi + 0x210]
5140        db 0x2b, 0xda       ; sub ebx, edx
5141        mov eax, 0x80
5142        db 0x2b, 0xc2       ; sub eax, edx
5143        shl eax, 6
5144        cdq
5145        idiv ebx
5146        mov ebx, dword [D_07b04]
5147        db 0x8b, 0xd0       ; mov edx, eax
5148        add edx, dword [D_07b18]
5149        mov dword [D_07afc], edx
5150        fild dword [D_07afc]
5151        fstp dword [ebx + 8]
5152        fild dword [D_07b10]
5153        fstp dword [ebx]
5154        fild dword [D_07b14]
5155        fadd dword [D_07b1c]
5156        fstp dword [ebx + 4]
5157        mov dword [esi + 0x20c], ebx
5158        call L_088aa
5159        add dword [D_07b04], 0x40
5160        ret
5161    L_08645:
5162        mov ebx, dword [esi + 0x20c]
5163        ret
5164        cmp dword [esi + 0x420c], -1
5165        jne short L_086c0
5166        db 0x33, 0xd2       ; xor edx, edx
5167        mov dl, byte [esi + 0x4200]
5168        movzx ebx, byte [esi + 0x4210]
5169        mov eax, 0x80
5170        db 0x2b, 0xc2       ; sub eax, edx
5171        shl eax, 6
5172        db 0x2b, 0xda       ; sub ebx, edx
5173        cdq
5174        idiv ebx
5175        mov ebx, dword [D_07b04]
5176        db 0x8b, 0xd0       ; mov edx, eax
5177        add edx, dword [D_07b18]
5178        mov dword [D_07afc], edx
5179        fild dword [D_07afc]
5180        fstp dword [ebx + 8]
5181        fild dword [D_07b10]
5182        fadd dword [D_07b1c]
5183        fstp dword [ebx]
5184        fild dword [D_07b14]
5185        fadd dword [D_07b1c]
5186        fstp dword [ebx + 4]
5187        mov dword [esi + 0x420c], ebx
5188        call L_088aa
5189        add dword [D_07b04], 0x40
5190        ret
5191    L_086c0:
5192        mov ebx, dword [esi + 0x420c]
5193        ret
5194        cmp dword [esi + 0x14], -1
```

```
5195        jne short L_0872c
5196        db 0x33, 0xd2      ; xor edx, edx
5197        mov dl, byte [esi + 0x10]
5198        movzx ebx, byte [esi + 0x4010]
5199        db 0x2b, 0xda      ; sub ebx, edx
5200        mov eax, 0x80
5201        db 0x2b, 0xc2      ; sub eax, edx
5202        shl eax, 6
5203        cdq
5204        idiv ebx
5205        mov ebx, dword [D_07b04]
5206        db 0x8b, 0xd0      ; mov edx, eax
5207        add edx, dword [D_07b10]
5208        mov dword [D_07afc], edx
5209        fild dword [D_07afc]
5210        fstp dword [ebx]
5211        fild dword [D_07b14]
5212        fstp dword [ebx + 4]
5213        fild dword [D_07b18]
5214        fadd dword [D_07b1c]
5215        fstp dword [ebx + 8]
5216        mov dword [esi + 0x14], ebx
5217        call L_088aa
5218        add dword [D_07b04], 0x40
5219        ret
5220    L_0872c:
5221        mov ebx, dword [esi + 0x14]
5222        ret
5223        cmp dword [esi + 0x18], -1
5224        jne short L_087aa
5225        db 0x33, 0xd2      ; xor edx, edx
5226        mov dl, byte [esi + 0x10]
5227        movzx ebx, byte [esi + 0x210]
5228        db 0x2b, 0xda      ; sub ebx, edx
5229        mov eax, 0x80
5230        db 0x2b, 0xc2      ; sub eax, edx
5231        shl eax, 6
5232        cdq
5233        idiv ebx
5234        mov ebx, dword [D_07b04]
5235        db 0x8b, 0xd0      ; mov edx, eax
5236        add edx, dword [D_07b14]
5237        mov dword [D_07afc], edx
5238        fild dword [D_07afc]
5239        fstp dword [ebx + 4]
5240        mov edx, dword [D_07b10]
5241        mov dword [D_07afc], edx
5242        fild dword [D_07afc]
5243        fstp dword [ebx]
5244        mov edx, dword [D_07b18]
5245        add edx, 0x40
5246        mov dword [D_07afc], edx
5247        fild dword [D_07afc]
5248        fstp dword [ebx + 8]
5249        mov dword [esi + 0x18], ebx
5250        call L_088aa
5251        add dword [D_07b04], 0x40
5252        ret
5253    L_087aa:
5254        mov ebx, dword [esi + 0x18]
5255        ret
5256        cmp dword [esi + 0x4018], -1
5257        jne short L_08822
5258        db 0x33, 0xd2      ; xor edx, edx
5259        mov dl, byte [esi + 0x4010]
5260        movzx ebx, byte [esi + 0x4210]
5261        db 0x2b, 0xda      ; sub ebx, edx
5262        mov eax, 0x80
5263        db 0x2b, 0xc2      ; sub eax, edx
```

```
5264        shl eax, 6
5265        cdq
5266        idiv ebx
5267        mov ebx, dword [D_07b04]
5268        db 0x8b, 0xd0      ; mov edx, eax
5269        add edx, dword [D_07b14]
5270        mov dword [D_07afc], edx
5271        fild dword [D_07afc]
5272        fstp dword [ebx + 4]
5273        fild dword [D_07b10]
5274        fadd dword [D_07b1c]
5275        fstp dword [ebx]
5276        fild dword [D_07b18]
5277        fadd dword [D_07b1c]
5278        fstp dword [ebx + 8]
5279        mov dword [esi + 0x4018], ebx
5280        call L_088aa
5281        add dword [D_07b04], 0x40
5282        ret
5283    L_08822:
5284        mov ebx, dword [esi + 0x4018]
5285        ret
5286        cmp dword [esi + 0x214], -1
5287        jne short L_0889d
5288        db 0x33, 0xd2      ; xor edx, edx
5289        mov dl, byte [esi + 0x210]
5290        movzx ebx, byte [esi + 0x4210]
5291        db 0x2b, 0xda      ; sub ebx, edx
5292        mov eax, 0x80
5293        db 0x2b, 0xc2      ; sub eax, edx
5294        shl eax, 6
5295        cdq
5296        idiv ebx
5297        mov ebx, dword [D_07b04]
5298        db 0x8b, 0xd0      ; mov edx, eax
5299        add edx, dword [D_07b10]
5300        mov dword [D_07afc], edx
5301        fild dword [D_07afc]
5302        fstp dword [ebx]
5303        fild dword [D_07b14]
5304        fadd dword [D_07b1c]
5305        fstp dword [ebx + 4]
5306        fild dword [D_07b18]
5307        fadd dword [D_07b1c]
5308        fstp dword [ebx + 8]
5309        mov dword [esi + 0x214], ebx
5310        call L_088aa
5311        add dword [D_07b04], 0x40
5312        ret
5313    L_0889d:
5314        mov ebx, dword [esi + 0x214]
5315        ret
5316    L_088a4:
5317        jmp dword [D_07b31]
5318    L_088aa:
5319        jmp dword [D_07b2d]
5320    L_088b0:
5321        mov byte [ebx + 0x3c], 1
5322        pushad
5323        fld dword [ebx]
5324        fistp dword [D_08a48]
5325        fld dword [ebx + 4]
5326        fistp dword [D_08a4c]
5327        fld dword [ebx + 8]
5328        fistp dword [D_08a50]
5329        mov esi, dword [D_08a48]
5330        add esi, 0x400
5331        add esi, dword [D_02055]
5332        and esi, 0xfff
```

```
5333        mov eax, dword [esi*4 + D_17cf4]
5334        mov esi, dword [D_08a48]
5335        neg esi
5336        sub esi, 0x400
5337        add esi, dword [D_02055]
5338        and esi, 0xfff
5339        sub eax, dword [esi*4 + D_17cf4]
5340        mov dword [D_08a54], eax
5341        mov esi, dword [D_08a4c]
5342        add esi, 0x400
5343        sub esi, dword [D_02055]
5344        and esi, 0xfff
5345        mov eax, dword [esi*4 + D_17cf4]
5346        mov esi, dword [D_08a4c]
5347        neg esi
5348        sub esi, 0x400
5349        add esi, dword [D_02055]
5350        add esi, dword [D_02055]
5351        add esi, dword [D_02055]
5352        add esi, dword [D_02055]
5353        and esi, 0xfff
5354        sub eax, dword [esi*4 + D_17cf4]
5355        mov dword [D_08a58], eax
5356        mov esi, dword [D_08a50]
5357        add esi, 0x400
5358        sub esi, dword [D_02055]
5359        add esi, 0xd7
5360        and esi, 0xfff
5361        mov eax, dword [esi*4 + D_17cf4]
5362        mov esi, dword [D_08a50]
5363        neg esi
5364        sub esi, 0x400
5365        add esi, dword [D_02055]
5366        add esi, dword [D_02055]
5367        add esi, dword [D_02055]
5368        and esi, 0xfff
5369        sub eax, dword [esi*4 + D_17cf4]
5370        mov dword [D_08a5c], eax
5371        mov eax, dword [D_08a48]
5372        sar eax, 1
5373        add dword [D_08a54], eax
5374        mov eax, dword [D_08a4c]
5375        sar eax, 1
5376        add dword [D_08a58], eax
5377        mov eax, dword [D_08a50]
5378        sar eax, 1
5379        add dword [D_08a5c], eax
5380        fild dword [D_08a54]
5381        fimul dword [D_08a54]
5382        fild dword [D_08a58]
5383        fimul dword [D_08a58]
5384        faddp st1
5385        fild dword [D_08a5c]
5386        fimul dword [D_08a5c]
5387        faddp st1
5388        fsqrt
5389        fdivr dword [D_08a60]
5390        fld st0
5391        fld st0
5392        fimul dword [D_08a54]
5393        fst dword [ebx + 0x2c]
5394        fstp dword [ebx + 0x20]
5395        fimul dword [D_08a58]
5396        fst dword [ebx + 0x30]
5397        fstp dword [ebx + 0x24]
5398        fimul dword [D_08a5c]
5399        fstp dword [ebx + 0x28]
5400        popad
5401        ret
```

```
5402        db 0x83, 0xf9, 0x22, 0x4b, 0x83, 0xf9, 0xa2, 0x4a
5403    D_08a48:
5404        db 0x00, 0x00, 0x00, 0x00
5405    D_08a4c:
5406        db 0x00, 0x00, 0x00, 0x00
5407    D_08a50:
5408        db 0x00, 0x00, 0x00, 0x00
5409    D_08a54:
5410        db 0x00, 0x00, 0x00, 0x00
5411    D_08a58:
5412        db 0x00, 0x00, 0x00, 0x00
5413    D_08a5c:
5414        db 0x00, 0x00, 0x00, 0x00
5415    D_08a60:
5416        db 0x00, 0x00, 0x00, 0x4b, 0x00, 0x00, 0x00, 0x47
5417    L_08a68:
5418        push ebp
5419        push esi
5420        db 0x8b, 0xf3     ; mov esi, ebx
5421        add esi, 0x400
5422        add esi, dword [D_02055]
5423        and esi, 0xfff
5424        mov eax, dword [esi*4 + D_13cf4]
5425        db 0x8b, 0xf3     ; mov esi, ebx
5426        neg esi
5427        sub esi, 0x400
5428        add esi, dword [D_02055]
5429        and esi, 0xfff
5430        add eax, dword [esi*4 + D_13cf4]
5431        db 0x8b, 0xf1     ; mov esi, ecx
5432        add esi, 0x400
5433        sub esi, dword [D_02055]
5434        and esi, 0xfff
5435        add eax, dword [esi*4 + D_13cf4]
5436        db 0x8b, 0xf1     ; mov esi, ecx
5437        neg esi
5438        sub esi, 0x400
5439        add esi, dword [D_02055]
5440        add esi, dword [D_02055]
5441        add esi, dword [D_02055]
5442        add esi, dword [D_02055]
5443        and esi, 0xfff
5444        add eax, dword [esi*4 + D_13cf4]
5445        db 0x8b, 0xf2     ; mov esi, edx
5446        add esi, 0x400
5447        sub esi, dword [D_02055]
5448        add esi, 0xd7
5449        and esi, 0xfff
5450        add eax, dword [esi*4 + D_13cf4]
5451        db 0x8b, 0xf2     ; mov esi, edx
5452        neg esi
5453        sub esi, 0x400
5454        add esi, dword [D_02055]
5455        add esi, dword [D_02055]
5456        add esi, dword [D_02055]
5457        and esi, 0xfff
5458        add eax, dword [esi*4 + D_13cf4]
5459        sar eax, 1
5460        mov ebp, dword [ebx*4 + D_0fcf0]
5461        add ebp, dword [ecx*4 + D_0fcf0]
5462        add ebp, dword [edx*4 + D_0fcf0]
5463        sar ebp, 0xa
5464        mov ebp, dword [ds:ebp*4 + D_0fcf0]
5465        sar ebp, 0xb
5466        mov ebp, dword [ds:ebp*4 + D_0fcf0]
5467        sar ebp, 0xa
5468        db 0x2b, 0xc5     ; sub eax, ebp
5469        add eax, 0x80
5470        jns short L_08b71
```

```
5471        db 0x33, 0xc0     ; xor eax, eax
5472    L_08b71:
5473        pop esi
5474        pop ebp
5475        ret
5476        times 24 db 0
5477        db 0x0d, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x00, 0xf6, 0xff, 0xff, 0xff, 0x05, 0x00
            , 0x00, 0x00
5478        db 0xf8, 0xff, 0xff, 0xff, 0x08
5479        times 27 db 0
5480        db 0x0f, 0x00, 0x00, 0x00, 0xf5, 0xff, 0xff, 0xff, 0xf6, 0xff, 0xff, 0xff, 0x07, 0x00
            , 0x00, 0x00
5481        db 0xfb, 0xff, 0xff, 0xff, 0xfa, 0xff, 0xff, 0xff
5482        times 24 db 0
5483        db 0x04, 0x00, 0x00, 0x00, 0xf7, 0xff, 0xff, 0xff, 0x0b, 0x00, 0x00, 0x00, 0x0a, 0x00
            , 0x00, 0x00
5484        db 0x07, 0x00, 0x00, 0x00, 0x05
5485        times 147 db 0
5486    L_08c94:
5487        fldz
5488        fldz
5489        fldz
5490        mov edx, 0x14
5491    L_08c9f:
5492        fld dword [ebx]
5493        fsub dword [edx + 0x8c4c]
5494        fld dword [ebx + 4]
5495        fsub dword [edx + 0x8c64]
5496        fld dword [ebx + 8]
5497        fsub dword [edx + 0x8c7c]
5498        fmul st0
5499        fxch st1
5500        fmul st0
5501        fxch st2
5502        fmul st0
5503        faddp st1
5504        faddp st1
5505        fld st0
5506        fsqrt
5507        fmulp st1
5508        fld1
5509        fdivrp st1
5510        fld dword [ebx]
5511        fsub dword [edx + 0x8c4c]
5512        fmul st1
5513        faddp st2
5514        fld dword [ebx + 4]
5515        fsub dword [edx + 0x8c64]
5516        fmul st1
5517        faddp st3
5518        fld dword [ebx + 8]
5519        fsub dword [edx + 0x8c7c]
5520        fmulp st1
5521        faddp st3
5522        sub edx, 4
5523        jns short L_08c9f
5524        fld st2
5525        fld st2
5526        fld st2
5527        fmul st0
5528        fxch st1
5529        fmul st0
5530        fxch st2
5531        fmul st0
5532        faddp st1
5533        faddp st1
5534        fsqrt
5535        fld dword [D_08d36]
5536        fdivrp st1
```

```
5537        fstp st4
5538        fmul st3
5539        fst dword [ebx + 0x20]
5540        fstp dword [ebx + 0x2c]
5541        fmul st2
5542        fst dword [ebx + 0x24]
5543        fstp dword [ebx + 0x30]
5544        fmulp st1
5545        fstp dword [ebx + 0x28]
5546        mov byte [ebx + 0x3c], 1
5547        ret
5548    D_08d36:
5549        db 0x00, 0x00, 0x00, 0x4b
5550    L_08d3a:
5551        push ebp
5552        push esi
5553        db 0x33, 0xed       ; xor ebp, ebp
5554        mov edi, 0x14
5555        nop
5556    L_08d44:
5557        mov esi, dword [edi + 0x8c04]
5558        db 0x2b, 0xf3       ; sub esi, ebx
5559        mov eax, dword [esi*4 + D_0fcf0]
5560        mov esi, dword [edi + 0x8c1c]
5561        db 0x2b, 0xf1       ; sub esi, ecx
5562        add eax, dword [esi*4 + D_0fcf0]
5563        mov esi, dword [edi + 0x8c34]
5564        db 0x2b, 0xf2       ; sub esi, edx
5565        add eax, dword [esi*4 + D_0fcf0]
5566        call L_0921d
5567        db 0x03, 0xe8       ; add ebp, eax
5568        sub edi, 4
5569        jns short L_08d44
5570        cmp ebp, 0xff
5571        jbe short L_08d8a
5572        mov ebp, 0xff
5573    L_08d8a:
5574        db 0x8b, 0xc5       ; mov eax, ebp
5575        pop esi
5576        pop ebp
5577        ret
5578    L_08d8f:
5579        db 0x8b, 0xd8       ; mov ebx, eax
5580        mov ecx, 0x14
5581    L_08d96:
5582        mov eax, dword [ecx + 0x8b8c]
5583        imul ebx
5584        lea eax, [eax + eax*2]
5585        sar eax, 5
5586        and eax, 0x3ff
5587        mov dword [ecx + 0x8b74], eax
5588        mov eax, dword [ecx + 0x8bbc]
5589        imul ebx
5590        lea eax, [eax + eax*2]
5591        sar eax, 5
5592        and eax, 0x3ff
5593        mov dword [ecx + 0x8ba4], eax
5594        mov eax, dword [ecx + 0x8bec]
5595        imul ebx
5596        lea eax, [eax + eax*2]
5597        sar eax, 5
5598        and eax, 0x3ff
5599        mov dword [ecx + 0x8bd4], eax
5600        sub ecx, 4
5601        jns short L_08d96
5602        mov ecx, 0x14
5603    L_08deb:
5604        mov ebx, dword [ecx + 0x8b74]
5605        mov eax, dword [ebx*4 + D_0a4ec]
```

```
5606        mov edx, dword [ecx + 0x8bd4]
5607        mov edx, dword [edx*4 + D_0a4ec]
5608        sar edx, 2
5609        add edx, 0x226
5610        push edx
5611        imul edx
5612        sar eax, 0xa
5613        mov dword [D_07afc], eax
5614        fild dword [D_07afc]
5615        fstp dword [ecx + 0x8c4c]
5616        mov dword [ecx + 0x8c04], eax
5617        pop edx
5618        mov eax, dword [ebx*4 + D_0a8ec]
5619        imul edx
5620        sar eax, 0xb
5621        sub eax, 0x200
5622        mov dword [D_07afc], eax
5623        fild dword [D_07afc]
5624        fstp dword [ecx + 0x8c64]
5625        mov dword [ecx + 0x8c1c], eax
5626        mov eax, dword [ecx + 0x8ba4]
5627        mov eax, dword [eax*4 + D_0a4ec]
5628        sar eax, 1
5629        mov dword [D_07afc], eax
5630        fild dword [D_07afc]
5631        fstp dword [ecx + 0x8c7c]
5632        mov dword [ecx + 0x8c34], eax
5633        sub ecx, 4
5634        jns near L_08deb
5635        ret
5636    L_08e84:
5637        mov edi, D_13cf4
5638        mov dword [D_02008], 0
5639    L_08e93:
5640        fldpi
5641        fdiv dword [D_08f9f]
5642        fimul dword [D_02008]
5643        fld st0
5644        fmul dword [D_08fbb]
5645        fsin
5646        fmul dword [D_08faf]
5647        fld st1
5648        fmul dword [D_08fab]
5649        fcos
5650        fmul dword [D_08fa7]
5651        faddp st1
5652        fld st1
5653        fmul dword [D_08fb3]
5654        fldpi
5655        faddp st1
5656        fsin
5657        fmul dword [D_08fa3]
5658        fchs
5659        faddp st1
5660        fld st1
5661        fmul dword [D_08fb7]
5662        fcos
5663        fmul dword [D_08fa7]
5664        faddp st1
5665        fchs
5666        fistp dword [edi]
5667        ffree st0
5668        add edi, 4
5669        inc dword [D_02008]
5670        cmp dword [D_02008], 0x1000
5671        jne short L_08e93
5672        mov edi, D_17cf4
5673        mov dword [D_02008], 0
5674    L_08f17:
```

```
5675        fldpi
5676        fdiv dword [D_08f9f]
5677        fimul dword [D_02008]
5678        fld st0
5679        fmul dword [D_08fbb]
5680        fcos
5681        fmul dword [D_08f8f]
5682        fld st1
5683        fmul dword [D_08fb7]
5684        fsin
5685        fmul dword [D_08f93]
5686        fchs
5687        faddp st1
5688        fld st1
5689        fmul dword [D_08fb3]
5690        fldpi
5691        faddp st1
5692        fcos
5693        fmul dword [D_08f97]
5694        fchs
5695        faddp st1
5696        fld st1
5697        fmul dword [D_08fab]
5698        fsin
5699        fmul dword [D_08f9b]
5700        fchs
5701        faddp st1
5702        fistp dword [edi]
5703        ffree st0
5704        add edi, 4
5705        inc dword [D_02008]
5706        cmp dword [D_02008], 0x1000
5707        jne short L_08f17
5708        ret
5709    D_08f8f:
5710        db 0x00, 0x00, 0x16, 0x43
5711    D_08f93:
5712        db 0x00, 0x00, 0x20, 0x43
5713    D_08f97:
5714        db 0x00, 0x00, 0x70, 0x42
5715    D_08f9b:
5716        db 0x00, 0x00, 0xa0, 0x42
5717    D_08f9f:
5718        db 0x00, 0x00, 0x00, 0x45
5719    D_08fa3:
5720        db 0x00, 0x00, 0xa0, 0x41
5721    D_08fa7:
5722        db 0x00, 0x00, 0x20, 0x42
5723    D_08fab:
5724        db 0x00, 0x00, 0x00, 0x40
5725    D_08faf:
5726        db 0x00, 0x00, 0xf0, 0x41
5727    D_08fb3:
5728        db 0x00, 0x00, 0x40, 0x40
5729    D_08fb7:
5730        db 0x00, 0x00, 0x80, 0x40
5731    D_08fbb:
5732        db 0x00, 0x00, 0xa0, 0x40, 0x00, 0x00, 0x00, 0x00, 0x3d, 0x00, 0x00, 0x01, 0x00, 0x0f
        , 0x82, 0x98
5733        db 0x00, 0x00, 0x00, 0x3d, 0x00, 0x00, 0x00, 0x01, 0x72, 0x49, 0x3d, 0x00, 0x00, 0x00
        , 0x10, 0x72
5734        db 0x21, 0x3d, 0x00, 0x00, 0x00, 0x40, 0x72, 0x0d, 0xc1, 0xe8, 0x18, 0x8a, 0x80, 0xf2
        , 0x90, 0x00
5735        db 0x00, 0xc1, 0xe0, 0x08, 0xc3, 0xc1, 0xe8, 0x16, 0x8a, 0x80, 0xf2, 0x90, 0x00, 0x00
        , 0xc1, 0xe0
5736        db 0x07, 0xc3, 0x3d, 0x00, 0x00, 0x00, 0x04, 0x72, 0x0d, 0xc1, 0xe8, 0x14, 0x8a, 0x80
        , 0xf2, 0x90
5737        db 0x00, 0x00, 0xc1, 0xe0, 0x06, 0xc3, 0xc1, 0xe8, 0x12, 0x8a, 0x80, 0xf2, 0x90, 0x00
        , 0x00, 0xc1
```

```
5738        db 0xe0, 0x05, 0xc3, 0x3d, 0x00, 0x00, 0x10, 0x00, 0x72, 0x21, 0x3d, 0x00, 0x00, 0x40
            , 0x00, 0x72
5739        db 0x0d, 0xc1, 0xe8, 0x10, 0x8a, 0x80, 0xf2, 0x90, 0x00, 0x00, 0xc1, 0xe0, 0x04, 0xc3
            , 0xc1, 0xe8
5740        db 0x0e, 0x8a, 0x80, 0xf2, 0x90, 0x00, 0x00, 0xc1, 0xe0, 0x03, 0xc3, 0x3d, 0x00, 0x00
            , 0x04, 0x00
5741        db 0x72, 0x0d, 0xc1, 0xe8, 0x0c, 0x8a, 0x80, 0xf2, 0x90, 0x00, 0x00, 0xc1, 0xe0, 0x02
            , 0xc3, 0xc1
5742        db 0xe8, 0x0a, 0x8a, 0x80, 0xf2, 0x90, 0x00, 0x00, 0xd1, 0xe0, 0xc3, 0x3d, 0x00, 0x01
            , 0x00, 0x00
5743        db 0x72, 0x45, 0x3d, 0x00, 0x10, 0x00, 0x00, 0x72, 0x1d, 0x3d, 0x00, 0x40, 0x00, 0x00
            , 0x72, 0x0a
5744        db 0xc1, 0xe8, 0x08, 0x8a, 0x80, 0xf2, 0x90, 0x00, 0x00
5745        ret
5746        shr eax, 6
5747        mov al, byte [eax + 0x90f2]
5748        shr eax, 1
5749        ret
5750        cmp eax, 0x400
5751        jb short L_090a5
5752        shr eax, 4
5753        mov al, byte [eax + 0x90f2]
5754        shr eax, 2
5755        ret
5756    L_090a5:
5757        shr eax, 2
5758        mov al, byte [eax + 0x90f2]
5759        shr eax, 3
5760        ret
5761        cmp eax, 0x10
5762        jb short L_090d3
5763        cmp eax, 0x40
5764        jb short L_090c6
5765        mov al, byte [eax + 0x90f2]
5766        shr eax, 4
5767        ret
5768    L_090c6:
5769        shl eax, 2
5770        mov al, byte [eax + 0x90f2]
5771        shr eax, 5
5772        ret
5773    L_090d3:
5774        cmp eax, 4
5775        jb short L_090e5
5776        shl eax, 4
5777        mov al, byte [eax + 0x90f2]
5778        shr eax, 6
5779        ret
5780    L_090e5:
5781        shl eax, 6
5782        mov al, byte [eax + 0x90f2]
5783        shr eax, 7
5784        ret
5785        db 0x00, 0x10, 0x16, 0x1b, 0x20, 0x23, 0x27
5786        db "*-02579;=@ACEGIKLNPQSTVWYZ[]^`abcefghjklmnpqrstu"
5787        jbe short L_091a2
5788        js short L_091a6
5789        db 0x7a, 0x7b     ; jp 0x91aa
5790        db 0x7c, 0x7d     ; jl 0x91ae
5791        db 0x7e, 0x80     ; jle 0x90b3
5792        add byte [ecx - 0x7a7b7c7e], 0x86
5793        xchg dword [eax - 0x73747577], ecx
5794        lea ecx, [esi - 0x6e6f6f71]
5795        xchg edx, eax
5796        xchg ebx, eax
5797        xchg esp, eax
5798        xchg ebp, eax
5799        xchg esi, eax
5800        xchg esi, eax
```

```
5801        xchg edi, eax
5802        cwde
5803        cdq
5804        db 0x9a, 0x9b, 0x9b, 0x9c, 0x9d, 0x9e, 0x9f     ; lcall 0x9f9e, 0x9d9c9b9b
5805        mov al, byte [0xa3a2a1a0]
5806        mov dword [0xa7a6a5a4], eax
5807        cmpsd
5808        test al, 0xa9
5809        stosb
5810        stosb
5811        stosd
5812        lodsb
5813        lodsd
5814        lodsd
5815        scasb
5816        scasd
5817        mov al, 0xb0
5818        mov cl, 0xb2
5819        mov dl, 0xb3
5820        mov ah, 0xb5
5821        mov ch, 0xb6
5822        mov bh, 0xb7
5823        mov eax, 0xbbbab9b9
5824        mov ebx, 0xbebdbdbc
5825        mov edi, 0xc1c1c0c0
5826        ret 0xc3c3
5827        db 0xc4, 0xc5, 0xc5, 0xc6, 0xc7, 0xc7, 0xc8, 0xc9, 0xc9, 0xca, 0xcb, 0xcb, 0xcc, 0xcc
            , 0xcd, 0xce
5828        db 0xce, 0xcf, 0xd0, 0xd0, 0xd1, 0xd1, 0xd2, 0xd3, 0xd3
5829    L_091a2:
5830        aam 0xd4
5831        aad 0xd6
5832    L_091a6:
5833        salc
5834        xlatb
5835        xlatb
5836        fcomp st1
5837        db 0xd9, 0xda     ; fstpnce st(2), st(0)
5838        fcmovu st0, st3
5839        fcmovnu st0, st4
5840        fstp st5
5841        db 0xde, 0xde, 0xdf, 0xe0, 0xe0, 0xe1, 0xe1, 0xe2, 0xe2, 0xe3, 0xe3, 0xe4, 0xe5, 0xe5
            , 0xe6, 0xe6
5842        db 0xe7, 0xe7, 0xe8, 0xe8, 0xe9, 0xea, 0xea, 0xeb, 0xeb, 0xec, 0xec, 0xed, 0xed, 0xee
            , 0xee, 0xef
5843        db 0xf0, 0xf0, 0xf1, 0xf1, 0xf2, 0xf2, 0xf3, 0xf3, 0xf4, 0xf4, 0xf5, 0xf5, 0xf6, 0xf6
            , 0xf7, 0xf7
5844        db 0xf8, 0xf8, 0xf9, 0xf9, 0xfa, 0xfa, 0xfb, 0xfb, 0xfc, 0xfc, 0xfd, 0xfd, 0xfe, 0xfe
            , 0xff
5845    L_091f2:
5846        mov edi, 0xb8f0
5847        mov eax, 1
5848        fild dword [D_0b8ec]
5849    L_09202:
5850        mov dword [D_02000], eax
5851        fild dword [D_02000]
5852        fsqrt
5853        fdivr st1
5854        fistp dword [edi]
5855        add edi, 4
5856        inc al
5857        jne short L_09202
5858        faddp st0
5859        ret
5860    L_0921d:
5861        cmp eax, 0x10000
5862        jb short L_09270
5863        cmp eax, 0x100000
5864        jb short L_0924e
```

```
5865        cmp eax, 0x400000
5866        jb short L_09240
5867        shr eax, 0x10
5868        mov eax, dword [eax*4 + D_0b8ec]
5869        shr eax, 4
5870        ret
5871    L_09240:
5872        shr eax, 0xe
5873        mov eax, dword [eax*4 + D_0b8ec]
5874        shr eax, 3
5875        ret
5876    L_0924e:
5877        cmp eax, 0x40000
5878        jb short L_09263
5879        shr eax, 0xc
5880        mov eax, dword [eax*4 + D_0b8ec]
5881        shr eax, 2
5882        ret
5883    L_09263:
5884        shr eax, 0xa
5885        mov eax, dword [eax*4 + D_0b8ec]
5886        shr eax, 1
5887        ret
5888    L_09270:
5889        cmp eax, 0x100
5890        jb short L_092c0
5891        cmp eax, 0x1000
5892        jb short L_0929d
5893        cmp eax, 0x4000
5894        jb short L_09290
5895        shr eax, 8
5896        mov eax, dword [eax*4 + D_0b8ec]
5897        ret
5898    L_09290:
5899        shr eax, 6
5900        mov eax, dword [eax*4 + D_0b8ec]
5901        shl eax, 1
5902        ret
5903    L_0929d:
5904        cmp eax, 0x400
5905        jb short L_092b2
5906        shr eax, 4
5907        mov eax, dword [eax*4 + D_0b8ec]
5908        shl eax, 2
5909        ret
5910    L_092b2:
5911        shr eax, 2
5912        mov eax, dword [eax*4 + D_0b8ec]
5913        shl eax, 3
5914        ret
5915    L_092c0:
5916        cmp eax, 0x10
5917        jb short L_092e3
5918        cmp eax, 0x40
5919        jb short L_092d5
5920        mov eax, dword [eax*4 + D_0b8ec]
5921        shl eax, 4
5922        ret
5923    L_092d5:
5924        shl eax, 2
5925        mov eax, dword [eax*4 + D_0b8ec]
5926        shl eax, 5
5927        ret
5928    L_092e3:
5929        cmp eax, 4
5930        jb short L_092f6
5931        shl eax, 4
5932        mov eax, dword [eax*4 + D_0b8ec]
5933        shl eax, 6
```

```
5934        ret
5935    L_092f6:
5936        shl eax, 6
5937        mov eax, dword [eax*4 + D_0b8ec]
5938        shl eax, 7
5939        ret
5940    L_09304:
5941        mov ecx, 0x500
5942        fldpi
5943        fdiv dword [D_0933d]
5944        fstp dword [D_09339]
5945        fldz
5946        mov edi, D_0a4ec
5947    L_0931e:
5948        fld st0
5949        fsin
5950        fmul dword [D_09341]
5951        fistp dword [edi]
5952        fadd dword [D_09339]
5953        add edi, 4
5954        dec ecx
5955        jne short L_0931e
5956        ffree st0
5957        ret
5958    D_09339:
5959        db 0x00, 0x00, 0x00, 0x00
5960    D_0933d:
5961        db 0x00, 0x00, 0x00, 0x44
5962    D_09341:
5963        db 0x00, 0x00, 0x80, 0x44
5964    D_09345:
5965        db 0x00, 0x00, 0x00, 0x00
5966    D_09349:
5967        times 8 db 0
5968    D_09351:
5969        db 0x00
5970    D_09352:
5971        db 0x66, 0x66, 0x66, 0x3f, 0x9a, 0x99, 0x99, 0x3e, 0xf6, 0x28, 0xe4, 0x40
5972    L_0935e:
5973        mov ebx, dword [D_02067]
5974        mov ebx, dword [ebx]
5975        mov dword [D_02004], ebx
5976        fild dword [D_02004]
5977        movzx eax, byte [D_09351]
5978        fmul dword [eax*4 + D_09352]
5979        fistp dword [D_093f2]
5980        cmp byte [D_09351], 0
5981        je short L_093b9
5982        cmp byte [D_09351], 1
5983        je short L_093d3
5984        mov dword [D_0947b], 0
5985        mov eax, dword [D_093f2]
5986        cmp eax, 0xff
5987        jbe short L_093b3
5988        mov eax, 0xff
5989    L_093b3:
5990        mov dword [D_09477], eax
5991        ret
5992    L_093b9:
5993        mov eax, dword [D_093f2]
5994        sub eax, 0x100
5995        cmp eax, 0
5996        jl short L_093cd
5997        mov eax, 0
5998    L_093cd:
5999        mov dword [D_09477], eax
6000        ret
6001    L_093d3:
6002        mov dword [D_09477], 0
```

```
6003        mov eax, 0x50
6004        sub eax, dword [D_093f2]
6005        jns short L_093ec
6006        db 0x33, 0xc0     ; xor eax, eax
6007    L_093ec:
6008        mov dword [D_0947b], eax
6009        ret
6010    D_093f2:
6011        db 0x00, 0x00, 0x00, 0x00
6012    D_093f6:
6013        db 0x00, 0x00, 0x00, 0x00
6014    D_093fa:
6015        db 0x00, 0x00, 0x00, 0x00
6016    D_093fe:
6017        db 0x00, 0x00, 0x00, 0x00
6018    L_09402:
6019        mov cl, 8
6020        sub cl, byte [D_0211c]
6021        mov ch, byte [D_02119]
6022        mov edi, dword [D_093f6]
6023        call L_09448
6024        mov cl, 8
6025        sub cl, byte [D_0211d]
6026        mov ch, byte [D_0211a]
6027        mov edi, dword [D_093fa]
6028        call L_09448
6029        mov cl, 8
6030        sub cl, byte [D_0211e]
6031        mov ch, byte [D_0211b]
6032        mov edi, dword [D_093fe]
6033    L_09448:
6034        mov ebx, 0xffffff00
6035    L_0944d:
6036        db 0x8b, 0xc3     ; mov eax, ebx
6037        cmp eax, 0xff
6038        jbe short L_09464
6039        jg short L_0945f
6040        mov eax, 0
6041        jmp short L_09464
6042    L_0945f:
6043        mov eax, 0xff
6044    L_09464:
6045        shr eax, cl
6046        db 0x86, 0xcd     ; xchg ch, cl
6047        shl eax, cl
6048        db 0x86, 0xcd     ; xchg ch, cl
6049        stosd
6050        inc ebx
6051        cmp ebx, 0x200
6052        jl short L_0944d
6053        ret
6054    D_09477:
6055        db 0x00, 0xff, 0xff, 0xff
6056    D_0947b:
6057        db 0x50, 0x00, 0x00, 0x00
6058    D_0947f:
6059        db 0x00, 0x00, 0x00, 0x00
6060    D_09483:
6061        db 0x00, 0x00, 0x00, 0x00
6062    D_09487:
6063        db 0x00
6064    D_09488:
6065        db 0x00, 0x00, 0x00
6066    L_0948b:
6067        mov eax, dword [D_09349]
6068        db 0xa3, 0x8e, 0x95, 0x00, 0x00     ; mov dword ptr [0x958e], eax
6069        inc eax
6070        db 0xa3, 0x9b, 0x95, 0x00, 0x00     ; mov dword ptr [0x959b], eax
6071        inc eax
```

```
6072        db 0xa3, 0xa8, 0x95, 0x00, 0x00     ; mov dword ptr [0x95a8], eax
6073        mov ebx, dword [D_09477]
6074        add ebx, 0x100
6075        shl ebx, 2
6076        mov eax, dword [D_093f6]
6077        db 0x03, 0xc3     ; add eax, ebx
6078        mov ecx, dword [eax]
6079        db 0xa3, 0x95, 0x95, 0x00, 0x00     ; mov dword ptr [0x9595], eax
6080        mov eax, dword [D_093fa]
6081        db 0x03, 0xc3     ; add eax, ebx
6082        or ecx, dword [eax]
6083        db 0xa3, 0xa2, 0x95, 0x00, 0x00     ; mov dword ptr [0x95a2], eax
6084        mov eax, dword [D_093fe]
6085        db 0x03, 0xc3     ; add eax, ebx
6086        or ecx, dword [eax]
6087        db 0xa3, 0xaf, 0x95, 0x00, 0x00     ; mov dword ptr [0x95af], eax
6088        jmp short L_094dd
6089        nop
6090    L_094dd:
6091        db 0x8b, 0xc1     ; mov eax, ecx
6092        shl eax, 0x10
6093        db 0x66, 0x8b, 0xc1     ; mov ax, cx
6094        mov edi, dword [ptr_framebuffer_src]
6095        mov edx, 0xc8
6096    L_094f0:
6097        mov ecx, 0x10
6098        rep stosd
6099        add edi, 0x200
6100        mov ecx, 0x10
6101        rep stosd
6102        dec edx
6103        jne short L_094f0
6104        mov esi, dword [D_09345]
6105        mov edi, dword [D_0947f]
6106        db 0x89, 0x3d, 0x76, 0x95, 0x00, 0x00     ; mov dword ptr [0x9576], edi
6107        db 0x89, 0x3d, 0x7c, 0x95, 0x00, 0x00     ; mov dword ptr [0x957c], edi
6108        mov ebx, dword [D_0947b]
6109        mov ebp, 0x10000
6110    L_0952a:
6111        mov al, byte [esi]
6112        mul bl
6113        mov byte [edi], ah
6114        inc esi
6115        inc edi
6116        dec ebp
6117        jne short L_0952a
6118        mov edi, dword [ptr_framebuffer_src]
6119        add edi, 0x40
6120        mov ecx, dword [D_09483]
6121        mov edx, dword [D_09487]
6122        db 0x33, 0xf6     ; xor esi, esi
6123        mov eax, dword [D_0947b]
6124        shr eax, 1
6125        neg eax
6126        db 0x8b, 0xf0     ; mov esi, eax
6127        shl eax, 8
6128        db 0x03, 0xf0     ; add esi, eax
6129        db 0x33, 0xc0     ; xor eax, eax
6130        mov dword [D_02018], 0xc8
6131    L_09568:
6132        mov dword [D_0200c], 0x100
6133    L_09572:
6134        db 0x33, 0xdb     ; xor ebx, ebx
6135        mov bl, byte [ecx + 0x12345678]
6136        mov bh, byte [edx + 0x12345678]
6137        db 0x66, 0x03, 0xde     ; add bx, si
6138        db 0x33, 0xc0     ; xor eax, eax
6139        inc cx
6140        inc dx
```

```
6141        lea ebx, [ebx + ebx*2]
6142        mov al, byte [ebx + 0x12345678]
6143        mov ebp, dword [eax*4 + 0x12345678]
6144        mov al, byte [ebx + 0x12345678]
6145        or ebp, dword [eax*4 + 0x12345678]
6146        mov al, byte [ebx + 0x12345678]
6147        or ebp, dword [eax*4 + 0x12345678]
6148        mov word [edi], bp
6149        inc esi
6150        add edi, 2
6151        dec dword [D_0200c]
6152        jne short L_09572
6153        add edi, 0x280
6154        sub edi, 0x200
6155        dec dword [D_02018]
6156        jne short L_09568
6157        ret
6158    L_095d7:
6159        mov eax, dword [D_09624]
6160        inc eax
6161        and eax, 3
6162        mov dword [D_09624], eax
6163        call L_04a65
6164        call L_0935e
6165        call L_099c6
6166        inc byte [D_09483]
6167        dec byte [D_09487]
6168        dec byte [D_09488]
6169        call L_0948b
6170        cmp byte [D_0993b], 1
6171        jne short L_09619
6172        call L_099d9
6173    L_09619:
6174        mov eax, dword [D_09624]
6175        call Blit_Or_Present
6176        ret
6177    D_09624:
6178        db 0x00, 0x00, 0x00, 0x00
6179    D_09628:
6180        db 0x00, 0x00, 0x00, 0x00
6181    L_0962c:
6182        db 0x33, 0xc0      ; xor eax, eax
6183        call Blit_Or_Present
6184        db 0x33, 0xc0      ; xor eax, eax
6185        call L_04a65
6186        mov esi, 0x316d8
6187        mov ebp, 0x100
6188        mov edi, dword [D_09628]
6189        mov dword [L_09751+4], edi
6190    L_09650:
6191        db 0x33, 0xc0      ; xor eax, eax
6192        mov al, byte [esi]
6193        mov cl, 8
6194        sub cl, byte [D_0211c]
6195        shr al, cl
6196        mov cl, byte [D_02119]
6197        shl eax, cl
6198        db 0x33, 0xdb      ; xor ebx, ebx
6199        mov bl, byte [esi + 1]
6200        mov cl, 8
6201        sub cl, byte [D_0211d]
6202        shr bl, cl
6203        mov cl, byte [D_0211a]
6204        shl ebx, cl
6205        db 0x0b, 0xc3      ; or eax, ebx
6206        db 0x33, 0xdb      ; xor ebx, ebx
6207        mov bl, byte [esi + 2]
6208        mov cl, 8
6209        sub cl, byte [D_0211e]
```

```
6210        shr bl, cl
6211        mov cl, byte [D_0211b]
6212        shl ebx, cl
6213        db 0x0b, 0xc3      ; or eax, ebx
6214        stosd
6215        add esi, 3
6216        dec ebp
6217        jne short L_09650
6218        push esi
6219        mov edi, dword [ptr_framebuffer_src]
6220        mov esi, dword [BufPtr_09a85]
6221        mov ebp, 0xc8
6222    L_096b1:
6223        push esi
6224        mov edx, 0x140
6225    L_096b7:
6226        movzx eax, byte [esi]
6227        mov cl, 8
6228        sub cl, byte [D_0211c]
6229        shr eax, cl
6230        mov cl, byte [D_02119]
6231        shl eax, cl
6232        movzx ebx, byte [esi + 1]
6233        mov cl, 8
6234        sub cl, byte [D_0211d]
6235        shr ebx, cl
6236        mov cl, byte [D_0211a]
6237        shl ebx, cl
6238        db 0x0b, 0xc3      ; or eax, ebx
6239        movzx ebx, byte [esi + 2]
6240        mov cl, 8
6241        sub cl, byte [D_0211e]
6242        shr ebx, cl
6243        mov cl, byte [D_0211b]
6244        shl ebx, cl
6245        db 0x0b, 0xc3      ; or eax, ebx
6246        add esi, 3
6247        mov word [edi], ax
6248        add edi, 2
6249        dec edx
6250        jne short L_096b7
6251        pop esi
6252        add esi, 0x300
6253        dec ebp
6254        jne short L_096b1
6255        pop esi
6256        movzx ebp, word [esi]
6257        mov dword [D_02018], ebp
6258        dec dword [D_02018]
6259        add esi, 2
6260        mov edi, dword [ptr_framebuffer_src]
6261        add edi, 0x1248
6262        db 0x33, 0xdb      ; xor ebx, ebx
6263    L_09733:
6264        db 0x33, 0xc0      ; xor eax, eax
6265        mov ecx, 1
6266        mov al, byte [esi]
6267        inc esi
6268        cmp al, 0xff
6269        jne short L_09751
6270        mov cl, byte [esi]
6271        mov dword [D_02008], 0
6272        inc esi
6273        mov al, 0xff
6274        dec ebp
6275    L_09751:
6276        mov dx, word [eax*4 + 0x12345678]
6277        cmp dx, -1
6278        je short L_09767
```

```
6279        mov word [edi], dx
6280        jmp near L_09841
6281    L_09767:
6282        pushad
6283        cmp ebx, 2
6284        jb near L_09840
6285        cmp ebx, 0x6e
6286        ja near L_09840
6287        cmp ecx, 8
6288        jb short L_097a0
6289        cmp ebp, dword [D_02018]
6290        je near L_09840
6291        cmp dword [D_02008], 8
6292        jae near L_09840
6293        mov ecx, dword [D_02008]
6294        jmp short L_097a9
6295    L_097a0:
6296        cmp ebp, 2
6297        jb near L_09840
6298    L_097a9:
6299        db 0x8b, 0xe9       ; mov ebp, ecx
6300        neg ebp
6301        mov ax, word [edi]
6302        mov cl, byte [D_0211c]
6303        mov edx, 1
6304        shl edx, cl
6305        dec edx
6306        mov cl, byte [D_02119]
6307        shl edx, cl
6308        db 0x23, 0xc2       ; and eax, edx
6309        shr eax, cl
6310        shr edx, cl
6311        neg eax
6312        db 0x03, 0xc2       ; add eax, edx
6313        push edx
6314        imul ebp
6315        sar eax, 3
6316        pop edx
6317        db 0x03, 0xc2       ; add eax, edx
6318        shl eax, cl
6319        db 0x8b, 0xf0       ; mov esi, eax
6320        mov ax, word [edi]
6321        mov cl, byte [D_0211d]
6322        mov edx, 1
6323        shl edx, cl
6324        dec edx
6325        mov cl, byte [D_0211a]
6326        shl edx, cl
6327        db 0x23, 0xc2       ; and eax, edx
6328        shr eax, cl
6329        shr edx, cl
6330        neg eax
6331        db 0x03, 0xc2       ; add eax, edx
6332        push edx
6333        imul ebp
6334        sar eax, 3
6335        pop edx
6336        db 0x03, 0xc2       ; add eax, edx
6337        shl eax, cl
6338        db 0x0b, 0xf0       ; or esi, eax
6339        mov ax, word [edi]
6340        mov cl, byte [D_0211e]
6341        mov edx, 1
6342        shl edx, cl
6343        dec edx
6344        mov cl, byte [D_0211b]
6345        shl edx, cl
6346        db 0x23, 0xc2       ; and eax, edx
6347        shr eax, cl
```

```
6348        shr edx, cl
6349        neg eax
6350        db 0x03, 0xc2      ; add eax, edx
6351        push edx
6352        imul ebp
6353        sar eax, 3
6354        pop edx
6355        db 0x03, 0xc2      ; add eax, edx
6356        shl eax, cl
6357        db 0x0b, 0xc6      ; or eax, esi
6358        mov word [edi], ax
6359    L_09840:
6360        popad
6361    L_09841:
6362        add edi, 2
6363        inc ebx
6364        cmp ebx, 0x70
6365        jb short L_09852
6366        add edi, 0x1a0
6367        db 0x33, 0xdb      ; xor ebx, ebx
6368    L_09852:
6369        inc dword [D_02008]
6370        dec ecx
6371        jne near L_09751
6372        dec ebp
6373        jne near L_09733
6374        db 0x33, 0xc0      ; xor eax, eax
6375        call Blit_Or_Present
6376        ret
6377        db 0x01
6378        times 75 db 0
6379    D_098ba:
6380        times 8 db 0
6381    L_098c2:
6382        db 0x33, 0xc0      ; xor eax, eax
6383        call L_04a65
6384        mov esi, 0x986e
6385        mov ebx, dword [ptr_framebuffer_src]
6386        sub ebx, 0x280
6387        mov ebp, 0x14
6388    L_098df:
6389        mov eax, dword [esi]
6390        cmp eax, 0xa
6391        ja short L_09907
6392        cmp eax, 0
6393        je short L_09907
6394        cmp eax, 4
6395        jne short L_098f3
6396        inc dword [esi + 4]
6397    L_098f3:
6398        inc dword [esi]
6399        mov ecx, 0x280
6400        mul ecx
6401        shr ecx, 2
6402        db 0x8b, 0xfb      ; mov edi, ebx
6403        db 0x03, 0xf8      ; add edi, eax
6404        db 0x33, 0xc0      ; xor eax, eax
6405        rep stosd
6406    L_09907:
6407        add esi, 4
6408        mov eax, 0xa
6409        imul eax, eax, 0x280
6410        db 0x03, 0xd8      ; add ebx, eax
6411        dec ebp
6412        jne short L_098df
6413        db 0x33, 0xc0      ; xor eax, eax
6414        call Blit_Or_Present
6415        ret
6416    L_09922:
```

```
6417        call Blit_Banked_Path
6418        call L_098c2
6419        cmp dword [D_098ba], 0xa
6420        jne short L_09922
6421        call L_098c2
6422        ret
6423    D_0993b:
6424        db 0x00
6425    D_0993c:
6426        db 0x00, 0x00, 0x00, 0x00
6427    L_09940:
6428        mov eax, dword [D_0993c]
6429        mov dword [L_099db+3], eax
6430        sub eax, 4
6431        db 0xa3, 0x23, 0x9a, 0x00, 0x00     ; mov dword ptr [0x9a23], eax
6432        mov eax, 1
6433        mov cl, byte [D_0211c]
6434        shl eax, cl
6435        dec eax
6436        mov cl, byte [D_02119]
6437        shl eax, cl
6438        mov ebx, 1
6439        mov cl, byte [D_0211e]
6440        shl ebx, cl
6441        dec ebx
6442        mov cl, byte [D_0211b]
6443        shl ebx, cl
6444        db 0x0b, 0xc3     ; or eax, ebx
6445        mov ebx, 1
6446        mov cl, byte [D_0211d]
6447        shl ebx, cl
6448        dec ebx
6449        mov cl, byte [D_0211a]
6450        shl ebx, cl
6451        db 0x8b, 0xc8     ; mov ecx, eax
6452        db 0x8b, 0xd3     ; mov edx, ebx
6453        shl ecx, 0x10
6454        shl edx, 0x10
6455        db 0x0b, 0xd9     ; or ebx, ecx
6456        db 0x0b, 0xc2     ; or eax, edx
6457        db 0xa3, 0xee, 0x99, 0x00, 0x00     ; mov dword ptr [0x99ee], eax
6458        db 0xa3, 0xf4, 0x99, 0x00, 0x00     ; mov dword ptr [0x99f4], eax
6459        db 0xa3, 0x0d, 0x9a, 0x00, 0x00     ; mov dword ptr [0x9a0d], eax
6460        db 0x89, 0x1d, 0xfa, 0x99, 0x00, 0x00     ; mov dword ptr [0x99fa], ebx
6461        db 0x89, 0x1d, 0x00, 0x9a, 0x00, 0x00     ; mov dword ptr [0x9a00], ebx
6462        db 0x89, 0x1d, 0x13, 0x9a, 0x00, 0x00     ; mov dword ptr [0x9a13], ebx
6463        ret
6464    L_099c6:
6465        mov eax, dword [ptr_framebuffer_src]
6466        db 0xa3, 0xe5, 0x99, 0x00, 0x00     ; mov dword ptr [0x99e5], eax
6467        sub eax, 4
6468        db 0xa3, 0x2a, 0x9a, 0x00, 0x00     ; mov dword ptr [0x9a2a], eax
6469        ret
6470    L_099d9:
6471        db 0x33, 0xf6     ; xor esi, esi
6472    L_099db:
6473        mov eax, dword [esi*4 + 0x12345678]
6474        mov ebx, dword [esi*4 + 0x12345678]
6475        db 0x8b, 0xc8     ; mov ecx, eax
6476        db 0x8b, 0xd3     ; mov edx, ebx
6477        and eax, 0x12345678
6478        and ebx, 0x12345678
6479        and ecx, 0x12345678
6480        and edx, 0x12345678
6481        db 0x03, 0xc3     ; add eax, ebx
6482        rcr eax, 1
6483        db 0x03, 0xca     ; add ecx, edx
6484        rcr ecx, 1
6485        and eax, 0x12345678
```

```
6486        and ecx, 0x12345678
6487        inc esi
6488        db 0x0b, 0xc1      ; or eax, ecx
6489        cmp esi, 0x7d00
6490        mov dword [esi*4 + 0x12345678], eax
6491        mov dword [esi*4 + 0x12345678], eax
6492        jb short L_099db
6493        ret
6494    D_09a31:
6495        db 0x00, 0x00, 0x00, 0x00
6496    D_09a35:
6497        db 0x00, 0x00, 0x00, 0x00
6498    D_09a39:
6499        db 0x00, 0x00, 0x00, 0x00
6500    D_09a3d:
6501        db 0x00, 0x00, 0x00, 0x00
6502    D_09a41:
6503        db 0x00, 0x00, 0x00, 0x00
6504    D_09a45:
6505        db 0x00, 0x00, 0x00, 0x00
6506    D_09a49:
6507        db 0x00, 0x00, 0x00, 0x00
6508    D_09a4d:
6509        times 30 db 0
6510        add byte [eax], al
6511        add byte [eax], al
6512        add byte [eax], al
6513        add byte [eax], al
6514        add byte [eax], al
6515        add byte [eax], al
6516        add byte [eax], al
6517        add byte [eax], al
6518        add byte [eax], al
6519        add byte [eax], al
6520        add byte [eax], al
6521    BufPtr_09a81:
6522        add byte [eax], al
6523        add byte [eax], al
6524    BufPtr_09a85:
6525        add byte [eax], al
6526        add byte [eax], al
6527    BufPtr_09a89:
6528        add byte [eax], al
6529        add byte [eax], al
6530    BufPtr_09a8d:
6531        add byte [eax], al
6532        add byte [eax], al
6533    BufPtr_09a91:
6534        add byte [eax], al
6535        add byte [eax], al
6536    BufPtr_09a95:
6537        add byte [eax], al
6538        add byte [eax], al
6539    BufPtr_09a99:
6540        add byte [eax], al
6541        add byte [eax], al
6542    BufPtr_09a9d:
6543        add byte [eax], al
6544        add byte [eax], al
6545    BufPtr_09aa1:
6546        add byte [eax], al
6547        add byte [eax], al
6548    BufPtr_09aa5:
6549        add byte [eax], al
6550        add byte [eax], al
6551    BufPtr_09aa9:
6552        add byte [eax], al
6553        add byte [eax], al
6554    BufPtr_09aad:
```

```
6555        add byte [eax], al
6556        add byte [eax], al
6557    BufPtr_09ab1:
6558        add byte [eax], al
6559        add byte [eax], al
6560    BufPtr_09ab5:
6561        add byte [eax], al
6562        add byte [eax], al
6563    L_09ab9:
6564        or eax, 0x75654e0a
6565        db 0x72, 0x61     ; jb 0x9b21
6566        insb
6567        and byte [edi + 0x61], dl
6568        db 0x76, 0x65     ; jbe 0x9b2b
6569        and byte [esi + 0x69], ah
6570        outsb
6571        popad
6572        insb
6573        and byte [esi + 0x65], dh
6574        jb short L_09b44
6575        imul ebp, dword [edi + 0x6e], 0x54202d20
6576        inc edx
6577        dec esp
6578        daa
6579        cmp dword [edi], esi
6580        or eax, 0x5c6240a
6581        push ecx
6582        xchg ebx, eax
6583        add byte [eax], al
6584        add dword [ecx + 0x2067], esp
6585        mov dword [eax], 0
6586        ret
6587        mov byte [D_09351], 2
6588        mov eax, dword [D_02067]
6589        mov dword [eax], 0
6590        ret
6591        mov dword [D_02059], 1
6592        ret
6593        mov dword [D_0205d], 0x100
6594        mov dword [D_02061], 0xfffffff0
6595        ret
6596        mov dword [D_02061], 0x10
6597        ret
6598
6599    ; ---------------------------------------------------------------------------
6600    ; Palette_SwapRB / Palette_SwapGB - kleurkanalen omwisselen
6601    ;
6602    ; Loopt over 64K RGB-triples en wisselt R↔B of G↔B.
6603    ; Nodig als texture-/palette-data in een andere byte-volgorde staat
6604    ; dan de videokaart verwacht.
6605    ; ---------------------------------------------------------------------------
6606    Palette_SwapRB:
6607        mov ecx, 0x10000
6608    Palette_SwapRB_Loop:
6609        mov al, byte [esi]
6610        mov bl, byte [esi + 2]
6611        mov byte [esi], bl
6612        mov byte [esi + 2], al
6613        add esi, 3
6614        dec ecx
6615    L_09b44:
6616        jne short Palette_SwapRB_Loop
6617        ret
6618    Palette_SwapGB:
6619        mov ecx, 0x10000
6620    Palette_SwapGB_Loop:
6621        mov al, byte [esi + 1]
6622        mov bl, byte [esi + 2]
6623        mov byte [esi + 1], bl
```

```
6624        mov byte [esi + 2], al
6625        add esi, 3
6626        dec ecx
6627        jne short Palette_SwapGB_Loop
6628        ret
6629    Timeline_Cursor:
6630        db 0x63, 0x9b, 0x00, 0x00, 0x00, 0x30, 0xe0, 0x9a, 0x00, 0x00, 0x01, 0x30, 0xf3, 0x9a
            , 0x00, 0x00
6631        db 0x02, 0x00, 0x06, 0x9b, 0x00, 0x00, 0x02, 0x00, 0x11, 0x9b, 0x00, 0x00, 0x03, 0x26
            , 0x26, 0x9b
6632        db 0x00, 0x00, 0x03, 0x34, 0x06, 0x9b, 0x00, 0x00, 0x03, 0x34, 0x11, 0x9b, 0x00, 0x00
            , 0x05, 0x20
6633        db 0x26, 0x9b, 0x00, 0x00, 0x06, 0x00, 0x06, 0x9b, 0x00, 0x00, 0x07, 0x00, 0x06, 0x9b
            , 0x00, 0x00
6634        db 0x0a, 0x00, 0x06, 0x9b, 0x00, 0x00, 0x0b, 0x30, 0x26, 0x9b, 0x00, 0x00, 0x0c, 0x00
            , 0x06, 0x9b
6635        db 0x00, 0x00, 0x0c, 0x00, 0x11, 0x9b, 0x00, 0x00, 0x0d, 0x20, 0x26, 0x9b, 0x00, 0x00
            , 0x0d, 0x3c
6636        db 0x06, 0x9b, 0x00, 0x00, 0xff, 0xff, 0xf3, 0x9b, 0x00, 0x00, 0xff, 0xff, 0xf3, 0x9b
            , 0x00, 0x00
6637        db 0xff, 0xff, 0xf3, 0x9b, 0x00, 0x00, 0xff, 0xff, 0xf3, 0x9b, 0x00, 0x00, 0xff, 0xff
            , 0xf3, 0x9b
6638        db 0x00, 0x00, 0xff, 0xff, 0xf3, 0x9b, 0x00, 0x00, 0xff, 0xff, 0xf3, 0x9b, 0x00, 0x00
            , 0xff, 0xff
6639        db 0xf3, 0x9b, 0x00, 0x00
6640        ret
6641
6642    ; ---------------------------------------------------------------------------
6643    ; [5c] Timeline - events op tijd
6644    ;
6645    ; Houdt een "speeltijd" bij (D_0205d / D_02061) en loopt door een tabel
6646    ; (Timeline_Cursor). Elke entry: (tijd_hi, tijd_lo, functie-pointer).
6647    ; Als de huidige tijd ver genoeg is → call die functie, schuif cursor op.
6648    ; Zo worden scènes en effecten gesynchroniseerd (vaak met de muziek).
6649    ; ---------------------------------------------------------------------------
6650    Timeline_Update:
6651        mov eax, dword [D_0205d]
6652        mov ebx, dword [D_02061]
6653        db 0x03, 0xc3       ; add eax, ebx
6654        jns short Timeline_ClampLow
6655        db 0x33, 0xc0       ; xor eax, eax
6656    Timeline_ClampLow:
6657        cmp eax, 0x100
6658        jb short Timeline_ClampHigh
6659        mov eax, 0x100
6660    Timeline_ClampHigh:
6661        mov dword [D_0205d], eax
6662        call D_233ca
6663        call D_34d2d
6664        mov byte [D_02065], ah
6665        mov byte [D_02066], al
6666        call D_35248
6667        mov ebx, 0x3d860
6668        mul ebx
6669        sub edx, dword [D_02073]
6670        mov eax, dword [D_0206b]
6671        cmp eax, dword [D_0206f]
6672        je short Timeline_AfterTimer
6673        add dword [D_02073], edx
6674        db 0x33, 0xd2       ; xor edx, edx
6675    Timeline_AfterTimer:
6676        mov dword [D_0206b], edx
6677        mov dword [D_0206f], edx
6678    Timeline_RunEvents:
6679        mov esi, dword [Timeline_Cursor]
6680        mov al, byte [esi]
6681        cmp al, 0xff
6682        je short Program_Entry_AfterTimeline
6683        cmp byte [D_02065], al
```

```
6684        jl short Program_Entry_AfterTimeline
6685        mov al, byte [esi + 1]
6686        cmp byte [D_02066], al
6687        jl short Program_Entry_AfterTimeline
6688        call dword [esi + 2]
6689        add esi, 6
6690        mov dword [Timeline_Cursor], esi
6691        jmp short Timeline_RunEvents
6692    Program_Entry_AfterTimeline:
6693        ret
6694
6695    ; ============================================================================
6696    ; [5] PROGRAMMA-START + HOOFDSTRUCTUUR
6697    ;
6698    ; Vanaf hier begint de echte demo-logica (na de helpers hierboven).
6699    ;
6700    ; Volgorde in mensentaal:
6701    ;   1. Segmentregisters en DOS32-basisadres klaarzetten
6702    ;   2. VESA-schermmodus initialiseren
6703    ;   3. Floating-point unit (FPU) configureren
6704    ;   4. Grote werkbuffers alloceren (effecten, textures, 3D, …)
6705    ;   5. Data laden / voorbereiden
6706    ;   6. Timeline + script-engines draaien (zie hieronder)
6707    ;
6708    ; Daarnaast bestaan er twee "mini-talen" (bytecode-interpreters):
6709    ;   • Script_Dispatch_Opcode  - scène-/effect-opdrachten
6710    ;   • ProcessList_Dispatch    - 3D/object-operaties ("processlist")
6711    ;   • Timeline_RunEvents      - tijdgebonden events (t.o.v. muziek/timer)
6712    ; ============================================================================
6713        mov ax, 0
6714        mov ds, eax
6715        mov es, eax
6716        mov word [D_02028], ax
6717        sti
6718        mov eax, 0xee02
6719        int 0x31
6720        mov dword [dos32_base_offset], ebx
6721        sti
6722        call VESA_Init
6723        call Video_PickBestMode
6724        jb near Dos_Terminate
6725        wait
6726        fninit
6727        wait
6728        fnstcw word [D_02000]
6729        wait
6730        and word [D_02000], 0xfcff
6731        fldcw word [D_02000]
6732        mov edx, 0x532320
6733        call Dos32_GetSelectorBase
6734        jb near Dos_Terminate
6735        mov dword [D_0204d], edx
6736        db 0x8b, 0xfa     ; mov edi, edx
6737        db 0x33, 0xc0     ; xor eax, eax
6738        mov ecx, 0x14c8c8
6739        rep stosd
6740        mov dword [Heap_Base], edx
6741        ; Grote werkheap opsplitsen in vaste buffers (effecten, textures, 3D):
6742        add edx, 0x1ca320
6743        mov dword [Buf_A], edx
6744        add edx, 0x30000
6745        mov dword [Buf_B], edx
6746        add edx, 0x30000
6747        mov dword [Buf_C], edx
6748        add edx, 0x30000
6749        mov dword [Buf_D], edx
6750        add edx, 0x30000
6751        mov dword [D_093f6], edx
6752        add edx, 0x1000
```

```
6753        mov dword [D_093fa], edx
6754        add edx, 0x1000
6755        mov dword [D_093fe], edx
6756        add edx, 0x1000
6757        mov dword [D_0947f], edx
6758        add edx, 0x10000
6759        mov dword [D_09628], edx
6760        add edx, 0x400
6761        mov dword [D_0993c], edx
6762        add edx, 0x20000
6763        mov dword [BufPtr_09a81], edx
6764        add edx, 0x10000
6765        mov dword [BufPtr_09a85], edx
6766        add edx, 0x30000
6767        mov dword [BufPtr_09a89], edx
6768        add edx, 0x30000
6769        mov dword [BufPtr_09a8d], edx
6770        add edx, 0x30000
6771        mov dword [BufPtr_09a91], edx
6772        add edx, 0x30000
6773        mov dword [BufPtr_09a95], edx
6774        add edx, 0x30000
6775        mov dword [BufPtr_09a99], edx
6776        add edx, 0x30000
6777        mov dword [BufPtr_09a9d], edx
6778        add edx, 0x30000
6779        mov dword [BufPtr_09aa1], edx
6780        add edx, 0x30000
6781        mov dword [BufPtr_09aa5], edx
6782        add edx, 0x30000
6783        mov dword [BufPtr_09aa9], edx
6784        add edx, 0x80800
6785        mov dword [BufPtr_09aad], edx
6786        add edx, 0x10400
6787        mov dword [BufPtr_09ab1], edx
6788        add edx, 0x20000
6789        mov dword [BufPtr_09ab5], edx
6790        add edx, 0x4000
6791        call L_091f2
6792        call L_09940
6793        call D_35614
6794        call D_355d1
6795        mov esi, 0x23c48
6796        mov ebx, dword [BufPtr_09ab5]
6797        mov ecx, 0x10000
6798        db 0x33, 0xd2     ; xor edx, edx
6799        call D_34a7c
6800        db 0x33, 0xc0     ; xor eax, eax
6801        mov al, byte [D_02119]
6802        mov dword [D_1bd38], eax
6803        mov al, byte [D_0211a]
6804        mov dword [D_1bd3c], eax
6805        mov al, byte [D_0211b]
6806        mov dword [D_1bd40], eax
6807        mov al, byte [D_0211c]
6808        mov dword [D_1bd2c], eax
6809        mov al, byte [D_0211d]
6810        mov dword [D_1bd30], eax
6811        mov al, byte [D_0211e]
6812        mov dword [D_1bd34], eax
6813        mov dword [D_1bd18], 0x140
6814        mov dword [D_1bd1c], 0xc8
6815        call D_233bd
6816        mov dword [D_1bd08], D_09a31
6817        mov eax, dword [BufPtr_09a85]
6818        mov dword [D_09a35], eax
6819        mov eax, dword [BufPtr_09a89]
6820        mov dword [D_09a39], eax
6821        mov eax, dword [BufPtr_09a8d]
```

```
6822        mov dword [D_09a3d], eax
6823        mov eax, dword [BufPtr_09a8d]
6824        mov dword [D_09a3d], eax
6825        mov eax, dword [BufPtr_09a91]
6826        mov dword [D_09a41], eax
6827        mov eax, dword [BufPtr_09a95]
6828        mov dword [D_09a45], eax
6829        mov eax, dword [BufPtr_09a99]
6830        mov dword [D_09a49], eax
6831        mov eax, dword [BufPtr_09a9d]
6832        mov dword [D_09a4d], eax
6833        call L_07a4f
6834        call L_09304
6835        push eax
6836        push edx
6837        mov edx, L_09ab9
6838        mov ah, 9
6839        int 0x21
6840        pop edx
6841        pop eax
6842        mov esi, 0x74a3
6843        mov edi, dword [BufPtr_09a85]
6844        call ProcessList_Start
6845        mov esi, 0x753b
6846        mov edi, dword [BufPtr_09a89]
6847        call ProcessList_Start
6848        mov esi, 0x7297
6849        mov edi, dword [BufPtr_09a95]
6850        call ProcessList_Start
6851        mov esi, 0x72eb
6852        mov edi, dword [BufPtr_09a99]
6853        call ProcessList_Start
6854        mov esi, 0x7375
6855        mov edi, dword [BufPtr_09a9d]
6856        call ProcessList_Start
6857        mov esi, 0x7407
6858        mov edi, dword [BufPtr_09aa1]
6859        ; ProcessList-scripts: 3D-objectlijsten uit data-stream opbouwen
6860        call ProcessList_Start
6861        mov eax, dword [BufPtr_09aa1]
6862        mov dword [D_09345], eax
6863        mov esi, 0x741e
6864        mov edi, dword [BufPtr_09aa5]
6865        call ProcessList_Start
6866        mov eax, dword [BufPtr_09aa5]
6867        mov dword [D_09349], eax
6868        call L_07c99
6869        call L_09402
6870        db 0x33, 0xc0     ; xor eax, eax
6871        call Video_SetMode_320x200
6872        db 0x33, 0xc0     ; xor eax, eax
6873        call D_350b7
6874        mov byte [D_0993b], 1
6875        call Scene_InitHelper
6876        mov dword [D_0205d], 0x100
6877        mov eax, 0x100
6878        call D_233ca
6879        mov byte [D_0993b], 0
6880        call Scene_ResetState
6881        mov dword [D_07955], 0x792d
6882        mov esi, 0x78eb
6883        mov dword [D_06126], 0x168
6884        ; Start script-stream (ESI wijst naar bytecode) + hulpsetup
6885        call Script_PushEsi
6886        fild dword [D_06126]
6887        fstp dword [D_06126]
6888        mov dword [D_1bd44], 0xff
6889        call Scene_RunHelper
6890        mov byte [D_0993b], 1
```

```
6891        call Scene_ResetState
6892        mov dword [D_1bd44], 0xff
6893        mov byte [D_07b28], 1
6894        mov dword [D_07955], 0x78c7
6895        mov esi, 0x783d
6896        mov dword [D_06126], 0x168
6897        call Script_PushEsi
6898        fild dword [D_06126]
6899        fstp dword [D_06126]
6900        mov dword [D_07b29], L_07e9d
6901        call Scene_InitHelper2
6902        call L_0962c
6903        mov dword [D_02059], 0
6904
6905    ; ------------------------------------------------------------------------
6906    ; MainLoop_PartA - kernlus van een scène
6907    ;
6908    ; 1. Toetsenbord lezen (ESC/ENTER kan de scène beëindigen)
6909    ; 2. Timeline_Update → tijdgebonden events / effecten
6910    ; 3. Herhaal tot interne vlag D_02059 == 1 (scène klaar)
6911    ; Daarna: presenteren/blit, volgende script-scène laden, enz.
6912    ; ------------------------------------------------------------------------
6913    MainLoop_PartA:
6914        call Keyboard_ReadNormalized
6915        je near MainLoop_PartA_End
6916        call Timeline_Update
6917        cmp dword [D_02059], 1
6918        jne short MainLoop_PartA
6919        call L_09922
6920        mov dword [D_02059], 0
6921        mov eax, 0
6922        call Blit_Or_Present
6923        mov dword [D_02051], 0
6924        mov byte [D_0993b], 0
6925        call Scene_ResetState
6926        mov dword [D_07955], 0x76d6
6927        mov esi, 0x75a0
6928        mov dword [D_06126], 0x1f4
6929        call Script_PushEsi
6930        fild dword [D_06126]
6931        fstp dword [D_06126]
6932        mov dword [D_02000], 0x76c
6933        fild dword [D_02000]
6934        fmul dword [D_0759c]
6935        fstp dword [D_07951]
6936        call L_07959
6937        call L_0612e
6938        mov eax, 0
6939        call D_233ca
6940        mov edi, dword [BufPtr_09ab1]
6941        call D_233d6
6942        call Scene_ResetState
6943        mov dword [D_1bd44], 1
6944        mov eax, dword [BufPtr_09ab1]
6945        mov dword [D_1bd48], eax
6946        mov byte [D_07b28], 5
6947        mov dword [D_07b29], L_07e35
6948        call Scene_RunHelper2
6949        mov dword [D_0206b], 0
6950        mov byte [D_0993b], 1
6951        call Scene_ResetState
6952        mov dword [D_07955], 0x76d6
6953        mov esi, 0x75a0
6954        mov dword [D_06126], 0x1f4
6955        call Script_PushEsi
6956        fild dword [D_06126]
6957        fstp dword [D_06126]
6958        mov dword [D_1bd44], 0xff
6959        call Scene_RunHelper
```

```
    mov esi, dword [BufPtr_09a85]
    call Palette_SwapRB
    mov esi, dword [BufPtr_09a89]
    call Palette_SwapGB
    mov byte [D_0993b], 0
    call Scene_ResetState
    mov dword [D_07955], 0x7819
    mov esi, 0x76fa
    mov dword [D_06126], 0xb4
    call Script_PushEsi
    fild dword [D_06126]
    fstp dword [D_06126]
    mov dword [D_1bd44], 0xff
    call Scene_RunHelper
    mov ecx, 0x200
L_0a204:
    push ecx
    db 0x8b, 0xc1     ; mov eax, ecx
    shr eax, 3
    call D_34d17
    call Blit_Banked_Path
    call Keyboard_ReadNormalized
    pop ecx
    je short MainLoop_PartA_End
    dec ecx
    jne short L_0a204
MainLoop_PartA_End:
    call D_35128
    mov eax, 3
    int 0x10
    jmp near Dos_Terminate
Scene_RunHelper:
    mov eax, dword [D_02067]
    mov dword [eax], 0x76c
L_0a23b:
    mov eax, dword [D_02051]
    inc eax
    and eax, 3
    mov dword [D_02051], eax
    call L_04a65
    call Timeline_Update
    call L_099c6
    mov eax, dword [D_02067]
    mov eax, dword [eax]
    mov dword [D_02000], eax
    fild dword [D_02000]
    fmul dword [D_0759c]
    fstp dword [D_07951]
    call L_07959
    call L_0612e
    mov edi, dword [ptr_framebuffer_src]
    call D_233d6
    cmp byte [D_0993b], 1
    jne short L_0a299
    call L_099d9
L_0a299:
    mov eax, dword [D_02051]
    call Blit_Or_Present
    cmp dword [D_02059], 1
    je short L_0a2b3
    call Keyboard_ReadNormalized
    jne short L_0a23b
L_0a2b3:
    mov dword [D_02059], 0
    ret
Scene_InitHelper:
    mov eax, dword [D_02067]
    mov dword [eax], 0
L_0a2c9:
```

```
7029        call Timeline_Update
7030        call L_095d7
7031        cmp dword [D_02059], 1
7032        je short L_0a2e3
7033        call Keyboard_ReadNormalized
7034        jne short L_0a2c9
7035    L_0a2e3:
7036        mov dword [D_02059], 0
7037        ret
7038    Scene_InitHelper2:
7039        mov eax, dword [D_02067]
7040        mov dword [eax], 0
7041        mov eax, dword [D_1bd0c]
7042        mov dword [D_0a410], eax
7043        inc dword [D_1bd0c]
7044        mov dword [D_02055], 0
7045        mov dword [D_02059], 0
7046    L_0a31d:
7047        mov eax, dword [D_02051]
7048        inc eax
7049        and eax, 3
7050        mov dword [D_02051], eax
7051        call L_04a65
7052        call Timeline_Update
7053        call L_099c6
7054        mov eax, dword [D_02067]
7055        mov eax, dword [eax]
7056        mov dword [D_02000], eax
7057        fild dword [D_02000]
7058        fmul dword [D_07c70]
7059        fstp dword [D_07c78]
7060        fild dword [D_02000]
7061        fmul dword [D_0759c]
7062        fstp dword [D_07951]
7063        fild dword [D_02000]
7064        fmul dword [D_0a40c]
7065        fistp dword [D_02055]
7066        mov eax, dword [D_02055]
7067        call L_08d8f
7068        mov eax, dword [D_0a410]
7069        call L_07f77
7070        call L_07959
7071        mov dword [D_0611a], 0
7072        mov dword [D_0611e], 0
7073        mov dword [D_06122], 0
7074        fld dword [D_06112]
7075        fadd dword [D_0a414]
7076        fstp dword [D_06112]
7077        call L_0612e
7078        mov edi, dword [ptr_framebuffer_src]
7079        call D_233d6
7080        cmp byte [D_0993b], 1
7081        jne short L_0a3e3
7082        call L_099d9
7083    L_0a3e3:
7084        mov eax, dword [D_02051]
7085        call Blit_Or_Present
7086        cmp dword [D_02059], 1
7087        je short L_0a401
7088        call Keyboard_ReadNormalized
7089        jne near L_0a31d
7090    L_0a401:
7091        mov dword [D_02059], 0
7092        ret
7093    D_0a40c:
7094        db 0x00, 0x00, 0x00, 0x40
7095    D_0a410:
7096        db 0x00, 0x00, 0x00, 0x00
7097    D_0a414:
```

```
7098        db 0x00, 0x00, 0x40, 0x44
7099    Scene_RunHelper2:
7100        mov eax, dword [D_02067]
7101        mov dword [eax], 0
7102        mov dword [D_1bd0c], 1
7103        mov dword [D_02055], 0
7104        mov dword [D_02061], 0
7105        mov dword [D_0205d], 0
7106        mov dword [D_02059], 0
7107    L_0a455:
7108        mov eax, dword [D_02051]
7109        inc eax
7110        and eax, 3
7111        mov dword [D_02051], eax
7112        call L_04a65
7113        call Timeline_Update
7114        mov eax, dword [D_02067]
7115        mov eax, dword [eax]
7116        mov dword [D_02000], eax
7117        fild dword [D_02000]
7118        fmul dword [D_07c70]
7119        fstp dword [D_07c78]
7120        fild dword [D_02000]
7121        fmul dword [D_0a4e5]
7122        fistp dword [D_02055]
7123        mov eax, 0
7124        call L_07f77
7125        mov eax, 0
7126        call L_07b35
7127        mov edi, dword [ptr_framebuffer_src]
7128        call D_233d6
7129        mov eax, dword [D_02051]
7130        call Blit_Or_Present
7131        cmp dword [D_02059], 1
7132        je short L_0a4da
7133        call Keyboard_ReadNormalized
7134        jne near L_0a455
7135    L_0a4da:
7136        mov dword [D_02059], 0
7137        ret
7138    D_0a4e5:
7139        db 0x00, 0x00, 0x00, 0x40, 0x87, 0xdb, 0x90
7140    D_0a4ec:
7141        times 1024 db 0
7142    D_0a8ec:
7143        times 4096 db 0
7144    D_0b8ec:
7145        db 0x33, 0x03
7146        times 18 db 0
7147
7148    ; ---- bulk data (graphics, tables, music...) ----
7149        times 17392 db 0
7150    D_0fcf0:
7151        times 16388 db 0
7152    D_13cf4:
7153        times 16384 db 0
7154    D_17cf4:
7155        times 16384 db 0
7156    D_1bcf4:
7157        db 0x00, 0x00, 0x00, 0x00
7158    D_1bcf8:
7159        db 0x00, 0x00, 0x00, 0x00
7160    D_1bcfc:
7161        db 0x00, 0x00, 0x00, 0x00
7162    D_1bd00:
7163        db 0x00, 0x00, 0x00, 0x00
7164    D_1bd04:
7165        db 0x00, 0x00, 0x00, 0x00
7166    D_1bd08:
```

```
7167        db 0x00, 0x00, 0x00, 0x00
7168    D_1bd0c:
7169        db 0x00, 0x00, 0x00, 0x00
7170    D_1bd10:
7171        db 0x00, 0x00, 0x00, 0x00
7172    D_1bd14:
7173        db 0x00, 0x00, 0x00, 0x00
7174    D_1bd18:
7175        db 0x40, 0x01, 0x00, 0x00
7176    D_1bd1c:
7177        db 0xc8, 0x00, 0x00, 0x00, 0x80, 0x02, 0x00, 0x00, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00
            , 0x00, 0x00
7178    D_1bd2c:
7179        db 0x05, 0x00, 0x00, 0x00
7180    D_1bd30:
7181        db 0x06, 0x00, 0x00, 0x00
7182    D_1bd34:
7183        db 0x05, 0x00, 0x00, 0x00
7184    D_1bd38:
7185        db 0x0b, 0x00, 0x00, 0x00
7186    D_1bd3c:
7187        db 0x05, 0x00, 0x00, 0x00
7188    D_1bd40:
7189        db 0x00, 0x00, 0x00, 0x00
7190    D_1bd44:
7191        db 0x00, 0x00, 0x00, 0x00
7192    D_1bd48:
7193        times 56 db 0
7194        incbin "assets/data_1bd80.bin"   ; 1013 bytes
7195        times 108 db 0
7196        incbin "assets/data_1c1e1.bin"   ; 1363 bytes
7197        times 2048 db 0
7198        incbin "assets/data_1cf34.bin"   ; 657 bytes
7199        times 28 db 0
7200        incbin "assets/data_1d1e1.bin"   ; 527 bytes
7201        times 12292 db 0
7202        incbin "assets/data_203f4.bin"   ; 1516 bytes
7203        times 75 db 0
7204        db 0x3f, 0x00, 0x00, 0x80, 0x47
7205        times 64 db 0
7206        incbin "assets/data_20a70.bin"   ; 1742 bytes
7207        times 64 db 0
7208        db 0x87, 0xdb, 0x00, 0x00, 0x00, 0x3f, 0x00, 0x00, 0x80, 0x47
7209        times 64 db 0
7210        incbin "assets/data_211c8.bin"   ; 2172 bytes
7211        times 107 db 0
7212        db 0x3f, 0x00, 0x00, 0x80, 0x47
7213        times 64 db 0
7214        incbin "assets/data_21af4.bin"   ; 2657 bytes
7215        times 112 db 0
7216        db 0x87, 0xdb, 0x90, 0x00, 0x00, 0x00, 0x3f, 0x00, 0x00, 0x80, 0x47
7217        times 64 db 0
7218        incbin "assets/data_22610.bin"   ; 1882 bytes
7219        times 72 db 0
7220        db 0x87, 0xdb, 0x00, 0x00, 0x00, 0x3f, 0x00, 0x00, 0x80, 0x47
7221        times 64 db 0
7222        incbin "assets/data_22dfc.bin"   ; 613 bytes
7223        times 40 db 0
7224        incbin "assets/data_23089.bin"   ; 592 bytes
7225        times 228 db 0
7226    D_233bd:
7227        db 0xe8, 0x1f, 0x9e, 0xff, 0xff, 0x33, 0xc0, 0xe8, 0x58, 0x9f, 0xff, 0xff, 0xc3
7228    D_233ca:
7229        db 0xe8, 0x52, 0x9f, 0xff, 0xff, 0xc3
7230    D_233d0:
7231        db 0xe8, 0x70, 0x9c, 0xff, 0xff, 0xc3
7232    D_233d6:
7233        incbin "assets/data_233d6.bin"   ; 2258 bytes
7234        times 240 db 0
```

```
7235        db 0x50, 0x07, 0x00, 0x00, 0x60, 0x08, 0x00, 0x00, 0x70, 0x09, 0x00, 0x00, 0x80, 0x0a
            , 0x00, 0x00
7236        db 0x90, 0x0b, 0x00, 0x00, 0xa0, 0x0c, 0x00, 0x00, 0xb0, 0x0d, 0x00, 0x00, 0xc0, 0x0e
            , 0x00, 0x00
7237        db 0xd0, 0x0f
7238        times 478 db 0
7239        db 0xe0, 0x10, 0x00, 0x00, 0x1e, 0x18, 0x00, 0x00, 0x08, 0x22, 0x00, 0x00, 0xbc, 0x2b
            , 0x00, 0x00
7240        db 0x24, 0x36, 0x00, 0x00, 0xc8, 0x46, 0x00, 0x00, 0xd6, 0x58, 0x00, 0x00, 0xb5, 0x63
            , 0x00, 0x00
7241        db 0xde, 0x6f
7242        times 990 db 0
7243        db 0x01
7244        times 106 db 0
7245        db 0xff, 0x00, 0xff, 0x00, 0x00, 0x40
7246        times 46 db 0
7247        db 0xff, 0x00, 0xff, 0x00, 0x00, 0x20
7248        times 94 db 0
7249        db 0x14, 0x00, 0x00, 0x80, 0x00, 0x00, 0x08, 0x40, 0x80, 0x00, 0x00, 0x00, 0x00, 0x01
7250        times 106 db 0
7251        db 0xff, 0x00, 0xff, 0x00, 0x00, 0x40
7252        times 46 db 0
7253        db 0xff, 0x00, 0xff, 0x00, 0x00, 0x20
7254        times 91 db 0
7255        db 0x81, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0b, 0x1d, 0x00, 0x00, 0x40, 0x80, 0xc8, 0x1c
            , 0x00, 0x00
7256        db 0x01
7257        times 106 db 0
7258        db 0xff, 0x00, 0xff, 0x00, 0x00, 0x40
7259        times 46 db 0
7260        db 0xff, 0x00, 0xff, 0x00, 0x00, 0x20
7261        times 91 db 0
7262        db 0x0c, 0x1d, 0x00, 0x00, 0x00, 0x00, 0x7f, 0x43, 0x00, 0x00, 0x40, 0x80, 0xc8, 0x1c
            , 0x00, 0x00
7263        db 0x01
7264        times 106 db 0
7265        db 0xff, 0x00, 0xff, 0x00, 0x00, 0x40
7266        times 46 db 0
7267        db 0xff, 0x00, 0xff, 0x00, 0x00, 0x20
7268        times 91 db 0
7269        db 0x80, 0x43, 0x00, 0x80, 0x43, 0x00, 0x00, 0x44, 0x00, 0x08, 0x40, 0x80, 0x68, 0x00
            , 0x00, 0x00
7270        db 0x01
7271        times 106 db 0
7272        db 0xff, 0x00, 0xff, 0x00, 0x00, 0x40
7273        times 46 db 0
7274        db 0xff, 0x00, 0xff, 0x00, 0x00, 0x20
7275        times 91 db 0
7276        db 0x01, 0x44, 0x00, 0x01, 0x44, 0x00, 0x81, 0x44, 0x00, 0x08, 0x40, 0x80, 0x68, 0x00
            , 0x00, 0x00
7277        db 0x01
7278        times 106 db 0
7279        db 0xff, 0x00, 0xff, 0x00, 0x00, 0x40
7280        times 46 db 0
7281        db 0xff, 0x00, 0xff, 0x00, 0x00, 0x20
7282        times 91 db 0
7283        db 0x82, 0x44, 0x00, 0x82, 0x44, 0x00, 0x02, 0x45, 0x00, 0x08, 0x40, 0x80, 0x68, 0x00
            , 0x00, 0x00
7284        db 0x01
7285        times 106 db 0
7286        db 0xff, 0x00, 0xff, 0x00, 0x00, 0x40
7287        times 46 db 0
7288        db 0xff, 0x00, 0xff, 0x00, 0x00, 0x20
7289        times 91 db 0
7290        db 0x03, 0x45, 0x00, 0x5f, 0x4b, 0x00, 0x9f, 0x4b, 0x00, 0x08, 0x40, 0x80, 0x00, 0x00
            , 0x00, 0x00
7291        db 0x01
7292        times 106 db 0
7293        db 0xff, 0x00, 0xff, 0x00, 0x00, 0x40
```

```
7294        times 46 db 0
7295        db 0xff, 0x00, 0xff, 0x00, 0x00, 0x20
7296        times 91 db 0
7297        db 0xa0, 0x4b, 0x00, 0x00, 0x00, 0x00, 0x1f, 0x53, 0x00, 0x00, 0x40, 0x80, 0xc8, 0x1c
            , 0x00, 0x00
7298        db 0x01
7299        times 106 db 0
7300        db 0xff, 0x00, 0xff, 0x00, 0x00, 0x40
7301        times 46 db 0
7302        db 0xff, 0x00, 0xff, 0x00, 0x00, 0x20
7303        times 91 db 0
7304        incbin "assets/data_24d18.bin"   ; 64868 bytes
7305    D_34a7c:
7306        incbin "assets/data_34a7c.bin"   ; 667 bytes
7307    D_34d17:
7308        db 0x55, 0xe8, 0x03, 0xfd, 0xff, 0xff, 0x88, 0x45, 0x01, 0x5d, 0xc3, 0x55, 0xe8, 0xf8
            , 0xfc, 0xff
7309        db 0xff, 0x8a, 0x45, 0x02, 0x5d, 0xc3
7310    D_34d2d:
7311        incbin "assets/data_34d2d.bin"   ; 906 bytes
7312    D_350b7:
7313        incbin "assets/data_350b7.bin"   ; 113 bytes
7314    D_35128:
7315        incbin "assets/data_35128.bin"   ; 288 bytes
7316    D_35248:
7317        incbin "assets/data_35248.bin"   ; 905 bytes
7318    D_355d1:
7319        incbin "assets/data_355d1.bin"   ; 67 bytes
7320    D_35614:
7321        incbin "assets/data_35614.bin"   ; 272 bytes
7322        times 91 db 0
7323        db 0x01
7324        times 79 db 0
7325        db 0x8a, 0x00, 0x00, 0x00, 0xa2, 0x00, 0x00, 0x00, 0xc9
7326        times 27 db 0
7327        db 0x03, 0x01
7328        times 170 db 0
7329        db 0x62, 0x00, 0x00, 0x00, 0x5e, 0x00, 0x00, 0x00, 0x5a, 0x00, 0x00, 0x00, 0x56, 0x00
            , 0x00, 0x00
7330        db 0x52
7331        times 103 db 0
7332        db 0xc9, 0x00, 0x00, 0x00
7333
7334    image_end:
7335        times 0x3591b-($-$$) db 0xcc   ; build fails if the image grew (negative count)
7336
```