

Make jumpman invincible

```
19B3 CD0828 CALL #2808 ; check for collisions with hostile sprites [set to NOPS to make mario invincible to enemy sprites]
```

Dus #19B3 3xNOP maken.

De HIGH SCORE aanpassen naar STOLEN LADDERS

```
368A:          96 76 17 11 1D          ..GAM
3690: 15 10 10 1F 26 15 22 3F 94 76 20 1C 11 29 15 22 E..OVER...PLAYER
36A0: 10 30 32 31 3F 94 76 20 1C 11 29 15 22 10 30 33 .<I>...PLAYER.<2
36B0: 31 3F 80 76 18 19 17 18 10 23 13 1F 22 15 3F 9F >...HIGH.SCORE..
36C0: 75 13 22 15 14 19 24 10 10 10 10 3F 5E 77 18 1F .CREDIT.....HO
36D0: 27 10 18 19 17 18 10 13 11 1E 10 29 1F 25 10 17 W.HIGH.CAN.YOU.G
36E0: 15 24 10 FB 10 3F 29 77 1F 1E 1C 29 10 01 10 20 ET.?....ONLY.1.P
```

De CREDIT opschuiven door de spaties aan het einde weg te halen:

```
#36C3 9F 75 13 22 15 14 19 24 3F
```

De aanroep van CREDIT ook aanpassen:

```
#3655: C3 36      5    ; #36C3 "CREDIT"
```

En dan:

```
#36B2: C0 76 23 24 1F 1C 15 1E 10 1C 11 14 14 15 22 23 3F
```

Het titelscherm aanpassen

De KONG letters 1 positie naar beneden verschuiven:

```
3D59: 05 30 77 05 10 77 02 F1 76 02 D0 76 02 D3 76 ; K
3D68: 05 90 76 05 70 76 01 50 76 01 54 76 05 30 76 ; O
3D77: 05 F0 75 02 D1 75 02 B2 75 05 90 75 ; N
3D83: 03 51 75 05 30 75 01 10 75 01 14 75 ; G (part 1)
3D8F: 01 F0 74 01 F2 74 01 F4 74 02 D2 74 ; G (part 2)
3D9B: 00 ; end code
```

En het TM symbool weghalen.

```
081C CD243F CALL #3F24 ; draw TM logo onscreen [patch? orig japanese had 3 NOPs here]
```

Hiermee wordt het stukje extra code dat het TM-symbool tekent ook vrijgemaakt:

```
#3F24 t/m #3F2F.
```

Dus op #081C weer 3 NOP's maken.

Dan kan het stuk van #3F00 t/m #3F23 ook vrijgemaakt worden en gebruikt worden voor de teksten in het titelscherm.

```
3687: 00 3F      1E      ; #3F00 "(C) 1981"  
3689: 09 3F      1F      ; #3F09 "NINTENDO OF AMERICA"
```

In #3F00 de copyright notice maken:

© 1981,2020 NINTENDO

5C 77 49 4A 10 01 09 08 01 43 02 00 02 00 10 1E 19 1E 24 15 1E 14 1F 3F

En de plek van de checksum ook aanpassen: INTEND: start op 3F10:

2441 21103F LD HL,#3F10 ; load HL with ROM area that has NINTENDO written

En dan in #3F8F de titel maken:

STOLEN LADDERS

CE 76 23 24 1F 1C 15 1E 10 1C 11 14 14 15 22 23 3F

En de verwijzing ernaar toe aanpassen:

3689: 8F 3F

Nu nog de Donkey Kong een regel naar beneden verplaatsen.

Wordt hier gedaan:

```
0820 210869 LD HL,#6908 ; load HL with start of kong sprite X pos  
0823 0E44 LD C,#44 ; load C with offset to add X  
0825 FF RST #38 ; draw kong in new position  
0826 210869 LD HL,#6908 ; load HL with start of kong sprite Y pos  
0829 0E78 LD C,#78 ; load C with offset to add Y  
082B FF RST #38 ; draw kong  
082C C9 RET ; return
```

Kan door aanpassen:

0829: 0E80 LD C,#80 ; load C with offset to add Y

Liefste een versie toevoegen aan het titel scherm. Eerste versie opnemen is v1.00.

De 1 is nog unused:

364B:	8B 36	0	; #368B "GAME OVER"
364D:	01 00	1	; unused ?
364F:	98 36	2	; #3698 "PLAYER <I>"
3651:	A5 36	3	; #36A5 "PLAYER <II>"
3653:	B2 36	4	; #36B2 "HIGH SCORE"
3655:	BF 36	5	; #36BF "CREDIT"
3657:	06 00	6	; unused ?
3659:	CC 36	7	; #36CC "HOW HIGH CAN YOU GET?"

Deze laten verwijzen naar het stukje met spaties in de Name:

3710:	24 24 1F 1E 3F 27 76 20 25 23 18 3F 06 77 1E 11	TTON...PUSH...NA
3720:	1D 15 10 22 15 17 19 23 24 22 11 24 19 1F 1E 3F	ME.REGISTRATION.
3730:	88 76 1E 11 1D 15 2E 3F E9 75 2D 2D 2D 10 10 10	..NAME:-----
3740:	10 10 10 10 10 10 3F 0B 77 11 10 12 10 13 10 14A.B.C.D
3750:	10 15 10 16 10 17 10 18 10 19 10 1A 3F 0D 77 1B	.E.F.G.H.I.J...K
3760:	10 1C 10 1D 10 1E 10 1F 10 20 10 21 10 22 10 23	.L.M.N.O.P.Q.R.S
3770:	10 24 3F 0F 77 25 10 26 10 27 10 28 10 29 10 2A	.T...U.V.W.X.Y.Z

Dus daartoe #373D veranderen in 3F.

En dan de versie opnemen in stuk daarna:

#373E: 3D 76 26 01 2B 00 00 10 3F (V1.00)

en

#364D: 3E 37

Dit moet dan aangeroepen worden bij het tekenen van het introscherm:

080C	111E03	LD	DE,#031E	; load task data for text "(C) 1981"
080F	CD9F30	CALL	#309F	; insert task to draw text
0812	13	INC	DE	; load task data for text "NINTENDO OF AMERICA"
0813	CD9F30	CALL	#309F	; insert task to draw text
0816	21CF39	LD	HL,#39CF	; load HL with table data for kong beating chest
0819	CD4E00	CALL	#004E	; update kong's sprites
081C	CD243F	CALL	#3F24	; draw TM logo onscreen [patch? orig japanese had 3 NOPs here]
081F	00	NOP		; no operation
0820	210869	LD	HL,#6908	; load HL with start of kong sprite X pos
0823	0E44	LD	C,#44	; load C with offset to add X
0825	FF	RST	#38	; draw kong in new position
0826	210B69	LD	HL,#690B	; load HL with start of kong sprite Y pos
0829	0E78	LD	C,#78	; load C with offset to add Y
082B	FF	RST	#38	; draw kong
082C	C9	RET		; return

We hadden de #081C drie NOP's gemaakt (niet meer TM logo tekenen). Daar kan dan mooi een aanroep naar een stuk additionele code:

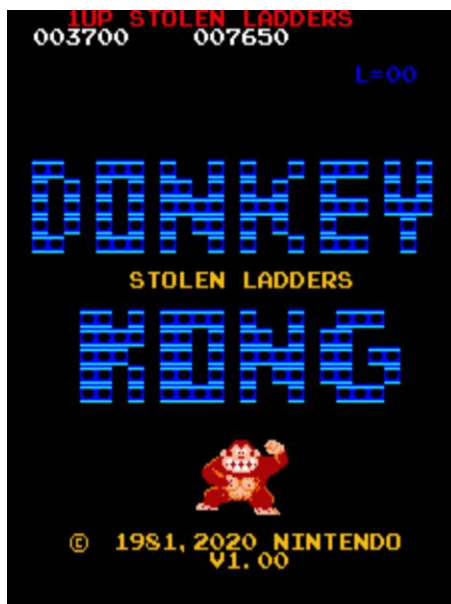
081C: C3863F JP #3F86 ; jump to additional code to display version

3F86: 110103 LD DE,#0301 ; load task data for text "V1.00"

3F89: CD9F30 CALL #309F ; insert task to draw text

3F8C: C31F08 JP #081F ; jump back

Het titelscherm ziet er nu zo uit:



In DK Twisted worden ook twee ladders toegevoegd. Eventjes terughalen hoe dat gedaan wordt. Dit wordt gedaan vanuit een stuk code die de hele tijd doorlopen wordt (vanuit de main) en die het switchen checkt (door hartje heen). Als dat gedaan wordt, dan worden de ladders getekend met:

```
2535:    CD F0 24    CALL #24F0            ; draw lower ladder
```

en

```
2563:    CD 07 25    CALL #2507            ; draw upper ladder
```

De stukjes die de ladders tekenen zijn dan:

```
24EA:  00 AB 55 AB 71 AA                ; test table data removed ladder
24F0:  11EA24    LD DE,#24EA            ; load start table data
24F3:  CDA70D    CALL #0DA7             ; draw screen
24F6:  21EA24    LD    HL,#24EA        ; load HL with start of table data for ladder
24F9:  DD210B63 LD    IX,#630B         ; #6300 is start, with 11 ladders added: 630B
24FD:  CD7524    CALL  #2475
2500:  C9          RET
```

en

```
2501:  00 93 38 93 54 AA                ; test table data removed ladder
2507:  110125    LD DE,#2501            ; load start table data
250A:  CDA70D    CALL #0DA7             ; draw screen
250D:  210125    LD    HL,#2501        ; load HL with start of table data for ladder
2510:  DD210C63 LD    IX,#630C         ; #6300 is start, with 11 ladders added: 630C
2514:  CD7524    CALL  #2475
2517:  C9          RET
```

De call naar #0DA7 tekent de ladder op het scherm. De call naar #2475 voegt de ladder toe aan de datastructuur met ladders zodat deze herkend wordt (invullen vanaf #6300 met index voor het aantal ladders dat reeds getekend is).

Het idee is om in een screen de 4 ladders die eventueel gestolen kunnen worden uit de velddefinitie te halen en in een aparte definitie te stoppen (of voor sommige screens twee setjes van 4 eruit halen). En dan de index te bepalen (hoeveel ladders zijn er dan al getekend) zodat je daar rekening mee kunt houden (voor sommige screens twee indexen bepalen).

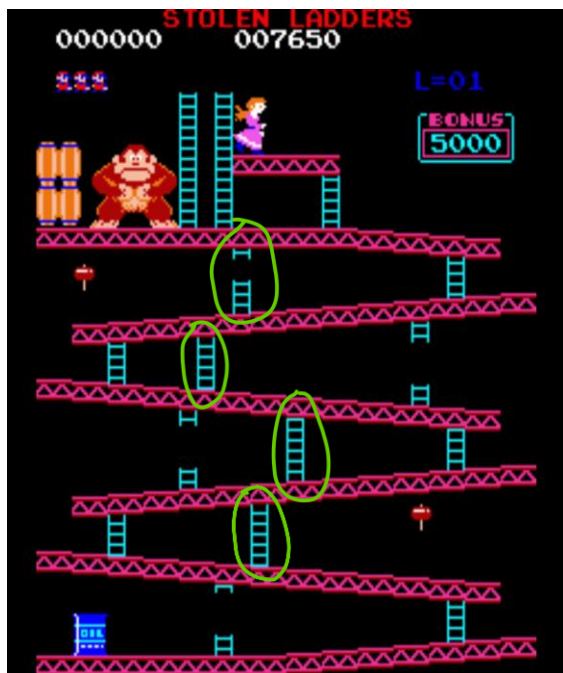
En dan na het tekenen van het scherm drie van de vier ladders één voor één toevoegen. Waarbij er dus één weggelaten wordt op basis van een random getal tussen 0 en 3.

De structuur van de set met vier ladders:

```
00 XT YT XB YB AA ; eerste ladder
01 XT YT XB YB AA ; tweede ladder
00 XT YT XB YB AA ; derde ladder
01 XT YT XB YB AA ; vierde ladder
```

Dit kost dus 4 bytes extra t.o.v. de structuur die er was (4 keer een afsluitende AA, nodig om de ladder te kunnen toevoegen: de calls die dat doen stoppen als er een AA gelezen wordt).

Eerst het barrels screen aanpassen. De volgende trappen gaan gebruiken:



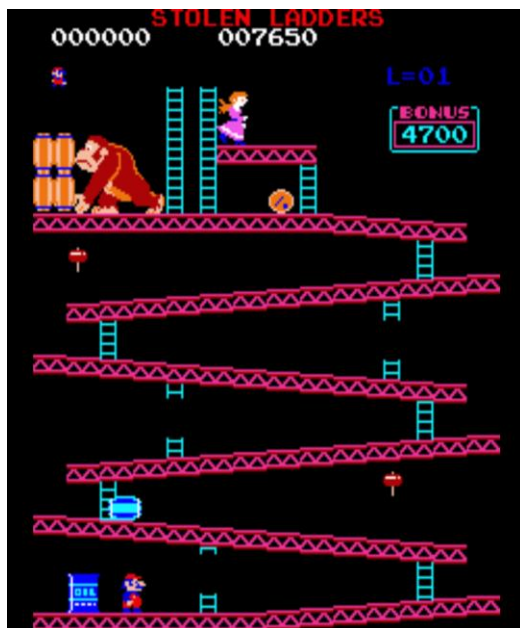
Deze dan uit de definitie halen:

```
3AE4: 02 97 38 68 38 ; top girder where girl sits
3AE9: 02 9F 54 10 54 ; girder where kong sits
3AED: 02 DF 58 A0 55 ; 1st slanted girder at top right
3AF3: 02 EF 6D 20 79 ; 2nd slanted girder (has hammer at left side)
3AF8: 02 DF 9A 10 8E ; 3rd slanted girder
3AFD: 02 EF AF 20 BB ; 4th slanted girder
3B02: 02 DF DC 10 D0 ; 5th slanted girder (has hammer at right side)
```

3B07: 02 FF F0 80 F7 ; bottom slanted girder
 3B0C: 02 7F F8 00 F8 ; bottom flat girder where mario starts
 3B11: 00 CB 57 CB 6F ; short ladder at top right
 3B16: 00 CB 99 CB B1 ; short ladder at center right
 3B1B: 00 CB DB CB F3 ; short ladder at bottom right
 3B20: 00 63 18 63 54 ; kong's ladder (right)
 3B25: 01 63 D5 63 F8 ; bottom broken ladder
 3B2A: 00 33 78 33 90 ; short ladder at left side under top hammer
 3B2F: 00 33 BA 33 D2 ; short ladder at left side above oil can
 3B34: 00 53 18 53 54 ; kong's ladder (left)
 3B39: 01 53 92 53 B8 ; second broken ladder from bottom, on 3rd girder
 3B3E: ~~00 5B 76 5B 92~~ ; longer ladder under the top left hammer
 3B43: ~~00 73 B6 73 D6~~ ; longer ladder to left of bottom hammer
 3B48: ~~00 83 95 83 B5~~ ; center longer ladder
 3B4D: 00 93 38 93 54 ; ladder leading to girl
 3B52: 01 BB 70 BB 98 ; third broken ladder on right side near top
 3B57: ~~01 6B 54 6B 75~~ ; fourth broken ladder near kong
 3B5C: AA ; AA code signals end of data

Eerst deze uit de code weghalen en de rest opschuiven. De definitie loopt nu van #3AE4 t/m #3B48.

Geeft dit scherm (dus de juiste weggehaald):



De weggehaalde delen vormen de volgende stolen ladder definitie:

3B49: 00 5B 76 5B 92 AA ; longer ladder under the top left hammer
 3B4F: 00 73 B6 73 D6 AA ; longer ladder to left of bottom hammer
 3B55: 00 83 95 83 B5 AA ; center longer ladder
 3B5B: 01 6B 54 6B 75 AA ; fourth broken ladder near kong

Deze definitie loopt nu van #3B49 t/m #3B60. Dit is dus een stuk in de definitie van het conveyors screen. Die moet later opgeschoven worden.

Het aantal ladders dat al wel getekend is 11: daardoor is de index #630B.

Nu eerst een stuk code maken die alle vier de ladders tekent en aan de definitie toevoegd zodat getest kan worden of dit werkt. Pas daarna aan die code het random weglaten van één van de ladders toevoegen.

Waar kun je dit aanroepen? lig na het tekenen van het scherm en na het toevoegen van de definitie van de overgebleven ladders.

Het eerste gebeurt in het stuk code tussen

Het tweede gebeurt in het stuk code tussen #2441 en #24B3. Dit wordt hier aangeroepen:

```
005F CD560F CALL #0F56 ; clear and initialize RAM values, compute initial timer, draw all initial sprites
0062 CD4124 CALL #2441 ;
0065 210960 LD HL,WaitTimerMSB ; load HL with timer addr.
0068 3640 LD (HL),#40 ; set timer to #40
006A 23 INC HL ; HL := GameMode2
006B 34 INC (HL) ; increase game mode2
006C 215C38 LD HL,#385C ; load HL with start of kong graphic table data
006F CD4E00 CALL #004E ; update kong's sprites

0072 110069 LD DE,#6900 ; set destination to girl sprite
0075 010800 LD BC,#0008 ; set counter to 8
0078 EDB0 LDIR ; draw the girl on screen

007A 3A2762 LD A,(#6227) ; load a with screen number
007D fe04 CP #04 ; is this rivets screen?
007f 280A JR Z,#0D8b ; if yes, jump ahead a bit
```

Dus de additionele code om de ladders te tekenen invoegen na de call naar #2441. Dus de #0D65 veranderen in een JP naar de addionele code en dan die code voor het terugspringen naar #0D68 aanvullen met de oorspronkelijke instructie van #06D5.

In de additionele code eerst testen op het screen. Dan afhankelijk van het screen de HL vullen met het startadres van de definitie van de stolen ladders en IX vullen met de ladder index. En dan springen naar de routine die het daadwerkelijke tekenen doet.

0D65: C3183F JP #3F18 ; jump to additional code

De additionele code wordt dan (op #3F18):

```
3F18: 3A2762 LD A,(#6227) ; load A with screen number
3F1B: FE01 CP #01 ; are we on the girders?
3F1D: C22A3F JP NZ,#3F2A ; no, skip ahead – jump to label AAAA
;
3F20: 21493B LD HL,#3B49 ; load HL with stolen ladders of girders
3F23: DD210B63 LD IX,#630B ; load IX with index for girders
3F27: CD303F CALL #3F30 ; call additional code to add stolen ladders
```

```

;
3F2A: 210960    LD  HL, WaitTimerMSB ; load HL with timer addr – label AAAA
3F2D: C3680D    JP  #0D68             ; jump back

en

3F30: 0604      LD  B,#04             ; for b = 1 to 4
;
3F32: E5        PUSH HL             ; save HL - label BBBB
3F33: C5        PUSH BC             ; save BC
3F34: EB        EX  DE,HL           ; DE <> HL
3F35: CDA70D    CALL #0DA7          ; draw ladder to screen
3F38: C1        POP  BC             ; restore BC
3F39: E1        POP  HL             ; restore HL
3F3A: CD7524    CALL #2475          ; define ladder and increase IX, HL is at AA at end
3F3D: 23        INC  HL             ; increment HL to point to next ladder
;
3F3E: 10F2      DJNZ #BBBB          ; next b – jump relative to label BBBB
3F40: C9        RET

```

De loop van vier werkt niet. Maar als #3F30, #3F31 NOP gemaakt worden en 3F3D een RET gemaakt wordt, dan wordt er wel één ladder getekend en daar kan je ook nog op. Maar als de loop er wel in zit worden er heel veel ladders getekend. Wellicht dat de relative jump niet goed is? Klopt was F1 maar moet F2 zijn. Aangepast en dan wel goed.

Hier dan het random weglaten van één van de vier ladders aan toevoegen. A als parameter gebruiken: als A 5 is, dan geen ladder weglaten. Als A 1 is, dan wel een ladder weglaten. Daarmee kun je blokken maken die later pas een ladder weglaten. Dus afhankelijk van level nr.

Stukje voorbeeld code om random nummer te genereren:

```

218C 3A1860    LD   A,(RngTimer1) ; load A with random timer ?
218F 4F        LD   C,A           ; store into C for later use ?
2190 E603      AND  #03           ; mask bits. result now random number between 0 and 3
2192 B8        CP   B             ; compare with value computed above based on skill

```

Random nummer genereren vóór de loop. En dan in de loop testen of B gelijk is aan het random nummer. Zo ja, dan geen ladder toevoegen, maar alleen de HL met 6 ophogen. HL ophogen met een getal kan alleen met een ADD HL,DE. Maar dan moet je DE eerst vullen dus dat is al 6 opcodes tezamen. Een simpele INC HL is 1 opcode. Dan kun je net zo goed 6 keer INC HL doen.

Stukje voorbeeld code om HL te verhogen:

```

0034 23        INC  HL           ; next table entry

```

De code wordt dan:

```

3F50: FE05      CP  #05             ; A = 5 ?
3F52: C2593F      JP  NZ, #3F59        ; no, skip next steps – jump to label CCCC
3F55: 4F          LD  C,A             ; C = A = 5

```



```

3F56: C3603F    JP #3F60          ; skip next steps – jump to label DDDD
3F59: 3A1860    LD A,(RngTimer1) ; load A with random timer ? – label CCCC
3F5C: E603       AND #03          ; mask bits: result is random # between 0 and 3
3F5E: 3C        INC A            ; increment A: result is random # between 1 and 4
3F5F: 4F        LD C,A          ; C = A = random # between 1 and 4
;
3F60: 0604     LD B,#04          ; for B = 1 to 4 – label DDDD
;
3F62: 78        LD A, B          ; A = B - label BBBB
3F63: B9        CP C            ; A = C ? : that is: B is equal to C?
3F64: C26F3F    JP NZ, #3F6F       ; no, skip next steps – jump to label EEEE
;
3F67: 23        INC HL          ; next table entry: 5 times to skip drawing this ladder
3F68: 23        INC HL          ; next table entry
3F69: 23        INC HL          ; next table entry
3F6A: 23        INC HL          ; next table entry
3F6B: 23        INC HL          ; next table entry
3F6C: C37A3F    JP #3F7A          ; skip next steps – jump to label FFFF
;
3F6F: E5        PUSH HL         ; save HL - label EEEE
3F70: C5        PUSH BC         ; save BC
3F71: EB        EX DE,HL        ; DE <> HL
3F72: CDA70D    CALL #0DA7         ; draw ladder to screen
3F75: C1        POP BC          ; restore BC
3F76: E1        POP HL          ; restore HL
3F77: CD7524    CALL #2475       ; define ladder and increase IX, HL is at AA at end
;
3F7A: 23        INC HL          ; increment HL to point to next ladder – label FFFF
;
3F7B: 10E5       DJNZ #3F62       ; next B – jump relative to label BBBB
3F7D: C9        RET

```

26 terug = #1A terug = #E6 relative jump

In de aanroep ook de parameter A vullen met 5 (alle ladders tekenen) of 1 (een ladder weglaten):

```

3F18: 3A2762    LD A,(#6227)       ; load A with screen number
3F1B: FE01       CP #01          ; are we on the girders?
3F1D: C22A3F    JP NZ,#3F2A        ; no, skip ahead – jump to label AAAA
;
3F20: 3E05       LD A,#05          ; A = x (5 = all ladders, 1 = remove one ladder)
3F22: 21493B    LD HL,#3B49        ; load HL with stolen ladders of girders
3F25: DD210B63   LD IX,#630B      ; load IX with index for girders
3F29: CD503F    CALL #3F50         ; call additional code to add stolen ladders
;
3F2C: 210960    LD HL, WaitTimerMSB ; load HL with timer addr – label AAAA
3F2F: C3680D    JP #0D68          ; jump back

```

Dit werkt nu goed. Een van de vier ladders wordt random weggelaten.

De code verplaatsen naar achteren (ruimte besparen).

Code verplaatsen:

```
3FD0: FE05      CP #05          ; A = 5 ?
3FD2: C2D93F    JP NZ, #3FD9      ; no, skip next steps – jump to label CCCC
3FD5: 4F        LD C,A           ; C = A = 5
3FD6: C3E03F    JP #3FE0         ; skip next steps – jump to label DDDD
3FD9: 3A1860    LD A,(RngTimer1) ; load A with random timer ? – label CCCC
3FDC: E603      AND #03          ; mask bits: result is random # between 0 and 3
3FDE: 3C        INC A            ; increment A: result is random # between 1 and 4
3FDF: 4F        LD C,A           ; C = A = random # between 1 and 4
;
3FE0: 0604      LD B,#04         ; for B = 1 to 4 – label DDDD
;
3FE2: 78        LD A, B          ; A = B - label BBBB
3FE3: B9        CP C            ; A = C ? : that is: B is equal to C?
3FE4: C2EF3F    JP NZ, #3FEF      ; no, skip next steps – jump to label EEEE
;
3FE7: 23        INC HL           ; next table entry: 5 times to skip drawing this ladder
3FE8: 23        INC HL           ; next table entry
3FE9: 23        INC HL           ; next table entry
3FEA: 23        INC HL           ; next table entry
3FEB: 23        INC HL           ; next table entry
3FEC: C3FA3F    JP #3FFA         ; skip next steps – jump to label FFFF
;
3FEF: E5        PUSH HL          ; save HL - label EEEE
3FF0: C5        PUSH BC          ; save BC
3FF1: EB        EX DE,HL         ; DE <> HL
3FF2: CDA70D    CALL #0DA7        ; draw ladder to screen
3FF5: C1        POP BC           ; restore BC
3FF6: E1        POP HL           ; restore HL
3FF7: CD7524    CALL #2475        ; define ladder and increase IX, HL is at AA at end
;
3FFA: 23        INC HL           ; increment HL to point to next ladder – label FFFF
;
3FFB: 10E5      DJNZ #3F62        ; next B – jump relative to label BBBB
3FFD: C9        RET
```

27 terug = #1B terug = #E5 relative jump

In de aanroep ook de parameter A vullen met 5 (alle ladders tekenen) of 1 (een ladder weggelaten):

```
3F18: 3A2762    LD A,(#6227)     ; load A with screen number
3F1B: FE01      CP #01           ; are we on the girders?
3F1D: C22A3F    JP NZ,#3F2A      ; no, skip ahead – jump to label AAAA
```

```

;
3F20: 3E05      LD  A,#05          ; A = x (5 = all ladders, 1 = remove one ladder)
3F22: 21493B     LD  HL,#3B49       ; load HL with stolen ladders of girders
3F25: DD210B63   LD  IX,#630B      ; load IX with index for girders
3F29: CDD03F     CALL #3FD0         ; call additional code to add stolen ladders
;
3F2C: 210960     LD  HL,WaitTimerMSB ; load HL with timer addr – label AAAA
3F2F: C3680D     JP  #0D68          ; jump back

```

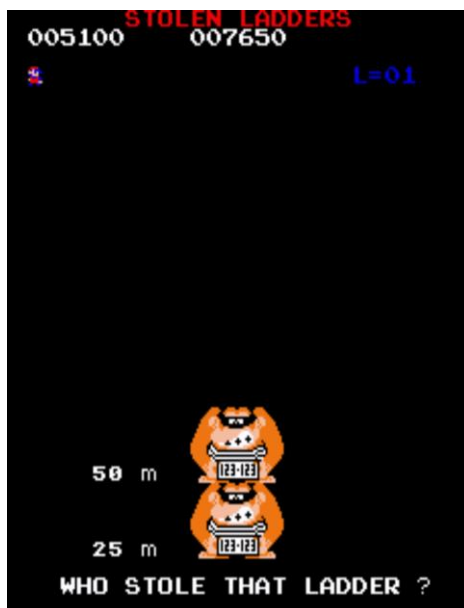
De Goofy Kong graphic veranderen in:



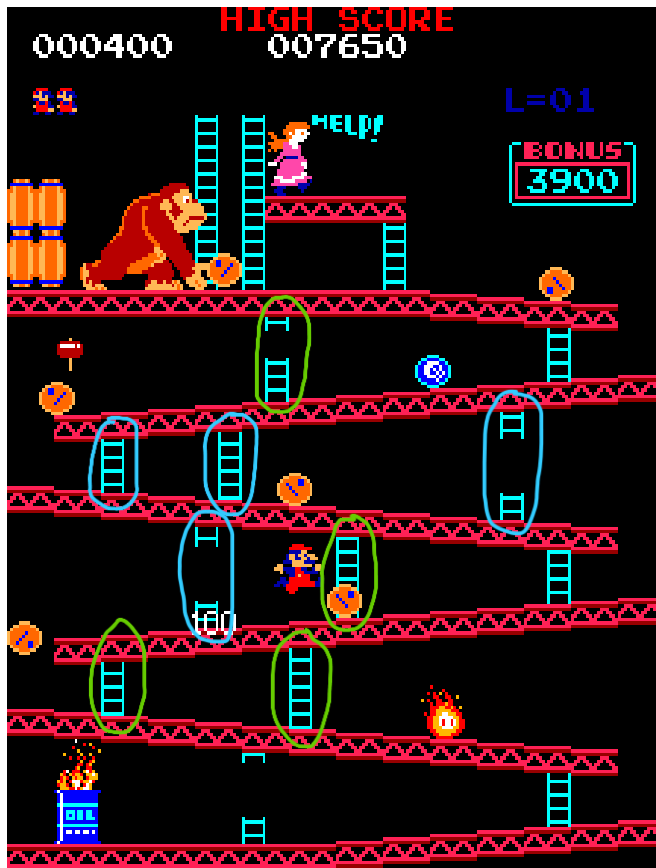
De tekst in het progress screen veranderen in WHO STOLE THAT LADDER ?

27 18 1F 10 23 24 1F 1C 15 10 24 18 11 24 10 1C 11 14 14 15 22 10 FB

Het progress screen ziet er nu zo uit:



De volgende ladder groepering maken voor barrels?



3AE4: 02 97 38 68 38 ; top girder where girl sits
 3AE9: 02 9F 54 10 54 ; girder where kong sits
 3AED: 02 DF 58 A0 55 ; 1st slanted girder at top right
 3AF3: 02 EF 6D 20 79 ; 2nd slanted girder (has hammer at left side)
 3AF8: 02 DF 9A 10 8E ; 3rd slanted girder
 3AFD: 02 EF AF 20 BB ; 4th slanted girder
 3B02: 02 DF DC 10 D0 ; 5th slanted girder (has hammer at right side)
 3B07: 02 FF F0 80 F7 ; bottom slanted girder
 3B0C: 02 7F F8 00 F8 ; bottom flat girder where mario starts
 3B11: 00 CB 57 CB 6F ; short ladder at top right
 3B16: 00 CB 99 CB B1 ; short ladder at center right
 3B1B: 00 CB DB CB F3 ; short ladder at bottom right
 3B20: 00 63 18 63 54 ; kong's ladder (right)
 3B25: 01 63 D5 63 F8 ; bottom broken ladder
 3B2A: ~~00 33 78 33 90~~ ; short ladder at left side under top hammer
 3B2F: ~~00 33 BA 33 D2~~ ; short ladder at left side above oil can

3B34: 00 53 18 53 54 ; kong's ladder (left)
 3B39: ~~01 53 92 53 B8~~ ; second broken ladder from bottom, on 3rd girder
 3B3E: ~~00 5B 76 5B 92~~ ; longer ladder under the top left hammer
 3B43: ~~00 73 B6 73 D6~~ ; longer ladder to left of bottom hammer
 3B48: ~~00 83 95 83 B5~~ ; center longer ladder
 3B4D: 00 93 38 93 54 ; ladder leading to girl
 3B52: ~~01 BB 70 BB 98~~ ; third broken ladder on right side near top
 3B57: ~~01 6B 54 6B 75~~ ; fourth broken ladder near kong
 3B5C: AA ; AA code signals end of data

Definitie van de groene ladders (index met 7 ladders: 6307):

3B35: 00 33 BA 33 D2 AA ; short ladder at left side above oil can
 3B3B: 00 73 B6 73 D6 AA ; longer ladder to left of bottom hammer
 3B41: 00 83 95 83 B5 AA ; center longer ladder
 3B47: 01 6B 54 6B 75 AA ; fourth broken ladder near kong

Definitie van de blauwe ladders (index met 11 ladders: 630B):

3B4D: 00 33 78 33 90 AA ; short ladder at left side under top hammer
 3B53: 01 53 92 53 B8 AA ; second broken ladder from bottom, on 3rd girder
 3B59: 00 5B 76 5B 92 AA ; longer ladder under the top left hammer
 3B5F: 01 BB 70 BB 98 AA ; third broken ladder on right side near top

De aanroep aanpassen en uitbreiden met deze nieuwe groep:

3F18:	3A2762	LD	A,(#6227)	; load A with screen number
3F1B:	FE01	CP	#01	; are we on the girders?
3F1D:	C2383F	JP	NZ,#3F38	; no, skip ahead – jump to label AAAA
;				
3F20:	3E05	LD	A,#05	; A = x (5 = all ladders, 1 = remove one ladder)
3F22:	21353B	LD	HL,#3B35	; load HL with stolen ladders of girders
3F25:	DD210763	LD	IX,#6307	; load IX with index for girders
3F29:	CDD03F	CALL	#3FD0	; call additional code to add stolen ladders
;				
3F2C:	3E05	LD	A,#05	; A = x (5 = all ladders, 1 = remove one ladder)
3F2E:	214D3B	LD	HL,#3B4D	; load HL with stolen ladders of girders
3F31:	DD210B63	LD	IX,#630B	; load IX with index for girders
3F35:	CDD03F	CALL	#3FD0	; call additional code to add stolen ladders
;				
3F38:	210960	LD	HL, WaitTimerMSB	; load HL with timer addr – label AAAA
3F3B:	C3680D	JP	#0D68	; jump back

Werkt goed.

Bij verder testen blijkt dit toch nog niet zo goed te werken. Het zijn steeds dezelfde combinaties van ladders die gekozen worden. Dat komt omdat ze dezelfde randomtimer gebruiken. Daarom zo aanpassen dat het randomgetal (in C) gekozen wordt buiten de functie.

De functie dan aanpassen:

```

3FD0: FE05      CP #05      ; A = 5 ?
3FD2: C2D93F    JP NZ, #3FD9 ; no, skip next steps – jump to label CCCC
3FD5: 4F        LD C,A      ; C = A = 5
3FD6: C3E03F    JP #3FE0    ; skip next steps – jump to label DDDD
3FD9: 3A1860    LD A,(RngTimer1) ; load A with random timer ? – label CCCC
3FDC: E603      AND #03     ; mask bits: result is random # between 0 and 3
3FDE: 3C        INC A       ; increment A: result is random # between 1 and 4
3FDF: 4F        LD C,A      ; C = A = random # between 1 and 4
;
3FE0: 0604      LD B,#04     ; for B = 1 to 4 – label DDDD
;
3FE2: 78        LD A, B      ; A = B - label BBBB
3FE3: B9        CP C        ; A = C ? : that is: B is equal to C?
3FE4: C2EF3F    JP NZ, #3FEF ; no, skip next steps – jump to label EEEE
;
3FE7: 23        INC HL      ; next table entry: 5 times to skip drawing this ladder
3FE8: 23        INC HL      ; next table entry
3FE9: 23        INC HL      ; next table entry
3FEA: 23        INC HL      ; next table entry
3FEB: 23        INC HL      ; next table entry
3FEC: C3FA3F    JP #3FFA     ; skip next steps – jump to label FFFF
;
3FEF: E5        PUSH HL     ; save HL - label EEEE
3FF0: C5        PUSH BC     ; save BC
3FF1: EB        EX DE,HL    ; DE <> HL
3FF2: CDA70D    CALL #0DA7   ; draw ladder to screen
3FF5: C1        POP BC      ; restore BC
3FF6: E1        POP HL      ; restore HL
3FF7: CD7524    CALL #2475   ; define ladder and increase IX, HL is at AA at end
;
3FFA: 23        INC HL      ; increment HL to point to next ladder – label FFFF
;
3FFB: 10E5      DJNZ #3F62   ; next B – jump relative to label BBBB
3FFD: C9        RET

```

In de aanroep moet dan wel de C (de nieuwe parameter) gezet worden en moet de aanroep aangepast worden naar het nieuwe adres (#3FE0):

```

3F18: 3A1860    LD A,(RngTimer1) ; load A with random timer ?
3F1B: E603      AND #03     ; mask bits: result is random # between 0 and 3
3F1D: 3C        INC A       ; increment A: result is random # between 1 and 4
3F1E: 4F        LD C,A      ; C = A = random # between 1 and 4
;
3F1F: 3A2762    LD A,(#6227)  ; load A with screen number
3F22: FE01      CP #01      ; are we on the girders?
3F24: C2423F    JP NZ,#3F42   ; no, skip ahead – jump to label AAAA

```

```

;
3F27: 21353B    LD    HL,#3B35        ; load HL with stolen ladders of girders
3F2A: DD210763 LD    IX,#6307        ; load IX with index for girders
3F2E: CDE03F    CALL #3FE0        ; call additional code to add stolen ladders
;
3F31: 3A1A60    LD    A,(FrameCounter) ; load A with random timer ?
3F34: E603      AND    #03        ; mask bits: result is random # between 0 and 3
3F36: 3C        INC    A            ; increment A: result is random # between 1 and 4
3F37: 4F        LD    C,A            ; C = A = random # between 1 and 4
;
3F38: 214D3B    LD    HL,#3B4D        ; load HL with stolen ladders of girders
3F3B: DD210B63 LD    IX,#630B        ; load IX with index for girders
3F3F: CDE03F    CALL #3FE0        ; call additional code to add stolen ladders
;
3F42: 210960    LD    HL,WaitTimerMSB ; load HL with timer addr – label AAAA
3F45: C3680D    JP    #0D68        ; jump back

```

Gaat nu wel beter, veel meer combinaties mogelijk.

Nu de definitie voor de rivets gaan aanpassen. Drie groepen maken:



```

3C8B: 00-7B-80-7B-A8 ; center ladder level 3
3C90: 00-7B-D0-7B-F8 ; bottom-center ladder
3C95: 00-33-58-33-80 ; top-left ladder
3C9A: 00-53-58-53-80 ; top-left ladder (right-side)

```

3C9F: ~~00 AB 58 AB 80~~ ; top right ladder (left side)
 3CA4: ~~00 CB 58 CB 80~~ ; top right ladder
 3CA9: ~~00 2B 80 2B A8~~ ; level 3 ladder left side
 3CAE: ~~00 D3 80 D3 A8~~ ; level 3 ladder right side
 3CB3: ~~00 23 A8 23 D0~~ ; level 2 ladder left side
 3CB8: ~~00 5B A8 5B D0~~ ; level 2 ladder #2 of 4
 3CBD: ~~00 A3 A8 A3 D0~~ ; level 2 ladder #3 of 4
 3CC2: ~~00 DB A8 DB D0~~ ; level 2 ladder right side
 3B65: 00 1B D0 1B F8 ; bottom left ladder
 3B6A: 00 E3 D0 E3 F8 ; bottom right ladder
 3B6F: 05 B7 30 48 30 ; girder above kong
 3B74: 05 CF 58 30 58 ; girder kong stands on
 3B79: 05 D7 80 28 80 ; level 4 girder
 3B7E: 05 DF A8 20 A8 ; level 3 girder
 3B83: 05 E7 D0 18 D0 ; level 2 girder
 3B88: 05 EF F8 10 F8 ; bottom level girder
 3B8D: AA ; end code

En dat dan op een andere plaats neerzetten (vanaf #3B65). En de verwijzingen naar deze datadefinitie ook aanpassen:

OCC0 328960 LD (#6089),A ; set music
 OCC3 11653B LD DE,#3B65 ; load DE with start of table data for rivets

en

246E 21653B LD HL,#3B65 ; otherwise we're on rivets. load HL with table data for rivets
 2471 DD210063 LD IX,#6300 ; #6300 is used for ladder positions?

Dat is nu goed. Dit levert de volgende laddergroep definities:

Definitie van de groene ladders (index met 2 ladders: 6302):

3B8E: 00 7B 80 7B A8 AA ; center ladder level 3
 3B94: 00 7B D0 7B F8 AA ; bottom center ladder
 3B9A: 00 2B 80 2B A8 AA ; level 3 ladder left side
 3BA0: 00 D3 80 D3 A8 AA ; level 3 ladder right side

Definitie van de blauwe ladders (index met 6 ladders: 6306):

3BA6: 00 23 A8 23 D0 AA ; level 2 ladder left side
 3BAC: 00 5B A8 5B D0 AA ; level 2 ladder #2 of 4
 3BB2: 00 A3 A8 A3 D0 AA ; level 2 ladder #3 of 4
 3BB8: 00 DB A8 DB D0 AA ; level 2 ladder right side

Definitie van de rode ladders (index met 10 ladders: 630A):

3BBE: 00 33 58 33 80 AA ; top left ladder

3BC4: 00 53 58 53 80 AA ; top left ladder (right side)
 3BCA: 00 AB 58 AB 80 AA ; top right ladder (left side)
 3BD0: 00 CB 58 CB 80 AA ; top right ladder

En dan het tekenen van deze groepen ook toevoegen aan het stuk code:

```

3F18: 3A1860    LD  A,(RngTimer1)    ; load A with random timer ?
3F1B: E603      AND #03      ; mask bits: result is random # between 0 and 3
3F1D: 3C        INC A        ; increment A: result is random # between 1 and 4
3F1E: 4F        LD  C,A      ; C = A = random # between 1 and 4
;
3F1F: 3A2762    LD  A,(#6227) ; load A with screen number
3F22: FE01      CP  #01      ; are we on the girders?
3F24: C2313F    JP  NZ,#3F31 ; no, skip ahead – jump to label AAAA
;
3F27: 21353B    LD  HL,#3B35 ; load HL with green group of girders
3F2A: DD210763  LD  IX,#6307 ; load IX with index for green group of girders
3F2E: C3383F    JP  #3F38    ; skip next steps – jump to label BBBB
;
3F31: 218E3B    LD  HL,#3B8E ; load HL with green group of rivets – label AAAA
3F34: DD210263  LD  IX,#6302 ; load IX with index for green group of rivets
;
3F38: CDE03F    CALL #3FE0    ; call additional code to add stolen ladders – label BBBB
;
3F3B: 3A1A60    LD  A,(FrameCounter) ; load A with random timer ?
3F3E: E603      AND #03      ; mask bits: result is random # between 0 and 3
3F40: 3C        INC A        ; increment A: result is random # between 1 and 4
3F41: 4F        LD  C,A      ; C = A = random # between 1 and 4
;
3F42: 3A2762    LD  A,(#6227) ; load A with screen number
3F45: FE01      CP  #01      ; are we on the girders?
3F47: C2543F    JP  NZ,#3F54 ; no, skip ahead – jump to label CCCC
;
3F4A: 214D3B    LD  HL,#3B4D ; load HL with blue group of girders
3F4D: DD210B63  LD  IX,#630B ; load IX with index for blue group of girders
3F51: C35B3F    JP  #3F5B    ; skip next steps – jump to label DDDD
;
3F54: 21A63B    LD  HL,#3BA6 ; load HL with blue group of rivets – label CCCC
3F57: DD210663  LD  IX,#6306 ; load IX with index for blue group of rivets
;
3F5B: CDE03F    CALL #3FE0    ; call additional code to add stolen ladders – label DDDD

```

```

;
3F5E: 3A2762 LD A,(#6227) ; load A with screen number
3F61: FE04 CP #04 ; are we on the rivets?
3F63: C2773F JP NZ,#3F77 ; no, skip ahead – jump to label EEEE
;
3F66: 3A1860 LD A,(RngTimer1) ; load A with random timer ?
3F69: E603 AND #03 ; mask bits: result is random # between 0 and 3
3F6B: 3C INC A ; increment A: result is random # between 1 and 4
3F6C: 4F LD C,A ; C = A = random # between 1 and 4
;
3F6D: 21BE3B LD HL,#3BBE ; load HL with red group of rivets
3F70: DD210A63 LD IX,#630A ; load IX with index for red group of rivets
;
3F74: CDE03F CALL #3FE0 ; call additional code to add stolen ladders
;
3F77: 210960 LD HL,WaitTimerMSB ; load HL with timer addr – label EEEE
3F7A: C3680D JP #0D68 ; jump back

```

Dit werkt nu goed.

Nu de definitie voor de elevators gaan aanpassen. Twee groepen maken:



```

3BD6: 00 63 18 63 58 ; kong's ladder (right)
3BEA: 00 63 88 63 D0 ; center ladder right
3BDB: 00 53 18 53 58 ; long's ladder (left)
3BF4: 00 53 88 53 D0 ; center ladder left
3BE0: 00 E3 68 E3 90 ; far top right ladder leading to purse

```

3BE5: 00 E3 B8 E3 D0 ; far bottom right ladder
 3C03: ~~00 CB 90 CB B0~~ ; ladder leading to purse (lower level)
 3BEA: 00 B3 58 B3 78 ; ladder leading to kong's level
 3C0D: ~~00 9B 80 9B A0~~ ; ladder to right of top right elevator
 3BEE: 00 93 38 93 58 ; ladder leading up to girl
 3BF4: 00 23 88 23 C0 ; long ladder on left side
 3C1C: ~~00 1B C0 1B E8~~ ; bottom left ladder
 3BF9: 02 97 38 68 38 ; girder girl is on
 3BFE: 02 B7 58 10 58 ; kong's girder
 3C03: 02 EF 68 E0 68 ; girder where purse is
 3C08: 02 D7 70 C8 70 ; girder to left of purse
 3C0D: 02 BF 78 B0 78 ; girder holding ladder that leads up to kong's level
 3C12: 02 A7 80 90 80 ; girder to right of top right elevator
 3C17: 02 67 88 48 88 ; top girder for central ladder section between elevators
 3C1C: 02 27 88 10 88 ; girder that holds the umbrella
 3C21: 02 EF 90 C8 90 ; girder under the girder that has the purse
 3C26: 02 A7 A0 98 A0 ; bottom girder for section to right of top right elevator
 3C2B: 02 BF A8 B0 A8 ; small floating girder
 3C30: 02 D7 B0 C8 B0 ; small girder
 3C35: 02 EF B8 E0 B8 ; small girder
 3C3A: 02 27 C0 10 C0 ; girder just above mario start
 3C3F: 02 EF D0 D8 D0 ; small girder on far right bottom
 3C44: 02 67 D0 50 D0 ; bottom girder for central ladder section between elevators
 3C49: 02 CF D8 C0 D8 ; small girder
 3C4E: 02 B7 E0 A8 E0 ; small girder
 3C53: 02 9F E8 88 E8 ; floating girder where the right side elevator gets off
 3C58: 02 27 E8 10 E8 ; girder where mario starts
 3C5D: 02 EF F8 10 F8 ; long bottom girder (mario dies if he gets that low)
 3C62: AA ; end code

En dat dan op een andere plaats neerzetten (vanaf #3BD6). En de verwijzingen naar deze datadefinitie ook aanpassen:

0CFA 11 **D63B** LD DE,#3BD6 ; load DE with start of table data for the elevators
 0CFD C3C60C JP #0CC6 ; jump back

en

2468 21 **D63B** LD HL,#3BD6 ; load HL with start of table data for elevators
 246B CA7124 JP Z,#2471 ; if elevators, skip ahead

Definitie van de groene ladders (index met 7 ladders: 6307):

3C63: 00 63 88 63 D0 AA ; center ladder right
 3C69: 00 53 88 53 D0 AA ; center ladder left
 3C6F: 00 CB 90 CB B0 AA ; ladder leading to purse (lower level)
 3C75: 00 1B C0 1B E8 AA ; bottom left ladder

Definitie van de rode ladder (index met 11 ladders: 630B):

3C7B: 00 9B 80 9B A0 AA ; ladder to right of top right elevator

En dan het tekenen van deze groepen ook toevoegen aan het stuk code (nog 9 bytes vrij: tot #3F85, dus moeten gaan splitsen). Eerst ergens een stuk code vrij spelen. Waar gaan we dat doen? Het liefste het progress screen ongemoeid laten. Dan maar het intro stuk pakken waarin Kong de ladder op klimt en dan gaat springen: dat later vervangen door iets eenvoudigers.

Het stuk waar Kong de ladder op klimt in de intro loopt van #0A76 t/m #0BD9

Dit hele stuk eventjes leeg maken en later weer vullen met een nieuwe animatie:

Dit kan door de volgende code:

```
0A76 210A60 LD HL,GameMode2
0A79 34 INC (HL) ; increase game mode2 (to 8?)
0A7A C9 RET
```

De code wordt dan:

```
3F18: 3A1860 LD A,(RngTimer1) ; load A with random timer ?
3F1B: E603 AND #03 ; mask bits: result is random # between 0 and 3
3F1D: 3C INC A ; increment A: result is random # between 1 and 4
3F1E: 4F LD C,A ; C = A = random # between 1 and 4
;
3F1F: 3A2762 LD A,(#6227) ; load A with screen number
;
3F22: FE04 CP #04 ; are we on the rivets?
3F24: CA3B3F JP Z,#3F3B ; yes, skip ahead – jump to label AAAA
;
3F27: FE03 CP #03 ; are we on the elevators?
3F29: CA453F JP Z,#3F45 ; yes, skip ahead – jump to label BBBB
;
3F2C: FE02 CP #02 ; are we on the conveyors?
3F2E: CA4F3F JP Z,#3F4F ; yes, skip ahead – jump to label CCCC
;
3F31: 21353B LD HL,#3B35 ; load HL with green group of girders
3F34: DD210763 LD IX,#6307 ; load IX with index for green group of girders
3F38: C3563F JP #3F56 ; skip next steps – jump to label DDDD
;
3F3B: 218E3B LD HL,#3B8E ; load HL with green group of rivets – label AAAA
3F3E: DD210263 LD IX,#6302 ; load IX with index for green group of rivets
3F42: C3563F JP #3F56 ; skip next steps – jump to label DDDD
;
3F45: 21633C LD HL,#3C63 ; load HL with green group of elevators – label BBBB
3F48: DD210763 LD IX,#6307 ; load IX with index for green group of elevators
3F4C: C3563F JP #3F56 ; skip next steps – jump to label DDDD
;
```

3F4F: 21C83F	LD	HL,#3FC8	; load HL with green group of conveyors – label CCCC
3F52: DD210763	LD	IX,#6307	; load IX with index for green group of conveyors
;			
3F56: CDE03F	CALL	#3FE0	; call additional code to add stolen ladders – label DDDD
;			
3F59: 3A2962	LD	A,(#6229)	; load A with level number
3F5C: FE02	CP	#02	; A >= 2 ?
3F5E: D2673F	JP	NC,#3F67	; yes, skip next steps – jump to label JJJJ
;			
3F61: 3E05	LD	A,#05	; A=5
3F63: 4F	LD	C,A	; C=A (=5) – don't remove a ladder from the blue group
3F64: C36E3F	JP	#3F6E	; skip next steps – jump to label KKKK
;			
3F67: 3A1A60	LD	A,(FrameCounter)	; load A with random timer ? – label JJJJ
3F6A: E603	AND	#03	; mask bits: result is random # between 0 and 3
3F6C: 3C	INC	A	; increment A: result is random # between 1 and 4
3F6D: 4F	LD	C,A	; C = A = random # between 1 and 4
;			
3F6E: 3A2762	LD	A,(#6227)	; load A with screen number – label KKKK
;			
3F71: FE04	CP	#04	; are we on the rivets?
3F73: CA6A0B	JP	Z,#0B6A	; yes, skip ahead – jump to label EEEE
;			
3F76: FE03	CP	#03	; are we on the elevators?
3F78: CA7E0B	JP	Z,#0B7E	; yes, skip ahead – jump to label HHHH (no blue group)
;			
3F7B: FE02	CP	#02	; are we on the conveyors?
3F7D: CA740B	JP	Z,#0B74	; yes, skip ahead – jump to label FFFF
;			
3F80: C3600B	JP	#0B60	; jump to additional code – jump to label YYYY

En additionele code in nieuw stuk:

0B60: 214D3B	LD	HL,#3B4D	; load HL with blue group of girders – label YYYY
0B63: DD210B63	LD	IX,#630B	; load IX with index for blue group of girders
0B67: C37B0B	JP	#0B7B	; skip next steps – jump to label GGGG
;			
0B6A: 21A63B	LD	HL,#3BA6	; load HL with blue group of rivets – label EEEE
0B6D: DD210663	LD	IX,#6306	; load IX with index for blue group of rivets
0B71: C37B0B	JP	#0B7B	; skip next steps – jump to label GGGG
;			
0B74: 21480B	LD	HL,#0B48	; load HL with blue group of conveyors – label FFFF
0B77: DD210B63	LD	IX,#630B	; load IX with index for blue group of conveyors
;			
0B7B: CDE03F	CALL	#3FE0	; call additional code to add stolen ladders – label GGGG
;			
0B7E: 3A2762	LD	A,(#6227)	; load A with screen number – label HHHH
;			

OB81: FE01	CP	#01	; are we on the girders?
OB83: CACA0B	JP	Z,#0BCA	; yes, skip ahead – jump to label XXXX
;			
OB86: FE02	CP	#02	; are we on the conveyors?
OB88: CACA0B	JP	Z,#0BCA	; yes, skip ahead – jump to label XXXX
;			
OB8B: FE03	CP	#03	; are we on the elevators?
OB8D: CAB20B	JP	Z,#0BB2	; yes, skip ahead – jump to label MMMM
;			
OB90: 3A2962	LD	A,(#6229)	; load A with level number
OB93: FE03	CP	#03	; A >= 3 ?
OB95: D29E0B	JP	NC,#0B9E	; yes, skip next steps – jump to label LLLL
;			
OB98: 3E05	LD	A,#05	; A=5
OB9A: 4F	LD	C,A	; C=A (=5) – don't remove a ladder from the red group
OB9B: C3A50B	JP	#0BA5	; skip next steps – jump to label KKKK
;			
OB9E: 3A1860	LD	A,(RngTimer1)	; load A with random timer ? – label LLLL
OBA1: E603	AND	#03	; mask bits: result is random # between 0 and 3
OBA3: 3C	INC	A	; increment A: result is random # between 1 and 4
OBA4: 4F	LD	C,A	; C = A = random # between 1 and 4
;			
OBA5: 21BE3B	LD	HL,#3BBE	; load HL with red group of rivets – label KKKK
OBA8: DD210A63	LD	IX,#630A	; load IX with index for red group of rivets
;			
OBAC: CDE03F	CALL	#3FE0	; call additional code to add stolen ladders
OBAF: C3CA0B	JP	#0BCA	; skip next steps, jump to label XXXX
;			
OB82: 3A2962	LD	A,(#6229)	; load A with level number – label MMMM
OB85: FE03	CP	#03	; A >= 3 ?
OB87: D2CA0B	JP	NC,#0BCA	; yes, skip next steps – jump to label XXXX
;			
OB8A: 117B3C	LD	DE, #3C7B	; load DE with red group of elevators
OB8D: CDA70D	CALL	#0DA7	; draw ladder to screen
OBC0: 217B3C	LD	HL,#3C7B	; load HL with red group of elevators
OBC3: DD210B63	LD	IX,#630B	; load IX with index for red group of rivets
OBC7: CD7524	CALL	#2475	; define ladder and increase IX, HL is at AA at end
;			
OBCA: 210960	LD	HL, WaitTimerMSB	; load HL with timer addr – label XXXX
OBCD: C3680D	JP	#0D68	; jump back

Level 1 is alleen groene groep meegenomen en Level 2 is groene groep en blauwe groep en level 3 is groene, blauwe en rode groep.

Nu de definitie voor de conveyors gaan aanpassen. Twee groepen maken:



Definitie starten vanaf: #3C81

3C81: 06 8F 90 70 90 ; central patch of XXX's
3C86: 06 8F 98 70 98 ; central patch of XXX's
3C8B: 06 8F A0 70 A0 ; central patch of XXX's
3C90: 00 63 18 63 58 ; kong's ladder (right)
 3B71: ~~00 63 80 63 A8~~ ; center ladder to left of oil can fire
 3B76: ~~00 63 D0 63 F8~~ ; bottom level ladder #2 of 4
3C95: 00 53 18 53 58 ; kong's ladder (left)
 3B80: ~~00 53 A8 53 D0~~ ; ladder under the hat
3C9A: 00 9B 80 9B A8 ; center ladder to right of oil can fire
 3B8A: ~~00 9B D0 9B F8~~ ; bottom level ladder #3 of 4
3C9F: 01 23 58 23 80 ; top broken ladder left side
3CA4: 01 DB 58 DB 80 ; top broken ladder right side
3CA9: 00 2B 80 2B A8 ; ladder on left platform with hammer
 3B9E: ~~00 D3 80 D3 A8~~ ; ladder on right platform with umbrella
 3BA3: ~~00 A3 A8 A3 D0~~ ; ladder to right of bottom hammer
 3BA8: ~~00 2B D0 2B F8~~ ; bottom level ladder #1 of 4
 3BAD: ~~00 D3 D0 D3 F8~~ ; bottom level ladder #4 of 4
3CAE: 00 93 38 93 58 ; ladder leading to girl
3CB3: 02 97 38 68 38 ; girder where girl sits
3CB8: 03 EF 58 10 58 ; top conveyor girder
3CBD: 03 F7 80 88 80 ; top right conveyor next to oil can
3CC3: 03 77 80 08 80 ; top left conveyor next to oil can
3CC8: 02 A7 A8 50 A8 ; center ledge
3CCD: 02 E7 A8 B8 A8 ; right center ledge
3CD2: 02 3F A8 18 A8 ; left center ledge (has hammer)

3CD6: 03 EF D0 10 D0 ; main lower conveyor girder (has hammer)

3CDC: 02 EF F8 10 F8 ; bottom level girder

3CE1: AA ; end code

En dat dan op een andere plaats neerzetten (vanaf #3C81). En de verwijzingen naar deze datadefinitie ook aanpassen:

0CDF 11813C LD DE,#3C81 ; load DE with start of table data for conveyors

0CE2 21867D LD HL,REG_PALETTE_A ; load HL with palette bank selector

en

2461 21813C ID HL,#3C81 ; load HL with start of table data for conveyors

2464 CA7124 JP Z,#2471 ; if conveyors, skip ahead

Definitie van de groene ladders (index met 7 ladders: 6307):

3FC8: 00 53 A8 53 D0 AA ; ladder under the hat

3FCE: 00 9B D0 9B F8 AA ; bottom level ladder #3 of 4

3FD4: 00 D3 80 D3 A8 AA ; ladder on right plantform with umbrella

3FDA: 00 2B D0 2B F8 AA ; bottom level ladder #1 of 4

Definitie van de blauwe ladders (index met 11 ladders: 630B):

0B48: 00 63 80 63 A8 AA ; center ladder to left of oil can fire

0B4E: 00 63 D0 63 F8 AA ; bottom level ladder #2 of 4

0B54: 00 A3 A8 A3 D0 AA ; ladder to right of bottom hammer

0B5A: 00 D3 D0 D3 F8 AA ; bottom level ladder #4 of 4

Dit werkt nu ook goed.

Aanpassen van de level definities:

Groene ladder gestolen (Barrels/Pies/Springs/Rivets)	Altijd
Blauwe ladder gestolen (Barrels/Pies/Rivets)	Vanaf instelbaar level: 3F59: 3A2962 LD A,(#6229) ; load A with level number 3F5C: FE02 CP #02 ; A >= 2 ?
Rode ladder gestolen (Rivets = bovenste 4 ladders)	Vanaf instelbaar level: 0B90: 3A2962 LD A,(#6229) ; load A with level number 0B93: FE03 CP #03 ; A >= 3 ?
Rode ladder gestolen (Springs = ladder naast elevator)	Vanaf instelbaar level: 0BB2: 3A2962 LD A,(#6229) ; load A with level number 0BB5: FE03 CP #03 ; A >= 3 ?

Probleem is nog dat de ladders over de girders getekend worden, maar dat is altijd met een girder van het barrel stage. Bij conveyors en rivets worden andere girder graphics gebruikt en dat ziet er niet mooi uit.


```

0B1F: 3A2762    LD    A,(#6227)    ; load A with screen number – label ZZZZ
;
0B22: FE01      CP    #01      ; are we on the girders?
0B24: CA430B    JP    Z,#0B43     ; yes, skip ahead – jump to label XXXX

```

```

;
OB27: FE03      CP   #03          ; are we on the elevators?
OB29: CA430B    JP   Z,#0B43      ; yes, skip ahead – jump to label XXXX
;
OB2C: FE02      CP   #02          ; are we on the conveyors?
OB2E: CA3D0B    JP   Z,#0B3D      ; yes, skip ahead – jump to label MMMM
;
OB31: 116F3B    LD   DE, #3B6F    ; load DE with repair girders for rivets
OB34: CDA70D    CALL #0DA7        ; redraw girders to screen
OB37: CD000D    CALL Z,#0D00      ; call sub to redraw the rivets
OB3A: C3430B    JP   #0B43        ; skip next steps – jump to label XXXX
;
OB3D: 11CC3C    LD   DE, #3CCC    ; load DE with repair girders for conveyors – label MMMM
OB40: CDA70D    CALL #0DA7        ; redraw girders to screen
;
OB43: 210960    LD   HL, WaitTimerMSB ; load HL with timer addr – label XXXX
OB46: C3680D    JP   #0D68        ; jump back

```

En dan de definities zodanig husselen dat de opnieuw te tekenen girders aan het einde staan: dan kun je daar naartoe springen:

Rivets:

```

3C81: 06 8F 90 70 90 ; central patch of XXX's
3C86: 06 8F 98 70 98 ; central patch of XXX's
3C8B: 06 8F A0 70 A0 ; central patch of XXX's
3C90: 00 63 18 63 58 ; kong's ladder (right)
3B71: 00 63 80 63 A8 ; center ladder to left of oil can fire
3B76: 00 63 D0 63 F8 ; bottom level ladder #2 of 4
3C95: 00 53 18 53 58 ; kong's ladder (left)
3B80: 00 53 A8 53 D0 ; ladder under the hat
3C9A: 00 9B 80 9B A8 ; center ladder to right of oil can fire
3B8A: 00 9B D0 9B F8 ; bottom level ladder #3 of 4
3C9F: 01 23 58 23 80 ; top broken ladder left side
3CA4: 01 DB 58 DB 80 ; top broken ladder right side
3CA9: 00 2B 80 2B A8 ; ladder on left platform with hammer
3B9E: 00 D3 80 D3 A8 ; ladder on right platform with umbrella
3BA3: 00 A3 A8 A3 D0 ; ladder to right of bottom hammer
3BA8: 00 2B D0 2B F8 ; bottom level ladder #1 of 4
3BAD: 00 D3 D0 D3 F8 ; bottom level ladder #4 of 4
3CAE: 00 93 38 93 58 ; ladder leading to girl
3CB3: 02 97 38 68 38 ; girder where girl sits
3CB8: 03 EF 58 10 58 ; top conveyor girder – ruilen X
3CBD: 03 F7 80 88 80 ; top right conveyor next to oil can – ruilen Y
3CC2: 03 77 80 08 80 ; top left conveyor next to oil can – ruilen Z
3CC7: 02 A7 A8 50 A8 ; center ledge
3CCC: 02 E7 A8 B8 A8 ; right center ledge – ruilen Z – start redraw girders here
3CD1: 02 3F A8 18 A8 ; left center ledge (has hammer) – ruilen Y

```

3CD6: 03 EF D0 10 D0 ; main lower conveyor girder (has hammer)
3CDB: 02 EF F8 10 F8 ; bottom level girder – ruilen X
3CE0: AA ; end code

Rivets:

3B65: 00 1B D0 1B F8 ; bottom left ladder
3B6A: 00 E3 D0 E3 F8 ; bottom right ladder
3B6F: 05 B7 30 48 30 ; girder above kong – start redraw girders here
3B74: 05 CF 58 30 58 ; girder kong stands on
3B79: 05 D7 80 28 80 ; level 4 girder
3B7E: 05 DF A8 20 A8 ; level 3 girder
3B83: 05 E7 D0 18 D0 ; level 2 girder
3B88: 05 EF F8 10 F8 ; bottom level girder
3B8D: AA ; end code

Werkt nu goed. De girders in rivets en conveyors worden goed getekend.

Huh. What happened here? Is de keuze groen /blauw inderdaad zodanig dat dit kan optreden?



Ja dus. Lekker dom !



De twee ladders op het tweede niveau moeten van dezelfde kleur zijn!



Dus de definities moeten aangepast:

Definitie van de groene ladders (index met 7 ladders: 6307):

3FC8: 00 63 80 63 A8 AA ; center ladder to left of oil can fire

3FCE: 00 9B D0 9B F8 AA ; bottom level ladder #3 of 4

3FD4: 00 D3 80 D3 A8 AA ; ladder on right platform with umbrella

3FDA: 00 2B D0 2B F8 AA ; bottom level ladder #1 of 4

Definitie van de blauwe ladders (index met 11 ladders: 630B):

0B48: 00 53 A8 53 D0 AA ; ladder under the hat

0B4E: 00 63 D0 63 F8 AA ; bottom level ladder #2 of 4

0B54: 00 A3 A8 A3 D0 AA ; ladder to right of bottom hammer

0B5A: 00 D3 D0 D3 F8 AA ; bottom level ladder #4 of 4

Aangepast. En getest. Beide ladders zijn te beklimmen. Dus de definities zijn goed.

Nu nog een nieuw intro screen maken. Er zijn in ieder geval ook nog een aantal definities van animaties die je kunt gebruiken. Eerst uitzoeken welke te overschrijven zijn.

Intro screen datablokken die vrijgemaakt kunnen worden:

#380D - #3849

#38B4 - #38CA

#38CB - #38DB

#38DC - #38F6

#392C - #3931

Dus dat is al met al niet heel veel.

Intro screen maken:

```
0A76 3A8563 LD A,(#6385) ; varies from 0 to X
0A79 EF RST #28 ; jump based on A
;
0A7A 8C0A 0 ; #0A8C ; draw barrels screen – jump to label AAAA
0A7C DA0A 1 ; #0ADA ; draw sprinkle – jump to label BBBB
0A7E F40A 2 ; #0AF4 ; erase sprinkle – jump to label CCCC
0A80 DA0A 3 ; #0ADA ; draw sprinkle – jump to label BBBB
0A82 F40A 4 ; #0AF4 ; erase sprinkle – jump to label CCCC
0A84 DA0A 5 ; #0ADA ; draw sprinkle – jump to label BBBB
0A86 F40A 6 ; #0AF4 ; erase sprinkle – jump to label CCCC
0A88 FA0A 7 ; #0AFA ; display text – jump to label DDDD
0A8A D10A 8 ; #0AD1 ; wait a number of time – jump to label EEEE
;
; arrive from #0A79 when intro screen indicator == 0
;
0A88: AF XOR A ; A := 0 – label AAAA
0A89: 32867D LD (REG_PALETTE_A),A ; clear palette bank selector
0A8C: 3C INC A ; A := 1
0A8D: 32877D LD (REG_PALETTE_B),A ; store into palette bank selector
0A90: 11E43A LD DE,#3AE4 ; load DE with start of table data
0A93: CDA70D CALL #0DA7 ; draw the screen
;
```

```

0A96: 3E05      LD A,#05          ; A := 5
0A98: 4F        LD C,A          ; C = A = 5 – draw all ladders
0A99: 21353B      LD HL,#3B35      ; load HL with green group of girders
0A9C: DD210763    LD IX,#6307  ; load IX with index for green group of girders
0AA0: CDE03F      CALL #3FE0    ; call additional code to add stolen ladders
;
0AA3: 3E02      LD A,#02          ; A := 2
0AA5: 4F        LD C,A          ; C = A = 2 – remove 2nd ladder
0AA6: 214D3B      LD HL,#3B4D  ; load HL with blue group of girders
0AA9: DD210B63    LD IX,#630B  ; load IX with index for blue group of girders
0AAD: CDE03F      CALL #3FE0    ; call additional code to add stolen ladders
;
0AB0 3E10      LD A,#10          ; A := #10
0AB2 32A376     LD (#76A3),A      ; erase a graphic near top of screen
0AB5 326376     LD (#7663),A      ; erase a graphic near top of screen
;
0AB8 CDB438     CALL #38B4        ; 3 NOP's om call voor tekenen Goofy Kong in te voegen
;
0ABB: 218A60     LD HL,#608A      ; load HL with music buffer
0ABE: 3601      LD (HL),#01      ; play scary music for start of game sound
0AC0: 23        INC HL          ; load HL with duration
0AC1: 3603      LD (HL),#03      ; set duration to 3
;
0AC3: 3E40      LD A,#40          ; A := #40
0AC5: 320960     LD (WaitTimerMSB),A ; set timer to #FF
0AC8: 218563     LD HL,#6385      ; load HL with intro screen counter
0ACB: 34        INC (HL)         ; increase
0ACC: C9       RET             ; return
;
; arrive from #0A79 when intro screen indicator == 2
;
0ACD: DF       RST #18          ; count down timer – label CCCC
0ACE: AF       XOR A           ; A := 0
0ACF: 328563     LD (#6385),A    ; reset intro screen counter to zero
0AD2: 34        INC (HL)         ; increase timer in WaitTimerMSB
0AD3: 23        INC HL          ; HL := GameMode2
0AD4: 34        INC (HL)         ; increase game mode2 (to 8?)
0AD5: C9       RET             ; return
;
; arrive from #0A79 when intro screen indicator == 1 or 3
;
0AD6: DF       RST #18          ; count down timer – label BBBB
0AD7: 3E63      LD A, #63        ; load A with sprite number of sparkle sprite
0AD9: 21206A     LD HL,#6A20      ; load HL with heart sprite – label CCCC
0ADC: 365B      LD (HL),#5B      ; set heart sprite x-position
0ADE: 23        INC HL          ; next
0ADF: 77        LD (HL),A        ; set heart sprite with correct sprite
0AD0: 23        INC HL          ; next
0AE1: 3609      LD (HL),#0C      ; set heart sprite color
0AE2: 23        INC HL          ; next

```

```

0AE4: 3688    LD HL,#88          ; set heart sprite y-position
;
0AE6: 3E80    LD  A,#80           ; A := #80
0AE8: 320960  LD  (WaitTimerMSB),A    ; set timer to #FF
0AEB: 218563  LD  HL,#6385          ; load HL with intro screen counter
0AEE: 34      INC  (HL)          ; increase
0AEF: C9      RET              ; return
;
; arrive from #0A79 when intro screen indicator == 2 or 4
;
0AF0: DF      RST  #18          ; count down timer – label BBBB
0AF1: 3E64    LD  A, #64        ; load A with sprite number of blank sprite
0AF3: C3D90A  JP  #0AD9          ; jump to code to set/reset sprite – jump to label CCCC
;
; arrive from #0A79 when intro screen indicator == 5
;
0AF6: 3EFF    LD  A, #FF
0AF8: 320960  LD  (WaitTimerMSB),A ; set timer to #FF
0AFB: 218563  LD  HL,#6385          ; load HL with intro screen counter
0AFE: 34      INC  (HL)          ; increase
0AFF: C9      RET              ; return

```

Goofy Kong op het scherm plaatsen:

```

38B4 21BC75  LD  HL,#75BC      ; load HL with screen location start for goofy kong
;
38B7 0E50    LD  C,#50        ; C := #50 = start graphic for goofy kong
;
38B9 71      LD  (HL),C        ; draw part of goofy kong
38BA 0C      INC  C            ; next graphic
38BB 2B      DEC  HL           ; next screen location
38BC 71      LD  (HL),C        ; draw part of goofy kong
38BD 0C      INC  C            ; next graphic
38BE 2B      DEC  HL           ; next screen location
38BF 71      LD  (HL),C        ; draw part of goofy kong
38C0 0C      INC  C            ; next graphic
38C1 2B      DEC  HL           ; next screen location
38C2 71      LD  (HL),C        ; draw part of goofy kong
38C3 79      LD  A,C           ; load A with graphic number
38C4 FE67    CP  #67           ; == #67 ? (are we done?)
38C6 CAD138  JP  Z,#38D1        ; yes, skip next 4 steps
;
38C9 0C      INC  C            ; next C
38CA 112300  LD  DE,#0023      ; load DE with offset
38CD 19      ADD  HL,DE        ; add to screen location
38CE C3B938  JP  #38B9           ; loop again
38D1 C9      RET

```

Deze wordt wel in de verkeerde kleuren getekend. Dat ziet er niet uit. Dus niet doen. Terug naar de versie zonder de Goofy Kong getekend.

Maar deze werkt wel:

```
38B4 21CF39 LD    HL,#39CF    ; load HL with table data for kong beating chest
38B7 CD4E00 CALL  #004E        ; update kong's sprites
38BA 210869 LD    HL,#6908    ; load HL with start of kong sprite X pos

38BD 0E44    LD    C,#44      ; load C with offset to add X
38BF FF      RST  #38        ; draw kong in new position
38C0 210B69 LD    HL,#690B    ; load HL with start of kong sprite Y pos
38C3 0E78    LD    C,#78      ; load C with offset to add Y
38C5 FF      RST  #38        ; draw kong
38C6 C9      RET
```

Playable demo maken:

Starten vanaf L=05.

```
095E 01 65 3A 01 00 00 00    ; #3A65 is start of table data for screens/levels
```

Waarin de eerste 01 gekopieerd wordt naar #6229 (is level #)

En de 653A gekopieerd wordt naar #622A (is pointer in level/screen table)

Als starten L=05 dan dit aanpassen naar 05 en 73 3A.

De moeilijkheidsgraad zo configureren:

Groene ladder gestolen (Barrels/Pies/Springs/Rivets)	Altijd
Blauwe ladder gestolen (Barrels/Pies/Rivets)	Vanaf instelbaar level: 3F59: 3A2962 LD A,(#6229) ; load A with level number 3F5C: FE06 CP #06 ; A >= 6 ?
Rode ladder gestolen (Rivets = bovenste 4 ladders)	Vanaf instelbaar level: 0B90: 3A2962 LD A,(#6229) ; load A with level number 0B93: FE07 CP #07 ; A >= 7 ?
Rode ladder gestolen (Springs = ladder naast elevator)	Vanaf instelbaar level: 0BB2: 3A2962 LD A,(#6229) ; load A with level number 0BB5: FE07 CP #07 ; A >= 7 ?

Er zit nog steeds een fout in de conveyors. Er zijn er bij het begin al twee weg en later wanneer er twee weg moeten zijn, zijn er some drie weg (en zelfs alle ladders op het tweede niveau).

Probleem is dat er een byte overlappend is:

Stuk code:

```
0B40: CDA70D    CALL #0DA7          ; redraw girders to screen
;
0B43: 210960    LD  HL, WaitTimerMSB ; load HL with timer addr – label XXXX
0B46: C3680D    JP  #0D68          ; jump back
```

En conveyors definitie:

```
0B48: 00 53 A8 53 D0 AA ; ladder under the hat
0B4E: 00 63 D0 63 F8 AA ; bottom level ladder #2 of 4
0B54: 00 A3 A8 A3 D0 AA ; ladder to right of bottom hammer
0B5A: 00 D3 D0 D3 F8 AA ; bottom level ladder #4 of 4
```

De conveyors definitie kan niet opgeschoven worden.

Daarom bug oplossen door springen naar additionele code:

```
0B43: C3040B    JP #0B04          ; jump to additional code – label XXXX

0B04: 210960    LD  HL, WaitTimerMSB ; load HL with timer addr
0B46: C3680D    JP  #0D68          ; jump back
```

Pies dicht op elkaar, kan wellicht door aanpassen:

```
2576 DD360001 LD  (IX+#00),#01 ; set pie active
257A DD36074B LD  (IX+#07),#4B  ; set pie sprite value
257E DD360908 LD  (IX+#09),#08  ; set pie size??? (width?)
2582 DD360A03 LD  (IX+#0A),#03  ; set pie size??? (height?)
2586 3E7C     LD  A,#7C       ; A := #7C
2588 329B63   LD  (#639B),A    ; store into pie timer for next pie deployment
```

De Pie timer kleiner maken: dan sneller weer een pie deployen. Testen met kleinere waarde dan #7C. #24 geeft een interessante pie verdeling. Soms 2 en soms 3 vlak achter elkaar.

Af en toe een afwijkende bakje laten komen afhankelijk van random waarde. Het zetten van de pie sprites gebeurt hier:

```
2576 DD360001 LD  (IX+#00),#01 ; set pie active
257A DD36074B LD  (IX+#07),#4B  ; set pie sprite value
257E DD360908 LD  (IX+#09),#08  ; set pie size??? (width?)
2582 DD360A03 LD  (IX+#0A),#03  ; set pie size??? (height?)
```

Als de pie-value veranderd wordt naar #4C dan komt het alternatieve bakje.

Dan het volgende doen:

```

2576: DD360001 LD    (IX+#00),#01 ; set pie active
257A: DD360908 LD    (IX+#09),#08 ; set pie size??? (width?)
257E: 3A1860 LD    A,(RngTimer1) ; load A with random timer ?
2581: E603 AND #03 ; mask bits: result is random # between 0 and 3
2583: A7 AND A ; check if A equals zero
2584: CAC738 JP Z, #38C7 ; if zero jump to additional code 1 – jump to label AAAA
2587: C3D738 JP #38D7 ; else jump to additonal code 2 – jump to label BBBB
258A: NOP

```

en

```

38C7: DD36074C LD    (IX+#07),#4C ; set pie sprite value – pie version 1 – label AAAA
38CB: DD360A01 LD    (IX+#0A),#01 ; set pie size??? (height?)
38CF: 3E24 LD    A,#24 ; A := #24
38D1: 329B63 LD    (#639B),A ; store into pie timer for next pie deployment
38D4: C38B25 JP    #258B ; jump back

```

en

```

38D7: DD36074B LD    (IX+#07),#4B ; set pie sprite value – pie version 1 – label BBBB
38DB: DD360A03 LD    (IX+#0A),#03 ; set pie size??? (height?)
38DF: 3E7C LD    A,#7C ; A := #7C
38E1: 329B63 LD    (#639B),A ; store into pie timer for next pie deployment
38E4: C38B25 JP    #258B ; jump back

```

Hammers aan/uit en alleen te pakken wanneer aan staan?

Eerst testen hoe een hamer uitgezet kan worden: stukje code toevoegen als call vanuit #19C2: een ongebruikt 'slot' uit de main routine: Daar dan een call naar een stukje code dat de eerste hamer sprite vervangt door een andere sprite.

Dat kan door de sprite value te veranderen van de eerste hamer: #6A19:

```

38E7: 3E07 LD    A, #07 ; load A with 07
38E9: 21196A LD    HL,#6A19 ; load HL with sprite value for first hammer
38EC: 77 LD    (HL),A ; set sprite value for first hammer
38ED: C9 RET ; return

```

En dit aanroepen vanaf #19C2:

```

19C2: CDE738 ; do blinking of hammers

```

Dat werkt. De eerste hamer verdwijnt inderdaad. Maar daar is noiet voldoende ruimte voor een meer intelligente manier van hammers tonen/verwijderen. Dus ergens ruimte zien vrij te maken.

Eventueel het intro screen weer vrijmaken:

Het intro screen kan uitgezet worden door:

```
0A76 210A60 LD HL,GameMode2
0A79 34 INC (HL) ; increase game mode2 (to 8?)
0A7A C9 RET
```

De vrije ruimte gebruiken voor de hamer blink:

```
0A7B: 3E07 LD A, #07 ; load A with 07
0A7D: 21196A LD HL,#6A19 ; load HL with sprite value for first hammer
0A80: 77 LD (HL),A ; set sprite value for first hammer
0A81: C9 RET ; return
```

En dit aanroepen vanaf #19C2:

```
19C2: CD7B0A ; do blinking of hammers
```

Dan nu een stuk code die de hamer onconditioneel laat knipperen op basis van de FrameCounter.

```
0A7B: 3A1A60 LD A,(FrameCounter) ; load A with timer
0A7E: E616 AND #16 ; test the 4th bit
0A80: CA880A JP Z, #0A88 ; if not set skip next steps – jump to label AAAA
;
0A83: 3E1E LD A, #1E ; set A with value for hammer sprite
0A85: C38A0A JP #0A8A ; skip next steps – jump to label BBBB
;
0A88: 3E07 LD A, #07 ; set A with value for blank sprite – label AAAA
;
0A8A: 21196A LD HL,#6A19 ; load HL with sprite value for first hammer – label BBBB
0A8D: 77 LD (HL),A ; set sprite value for first hammer
0A8E: 211D6A LD HL,#6A1D ; load HL with sprite value for second hammer
0A91: 77 LD (HL),A ; set sprite value for second hammer
0A92: C9 RET ; return
```

Werkt niet zo heel erg goed. Lijkt ook te vertragen, maar dat kan ook gevoel zijn. Wellicht anders doen op basis van eigen counter. Hiertoe adres #6385 dat in het intro screen gebruikt wordt.

```
0A7B: 3A2762 LD A,(#6227) ; load A with screen number
0A7E: FE03 CP #03 ; are we on the elevators?
0A80: C8 RET Z ; yes, return
;
0A81: 3A8563 LD A,(#6385) ; load A with blink counter
0A84: 3C INC A ; increment blink counter
0A85: 328563 LD (#6385), A ; store blink counter
;
0A88: FE50 CP #50 ; is blink counter #50?
0A8A: CA9B0A JP Z, #0A9B ; yes: make hammer invisible – jump to label AAAA
;
```

```

0A8D: FEA0      CP #A0          ; is blink counter #A0?
0A8F: CAA40A    JP Z, #0AA4      ; yes: make hammer visible – jump to label BBBB
;
0A92: FEC0      CP #C0          ; is blink counter greater than #C0?
0A94: D8        RET C           ; no return
0A95: 3E00      LD A, #00        ; load A with #00
0A97: 328563    LD (#6385), A    ; store blink counter
0A9A: C9        RET             ; return
;
0A9B: 3E07      LD A, #07        ; load A with value empty sprite – label AAAA
0A9D: 32196A    LD (#6A19), A    ; make first hammer invisible
0AA0: 321D6A    LD (#6A1D), A    ; make second hammer invisible
0AA3: C9        RET             ; return
;
0AA4: 3E1E      LD A, #1E        ; load A with hammer sprite – label BBBB
0AA6: 32196A    LD (#6A19), A    ; make first hammer visible
0AA9: 321D6A    LD (#6A1D), A    ; make second hammer visible
0AAC: C9        RET             ; return

```

Dit gaat nu goed. Als hamer gepakt dan knippert deze niet. En wanneer hamer gebruikt, dan wordt deze ook niet meer getoond.

Nu nog zo maken dat de hamers alleen gepakt kunnen worden wanneer de hamer zichtbaar is. Daartoe check inbouwen bij het grabben dat er gecheckt wordt of de waarde in 6A19 (eerste hamer) en 6A1D (tweede hamer) op #1E staat (zichtbare hamer). Maar welke je checkt maakt niet uit. Ze worden toch steeds beiden gezet. Dus pak gewoon de #6A19.

Hammer grab check wordt hier gedaan:

; check for hammer grab ?

```

2974 FD210062 LD IY,#6200 ; load IY with start of mario sprite values
2978 3A0562 LD A,(#6205) ; load A with mario's Y position
297B 4F LD C,A ; copy to C
297C 210804 LD HL,#0408 ; H := 4, L := 8
297F 0602 LD B,#02 ; B := 2 for the 2 hammers (?)
2981 111000 LD DE,#0010 ; offset for each hammer
2984 DD218066 LD IX,#6680 ; load IX with start of hammer sprites ?
2988 CD1329 CALL #2913 ; check for collision with hammer
298B C9 RET ; return

```

Deze hele check hoeft niet als de waarde van #6A19 gelijk is aan #07, dan meteen een return. Dit aanpassen door extra check in additionele code te doen:

```

2974: C3B00A JP #0AB0 ; jump to additonal code – jump to label AAAA
2977: 00      NOP
en

```

```

0AB0: 3A196A    LD A,(#6A19)    ; load A with sprite value first hammer – label AAAA
0AB3: FE07      CP #07      ; check if sprite value is #07: invisible hammer
0AB5: C8        RET Z      ; return if invisible: no grab
0AB6: FD210062   LD IY,#6200    ; load IY with start of mario sprite values
0ABA: C37829    JP #2978    ; jump back to code and continue check hammer grab

```

Dit werkt niet goed. Pakt toch nog de hamer, maar bij het timmeren ermee, worden de barrels niet meer geraakt. Dus op een verkeerde plek tussengevoegd. Teruggezet. Kijken waar de hammer grab dan gecheckt wordt.

Bovenstaande code wordt eerder aangeroepen:

```

2954 3E0B    LD A,#0B    ; A := #0B = 1011 binary
2956 F7      RST #30     ; if level is elevators RET from this sub now. no hammers on elevators.
2957 CD7429   CALL #2974 ; load A with 1 if hammer is grabbed, 0 if no grab

```

Dat dan vervangen:

```

2954 C3B00A    JP #0AB0    ; jump to additional code – jump to label AAAA

```

en

```

0AB0: 3E0B    LD A,#0B    ; A := #0B = 1011 binary – label AAAA
0AB2: F7      RST #30     ; if level is elevators RET from this sub now.
0AB3: 3A196A    LD A,(#6A19)    ; load A with sprite value first hammer
0AB6: FE07      CP #07      ; check if sprite value is #07: invisible hammer
0AB8: C8        RET Z      ; return if invisible: no grab
0AB9: C35729    JP #2957    ; jump back to code and continue check hammer grab

```

Dit werkt prima. Hamer is niet pakbaar wanneer onzichtbaar. De interval onzichtbaar nog iets langer maken om het irriterender te maken.

Nu de springs voorzien van een extra spring ertussen door. Er is een routine die bepaalt wanneer er een nieuwe ‘bouncer’ gedeployed moet worden. Deze werkt gewoon op een timer die terugloopt en wanneer deze nul is, dient er een spring gedeployed te worden:

```

2FCB 3E0E    LD A,#0E    ; A := #E = 1110 binary
2FCD F7      RST #30     ; is this the girders? if so, return immediately

2FCE 21B462   LD HL,#62B4    ; else load HL with timer
2FD1 35      DEC (HL)    ; count down timer. at zero?
2FD2 C0      RET NZ      ; no, return

2FD3 3E03    LD A,#03     ; else A := 3
2FD5 32B962   LD (#62B9),A    ; store into fire release - a new fire can be released
2FD8 329663   LD (#6396),A    ; store into bouncer release - a new bouncer can be deployed

```

De timer is #62B4. Waar wordt deze gereset? Wat is de interval? Lijkt gereset te worden met de waarde in #62B3. Deze wordt volgens de comments gezet op #50.

Snellere bouncer releases door deze code uit te breiden met een extra stukje dat al op een eerdere waarde van de timer een bouncer release zet.

```
2FCE: C3BC0A    JP #0ABC          ; jump to additional code – jump to label AAAA

0ABC: 21B462    LD    HL,#62B4    ; load HL with timer – label AAAA
0ABF: 7E        LD A, (HL)    ; load A with timer
0AC0: FE18      CP #18      ; time to potentially release an extra bouncer?
0AC2: C2D12F    JP NZ, #2FD1    ; no, jump back
;
0AC5: 3A1860    LD  A,(RngTimer1) ; load A with random timer ?
0AC8: E607      AND #07      ; mask bits: result is random # between 0 and 7
0ACA: A7        AND A        ; check if A equals zero
0ACB: C2D12F    JP NZ, #2FD1    ; jump back
;
0ACE: 3E03      LD   A,#03     ; else A := 3
0AD0: 329663    LD   (#6396),A    ; store into bouncer release - a new bouncer can be deployed
0AD3: C3D12F    JP #2FD1      ; jump back
```

Gaat goed. Af en toe wordt een extra bouncer toegevoegd. Wellicht een geluid geven als dat gedaan wordt?

```
0ACE: 3E03      LD   A,#03     ; else A := 3
0AD0: 329663    LD   (#6396),A    ; store into bouncer release - a new bouncer can be deployed
0AD3: E5        PUSH HL
0AD4: 218560    LD   HL,#6085    ; else load HL with bonus sound address
0AD7: 3603      LD   (HL),#03    ; play bonus sound for 3 duration
0AD9: E1        POP  HL
0ADA: C3D12F    JP #2FD1      ; jump back
```

Gaat goed. Maar nu ook geluid bij de conveyors. Dat is ongewenst. Dus check inbouwen dat deze extra code alleen gedaan wordt op de elevators.

```
0ABC: 21B462    LD    HL,#62B4    ; load HL with timer – label AAAA
;
0ABF: 3A2762    LD    A,(#6227)    ; load A with screen number
0AC2: FE03      CP    #03      ; are we on elevators?
0AC4: C2D12F    JP    NZ,#2FD1    ; no, jump back
;
0AC7: 7E        LD A, (HL)    ; load A with timer
0AC8: FE18      CP #18      ; time to potentially release an extra bouncer?
0ACA: C2D12F    JP NZ, #2FD1    ; no, jump back
;
0ACD: 3A1860    LD  A,(RngTimer1) ; load A with random timer ?
0AD0: E607      AND #07      ; mask bits: result is random # between 0 and 7
```

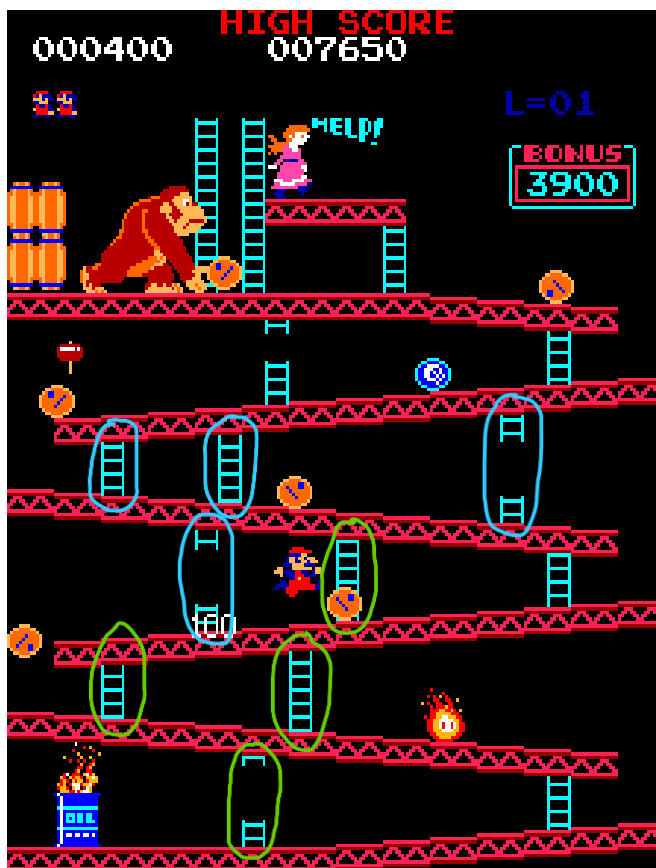
```

0AD2: A7      AND A      ; check if A equals zero
0AD3: C2D12F   JP NZ, #2FD1 ; jump back
;
0AD6: 3E03     LD  A,#03    ; else A := 3
0AD8: 329663   LD  (#6396),A ; store into bouncer release - a new bouncer can be deployed
0ADB: E5       PUSH HL
0ADC: 218560   LD  HL,#6085 ; else load HL with bonus sound address
0ADF: 3603     LD  (HL),#03 ; play bonus sound for 3 duration
0AE1: E1       POP HL
0AE2: C3D12F   JP #2FD1    ; jump back

```

Dit gaat nu goed. Dit eventjes testen met de eerste levels? Qua snelheid van springs en pies?

Aanpassen dat de top broken ladder in barrels niet weggehaald kan worden. Andere groepsverdeling maken. Ipv de top broken ladder, de bottom broken ladder in de groene groep opnemen.



```

3AE4: 02 97 38 68 38 ; top girder where girl sits
3AE9: 02 9F 54 10 54 ; girder where kong sits
3AED: 02 DF 58 A0 55 ; 1st slanted girder at top right
3AF3: 02 EF 6D 20 79 ; 2nd slanted girder (has hammer at left side)
3AF8: 02 DF 9A 10 8E ; 3rd slanted girder
3AFD: 02 EF AF 20 BB ; 4th slanted girder

```

3B02: 02 DF DC 10 D0 ; 5th slanted girder (has hammer at right side)
 3B07: 02 FF F0 80 F7 ; bottom slanted girder
 3B0C: 02 7F F8 00 F8 ; bottom flat girder where mario starts
 3B11: 00 CB 57 CB 6F ; short ladder at top right
 3B16: 00 CB 99 CB B1 ; short ladder at center right
 3B1B: 00 CB DB CB F3 ; short ladder at bottom right
 3B20: 00 63 18 63 54 ; kong's ladder (right)
~~3B25: 01 63 D5 63 F8 ; bottom broken ladder~~
 3B25: 01 6B 54 6B 75 ; fourth broken ladder near kong
 3B2A: ~~00 33 78 33 90 ; short ladder at left side under top hammer~~
 3B2F: ~~00 33 BA 33 D2 ; short ladder at left side above oil can~~
 3B34: 00 53 18 53 54 ; kong's ladder (left)
 3B39: ~~01 53 92 53 B8 ; second broken ladder from bottom, on 3rd girder~~
 3B3E: ~~00 5B 76 5B 92 ; longer ladder under the top left hammer~~
 3B43: ~~00 73 B6 73 D6 ; longer ladder to left of bottom hammer~~
 3B48: ~~00 83 95 83 B5 ; center longer ladder~~
 3B4D: 00 93 38 93 54 ; ladder leading to girl
 3B52: ~~01 BB 70 BB 98 ; third broken ladder on right side near top~~
 3B57: ~~01 6B 54 6B 75 ; fourth broken ladder near kong~~
 3B5C: AA ; AA code signals end of data

Definitie van de groene ladders (index met 7 ladders: 6307):

3B35: 00 33 BA 33 D2 AA ; short ladder at left side above oil can
 3B3B: 00 73 B6 73 D6 AA ; longer ladder to left of bottom hammer
 3B41: 00 83 95 83 B5 AA ; center longer ladder
~~3B47: 01 6B 54 6B 75 AA ; fourth broken ladder near kong~~
 3B47: 01 63 D5 63 F8 AA ; bottom broken ladder

Definitie van de blauwe ladders (index met 11 ladders: 630B):

3B4D: 00 33 78 33 90 AA ; short ladder at left side under top hammer
 3B53: 01 53 92 53 B8 AA ; second broken ladder from bottom, on 3rd girder
 3B59: 00 5B 76 5B 92 AA ; longer ladder under the top left hammer
 3B5F: 01 BB 70 BB 98 AA ; third broken ladder on right side near top

Gedaan. De onderste broken ladder wordt nu af en toe verwijderd en de bovenste broken ladder wordt nu niet meer verwijderd.

Titel in het titelscherm veranderen in ON THE RUN:

in #3F8F de titel maken:

CE 76 10 10 1F 1E 10 24 18 15 10 22 25 1E 10 10 3F

De titel boven aan het scherm aanpassen aan ON THE RUN

Was: #36B2: C0 76 23 24 1F 1C 15 1E 10 1C 11 14 14 15 22 23 3F

Nu: #36B2: C0 76 10 10 1F 1E 10 24 18 15 10 22 25 1E 10 10 3F

De tekst in het progress screen veranderen in SABOTAGED ? REALLY ?

De tekst was: WHO STOLE THAT LADDER ?

27 18 1F 10 23 24 1F 1C 15 10 24 18 11 24 10 1C 11 14 14 15 22 10 FB

Veranderen in:

10 23 11 12 1F 24 11 17 15 14 10 FB 10 22 15 11 1C 1C 29 10 FB FB 10

Starten vanaf L=05.

095E 01 65 3A 01 00 00 00 ; #3A65 is start of table data for screens/levels

Waarin de eerste 01 gekopieerd wordt naar #6229 (is level #)

En de 653A gekopieerd wordt naar #622A (is pointer in level/screen table)

Als starten L=05 dan dit aanpassen naar 05 en 73 3A.

Jumpman aangepast naar politieagent.

Items aangepast naar politie gerelateerde items:

Hat = Donut - kleurcode 0A

Purse = Badge - kleurcode 01

Umbrella = Money bag - kleurcode 0F

Kijken of we de elevators kunnen laten stoppen. Elevators move gebeurd hier:

; elevator is moving up

27AE DD7E05 LD A,(IX+#05) ; load A with elevator Y position

27B1 3D DEC A ; decrement (move up)

27B2 DD7705 LD (IX+#05),A ; store result

27B5 FE60 CP #60 ; at top of elevator ?

27B7 C2C227 JP NZ,#27C2 ; no, skip next 2 steps

27BA DD360377 LD (IX+#03),#77 ; set X position to right side of elevators

27BE DD360D04 LD (IX+#0D),#04 ; set direction to down

27C2 DD19 ADD IX,DE ; add offset for next elevator

27C4 10DA DJNZ #27A0 ; next B

27C6 C9 RET ; return

; elevator is moving down

27C7 DD7E05 LD A,(IX+#05) ; load A with elevator Y position

```

27CA 3C      INC  A      ; increase (move down)
27CB DD7705 LD   (IX+#05),A ; store result
27CE FEF8    CP   #F8    ; at bottom of shaft ?
27D0 C2C227 JP   NZ,#27C2 ; no, loop for next

27D3 DD360000 LD (IX+#00),#00 ; yes, make this elevator inactive
27D7 C3C227 JP   #27C2      ; jump back and loop for next elevator

```

Dus de eerste aanpassen naar een jump to additional code en daar dan testen op de gezette waarde van een geheugenpositie. Als gezet dan geen increase of decrease. De geheugenplaats hiervoor te gebruiken is #638D.

```

27AE: CD0A0B CALL #0B0A      ; jump to additional code – jump to label AAAA
27B1: 00      NOP

```

en

```

27C7: CD0A0B CALL #0B0A      ; jump to additional code – jump to label AAAA
27CA: 00      NOP

```

en

```

0B0A: DD7E05 LD   A,(IX+#05) ; load A with elevator Y position – label AAAA
0B0D: 47      LD   B,A      ; temporarily store A to B
0B0E: 3A8D63 LD   A,(#638D) ; load A with elevator stop indicator
0B11: FE01    CP   #01    ; is elevator stop indicator set ?
0B13: 78      LD A,B      ; restore B to A
0B14: C8      RET Z      ; yes, return
;
0B15: DDCB0D5E BIT  3,(IX+#0D) ; is this elevator moving down ?
0B19: 2802    JR   Z,#BBBB ; yes, skip ahead – jump to label BBBB
;
0B1B: 3D      DEC  A      ; decrement (move up)
0B1C: C9      RET      ; return
;
0B1D: 3C      INC  A      ; increase (move down) – label BBBB
0B1E: C9      RET      ; return – label

```

Deze dan zetten vanuit de extra routine die ook de hammer blink regelt:

```

19C2: CD7B0A      ; do blinking of hammers and stopping of elevators

```

```

0A7B: 3A2762 LD   A,(#6227) ; load A with screen number
0A7E: FE03    CP   #03    ; are we on the elevators?
0A80: C8      RET Z      ; yes, return
;

```

```

0A7B: C3E50A JP 0AE5 ; jump to additional code
0A7E: 000000 NOP,NOP,NOP ; no operations to fill in space
;
0A81: 3A8563 LD A,(#6385) ; load A with blink counter
0A84: 3C INC A ; increment blink counter
0A85: 328563 LD (#6385), A ; store blink counter
;
0A88: FE50 CP #50 ; is blink counter #50?
0A8A: CA9B0A JP Z, #0A9B ; yes: make hammer invisible – jump to label AAAA
;
0A8D: FEA0 CP #A0 ; is blink counter #A0?
0A8F: CAA40A JP Z, #0AA4 ; yes: make hammer visible – jump to label BBBB
;
0A92: FEC0 CP #C0 ; is blink counter greater than #C0?
0A94: D8 RET C ; no return
0A95: 3E00 LD A, #00 ; load A with #00
0A97: 328563 LD (#6385), A ; store blink counter
0A9A: C9 RET ; return
;
0A9B: 3E07 LD A, #07 ; load A with value empty sprite – label AAAA
0A9D: 32196A LD (#6A19), A ; make first hammer invisible
0AA0: 321D6A LD (#6A1D),A ; make second hammer invisible
0AA3: C9 RET ; return
;
0AA4: 3E1E LD A, #1E ; load A with hammer sprite – label BBBB
0AA6: 32196A LD (#6A19), A ; make first hammer visible
0AA9: 321D6A LD (#6A1D),A ; make second hammer visible
0AAC: C9 RET ; return

en

0AE5: 3A2762 LD A,(#6227) ; load A with screen number
0AE8: FE03 CP #03 ; are we on the elevators?
0AEA: C2810A JP NZ, #0A81 ; no, jump back to hammer blink code
;
0AED: 3A1A60 LD A,(FrameCounter) ; load A with framecounter
0AF0: FE80 CP #80 ; is A greater than 80?
0AF2: D2FE0A JP NC,#0AFE ; yes, reset elevator stop indicator – jump to label EEEE
;
0AF5: FE60 CP #60 ; is A less than 60?
0AF7: DAFE0A JP C,#0AFE ; yes, reset elevator stop indicator – jump to label EEEE
;
0AFA: 3E01 LD A, #01 ; set elevator stop indicator
0AFC: 1802 JR #FFFF ; skip next steps – jump to label FFFF
;
0AFE: 3E00 LD A,#00 ; reset elevator stop indicator – label EEEE
0B00: 328D63 LD (#638D), A ; configure elevator stop indicator – label FFFF
0B03: C9 RET
;

```

Dit aangepast. Gaat niet goed. Tijdens testen de elevators helemaal stopgezet. Dat werkt, maar wanneer je erop springt, gaat jumpman nog wel omhoog. Dus beweging van jumpman zou je dan ook afhankelijk moeten maken van de elevator stop indicator. En daar is geen ruimte meer voor.

Dus teruggaan naar de vorige versie.

Dan proberen om de poster afgebeeld te krijgen in het pies level (omdat daar de kleuren goed zijn).

De code voor het tekenen van de Goofy Kong is de volgende (plaatsen vanaf #0AE5)

```
0AE5: 21BC75 LD HL,#75BC ; load HL with screen location start for goofy kong
;
0AE8: 0E90 LD C,#90 ; C := #90 = start graphic for wanted poster
;
0AEA: 71 LD (HL),C ; draw part of goofy kong
0AEB: 0C INC C ; next graphic
0AEC: 2B DEC HL ; next screen location
0AED: 71 LD (HL),C ; draw part of goofy kong
0AEE: 0C INC C ; next graphic
0AEF: 2B DEC HL ; next screen location
0AF0: 71 LD (HL),C ; draw part of goofy kong
0AF1: 0C INC C ; next graphic
0AF2: 2B DEC HL ; next screen location
0AF3: 71 LD (HL),C ; draw part of goofy kong
0AF4: 0C INC C ; next graphic
0AF5: 2B DEC HL ; next screen location
0AF6: 71 LD (HL),C ; draw part of goofy kong
0AF7: 79 LD A,C ; load A with graphic number
0AF8: FE A3 CP #A3 ; == #A3 ? (are we done?)
0AFA: 2808 JR Z,#0B03 ; yes, skip next 4 steps
;
0AFC: 0C INC C ; next C
0AFD: 112400 LD DE,#0024 ; load DE with offset
0B00: 19 ADD HL,DE ; add to screen location
0B01: 18E7 JR #0AEA ; loop again
;
0B03: C9 RET ; return
```

Dit tekenen nog ergens tussenvoegen op conveyors:

Einde van het tekenen van extra zaken voor conveyors is de volgende code:

```
1081 3E01 LD A,#01 ; A := 1
1083 32B962 LD (#62B9),A ; store into fire release
1086 C9 RET ; return
```

Veranderen in:

1081: C30A0B JP #0B0A ; jump to additional code

```
0B0A: 3E01 LD A,#01 ; A := 1
0B0C: 32B962 LD (#62B9),A ; store into fire release
0B0E: CDE50A CALL #0AE5 ; draw wanted poster
;
0B11: C9 RET ; return
```

Ook tekenen in de barrels stage.

```
0D84 210B69 LD HL,#690B ; load HL with start of kong sprite
0D87 0EFC LD C,#FC ; set to move by -4
0D89 FF RST #38 ; move kong
0D8A C9 RET ; return
```

Dit vervangen door:

```
0D84: C3130B JP #0B13 ; jump to additional code

0B13: 210B69 LD HL,#690B ; load HL with start of kong sprite
0B16: 0EFC LD C,#FC ; set to move by -4
0B18: FF RST #38 ; move kong
0B19: CDE50A CALL #0AE5 ; draw wanted poster
0B1C: C9 RET ; return
```

De poster behouden in de barrels stage en verwijderen uit de conveyor stage.

Als de ladder links naast de oil can gestolen is, kunnen de fireballs niet naar beneden.



Dus de definities moeten aangepast:

Definitie van de groene ladders (index met 7 ladders: 6307):

3FC8: 00 63 80 63 A8 AA ; center ladder to left of oil can fire

3FCE: 00 9B D0 9B F8 AA ; bottom level ladder #3 of 4

3FD4: 00 D3 80 D3 A8 AA ; ladder on right platform with umbrella

3FDA: 00 2B D0 2B F8 AA ; bottom level ladder #1 of 4

Dus de groene ladder op 3FC8 moet altijd getekend worden.

Omwisselen met:

3CA9: 00 2B 80 2B A8 ; ladder on left platform with hammer

Zodat de volgende configuratie ontstaat:



Aangepast. En getest. Beide ladders zijn te beklommen. Dus de definities zijn goed.

In overleg de ladders toch weer terug naar de oude configuratie. De fireballs gaan nu snel omhoog en dat is feitelijk ook een extra moeilijkheid. Wellicht strategie om in dat gevla snel naar rechts te lopen zodat er ook een paar aan de rechterkant spawnen.

De moeilijkheidsgraad zo configureren:

Groene ladder gestolen (Barrels/Pies/Springs/Rivets)	Altijd
Blauwe ladder gestolen (Barrels/Pies/Rivets)	Vanaf instelbaar level: 3F59: 3A2962 LD A,(#6229) ; load A with level number 3F5C: FE05 CP #05 ; A >= 5 ?
Rode ladder gestolen (Rivets = bovenste 4 ladders)	Vanaf instelbaar level: 0B90: 3A2962 LD A,(#6229) ; load A with level number 0B93: FE0F CP #0F ; A >= 15 ?
Rode ladder gestolen (Springs = ladder naast elevator)	Vanaf instelbaar level: 0BB2: 3A2962 LD A,(#6229) ; load A with level number 0BB5: FE0C CP #0C ; A >= 12 ?

Starten vanaf L=05.

095E 01 65 3A 01 00 00 00 ; #3A65 is start of table data for screens/levels

Waarin de eerste 01 gekopieerd wordt naar #6229 (is level #)
En de 653A gekopieerd wordt naar #622A (is pointer in level/screen table)

Als starten L=04 dan dit aanpassen naar 04 en 6E 3A. Getest overgang 04 / 05: **blauw erbij**.

Als starten L=11 dan dit aanpassen naar 0B en 73 3A. Getest overgang 11 / 12: **rood elevators erbij**.

Als starten L=14 dan dit aanpassen naar 0E en 73 3A. Getest overgang 14 / 15: **rood rivets erbij**.

Nu weer terugzetten naar originele start.

Nu nog de invincibility opheffen.

19B3 CD0828 CALL #2808 ; check for collisions with hostile sprites [set to NOPS to make mario invincible to enemy sprites]

Kijken naar review opmerkingen JP.

Closely spaced pies vaker laten voorkomen.

2576: DD360001 LD (IX+#00),#01 ; set pie active
257A: DD360908 LD (IX+#09),#08 ; set pie size??? (width?)
257E: 3A1860 LD A,(RngTimer1) ; load A with random timer ?
2581: E603 AND #03 ; mask bits: result is random # between 0 and 3
2583: A7 AND A ; check if A equals zero
2584: CAC738 JP Z, #38C7 ; if zero jump to additional code 1 – jump to label **AAAA**
2587: C3D738 JP #38D7 ; else jump to additional code 2 – jump to label **BBBB**
258A: NOP

en

38C7 DD36074C LD (IX+#07),#4C ; set pie sprite value – pie version 1 – label **AAAA**
38CB: DD360A01 LD (IX+#0A),#01 ; set pie size??? (height?)

```

38CF: 3E24      LD   A,#24      ; A := #24
38D1: 329B63    LD   (#639B),A    ; store into pie timer for next pie deployment
38D4: C38B25    JP    #258B      ; jump back

```

en

```

38D7: DD36074B LD   (IX+#07),#4B ; set pie sprite value – pie version 1 – label BBBB
38DB: DD360A03 LD   (IX+#0A),#03 ; set pie size??? (height?)
38DF: 3E7C      LD   A,#7C      ; A := #7C
38E1: 329B63    LD   (#639B),A    ; store into pie timer for next pie deployment
38E4: C38B25    JP    #258B      ; jump back

```

Mogelijk door de AND in #2581 te veranderen in een AND met #02. Dan kans van 50% ipv 25% op een closely spaced pie. Dat werkt zeker. Je krijgt dan ook meer pies in het scherm omdat ze sneller spawned worden.

Kan ook kijken of de pie timer voor een normale deployment ook iets lager gezet kan worden. Dan krijg je sneller een pie (waardoor dus meer op het scherm). Dat kan door de waarde in #38DF te verlagen. Bijvoorbeeld naar #50. Nee. Dat is niet echt een verbetering.

Vrije ruimte eventueel nog:

```

3CE1: 15 posities.
3FBA: 6 posities.
0BCD: 13 posities.

```

34 bytes free rom-space left. Dat is niet heel erg veel!

Barrels moeilijker maken. Kan door tweaken van het mechanisme wellicht. De kans op wil barrels verhogen? Of een fireball spawnen bovenin het veld? Fireball spawnen kost 24 posities.

Vanaf #2C03 is het stukje code van barrel deployment. Mogelijk om kans op crazy barrel te vergroten. Gebaseerd op difficulty level 1 t/m 5. Maar voordat je gaat vergelijken de ingelezen waarde verhogen met 5? Daarmee eens testen.

```

2C23 3A8063 LD   A,(#6380)    ; if not, then load A with difficulty from 1 to 5
2C26 47      LD   B,A         ; For B = 1 to difficulty
2C27 3A1A60 LD   A,(FrameCounter) ; load A with timer value.
2C2A E61F    AND   #1F        ; zero out left 3 bits. the result is between 0 and #1F
;
2C2C B8      CP    B          ; compare with Loop counter B (between 1 and 5)
2C2D CA332C JP    Z,#2C33     ; if it equal then jump ahead to check for a crazy barrel
;
2C30 10FA    DJNZ  #2C2C      ; else Next B

2C23 CDCD0B JP    #0BCD       ; call to label AAAA

0BCD: 3A8063 LD   A,(#6380)    ; load A with difficulty from 1 to 5

```



```

OBD0: 3C      INC A
OBD1: 3C      INC A
OBD2: 3C      INC A
OBD3: 3C      INC A
OBD4: C9      RET

```

Niet echt een groot verschil. Gaat hem niet worden.

Twee keer zoveel kans op wild barrel:

```

2C44 E60F    AND    #0F      ; mask out left 4 bits to zero. A becomes a number between 0 and
#F

```

#0F veranderen in #07

Nu nog kijken of het type aangepast kan worden. Nu nog veel drop barrels (recht naar beneden).
Hoe wordt dat bepaald? Nee, geen idee. Voorlopig het aantal wild barrels weer terugzetten.

Kijken of het toch niet mogelijk is om de barrels broken te maken ipv te verwijderen.

In plaats van een skip van de ladder (een aantal INC's door de ladder groep), zou er ook een variabele gezet kunnen worden. Als deze gezet, dan bij het tekenen en definiëren van de ladder condities maken die testen op deze variabele en dan een override doen: broken maken.

De volgende variabele daarvoor gebruiken: #638E (wordt normaal gebruikt in het intro screen, maar dat is er niet meer).

De code voor het niet tekenen van een ladder was:

```

3FE2: 78      LD A, B          ; A = B - label BBBB
3FE3: B9      CP C             ; A = C ? : that is: B is equal to C?
3FE4: C2EF3F  JP NZ, #3FEF     ; no, skip next steps – jump to label EEEE
;
3FE7: 23      INC HL           ; next table entry: 5 times to skip drawing this ladder
3FE8: 23      INC HL           ; next table entry
3FE9: 23      INC HL           ; next table entry
3FEA: 23      INC HL           ; next table entry
3FEB: 23      INC HL           ; next table entry
3FEC: C3FA3F  JP #3FFA         ; skip next steps – jump to label FFFF
;
3FEF: <code die de ladder toevoegt>

```

Nu de INC's niet doen (niet door groep heen te skippen) en ook laatste regel niet nodig: moet wel gewoon ladders gaan toevoegen. In het vrijgekomen stuk de variabele zetten:

```

3FE2: 78      LD A, B      ; A = B - label BBBB
3FE3: B9      CP C        ; A = C ? : that is: B is equal to C?
3FE4: 3E00    LD A, #00    ; load A with 0, carry of previous CP is preserved
3FE6: C2EB3F  JP NZ, #3FEB ; no, skip next steps – jump to label EEEE
;
3FE9: 3E01    LD A, #01    ; load A with 1, override and make broken
3FEB: 328E63  LD (#638E), A ; set broken ladder override – label EEEE
3FEE: 00      NOP
;
3FEF: <code die de ladder toevoegt>

```

Bij het tekenen van de ladder, wordt hier bepaald of het een broken ladder wordt:

```

0E0A D630 SUB #30 ; subtract #30. now the element to draw is a ladder
0E0C 77 LD (HL),A ; draw element to screen = top of ladder
0E0D 3AB363 LD A, (#63B3) ; load A with original data item
0E10 FE01 CP #01 ; == 1 ? (is this a broken ladder?)
0E12 C2190E JP NZ, #0E19 ; no, skip next 2 steps

```

Op #0E0D kan je springen naar een stukje additionele code die de override doet:

```

0E0D: C3E13C JP #3CE1 ; jump to additional code – jump to label AAAA

```

en

```

3CE1: 3A8E63 LD A, (#638E) ; load A with broken ladder override – label AAAA
3CE4: FE01 CP #01 ; == 1 ? (is override set?)
3CE6: CA150E JP Z, #0E15 ; yes, draw broken ladder
3CE9: 3AB363 LD A, (#63B3) ; load A with original data item
3CEC: C3100E JP #0E10 ; jump back

```

Dit werkt in principe goed. Trap wordt broken getekend en je kunt er nog wel gebruik van maken (moet nog aangepast worden). Alleen het onderste stuk van de broken trap wordt niet getekend. Uitzoeken hoe dat komt.

Voor het onderste deel is er ook een zelfde stukje:

```

0E33 3AB363 LD A, (#63B3) ; load A with original data item
0E36 FE01 CP #01 ; == 1 ? (is this a broken ladder ?)
0E38 C23F0E JP NZ, #0E3F ; no, skip next 3 steps

```

```

0E33: C3400C JP #0C40 ; jump to additional code – jump to label AAAA

```

```

0C40: 3A8E63 LD A, (#638E) ; load A with broken ladder override – label AAAA
0C43: FE01 CP #01 ; == 1 ? (is override set?)
0C45: CA3B0E JP Z, #0E3B ; yes, draw broken ladder

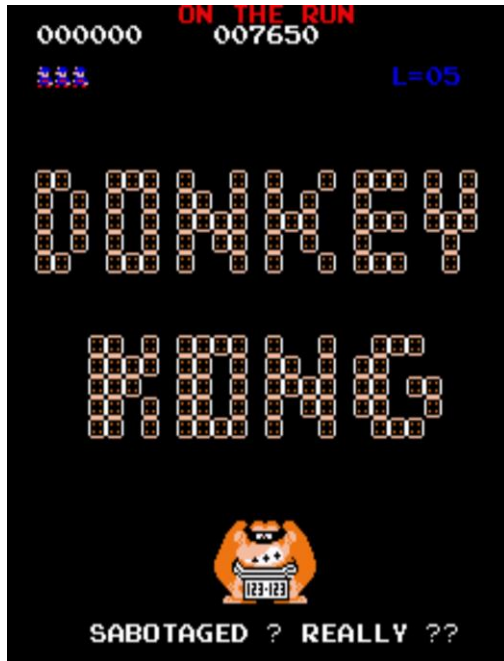
```

```

0C48: 3AB363    LD    A, (#63B3)    ; load A with original data item
0C4B: C3360E    JP    #0E36        ; jump back

```

Ruimte vrijmaken bij het tekenen van de Goofy Kong? Ze niet meer gaan stapelen? Ja, aangepast. Er wordt één Goofy Kong getekend. Daarboven wordt het Kong logo getekend. Ziet er zo uit:



Hiermee is de volgende ruimte vrijgekomen: #0C40 t/m #0C90.

Bij het toevoegen van de ladder aan de definitie, wordt hier bepaald of het een broken ladder wordt:

```

2471 DD210063 LD    IX,#6300    ; #6300 is used for ladder positions?
2475 110500 LD    DE,#0005    ; DE := 5 = offset

2478 7E      LD    A,(HL)      ; load A with the next item of data
2479 A7      AND    A          ; is this item == 0 ?
247A CA8824 JP    Z,#2488      ; yes, jump ahead

247D 3D      DEC    A          ; no, decrease, was this item == 1 ?
247E CA9E24 JP    Z,#249E      ; yes, jump down instead

2481 FEA9    CP    #A9         ; was the item == #AA ?
2483 C8      RET    Z          ; yes, return, we are done with this. AA is at the end of each table

2484 19      ADD    HL,DE      ; if neither then add offset for next HL
2485 C37824 JP    #2478        ; loop again

```

Dus ook daar kan je springen naar een stukje code die de override doet:

```

2478: C3400C    JP    #0C40        ; jump to additional code
247B: 00          NOP
247C: 00          NOP
en

```

```

0C40: 3A8E63    LD    A, (#638E)    ; load A with broken ladder override – label AAAA

```

```

0C43: FE01      CP      #01          ; == 1 ? (is override set?)
0C45: CA9E24    JP      Z, #249E      ; yes, add broken ladder
0C48: 7E        LD      A,(HL)        ; load A with the next item of data
0C49: A7        AND     A              ; is this item == 0 ?
0C4A: CA8824    JP      Z,#2488      ; yes, jump ahead
0C4D: C37D24    JP      #247D          ; jump back

```

Dit gaat nog niet goed. Wellicht komt dit omdat deze routine ook gebruikt wordt voor het tekenen van de andere elementen (zoals de girders). Dan is het raar om ook te checken op de override. Dus de check op de override alleen doen als er een fixed ladder gevonden is.

Dus wordt dan:

```

247A: CACD0B    JP      Z, #0BCD          ; jump to additional code

```

en

```

0BCD: 3A8E63    LD      A,(#638E)      ; load A with broken ladder override – label AAAA
0BD0: FE01      CP      #01          ; == 1 ? (is override set?)
0BD2: CA9E24    JP      Z, #249E      ; yes, add broken ladder
0BD5: C38824    JP      #2488          ; no, add fixed ladder

```

Werkt nu in principe goed. Twee problemen nog:

- Barrels: als teveel broken ladders dan niet speelbaar:
 - Definities groepen aanpassen, lastig: goed over nadenken.
- Algemeen: broken ladder override van de vorige keer beïnvloed tekenen rest van screen:
 - De broken ladder override variabele initialiseren voor tekenen scherm.

Tekenen van het scherm gebeurt hier:

```

0CC6 CDA70D    CALL   #0DA7          ; draw the screen

```

Veranderen in:

```

0CC6: CD4E0C    CALL   #0C4E          ; jump to additional code – 0C61 tijdelijk

```

en

```

0C4E: 3E00      LD A, #00          ; load A with 0 – label AAAA
0C50: 328E63    LD (#638E), A      ; reset broken ladder override
0C53: CDA70D    CALL   #0DA7          ; draw the screen
0C57: C9        RET              ; return

```

Barrels teveel broken: daarom ze eventjes allemaal fixed maken:

```

3AE4: 02 97 38 68 38 ; top girder where girl sits
3AE9: 02 9F 54 10 54 ; girder where kong sits
3AED: 02 DF 58 A0 55 ; 1st slanted girder at top right

```

3AF3: 02 EF 6D 20 79 ; 2nd slanted girder (has hammer at left side)
 3AF8: 02 DF 9A 10 8E ; 3rd slanted girder
 3AFD: 02 EF AF 20 BB ; 4th slanted girder
 3B02: 02 DF DC 10 D0 ; 5th slanted girder (has hammer at right side)
 3B07: 02 FF F0 80 F7 ; bottom slanted girder
 3B0C: 02 7F F8 00 F8 ; bottom flat girder where mario starts
 3B11: 00 CB 57 CB 6F ; short ladder at top right
 3B16: 00 CB 99 CB B1 ; short ladder at center right
 3B1B: 00 CB DB CB F3 ; short ladder at bottom right
 3B20: 00 63 18 63 54 ; kong's ladder (right)
 3B25: 01 63 D5 63 F8 ; bottom broken ladder
 3B2A: 00 53 18 53 54 ; kong's ladder (left)
 3B2F: 00 93 38 93 54 ; ladder leading to girl
 3B34: AA ; AA code signals end of data

Definitie van de groene ladders (index met 7 ladders: 6307):

3B35: 00 33 BA 33 D2 AA ; short ladder at left side above oil can
 3B3B: 00 73 B6 73 D6 AA ; longer ladder to left of bottom hammer
 3B41: 00 83 95 83 B5 AA ; center longer ladder
 3B47: 01 6B 54 6B 75 AA ; fourth broken ladder near kong

Definitie van de blauwe ladders (index met 11 ladders: 630B):

3B4D: 00 33 78 33 90 AA ; short ladder at left side under top hammer
 3B53: 01 53 92 53 B8 AA ; second broken ladder from bottom, on 3rd girder
 3B59: 00 5B 76 5B 92 AA ; longer ladder under the top left hammer
 3B5F: 01 BB 70 BB 98 AA ; third broken ladder on right side near top

01 – 00 gemaakt voor test

Met de barrels de volgende twee uitdagingen:

- De gebroken trappen weer heel maken werkt alleen voor de gebroken trap rechtsboven, da andere trappen werkt het niet voor. Jumpman blijft doorklimmen.
 - Slim hiermee omgaan bij het opstellen van de groepen.
- De kleine trappen aan de uiteinden kunnen wel gebroken gemaakt worden, maar dat is niet zichtbaar omdat de trapuiteinden boven en beneden elkaar overlappen.
 - Wellicht conditie: als op girders en kleine trap (verschil y-waarden), dan bovenste deel niet tekenen?

Testen met het tekenen van alleen de onderkant van de ladder (dat is ook alleen de kant waarop je de ladder op kunt gaan).

In de originele code wordt het bovenste stuk van de ladder getekend door #0E07. Eigenlijk moet je die weer wissen als het een gebroken ladder is.

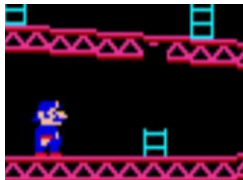
Dus de code wordt dan (past niet meer in stuk van 3CE1 t/m 3CED):

0E0D: C3570C JP #0C57 ; jump to additional code – jump to label AAAA

en

```
0C57: 3A8E63 LD A,(#638E) ; load A with broken ladder override – label AAAA
0C5A: FE01 CP #01 ; == 1 ? (is override set?)
0C5C: CA650C JP Z, #0C65 ; yes, draw broken ladder – jump to label BBBB
0C5F: 3AB363 LD A, (#63B3) ; load A with original data item
0C62: C3100E JP #0E10 ; jump back
;
0C65: 3610 LD (HL), #10 ; erase already drawn top of ladder – label BBBB
0C67: C3150E JP #0E15 ;
```

Dit gaat ook niet goed omdat dan de ook de girder gesloopt wordt. En jumpman en de fireballs kunnen er in blijven vastzitten. Geen goede oplossing dus.



Terug naar vorige versie en proberen of het werkt met het weglaten van het onderste deel.

0E33: C3400C JP #0C40 ; jump to additional code – jump to label AAAA

```
0C40: 3A8E63 LD A,(#638E) ; load A with broken ladder override – label AAAA
0C43: FE01 CP #01 ; == 1 ? (is override set?)
0C45: CA3B0E JP Z, #0E3B ; yes, draw broken ladder
0C48: 3AB363 LD A, (#63B3) ; load A with original data item
0C4B: C3360E JP #0E36 ; jump back
```

Verander de 0C45 naar JP Z,#0E3F voor een skip van het tekenen van een stukje ladder.

Dan ziet de kleine ladder links beneden er zo uit:



En de kleine ladder links boven zo:



De kleine ladders aan de rechterkant worden nooit gekozen.

Dus eigenlijk checken of op barrels en x-waarde gelijk aan die van de kleine ladders: dan springen naar #0E3F en anders naar #0E3B.

#63B3 lijkt het eerste data-item uit de reeks te bevatten.

Dus dan de x (is het tweede data-item) als volgt te bepalen:

Uit de voorgaand uitgevoerde code lijkt het alsof DE de pointer in de datastructuur is en dat deze inmiddels wijst naar de laatste y-coördinaat. De x-coördinaat staat daar voor:

```
PUSH DE
DEC DE
LD A, (DE)
POP DE
```

Dit integreren in de code:

```
0E33: C3400C    JP    #0C40          ; jump to additional code – jump to label AAAA

0C40: 3A8E63    LD    A,(#638E)        ; load A with broken ladder override – label AAAA
0C43: FE01      CP    #01              ; == 1 ? (is override set?)
0C45: CA4E0C    JP    Z, #0C4E          ; yes force broken ladder – jump to label BBBB
0C48: 3AB363    LD    A, (#63B3)        ; load A with original data item
0C4B: C3360E    JP    #0E36            ; jump back
;
0C4E: 3A2762    LD    A,(#6227)        ; load a with screen number – label BBBB
0C51: 3D        DEC    A              ; is this the barrels?
0C52: C23B0E    JP    NZ,#0E3B          ; no, draw normal broken ladder
0C55: D5        PUSH   DE              ; store DE
0C56: 1B        DEC    DE              ; decrement DE, DE points to data-item with x-position
0C57: 1A        LD    A, (DE)          ; A = x-position ladder
0C58: D1        POP    DE          ; restore DE
0C59: FE33      CP    #33              ; check if short ladder at left side
0C5B: CA3F0E    JP    Z, #0E3F          ; yes, draw short broken ladder
0C5E: C33B0E    JP    #0E3B            ; no, draw normal broken ladder
```

Dit werkt: de korte ladders worden goed getekend en de lange gebroken ladders worden ook goed getekend.

Nu eventjes kijken naar de manier om de rode groep van de rivets minder moeilijk te maken. Twee mogelijke oplossingen:

- Rode groep 2x2 dezelfde trappen en de andere twee altijd tekenen
 - Vraag is of voor dat laatste ruimte is ?
- Bij vaststellen van het random getal extra code toevoegen die bij rivets/rood de ongewenste trappen mapt op een gewenste trap.
 - Vraag is of je kunt bepalen welke groep het is ?

Dit is de definitie van de rode groep van rivets:

```
3BBE: 00 33 58 33 80 AA ; top left ladder
3BC4: 00 53 58 53 80 AA ; top left ladder (right side)
3BCA: 00 AB 58 AB 80 AA ; top right ladder (left side)
3BD0: 00 CB 58 CB 80 AA ; top right ladder
```

Een call naar #3FE0 zorgt voor het tekenen van een groep. Voor het aanroepen van het tekenen van de rode groep van rivets wordt de HL gevuld met beginadres van de groepsdefinitie (#3BBE). Dus daar zou op getest kunnen worden: De BE is uniek: dit zijn de beginadressen van de verschillende groepen:

3B35: Barrels groen
3B4D: Barrels blauw
3B8E: Rivets groen
3BA6: Rivets blauw
3BBE: Rivets rood
3C63: Springs groen
3C7B: Springs rood
3FCB: Pies groen
0B48: Pies blauw

Voorlopig eventjes laten. Wellicht is dit al een heel ander verhaal wanneer de ladders gebroken zijn ipv gestolen.

Toch meer wild barrels laten komen:

Twee keer zoveel kans op wild barrel:

2C44 E60F AND #0F ; mask out left 4 bits to zero. A becomes a number between 0 and #F

#0F veranderen in #07

Hamer sequence verschuiven:

0A88: FE10	CP #10	; is blink counter #10 (was 40)?
0A8A: CA9B0A	JP Z, #0A9B	; yes: make hammer invisible – jump to label AAAA
;		
0A8D: FE80	CP #80	; is blink counter #80 (was B0)?
0A8F: CAA40A	JP Z, #0AA4	; yes: make hammer visible – jump to label BBBB
;		
0A92: FEC0	CP #C0	; is blink counter greater than #C0?
0A94: D8	RET C	; no return
0A95: 3E00	LD A, #00	; load A with #00
0A97: 328563	LD (#6385), A	; store blink counter
0A9A: C9	RET	; return

Hierdoor is de hamer te pakken bij de run op het Kill Screen...

Eventjes Kill Screen proberen:

Starten vanaf L=05.

095E 01 65 3A 01 00 00 00 ; #3A65 is start of table data for screens/levels

Waarin de eerste 01 gekopieerd wordt naar #6229 (is level #)
En de 653A gekopieerd wordt naar #622A (is pointer in level/screen table)

Als starten L=04 dan dit aanpassen naar 04 en 6E 3A. Getest overgang 04 / 05: **blauw erbij**.

Als starten L=11 dan dit aanpassen naar 0B en 73 3A. Getest overgang 11 / 12: **rood elevators erbij**.

Als starten L=14 dan dit aanpassen naar 0E en 73 3A. Getest overgang 14 / 15: **rood rivets erbij**.

Kan hamer pakken als je er in een keer naar toe rent. En ook in de attract mode kan jumpman de hammer pakken (als hij nog niet afgegaan is).

Versie ophogen naar 1.01.

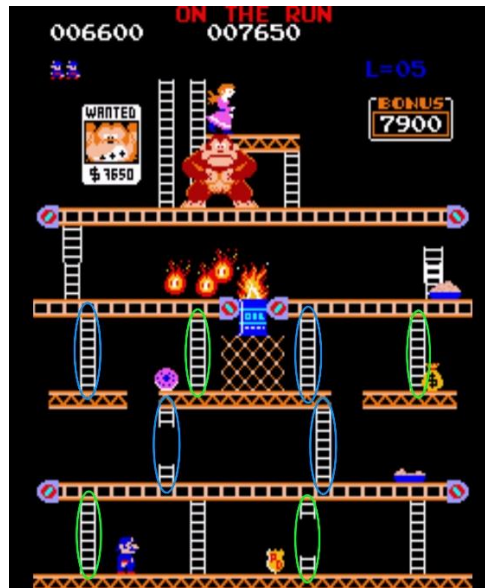
#373E: 3D 76 26 01 2B 00 01 10 3F (V1.01)

Pies ladders blauw en groen omdraaien.

Dit was de definitie:



Aanpassen naar:



Blauwe ladders definitie was:

0B48: 00 53 A8 53 D0 AA ; ladder under the hat

0B4E: 00 63 D0 63 F8 AA ; bottom level ladder #2 of 4

0B54: 00 A3 A8 A3 D0 AA ; ladder to right of bottom hammer

0B5A: 00 D3 D0 D3 F8 AA ; bottom level ladder #4 of 4

Wisselen met:

3C9A: 00 9B 80 9B A8 ; center ladder to right of oil can fire

3CA9: 00 2B 80 2B A8 ; ladder on left platform with hammer

Dus dat wordt dan:

0B48: 00 53 A8 53 D0 AA ; ladder under the hat

0B4E: 00 9B 80 9B A8 AA ; bottom level ladder #2 of 4

0B54: 00 A3 A8 A3 D0 AA ; ladder to right of bottom hammer

0B5A: 00 2B 80 2B A8 AA ; bottom level ladder #4 of 4

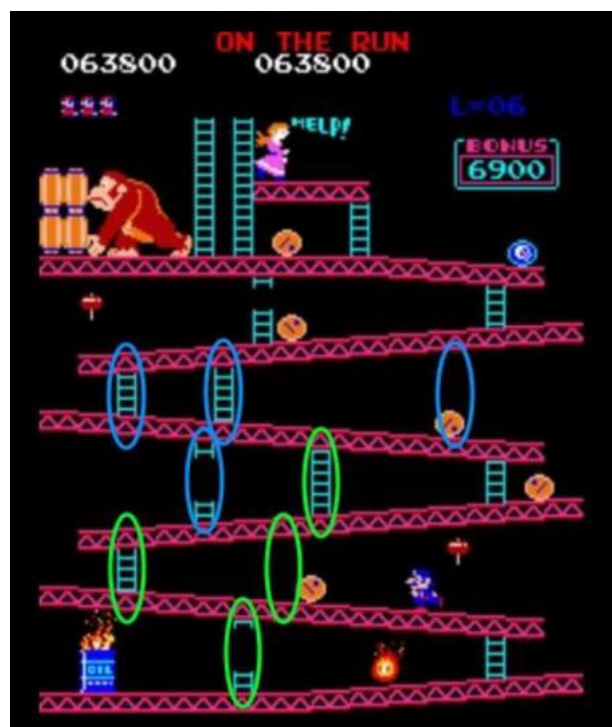
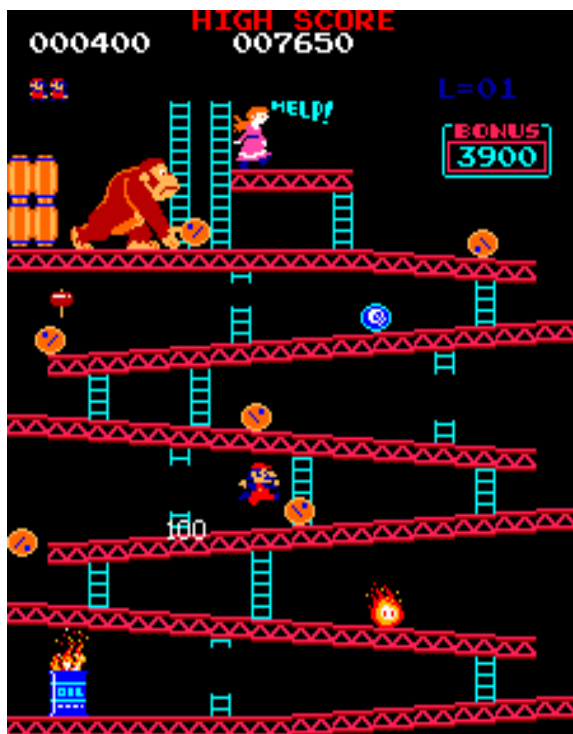
3C9A: 00 63 D0 63 F8 ; center ladder to right of oil can fire

3CA9: 00 D3 D0 D3 F8 ; ladder on left platform with hammer

Aangepast en werkt goed.

Testen van de rode rivets op level L=15. Best toch nog wel lastig. Misschien ook aanpassen naar andere indeling. Nee. Beter om de groene groep te deactiveren wanneer de rode groep actief wordt.

Nadenken barrels. Originele veld:



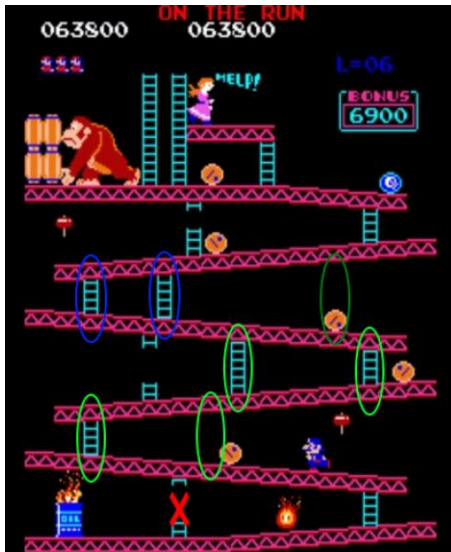
Ladder die al gebroken is, kan niet opnieuw gebroken worden.

Maximum van vier gebroken ladders.

- De gebroken ladder naast oil can laten vervallen, dan heb je er één vrijgespeeld. Dan één groep maken met de twee ladders boven en met de lange ladders van midden en onder.
- De gebroken ladder naast oil can laten vervallen en de gebroken ladder rechtsboven fixed maken, dan heb je er twee vrijgespeeld. Dan kan je twee groepen maken: één met de twee ladders boven en één met de twee ladders midden en de twee ladders onder.

De eerste optie nemen.

Dan de volgende indeling maken:



Barrels startveld:

3AE4: 02 97 38 68 38 ; top girder where girl sits
3AE9: 02 9F 54 10 54 ; girder where kong sits
3AED: 02 DF 58 A0 55 ; 1st slanted girder at top right
3AF3: 02 EF 6D 20 79 ; 2nd slanted girder (has hammer at left side)
3AF8: 02 DF 9A 10 8E ; 3rd slanted girder
3AFD: 02 EF AF 20 BB ; 4th slanted girder
3B02: 02 DF DC 10 D0 ; 5th slanted girder (has hammer at right side)
3B07: 02 FF F0 80 F7 ; bottom slanted girder
3B0C: 02 7F F8 00 F8 ; bottom flat girder where mario starts
3B11: 00 CB 57 CB 6F ; short ladder at top right
3B16: 00 CB 99 CB B1 ; short ladder at center right
3B1B: 00 CB DB CB F3 ; short ladder at bottom right
3B20: 00 63 18 63 54 ; kong's ladder (right)
3B25: 01 63 D5 63 F8 ; bottom broken ladder
3B2A: 00 53 18 53 54 ; kong's ladder (left)
3B2F: 00 93 38 93 54 ; ladder leading to girl
3B34: AA ; AA code signals end of data

Definitie van de groene ladders (index met 7 ladders: 6307):

3B35: 00 33 BA 33 D2 AA ; short ladder at left side above oil can
3B3B: 00 73 B6 73 D6 AA ; longer ladder to left of bottom hammer
3B41: 00 83 95 83 B5 AA ; center longer ladder
3B47: ~~01 63 D5 63 F8 AA ; bottom broken ladder~~

Definitie van de blauwe ladders (index met 11 ladders: 630B):

3B4D: 00 33 78 33 90 AA ; short ladder at left side under top hammer
3B53: 01 53 92 53 B8 AA ; second broken ladder from bottom, on 3rd girder -> naar startveld
3B59: 00 5B 76 5B 92 AA ; longer ladder under the top left hammer
3B5F: 01 BB 70 BB 98 AA ; third broken ladder on right side near top

In het startveld aanpassen:

3B16: 01 53 92 53 B8 ; second broken ladder from bottom, on 3rd girder

De groene groep wordt dan:

3B35: 00 33 BA 33 D2 AA ; short ladder at left side above oil can
3B3B: 00 73 B6 73 D6 AA ; longer ladder to left of bottom hammer
3B41: 00 83 95 83 B5 AA ; center longer ladder
3B47: 00 CB 99 CB B1 ; short ladder at center right

De donkergroene groep wordt dan:

3B4D: 00 BB 70 BB 98 AA ; third broken ladder on right side near top

De blauwe groep wordt dan:

3B53: 00 33 78 33 90 AA ; short ladder at left side under top hammer
3B59: 00 5B 76 5B 92 AA ; longer ladder under the top left hammer

Het startveld aangepast en de groene groep definitie aangepast. De blauwe groep eventjes niet laten meedoen.

De volgende call regelt de groene groep. NOPS maken:

3F56: CDE03F CALL #3FE0 ; call additional code to add stolen ladders – label DDDD

De volgende call regelt de blauwe groep. NOPS maken:

0B7B: CDE03F CALL #3FE0 ; call additional code to add stolen ladders – label GGGG

Het startveld klopt. Nu de groene groep laten meedoen: #3F56 herstellen.

De groene groep werkt ook goed. Nu de groene groep aanroepen vanuit een additionele code, zodat daar het tekenen van de donkergroene ook aan toegevoegd kan worden.

Stuk code voor de groene groep voor barrels:

3F31: 21353B LD HL,#3B35 ; load HL with green group of girders
3F34: DD210763 LD IX,#6307 ; load IX with index for green group of girders
3F38: C3563F JP #3F56 ; skip next steps – jump to label DDDD

Aanpassen naar:

```

3F31: 21353B    LD    HL,#3B35    ; load HL with green group of girders
3F34: DD210763 LD    IX,#6307    ; load IX with index for green group of girders
3F38: C36A0C    JP     #0C6A    ; jump to additional code – jump to label AAAA

```

en:

```

0C6A: CDE03F    CALL #3FE0    ; call additional code to add stolen ladders – label AAAA
0C6D: 0E05      LD C, #05    ; don't make ladder stolen
0C6F: 0601      LD B, #01    ; loop counter, process one ladder
0C71: CDE23F    CALL #3FE2    ; call additional code to add stolen ladders
0C74: C3593F    JP     #3F59    ; start processing of blue group

```

Gaat veel te makkelijk via de fixed ladder rechts boven naar boven. Daarom deze broken maken:

3B4D: 01 BB 70 BB 98 AA ; third broken ladder on right side near top

3B25: 00 6B 54 6B 75

De blauwe groep voor barrels activeren, met kans 50% elk en 2 ladders tekenen:

Stuk code voor de blauwe groep voor barrels:

```

0B60: 214D3B    LD    HL,#3B4D    ; load HL with blue group of girders – label YYYY
0B63: DD210B63 LD    IX,#630B    ; load IX with index for blue group of girders
0B67: C37B0B    JP     #0B7B    ; skip next steps – jump to label GGGG

```

Aanpassen naar:

```

0B60: 21533B    LD    HL,#3B53    ; load HL with blue group of girders – label YYYY
0B63: DD210C63 LD    IX,#630C    ; load IX with index for blue group of girders
0B67: C3770C    JP     #0C77    ; jump to additional code – jump to label AAAA

```

(index is one higher because of the single ladder that is added in between green and blue groups).

en:

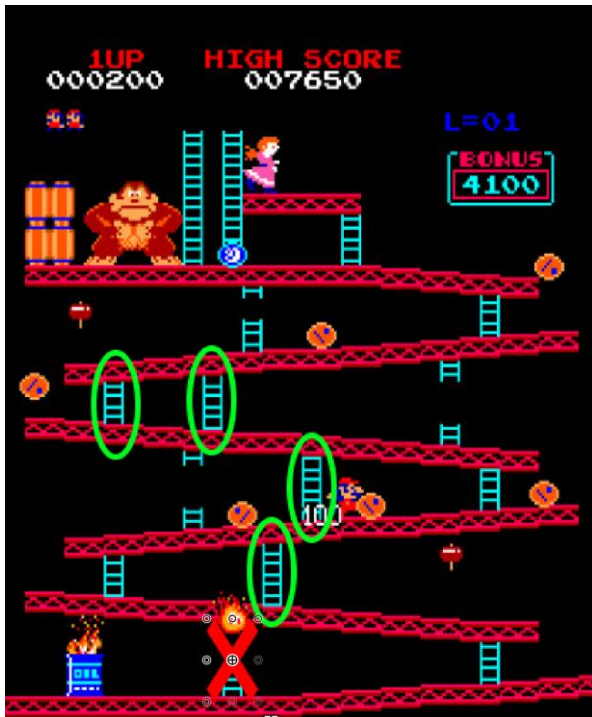
```

0C77: 79        LD A,C    ; load a with chosen ladder (1 to 4) – label AAAA
0C78: E601      AND #01    ; a is number between 0 and 1
0C7A: 3C        INC A    ; a is number between 1 and 2
0C7B: 4F        LD C,A    ; put back in c
0C7C: 0602      LD B, #02    ; loop counter, process two ladders
0C7E: CDE23F    CALL #3FE2    ; call additional code to add stolen ladders
0C81: C37E0B    JP     #0B7E    ; start processing of blue group

```

Werkt allemaal goed. Maar één probleem. De ladder rechts naast Kong is nu fixed, maar dat kan niet want jumpman blijft daar doorlopen naar boven. Dit kan niet afgevangen worden. Daar is géén ruimte voor.

Dus echt gaan voor een enkele groene groep:



De groene groep wordt dan:

```
3B35: 00 73 B6 73 D6 AA ; longer ladder to left of bottom hammer
3B3B: 00 83 95 83 B5 AA ; center longer ladder
3B41: 00 33 78 33 90 AA ; short ladder at left side under top hammer
3B47: 00 5B 76 5B 92 AA ; longer ladder under the top left hammer
```

De donkergroene groep wordt dan:

```
3B4D: 00 33 BA 33 D2 AA ; short ladder at left side above oil can
3B53: 01 53 92 53 B8 AA ; second broken ladder from bottom, on 3rd girder
3B59: 01 BB 70 BB 98 AA ; third broken ladder on right side near top
```

De volgende call regelt de groene groep. NOPS maken:

```
3F56: CDE03F CALL #3FE0 ; call additional code to add stolen ladders – label DDDD
```

De volgende call regelt de blauwe groep. NOPS maken:

```
0B7B: CDE03F CALL #3FE0 ; call additional code to add stolen ladders – label GGGG
```

De groene groep definiëren en weer aanzetten. Dat werkt.

De donkergroene groep definiëren en aanzetten:

```
3F31: 21353B LD HL,#3B35 ; load HL with green group of girders
3F34: DD210763 LD IX,#6307 ; load IX with index for green group of girders
3F38: C36A0C JP #0C6A ; jump to additional code – jump to label AAAA
```


en:

```
0C6A: CDE03F    CALL #3FE0          ; call additional code to add stolen ladders – label AAAA
0C6D: 0E05      LD C, #05          ; don't make ladder stolen
0C6F: 0603      LD B, #03          ; loop counter, process three ladders
0C71: CDE23F    CALL #3FE2          ; call additional code to add stolen ladders
0C74: C3593F    JP #3F59           ; start processing of blue group
```

Werkt goed.

Nu de blauwe groep weer aanzetten, maar voor girders daar dan niets doen:

```
3F80: C37E0B    JP #0B7E           ; jump to additional code – jump to label YYYY
```

Werkt nu goed.

Het gearceerde stuk tekent het Donkey Kong logo:

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000B40	CD	A7	0D	C3	04	0B	00	00	00	53	A8	53	D0	AA	00	9B	is.Ä....S"SD".>
00000B50	80	9B	A8	AA	00	A3	A8	A3	D0	AA	00	2B	80	2B	A8	AA	€>"*.Ä"SD".+E+"*
00000B60	21	4D	3B	DD	21	0B	63	C3	7B	0B	21	A6	3B	DD	21	06	!M;Ý!.cÄ{.!.;Ý!.
00000B70	63	C3	7B	0B	21	48	0B	DD	21	0B	63	00	00	00	3A	27	cÄ{.!H.Ý!.c....'
00000B80	62	FE	01	CA	CA	0B	FE	02	CA	CA	0B	FE	03	CA	B2	0B	bp.ÊÊ.p.ÊÊ.p.ÊÊ.
00000B90	3A	29	62	FE	0F	D2	9E	0B	3E	05	4F	C3	A5	0B	3A	18	:)bp.Ôž.>.OÄ¥..
00000BA0	60	E6	03	3C	4F	21	BE	3B	DD	21	0A	63	CD	E0	3F	C3	`æ.<O!%;Ý!.cîa?Ä
00000BB0	CA	0B	3A	29	62	FE	0C	D2	CA	0B	11	7B	3C	CD	A7	0D	Ê.:)bp.ÔÊ..{<is.
00000BC0	21	7B	3C	DD	21	0B	63	CD	75	24	C3	1F	0B	3A	8E	63	!{<Ý!.ciusÄ..:Žc
00000BD0	FE	01	CA	9E	24	C3	88	24	00	00	CD	1C	01	DF	CD	74	p.ÊžsÄ^\$.!i..Bit
00000BE0	08	16	06	3A	00	62	5F	CD	9F	30	21	86	7D	36	01	23b iÝ0!+}6.#
00000BF0	36	00	21	8A	60	36	02	23	36	03	21	A7	63	36	00	21	6.!Š`6.#6.!\$c6.!
00000C00	BC	75	0E	50	71	0C	2B	71	0C	2B	71	0C	2B	71	79	FE	4u.Pq.+q.+q.+qyb
00000C10	67	CA	1C	0C	0C	11	23	00	19	C3	04	0C	21	08	3D	3E	gÊ....#...Ä..!>=>
00000C20	FA	46	23	5E	23	56	12	13	10	FC	23	7E	FE	00	C2	1F	üF#^#V...ü#~p.Ä.
00000C30	0C	11	07	03	CD	9F	30	21	09	60	36	A0	23	34	34	C9	...iÝ0!..`6 #44Ê
00000C40	3A	8E	63	FE	01	CA	4E	0C	3A	B3	63	C3	36	0E	3A	27	:Žcp.ÊN.:`cÄ6..'
00000C50	62	3D	C2	3B	0E	D5	1B	1A	D1	FE	33	CA	3F	0E	C3	3B	b=Ä;.Ö..Ŋp3Ê?.Ä;
00000C60	0E	3E	00	32	8E	63	CD	A7	0D	C9	00	00	00	00	00	00	>.2Žcîs.Ê.....
00000C70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000C80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000C90	00	DF	CD	74	08	AF	32	8C	63	11	01	05	CD	9F	30	21	.Bit.-2æc...iÝ0!
00000CA0	86	7D	36	00	23	36	01	3A	27	62	3D	CA	D4	0C	3D	CA	+}6.#6.:`b=ÊÖ.=Ê

Dat eventueel vrijmaken. En de goofy kong en de tekst een stuk naar het midden.

De Goofy Kong start positie zit op #0C26:

```
0C1F 322F62 LD (#622F),A ; store A into current screen/level
0C22 3A2E62 LD A,(#622E) ; load A with number of goofys to draw
0C25 47 LD B,A ; copy to B for use as loop counter, refer to #0C7E
0C26 21BC75 LD HL,#75BC ; load HL with screen location start for goofy kong

0C29 0E50 LD C,#50 ; C := #50 = start graphic for goofy kong

0C2B 71 LD (HL),C ; draw part of goofy kong
0C2C 0C INC C ; next graphic
```

Maar dat is nu op: #0BFF

0BFF: 21BC75 LD HL,#75BC ; load HL with screen location start for goofy kong

Omhoog is aantal regels omhoog van dit getal aftrekken: zeg 10 regels omhoog: dan 75B2.

De tekst 'SABOTAGED REALLY' ook omhoog. #36CC is positie van de tekst: 775E. Deze 9 posities omhoog: 7755

Ziet er nu goed uit.

Nu kijken naar de red ladder in springs. Wordt actief vanaf level 12. Even starten met L12 en springs.

095E 01 65 3A 01 00 00 00 ; #3A65 is start of table data for screens/levels

Waarin de eerste 01 gekopieerd wordt naar #6229 (is level #)

En de 653A gekopieerd wordt naar #622A (is pointer in level/screen table)

Als starten L=04 dan dit aanpassen naar 04 en 6E 3A. Getest overgang 04 / 05: blauw erbij.

Als starten L=11 dan dit aanpassen naar 0B en 73 3A. Getest overgang 11 / 12: rood elevators erbij.

Als starten L=14 dan dit aanpassen naar 0E en 73 3A. Getest overgang 14 / 15: rood rivets erbij.

Ja, die wordt dus weggehaald. Zo maken dat deze ook broken wordt.

```
0BB2: 3A2962 LD A,(#6229) ; load A with level number – label MMMM
0BB5: FE03 CP #03 ; A >= 3 ?
0BB7: D2CA0B JP NC,#0BCA ; yes, skip next steps – jump to label XXXX
;
0BBA: 117B3C LD DE, #3C7B ; load DE with red group of elevators
0BBD: CDA70D CALL #0DA7 ; draw ladder to screen
0BC0: 217B3C LD HL,#3C7B ; load HL with red group of elevators
0BC3: DD210B63 LD IX,#630B ; load IX with index for red group of rivets
0BC7: CD7524 CALL #2475 ; define ladder and increase IX, HL is at AA at end
;
0BCA: 210960 LD HL, WaitTimerMSB ; load HL with timer addr – label XXXX
0BCD: C3680D JP #0D68 ; jump back
```

Wordt:

```
0BB2: 3A2962 LD A,(#6229) ; load A with level number – label MMMM
0BB5: FE0C CP #0C ; A >= 12 ?
0BB7: 3004 JR NC,#0BBD ; yes, skip next steps – jump relative to label AAAA
;
0BB9: 0E05 LD C, #05 ; C = 5 – do not make ladder broken
0BBB: 1802 JR #0BBF ; skip next steps, jump relative to label BBBB
;
```



```

0BBD: 0E01      LD   C, #01          ; C = 1 – make ladder broken – label AAAA
;
0BBF: 0601      LD   B, #01          ; B = 1 – process a single ladder – label BBBB
;
0BC1: 217B3C    LD   HL,#3C7B        ; load HL with red group of elevators
0BC4: C3770C    JP   #0C77          ; jump to additional code
;
0BC7,8,9: NOP's
;
0BCA: 210960    LD   HL, WaitTimerMSB ; load HL with timer addr – label XXXX
0BCD: C3680D    JP   #0D68          ; jump back

```

En additionele code:

```

0C77: DD210B63 LD   IX,#630B        ; load IX with index for red group of elevators
0C7B: CDE23F    CALL #3FE2          ; process ladders
0C7F: C3CA0B    JP   #0BCA          ; jump back

```

Dat werkt nu goed.

Nu de groene groep op rivets niet doen als level >= 15.

De groene groep van rivets:

```

3F3B: 218E3B    LD   HL,#3B8E        ; load HL with green group of rivets – label AAAA
3F3E: DD210263 LD   IX,#6302        ; load IX with index for green group of rivets
3F42: C3563F    JP   #3F56          ; skip next steps – jump to label DDDD

```

Aanpassen naar:

```

3F42: C32B0C    JP   #0C2B          ; jump to additional code – jump to label BBBB

```

En additionele code:

```

0C2B: 3A2962    LD   A,(#6229)          ; load A with level number – label BBBB
0C2E: FE0F      CP   #0F              ; A >= 15 ?
0C30: 3802      JR   C, #0C34          ; no, skip next steps – jump relative to label CCCC
0C32: 0E05      LD   C, #05            ; C := 5 – do not make green group rivets broken
0C34: C3563F    JP   #3F56          ; skip next steps – jump to label DDDD

```

Werkt goed.

Nu verschillende groepen testen bij de verschillende overgangen:

```

095E 01 65 3A 01 00 00 00          ; #3A65 is start of table data for screens/levels

```

Waarin de eerste 01 gekopieerd wordt naar #6229 (is level #)

En de 653A gekopieerd wordt naar #622A (is pointer in level/screen table)

Als starten L=04 dan dit aanpassen naar 04 en 6E 3A. Getest overgang 04 / 05: **blauw erbij**.
Als starten L=11 dan dit aanpassen naar 0B en 73 3A. Getest overgang 11 / 12: **rood elevators erbij**.
Als starten L=14 dan dit aanpassen naar 0E en 73 3A. Getest overgang 14 / 15: **rood rivets erbij**.

Alle overgangen gaan goed.

Nu terugzetten naar level 01 als start.
En invincibility aanzetten.

19B3 CD0828 CALL #2808 ; check for collisions with hostile sprites

Deze versie is de finale versie.

Kleine test met herintroduceren kleine ladder beneden:

De donkergroene groep wordt dan:

3B4D: 00 33 BA 33 D2 AA ; short ladder at left side above oil can
3B53: 01 53 92 53 B8 AA ; second broken ladder from bottom, on 3rd girder
3B59: 01 BB 70 BB 98 AA ; third broken ladder on right side near top
3B5F: ~~01 63 D5 63 F8 AA ; bottom broken ladder~~

De volgende call regelt de groene groep. NOPS maken:

3F56: CDE03F CALL #3FE0 ; call additional code to add stolen ladders – label **DDDD**

De volgende call regelt de blauwe groep. NOPS maken:

0B7B: CDE03F CALL #3FE0 ; call additional code to add stolen ladders – label **GGGG**

De groene groep definiëren en weer aanzetten. Dat werkt.

De donkergroene groep definiëren en aanzetten:

3F31: 21353B LD HL,#3B35 ; load HL with **green group of girders**
3F34: DD210763 LD IX,#6307 ; load IX with index for green group of girders
3F38: C36A0C JP #0C6A ; jump to additional code – jump to label **AAAA**

en:

0C6A: CDE03F CALL #3FE0 ; call additional code to add stolen ladders – label **AAAA**
0C6D: 0E05 LD C, #05 ; don't make ladder stolen
0C6F: 0603 LD B, #03 ; loop counter, process three ladders
0C71: CDE23F CALL #3FE2 ; call additional code to add stolen ladders
0C74: C3593F JP #3F59 ; start processing of blue group

Gaat inderdaad meteen fout: blijft hangen.

Het is echter werkend te krijgen door de volgende aanpassingen:

```
244C FD211063 LD IY,#6310 ;
```

Aanpassen naar:

```
244C FD210F63 LD IY,#630F ;
```

En

```
1B23 3E04 LD A,#04 ; A := 4
1B25 B9 CP C ; compare. is the ladder broken?
```

Aanpassen naar:

```
1B23 3E04 LD A,#05 ; A := 5
1B25 B9 CP C ; compare. is the ladder broken?
```

Dit werkt alleen niet voor Pies. Je bent grens fixed/broken aan het aanpassen, maar waarschijnlijk door het grote aantal ladders in Pies, zijn de retractable ladders hierdoor fixed geworden.

Maar wat als deze twee aanpassingen alleen gedaan worden wanneer barrels?

```
244C FD211063 LD IY,#6310 ;
2450 A7 AND A ; A == 0 ? checksum OK ?
2451 CA5624 JP Z,#2456 ; yes, skip next step

2454 FD23 INC IY ; running this step will break the game ? loops at #2371 forever

2456 3A2762 LD A,(#6227) ; load A with screen number
2459 3D DEC A ; is this the girders?
245A 21E43A LD HL,#3AE4 ; load HL with start of table data for girders
245D CA7124 JP Z,#2471 ; if girders, skip ahead
```

```
245D: CA2B0C JP Z,#0C2B ; if girders, go to additional code – jump to label AAAA
```

en:

```
0C2B: FD210F63 LD IY, #630F ; load alternative IY for barrels – label AAAA
0C2F: C37124 JP #2471 ; jump back
```

1B16	F5	PUSH	AF	; save AF for later
1B17	210762	LD	HL,#6207	; load HL with movement indicator
1B1A	7E	LD	A,(HL)	; load movement
1B1B	E680	AND	#80	; mask bits
1B1D	F606	OR	#06	; mask bits
1B1F	77	LD	(HL),A	; store movement
1B20	211A62	LD	HL,#621A	; load HL with ladder type address
1B23	3E04	LD	A,#04	; A := 4
1B25	B9	CP	C	; compare. is the ladder broken?
1B26	3601	LD	(HL),#01	; store 1 into ladder type = broken ladder by default
1B28	D22C1B	JP	NC,#1B2C	; if ladder broken, skip next step
1B2B	35	DEC	(HL)	; set indicator to unbroken ladder
1B2C	F1	POP	AF	; restore AF
1B2D	A7	AND	A	; A == 0 ? is mario at bottom of ladder?
1B2E	CA4E1B	JP	Z,#1B4E	; yes, skip ahead

1B20 C3810C JP #0C81 ; jump to additional code – jump to label **AAAA**

en:

0C81:	211A62	LD	HL,#621A	; load HL with ladder type address – label AAAA
0C84:	3A2762	LD	A,(#6227)	; load A with screen number
0C87:	3D	DEC	A	; decrease by 1
0C88:	C2231B	JP	NZ, #1B23	; if not zero, not on girders – jump back
;				
0C8B:	3E05	LD	A, #05	; A := 5
0C8D:	C3251B	JP	#1B25	; jump back

Werkt goed.

Nu proberen met de volgende groepen voor barrels:



Definitie van de groene ladders (index met 7 ladders: 6307):

3B35: 00 33 BA 33 D2 AA ; short ladder at left side above oil can
3B3B: 00 73 B6 73 D6 AA ; longer ladder to left of bottom hammer
3B41: 00 33 78 33 90 AA ; short ladder at left side under top hammer
3B47: 00 5B 76 5B 92 AA ; longer ladder under the top left hammer

Definitie van de blauwe ladders (index met 11 ladders: 630B):

3B4D: 00 83 95 83 B5 AA ; center longer ladder
3B47: 01 63 D5 63 F8 AA ; bottom broken ladder
3B4D: 01 53 92 53 B8 AA ; second broken ladder from bottom, on 3rd girder
3B53: 01 BB 70 BB 98 AA ; third broken ladder on right side near top

Groene groep weer herstellen, naar originele zonder de donkergroene groep:

Dat was het volgende:

3F31: 21353B LD HL,#3B35 ; load HL with green group of girders
3F34: DD210763 LD IX,#6307 ; load IX with index for green group of girders
3F38: C3563F JP #3F56 ; skip next steps – jump to label DDDD

Code voor de groene en donkergroene groep komt weer vrij: 0C6A t/m 0C76

Code voor de groene en donkergroene groep:

3F31: 21353B LD HL,#3B35 ; load HL with green group of girders
3F34: DD210763 LD IX,#6307 ; load IX with index for green group of girders
3F38: C36A0C JP #0C6A ; jump to additional code – jump to label AAAA

en:

0C6A: CDE03F CALL #3FE0 ; call additional code to add stolen ladders – label AAAA
0C6D: 0E05 LD C, #05 ; don't make ladder stolen
0C6F: 0603 LD B, #03 ; loop counter, process three ladders
0C71: CDE23F CALL #3FE2 ; call additional code to add stolen ladders
0C74: C3593F JP #3F59 ; start processing of blue group

De index voor de blauwe groep voor rivets aanpassen:

0B60: 214D3B LD HL,#3B4D ; load HL with blue group of girders – label YYYY
0B63: DD210B63 LD IX,#630B ; load IX with index for blue group of girders
0B67: C37B0B JP #0B7B ; skip next steps – jump to label GGGG

De blauwe groep weer activeren:

3F80: C3600B JP #0B60 ; jump to additional code – jump to label YYYY

De volgende call regelt de blauwe groep. NOPS maken:

0B7B: CDE03F CALL #3FE0 ; call additional code to add stolen ladders – label GGGG

Dat werkt nu allemaal weer. Eventjes backup maken. Maar gaat hangen omdat er nu 6 gebroken ladders zijn, dus de grens moet nog opgeschoven worden.

Grens opschuiven: twee aanpassingen:

0C2B: FD210E63 LD IY, #630E ; load alternative IY for barrels – label AAAA

...

0C8B: 3E06 LD A, #06 ; A := 6

Ja dat werkt ook goed.

Nu nog maken dat de blauwe groep altijd die trap broken maakt. Dat betekent een C waarde van 1.

Stuk code voor blauwe groep girders:

0B60: 214D3B LD HL, #3B4D ; load HL with blue group of girders – label YYYY

0B63: DD210B63 LD IX, #630B ; load IX with index for blue group of girders

0B67: C37B0B JP #0B7B ; skip next steps – jump to label GGGG

Aanpassen naar:

0B67: C36A0C JP #0C6A ; jump to additional code – jump to label AAAA

0C6A: 79 LD A, C ; a := c – label AAAA

0C6B: FE05 CP #05 ; is c equal to 5?

0C6D: 2802 JR Z, 0C6F ; yes, skip next step

0C6F: 0E04 LD C, #04 ; c := 04

0C71: C37B0B JP #0B7B ; skip next steps – jump to label GGGG

Nu verschillende groepen testen bij de verschillende overgangen:

095E 01 65 3A 01 00 00 00 ; #3A65 is start of table data for screens/levels

Waarin de eerste 01 gekopieerd wordt naar #6229 (is level #)

En de 653A gekopieerd wordt naar #622A (is pointer in level/screen table)

Als starten L=04 dan dit aanpassen naar 04 en 6E 3A. Getest overgang 04 / 05: blauw erbij.

Als starten L=11 dan dit aanpassen naar 0B en 73 3A. Getest overgang 11 / 12: rood elevators erbij.

Als starten L=14 dan dit aanpassen naar 0E en 73 3A. Getest overgang 14 / 15: rood rivets erbij.

Alle overgangen gaan goed.

Nu terugzetten naar level 01 als start.
En invincibility aanzetten.

19B3 CD0828 CALL #2808 ; check for collisions with hostile sprites

Deze versie is de finale versie.

Kijken of de blauwe groep uitgebreid kan worden met de ladder links ervan.

Definitie van de blauwe ladders (index met 11 ladders: 630B):

3B4D: 00 83 95 83 B5 AA ; center longer ladder

3B53: 01 63 D5 63 F8 AA ; bottom broken ladder

3B59: 01 53 92 53 B8 AA ; second broken ladder from bottom, on 3rd girder

3B5F: 01 BB 70 BB 98 AA ; third broken ladder on right side near top

Broken maken 1 of 2, maar vanaf de 4 beginnen, dus juist de laatste twee blauw maken:

3B4D: 01 63 D5 63 F8 AA ; bottom broken ladder

3B53: 01 BB 70 BB 98 AA ; third broken ladder on right side near top

3B59: 00 83 95 83 B5 AA ; center longer ladder

3B5F: 00 53 92 53 B8 AA ; second broken ladder from bottom, on 3rd girder

0B67: C36A0C JP #0C6A ; jump to additonal code – jump to label AAAA

0C6A: 79 LD A, C ; a := c – label AAAA

0C6B: FE05 CP #05 ; is c equal to 5?

0C6D: 2804 JR Z, 0C73 ; yes, skip next steps

0C6F: E601 AND #01 ; a is number between 0 and 1

0C71: 3C INC A ; increment a; a is number between 1 and 2

0C72: 4F LD C,A ; put back in c

0C73: C37B0B JP #0B7B ; skip next steps – jump to label GGGG

Nee, gaat niet goed. Dat was een ladder die niet fixed gemaakt kon worden. Dan blijft jumpman doorklimmen.

Op te lossen door de ladder één plek naar links te verplaatsen (en de uiteinden ook goed te zetten):

3B5F: 00 4B 91 4B B9 AA ; second broken ladder from bottom, on 3rd girder

Ander alternatief is wellicht de kleine trap rechts aan de blauwe groep toe te voegen, maar dan moet deze wel gebroken getekend kunnen worden. Daartoe moet de code hiervoor uitgebreid worden:

0C40: 3A8E63 LD A, (#638E) ; load A with broken ladder override – label AAAA

0C43: FE01 CP #01 ; == 1 ? (is override set?)

0C45: CA4E0C JP Z, #0C4E ; yes force broken ladder – jump to label BBBB

0C48: 3AB363 LD A, (#63B3) ; load A with original data item

```

0C4B: C3360E    JP    #0E36        ; jump back
;
0C4E: 3A2762    LD    A,(#6227)    ; load a with screen number – label BBBB
0C51: 3D         DEC    A        ; is this the barrels?
0C52: C23B0E    JP    NZ,#0E3B     ; no, draw normal broken ladder
0C55: D5         PUSH   DE        ; store DE
0C56: 1B         DEC    DE        ; decrement DE, DE points to data-item with x-position
0C57: 1A         LD     A, (DE)    ; A = x-position ladder
0C58: D1         POP    DE        ; restore DE
0C59: FE33       CP     #33       ; check if short ladder at left side
0C5B: CA3F0E    JP    Z, #0E3F     ; yes, draw short broken ladder
0C5E: C33B0E    JP    #0E3B     ; no, draw normal broken ladder

```

Benodigde aanpassing is dan:

```

0C5E: C3320C    JP    #C30C32    ; jump to additional code

```

en:

```

0C32: FECB       CP     #CB        ; check if short ladder at right side
0C34: CA3F0E    JP    Z, #0E3F     ; yes, draw short broken ladder
0C37: C33B0E    JP    #0E3B     ; no, draw normal broken ladder

```

De groepindeling van blauw wordt dan:

Broken maken 1 of 2, maar vanaf de 4 beginnen, dus juist de laatste twee blauw maken:

```

3B4D: 01 63 D5 63 F8 AA ; bottom broken ladder
3B53: 01 BB 70 BB 98 AA ; third broken ladder on right side near top
3B59: 00 83 95 83 B5 AA ; center longer ladder
3B5F: 00 CB 99 CB B1 AA ; short ladder at center right

```

En in de definitie van het barrels board:

```

3B16: 01 53 92 53 B8 ; second broken ladder from bottom, on 3rd girder

```

Dat werkt heel erg goed.

Nu verschillende groepen testen bij de verschillende overgangen:

```

095E 01 65 3A 01 00 00 00 ; #3A65 is start of table data for screens/levels

```

Waarin de eerste 01 gekopieerd wordt naar #6229 (is level #)

En de 653A gekopieerd wordt naar #622A (is pointer in level/screen table)

Als starten L=04 dan dit aanpassen naar 04 en 6E 3A. Getest overgang 04 / 05: blauw erbij.

Als starten L=11 dan dit aanpassen naar 0B en 73 3A. Getest overgang 11 / 12: rood elevators erbij.

Als starten L=14 dan dit aanpassen naar 0E en 73 3A. Getest overgang 14 / 15: rood rivets erbij.

Alle overgangen gaan goed.

Nu terugzetten naar level 01 als start.

En invincibility aanzetten.

19B3 CD0828 CALL #2808 ; check for collisions with hostile sprites

Deze versie is de finale versie.

En dan nu nogmaals aan een nieuwe optie werken:

3B4D: 01 63 D5 63 F8 AA ; bottom broken ladder

3B53: 00 CB 99 CB B1 AA ; short ladder at center right

3B59: 00 83 95 83 B5 AA ; center longer ladder

3B5F: 00 4B 91 4B B9 AA ; second broken ladder from bottom, on 3rd girder

En in de definitie van het barrels board:

3B16: 01 BB 70 BB 98 ; third broken ladder on right side near top

Code aanpassen die de broken ladder bepaalt:

0B67: C36A0C JP #0C6A ; jump to additional code – jump to label AAAA

Alle vier processen, maar broken maken mag alleen 1,2 en 3:

0C6A: 79 LD A, C ; a := c – label AAAA

0C6B: FE04 CP #04 ; is c equal to 4?

0C6D: 2002 JR NZ, 0C71 ; no, skip next steps

0C6F: 3D DEC A ; decrement a, 4 becomes 3

0C70: 4F LD C,A ; put back in c

0C71: C3 7B0B JP #0B7B ; skip next steps – jump to label GGGG

Dat werkt nu.

Nu verschillende groepen testen bij de verschillende overgangen:

095E 01 65 3A 01 00 00 00 ; #3A65 is start of table data for screens/levels

Waarin de eerste 01 gekopieerd wordt naar #6229 (is level #)

En de 653A gekopieerd wordt naar #622A (is pointer in level/screen table)

Als starten L=04 dan dit aanpassen naar 04 en 6E 3A. Getest overgang 04 / 05: blauw erbij.

Als starten L=11 dan dit aanpassen naar 0B en 73 3A. Getest overgang 11 / 12: rood elevators erbij.

Als starten L=14 dan dit aanpassen naar 0E en 73 3A. Getest overgang 14 / 15: rood rivets erbij.

Alle overgangen gaan goed.

Nu terugzetten naar level 01 als start.
En invincibility aanzetten.

19B3 CD0828 CALL #2808 ; check for collisions with hostile sprites

Deze versie is de finale versie v1.02.