

Aanzetten onsterfelijkheid:

```
19B0 CDFB03 CALL #03FB ; check for kong beating chest and animate girl and her screams  
19B3 CD0828 CALL #2808 ; check for collisions with hostile sprites [set to NOPS to make mario invincible to enemy sprites]  
19B6 CD1D28 CALL #281D ; do stuff for hammers
```

Aanpassen van 19B3 naar 3x NOP:

19B3 000000 NOP, NOP, NOP

De girders aan de rechterkant korter maken:

Beginnen met de onderste girder:

3B07: 02 DF F2 80 F7 ; bottom slanted girder

Dit maakt de onderste girder wel korter, maar aan de rechterkant kan Jumpman eraf vallen. Valt dan beneden uit scherm en komt boven tevoorschijn. Gaat dan dood, maar doordat hij bovenin verschijnt is de Y-coordinaat kleiner dan het platform met Pauline en wordt dit gezien als level cleared. Dat is niet de bedoeling. Dus de onderste maar origineel laten:

3B07: 02 FF F0 80 F7 ; bottom slanted girder

Dan de volgende girder korter maken:

3B02: 02 BF DA 10 D0 ; 5th slanted girder (has hammer at right side)

En de volgende girder:

3AFD: 02 CF B1 20 BB ; 4th slanted girder

En de volgende girder:

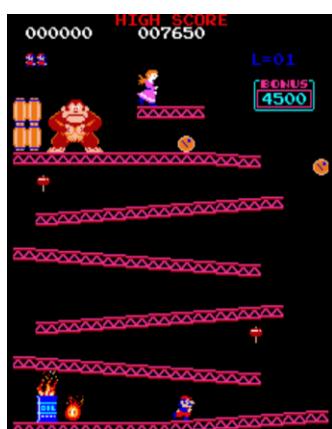
3AF8: 02 BF 98 10 8E ; 3rd slanted girder

En de volgende girder:

3AF3: 02 CF 6F 20 79 ; 2nd slanted girder (has hammer at left side)

En de laatste girder:

3AED: 02 BF 56 A0 55 ; 1st slanted girder at top right



De barrels vliegen nu rechts helemaal uit beeld naar beneden en verdwijnen.

Nu kijken of de 'rand' die de barrels zien naar links verplaatst kan worden. Dit wordt hier geregeld:

```
2017 DD7E03 LD A,(IX+#03) ; Load A with Barrels' X position  
201A FE1C CP #1C ; have we arrived at left edge of girder?  
201C DA2F20 JP C,#202F ; yes, jump ahead to handle  
  
201F FEE4 CP #E4 ; else , have we arrived at right edge of girder?  
2021 DABA21 JP C,#21BA ; no, jump way ahead - we're done, store values and try next barrel
```

E4 is de x-coördinaat van de rand, voor girders die eindigen op DF. De nieuwe girders eindigen nu op BF (#DF-#BF = #20). Dus 'rand' zou moeten worden #E4-#20 = #C4.

Dus aanpassing wordt dan:

```
201F FEC4 CP #C4 ; else , have we arrived at right edge of girder?
```

En dat werkt uitstekend. De barrels rollen netjes naar beneden. Op deze manier is er een smaller barrels structuur gemaakt.



Nu de ladders weer terugzetten en de rechter ladders naar links verplaatsen.



De rechter kleine trappen worden dan:

```
3B11: 00 AB 55 AB 71 ; short ladder at top right  
3B16: 00 AB 97 AB B3 ; short ladder at center right  
3B1B: 00 AB D9 AB F5 ; short ladder at bottom right
```



Dit werkt allemaal goed. Alleen de broken ladder twee girders boven de rechter hamer moet nog opgeschoven worden:

3B52: 01 9B 72 9B 96 ; third broken ladder on right side near top



Dit werkt allemaal goed.

Nu additionele girders maken aan de rechterkant, met trappetjes ertussen. Dit gaat voor nu eventjes door de definitie van het conveyors screen heen.

Let op de girders moeten ook schuin zijn, omdat jumpman anders door de girders zakt.

3B5C: 02 FE D2 D0 D1 ; onderste platform

3B61: 02 FE A1 E0 A1 ; tweede platform

3B66: 00 E3 A1 E3 D1



Een tweede stuk maken:

3B6B: 02 FE 6F E0 6F ; onderste platform
3B70: 02 FE 4F E0 4F ; tweede platform
3B75: 00 E3 4F E3 6F



Nu nog een ladder tussen de beide rechter delen:

3B7A: 00 EB 6F EB A1

Dat gaat niet goed. De ladder werkt niet. Komt dat door geen veelvoud van 8?

Aanpassingen levert de volgende structuur rechts op:



3B5C: 02 FE D2 D0 D1 ; Onderste platform
3B61: 02 FE A1 E0 A1 ; Tweede platform
3B66: 00 E3 A1 E3 D1 ; Ladder tussen 1 en 2
3B6B: 02 FE 71 E0 71 ; Derde platform
3B70: 02 FE 51 E0 51 ; Vierde platform
3B75: 00 E3 51 E3 71 ; Ladder tussen 3 en 4
3B7A: 00 EB 71 EB A1 ; Ladder tussen 2 en 3

Nu een firefox starten op structuur rechts. Firefox aanmaken kan met:

```
10E9 DD210064 LD    IX,#6400      ; load IX with start of fire #1
10ED DD360001 LD    (IX+#00),#01 ; set fire active
10F1 DD360358 LD    (IX+#03),#58 ; set fire X position
10F5 DD360E58 LD    (IX+#0E),#58 ; set fire X position #2
10F9 DD360580 LD    (IX+#05),#80 ; set fire Y position
10FD DD360F80 LD    (IX+#0F),#80 ; set fire Y position #2
```

Dus dit moet ergens tussengevoegd worden met de fire op de juiste positie. Eerst maar eens op de onderste girder.

Proberen tussen te voegen na het plaatsen van de hamers? Dat gebeurd hier:

```

1000 210C3E LD HL,#3E0C ; load HL with table data for hammers on girders
1003 CDA611 CALL #11A6 ; ???

```

Dit testen door 1003 aan te passen naar 3xNOP: dan worden de hamers niet getekend. Dat klopt inderdaad. De code voor start fire invoegen in ongebruikte stuk achteraan:

```

3F30: 50 52 4F 47 52 41 4D 2C 57 45 20 57 4F 55 4C 44 PROGRAM,WE WOULD
3F40: 20 54 45 41 43 48 20 59 4F 55 2E 2A 2A 2A 2A 2A TEACH YOU.*****
3F50: 54 45 4C 2E 54 4F 4B 59 4F 2D 4A 41 50 41 4E 20 TEL.TOKYO-JAPAN
3F60: 30 34 34 28 32 34 34 29 32 31 35 31 20 20 20 044(244)2151
3F70: 45 58 54 45 4E 54 49 4F 4E 20 33 30 34 20 20 20 EXTENTION 304
3F80: 53 59 53 54 45 4D 20 44 45 53 49 47 4E 20 20 20 SYSTEM DESIGN
3F90: 49 4B 45 47 41 4D 49 20 43 4F 2E 20 4C 49 4D 2E IKEGAMI CO. LIM.

```

Firefox plaatsen op plaats: #D8, #C9.

Dus vanaf 3F30 het volgende invoegen:

```

3F30 CDA611 CALL #11A6 ; draw hammers
3F33 DD210064 LD IX,#6400 ; load IX with start of fire #1
3F37 DD360001 LD (IX+#00),#01 ; set fire active
3F3B DD3603D8 LD (IX+#03),#D8 ; set fire X position
3F3F DD360ED8 LD (IX+#0E),#D8 ; set fire X position #2
3F43 DD3605C8 LD (IX+#05),#C8 ; set fire Y position
3F47 DD360FC8 LD (IX+#0F),#C8 ; set fire Y position #2
3F4B C9 RET ; return

```

En dan moet de aanroep hiernaar toe ook aangepast worden:

```
1003 CD303F CALL #3F30 ; draw hammers and firefox
```



Werkt goed. Alleen als de firefox helemaal boven is, gaat deze niet meer naar beneden. Dat is hetzelfde als in de conveyors screen.

Dit oplossen door het bovenste platform te verlagen en de ladder daarop aan te passen:

```

3B6B: 02 FE 71 E0 71 ; Derde platform
3B70: 02 FE 61 E0 61 ; Vierde platform
3B75: 00 E3 61 E3 71 ; Ladder tussen 3 en 4

```

Dat helpt wel: gaat nu wel naar beneden. Firefox daarboven is erg lastig. Daarom weer terug naar oorspronkelijke platform. Maar nu de ladder tussen 2 en 3 weghalen.

```

3B6B: 02 FE 71 E0 71 ; Derde platform
3B70: 02 FE 51 E0 51 ; Vierde platform
3B75: 00 E3 51 E3 71 ; Ladder tussen 3 en 4

```



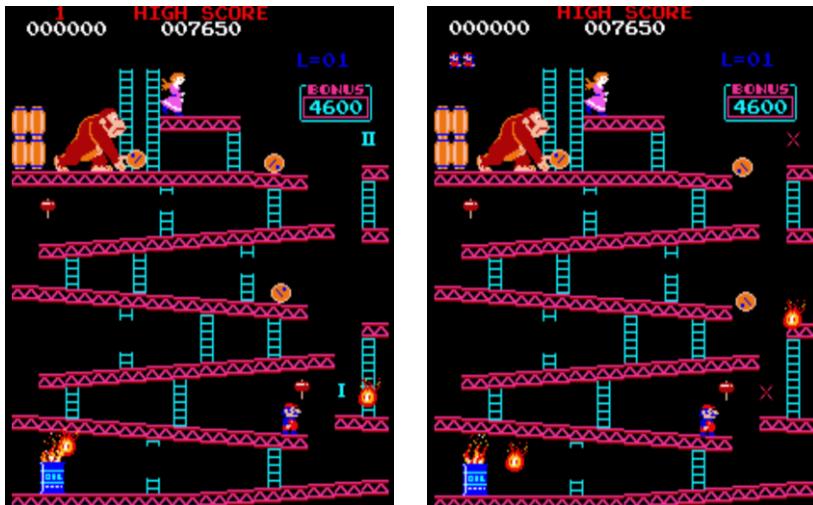
Markeren plaatsen switch. Een karakter in een video geheugenplaats neerzetten:

De voorgaande return vervangen door de onderstaande code:

```

3F4B 21B874 LD HL,#74B8 ; video memory address to clear
3F4E 3E32 LD A,#32 ; load a with I character
3F50 77 LD (HL),A ; clear memory address
3F51 21A174 LD HL,#74A1 ; video memory address to clear
3F54 3E33 LD A,#33 ; load a with II character
3F56 77 LD (HL),A ; clear memory address
3F57 C9 RET ; return
    
```

Of allebei met een rood kruis: is character #FE !



Vooralsnog vind ik de kruisen mooier.

Kleine test doen of jumpman door kruis in girder heen valt.

Toevoegen:

```

3F57 219776 LD HL,#7697 ; video memory address to clear
3F5A 3EFE LD A,#FE ; load a with II character
3F5C 77 LD (HL),A ; clear memory address
3F5D C9 RET ; return
    
```

Gaat goed. Jumpman valt daar inderdaad doorheen.

Welk stuk van de girders zo maken? Ook afhankelijk van de y-coordinaat, omdat de girder soms verschoven zit: niet precies de karakterposities. Aantal geprobeerd. Werkt niet goed.

Toch kijken of je op een of andere manier de trappen erbij kan maken. Maar dan moet je weten hoeveel je er al hebt en dan eentje toevoegen. Je moet de index in de gebruikte tabel voor trappen weten. Deze start op #6300, maar tot hoever gevuld? Roeft ook de vraag op of er een maximum zit aan het aantal trappen.

Alsnog een trap tekenen:

```
3F57 115E3F LD DE,#3F5E ; load start table data  
3F5A CDA70D CALL #0DA7 ; draw screen  
3F5D C9 RET ; return  
3F5E 06 8F 90 70 90 AA ; test table data patch of X characters
```

En dat werkt inderdaad:



Nu kijken of je er een trap mee kun tekenen. In principe de trap naar de bovenste grider laten toevoegen. Eerst deze in de definitie van het veld weghalen.

Is deze trap:

```
3B11: 00 AB 55 AB 71 ; short ladder at top right
```

Dus die ertussen uit en dan alle opschuiven. Levert het volgende resultaat:



En dan de XXXX vervangen door het tekenen van de verwijderde ladder:

```
3F5E 00 AB 55 AB 71 ; test table data removed ladder
```

Gaat goed. Tekent de trap op de juiste plaats. Alleen je kan er nog niet op omdat de trap niet herkend wordt. Dat alsnog voor elkaar krijgen.

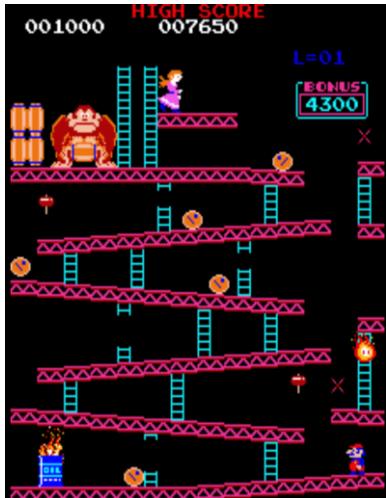
Omdat we met de trappen indexen moeten gaan spelen, is het belangrijk dat de andere trap: die naar Pauline ook alvast verwijderen.

Is deze trap:

3B4D: 00 93 38 93 54 ; ladder leading to girl

Zit nu door eerdere verschuiven op 3B48.

Dus die ertussen uit en dan alle oorschuiven. Levert het volgende resultaat:



De trap weer activeren kan eventueel door de volgende code:

```
21XXXX LD HL,#XXXX ; load HL with start of table data for ladder  
DD210B63 LD IX,#630B ; #6300 is start, with 11 ladders added: 630B  
CD7524 CALL #2475
```

Springt in de volgende code: die door de tabel heen loopt en de ladder aan de tabel toevoegt.

```
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
```

Eerst moet de tabel definitie met de ladder verplaatst worden, zodat dit stukje code aan de routine toegevoegd kan worden.

Stuk van conveyors functionaliteit gebruiken voor de nieuwe functionaliteit. Het volgende stuk gaat over de afhandeling van de taarten.: #24EA – 25EF.

Functionaliteit voor het tekenen van de ladder vanaf #24EA:

```

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: C9 RET ; return

```

En dan dit als test aanroepen vanaf de 3F57:

```

3F57 CDF024 CALL #24F0 ; draw ladder
3F5A C9 RET ; return

```

Dit werkt goed. Tekent de trap inderdaad zonder dat je erop kan klimmen. Nu de code voor het activeren van de trap aan de trap routine toevoegen:

```

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 inderdaad, dit werkt helemaal goed. De trap kan nu opgelopen worden. Dus met deze code wordt de trap juist geactiveerd. Nu de code voor de tweede trap maken. Kan er meteen achteraan.

```

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

```

En dan dit als test aanroepen vanaf de 3F5A:

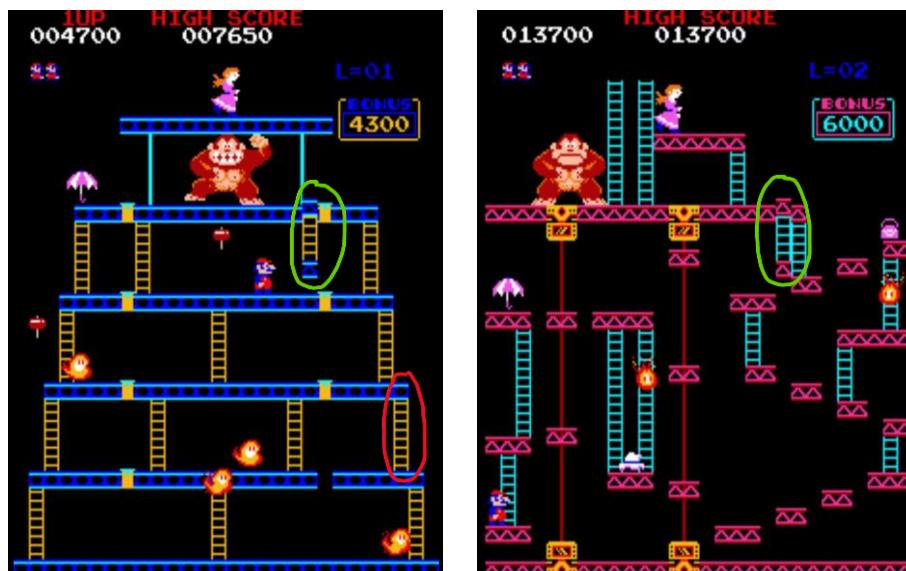
```

3F5A CD0725 CALL #2507 ; draw ladder
3F5D C9 RET ; return

```

Dus hiermee is de code vanaf 3F57 nu de code om de twee trappen als test te tekenen. De code daarvoor (met 3F57 gevuld met C9 = RET) de code om in het scherm de twee kruisen te tekenen. Dit moet later hiernaar teruggezet worden.

Lijkt allemaal in eerste scherm goed te gaan, maar gaat in latere schermen niet goed. In de rivets wordt een extra trap getekend. Daar kan je wel op lopen. De trap rechts op het tweede niveau kan je echter niet gebruiken. De barrels zijn daarna weer gewoon goed te spelen. De elevators komt daarna. De trap links onderin kan niet beklimmen worden, het spel is heel traag, de elevators staan stil en na een keer springen wordt je teruggezet naar het score scherm.



Hoe komt dit. Wordt de eerste trap nu ook getekend in de andere velden? De aanroep naar de twee nieuwe functies wordt niet gedaan lijkt me? Of wordt de code (beginnend op #24EA) nog ergens anders vandaan aangeroepen? Ja dus. In de algemene routine wordt 24EA altijd aangeroepen:

```
; arrive here from #0701 when playing

197A CDBD1D CALL #1DBD ; check for bonus items and jumping scores, rivets
197D CD8C1E CALL #1E8C ; do stuff for items hit with hammer
1980 CDC31A CALL #1AC3 ; check for jumping
1983 CD721F CALL #1F72 ; roll barrels
1986 CD8F2C CALL #2C8F ; roll barrels ?
1989 CD032C CALL #2C03 ; do barrel deployment ?
198C CDED30 CALL #30ED ; update fires if needed
198F CD042E CALL #2E04 ; update bouncers if on elevators
1992 CDEA24 CALL #24EA ; do stuff for pie factory
1995 CDBB2D CALL #2DDB ; deploy fireball/firefoxes for conveyors and rivets
1998 CDD42E CALL #2ED4 ; do stuff for hammer
199B CD0722 CALL #2207 ; do stuff for conveyors
199E CD331A CALL #1A33 ; check for and handle running over rivets
```

Ik hoop dat je dat kan voorkomen door #1992 in drie NOPS te veranderen.

1992: 000000 NOP NOP NOP

Ja dat heeft het probleem opgelost. Alle drie de levels werken nu weer zoals het zou moeten.

Nu stuk code schrijven die elke keer uitgevoerd wordt en die test of de 'switch' geraakt wordt en dan de trap tekent.

Vastleggen status van switch in variabele #6398. Deze wordt oorspronkelijk gebruikt als pie deployment counter.

Bit 1 = switch 1 geactiveerd
Bit 2 = switch 2 geactiveerd

Aan het einde van het tekenen van het level het nu ter test tekenen van de trappen weer verwijderen en deze variabele op 0 initialiseren.

```
3F57 CDF024 CALL #24F0 ; draw ladder
3F5A CD0725 CALL #2507 ; draw ladder
3F5D C9 RET ; return
```

Vervangen door:

```
3F57: 3E00 LD A,#00 ; A := #00
3F59: 329B63 LD (#639B),A ; store into switch status variable
3F5C: C9 RET
```

Aangepast. De trappen worden nu niet meer getekend. Nu dan toevoegen code.

Dat kan vanaf #2518

```
2518: 3A9B63 LD A,(#639B) ; load switch status variable in A
251B: E601 AND #01 ; check bit 1 set
251D: C23325 JP NZ,#2533 ; jump to next switch if switch already activated
2520: 3A0362 LD A,(#6203) ; load A with jumpman's x-position
2523: FED8 CP #D8 ; check x-position = #D8
2525: C23325 JP NZ,#2533 ; jump to next switch if x-position is not correct
2528: 3A0562 LD A,(#6205) ; load A with jumpman's y-position
252B: FED1 CP #D1 ; check y-position = #D0
252D: C23325 JP NZ,#2533 ; jump to next switch if y-position is not correct
2530: CDF024 CALL #24F0 ; draw lower ladder
2533: C9
```

Dit moet elke keer aangeroepen worden. Dat kan in de verwijderde call naar de afhandeling van conveyors (waren 3x NOP gemaakt). Nu wordt dat dan:

1992: CD1825 CALL #2518

Werkt niet meteen. Waarschijnlijk y-coordinaat niet goed. Daarom deze test eventjes uitzetten door de code door NOP te vervangen:

252B: NOP NOP NOP NOP NOP

Dat werkt goed. Dus nu nog de juiste Y-waarde vinden. Proefondervindelijk terugschroeven vanaf #D0 terug. Eerder geplaatste firefox op die girder begint met y-coordinaat #C8. Dit eerst proberen.

252B: FEC8 CP #C8 ; check y-position = #C8
252D: C23325 JP NZ,#2533 ; jump to next switch if y-position is not correct

Dat werkt nu goed.

Na het tekenen van de ladder de switch status aanpassen. En dan een geluid toevoegen zodat we kunnen checken of de switch inderdaad maar één keer geactiveerd wordt.

2530: CDF024 CALL #24F0 ; draw lower ladder
2533: 3A9B63 LD A,(#639B) ; load switch status variable in A
2536: F601 OR #01 ; set bit 1
2538: 329B63 LD (#639B),A ; write back to switch status variable
253B: 3E03 LD A,#03 ; load sound duration of 3
253D: 328460 LD (#6084),A ; play sound for falling bouncer
2540: C9 RET

En de jumps aanpassen naar de nieuwe return: #2540.

Werkt goed. Geluid is echter hetzelfde als wanneer jumpman dood valt. Ander geluid kiezen? Ja 6085. Is geluid voor bonus item pick up.

Dan nog de switch ook weghalen. Dit kan door volgende code toe te voegen:

2540: 21B874 LD HL,#74B8 ; video memory address to clear
2543: 3E10 LD A,#10 ; load a with empty
2545: 77 LD (HL),A ; clear memory address
2546: C9 RET

En de jumps aanpassen naar de nieuwe return: #2546.

Werkt ook. Nu nog de andere ladder ook toevoegen. Dit kan met de volgende code:

2546: 3A9B63 LD A,(#639B) ; load switch status variable in A
2549: E602 AND #02 ; check bit 2 set
254B: C27425 JP NZ,#2574 ; jump to end if switch already activated
254E: 3A0362 LD A,(#6203) ; load A with jumpman's x-position
2551: FEE3 CP #E3 ; check x-position = #E3
2553: C27425 JP NZ,#2574 ; jump to end if x-position is not correct
2556: 3A0562 LD A,(#6205) ; load A with jumpman's y-position
2559: FE49 CP #49 ; check y-position = #49
255B: C27425 JP NZ,#2574 ; jump to end if y-position is not correct
255E: CD0725 CALL #2507 ; draw upper ladder

```

2561: 3A9B63 LD A,(#639B) ; load switch status variable in A
2564: F602 OR #02 ; set bit 2
2566: 329B63 LD (#639B),A ; write back to switch status variable
2569: 3E03 LD A,#03 ; load sound duration of 3
256B: 328560 LD (#6085),A ; play sound for falling bouncer
256E: 216874 LD HL,#7468 ; video memory address to clear
2571: 3E10 LD A,#10 ; load a with empty
2573: 77 LD (HL),A ; clear memory address
2574: C9 RET

```

Werkt toch nog niet helemaal. Zoals al eerder opgemerkt kan de rechter trap op het tweede niveau van de rivets niet beklimmen worden. Hoe kan dat? Daar heb ik nog niet naar gekeken. Wel naar de trap die toen nog verkeerd getekend werd, maar niet naar de trap die niet gebruikt kon worden. Nogmaals goed testen. In een volgende test kan ik weer die ladder niet op, maar wordt gedurende het spelen, de ladder verkeerd getekend.

De code wordt voor alle levels uitgevoerd. En dat moet niet. Er moet dus een test voor die checkt of het wel het barrels level is. En anders meteen return. Dit betekent dat alles opgeschoven moet worden en dat de relatieve spronges ook aangepast moeten worden. Is wel eventjes wat werk.

```

2518: 3A 27 62 LD A,(#6227) ; load A with screen number
251B: 3D DEC A ; is this the girders?
251C: C0 RET NZ ; if not return

251D: 3A 9B 63 LD A,(#639B) ; load switch status variable in A
2520: E6 01 AND #01 ; check bit 1 set
2522: C2 4B 25 JP NZ,#254B ; jump to next switch if switch already activated
2525: 3A 03 62 LD A,(#6203) ; load A with jumpman's x-position
2528: FE D8 CP #D8 ; check x-position = #D8
252A: C2 4B 25 JP NZ,#254B ; jump to next switch if x-position is not correct
252D: 3A 05 62 LD A,(#6205) ; load A with jumpman's y-position
2530: FE C8 CP #C8 ; check y-position = #C8
2532: C2 4B 25 JP NZ,#254B ; jump to next switch if y-position is not correct
2535: CD F0 24 CALL #24F0 ; draw lower ladder
2538: 3A 9B 63 LD A,(#639B) ; load switch status variable in A
253B: F6 01 OR #01 ; set bit 1
253D: 32 9B 63 LD (#639B),A ; write back to switch status variable
2540: 3E 03 LD A,#03 ; load sound duration of 3x
2542: 32 85 60 LD (#6085),A ; play sound for item pickup
2545: 21 B8 74 LD HL,#74B8 ; video memory address to clear
2548: 3E 10 LD A,#10 ; load a with empty
254A: 77 LD (HL),A ; clear memory address = remove switch
;
254B: 3A 9B 63 LD A,(#639B) ; load switch status variable in A
254E: E6 02 AND #02 ; check bit 2 set
2550: C2 79 25 JP NZ,#2579 ; jump to end if switch already activated
2553: 3A 03 62 LD A,(#6203) ; load A with jumpman's x-position
2556: FE E3 CP #E3 ; check x-position = #E3
2558: C2 79 25 JP NZ,#2579 ; jump to end if x-position is not correct
255B: 3A 05 62 LD A,(#6205) ; load A with jumpman's y-position
255E: FE 49 CP #49 ; check y-position = #49
2560: C2 79 25 JP NZ,#2579 ; jump to end if y-position is not correct
2563: CD 07 25 CALL #2507 ; draw upper ladder
2566: 3A 9B 63 LD A,(#639B) ; load switch status variable in A
2569: F6 02 OR #02 ; set bit 2
256B: 32 9B 63 LD (#639B),A ; write back to switch status variable
256E: 3E 03 LD A,#03 ; load sound duration of 3
2570: 32 85 60 LD (#6085),A ; play sound for item pickup
2573: 21 68 74 LD HL,#7468 ; video memory address to clear
2576: 3E 10 LD A,#10 ; load a with empty
2578: 77 LD (HL),A ; clear memory address = remove switch
2579: C9 RET

```

Nu werkt het wel weer goed in de rivets.

Aanpassen naar andere graphics: heart sprite tonen (eerst beneden en na activeren ervan boven) en bij het tekenen van de ladder ook een sprite laten knipperen.

Eerst de heart sprite tekenen in plaats van de kruisen:

Stuk is nu:

```
3F4B: 21B874 LD HL,#74B8 ; video memory address
3F4E: 3EFE LD A,#32 ; load a with red cross character
3F50: 77 LD (HL),A ; write memory address
3F51: 216874 LD HL,#7468 ; video memory address
3F54: 3EFE LD A,#FE ; load a with red cross character
3F56: 77 LD (HL),A ; write memory address
3F57: 3E00 LD A,#00 ; A := #00
3F59: 329B63 LD (#639B),A ; store into switch status variable
3F5C: C9 RET
```

Wordt nu dan:

```
3F4B: 21206A LD HL,#6A20 ; load HL with heart sprite
3F4E: 36D6 LD (HL),#D6 ; set heart sprite x-position
3F50: 23 INC HL ; next
3F51: 3676 LD (HL),#76 ; set heart sprite
3F53: 23 INC HL ; next
3F54: 3609 LD (HL),#09 ; set heart sprite color
3F56: 23 INC HL ; next
3F57: 36C8 LD HL,#C8 ; set heart sprite y-position
3F59: 3E00 LD A,#00 ; A := #00
3F5B: 329B63 LD (#639B),A ; store into switch status variable
3F5E: C9 RET
```

Positie heart sprite beneden is: (D6,C8)

Positie heart sprite boven is: (E3,48)

Sprite waarde 63 is de vonkjes die spatten. Gebruiken bij het tekenen van een ladder.

Testje met aanpassen van alleen de sprite vorm:

```
3F5E: 21216A LD HL,#6A21
3F61: 3663 LD (HL),#63 ; set sparkle sprite
3F63: C9 RET
```

Dat gaat goed, tekent een sparkle sprite in plaats van de heart sprite.

Nu nog testen met wissen van de sprite. Sprite waarde 64. Klopt geen heart sprite te zien.
Dus sparkle aan/uit is wisselen tussen waarde 63 en 64.

Weer terug naar de heart sprite alleen.

Dan nu de logica uitwerken voor de heart sprite eerst beneden en daarna de vonkjes laten knipperen. Dan heart sprite boven en nogmaals laten knipperen.

#639B is de switch variable.

#65A0 is de blink counter

```
2518: 3A 27 62 LD A,(#6227) ; load A with screen number
251B: 3D DEC A ; is this the girders?
251C: C0 RET NZ ; if not return
;
251D: 3A 9B 63 LD A,(#639B) ; load switch status variable in A
2520: E6 04 AND #04 ; check bit 3 set: blink sparkle
```

2522:	CA6925	JP Z,#2569	; if not set then do not blink
;			
2525:	21216A	LD HL,#6A21	; load HL with heart sprite
2528:	3AA065	LD A,(#65A0)	; load A with blink counter
252B:	3C	INC A	; increment blink counter
252C:	32A065	LD (#65A0), A	; store blink counter
;			
252F:	E616	AND #16	; check bit 5 set
2531:	C23925	JP NZ, #2539	; jump if not set
;			
2534:	3663	LD (HL),#63	; set sparkle sprite
2536:	C33B25	JP #253B	; skip next
;			
2539:	3664	LD (HL),#64	; set blank sprite
;			
253B:	3AA065	LD A,(#65A0)	; load A with blink counter
253E:	FE96	CP #96	; check max blinks reached
2540:	C0	RET NZ	; return if not ready blinking
;			
2541:	3A 9B 63	LD A,(#639B)	; load switch status variable in A
2544:	E602	AND #02	; check bit 2 set
2546:	C25D25	JP NZ, #255D	; jump if switch 2 activated
;			
2549:	CD F0 24	CALL #24F0	; draw lower ladder
254C:	21206A	LD HL,#6A20	; load HL with heart sprite
254F:	36E3	LD (HL),#E3	; set heart sprite x-position
2551:	23	INC HL	; next
2552:	3676	LD (HL),#76	; set heart sprite
2554:	23	INC HL	; next
2555:	3609	LD (HL),#09	; set heart sprite color
2557:	23	INC HL	; next
2558:	3648	LD HL,#48	; set heart sprite y-position
255A:	C36025	JP #2560	; skip next
;			
255D:	CD 07 25	CALL #2507	; draw upper ladder
;			
2560:	3A9B63	LD A,(#639B)	; load switch status variable in A
2563:	E6 FB	AND #FB	; reset bit 3 , stop blinking sparkle
2565:	32 9B 63	LD (#639B), A	; write back to switch status variable
2568:	C9	RET	; return
;			
2569:	3A 9B 63	LD A,(#639B)	; load switch status variable in A
256C:	E6 01	AND #01	; check bit 1 set
256E:	C2 90 25	JP NZ,#2590	; jump to next switch if switch already activated
2571:	3A 03 62	LD A,(#6203)	; load A with jumpman's x-position
2574:	FE D8	CP #D8	; check x-position = #D8
2576:	C0	RET NZ	; return if x-position is not correct
2577:	3A 05 62	LD A,(#6205)	; load A with jumpman's y-position
257A:	FE C8	CP #C8	; check y-position = #C8
257C:	C0	RET NZ	; return if y-position is not correct
257D:	3A 9B 63	LD A,(#639B)	; load switch status variable in A
2580:	F6 05	OR #05	; set bit 1 and bit 3, switch 1 activated, blink
2582:	32 9B 63	LD (#639B), A	; write back to switch status variable
2585:	3E 03	LD A,#03	; load sound duration of 3x
2587:	32 85 60	LD (#6085),A	; play sound for item pickup
258A:	3E 00	LD A,#00	; A:=0
258C:	32A065	LD (#65A0), A	; reset blink counter
258F:	C9	RET	; return
;			
2590:	3A 9B 63	LD A,(#639B)	; load switch status variable in A
2593:	E6 02	AND #02	; check bit 2 set
2595:	C0	RET NZ	; return if switch already activated
2596:	3A 03 62	LD A,(#6203)	; load A with jumpman's x-position

2599:	FE E3	CP #E3	; check x-position = #E3
259B:	C0	RET NZ	; return if x-position is not correct
259C:	3A 05 62	LD A,(#6205)	; load A with jumpman's y-position
259F:	FE 49	CP #49	; check y-position = #49
25A1:	C0	RET NZ	; return if y-position is not correct
25A2:	3A 9B 63	LD A,(#639B)	; load switch status variable in A
25A5:	F6 06	OR #06	; set bit 2 and 3, switch 2 activated, blink
25A7:	32 9B 63	LD (#639B), A	; write back to switch status variable
25AA:	3E 03	LD A,#03	; load sound duration of 3
25AC:	32 85 60	LD (#6085),A	; play sound for item pickup
25AF:	3E 00	LD A,#00	; A:=0
25B1:	32A065	LD (#65A0), A	; reset blink counter
25B4:	C9	RET	; return

Is bijna goed. Maar sparkle knippert nog te snel en de sparkle verhuist naar boven en blijft daar knipperen. De sparkle moet overigens gaan knipperen op de plek van de nieuwe ladder en niet op de plaats van het hartje. Daarnaast wordt de bovenste ladder vanzelf getekend.

Is nu aangepast. Logica gaat goed. Knippert nog wel op verkeerde plek en knippert raar snel. Knipperen anders doen: niet steeds sprite zetten, maar alleen als counter op een waarde staat. Dan kun je het knipperen ook reguleren.

2518:	3A 27 62	LD A,(#6227)	; load A with screen number
251B:	3D	DEC A	; is this the girders?
251C:	C0	RET NZ	; if not return
;			
251D:	3A 9B 63	LD A,(#639B)	; load switch status variable in A
2520:	E6 04	AND #04	; check bit 3 set: blink sparkle
2522:	CA6925	JP Z,#2569	; if not set then do not blink
;			
2525:	21216A	LD HL,#6A21	; load HL with heart sprite
2528:	3AA065	LD A,(#65A0)	; load A with blink counter
252B:	3C	INC A	; increment blink counter
252C:	32A065	LD (#65A0), A	; store blink counter
;			
252F:	E616	AND #16	; check bit 5 set
2531:	C23925	JP NZ, #2539	; jump if not set
;			
2534:	3663	LD (HL),#63	; set sparkle sprite
2536:	C33B25	JP #253B	; skip next
;			
2539:	3664	LD (HL),#64	; set blank sprite
;			
253B:	3AA065	LD A,(#65A0)	; load A with blink counter
253E:	FE96	CP #96	; check max blinks reached
2540:	C0	RET NZ	; return if not ready blinking
;			
2541:	3A 9B 63	LD A,(#639B)	; load switch status variable in A
2544:	E602	AND #02	; check bit 2 set
2546:	C25D25	JP NZ, #255D	; jump if switch 2 activated
;			
2549:	CD F0 24	CALL #24F0	; draw lower ladder
254C:	21206A	LD HL,#6A20	; load HL with heart sprite
254F:	36E3	LD (HL),#E3	; set heart sprite x-position
2551:	23	INC HL	; next
2552:	3676	LD (HL),#76	; set heart sprite
2554:	23	INC HL	; next
2555:	3609	LD (HL),#09	; set heart sprite color
2557:	23	INC HL	; next
2558:	3648	LD HL,#48	; set heart sprite y-position
255A:	C36025	JP #2560	; skip next
;			
255D:	CD 07 25	CALL #2507	; draw upper ladder

```

;
2560: 3A9B63 LD A,(#639B) ; load switch status variable in A
2563: E6 FB AND #FB ; reset bit 3 , stop blinking sparkle
2565: 32 9B 63 LD (#639B), A ; write back to switch status variable
2568: C9 RET ; return
;
2569: 3A 9B 63 LD A,(#639B) ; load switch status variable in A
256C: E6 01 AND #01 ; check bit 1 set
256E: C2 9E 25 JP NZ,#259E ; jump to next switch if switch already activated
2571: 3A 03 62 LD A,(#6203) ; load A with jumpman's x-position
2574: FE D8 CP #D8 ; check x-position = #D8
2576: C0 RET NZ ; return if x-position is not correct
2577: 3A 05 62 LD A,(#6205) ; load A with jumpman's y-position
257A: FE C8 CP #C8 ; check y-position = #C8
257C: C0 RET NZ ; return if y-position is not correct
257D: 3A 9B 63 LD A,(#639B) ; load switch status variable in A
2580: F6 05 OR #05 ; set bit 1 and bit 3, switch 1 activated, blink
2582: 32 9B 63 LD (#639B), A ; write back to switch status variable
2585: 3E 03 LD A,#03 ; load sound duration of 3x
2587: 32 85 60 LD (#6085),A ; play sound for item pickup
;
258A: 21206A LD HL,#6A20 ; load HL with heart sprite
258D: 36AB LD (HL),#AB ; set heart sprite x-position
258F: 23 INC HL ; next
2590: 3663 LD (HL),#63 ; set sparkle sprite
2592: 23 INC HL ; next
2593: 3609 LD (HL),#09 ; set heart sprite color
2595: 23 INC HL ; next
2596: 3667 LD HL,#67 ; set heart sprite y-position
;
2598: 3E 00 LD A,#00 ; A:=0
259A: 32A065 LD (#65A0), A ; reset blink counter
259D: C9 RET ; return
;
259E: 3A 9B 63 LD A,(#639B) ; load switch status variable in A
25A1: E6 02 AND #02 ; check bit 2 set
25A3: C0 RET NZ ; return if switch already activated
25A4: 3A 03 62 LD A,(#6203) ; load A with jumpman's x-position
25A7: FE E3 CP #E3 ; check x-position = #E3
25A9: C0 RET NZ ; return if x-position is not correct
25AA: 3A 05 62 LD A,(#6205) ; load A with jumpman's y-position
25AD: FE 49 CP #49 ; check y-position = #49
25AF: C0 RET NZ ; return if y-position is not correct
25B0: 3A 9B 63 LD A,(#639B) ; load switch status variable in A
25B3: F6 06 OR #06 ; set bit 2 and 3, switch 2 activated, blink
25B5: 32 9B 63 LD (#639B), A ; write back to switch status variable
25B8: 3E 03 LD A,#03 ; load sound duration of 3
25BA: 32 85 60 LD (#6085),A ; play sound for item pickup
;
25BD: 21206A LD HL,#6A20 ; load HL with heart sprite
25C0: 3693 LD (HL),#93 ; set heart sprite x-position
25C2: 23 INC HL ; next
25C3: 3663 LD (HL),#63 ; set sparkle sprite
25C5: 23 INC HL ; next
25C6: 3609 LD (HL),#09 ; set heart sprite color
25C8: 23 INC HL ; next
25C9: 364A LD HL,#4A ; set heart sprite y-position
;
25CB: 3E 00 LD A,#00 ; A:=0
25CD: 32A065 LD (#65A0), A ; reset blink counter
25D0: C9 RET ; return

```

Sparkles worden nu op de goede plaats getoond. Alleen flikkeren nog wel.

2518:	3A 27 62	LD A,(#6227)	; load A with screen number
251B:	3D	DEC A	; is this the girders?
251C:	C0	RET NZ	; if not return
;			
251D:	3A 9B 63	LD A,(#639B)	; load switch status variable in A
2520:	E6 04	AND #04	; check bit 3 set: blink sparkle
2522:	CA8225	JP Z,#2582	; if not set then do not blink – jump label AAAA
;			
2525:	21216A	LD HL,#6A21	; load HL with heart sprite
2528:	3AA065	LD A,(#65A0)	; load A with blink counter
252B:	3C	INC A	; increment blink counter
252C:	32A065	LD (#65A0), A	; store blink counter
;			
252F:	47	LD B,A	
2530:	FE 16	CP #16	
2532:	CA 54 25	JP Z, #2554	; sprite off – jump label HHHH
;			
2535:	78	LD A,B	
2536:	FE 32	CP #32	
2538:	CA 57 25	JP Z, #2557	; sprite on – jump label III
;			
253B:	78	LD A,B	
253C:	FE 48	CP #48	
253E:	CA 54 25	JP Z, #2554	; sprite off – jump label HHHH
;			
2541:	78	LD A,B	
2542:	FE 64	CP #64	
2544:	CA 57 25	JP Z, #2557	; sprite on – jump label III
;			
2547:	78	LD A,B	
2548:	FE 80	CP #80	
254A:	CA 54 25	JP Z, #2554	; sprite off – jump label HHHH
;			
254D:	78	LD A,B	
254E:	FE 96	CP #96	
2550:	CA 5A 25	JP Z, #255A	; stop blinking – jump label GGGG
;			
2553:	C9	RET	; no action continue blinking
;			
2554:	3664	LD (HL),#64	; set blank sprite – label HHHH
2556:	C9	RET	; continue blinking
;			
2557:	3663	LD (HL),#63	; set sparkle sprite – jump label II
2559:	C9	RET	; continue blinking
;			
255A:	3A 9B 63	LD A,(#639B)	; load switch status variable in A – label GGGG
255D:	E602	AND #02	; check bit 2 set
255F:	C27625	JP NZ, #2576	; jump if switch 2 activated – jump label DDDD
;			
2562:	CD F0 24	CALL #24F0	; draw lower ladder
2565:	21206A	LD HL,#6A20	; load HL with heart sprite
2568:	36E3	LD (HL),#E3	; set heart sprite x-position
256A:	23	INC HL	; next
256B:	3676	LD (HL),#76	; set heart sprite
256D:	23	INC HL	; next
256E:	3609	LD (HL),#09	; set heart sprite color
2570:	23	INC HL	; next
2571:	3648	LD HL,#48	; set heart sprite y-position
2573:	C37925	JP #2579	; skip next – jump label EEEE
;			
2576:	CD 07 25	CALL #2507	; draw upper ladder – label DDDD
;			
2579:	3A9B63	LD A,(#639B)	; load switch status variable in A – label EEEE

257C:	E6 FB	AND #FB	; reset bit 3 , stop blinking sparkle
257E:	32 9B 63	LD (#639B), A	; write back to switch status variable
2581:	C9	RET	; return
;			
2582:	3A 9B 63	LD A,(#639B)	; load switch status variable in A – label AAAA
2585:	E6 01	AND #01	; check bit 1 set
2587:	C2 B7 25	JP NZ,#25B7	; jump to next switch – jump label FFFF
258A:	3A 03 62	LD A,(#6203)	; load A with jumpman's x-position
258D:	FE D8	CP #D8	; check x-position = #D8
258F:	C0	RET NZ	; return if x-position is not correct
2590:	3A 05 62	LD A,(#6205)	; load A with jumpman's y-position
2593:	FE C8	CP #C8	; check y-position = #C8
2595:	C0	RET NZ	; return if y-position is not correct
2596:	3A 9B 63	LD A,(#639B)	; load switch status variable in A
2599:	F6 05	OR #05	; set bit 1 and bit 3, switch 1 activated, blink
259B:	32 9B 63	LD (#639B), A	; write back to switch status variable
259E:	3E 03	LD A,#03	; load sound duration of 3x
25A0:	32 85 60	LD (#6085),A	; play sound for item pickup
;			
25A3:	21206A	LD HL,#6A20	; load HL with heart sprite
25A6:	36AB	LD (HL),#AB	; set heart sprite x-position
25A8:	23	INC HL	; next
25A9:	3663	LD (HL),#63	; set sparkle sprite
25AB:	23	INC HL	; next
25AC:	3609	LD (HL),#0C	; set heart sprite color
25AE:	23	INC HL	; next
25AF:	3667	LD HL,#67	; set heart sprite y-position
;			
25B1:	3E 00	LD A,#00	; A:=0
25B3:	32A065	LD (#65A0), A	; reset blink counter
25B6:	C9	RET	; return
;			
25B7:	3A 9B 63	LD A,(#639B)	; load switch status variable in A – label FFFF
25BA:	E6 02	AND #02	; check bit 2 set
25BC:	C0	RET NZ	; return if switch already activated
25BD:	3A 03 62	LD A,(#6203)	; load A with jumpman's x-position
25C0:	FE E3	CP #E3	; check x-position = #E3
25C2:	C0	RET NZ	; return if x-position is not correct
25C3:	3A 05 62	LD A,(#6205)	; load A with jumpman's y-position
25C6:	FE 49	CP #49	; check y-position = #49
25C8:	C0	RET NZ	; return if y-position is not correct
25C9:	3A 9B 63	LD A,(#639B)	; load switch status variable in A
25CC:	F6 06	OR #06	; set bit 2 and 3, switch 2 activated, blink
25CE:	32 9B 63	LD (#639B), A	; write back to switch status variable
25D1:	3E 03	LD A,#03	; load sound duration of 3
25D3:	32 85 60	LD (#6085),A	; play sound for item pickup
;			
25D6:	21206A	LD HL,#6A20	; load HL with heart sprite
25D9:	3693	LD (HL),#93	; set heart sprite x-position
25DB:	23	INC HL	; next
25DC:	3663	LD (HL),#63	; set sparkle sprite
25DE:	23	INC HL	; next
25DF:	3609	LD (HL),#0C	; set heart sprite color
25E1:	23	INC HL	; next
25E2:	364A	LD HL,#4A	; set heart sprite y-position
;			
25E4:	3E 00	LD A,#00	; A:=0
25E6:	32A065	LD (#65A0), A	; reset blink counter
25E9:	C9	RET	; return

Dit werkt nu goed. Displayed de sparkle sprite netjes zonder flikkeren.

De nieuwe elevators definitie inplakken na de barrels:

Barrels: #3AE4 – #3B70

Elevators: #3B71 - #3C0C

En de aanroep(en) ernaar toe ook aanpassen. #3BE5 (oorspronkelijk) moet worden #3B71.

0CFA: 11713B LD DE,#3B71 ; load DE with start of table data for the elevators

2468: 21713B LD HL,#3B71 ; load HL with start of table data for elevators

Nu wel het veld goed. Maar nog de volgende dingen regelen: Elevator verplaatsen, items verplaatsen, firefoxes verplaatsen. Dit allemaal halen uit de beschrijving van prototype 4.

Aangepast en werkt goed.

Nu de rivets velden gaan toevoegen. Eerst een standaard veld tekenen en de ladders op de onderste levels weglaten.

Invoegen vanaf 3C0D:

3C0D: 00 33 58 33 80 ; top left ladder
3C12: 00 53 58 53 80 ; top left ladder (right side)
3C17: 00 AB 58 AB 80 ; top right ladder (left side)
3C1C: 00 CB 58 CB 80 ; top right ladder
3C21: 05 B7 30 48 30 ; girder above kong
3C26: 05 CF 58 30 58 ; girder kong stands on
3C2B: 05 D7 80 28 80 ; level 4 girder
3C30: 05 DF A8 20 A8 ; level 3 girder
3C35: 05 E7 D0 18 D0 ; level 2 girder
3C3A: 05 EF F8 10 F8 ; bottom level girder
3C3F: AA ; end code

Locatie van de Rivets definitie aanpassen. Was #3C8B en wordt nu #3C0D.

0CC3 110D3C LD DE,#3C0D ; load DE with start of table data for rivets
246E 210D3C LD HL,#3C0D ; load HL with table data for rivets

Ook eventjes aanpassen alleen rivets doen:

3A65: 04 04 ; level 1

Inderdaad nu het level zonder de ladders:



Resterende trappen random normaal/broken maken.

Aantal variabelen benodigd:

- Variabele laatste broken (1) / normal (0): #65A0
- Variabele aantal broken ladders (max 4): #65A1
- Variabele ladder index (starten vanaf 3): #65A2
- Ladder definitions index: #65A3
- Single ladder definition (type, bx, by, ox, oy, end): #65A4 t/m #65A9

Single ladder definition: #65A4 = 00 voor normale ladder en 01 voor broken ladder
 #65A5 = x-coordinaat bovenkant ladder
 #65A6 = y-coordinaat bovenkant ladder
 #65A7 = x-coordinaat onderkant ladder
 #65A8 = y-coordinaat onderkant ladder
 #65A9 = AA

De ladders (zonder de eerste byte die type aangeeft) in een aparte definitie opnemen en in volgorde :

Vanaf #3C40:

```
3C40: 1B D0 1B F8 ; bottom left ladder
3C44: 7B D0 7B F8 ; bottom center ladder
3C48: E3 D0 E3 F8 ; bottom right ladder
3C4C: 23 A8 23 D0 ; level 2 ladder left side
3C50: 5B A8 5B D0 ; level 2 ladder #2 of 4
3C54: A3 A8 A3 D0 ; level 2 ladder #3 of 4
3C58: DB A8 DB D0 ; level 2 ladder right side
3C5C: 2B 80 2B A8 ; level 3 ladder left side
3C60: 7B 80 7B A8 ; center ladder level 3
3C64: D3 80 D3 A8 ; level 3 ladder right side
3C68: AA             ; end code
```

Normaal rivets tekenen aanroep hier:

```
0CC9 3A2762 LD A,(#6227) ; load A with screen number
0CCC FE04 CP #04         ; screen is rivets level?
0CCE CC000D CALL Z,#0D00 ; yes, call sub to draw the rivets
```

En rivets tekenen hier:

```
; For the rivets level only - draw the rivets

0D00 0608 LD B,#08     ; for B = 1 to 8 rivets to draw
0D02 21170D LD HL,#0D17 ; load HL with start of table data below

0D05 3EB8 LD A,#B8     ; load A with #B8 = start code for rivet
0D07 0E02 LD C,#02     ; For C = 1 to 2
0D09 5E LD E,(HL)     ; load E with the high byte of the address
0D0A 23 INC HL         ; next HL
0D0B 56 LD D,(HL)     ; load D with the low byte of the address
0D0C 23 INC HL         ; next HL

0D0D 12 LD (DE),A     ; draw rivet onscreen
0D0E 3D DEC A         ; next graphic
0D0F 13 INC DE         ; next screen address
0D10 0D DEC C         ; Next C
0D11 C20D0D JP NZ,#0D0D ; loop until done

0D14 10EF DJNZ #0D05   ; Next B

0D16 C9 RET            ; return
```

Deze code toevoegen aan het stuk van de ladders tekenen (vanaf 25F2) en dan de aanroep ernaar toe ook aanpassen:

Dus de aanroep wordt:

0CCE CCF225 CALL Z,#25F2 ; yes, call sub to draw ladders and rivets

En de code om de rivets en de ladders te tekenen:

```
25F2: 0608 LD B,#08 ; for B = 1 to 8 rivets to draw
25F4: 21170D LD HL,#0D17 ; load HL with start of table data below
;
25F7: 3EB8 LD A,#B8 ; load A with #B8 = start code for rivet
25F9: 0E02 LD C,#02 ; For C = 1 to 2
25FB: 5E LD E,(HL) ; load E with the high byte of the address
25FC: 23 INC HL ; next HL
25FD: 56 LD D,(HL) ; load D with the low byte of the address
25FE: 23 INC HL ; next HL
;
25FF: 12 LD (DE),A ; draw rivet onscreen
2600: 3D DEC A ; next graphic
2601: 13 INC DE ; next screen address
2602: 0D DEC C ; Next C
2603: C2522F JP NZ,#2F52 ; loop until done
;
2606: 10EF DJNZ #25F7 ; Next B
;

25F2: CC000D CALL Z,#0D00 ; yes, call sub to draw the rivets
;
Allemaal NOPS ter test!
;
2608: 3E00 LD A,#00 ; A:=0
260A: 32A065 LD (#65A0),A ; set normal/broken switch to normal (0)
260D: 32A165 LD (#65A1),A ; set number of broken ladders to zero
2610: DD210463 LD IX,#6304 ; ladder index, start 6300, already 4 ladders drawn
2614: 21403C LD HL,#3C40 ; load HL with start table data ladders
;
2617: 7E LD A,(HL) ; load entry from table data - label CCCC
2618: FEAA CP #AA ; if AA then ...
261A: C8 RET Z ; ... end of table, return, all ladders drawn
;
261B: 3AA165 LD A,(#65A1) ; load number of broken ladders
261E: FE04 CP #04 ; if 4 then maximum nr of broken ladders reached
2620: CA3E26 JP Z,#263E ; jump to draw normal ladder - jump label AAAA
;
2623: 3AA065 LD A,(65A0) ; load normal/broken switch
2626: FE01 CP #01 ; if 1 then last ladder was broken
2628: CA3E26 JP Z,#263E ; jump to draw normal ladder - jump label AAAA
;
262B: CD5700 CALL #0057 ; load A with a random number
262E: FE80 CP #80 ; A<=80 ?, 50% chance
2630: CA3E26 JP Z,#263E ; jump to draw normal ladder - jump label AAAA
;
2633: 3E01 LD A,#01 ; A:=1; draw broken ladder
2635: 32A065 LD (#65A0),A ; set normal/broken switch to broken (1)
2638: 32A465 LD (#65A4),A ; set single ladder definition [1] to broken (#01)
263B: C34626 JP #2646 ; jump to copy ladder positions - jump label BBBB
;
263E: 3E00 LD A,#00 ; A:=0; draw normal ladder - label AAAA
2640: 32A065 LD (#65A0),A ; set normal/broken switch to normal (0)
2643: 32A465 LD (#65A4),A ; set single ladder definition [1] to normal (#00)
;
2646: 7E LD A,(HL) ; reload entry from table - label BBBB
2647: 32A565 LD (#65A5),A ; set single ladder definition [2] to x-pos of top
```

```

264A: 23      INC  HL          ; next table entry
264B: 7E      LD   A,(HL)      ; load entry from table
264C: 32A665  LD   (#65A6),A  ; set single ladder definition [3] to y-pos of top
264F: 23      INC  HL          ; next table entry
2650: 7E      LD   A,(HL)      ; load entry from table
2651: 32A765  LD   (#65A7),A  ; set single ladder definition [4] to x-pos of bottom
2654: 23      INC  HL          ; next table entry
2655: 7E      LD   A,(HL)      ; load entry from table
2656: 32A865  LD   (#65A8),A  ; set single ladder definition [5] to y-pos of bottom
2659: 23      INC  HL          ; next table entry
265A: 3EAA    LD   A,#AA       ; A:=#AA
265C: 32A965  LD   (#65A9),A  ; set single ladder definition [6] to AA, means end
265F: 23      INC  HL          ; next table entry
;
2660: 11A465  LD  DE,#65A4     ; load start table data
2663: CDA70D  CALL #0DA7      ; draw screen
;
2666: E5      PUSH HL         ; save HL for later
;
2667: 21A465  LD  HL,#65A4     ; load HL with start of table data for ladder
266A: CD7524  CALL #2475;    ;
;
266D: E1      POP  HL         ; restore HL
266E: DD23    INC  IX         ; increment ladder index
;
2670: C31726  JP  #2617      ; process next ladder—jump label CCCC

2666: DD210463 LD  IX,#6304
;
266A: 21EA24  LD  HL,#24EA     ; load HL with start of table data for ladder
266D: CD7524  CALL #2475;    ;
;
2670: C9      RET          

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

```

Nu nog bepalen waar we dit gaan tussenvoegen?

Code van #25F2 t/m #26F9 kan gebruikt worden: is voor conveyors direction en pulleys.

Maar dan wel de call uit de main routine aanpassen naar 3x NOP:

```

19A7  CDFA26  CALL  #26FA      ; do stuff for elevators
19AA  CDF225  CALL  #25F2      ; handle conveyor directions, adjust Mario's speed based on conveyor directions
19AD  CDDA19  CALL  #19DA      ; check for mario picking up bonus item

```

19AA 00 00 00 NOP NOP NOP

Aanroep andersom. Nu eerst girders tekenen en dan de ladders. Dat gaat niet goed omdat de ladders getekend worden met boven girder en onder girder met verkeerde graphic (die van barrels ipv girder met gat).

```

; other screens return here

0CC6 CDA70D CALL #0DA7 ; draw the screen

0CC9 3A2762 LD A,(#6227) ; load A with screen number
0CC9 FE04 CP #04 ; screen is rivets level?
0CCE CC000D CALL Z,#0D00 ; yes, call sub to draw the rivets

```

Wordt dan:

```

0CC6 3A2762 LD A,(#6227) ; load A with screen number
0CC9 FE04 CP #04 ; screen is rivets level?
0CCB CCF225 CALL Z,#25F2 ; yes, call sub to draw ladders and rivets
;
0CCE CDA70D CALL #0DA7 ; draw the screen

```

En jumps naar #0CC6 aanpassen naar #0CCF.

```

0CDC C3CE0C JP #0CCE ; jump back
0CEF C3CE0C JP #0CCE ; jump back
0CFD C3CE0C JP #0CCE ; jump back

```

Dit gaat niet goed. Weer terugzetten.

Misschien gaat het niet goed omdat het stuk op een verkeerde plek aangeroepen wordt. Daarna wordt waarschijnlijk pas het originele activeren van de trappen aangeroepen. Misschien reset dat wel het activeren van de random trappen.

Kijken waar wel invoegen dan.

Gaat allemaal niet lukken zo. Probleem is met volgorde tekenen. Volgorde normaal is dat trappen eerst getekend worden, dan de girders waardoor de boven- en onderkant van de girder goed wordt (van barreltype/draad naar rivet type/gat) en dan de rivets die de hele girders onderbreken. Als je dit gaat omgooien (omdat activeren trappen aan het einde moet, dan kloppen de boven- en onderkant niet). Dus heel lastig allemaal.

Gaan voor een andere oplossing: twee (of drie) verschillende velden maken en die op basis van level tekenen!

Eerst kiezen welke twee. Pak versies 0.0.2 en 0.0.3. Kan later ook nog getuned worden. Als je er maar vast twee hebt.

Eerst de eerste rivets aanpassen. Hiervoor nemen we versie 0.0.2.
Invoegen vanaf #3C0D t/m 3C71.

Rivets begon origineel op 3C8B dus de aanroepen van deze data tabel aanpassen naar 3C0D!

```

0CC3 110D3C LD DE,#3C0D ; load DE with start of table data for rivets
246E 210D3C LD HL,#3C0D ; Load table data for rivets

```

Dat is gelukt.

Nu de tweede rivets toevoegen. Hiervoor nemen we versie 0.0.1.

Deze invoegen vanaf #3C72. En even los testen door aanpassen bovenstaande twee table data referenties.

Dat is gelukt.

Dus rivets definities: #3C0D en #3C72.

Nu dan op de twee plekken ervoor zorgen dat de rivets table data gekozen wordt op basis van het level dat gespeeld wordt.

#6229 is het level nr.

Checken even/oneven level kan als volgt:

```
1913 218A60 LD HL,#608A ; load HL with sound address
1916 360C LD (HL),#0C ; play sound for rivets cleared
1918 3A2962 LD A,(#6229) ; load A with level #
191B 0F RRCA ; roll a right . is this an odd level ?
191C 3802 JR c,#1920 ; Yes, skip next step

191E 3605 LD (HL),#05 ; else play sound for even numbered rivets

1920 23 INC HL ; HL := #608B = sound duration
```

De extra keuze stukken invoegen vanaf 25F2. En de call naar 25F2 NOP maken:

19AA CDF225 CALL #25F2

Aanpassen naar

19AA 000000 NOPNOPNOP

De eerste keuze rivets data op #0CC3:

0CC3: C3F225 JP #25F2

...

```
25F2: 110D3C LD DE,#3C0D ; load DE with start of table data for odd rivets
25F5: 3A2962 LD A,(#6229) ; load A with level #
25F8: 0F RRCA ; roll a right . is this an odd level ?
25F9: 3803 JR C,#25FE ; Yes, skip next step
25FB: 11723C LD DE,#3C72 ; load DE with start of table data for even rivets
25FE: C3C60C JP #0CC6 ; jump back
```

De tweede keuze rivets data op #246E

246E: C30126 JP #2601

...

```
2601: 210D3C LD HL,#3C0D ; load DE with start of table data for odd rivets
2604: 3A2962 LD A,(#6229) ; load A with level #
2607: 0F RRCA ; roll a right . is this an odd level ?
2608: 3803 JR C,#260D ; Yes, skip next step
260A: 21723C LD HL,#3C72 ; load DE with start of table data for even rivets
260D: C37124 JP #2471 ; jump back
```

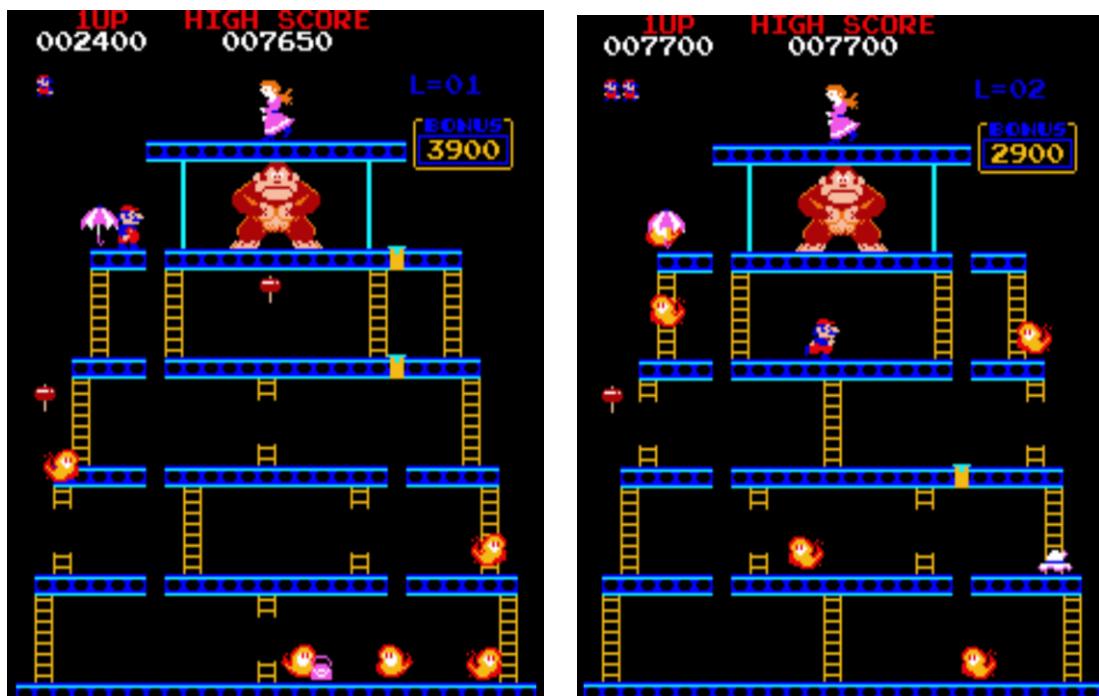
Bleef hangen. Lijkt opgelost door maar 4 broken ladders toe te passen.

Level designen:

3C8B: 00 7B 80 7B A8	center ladder level 3	X	
3C90: 00 7B D0 7B F8	bottom center ladder	X	
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		X

3CAE: 00 D3 80 D3 A8	level 3 ladder right side		X
3CB3: 00 23 A8 23 D0	level 2 ladder left side	X	
3CB8: 00 5B A8 5B D0	level 2 ladder #2 of 4		X
3CBD: 00 A3 A8 A3 D0	level 2 ladder #3 of 4	X	X
3CC2: 00 DB A8 DB D0	level 2 ladder right side		
3CC7: 00 1B D0 1B F8	bottom left ladder		
3CCC: 00 E3 D0 E3 F8	bottom right ladder		
3CD1: 05 B7 30 48 30	girder above kong		
3CD6: 05 CF 58 30 58	girder kong stands on		
3CDB: 05 D7 80 28 80	level 4 girder		
3CE0: 05 DF A8 20 A8	level 3 girder		
3CE5: 05 E7 D0 18 D0	level 2 girder		
3CEA: 05 EF F8 10 F8	bottom level girder		
3CEF: AA	end code		

Zo nu gemaakt. Wel erg lastig. Gaan testen.



Proberen of het eenvoudiger wordt wanneer de firefoxes ook naar beneden kunnen wanneer je op gelijk of hoger niveau staat.

Dit lijkt geregel te worden door dit stuk:

```
; Else there is a ladder nearby to go down
3360 DD701F LD    (IX+#1F),B      ; Store B into +#1F = Y-position of bottom of ladder
3363 3A0562 LD    A,(#6205)      ; \ Return without taking the ladder if Mario is at or above the Y-position of the fireball
3366 47      LD    B,A          ; |
3367 DD7E0F LD    A,(IX+#0F)      ; |
336A 90      SUB   B            ; |
336B D0      RET   NC          ; /
336C DD360D04 LD    (IX+#0D),#04  ; Else set direction to descending ladder
3370 C9      RET   ; return
```

Dit regelt bijvoorbeeld dat je in de elevators screen een stukje de linker ladder moet afdalen om de fireball ernaast ook een van de twee ladders te laten nemen.

Je zou deze beperking kunnen verwijderen door 3363 t/m 336B in NOP's te veranderen.

En dat maakt het inderdaad een stuk makkelijker. Maar door de gebroken trappen toch ook nog wel moeilijk genoeg.

Kijken hoe het eruit ziet als de girders beneden smal en boven breed zijn:
Mee getest, maar niet echt leuk. Ziet er vreemd uit. Daarnaast staan de items dan niet op de goede plaats en worden de fireboxes op de verkeerde plaats spawned (waardoor sommige stationair in de lucht blijven hangen). Dit dan allemaal aanpassen naar een nieuwe plek, is de moeite niet waard.

Nu aan de slag met de conveyors stage.

```
00 63 18 63 58 ; kong's ladder (right)
00 53 18 53 58 ; kong's ladder (left)
00 93 38 93 58 ; ladder leading to girl
02 97 38 68 38 ; girder where girl sits
03 EF 58 10 58 ; top conveyor girder
02 EF F8 10 F8 ; bottom level girder
```

De definitie op een nieuwe plek aanmaken. Was op #3B5D.

Veld definitie vooralsnog vanaf #2670 Als veld af dan altijd nog te verschuiven.
Aanroep aanpassen:

```
0CDF 117026 LD DE,#2670 ; load DE with start of table data for conveyors
2461 217026 ID HL,#2670 ; load HL with start of table data for conveyors
```

De draaipunten van de conveyors worden nog getekend. Dat is eenmalig. Waar is dat? Daar kun je andere eenmalige configuraties voor de conveyors invoegen. Dit begint op 101F

Platform blink counter vullen op de volgende plek:

```
103A DD21A065 LD IX,#65A0 ; load IX with start of pies
103E 21B869 LD HL,#69B8 ; load HL with sprites for pies
1041 111000 LD DE,#0010 ; DE := #10
1044 0606 LD B,#06 ; B := 6
1046 CDD311 CALL #11D3

103A: 3E00 LD A,#00 ; A := #00
103C: 329B63 LD (#639B),A ; store into switch status variable
103F: 32A065 LD (#65A0),A ; store in blink counter
1042-1048 NOP's
```

En dan in de normale gameflow het volgende aanroepen:

```
2610: 3A2762 LD A,(#6227) ; load A with screen number
2613: FE02 CP #02 ; are we on the conveyors?
2615: C21304 RET NZ ; no, return
;
2618: 3AA065 LD A,(#65A0) ; load A with blink counter
261B: 3C INC A ; increment blink counter
261C: 32A065 LD (#65A0), A ; store blink counter
;
261F: 47 LD B,A ; temporary store blink counter in B
2620: FE 50 CP #50 ; is blink counter #50?
2622: CA 2E 26 JP Z, #262E ; yes, play sound - jump label AAAA
;
2625: 78 LD A,B ; reload A with blink counter from B
2626: FE 60 CP #60 ; is blink counter #60?
2628: CA 32 26 JP Z, #2632 ; yes, change platform state - jump label BBBB
;
```

```

262B: C9      RET          ; it is not time to blink yet, return
;
262C: 3E03    LD A,#03      ; load sound duration – label AAAA
262E: 328360  LD (#6083),A   ; plays the coin insert sound
2631: C9      RET          ; it is not time to blink yet, return
;
2632: 3A9B63  LD A,(#639B)  ; load A with switch status – label BBBB
2635: FE01    CP #01        ; check if platform is visible
2637: CA 43 26 JP Z, #2643  ; yes, platform is visible – jump label CCCC
;
263A: 3C      INC A         ; make A #01: make platform visible
263B: 329B63  LD (#639B), A  ; store switch status
263E: 3EB0    LD A,#B0       ; load A with girder character
2640: C3 49 26 JP, #2649    ; skip next step – jump label DDDD
;
2643: 3D      DEC A         ; make A #00: make platform invisible – label CCCC
2644: 329B63  LD (#639B), A  ; store switch status
2647: 3E10    LD A,#10       ; load A with blank character
;
2649: 217876  LD HL,#7678    ; video memory address to write – label DDDD
264C: 77      LD (HL),A      ; write memory address
264D: 219876  LD HL,#7698    ; video memory address to write
2650: 77      LD (HL),A      ; write memory address
2651: 21B876  LD HL,#76B8    ; video memory address to write
2654: 77      LD (HL),A      ; write memory address
2655: 21D876  LD HL,#76D8    ; video memory address to write
2658: 77      LD (HL),A      ; write memory address
;
2659: C9      RET          ; return

```

De aanroep voor elkaar krijgen door de aanroep in te voegen:

```
19AA CD1026 CALL #2610 ; handle platform appear/disappear
```

Werkt goed: platform wordt getekend en weer verwijderd, enz, enz. Geluid nog niet echt goed. Misschien zelfs verwijderen?

Nu platform en trap maken:

```
02 30 C0 10 C0
00 23 C0 23 F8
```

Werkt heel erg goed. Jumpman kan op het blinking platform springen, en als het platform weer verdwijnt, valt jumpman naar beneden. Daar hoeft geen logica voor gemaakt te worden.



Tekenen van de conveyor katrollen:

```
1065 21243E LD HL,#3E24 ; set source table data
1068 11E469 LD DE,#69E4 ; set destination RAM sprites
106B 011800 LD BC,#0018 ; set counter
106E EDB0 LDIR ; draw pulleys
```

Definitie van de 6 conveyor katrollen:

```
; the 6 conveyor pulleys

3E24: 17 50 00 5C ; 50 = edge of conveyor pulley
3E28: E7 D0 00 5C ; D0 = edge of conveyor pulley inverted
3E2C: 8C 50 00 84
3E30: 73 D0 00 84
3E34: 17 50 00 D4
3E38: E7 D0 00 D4
```

Dus de katrollen weghalen door #1065 t/m 106F in NOPs te veranderen. De #3E24 t/m 3E3B komen daarmee ook beschikbaar. Aangepast en de katrollen zijn inderdaad verdwenen.

Nu zodanig aanpassen dat er vier platformen getekend kunnen worden.

```
2649; 217876 LD HL,#7678 ; video memory address to write – label DDDD
264C: CD6226 CALL #2662 ; call draw routine – call label EEEE
264F; 21F775 LD HL,#75F7 ; video memory address to write
2652: CD6226 CALL #2662 ; call draw routine – call label EEEE
2655: 217675 LD HL,#7576 ; video memory address to write
2658: CD6226 CALL #2662 ; call draw routine – call label EEEE
265B: 21F574 LD HL,#74F5 ; video memory address to write
265E: CD6226 CALL #2662 ; call draw routine – call label EEEE
;
2661: C9 RET ; return
;
2662: 112000 LD DE,#0020 ; DE = #20; offset to next vertical line – label EEEE
;
2665: 77 LD (HL),A ; write memory address
2666: 19 ADD HL,DE ; add offset
2667: 77 LD (HL),A ; write memory address
2668: 19 ADD HL,DE ; add offset
2669: 77 LD (HL),A ; write memory address
266A: 19 ADD HL,DE ; add offset
266B: 77 LD (HL),A ; write memory address
;
266C: C9 RET ; return
```



Het tempo is niet aanpasbaar. Dit komt doordat de blink counter niet op nul gezet wordt. Dat moet wel. Dat dus nog tussenvoegen.

```
2649; 217876 LD HL,#7678 ; video memory address to write - label DDDD
264C: CD6726 CALL #2667 ; call draw routine - call label EEEE
264F; 21F775 LD HL,#75F7 ; video memory address to write
2652: CD6726 CALL #2667 ; call draw routine - call label EEEE
2655; 217675 LD HL,#7576 ; video memory address to write
2658: CD6726 CALL #2667 ; call draw routine - call label EEEE
265B; 21F574 LD HL,#74F5 ; video memory address to write
265E: CD6726 CALL #2667 ; call draw routine - call label EEEE
;
2661: 3E00 LD A,#00 ; A := #00
2663: 32A065 LD (#65A0),A ; store in blink counter
;
2666: C9 RET ; return
;
2667: 112000 LD DE,#0020 ; DE = #20; offset to next vertical line - label EEEE
;
266A: 77 LD (HL),A ; write memory address
266B: 19 ADD HL,DE ; add offset
266C: 77 LD (HL),A ; write memory address
266D: 19 ADD HL,DE ; add offset
266E: 77 LD (HL),A ; write memory address
266F: 19 ADD HL,DE ; add offset
2670: 77 LD (HL),A ; write memory address
;
2671: C9 RET ; return
```

De definitie van de conveyors moet hierdoor opschuiven naar #2672.

Aanroep ook aanpassen:

Aanroep aanpassen:

```
0CDF 117226 LD DE,#2672 ; load DE with start of table data for conveyors
2461 217226 LD HL,#2672 ; load HL with start of table data for conveyors
```

Nu wel tempo aan te passen! Geluid is uit te zetten door de betreffende compare hoger te maken dan de compare van het tekenen.

Twee girders toevoegen naast de oil can. Zodat de firefoxes kunnen gaan bewegen. Blijven nu in de lucht hangen.

```
02 F7 80 88 80 ; top right conveyor next to oil can
02 77 80 08 80 ; top left conveyor next to oil can
```

Raar want de extractable ladders lopen niet tot de girder. Daartoe broken ladders toevoegen:

```
01 23 58 23 80 ; top broken ladder left side
01 DB 58 DB 80 ; top broken ladder right side
```

Op termijn de linker extractable ladder nog verwijderen!

Voordat we het level verder gaan ontwerpen, eerst zorgen dat het einde goed is. Daarom ter test een lange ladder toevoegen naar boven:

```
01 A3 80 A3 F8
```

Hiermee kun je tot de rechter girder naast de oil can. Je kunt nu de retractable ladder op en zodra je op het niveau naast kong bent is het level klaar. Maar dan gebeurt er niets. Komt waarschijnlijk omdat Kong dan moet moven, maar dat gebeurd niet. Wellicht hetzelfde maken als de girders. Dus wellicht andere jumps e.d. Proefondervindelijk testen.



Eigenlijk level pas klaar als bij Pauline. Dan moeten we de waardes aanpassen.

```
1E66 fe51      CP      #51          ; else on the conveyors.  is mario high enough to end level?
1E68 d0        RET     nc           ; no, return
```

Bij girders en elevators wordt getest op #31. Hiermee testen. Ja dat werkt.

Nu nog wel dat je de retractable ladder niet af kan. En je kunt naar links doorlopen, voor Donkey Kong langs zonder af te gaan. Eigenlijk dezelfde check toepassen als bij girders, zodat je niet verder kan lopen dan de trap. Hoe zit dat bij de elevators eigenlijk? Je kunt dan wel verder lopen dan de trap. Maar ga je dan af als je bij Kong komt?

DK end level check goedmaken.

5 fasen 1 t/m 4 die hier gesprongen worden:

```
; else the conveyors

1633 3A8863  LD    A,(#6388)    ; load A with this usually zero, counts from 1 to 5 when the level is complete
1636 EF       RST   #28         ; jump based on A

1637 A3 16      ; #16A3        ; 0
1639 BB 16      ; #16BB        ; 1
163B 32 17      ; #1732        ; 2
163D 57 17      ; #1757        ; 3
163F 8E 17      ; #178E        ; 4
```

Deze rij komt deels overeen met de fasen die voor girders en elevators uitgevoerd worden:

```

; arrive when GameMode2 == #16 (level completed). called from #0701

1615 CDBD30 CALL #30BD ; clear sprites
1618 3A2762 LD A,(#6227) ; load a with screen number
161B 0F RRCA ; roll right with carry. is this the rivets or the conveyors?
161C d22f16 JP NC,#162F ; yes, skip ahead to #162F

; handle for girders or elevators, they are same here

161F 3A8863 LD A,(#6388) ; load A with this counter usually zero, counts from 1 to 5 when the level is complete
1622 EF RST #28 ; jump based on A

1623 54 16 ; #1654 ; 0
1625 70 16 ; #1670 ; 1
1627 8A 16 ; #168A ; 2
1629 32 17 ; #1732 ; 3
162B 57 17 ; #1757 ; 4
162D 8E 17 ; #178E ; 5

162F 0F RRCA ; roll right again. is this the rivets ?
1630 D24116 JP NC,#1641 ; yes, skip ahead

; else the conveyors

1633 3A8863 LD A,(#6388) ; load A with this usually zero, counts from 1 to 5 when the level is complete
1636 EF RST #28 ; jump based on A

1637 A3 16 ; #16A3 ; 0
1639 BB 16 ; #16BB ; 1
163B 32 17 ; #1732 ; 2
163D 57 17 ; #1757 ; 3
163F 8E 17 ; #178E ; 4

; rivets

1641 CDBD1D CALL #1DBD ; check for bonus items and jumping scores, rivets

```

Eigenlijk klinkt het logisch dat we dezelfde einde procedure gebruiken als voor girders en elevators. Dit betekent dat de selectie aangepast dient te worden en dan kijken of dat werkt.

```

1615 CDBD30 CALL #30BD ; clear sprites
;
1618 3A2762 LD A,(#6227) ; load a with screen number
161B 0F RRCA ; roll right with carry. is this the rivets or the conveyors?
161C d22f16 JP NC,#162F ; yes, skip ahead to #162F
;
1618 3A2762 LD A,(#6227) ; load a with screen number
161B FE04 CP #04 ; is this rivets screen?
161D CA4116 JP Z,#1641 ; if yes, jump ahead
;
1620 3A8863 LD A,(#6388) ; counts from 1 to 5 when the level is complete
1623 EF RST #28 ; jump based on A

1624 54 16 ; #1654 ; 0
1626 70 16 ; #1670 ; 1
1628 8A 16 ; #168A ; 2
162A 32 17 ; #1732 ; 3
162C 57 17 ; #1757 ; 4
162E 8E 17 ; #178E ; 5
;
162F 0F RRCA ; roll right again. is this the rivets ?
1630 D24116 JP NC,#1641 ; yes, skip ahead
;
; else the conveyors
;
1633 3A8863 LD A,(#6388) ; counts from 1 to 5 when the level is complete
1636 EF RST #28 ; jump based on A

```

```

1637 A3 16      ; #16A3      ; 0
1639 BB 16      ; #16BB      ; 1
163B 32 17      ; #1732      ; 2
163D 57 17      ; #1757      ; 3
163F 8E 17      ; #178E      ; 4
;
; rivets
;
1641 CDBD1D    CALL   #1DBD      ; check for bonus items and jumping scores, rivets

```

Stuk van #1630 t/m #1640 kan NOP worden en wordt niet gebruikt. Wellicht te gebruiken voor de test of je niet voorbij Kong's ladders kunt komen?

Vreemde overgang in logica van #162D naar #162F. Bij #162F begint het stuk voor de oude rivets en conveyors. Loopt het stuk van de girders en elevators daar dan niet in door? Nee, waarschijnlijk niet. Er wordt gesprongen naar #178E en dat is een boel code voor het opnieuw schonen van het scherm en initialiseren waarden. Op #1825 wordt daarvandaan vervolgens een return gedaan. En dat is een return van een overkoepelende call.

Dit werkt nu. Einde level gaat goed. Met animatie van DK die Pauline pakt en met de trappen naar boven verdwijnt.

De check toevoegen dat je niet voorbij Kong's ladders kunt komen. Dit wordt hier geregeld:

```

242D 1D      DEC  E      ; no, DE:= #0000
242E 3A2762  LD   A,(#6227) ; load A with screen number (01, 10, 11 or 100)
2431 0F      RRCA          ; rotate right with carry. is this the girders or elevators?
2432 d0      RET   nc      ; no, return

2433 3A0562  LD   A,(#6205) ; otherwise load A with mario's Y position
2436 fe58    CP   #58      ; is this > #58 ?
2438 d0      RET   nc      ; Yes, return

2439 3A0362  LD   A,(#6203) ; else load A with mario's X position
243C FE6C    CP   #6C      ; is this > #6C ?
243E D0      RET   NC      ; Yes, return

243F 14      INC   D      ; else DE := #0100
2440 C9      RET   ; and return

```

Deze test wordt alleen uitgevoerd voor girders en elevators: de enige screens waarbij je door speelt op de bovenste girder en naar Pauline toe moet klimmen. Hier moet de test dan ook aangepast worden.

```

242D 1D      DEC  E      ; no, DE:= #0000
;
242E C33016  JP   #1630      ; jump to the additional code at #1630
;
2431 0000    NOP, NOP      ;
;
242E 3A2762  LD   A,(#6227) ; load A with screen number (01, 10, 11 or 100)
2431 0F      RRCA          ; rotate right with carry. is this the girders or elevators?
2432 D0      RET   NC      ; no, return
;
2433 3A0562  LD   A,(#6205) ; otherwise load A with mario's Y position
2436 fe58    CP   #58      ; is this > #58 ?
2438 d0      RET   NC      ; Yes, return
;
2439 3A0362  LD   A,(#6203) ; else load A with mario's X position
243C FE6C    CP   #6C      ; is this > #6C ?
243E D0      RET   NC      ; Yes, return
;
243F 14      INC   D      ; else DE := #0100

```

```
2440 C9      RET          ; and return
```

Volgende code toevoegen:

```
1630 3A2762 LD A,(#6227) ; load a with screen number  
1633 FE04 CP #04        ; is this rivets screen?  
1635 C8 RET Z           ; if yes, return  
1636 C33324 JP #2433    ; jump back
```

Dat werkt goed. Je kunt nu niet meer verder naar links bij de trap lopen en ook niet tegen Donkey Kong aan lopen. Retractable ladder links verwijderen/verplaatsen. Sowieso kijken of de retractable ladders ergens anders in het level geplaatst kunnen worden. Retractable ladder werkt wel vreemd. Als je erop omhoog klimt tot op de bovenste girder, dan schuif je door de girder naar beneden wanneer de ladder weer naar beneden gaat. En daarna val je en ga je af. Dit pleit ervoor om de retractable ladder te verwijderen.

Nu eerst de platformen op de juiste plaats zetten (2 links en 2 rechts boven elkaar). De code die de platformen tekent is:

```
2649; 219476 LD HL,#7694 ; video memory address to write: upper left  
264C: CD6726 CALL #2667   ; call draw routine - call label EEEE  
264F; 211475 LD HL,#7514 ; video memory address to write: upper right  
2652: CD6726 CALL #2667   ; call draw routine - call label EEEE  
2655; 219B96 LD HL,#769B ; video memory address to write: lower left  
2658: CD6726 CALL #2667   ; call draw routine - call label EEEE  
265B; 211B75 LD HL,#751B ; video memory address to write: lower right  
265E: CD6726 CALL #2667   ; call draw routine - call label EEEE
```

Eerste platform naar boven verschuiven: video ram - 1.

Nu eerst het linker start platform en de trap aanpassen.

De definitie van het conveyors veld zit op #2670.

```
#2690: 02 28 D8 10 D8 ; girder left bottom  
#2695: 00 23 D8 23 F8 ; ladder left bottom
```

```
#26B3: 02 88 D8 70 D8 ; girder middle bottom
```

AA

Ziet er nu zo uit:



Girder aan de rechterkant maken:

#26B8: 02 EF D8 D0 D8

AA

Bovenste twee knipperende platformen een regel naar beneden en de onderste naar boven:

2649;	219576	LD	HL,#7695	; video memory address to write: upper left
264C:	CD6726	CALL	#2667	; call draw routine - call label EEEE
264F;	211575	LD	HL,#7515	; video memory address to write: upper right
2652:	CD6726	CALL	#2667	; call draw routine - call label EEEE
2655;	219A96	LD	HL,#769A	; video memory address to write: lower left
2658:	CD6726	CALL	#2667	; call draw routine - call label EEEE
265B;	211A75	LD	HL,#751A	; video memory address to write: lower right
265E:	CD6726	CALL	#2667	; call draw routine - call label EEEE

Lange trap veranderen in girder aan de rechterkant:

#26AE: 02 EF B0 D0 B0

Interessant. De platformen hoger dan de girders. Lastiger te springen. Eventjes zo laten.



Ladder tussen de twee girders rechts maken:

#26BD: 00 E3 B0 E3 D8

En de middelste girder en linker girder maken:

#26C2: 02 88 B0 70 B0

#26C7: 02 28 B0 10 B0

Dat geeft:



Nu een aantal ladders invoegen: twee ladders aan beide zijden van de oil can:

```
#26CC: 01 73 80 73 B0  
#26D1: 01 8A 80 8A B0
```

Ziet er nu zo uit:



Vanaf nu eventjes dit testen met level 3. Level 1 en Level 2 kennen geen conveyors screen. Dus het aantal firefoxes klopt niet.

```
01D1 3E01 LD A,#01 ; A := 1  
01D3 320760 LD (#NoCredits),A ; store into credit indicator == no credits exist  
01D6 322962 LD (#6229),A ; initialize level to 1  
01D9 322862 LD (#6228),A ; set number of lives remaining to 1
```

Wat gebeurt er als je #01D1 verandert in 3E03? Dan gebeurt er niets?

Er is nog een stuk waar het level geïnitialiseerd wordt:

```
076B 3E01 LD A,#01 ; A := 1  
076D 322762 LD (#6227),A ; load screen number with 1  
0770 322962 LD (#6229),A ; load level # with 1  
0773 322862 LD (#6228),A ; load number of lives with 1
```

Wat gebeurt er als je #076B verandert in 3E03? Dan gebeurt er ook niets. Allebei weer terugzetten.

Level anders zetten in het progress screen. Daar wordt een muziekje gespeeld. Dit tijdelijk weghalen en daar het level zetten:

```
0BF2 218A60 LD HL,#608A ; load HL with tune address  
0BF5 3602 LD (HL),#02 ; play how high can you get sound?  
0BF7 23 INC HL ; HL := #608B . load HL with music timer ?  
0BF8 3603 LD (HL),#03 ; set to 3 units
```

```
0BF2 3E03 LD A,#03 ; A := 3  
0BF4 322962 LD (#6229),A ; load level # with 3  
0BF7 NOP NOP NOP
```

Dat werkt wel. Wel later weer terugzetten.

Nu trap maken in midden tussen onderste en middelste girder:

```
#26D6: 00 83 B0 83 D8
```

Retractable ladders weghalen. De code voor het initieel tekenen van de retractable ladders is:

```

105A 211C3E LD HL,#3E1C ; load HL with start of table data
105D 114469 LD DE,#6944 ; load DE with sprite start for moving ladders
1060 010800 LD BC,#0008 ; set byte counter to 8
1063 EDB0 LDIR ; draw moving ladders

```

Dit stuk allemaal NOPS maken.

De code voor het bewegen van de retractable ladders is #2207 t/m #22CA. Deze code kan opnieuw gebruikt worden. Eerst maar meteen een return geven:

```
2207: C9 RET ; return, don't move the retractable ladders
```

Werkt goed. Alleen de twee ladders zijn broken. Dat zijn de volgende ladders:

```
00 DB 80 DB B0 ; middle ladder right side
00 D3 58 D3 80 ; top ladder right side
```

```
#26DB: 00 23 80 23 B0 ; top ladder left side
#26E0: 01 1B B0 1B D8 ; middle broken ladder left side
#26E5: 01 DB D8 DB F8 ; bottom broken ladder right side
```

Onderste twee platformen naar beneden:

```

2655; 219B96 LD HL,#769B ; video memory address to write: lower left
2658: CD6726 CALL #2667 ; call draw routine – call label EEEE
265B; 211B75 LD HL,#751B ; video memory address to write: lower right
265E: CD6726 CALL #2667 ; call draw routine – call label EEEE

```

Ziet er nu zo uit:



Waar komen de twee korte stukken trap vandaan? Zijn dat nog restanten van de retractable ladders? Maar die zijn helemaal verwijderd? Uitzoeken.

Onderste platformen en girders naar beneden.

```

2655; 219C76 LD HL,#769C ; video memory address to write: lower left
2658: CD6726 CALL #2667 ; call draw routine – call label EEEE
265B; 211C75 LD HL,#751C ; video memory address to write: lower right
265E: CD6726 CALL #2667 ; call draw routine – call label EEEE

```

De girders met Y-waarde D8 veranderen in E0.

Bovenste girders naar boven. De girders met Y-waarde B0 veranderen in A8.
Nog een stuk naar boven: De girders met Y-waarde A8 veranderen in A0.

Bovenste platformen naar boven.

```
2649; 219476 LD HL,#7694 ; video memory address to write: upper left  
264C: CD6726 CALL #2667 ; call draw routine - call label EEEE  
264F; 211475 LD HL,#7514 ; video memory address to write: upper right  
2652: CD6726 CALL #2667 ; call draw routine - call label EEEE
```

De oil can naar boven: Y-waarde veranderen van 88 naar 80.

```
3E00 27 49 0C F0 ; oil can for girders  
3E04 7F 49 0C 80 ; oil can for conveyors ?
```

Vlammen boven de oil van naar boven: Y-waarde veranderen van 78 naar 70.

```
3DF4: 27 70 01 E0 00 00 ; initial data for fires on girders ?  
3DFA: 7F 40 01 70 02 00 ; initial data for conveyors to release a fire ?
```

Nu nog de fireballs die spawnen omhoog doen.

; controls the positions of fires coming out of the oil can on the conveyors

```
3AAC: 80 7B 78 76 74 73 72 71 70 70 6F 6F 6F 70 70 71 72 73 74 75 76 77 78  
3AC3: AA ; end code
```

Serie allemaal met 8 verlagen?: lijkt een serie met Y-waarden die de boog beschrijven:

```
3AAC: 78 73 70 6E 6C 6B 6A 69 68 68 67 67 67 68 68 69 6A 6B 6C 6D 6E 6F 70
```

Ja dat werkt. Nu nog de twee girders naast de oil can ook omhoog: Y-waarde veranderen van 80 naar 78.

```
02 F7 78 88 78 ; top right conveyor next to oil can  
02 77 78 08 78 ; top left conveyor next to oil can
```

En daarna de hele definitie verplaatsen. En de extra girders en ladders toevoegen. En dan de platformen afwisselend knipperen. En dan de items plaatsen en uitgebreid testen.

De hele conveyors definitie verplaatsen: #2672 t/m #26EA naar #2210 t/m #2288.

Aanroep ook aanpassen:

Aanroep aanpassen:

```
0CDF 111022 LD DE,#2210 ; load DE with start of table data for conveyors  
2461 211022 LD HL,#2210 ; load HL with start of table data for conveyors
```

Additionele girder over hele lengte:

```
02 F7 C0 08 C0 ; long middle conveyor
```

Trap rechtsonder gewoon maken:

01 DB D8 DB F8 veranderen in 00 DB D8 DB F8.

De ladders tussen de girders naast de platformen korter maken: tot het nieuwe lange platform.

```
#26E0: 01 1B C0 1B D8 ; middle broken ladder left side
```

Extra ladders maken boven het lange platform:

```
00 23 A0 23 C0 ; left ladder above long girder  
00 DB A0 DB C0 ; right ladder above long girder
```

00 7B A0 7B C0 ; middle ladder above long girder

De kruisen onder de oil can tekenen:

06 80 90 78 90
06 80 88 78 88
06 80 98 78 98

De platformen om en om tekenen. Eerst de code weer terughalen:

```
2610: 3A2762 LD A,(#6227) ; load A with screen number
2613: FE02 CP #02 ; are we on the conveyors?
2615: C21304 RET NZ ; no, return
;
2618: 3AA065 LD A,(#65A0) ; load A with blink counter
261B: 3C INC A ; increment blink counter
261C: 32A065 LD (#65A0), A ; store blink counter
;
261F: 47 LD B,A ; temporary store blink counter in B
2620: FE BA CP #BA ; is blink counter #BA? – effectively no sound !
2622: CA 2E 26 JP Z, #262E ; yes, play sound – jump label AAAA
;
2625: 78 LD A,B ; reload A with blink counter from B
2626: FE B0 CP #B0 ; is blink counter #B0?
2628: CA 32 26 JP Z, #2632 ; yes, change platform state - jump label BBBB
;
262B: C9 RET ; it is not time to blink yet, return
;
262C: 3E03 LD A,#03 ; load sound duration – label AAAA
262E: 328360 LD (#6083),A ; plays the coin insert sound
2631: C9 RET ; it is not time to blink yet, return
;
2632: 3A9B63 LD A,(#639B) ; load A with switch status – label BBBB
2635: FE01 CP #01 ; check if platform is visible
2637: CA 45 26 JP Z, #2645 ; yes, platform is visible – jump label CCCC
;
263A: 3C INC A ; make A #01: make platform visible
263B: 329B63 LD (#639B), A ; store switch status
263E: 3EB0 LD A,#B0 ; load A with girder character
2640: 0610 LD B,#10 ; load B with blank character
2642: C3 4D 26 JP, #264D ; skip next step – jump label DDDD
;
2645: 3D DEC A ; make A #00: make platform invisible – label CCCC
2646: 329B63 LD (#639B), A ; store switch status
2649: 3E10 LD A,#10 ; load A with blank character
264B: 06B0 LD B,#B0 ; load B with girder character
;
264D: 219476 LD HL,#7694 ; video memory address to write – label DDDD
2650: CD6C26 CALL #266C ; call draw routine – call label EEEE
2653: 211C75 LD HL,#751C ; video memory address to write
2656: CD6C26 CALL #266C ; call draw routine – call label EEEE
2659: 78 LD A,B ; load A with B
265A: 219C76 LD HL,#769C ; video memory address to write
265D: CD6C26 CALL #266C ; call draw routine – call label EEEE
2660: 211475 LD HL,#7514 ; video memory address to write
2663: CD6C26 CALL #266C ; call draw routine – call label EEEE
;
2666: 3E00 LD A,#00 ; A := #00
2668: 32A065 LD (#65A0),A ; store in blink counter
;
266B: C9 RET ; return
```

```

266C: 112000 LD DE,#0020 ; DE = #20; offset to next vertical line – label EEEE
;
266F: 77 LD (HL),A ; write memory address
2670: 19 ADD HL,DE ; add offset
2671: 77 LD (HL),A ; write memory address
2672: 19 ADD HL,DE ; add offset
2673: 77 LD (HL),A ; write memory address
2674: 19 ADD HL,DE ; add offset
2675: 77 LD (HL),A ; write memory address
;
2676: C9 RET ; return

```

Werkt prima.

Spawnen van fireballs aan beide kanten.

Dit wordt geregeld in dit stuk:

```

; This subroutine handles fireball movement as it spawns out of the oilcan on conveyors.
; Called from #32D2

3478 DD6E1A LD L,(IX+#1A) ; \ Load HL with address into Y-position table
347B DD661B LD H,(IX+#1B) ; /
347E AF XOR A ; \ Jump if HL is non-zero (i.e., if this is not the very first spawning frame)
347F 010000 LD BC,#0000 ; |
3482 ED4A ADC HL,BC ; |
3484 C29A34 JP NZ,#349A ; /

3487 21AC3A LD HL,#3AAC ; load HL with start of table data
348A 3A0362 LD A,(#6203) ; \ Jump if Mario is on left side of the screen, in this case we spawn the fireball on the left
348D CB7F BIT 7,A ; |
348F CAA834 JP Z,#34A8 ; |

3492 DD360D01 LD (IX+#0D),#01 ; Set fireball direction to "right"
3496 DD36037E LD (IX+#03),#7E ; Initialize X position to #7E

349A DD7E0D LD A,(IX+#0D) ; \ Jump if fireball moving left
349D FE01 CP #01 ; |
349F C2B334 JP NZ,#34B3 ; |

34A2 DD3403 INC (IX+#03) ; Moving right, Increment X-position
34A5 C34534 JP #3445 ; Jump back, remainder of subroutine shared with the above subroutine

34A8 DD360D02 LD (IX+#0D),#02 ; Set fireball direction to "special" left (This isn't actually used at all after spawning, since
; after spawning it will check to reverse rection and receive a direction of either "right" or "l"
34AC DD360380 LD (IX+#03),#80 ; Initialize X position to #80
34B0 C39A34 JP #349A ; Jump back [why there? after setting direction, we should jump directly to #34B3]

34B3 DD3503 DEC (IX+#03) ; Moving left, Decrement X-position
34B6 C34534 JP #3445 ; Jump back, remainder of subroutine shared with the above subroutine

```

Kant waarop spawnen is nu afhankelijk van de positie van jumpman (left or right gebaseerd op linker bit van de positie van Mario: 0 is links, 1 is rechts).

Eventueel RngTimer2 (=1960) gebruiken als pseudo random generator en dan testen op de rechter bit hiervan? Hopende dat RngTimer2 inderdaad voldoende varieert.
Dit kan eventueel door deze code:

348A 3A1960 LD A,(RngTimer2) ; Determine side to spawn fireball based on timer value
348D CB47 BIT 0,A ; If timer value is even . . .
348F CAA834 JP Z,#34A8 ; . . . jump and spawn the fireball at left side

Fireballs helemaal boven wel naar beneden laten gaan.

In het aanroepende stuk de aanroep weghalen. Alles een klein beetje terugschuiven en dan in geval dat de fireball omhoog wil aanroepen naar dit stuk.

```

334A CDA133 CALL #33A1 ; Return without taking ladder if fireball is on the top girder and the
; screen is not rivets
334A DD7E0F LD A,(IX+#0F) ; \ D := Y-position of bottom of fireball
334D C608 ADD A,#08 ; |
334F 57 LD D,A ; /
3350 DD7E0E LD A,(IX+#0E) ; A := fireball's X-position
3353 011500 LD BC,#0015 ; BC := #0015, the number of ladders to check
3356 CD6E23 CALL #236E ; Check for ladders nearby, return if none, else A := 0 if at bottom of
ladder, A := 1 if at top
3359 A7 AND A ; \ Jump if there is a ladder nearby to go up
335A CA9633 JP Z,#3396 ; / Changed call because of shifted code

; Else there is a ladder nearby to go down
335D DD701F LD (IX+#1F),B ; Store B into +#1F = Y-position of bottom of ladder
3360 CA9622 LD A,(#730E) ; \ Return without taking the ladder if Mario is at or above the Y-
position of the fireball
3363 47 LD B,A ; |
3364 DD7E0F LD A,(IX+#0F) ; | Allemal NOPS removed
3367 90 SUB B ; |
3368 D0 RST NC ; |

3369 DD360D04 LD (IX+#0D),#04 ; Else set direction to descending ladder
336D C9 RET ; return

; Arrived because fireball is moving up a ladder
336E DD7E0F LD A,(IX+#0F) ; \ Return if fireball is not at the top of the ladder
3371 C608 ADD A,#08 ; |
3373 DD461F LD B,(IX+#1F) ; |
3376 B8 CP B ; |
3377 C0 RET NZ ; |

; Fireball at top of ladder
3378 DD360D00 LD (IX+#0D),#00 ; Set fireball direction to left
337C DD7E19 LD A,(IX+#19) ; \ If freezer mode is engaged then set the freeze flag and return,
otherwise just return.
337F FE02 CP #02 ; |
3381 C0 RET NZ ; |
3382 DD361D01 LD (IX+#1D),#01 ; |
3386 C9 RET ; |

; Arrive because fireball is moving down a ladder
3387 DD7E0F LD A,(IX+#0F) ; \ Return if fireball is not at the bottom of the ladder
338A C608 ADD A,#08 ; |
338C DD461F LD B,(IX+#1F) ; |
338F B8 CP B ; |
3390 C0 RET NZ ; |

3391 DD360D00 LD (IX+#0D),#00 ; Fireball has reached the bottom, set the direction to left
3395 C9 RET ; return

; Arrive because there is a ladder nearby to go up
3396 7726 CALL #2677 ; Moved check prevent going up
3399 DD701F LD (IX+#1F),B ; Store B into +#1F = Y-position of top of ladder
339C DD360D08 LD (IX+#0D),#08 ; Else set direction to ascending ladder
33A0 C9 RET ; return

```

Groene stuk moeten NOPs zijn voor de eerdere wijziging 'In conveyor level firefoxes niet gericht naar jumpman laten gaan'.

Stuk van #33A1 t/m #33AC moet aangepast worden zodat de waarde van B behouden blijft wanneer de RST aangeroepen wordt. De RST past namelijk de B aan, waardoor de Y-waarde van de bovenkant van de ladder niet goed gezet wordt (die zit namelijk in B). De fireballs blijven daardoor naar boven lopen wanneer aan bovenkant ladder, verdwijnen boven uit scherm en verschijnen onder weer. Alleen deze aanpassing past daar niet. Daarom ergens anders neerzetten:

```

2677: C5 PUSH BC ; save B for later
2678: 3E07 LD A,#07 ; \ Return if immediately we are on rivets, fireballs do
267A: F7 RST #30 ; / not get stuck on the top in this case
267B: C1 POP BC ; restore B
;
267C: DD7E0F LD A,(IX+#0F) ; \ Return if Y-position is >= 59
267F: FE59 CP #59 ; | (i.e., fireball is not on the top girder)
2681: D0 RET NC ; /

2682: 33 INC SP ; \ Else return to higher subroutine.

```

```

2683: 33      INC   SP      ; |
2684: C9      RET      ; /

```

Vlak ervoor wordt op twee plekken naar plekken binnen de verschoven code gespongen: dit moet aangepast worden:

333D	DD7E0D	LD	A, (IX+#0D)	; \ Jump if fireball is climbing up a ladder
3340	FE08	CP	#08	;
3342	CA 6E33	JP	Z, # 336E	; / Was #3371
3345	FE04	CP	#04	; \ Jump if fireball is climbing down a ladder
3347	CA 8733	JP	Z, # 3387	; / Was #338A

Veld toch iets anders maken. Maar eerst huidige veld eventjes uitwerken:



```

00 63 18 63 58 ; ladder next to kong (right of two)
00 53 18 53 58 ; ladder next to kong (left of two)
00 93 38 93 58 ; ladder leading to girder with pauline
02 97 38 68 38 ; girder where Pauline stands on
02 EF 58 10 58 ; girder where kong stands on
02 EF F8 10 F8 ; bottom girder where jumpman starts
02 28 E0 10 E0 ; girder left - 1st level
00 23 E0 23 F8 ; ladder left - bottom to 1st level
02 F7 78 88 78 ; girder right side of oil can
02 77 78 08 78 ; girder left side of oil can
00 E3 78 E3 A0 ; ladder right - 3rd level to girder next oil can
00 D3 58 D3 78 ; ladder leading to girder with kong
02 EF A0 D0 A0 ; girder right - 3rd level
02 88 E0 70 E0 ; girder middle - 1st level
02 EF E0 D0 E0 ; girder right - 1st level
00 E3 C0 E3 E0 ; ladder right - 1st level to 2nd level
02 88 A0 70 A0 ; girder middle - 3rd level under oil can
02 28 A0 10 A0 ; girder left 3rd level
01 73 78 73 A0 ; broken ladder left of oil can
01 8A 78 8A A0 ; broken ladder right of oil can
00 83 C0 83 E0 ; ladder middle - 1st level to 2nd level
00 1B 78 1B A0 ; ladder left - 3rd level to girder next oil can
01 1B C0 1B E0 ; broken ladder left - 1st level to 2nd level
00 DB E0 DB F8 ; ladder right - bottom to 1st level
02 F7 C0 08 C0 ; long girder 2nd level
00 23 A0 23 C0 ; ladder left - 2nd level to 3rd level
00 DB A0 DB C0 ; ladder right - 2nd level to 3rd level
00 7B A0 7B C0 ; ladder middle - 2nd level to 3rd level

```

```

06 80 90 78 90 ; patch of X's under oil can
06 80 88 78 88 ; patch of X's under oil can
06 80 98 78 98 ; patch of X's under oil can
AA

```

Nieuwe veld (vanaf #2210):

```

00 63 18 63 58 ; ladder next to kong (right of two)
00 53 18 53 58 ; ladder next to kong (left of two)
00 93 38 93 58 ; ladder leading to girder with pauline
02 97 38 68 38 ; girder where Pauline stands on
02 EF 58 10 58 ; girder where kong stands on
02 EF F8 10 F8 ; bottom girder where jumpman starts
02 28 E0 10 E0 ; girder left - 1st level
00 23 E0 23 F8 ; ladder left - bottom to 1st level
02 F7 78 88 78 ; girder right side of oil can
02 77 78 08 78 ; girder left side of oil can
00 E3 78 E3 A0 ; ladder right - 3rd level to girder next oil can
00 D3 58 D3 78 ; ladder leading to girder with kong
02 EF A0 D0 A0 ; girder right - 3rd level
02 88 E0 70 E0 ; girder middle - 1st level
02 EF E0 D0 E0 ; girder right - 1st level
02 88 A0 70 A0 ; girder middle - 3rd level under oil can
02 28 A0 10 A0 ; girder left 3rd level
01 73 78 73 A0 ; broken ladder left of oil can
01 8A 78 8A A0 ; broken ladder right of oil can
00 8B A0 8B E0 ; ladder middle - 1st level to 3rd level
00 1B 78 1B A0 ; ladder left - 3rd level to girder next oil can
01 1B C0 1B E0 ; broken ladder left - 1st level to 2nd level
00 DB E0 DB F8 ; ladder right bottom to 1st level
00 E3 C0 E3 E0 ; ladder right - 1st level to 2nd level
02 60 C0 08 C0 ; left girder 2nd level
00 23 A0 23 C0 ; ladder left - 2nd level to 3rd level
01 DB A0 DB C0 ; broken ladder right - 2nd level to 3rd level
00 73 A0 73 E0 ; ladder middle - 1st level to 3rd level
06 80 90 78 90 ; patch of X's under oil can
06 80 88 78 88 ; patch of X's under oil can
06 80 98 78 98 ; patch of X's under oil can
02 F7 C0 98 C0 ; right girder 2nd level
02 80 C0 78 C0 ; middle girder 2nd level
AA

```

De platformen korter maken.

```

266C: 112000 LD DE,#0020 ; DE = #20; offset to next vertical line - label EEEE
;
266F: 77 LD (HL),A ; write memory address
2670: 19 ADD HL,DE ; add offset
2671: 77 LD (HL),A ; write memory address
2672: 19 ADD HL,DE ; add offset
2673: 77 LD (HL),A ; write memory address
2674: 19 ADD HL,DE ; add offset
2675: 77 LD (HL),A ; write memory address
;
2676: C9 RET ; return

```

De return eventjes plaatsen op #2672.

De platformen moeten verschoven worden:

```

264D: 217476 LD HL,#7674 ; video memory address to write - label DDDD
2650: CD6C26 CALL #266C ; call draw routine - call label EEEE
2653: 217C75 LD HL,#757C ; video memory address to write

```

```

2656: CD6C26 CALL #266C ; call draw routine - call label EEEE
2659: 78 LD A,B ; load A with B
265A; 217C76 LD HL,#767C ; video memory address to write
265D: CD6C26 CALL #266C ; call draw routine - call label EEEE
2660; 217475 LD HL,#7574 ; video memory address to write
2663: CD6C26 CALL #266C ; call draw routine - call label EEEE

```

De girders naast de platformen langer maken (vanaf #2210):

```

00 63 18 63 58 ; ladder next to kong (right of two)
00 53 18 53 58 ; ladder next to kong (left of two)
00 93 38 93 58 ; ladder leading to girder with pauline
02 97 38 68 38 ; girder where Pauline stands on
02 EF 58 10 58 ; girder where kong stands on
02 EF F8 10 F8 ; bottom girder where jumpman starts
02 48 E0 10 E0 ; girder left - 1st level
00 23 E0 23 F8 ; ladder left - bottom to 1st level
02 F7 78 88 78 ; girder right side of oil can
02 77 78 08 78 ; girder left side of oil can
00 E3 78 E3 A0 ; ladder right - 3rd level to girder next oil can
00 D3 58 D3 78 ; ladder leading to girder with kong
02 EF A0 B0 A0 ; girder right - 3rd level
02 88 E0 70 E0 ; girder middle - 1st level
02 EF E0 B0 E0 ; girder right - 1st level
02 88 A0 70 A0 ; girder middle - 3rd level under oil can
02 48 A0 10 A0 ; girder left 3rd level
00 8B A0 8B E0 ; ladder middle - 1st level to 3rd level
00 1B 78 1B A0 ; ladder left - 3rd level to girder next oil can
01 1B C0 1B E0 ; broken ladder left - 1st level to 2nd level
00 E3 C0 E3 E0 ; ladder right - 1st level to 2nd level
02 60 C0 08 C0 ; left girder 2nd level
00 23 A0 23 C0 ; ladder left - 2nd level to 3rd level
01 DB A0 DB C0 ; broken ladder right - 2nd level to 3rd level
00 73 A0 73 E0 ; ladder middle - 1st level to 3rd level
06 80 90 78 90 ; patch of X's under oil can
06 80 88 78 88 ; patch of X's under oil can
06 80 98 78 98 ; patch of X's under oil can
02 F7 C0 98 C0 ; right girder 2nd level
02 80 C0 78 C0 ; middle girder 2nd level
AA

```

De platformen sneller laten knipperen:

```

2625: 78 LD A,B ; reload A with blink counter from B
2626: FE 90 CP #90 ; is blink counter #90?
2628: CA 32 26 JP Z, #2632 ; yes, change platform state - jump label BBBB

```

Ziet er nu zo uit:



Alleen vreemd dat jumpman naast het middelste platform tussen de twee lange ladders kan staan. De code ziet de ladder als verlenging van het platform.

Wellicht op te lossen door er een broken ladder van te maken? Ja dat gaat goed. Alleen dan nog wel voor het zicht een stuk extra ladder erbij tekenen.

Nog doen:

Plaatsen items en hamers

Plaatsen firefoxes

Plaatsen additionele trapdelen in middenstuk

Verwijderen trapdelen van retractable ladders

Plaatsen items:

; bonus items on conveyors

```
3E3C 53 73 0A A0      ; position of hat on pies is 53,A0
3E40 8B 74 0A F0      ; position of purse on pies is 8B,F0
3E44 DB 75 0A A0      ; umbrella on the pies is at DB,A0
```

Aanpassen naar:

; bonus items on conveyors

```
3E3C 53 73 0A 70      ; position of hat on pies is 53,A0
3E40 BB 74 0A F0      ; position of purse on pies is 8B,F0
3E44 3B 75 0A 98      ; umbrella on the pies is at DB,A0
```

Plaatsen hamers:

3E10: 23 8D 7B B4 ; for conveyors

Aanpassen naar:

3E10: C3 60 7B E8 ; for conveyors

Ziet er nu zo uit:



Plaatsen van een firefox (voorbeeld van barrels screen waar dat ook gedaan wordt):

```

3F33 DD210064 LD    IX,#6400      ; load IX with start of fire #1
3F37 DD360001 LD    (IX+#00),#01  ; set fire active
3F3B DD3603D8 LD    (IX+#03),#D8  ; set fire X position
3F3F DD360ED8 LD    (IX+#0E),#D8  ; set fire X position #2
3F43 DD3605C8 LD    (IX+#05),#C8  ; set fire Y position
3F47 DD360FC8 LD    (IX+#0F),#C8  ; set fire Y position #2

```

Iets soortgelijks doen voor de firefox in het middendeel:

```

2690: DD 21 00 64 LD    IX,#6400      ; load IX with start of fire #1
2694: DD 36 00 01 LD    (IX+#00),#01  ; set fire active
2698: DD 36 03 78 LD    (IX+#03),#78  ; set fire X position
269C: DD 36 0E 78 LD    (IX+#0E),#78  ; set fire X position #2
26A0: DD 36 05 98 LD    (IX+#05),#98  ; set fire Y position
26A4: DD 36 0F 98 LD    (IX+#0F),#98  ; set fire Y position #2
;
26A8: C9             RET           ; return

```

Dit ergens tussenvoegen en dan aanroepen bij tekenen van het conveyorsscherm:

```
105A: CD9026 CALL #2690 ; call drawing of the firefox
```

Nog 2 firexes al standard spawnen:

```

26A8: DD 36 20 01 LD    (IX+#20),#01  ; set fire active
26AC: DD 36 23 40 LD    (IX+#23),#40  ; set fire X position
26B0: DD 36 2E 40 LD    (IX+#2E),#40  ; set fire X position #2
26B4: DD 36 25 B8 LD    (IX+#25),#B8  ; set fire Y position
26B8: DD 36 2F B8 LD    (IX+#2F),#B8  ; set fire Y position #2
;
26BC: DD 36 40 01 LD    (IX+#40),#01  ; set fire active
26C0: DD 36 43 B8 LD    (IX+#43),#B8  ; set fire X position
26C4: DD 36 4E B8 LD    (IX+#4E),#B8  ; set fire X position #2
26C8: DD 36 45 B8 LD    (IX+#45),#B8  ; set fire Y position
26CC: DD 36 4F B8 LD    (IX+#4F),#B8  ; set fire Y position #2
;
26D0: C9             RET           ; return

```

Alle trappen heel maken (behalve de trappen in het middengedeelte).

Extra trap

00 DB E0 DB F8 ; ladder right – 1st level to 2nd level

Eigenlijk fireball ook niet omhoog naar de girder van Kong ! Daartoe moet de waarde van de test aangepast worden:

```
2677: C5      PUSH BC          ; save B for later
2678: 3E07    LD   A,#07        ; \ Return if immediately we are on rivets, fireballs do
267A: F7      RST   #30        ; /     not get stuck on the top in this case
267B: C1      POP BC          ; restore B
;
267C: DD7E0F  LD   A,(IX+#0F) ; \ Return if Y-position is >= 79
267F: FE79    CP   #79          ; |     (i.e., fireball is not on the top girder)
2681: D0      RET   NC          ; /
;
2682: 33      INC   SP          ; \ Else return to higher subroutine.
2683: 33      INC   SP          ; |
2684: C9      RET   ;           ; /
```

Hamers verplaatsen:

3E10: E3 60 BB 88 ; for conveyors

Extra trap rechts bovenin maken:

00 C3 78 C3 A0 ; ladder right – 3rd level to girder next oil can

Bovenste hame verplaatsen:

3E10: 3B A8 BB 88 ; for conveyors

De umbrella bovenop het middenstuk en de purse onderop het middenstuk:

```
3E3C 53 73 0A 70          ; position of hat
3E40 80 74 0A D8          ; position of purse
3E44 78 75 0A 98          ; position of umbrella
```

Middelste ladders klimbaar maken:

```
00 63 18 63 58 ; ladder next to kong (right of two)
00 53 18 53 58 ; ladder next to kong (left of two)
00 93 38 93 58 ; ladder leading to girder with pauline
02 97 38 68 38 ; girder where Pauline stands on
02 EF 58 10 58 ; girder where kong stands on
02 EF F8 10 F8 ; bottom girder where jumpman starts
02 48 E0 10 E0 ; girder left - 1st level
00 23 E0 23 F8 ; ladder left - bottom to 1st level
02 F7 78 88 78 ; girder right side of oil can
02 77 78 08 78 ; girder left side of oil can
00 E3 78 E3 A0 ; ladder right – 3rd level to girder next oil can
00 D3 58 D3 78 ; ladder leading to girder with kong
02 EF A0 B0 A0 ; girder right – 3rd level
02 88 E0 70 E0 ; girder middle – 1st level
02 EF E0 B0 E0 ; girder right – 1st level
02 88 A0 70 A0 ; girder middle – 3rd level under oil can
02 48 A0 10 A0 ; girder left 3rd level
00 8B C0 8B E0 ; ladder middle – 1st level to 3rd level
00 1B 78 1B A0 ; ladder left – 3rd level to girder next oil can
```

```

01 1B C0 1B E0 ; broken ladder left - 1st level to 2nd level
00 E3 C0 E3 E0 ; ladder right - 1st level to 2nd level
02 58 C0 08 C0 ; left girder 2nd level
00 23 A0 23 C0 ; ladder left - 2nd level to 3rd level
01 DB A0 DB C0 ; broken ladder right - 2nd level to 3rd level
00 73 A0 73 C0 ; ladder middle - 1st level to 3rd level
06 80 90 78 90 ; patch of X's under oil can
06 80 88 78 88 ; patch of X's under oil can
06 80 98 78 98 ; patch of X's under oil can
02 F7 C0 A0 C0 ; right girder 2nd level
02 80 C0 78 C0 ; middle girder 2nd level
AA

```

Het blijkt dat de rivets niet goed meer speelt, geeft een reset. Vanaf welke versie is dat? Testen.

V0.3.0 is nog OK, V0.3.4 is nog OK, V0.3.7 is fout, v0.3.6 is fout, v0.3.5 is fout.

Gaat dus fout vanaf het voorkomen dat de firefoxes op de bovenste regel naar beneden kunnen, is stuk code op pagina 40.

```

2677: C5      PUSH   BC          ; save B for later
2678: 3E07    LD     A,#07       ; \ Return if immediately we are on rivets, fireballs do
267A: F7      RST    #30         ; / not get stuck on the top in this case
267B: C1      POP    BC          ; restore B
;
267C: DD7E0F  LD     A,(IX+#0F) ; \ Return if Y-position is >= 59
267F: FE59    CP     #79         ; | (i.e., fireball is not on the top girder)
2681: D0      RET    NC          ; /
;
2682: 33      INC    SP          ; \ Else return to higher subroutine.
2683: 33      INC    SP          ; |
2684: C9      RET    ; / 

```

Het probleem is waarschijnlijk dat een push ook de stack aanpas en daardoor de RST beïnvloed wordt. Dus de B eigenlijk anders bewaren (niet met PUSH en POP).

```

2677: 78      LD A,B          ; save B for later
2678: 4F      LD C,A          ;
2679: 3E07    LD     A,#07       ; \ Return if immediately we are on rivets, fireballs do
267B: F7      RST    #30         ; / not get stuck on the top in this case
267C: 79      LD A,C          ; restore B
267D: 47      LD B,A          ;
;
267E: DD7E0F  LD     A,(IX+#0F) ; \ Return if Y-position is >= 59
2681: FE79    CP     #79         ; | (i.e., fireball is not on the top girder)
2683: D0      RET    NC          ; /
;
2684: 33      INC    SP          ; \ Else return to higher subroutine.
2685: 33      INC    SP          ; |
2686: C9      RET    ; / 

```

Dit werkt wel goed in de conveyors stage. In de rivets stage nu geen reset, maar daar blijft de firefox aan het einde van de trap doorlopen: dus dan de B ook niet teruggezet. Dat moet dan gebeuren in de RST30 code.

```

;
; RST #30
;
```

```
0030 1812 JR #0044 ; this core sub is actually at #0044
```

Is alleen een verwijzing naar stuk code op #0044:

```
; continuation of rst #30  
; used to check a screen number. if it doesn't match, the 2nd level of subroutine is returned  
; A is preloaded with the check value, in binary
```

```
0044 212762 Id HL,#6227 ; Load HL with address of Screen #  
0047 46 Id B,(HL) ; load B with Screen #, For B = 1 to screen # (1, 2, 3 or 4)  
  
0048 0F RRCA ; Rotate A right with carry  
0049 10Fd DJNZ #0048 ; Next B  
  
004B d8 RET c ; return if carry  
  
004C E1 POP HL ; otherwise HL gets the stack = return to higher subroutine  
004D C9 RET ; return
```

Aanpassen:

```
0044: C3E026 JP #26E0 ; jump to additional code
```

en

```
26E0: 212762 Id HL,#6227 ; Load HL with address of Screen #  
26E3: 46 Id B,(HL) ; load B with Screen #, For B = 1 to screen # (1, 2, 3 or 4)  
;  
26E4: 00480F RRCA ; Rotate A right with carry  
26E7: 10Fd DJNZ #0048 ; Next B  
;  
26E9: d8 RET c ; return if carry  
;  
26EA: 79 LD A,C ; restore B  
26EB: 47 LD B,A  
;  
26EC: E1 POP HL ; otherwise HL gets the stack = return to higher subroutine  
26ED: C9 RET ; return
```

Dat helpt niet: dus terugzetten.

```
; Arrive because there is a ladder nearby to go up  
3396 CD7726 CALL #2677 ; Moved check prevent going up  
3399 DD701F LD (IX+#1F),B ; Store B into +#1F = Y-position of top of ladder  
339C DD360D08 LD (IX+#0D),#08 ; Else set direction to ascending ladder  
33A0 C9 RET ; return
```

Je zou zeggen dat de B in 3399 gebruikt wordt. Dus daarvoor terugzetten uit de gesavede C waarde:

```
3396: CD7726 CALL #2677 ; Moved check prevent going up  
3399: C3E026 JP #26E0 ; jump to additional code
```

en

```
26E0: 79 LD A,C ; restore B  
26E1: 47 LD B,A  
26E2: DD701F LD (IX+#1F),B ; Store B into +#1F = Y-position of top of ladder  
26E5: DD360D08 LD (IX+#0D),#08 ; Else set direction to ascending ladder  
26E9: C9 RET ; return
```

En nu werkt het goed in de rivets en in de conveyors.

De extra trapdelen nog weghalen.

Tekenen van een character vanuit het stuk dat de platformen ook tekent:

```
264D: 219476 LD HL,#7694 ; video memory address to write - label DDDD
2650: CD6C26 CALL #266C ; call draw routine - call label EEEE
2653: 211C75 LD HL,#751C ; video memory address to write
2656: CD6C26 CALL #266C ; call draw routine - call label EEEE
2659: 78 LD A,B ; load A with B
265A: 219C76 LD HL,#769C ; video memory address to write
265D: CD6C26 CALL #266C ; call draw routine - call label EEEE
2660: 211475 LD HL,#7514 ; video memory address to write
2663: CD6C26 CALL #266C ; call draw routine - call label EEEE
;
2666: 3E00 LD A,#00 ; A := #00
2668: 32A065 LD (#65A0),A ; store in blink counter
;
266B: C9 RET ; return
```

Veranderen in:

```
264D: 219476 LD HL,#7694 ; video memory address to write - label DDDD
2650: CD6C26 CALL #266C ; call draw routine - call label EEEE
2653: 211C75 LD HL,#751C ; video memory address to write
2656: CD6C26 CALL #266C ; call draw routine - call label EEEE
2659: 78 LD A,B ; load A with B
265A: 219C76 LD HL,#769C ; video memory address to write
265D: CD6C26 CALL #266C ; call draw routine - call label EEEE
2660: 211475 LD HL,#7514 ; video memory address to write
2663: CD6C26 CALL #266C ; call draw routine - call label EEEE
;
2666: C3D126 JP #26D1 ; jump to additional code
```

en

```
26D1: 3E00 LD A,#00 ; A := #00
26D3: 32A065 LD (#65A0),A ; store in blink counter
26D6: 3E10 LD A,#10 ; A := #FE
26D8: 326E77 LD (#776E),A ; store in blink counter
26DB: 328E74 LD (#748E),A ; store in blink counter
;
26DE: C9 RET ; return
;
```

Werkt wel, maar worden erg laat weggehaald.

Terugzetten.

Misschien beter om hier te doen:

```
2618: 3AA065 LD A,(#65A0) ; load A with blink counter
261B: 3C INC A ; increment blink counter
261C: 32A065 LD (#65A0), A ; store blink counter
;
261F: 47 LD B,A ; temporary store blink counter in B
2620: FE 50 CP #50 ; is blink counter #50?
2622: CA 2E 26 JP Z, #262E ; yes, play sound - jump label AAAA
```

Veranderen in:

```
2618: C3D126 JP #26D1 ; jump to additional code
```

en

```
26D1: 3E10 LD A,#10 ; A := #FE
26D3: 326E77 LD (#776E),A ; store in blink counter
26D6: 328E74 LD (#748E),A ; store in blink counter
26D9: 3AA065 LD A,(#65A0) ; load A with blink counter
26DC: C31B26 JP #261B ; jump back
```

Dat werkt wel, maar toch nog eerst zichtbaar.

De retractable ladders worden hier aangepast:

```
0CD1 C3A03F JP #3FA0 ; fix retractable ladders for pie factory and returns to
                        ; #0D5F. [orig code was JP #0D5F ?]
```

en

```
3FA0 CDA63F CALL #3FA6 ; call sub below
3FA3 C35F0D JP #0D5F ; return to program [this was original line wiped by patch ?]
```

; called from #3FA0 above

```
3FA6 3E02 LD A,#02 ; A := 2
3FA8 F7 RST #30 ; check to see if the level is pie factory

3FA9 0602 LD B,#02 ; for B = 1 to 2
3FAB 216C77 LD HL,#776C ; load HL with video RAM address for top retractable ladder

3FAE 3610 LD (HL),#10 ; clear the top of the ladder
3FB0 23 INC HL
3FB1 23 INC HL ; next address
3FB2 36C0 LD (HL),#C0 ; draw a ladder 2 rows down
3FB4 218C74 LD HL,#748C ; set HL for next loop - does the other side of the screen
3FB7 10F5 DJNZ #3FAE ; Next B

3FB9 C9 RET ; return [to #3FA3, then jump to #0D5F]
```

Dit lijkt helemaal niet nodig. Dan kan de sprong naar 3FA0 vervangen worden door de oorspronkelijke sprong naar 0D5F:

```
0CD1 C35F0D JP #0D5F
```

Dat werkt goed. De eerdere wijzigingen t.a.v. het overschrijven van de graphic in het videomemory is niet benodigd.

Nog het level weer terugzetten:

```
0BF2 218A60 LD HL,#608A ; load HL with tune address
0BF5 3602 LD (HL),#02 ; play how high can you get sound?
0BF7 23 INC HL ; HL := #608B . load HL with music timer ?
0BF8 3603 LD (HL),#03 ; set to 3 units
```

De attract mode aanpassen zodat deze klopt met het nieuwe barrels scherm.

; data used in sub below for attract mode movement
; first byte is movement, second is duration

#21D1:

```
80 ; jump
CE 01 ; run right
10 81 ; jump right
20 01 ; run right
```

```
80 04 ; up = climb ladder
50 02 ; run left
10 82 ; jump left
60 02 ; run left
10 82 ; jump left
CA 01 ; run right
10 81 ; jump right (gets hammer)
FF 02 ; run left
38 01 ; run right - mario dies falling over right edge
80 02 ; run left
FF 04 ; up
80 04 ; up
80      ; ?
```

HIGH SCORE aanpassen naar TWIST:

#36B2: 80 76 10 10 10 24 27 19 23 24 10 10 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 #3F18 de titel maken:

TWISTED

6E 76 24 27 19 23 24 15 14 3F

En de verwijzing ernaar toe aanpassen:

3689: 18 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 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

```

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 14 .....A.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:

```

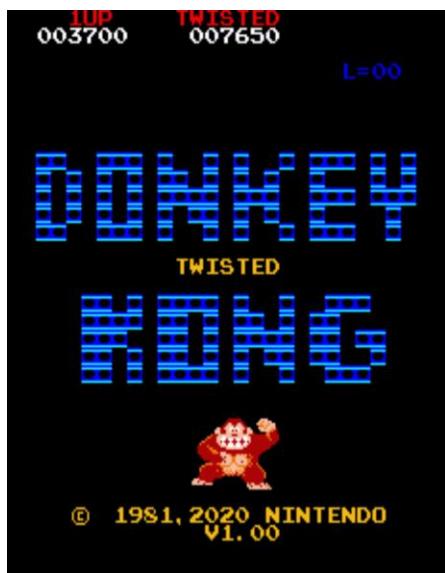
080C 111E03 LD DE,#031E ; load task data for text "(C) 1981"
080F CD9F30 CALL #309F ; insert task to draw text
0812 13 INC DE ; load task data for text "NINTENDO OF AMERICA"
0813 CD9F30 CALL #309F ; insert task to draw text
0816 21CF39 LD HL,#39CF ; load HL with table data for kong beating chest
0819 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: C3223F JP #3F22 ; jump to additional code to display version  
3F22: 110103 LD DE,#0301 ; load task data for text "V1.00"  
3F25: CD9F30 CALL #309F ; insert task to draw text  
3F28: C31F08 JP #081F ; jump back
```

Het titelscherm ziet er met versie nu zo uit:



Testen met verschillende screens met level 05:

Aanpassen start level:

```
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.

Springs is goed / Rivets is te moeilijk / Barrels is lastig / Pies is te makkelijk.

Ideeën:

- Bij de barrels de korte trappen naar het uiteinde van de girder verplaatsen?
- Bij de rivets de gebroken trappen kiezen uit een groep (daardoor meer random)?
- Bij de rivets de rivets terug laten groeien, zodra terug: nog een keer verwijderen?
- Bij de rivets, de rivets één keer verwijderen en daarna terug en dan nog een keer doen?
- Bij de pies ander veld samenstellen: route met springen en dan obstakels inbrengen?
- Bij de pies wellicht een free floating iets, dat achter jumpman aangaat?
- Bij de pies wellicht een mogelijkheid tot warpen?

De fireballs gaan nu ook naar beneden als je hoger of op gelijke hoogte met ze bent. Dat is niet zoals de echte spelers het gewend zijn. Is voor deze hack ook niet echt noodzakelijk dus dat weer terugdraaien:

Dit lijkt geregeld te worden door dit stuk:

```

; Else there is a ladder nearby to go down
3360 DD701F LD (IX+#1F),B ; Store B into +#1F = Y-position of bottom of ladder
3363 3A0562 LD A,(#6205) ; \ Return without taking the ladder if Mario is at or above the Y-position of the fireball
3366 47 LD B,A ; |
3367 DD7E0F LD A,(IX+#0F) ; |
336A 90 SUB B ; |
336B D0 RET NC ; /
336C DD360D04 LD (IX+#0D),#04 ; Else set direction to descending ladder
3370 C9 RET ; return

```

Dit regelt bijvoorbeeld dat je in de elevators screen een stukje de linker ladder moet afdalen om de fireball ernaast ook een van de twee ladders te laten nemen.

Je zou deze beperking kunnen verwijderen door 3363 t/m 336B in NOP's te veranderen. Nu dus weer herstellen. Gedaan!

De rivets stages makkelijker maken: maar twee trappen broken maken: de twee center trappen.

Rivets definities: #3C0D en #3C72.

3C8B: 00 7B 80 7B A8	center ladder level 3	X	
3C90: 00 7B D0 7B F8	bottom center ladder	X	
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	X	
3CAE: 00 D3 80 D3 A8	level 3 ladder right side	X	
3CB3: 00 23 A8 23 D0	level 2 ladder left side	X	
3CB8: 00 5B A8 5B D0	level 2 ladder #2 of 4	X	
3CBD: 00 A3 A8 A3 D0	level 2 ladder #3 of 4	X	X
3CC2: 00 DB A8 DB D0	level 2 ladder right side		
3CC7: 00 1B D0 1B F8	bottom left ladder		
3CCC: 00 E3 D0 E3 F8	bottom right ladder		
3CD1: 05 B7 30 48 30	girder above kong		
3CD6: 05 CF 58 30 58	girder kong stands on		
3CDB: 05 D7 80 28 80	level 4 girder		
3CE0: 05 DF A8 20 A8	level 3 girder		
3CE5: 05 E7 D0 18 D0	level 2 girder		
3CEA: 05 EF F8 10 F8	bottom level girder		
3CEF: AA	end code		

Het eerste veld is aangepast. Heeft nu twee broken ladders en is nog lastig genoeg. Het andere veld toch gaan omdraaien: en dan de items aanpassen en ook de spawn plekken aanpassen:

3C8B: 00 7B 80 7B A8	center ladder level 3	X	
3C90: 00 7B D0 7B F8	bottom center ladder	X	
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	X	
3CAE: 00 D3 80 D3 A8	level 3 ladder right side	X	
3CB3: 00 23 A8 23 D0	level 2 ladder left side	X	
3CB8: 00 5B A8 5B D0	level 2 ladder #2 of 4	X	
3CBD: 00 A3 A8 A3 D0	level 2 ladder #3 of 4	X	X
3CC2: 00 DB A8 DB D0	level 2 ladder right side		
3CC7: 00 1B D0 1B F8	bottom left ladder		

3CCC: 00 E3 D0 E3 F8	bottom right ladder		
3CD1: 05 B7 30 48 30	girder above kong		
3CD6: 05 CF 58 30 58	girder kong stands on		
3CDB: 05 D7 80 28 80	level 4 girder		
3CE0: 05 DF A8 20 A8	level 3 girder		
3CE5: 05 E7 D0 18 D0	level 2 girder		
3CEA: 05 EF F8 10 F8	bottom level girder		
3CEF: AA	end code		

De girder breedtes aanpassen:

3CB8: 05 B7 30 48 30	girder above kong	05 B7 30 48 30
3CBD: 05 CF 58 30 58	girder kong stands on	05 E7 58 18 58
3CC2: 05 D7 80 28 80	level 4 girder	05 DF 80 20 80
3CC7: 05 DF A8 20 A8	level 3 girder	05 D7 A8 28 A8
3CCC: 05 E7 D0 18 D0	level 2 girder	05 CF D0 30 D0
3CD1: 05 EF F8 10 F8	bottom level girder	05 EF F8 10 F8
3CD6:	Extra side girder left	05 18 D0 10 D0
3CDB:	Extra side girder right	05 EF D0 E3 D0

En dan nu nog de trappen goedzetten:

3C72: 00 7B 80 7B A8	center ladder level 3	00 7B 80 7B A8
3C77: 00 7B D0 7B F8	bottom center ladder	00 7B D0 7B F8
3C7C: 00 33 58 33 80	top left ladder	00 23 58 23 80
3C81: 00 53 58 53 80	top left ladder (right side)	00 53 58 53 80
3C86: 00 AB 58 AB 80	top right ladder (left side)	00 AB 58 AB 80
3C8B: 00 CB 58 CB 80	top right ladder	00 DB 58 DB 80
3C90: 00 2B 80 2B A8	level 3 ladder left side	00 2B 80 2B A8
3C95: 00 D3 80 D3 A8	level 3 ladder right side	00 D3 80 D3 A8
3C9A: 00 23 A8 23 D0	level 2 ladder left side	00 33 A8 33 D0
3C9F: 00 5B A8 5B D0	level 2 ladder #2 of 4	00 5B A8 5B D0
3CA4: 00 A3 A8 A3 D0	level 2 ladder #3 of 4	00 A3 A8 A3 D0
3CA9: 00 DB A8 DB D0	level 2 ladder right side	00 CB A8 CB D0
3CAE: 00 1B D0 1B F8	bottom left ladder	00 3B D0 3B F8
3CB3: 00 E3 D0 E3 F8	bottom right ladder	00 C3 D0 C3 F8

Die staan nu goed. Alleen de fireballs spawnen nu soms buiten de girders en dan blijven ze op hun plaats hangen. En de hat item bevindt zich buiten de girders, zodat je deze niet te pakken kunt krijgen.

Spawnen van de fireballs op level 2, 3 en 4 aanpassen? Allemaal boven elkaar. Op de bottom level spawnen gaat goed dus dat niet aanpassen. Tabel met spawn posities is hier:

```

; Possible X and Y positions to spawn a fireball on the right side of the screen
; First value is X position, 2nd value is Y position

; 3AC4: EE F0 ; bottom, right
; 3AC6: DB A0 ; middle, right
; 3AC8: E6 C8 ; 2nd from bottom, right
; 3ACA: D6 78 ; 2nd from top, right
; 3ACC: EB F0 ; unused?
; 3ACE: DB A0 ; unused?
; 3AD0: E6 C8 ; unused?
; 3AD2: E6 C8 ; unused?

; Possible X and Y positions to spawn a fireball on the left side of the screen
; First value is X position, 2nd value is Y position

; 3AD4: 1B C8 ; 2nd from bottom, left
; 3AD6: 23 A0 ; middle, left
; 3AD8: 2B 78 ; 2nd from top, left
; 3ADA: 12 F0 ; bottom, left
; 3ADC: 1B C8 ; unused?
; 3ADE: 23 A0 ; unused?
; 3AE0: 12 F0 ; unused?
; 3AE2: 1B C8 ; unused?

```

; table data referenced in #34C7

3AC4: EE F0 D0 A0 D0 C8 D0 78 EB F0 D0 A0 D0 C8 D0 C8

; table data referenced in #34ED

3AD4: 30 C8 30 A0 30 78 12 F0 30 C8 30 A0 12 F0 30 C8

Right allemaal (behalve bottom) spawnen op D0
Left allemaal (behalve bottom) spawnen op 2B

Is aangepast. Spawnen nu wel goed.

Nu de hat-item nog verplaatsen. Items op rivets definitie:

```

; bonus items for rivets

3E54 DB 73 0A C8          ; hat on rivets at DB,C8
3E58 93 74 0A F0          ; purse on rivets at 93,F0
3E5C 33 75 0A 50          ; umbrella on rivets at 33,50

```

Hat naar rechts verplaatsen:

3E54: E3 73 0A C8

Nu de andere rivets aanpassen:

Alleen de bottom center trap broken houden.

De level 3 girder omhoog: kan niet spawnen gaat dan niet goed.
Ladders verplaatsen.

3C0D: 00 7B 80 7B A8	center ladder level 3	00 7B 80 7B A8
3C12: 00 7B D0 7B F8	bottom center ladder	
3C17: 00 33 58 33 80	top left ladder	00 3B 58 3B 80
3C1C: 00 53 58 53 80	top left ladder (right side)	00 5B 58 5B 80
3C21: 00 AB 58 AB 80	top right ladder (left side)	00 A3 58 A3 80
3C26: 00 CB 58 CB 80	top right ladder	00 C3 58 C3 80
3C2B: 00 2B 80 2B A8	level 3 ladder left side	
3C30: 00 D3 80 D3 A8	level 3 ladder right side	
3C35: 00 23 A8 23 D0	level 2 ladder left side	00 3B A8 3B D0
3C3A: 00 5B A8 5B D0	level 2 ladder #2 of 4	00 63 A8 63 D0

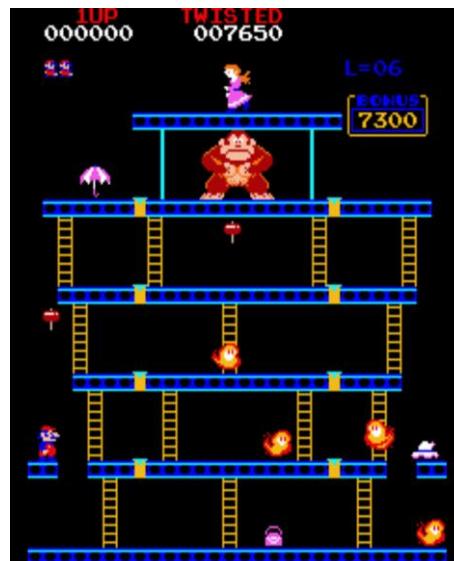
3C3F: 00 A3 A8 A3 D0	level 2 ladder #3 of 4	00 9B A8 9B D0
3C44: 00 DB A8 DB D0	level 2 ladder right side	00 C3 A8 C3 D0
3C49: 00 1B D0 1B F8	bottom left ladder	00 23 D0 23 F8
3C4E: 00 E3 D0 E3 F8	bottom right ladder	00 DB D0 DB F8
3C53: 05 B7 30 48 30	girder above kong	
3C58: 05 CF 58 30 58	girder kong stands on	
3C5D: 05 D7 80 28 80	level 4 girder	
3C62: 05 DF A8 20 A8	level 3 girder	
3C67: 05 E7 D0 18 D0	level 2 girder	
3C6C: 05 EF F8 10 F8	bottom level girder	
3C72: AA	end code	

De twee rivets velden zijn nu:

Oneven:



Even:



Nu het barrels screen iets makkelijker maken door extra girder rechts iets lager te maken, zodat er makkelijker naartoe gesprongen kan worden. En door de rechter ladders iets naar rechts te verplaatsen.

Eerst de rechter trap tussen de 2e en 3e girder:

3B11: 00 BB 98 BB B2 ; short ladder at center right

Deze werkt niet: Jumpman kan wel de ladder op, maar niet af.

3B11: 00 B3 98 B3 B2 ; short ladder at center right

Deze werkt ook niet: Jumpman kan wel de ladder op, maar niet af.

Dit heeft dus geen zin.

Wellicht ook nog eenvoudiger te maken door steering sneller te laten optreden? De level afhankelijke settings hiervan aanpassen zodat het eerder op het hoogste niveau zit en er dus beter gestuurd kan worden?

De extra girder rechts iets naar beneden plaatsen:

3B5C: 02 FE D2 D0 D1 ; Onderste platform

3B61: 02 FE A1 E0 A1 ; Tweede platform

3B66: 00 E3 A1 E3 D1 ; Ladder tussen 1 en 2

Aanpassen naar:

3B52: 02 FE D2 D0 D1 ; Onderste platform	02 FE DA D0 D9
3B57: 02 FE A1 E0 A1 ; Tweede platform	
3B5C: 00 E3 A1 E3 D1 ; Ladder tussen 1 en 2	00 E3 A1 E3 D9

Dan nog de fireball aanpassen die gespawned wordt:

```
3F30 CDA611 CALL #11A6      ; draw hammers
3F33 DD210064 LD IX,#6400    ; load IX with start of fire #1
3F37 DD360001 LD (IX+#00),#01 ; set fire active
3F3B DD3603D8 LD (IX+#03),#D8 ; set fire X position
3F3F DD360ED8 LD (IX+#0E),#D8 ; set fire X position #2
3F43 DD3605D0 LD (IX+#05),#D0 ; set fire Y position
3F47 DD360FD0 LD (IX+#0F),#D0 ; set fire Y position #2
3F4B C9      RET           ; return
```

Nu de heart sprite ook 8 posities naar beneden (Y+8):

```
3F4B: 21206A LD HL,#6A20    ; load HL with heart sprite
3F4E: 36D6 LD (HL),#D6    ; set heart sprite x-position
3F50: 23   INC HL        ; next
3F51: 3676 LD (HL),#76    ; set heart sprite
3F53: 23   INC HL        ; next
3F54: 3609 LD (HL),#09    ; set heart sprite color
3F56: 23   INC HL        ; next
3F57: 36D0 LD HL,#D0    ; set heart sprite y-position
3F59: 3E00 LD A,#00      ; A := #00
3F5B: 329B63 LD (#639B),A ; store into switch status variable
3F5E: C9      RET           ; return
```

De detectie van het pakken van de heart sprite ook aanpassen:

```
258A: 3A 03 62 LD A,(#6203) ; load A with jumpman's x-position
258D: FE D8 CP #D8        ; check x-position = #D8
258F: C0      RET NZ     ; return if x-position is not correct
2590: 3A 05 62 LD A,(#6205) ; load A with jumpman's y-position
2593: FE D0 CP #D0        ; check y-position = #D0
2595: C0      RET NZ     ; return if y-position is not correct
```

Werkt nu allemaal. Grappige is wel dat je de fireball kunt kapot maken met de hamer. Dan heb je daar geen last meer van. Dan kun je de girder gebruiken als een veilige plek om voor lastige barrel combinaties te vluchten. Hier eens uitgebreid mee testen.

Nu rivets zodanig maken dat je er twee keer overeen moet voordat deze meetelt. Eerst alleen het bovenste stukje weghalen. En de tweede keer pas helemaal weghalen en punten toekennen.

De code hiervoor is deze:

```
1A79 CBC0 SET 0,B       ; B := B + 1
1A7B 219262 LD HL,#6292 ; load HL with start of array of rivets
1A7E 78   LD A,B        ; A := B
1A7F 85   ADD A,L       ; add #92
1A80 6F   LD L,A        ; copy to L
1A81 7E   LD A,(HL)     ; get the status of the rivet mario is crossing
1A82 A7   AND A          ; has this rivet already been traversed?
1A83 C8   RET Z         ; yes, return
```

En de code die de afhandeling regelt als je de rivet over gaat:

```

; a rivet has been traversed

1A84 3600 LD (HL),#00 ; set this rivet as cleared
1A86 219062 LD HL,#6290 ; load HL with address of number of rivets remaining
1A89 35 DEC (HL) ; decrease number of rivets
1A8A 78 LD A,B ; A := B
1A8B 010500 LD BC,#0005 ; load BC with offset of 5
1A8E 1F RRA ; rotate right. carry? (is this rivet on right side?)
1A8F DABD1A JP C,#1ABD ; yes, skip ahead and load HL with #012B and return to #1A95

1A92 21CB02 LD HL,#02CB ; else load HL with master offset for rivets

1A95 A7 AND A ; A == 0 ?
1A96 CA9E1A JP Z,#1A9E ; yes, skip next 3 steps

1A99 09 ADD HL,BC ; add offset to HL
1A9A 3D DEC A ; decrease A. zero?
1A9B C2991A JP NZ,#1A99 ; no, loop again

1A9E 010074 LD BC,#7400 ; start of video RAM is #7400
1AA1 09 ADD HL,BC ; add offset computed based on which rivet is cleared
1AA2 3E10 LD A,#10 ; A := #10 = clear space
1AA4 77 LD (HL),A ; erase the rivet
1AA5 2D DEC L ; next video memory
1AA6 77 LD (HL),A ; erase the top of the rivet
1AA7 2C INC L ; next video memory
1AA8 2C INC L ; next video memory
1AA9 77 LD (HL),A ; erase underneath the rivet [ not needed , there is nothing there to erase ???]

```

De code in #1AA6 die wist het bovenste streepje van de rivet. Dus die gebruiken wanneer je er voor de eerste keer overheen gaat.

In #1A81 t/ 1A83 wordt gecheckt of er al over de rivet gegaan is. Deze check vervangen door een jump naar additionele code.

1A81: C37B0A JP #0A7B ; jump to additional code – jump to label **AAAA**

en

```

0A7B: 7E LD A,(HL) ; get the status of the rivet that is crossed – label AAAA
0A7C: FE0A CP #0A ; is rivet status A?
0A7E: CA841A JP Z, #1A84 ; yes, rivet is crossed for second time, erase completely
;
0A81: A7 AND A ; is rivet status 0?
0A82: C8 RET Z ; yes, rivet is already cleared, return
0A83: NOP ;
; rivet is crossed for the first time
;
0A84: 360A LD (HL),#0A ; set this rivet as crossed for the first time
0A86: 78 LD A,B ; A := B
0A87: 010500 LD BC,#0005 ; load BC with offset of 5
0A8A: 1F RRA ; rotate right. carry? (is this rivet on right side?)
0A8B: DAA40A JP C,#0AA4 ; yes, skip ahead - jump to label BBBB
;
0A8E: 21CB02 LD HL,#02CB ; else load HL with master offset for rivets
;
0A91: A7 AND A ; A == 0 ? – label CCCC
0A92: CA9A0A JP Z,#0A9A ; yes, skip next 3 steps – jump to label DDDD
;
0A95: 09 ADD HL,BC ; add offset to HL – label EEEE
0A96: 3D DEC A ; decrease A. zero?
0A97: C2950A JP NZ,#0A95 ; no, loop again – jump to label EEEE

```

```

;
0A9A: 010074 LD BC,#7400 ; start of video RAM is #7400 – label DDDD
0A9D: 09 ADD HL,BC ; add offset computed based on which rivet is cleared
0A9E: 3E10 LD A,#10 ; A := #10 = clear space
0AA0: 2D DEC L ; next video memory
0AA1: 77 LD (HL),A ; erase the top of the rivet
0AA2: 2C INC L ;
0AA3: C9 RET ; else return
;
0AA4: 212B01 LD HL,#012B ; load HL with alternate master offset – label BBBB
0AA7: C3910A JP #0A91 ; jump back to program and resume – jump to label CCCC

```

Ruimte voor additionele code verkrijgen door het intro scherm niet meer te tonen.

Intro scherm overslaan door:

```

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

```

Het hele stuk van #0A7B t/m #0BD9 komt hiermee vrij. Ruimte te gebruiken voor additonele mechanismen voor pies o.a.

Gaat goed. Rivets moeten twee keer overgestoken worden. Eerste keer haalt alleen het bovenste stukje weg. Tweede keer haalt hele rivet weg.

Nu aan de slag met een leukere, lastigere versie van de pies (die geen pies meer zijn 😊).

Idee is om de items in hartjes te veranderen en dat je ze alle drie moet verzamelen om op de girder bij Kong een knipperend platform te krijgen. Verder een soort doolhof met rondwarende fireballs en hamers om ze te slaan. Daarnaast knipperende platformen om bij de items te komen die ook nog eens een trap hebben met ook een fireball.

Eerst de items in hartjes veranderen. Items van pies:

; bonus items on conveyors

```

3E3C 53 73 0A A0 ; position of hat on pies is 53,A0
3E40 8B 74 0A F0 ; position of purse on pies is 8B,F0
3E44 DB 75 0A A0 ; umbrella on the pies is at DB,A0

```

Ervan uitgaande dat de tweede positie de sprite is en de derde positie de kleur (of palette), deze aanpassen naar een hartje in de kleur roze: veranderen in 76 en 09. Gedaan en dat klopt inderdaad.

Veld aanpassen zodat alle girders dezelfde Y-afstand hebben.

Dit veld is vanaf #2210:

```

00 63 18 63 58 ; ladder next to kong (right of two)
00 53 18 53 58 ; ladder next to kong (left of two)
00 93 38 93 58 ; ladder leading to girder with pauline
02 97 38 68 38 ; girder where Pauline stands on
02 EF 58 10 58 ; girder where kong stands on
02 EF F8 10 F8 ; bottom girder where jumpman starts
02 48 E0 10 E0 ; girder left - 1st level
00 23 E0 23 F8 ; ladder left - bottom to 1st level
02 F7 78 88 78 ; girder right side of oil can
02 77 78 08 78 ; girder left side of oil can
00 E3 78 E3 A0 ; ladder right – 3rd level to girder next oil can
00 D3 58 D3 78 ; ladder leading to girder with kong
02 EF A0 B0 A0 ; girder right – 3rd level
02 88 E0 70 E0 ; girder middle – 1st level

```

```

02 EF E0 B0 E0 ; girder right – 1st level
02 88 A0 70 A0 ; girder middle – 3rd level under oil can
02 48 A0 10 A0 ; girder left 3rd level
00 8B C0 8B E0 ; ladder middle – 1st level to 3rd level
00 1B 78 1B A0 ; ladder left – 3rd level to girder next oil can
01 1B C0 1B E0 ; broken ladder left – 1st level to 2nd level
00 E3 C0 E3 E0 ; ladder right – 1st level to 2nd level
02 58 C0 08 C0 ; left girder 2nd level
00 23 A0 23 C0 ; ladder left – 2nd level to 3rd level
01 DB A0 DB C0 ; broken ladder right – 2nd level to 3rd level
00 73 A0 73 C0 ; ladder middle – 1st level to 3rd level
06 80 90 78 90 ; patch of X's under oil can
06 80 88 78 88 ; patch of X's under oil can
06 80 98 78 98 ; patch of X's under oil can
02 F7 C0 A0 C0 ; right girder 2nd level
02 80 C0 78 C0 ; middle girder 2nd level
AA

```

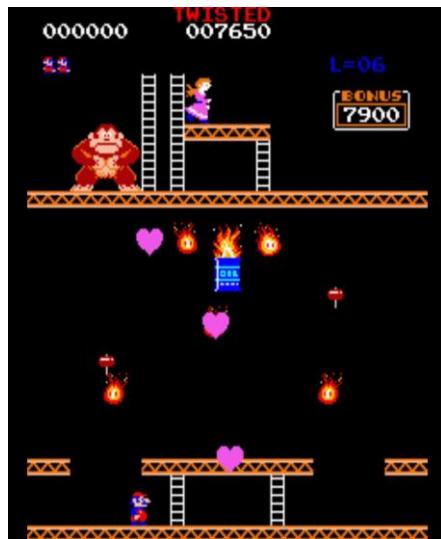
Opnieuw gaan opbouwen:

```

2210: 00 63 18 63 58 ; ladder next to kong (right of two)
2215: 00 53 18 53 58 ; ladder next to kong (left of two)
221A: 00 93 38 93 58 ; ladder leading to girder with pauline
221F: 02 97 38 68 38 ; girder where Pauline stands on
2224: 02 EF 58 10 58 ; girder where kong stands on
2229: 02 EF F8 10 F8 ; bottom girder where jumpman starts
222E: 02 20 D8 10 D8 ; girder 1st level left
2233: 02 B0 D8 48 D8 ; girder 1st level middle
2238: 02 EF D8 DF D8 ; girder 1st level right
223D: 00 5B D8 5B F8 ; left bottom ladder
2242: 00 A3 D8 A3 F8 ; right bottom ladder
AA

```

Geeft volgende scherm:



De knipperende platformen ertussen laten knipperen, en de andere twee voorlopig uitzetten:

```

264D: 211B77 LD HL,#771B ; video memory address to write – label DDDD
2650: CD6C26 CALL #266C ; call draw routine – call label EEEE
2653: 21DB74 LD HL,#74DB ; video memory address to write
2656: CD6C26 CALL #266C ; call draw routine – call label EEEE
2659: 78 LD A,B ; load A with B

```

```

265A: 217C76 LD HL,#767C ; video memory address to write
265D: CD6C26 CALL #266C ; call draw routine call label EEEE
265D: 00 00 00 NOP NOP NOP
2660: 217475 LD HL,#7574 ; video memory address to write
2663: CD6C26 CALL #266C ; call draw routine call label EEEE
2663: 00 00 00 NOP NOP NOP

```

De twee eerste hartjes op de platformen plaatsen (hat: links en purse: rechts):

; bonus items on conveyors

```

3E3C 1B 76 09 D0 ; position of hat on pies
3E40 E4 76 09 D0 ; position of purse on pies
3E44 DB 76 09 A0 ; umbrella on the on pies

```

Het veld ziet er nu zo uit:



Extra girders toevoegen:

```

2247: 02 EF B8 10 B8 ; girder 2nd level
224C: 02 68 98 10 98 ; girder 3rd level left
2251: 02 EF 98 90 98 ; girder 3rd level right
2256: 02 F7 78 88 78 ; girder right side of oil can
225B: 02 77 78 08 78 ; girder left side of oil can
AA

```

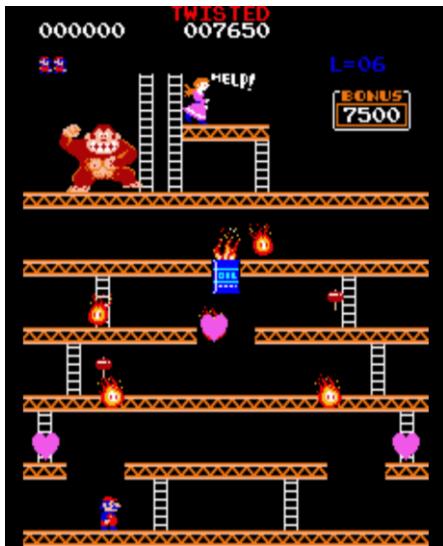
Extra ladders toevoegen:

```

2260: 00 1B B8 1B D8 : left ladder 2nd level
2265: 00 E3 B8 E3 D8 : right ladder 2nd level
226A: 00 2B 98 2B B8 : left ladder 3rd level
226F: 00 D3 98 D3 B8 : right ladder 3rd level
2274: 00 3B 78 3B 98 : left ladder 4th level
2279: 00 C3 78 C3 98 : right ladder 4th level

```

Veld ziet er nu zo uit:



Nu de drie fireballs initieel op de juiste plek spawnen:

```

2690: DD 21 00 64 LD    IX,#6400      ; load IX with start of fire #1
2694: DD 36 00 01 LD    (IX+#00),#01  ; set fire active
2698: DD 36 03 7F LD    (IX+#03),#7F  ; set fire X position
269C: DD 36 0E 7F LD    (IX+#0E),#7F  ; set fire X position #2
26A0: DD 36 05 D0 LD    (IX+#05),#D0  ; set fire Y position
26A4: DD 36 0F D0 LD    (IX+#0F),#D0  ; set fire Y position #2
;
26A8: DD 36 20 01 LD    (IX+#20),#01  ; set fire active
26AC: DD 36 23 1B LD    (IX+#23),#1B  ; set fire X position
26B0: DD 36 2E 1B LD    (IX+#2E),#1B  ; set fire X position #2
26B4: DD 36 25 D0 LD    (IX+#25),#D0  ; set fire Y position
26B8: DD 36 2F D0 LD    (IX+#2F),#D0  ; set fire Y position #2
;
26BC: DD 36 40 01 LD    (IX+#40),#01  ; set fire active
26C0: DD 36 43 E4 LD    (IX+#43),#E4  ; set fire X position
26C4: DD 36 4E E4 LD    (IX+#4E),#E4  ; set fire X position #2
26C8: DD 36 45 D0 LD    (IX+#45),#D0  ; set fire Y position
26CC: DD 36 4F D0 LD    (IX+#4F),#D0  ; set fire Y position #2
;
26D0: C9          RET           ; return

```

Nu het derde platform laten knipperen onder de oil can:

```

264D: 211B77 LD    HL,#771B      ; video memory address to write - label DDDD
2650: CD6C26 CALL  #266C        ; call draw routine - call label EEEE
2653: 21DB74 LD    HL,#74DB      ; video memory address to write
2656: CD6C26 CALL  #266C        ; call draw routine - call label EEEE
2659: 78          LD A,B        ; load A with B
265A: 21F375 LD    HL,#75F3      ; video memory address to write
265D: CD6C26 CALL  #266C        ; call draw routine - call label EEEE
2660: 217475 LD    HL,#7574      ; video memory address to write
2663: CD6C26 CALL  #266C        ; call draw routine - call label EEEE
2663: 00 00 00 NOP NOP NOP

```

Het derde hartje (umbrella) boven het knipperende platform plaatsen.

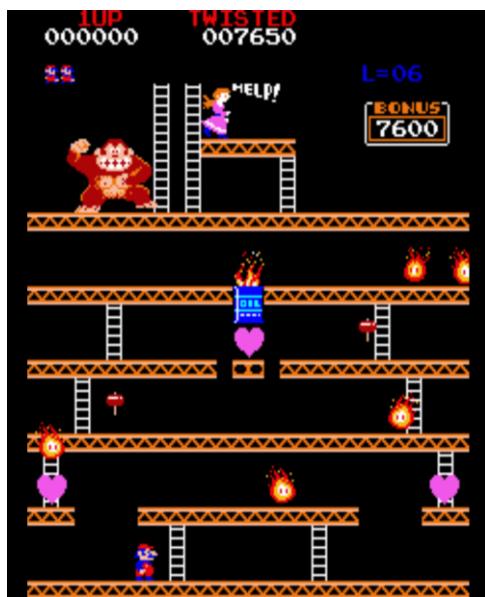
; bonus items on conveyors

```

3E3C 1B 76 09 D0      ; position of hat on pies
3E40 E4 76 09 D0      ; position of purse on pies
3E44 7F 76 09 90      ; umbrella on the on pies

```

Het veld ziet er nu zo uit:



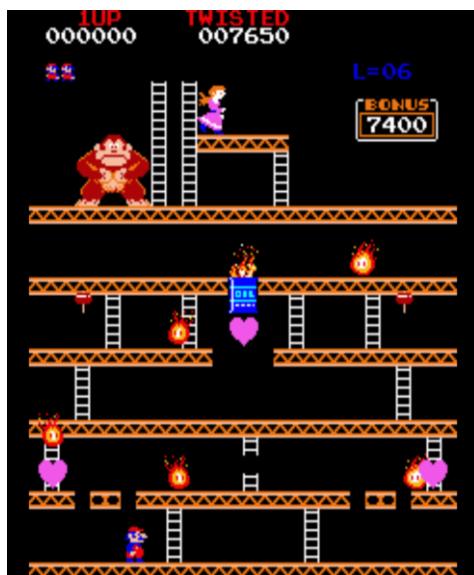
Nu nog extra ladders zodat de fireballs beter kunnen ronddwalen. Eventueel een broken center ladder op 2nd level toevoegen.

2260: 00 1B B8 1B D8 : left ladder 2nd level
2265: 00 E3 B8 E3 D8 : right ladder 2nd level
226A: 00 2B 98 2B B8 : left ladder 3rd level
226F: 00 D3 98 D3 B8 : right ladder 3rd level
2274: 00 3B 78 3B 98 : left ladder 4th level
2279: 00 C3 78 C3 98 : right ladder 4th level
227E: 00 63 78 63 98 ; ladder 4th level left of oil can
2283: 00 9B 78 9B 98 ; ladder 4th level right of oil can
2288: 01 83 B8 83 D8 ; broken ladder center 2nd level

De twee hamers op 4th level hangen:

3E10: 2B 80 D3 80 ; for conveyors

Veld ziet er nu zo uit:



Nog twee trappen op het 3rd level toevoegen:

2260: 00 1B B8 1B D8 : left ladder 2nd level
2265: 00 E3 B8 E3 D8 : right ladder 2nd level
226A: 00 2B 98 2B B8 : left ladder 3rd level
226F: 00 D3 98 D3 B8 : right ladder 3rd level
2274: 00 3B 78 3B 98 : left ladder 4th level
2279: 00 C3 78 C3 98 : right ladder 4th level
227E: 00 63 78 63 98 ; ladder 4th level left of oil can
2283: 00 9B 78 9B 98 ; ladder 4th level right of oil can
2288: 01 83 B8 83 D8 ; broken ladder center 2nd level
228D: 00 53 98 53 B8 ; second ladder 3rd level
2292: 00 AB 98 AB B8 ; third ladder 3rd level

Ook nog trap naar het bovenste level toevoegen:

2297: 00 E3 58 E3 78 ; ladder to Kong girder

En de broken trap gewoon maken:

2288: 00 83 B8 83 D8 ; broken ladder center 2nd level

Probleem met alle fireballs boven, is dat je niet kunt ontsnappen.

Wellicht een escape route maken aan de rechterkant? Waar je op twee niveau's naar toe kunt springen?

Daartoe twee girders korter maken:

2251: 02 CF 98 90 98 ; girder 3rd level right
2256: 02 CF 78 88 78 ; girder right side of oil can

En een korte girder toevoegen en de ladder naar Kong's girder langer maken:

229C: 02 F7 98 DF 98 ; short girder 3rd level right
2297: 00 E3 58 E3 98 ; ladder to Kong girder

En ladders verplaatsen:

226F: 00 CB 98 CB B8 : right ladder 3rd level
2279: 00 BB 78 BB 98 : right ladder 4th level
226A: 00 33 98 33 B8 : left ladder 3rd level
2274: 00 43 78 43 98 : left ladder 4th level

Nu teveel ladders, kan niet naar de girder met Kong klimmen.
Is weer opgelost als we de center ladder 2nd level weer broken maken.

De hamers verschuiven:

3E10: **2B 80 D3 80** ; for conveyors

Veld ziet er nu zo uit:



Proberen met slechts alleen de middelste fireball spawnen:

```

2690: DD 21 00 64 LD    IX,#6400      ; load IX with start of fire #1
2694: DD 36 00 01 LD    (IX+#00),#01  ; set fire active
2698: DD 36 03 7F LD    (IX+#03),#7F  ; set fire X position
269C: DD 36 0E 7F LD    (IX+#0E),#7F  ; set fire X position #2
26A0: DD 36 05 D0 LD    (IX+#05),#D0  ; set fire Y position
26A4: DD 36 0F D0 LD    (IX+#0F),#D0  ; set fire Y position #2
;
26A8: C9             RET          ; return
;
26A8: DD 36 20 01 LD    (IX+#20),#01  ; set fire active
26AC: DD 36 23 1B LD    (IX+#23),#1B  ; set fire X position
26B0: DD 36 2E 1B LD    (IX+#2E),#1B  ; set fire X position #2
26B4: DD 36 25 D0 LD    (IX+#25),#D0  ; set fire Y position
26B8: DD 36 2F D0 LD    (IX+#2F),#D0  ; set fire Y position #2
;
26BC: DD 36 40 01 LD    (IX+#40),#01  ; set fire active
26C0: DD 36 43 E4 LD    (IX+#43),#E4  ; set fire X position
26C4: DD 36 4E E4 LD    (IX+#4E),#E4  ; set fire X position #2
26C8: DD 36 45 D0 LD    (IX+#45),#D0  ; set fire Y position
26CC: DD 36 4F D0 LD    (IX+#4F),#D0  ; set fire Y position #2
;
26D0: C9             RET          ; return

```

Terug naar de symmetrische versie en dan zo maken dat het level klaar is wanneer je op het niveau van Kong bent en zorgen dat de fireballs wel naar dat niveau gaan, maar niet meer naar beneden.

Klaar als op niveau van Kong. Dat was aangepast. Weer terugzetten naar oorspronkelijke waarden:

```

1E66 fe51     CP      #51          ; else on the conveyors. is mario high enough to end level?
1E68 d0       RET     nc          ; no, return

```

Weer teruggezet op 51 en level is nu klaar wanneer op niveau van Kong.

Ander idee is om de twee heart sprites beneden ook boven de knipperende plateau's te hangen!. Dus deze (hat en purse) gaan verplaatsen:

; bonus items on conveyors

```

3E3C 3B 76 09 D0      ; position of hat on pies

```

```
3E40 C4 76 09 D0 ; position of purse on pies  
3E44 7F 76 09 90 ; umbrella on the on pies
```

De fireballs wel naar de girder van Kong laten gaan:

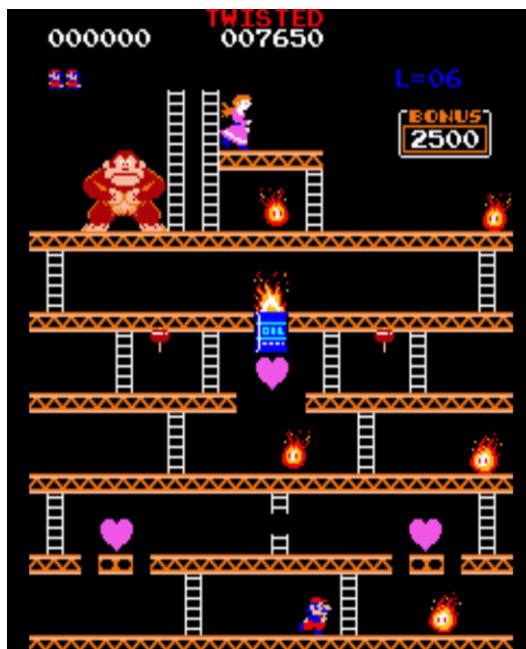
Hele spullenboeltje is weer teruggezet naar oorspronkelijke code. De fireballs gaan nu wel naar de girder met Kong en blijven daar ook. Nog wel ook een ladder maken aan de linkerkant. En twee ladders weghalen:

```
2260: 00 1B B8 1B D8 : left ladder 2nd level  
2265: 00 E3 B8 E3 D8 : right ladder 2nd level  
226A: 00 2B 98 2B B8 : left ladder 3rd level  
226F: 00 D3 98 D3 B8 : right ladder 3rd level  
226A: 00 3B 78 3B 98 : left ladder 4th level  
226F: 00 C3 78 C3 98 : right ladder 4th level  
2274: 00 63 78 63 98 ; ladder 4th level left of oil can  
2279: 00 9B 78 9B 98 ; ladder 4th level right of oil can  
227E: 01 83 B8 83 D8 ; broken ladder center 2nd level  
2283: 00 53 98 53 B8 ; second ladder 3rd level  
2288: 00 AB 98 AB B8 ; third ladder 3rd level  
228D: 00 E3 58 E3 78 ; right ladder to Kong girder  
2292: 00 1B 58 1B 78 ; left ladder to Kong girder
```

Alleen de middelste fireball vanaf begin plaatsen. De andere twee gewoon vanuit de oil can laten spawnnen. Zie hierboven:

```
26A8: C9 RET ; return
```

Level ziet er nu zo uit:



Nu zo maken dat de ladders naar de girder met Kong niet helemaal tot boven beklimmen kunnen worden. Dus bij einde level check, ook testen of jumpman een Y-waarde heeft net onder de Y-waarde van de girder (dan zit hij dus bovenaan de trap). Op dat moment testen of alle drie de bonus items opgepakt zijn (op basis van items left counter: variabele #6385: als deze 0 is, dan alle items opgepakt). Als ze niet alle drie opgepakt zijn, dan de Y-waarde van jumpman verhogen met 1 (1 plek naar beneden op scherm). Anders door laten klimmen.

Visueel de ladders (die niet broken zijn) toch onderbreken door een leeg blok op een plek over de ladder heen te tekenen.

```
; checks for end of level ?

1E57 3A2762 LD A,(#6227) ; load a with screen number
1E5A cb57 BIT 2,A ; are we on the rivets?
1E5C c2801E JP NZ,#1E80 ; yes, skip ahead to handle

1E5F 1F rra ; else rotate right with carry
1E60 3A0562 LD A,(#6205) ; load A with y position of mario
1E63 dA7A1E JP c,#1E7A ; skip ahead on girders and elevators

1E66 fe51 CP #51 ; else on the conveyors. is mario high enough to end level?
1E68 d0 RET nc ; no, return
```

De #1E66 vervangen door aanroep naar additionele code:

1E66: C39822 JP #2298 ; jump to additional code – jump to label **AAAA**

en

```
2298: FE60 CP #60 ; is jumpman high enough on ladder? – label AAAA
229A: D0 RET NC ; no return
;
229B: 3A8563 LD A,(#6385) ; load A with items left counter
229E: FE00 CP #00 ; are no items left?
22A0: CAAE22 JP Z, #22AE ; yes, skip next lines – jump to label BBBB
;
22A3: 3A0562 LD A,(#6205) ; load A with y-position of jumpman
22A6: 3C INC A ; move jumpman 4 positions down
22A7: 3C INC A ;
22A8: 3C INC A ;
22A9: 3C INC A ;
22AA: 320562 LD (#6205),A ; store changed y-position of jumpman
22AD: C9 RET ; no end of level: return
;
22AE: 3A0562 LD A,(#6205) ; load A with y-position of jumpman – label BBBB
22B1: FE51 CP #51 ; is jumpman high enough to end level?
22B3: D0 RET NC ; no, return
;
22B4: C3691E JP #1E69 ; yes, jump back
```

Bij y-positie 60 het mooiste resultaat. Als lager, dan beïnvloed dat ook de jump op de girder er onder, wat niet de bedoeling is. De y-positie moet niet 1 lager gemaakt worden: dan blijft jumpman bij naar beneden klimmen doorklimmen voorbij de onderkant van de ladder. Pas bij verlagen met 4 gaat het goed.

Na het tekenen van het conveyor veld de trappen onderbroken maken door er een spatie over heen te tekenen. En dan de items left counter (#6385) op 3 zetten. Eventjes kijken waar dat kan. Er is al extra code voor conveyors toegevoegd (om stukjes broken/retractable ladder te verwijderen), maar dat is neem ik aan niet meer nodig.

In onderstaande stukje worden de andere variabelen voor pies gevuld. Daar is nog ruimte dus daar zou de initialisatie van de items left counter aan toegevoegd kunnen worden:

```
103A: 3E00 LD A,#00 ; A := #00
103C: 329B63 LD (#639B),A ; store into switch status variable
103F: 32A065 LD (#65A0),A ; store in blink counter
1042-1048 NOP's
```

```

1042: 3E03 LD A,#03      ; A:= #03
1044: 328563 LD (#6385),A ; store in items left counter

```

Ook nog de ladder een stukje gebroken maken. Daar is nog ruimte voor. De plek waar origineel de retractable ladders en de conveyor pulleys getekend werden is nog deels leeg: #105D t/m #106F.

```

105D: 3E10 LD A,#10      ; load A with rivet character (lijkt op gladde trap)
105F: 216C74 LD HL,#746C ; video memory address to change
1062: 77 LD (HL),A      ; set memory address = make ladder broken
1063: 218C77 LD HL,#778C ; video memory address to change
1066: 77 LD (HL),A      ; set memory address = make ladder broken

```

Bonus item pickup wordt getest vanaf #19DA. Daar wordt eerst de HL geladen met de start van het eerste bonus item (= #6A0C). En dan een loop om ze alle drie te testen (steeds 4 geheugenplaatsen verder de volgende). De eerste geheugenplaats is de X-waarde, de tweede geheugenplaats is de sprite, de derde geheugenplaats is het palette en de vierde geheugenplaats is de Y-waarde. Er wordt daar gekeken of het bonus-item opgepakt is door de sprite te bekijken. Als bit 3 daarvan 0 is, dan is het item al opgepakt.

HL = #6A0C,
 HL = X-waarde bonus item
 HL + INC + INC + INC = Y-waarde bonus item
 HL + INC + INC + INC + DEC + DEC = sprite waarde (= HL + INC!)
 Testen dan is BIT 3,(HL) als NZ dan al opgepikt.

Wanneer een bonus item opgepakt, springen naar extra stuk code:

```

; bonus item has been picked up

19FA 2D DEC L      ; decrease L. HL now has the starting address of the sprite that was picked up
19FB 224363 LD (#6343),HL ; store into this temp memory. read from at #1E18
19FE AF XOR A      ; A := 0
19FF 324263 LD (#6342),A ; store into ???. read from at #1DD6
1A02 3C INC A      ; A := 1
1A03 324063 LD (#6340),A ; store into #6340 - usually 0, changes when mario picks up bonus item. jumps over item turns to 1
1A06 C9 RET        ; return

```

Dus op 19FA springen naar additionele code. In de additionele code de waarde van de items left counter (#6385) met één verlagen. Als deze 0 geworden is, de trapdelen weer intekenen. Altijd (onafhankelijk van het aantal opgepakte bonus-items) daarna stukje van de verwijderde code op #19FA en #19FB uitvoeren en terugspringen naar #19FE.

```

19FA: C3AA0A JP #0AAA      ; jump to additional code – jump to label AAAA
19FD: NOP

```

en

```

0AAA: 3A8563 LD A, (#6385) ; load A with items left counter – label AAAA
0AAD: 3D DEC A           ; decrement items left counter
0AAE: 328563 LD (#6385), A ; set items left counter
;
0AB1: FE00 CP #00         ; check if A is 0: no items left anymore
0AB3: C2C20A JP NZ, 0AC2 ; no, skip next steps – jump to label BBBB
;
0AB6: E5 PUSH HL          ; save HL
0AB7: 3EC0 LD A,#C0        ; load A with ladder character
0AB9: 216C74 LD HL,#746C ; video memory address to change
0ABC: 77 LD (HL),A        ; set memory address = add ladder piece
0ABD: 218C77 LD HL,#778C ; video memory address to change
0AC0: 77 LD (HL),A        ; set memory address = add ladder piece
0AC1: E1 POP HL           ; restore HL

```

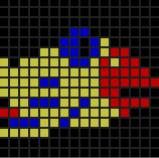
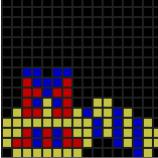
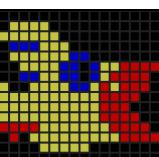
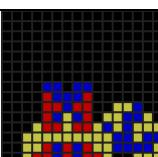
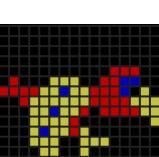
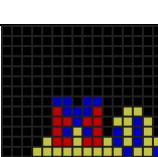
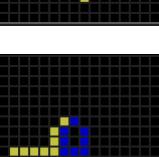
```

;
0AC2: 2D      DEC L          ; decrease L. HL = picked sprite - label BBBB
0AC3: 224363  (#6343), HL    ; store into this temp memory
0AC6: C3FE19  JP #19FE       ; jump back

```

Nu de extra enemies gaan toevoegen. In DKjr zitten twee vogels en de snappers. In DK3 zitten de rupsjes. Al deze graphics in de graphics file toevoegen. Dan kun je later nog kijken welke je wel/niet wilt gebruiken.

De volgende sprites zijn toegevoegd:

#58			#68		
#59			#69		
#5A			#6A		
#5B					
#5C					
#5D					

Extra code kan in de gameflow toegevoegd worden, door op #19C2 een call toe te voegen (is nu nog leeg met 2 NOP's).

Proberen 1 vogel sprite op het scherm te krijgen d.m.v. de geheugenplaatsen die normaal voor de pies gebruikt worden:

```

; #6500 - #65AF = the ten bouncer values, 6510, 6520, etc. are starting values
;      +3 is the X pos, +5 is the Y pos

; #65A0 - #65?? = values for the 6 pies

; #6600 - 665F = the 6 elevator values. 6610, 6620, 6630, 6640 ,6650 are starting values
;      + 3 is the X position, + 5 is the Y position

```

En

; copies pie buffer to pie sprites

```

24EA 3E02    LD   A,#02      ; check level for conveyors
24EC F7      RST  #30      ; if not conveyors, RET, else continue
24ED CD2325  CALL #2523  ; check for deployment of new pies
24F0 CD9125  CALL #2591  ; update all pies positions based on direction of trays, remove pies in fire or off edge
24F3 DD21A065 LD   IX,#65A0 ; load IX with start of pies
24F7 0606    LD   B,#06      ; for B = 1 to 6 pies
24F9 21B869  LD   HL,#69B8 ; load HL with hardware address for pies

24FC DD7E00  LD   A,(IX+#00) ; load A with sprite status
24FF A7      AND  A        ; is this sprite active ?
2500 CA1C25  JP   Z,#251C  ; no, add 4 to L and loop again

2503 DD7E03  LD   A,(IX+#03) ; load A with pie X position
2506 77      LD   (HL),A    ; store into sprite
2507 2C      INC  L        ; next address
2508 DD7E07  LD   A,(IX+#07) ; load A with pie sprite value
250B 77      LD   (HL),A    ; store into sprite
250C 2C      INC  L        ; next address
250D DD7E08  LD   A,(IX+#08) ; load A with pie color
2510 77      LD   (HL),A    ; store into sprite
2511 2C      INC  L        ; next address
2512 DD7E05  LD   A,(IX+#05) ; load A with pie Y position
2515 77      LD   (HL),A    ; store into sprite
2516 2C      INC  L        ; next address

2517 DD19    ADD  IX,DE      ; add offset for next pie
2519 10E1    DJNZ #24FC    ; next B

251B C9      RET           ; return

```

De structuur vanaf #65A0 is administratie rondom de pies en de daadwerkelijke hardware sprite adressen zijn vanaf #69B8. En dan steeds vier geheugenplaatsen:

#69B8 = x-positie, #69B9 = sprite value, #69BA = sprite palette, #69BB = y-positie.

Stukje code om vogel te tekenen:

```
19C2: CDC90A    CALL 0AC9      ; do stuff for birds and snappers – call to AAAA
```

en

```

0AC9: 3A2762  LD A,(#6227)  ; load A with screen number – label AAAA
0ACC: FE02    CP #02      ; are we on the conveyors?
0ACE: C0      RET NZ     ; no, return
;
0ACF: 21B869  LD HL, #69B8  ; load HL with start sprite address
0AD2: 3638    LD (HL),#38  ; x-position = 38
0AD4: 23      INC HL      ; 
0AD5: 365A    LD (HL),#5A  ; sprite value of bird
0AD7: 23      INC HL      ; 
0AD8: 3601    LD (HL),#01  ; sprite palette
0ADA: 23      INC HL      ;

```

```

0ADB: 36A6 LD (HL),#A6 ; y-position = A6
0ADD: 23 INC HL
0ADE: C9 RET

```

Sprite palette: 01



Sprite palette: 02



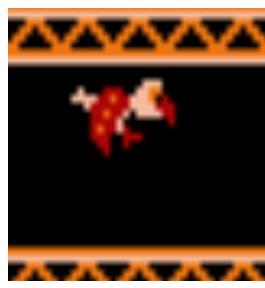
Sprite palette: 03



Sprite palette: 07



Sprite palette: 08



Sprite palette: 09



Sprite palette: 0A



Sprite palette: 0B



De palettes die het beste uitkomen zijn 01, 03 en 07 door de wit/zwarte ogen.

Nu de vogel laten gaan vliegen. Dit door eenvoudig de x-positie te gaan verhogen.

```

0AC9: 3A2762 LD A,(#6227) ; load A with screen number – label AAAA
0ACC: FE02 CP #02 ; are we on the conveyors?
0ACE: C0 RET NZ ; no, return
;
0ACF: 3AA463 LD A,(#63A4) ; load A with bird movement counter
0AD2: 3C INC A ; increment bird movement counter
0AD3: 32A463 LD (#63A4), A ; store bird movement counter
;
0AD6: FE04 CP #04 ; A=04?
0AD8: C0 RET NZ ; no, return
;
0AD9: 3E00 LD A, #00 ; yes, A:=00
0ADB: 32A463 LD (#63A4), A ; reset bird movement counter
;
0ADE: 21B869 LD HL, #69B8 ; load HL with start sprite address
0AE1: 34 INC (HL) ; move bird one place to the right
0AE2: 34 INC (HL) ; move bird one place to the right
;
0AE3: 23 INC HL
0AE4: 3A1A60 LD A,(FrameCounter)
0AE7: E610 AND #10 ; framecounter bit 2 is set?
0AE9: 2004 JR NZ, #BBBB ; no, skip next lines – jump to label BBBB
0AEB: 365A LD (HL),#5A ; sprite value of bird – variant A
0AED: 1802 JR #CCCC ; skip next lines – jump to label CCCC
0AEF: 365B LD (HL),#5B ; sprite value of bird – variant B – label BBBB
;
0AF1: 23 INC HL ; label CCCC
0AF2: 3601 LD (HL),#01 ; sprite palette

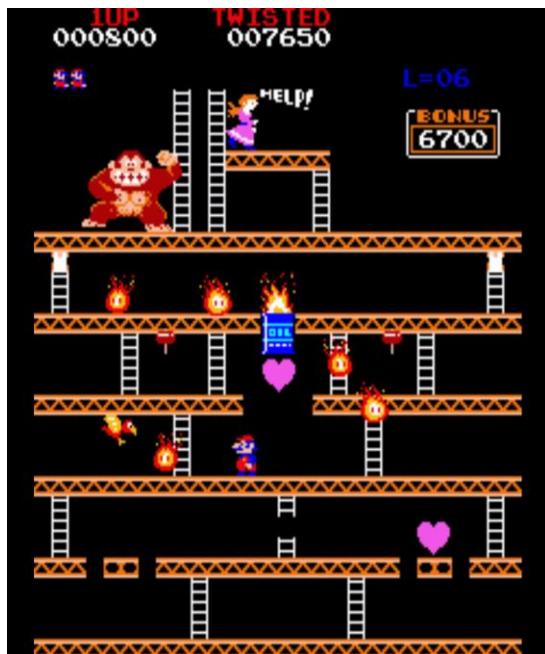
```

```

;
0AF4: 23      INC HL
0AF5: 36A4    LD (HL),#A4 ; y-position = A4
;
0AF7: C9      RET

```

Gaat goed:



Wanneer juiste snelheid en animatie, dan de sprite informatie table ook vullen met informatie. De sprite op actief zetten , de x-coordinaat en de y-coordinaat en de breedte en de hoogte. Als dat gevuld is, kijken of de collision detection dan werkt. Dat zou mooi zijn. Dan hoef je dat niet zelf te bouwen.

Informatiestructuur is: pies: 65A0 / 65B0 / 65C0 /65D0 / 65E0 en 65F0 (6 stuks)

+00 = sprite active: als 01 dan active

+03 = sprite x-position

+05 = sprite y-position

+07 = sprite y-value

+09 = sprite size width

+0A = sprite size height

```

0AF7: DD21B065 LD IX,#65B0      ; load IX with start of pies sprites table
0AFB: DD3605A4 LD (IX+#05),#A4  ; store #A4 into bird Y position
0AFF: DD360001 LD (IX+#00),#01  ; set bird active
0B03: DD360908 LD (IX+#09),#08  ; set bird size (width?)
0B07: DD360A08 LD (IX+#0A),#05  ; set bird size (height?)

;
0B0B: 21B869 LD HL, #69B8      ; load HL with start sprite address
0B0E: 7E      LD A,(HL)        ; A = sprite x-position
0B0F: DD7703 LD (IX+#03),A     ; store into bird X position
;
0B12: C9      RET

```

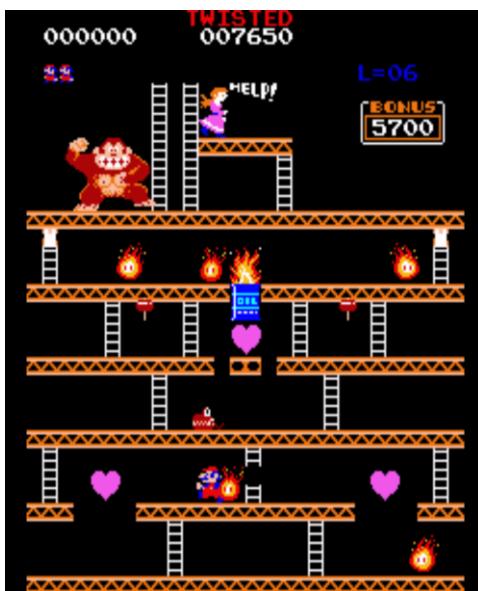
Werkte eerst niet #65A0 is de blink counter. Dus die niet nemen. Dan maar #65B0 genomen. Zorgt er wel voor dat we één positie minder hebben voor enemy sprites. Maar dat gaan we dan wel zien.

Testen met invincibility weer uitgezet:

19B3: CD0828

	Vogel beneden	Vogel boven	Snapper beneden
Y-positie sprite	#A4	#64	#B0
Y-positie array	#A4	#64	#B4
Width sprite	#08	#08	#05
Height sprite	#05	#05	#06
Move counter	#04	#04	#04
Move positions	2x INC	2x INC	2x DEC
Animation	#10	#10	#08
Sprite value A	#5A	#5A	#5C
Sprite value B	#5B	#5B	#5D
Sprite palette	#01	#01	#07

Uitgewerkt met snapper:



Nu alles samenbrengen. Twee vogels boven, één vogel onderin en twee snapper onderin.

#63A4 is de enemy movement counter.
#65A1 is de enemy deploy timer

69B8 / 65B0: vogel boven, start meteen
69C8 / 65C0: vogel boven, start na deployment timer 80 (nog fine tunen)
69D8 / 65D0: vogel onder, start meteen
69E8 / 65E0: snapper, start meteen
69F8 / 65F0: snapper, start na deployment timer 80 (nog fine tunen)

#69B8 = x-positie, #69B9 = sprite value, #69BA = sprite palette, #69BB = y-positie.

```

0AC9: 3A2762 LD A,(#6227) ; load A with screen number - label AAAA
0ACC: FE02 CP #02 ; are we on the conveyors?
0ACE: C0 RET NZ ; no, return
;
0ACF: 3AA165 LD A,(#65A1) ; load A with enemy deploy timer
0AD2: FE00 CP #00 ; is deploy timer 00?
0AD4: CC600B CALL Z, #0B60 ; yes, initialize sprites - call to label ZZZZ
;
0AD7: FEF0 CP #F0 ; is deploy timer F0?
0AD9: 2804 JR Z, BBBB ; yes, skip next steps - jump relative to BBBB
;
0ADB: 3C INC A ; increment A

```

```

0ADC: 32A165 LD (#65A1), A ; store into enemy deploy timer
;
; perform sprite animation
;
0ADF: 3A1A60 LD A,(FrameCounter) ; - label BBBB
0AE2: 47 LD B, A ; save A into B
;
0AE3: E610 AND #10 ; framecounter bit 5 is set?
0AE5: 2004 JR NZ, #CCCC ; no, skip next lines – jump relative to label CCCC
;
0AE7: 3E5A LD A, #5A ; load A with sprite # of bird variant A
0AE9: 1802 JR, #DDDD ; skip next lines – jump relative to label DDDD
;
0AEB: 3E5B LD A, #5B ; load A with sprite # of bird variant B – label CCCC
;
0AED: 32B969 LD (#69B9), A ; set sprite upper bird 1 – label DDDD
0AF0: 32BD69 LD (#69BD), A ; set sprite upper bird 2
0AF3: 32C169 LD (#69C1), A ; set sprite lower bird [verandert een fireball sprite]
;
0AF6: 78 LD A, B ; restore A from B
;
0AF7: E608 AND #08 ; framecounter bit 4 is set?
0AF9: 2004 JR NZ, #EEEE ; no, skip next lines – jump relative to label EEEE
;
0AFB: 3E5C LD A, #5C ; load A with sprite # of snapper variant A
0AFD: 1802 JR, #FFFF ; skip next lines – jump relative to label FFFF
;
0AFF: 3E5D LD A, #5D ; load A with sprite # of snapper variant B – label EEEE
;
0B01: 32C569 LD (#69C5), A ; set sprite snapper 1 – label FFFF
0B04: 32C969 LD (#69C9), A ; set sprite snapper 2
;
; perform sprite movement
;
0B07: 3AA463 LD A,(#63A4) ; load A with enemy movement counter
0B0A: 3C INC A ; increment enemy movement counter
0B0B: 32A463 LD (#63A4), A ; store enemy movement counter
;
0B0E: FE03 CP #03 ; A=03?
0B10: C0 RET NZ ; no, return
;
0B11: 3E00 LD A, #00 ; yes, A:=00
0B13: 32A463 LD (#63A4), A ; reset bird movement counter
;
0B16: 3AA165 LD A, (#65A1) ; load a with enemy deploy timer
;
0B19: 21B869 LD HL, #69B8 ; load HL with upper bird 1 hw x position
0B1C: 34 INC (HL) ; move bird one place to the right
0B1D: 34 INC (HL) ; move bird one place to the right
0B1E: 21B365 LD HL, #65B3 ; load HL with upper bird 1 definition sprite x position
0B21: 34 INC (HL) ; move bird one place to the right
0B22: 34 INC (HL) ; move bird one place to the right
;
0B23: FE80 CP #80 ; time to move upper bird 2 yet?
0B25: 380A JR C #GGGG ; no, skip next steps – jump relative to GGGG
;
0B27: 21BC69 LD HL, #69BC ; load HL with upper bird 2 hw x position
0B2A: 34 INC (HL) ; move bird one place to the right
0B2B: 34 INC (HL) ; move bird one place to the right
0B2C: 21C365 LD HL, #65C3 ; load HL with upper bird 2 definition sprite x position
0B2F: 34 INC (HL) ; move bird one place to the right
0B30: 34 INC (HL) ; move bird one place to the right
;

```

0B31: 21C069	LD HL, #69C0	; load HL with lower bird hw x position – label GGGG
0B34: 34	INC (HL)	; move bird one place to the right
0B35: 34	INC (HL)	; move bird one place to the right
0B36: 21D365	LD HL, #65D3	; load HL with lower bird definition sprite x position
0B39: 34	INC (HL)	; move bird one place to the right
0B3A: 34	INC (HL)	; move bird one place to the right
;		
0B3B: 21C469	LD HL, #69C4	; load HL with snapper 1 hw x position
0B3E: 35	DEC (HL)	; move bird one place to the right
0B3F: 35	DEC (HL)	; move bird one place to the right
0B40: 21E365	LD HL, #65E3	; load HL with snapper 1 definition sprite x position
0B43: 35	DEC (HL)	; move bird one place to the right
0B44: 35	DEC (HL)	; move bird one place to the right
;		
0B45: FE80	CP #80	; time to move upper bird 2 yet?
0B47: 380A	JR C #HHHH	; no, skip next steps – jump relative to HHHH
;		
0B49: 21C869	LD HL, #69C8	; load HL with snapper 2 hw x position
0B4C: 35	DEC (HL)	; move bird one place to the right
0B4D: 35	DEC (HL)	; move bird one place to the right
0B4E: 21F365	LD HL, #65F3	; load HL with snapper 2 definition sprite x position
0B51: 35	DEC (HL)	; move bird one place to the right
0B52: 35	DEC (HL)	; move bird one place to the right
;		
0B53: C9	RET	; return – label HHHH

En

0B60: 3E01	LD A, #01	; A := 01 – label ZZZZ
0B62: 32B065	LD (#65B0), A	; activate upper bird 1
0B65: 32C065	LD (#65C0), A	; activate upper bird 2
0B68: 32D065	LD (#65D0), A	; activate lower bird
0B6B: 32E065	LD (#65E0), A	; activate snapper 1
0B6E: 32F065	LD (#65F0), A	; activate snapper 2
0B71: 32BA69	LD (#69BA), A	; palette upper bird 1
0B74: 32BE69	LD (#69BE), A	; palette upper bird 2
0B77: 32C269	LD (#69C2), A	; palette lower bird
;		
0B7A: 3E08	LD A, #08	; A := 08
0B7C: 32B965	LD (#65B9), A	; width upper bird 1
0B7F: 32C965	LD (#65C9), A	; width upper bird 2
0B82: 32D965	LD (#65D9), A	; width lower bird
;		
0B85: 3E05	LD A, #05	; A := 05
0B87: 32BA65	LD (#65BA), A	; height upper bird 1
0B8A: 32CA65	LD (#65CA), A	; height upper bird 2
0B8D: 32DA65	LD (#65DA), A	; height lower bird
0B90: 32E965	LD (#65E9), A	; width snapper 1
0B93: 32F965	LD (#65F9), A	; width snapper 2
;		
0B96: 3E06	LD A, #06	; A := 06
0B98: 32EA65	LD (#65EA), A	; height snapper 1
0B9B: 32FA65	LD (#65FA), A	; height snapper 2
;		
0B9E: 3E07	LD A, #07	; A := 07
0BA0: 32C669	LD (#69C6), A	; palette snapper 1
0BA3: 32CA69	LD (#69CA), A	; palette snapper 2
;		
0BA6: 3E64	LD A, #64	; A := 64
0BA8: 32B565	LD (#65B5), A	; y-pos def upper bird 1
0BAB: 32C565	LD (#65C5), A	; y-pos def upper bird 2
0BAE: 32BB69	LD (#69BB), A	; y-pos hw upper bird 1
0BB1: 32BF69	LD (#69BF), A	; y-pos hw upper bird 2

```

;
0BB4: 3EA4      LD A, #A4      ; A := A4
0BB6: 32D565    LD (#65D5), A  ; y-pos def lower bird
0BB9: 32C369    LD (#69C3), A  ; y-pos hw upper bird 1
;
0BBC: 3EB0      LD A, #B0      ; A := B0
0BBE: 32C769    LD (#69C7), A  ; y-pos hw snapper 1
0BC1: 32CB69    LD (#69CB), A  ; y-pos hw snapper 2
;
0BC4: 3EB4      LD A, #B4      ; A := B4
0BC6: 32E565    LD (#65E5), A  ; y-pos def snapper 1
0BC9: 32F565    LD (#65F5), A  ; y-pos def snapper 2
;
0BCC: C9        RET          ; return

```

Uitgesteld beginnen met bewegen gaat niet goed. Eerst de twee timers initialiseren met 0. En dan aanpassen dat opvolgende CP niet beïnvloed wordt door de CALL (save/restore). Dan gaat de onderste vogel niet goed: blijkbaar een fireball. Nog te verplaatsen?

0B60 t/m 0BCB één verschuiven

0B60: 47 LD B,A

Verschoven stuk

0BCD: 78 LD A,B
0BCE: C9 RET

Dat helpt inderdaad.

Maar de hardware adressen van de sprites met 4 ertussen?
Dus dan 69B8 / 69BC / 69C0 / 69C4 / 69C8

Dat gaat nu goed. Nu nog beter spreiden.

De onderste vogel met vertraging laten verschijnen.

0B31 t/m 0B53 vier plekken opschuiven en het volgende ertussen zetten

0B23: FE40 CP #40 ; time to move upper bird 2 yet?
0B25: 380A JR C #GGGG ; no, skip next steps – jump relative to GGGG

0B31: 21C069	LD HL, #69C0	; load HL with lower bird hw x position – label GGGG
0B34: 34	INC (HL)	; move bird one place to the right
0B35: 34	INC (HL)	; move bird one place to the right
0B36: 21D365	LD HL, #65D3	; load HL with lower bird definition sprite x position
0B39: 34	INC (HL)	; move bird one place to the right
0B3A: 34	INC (HL)	; move bird one place to the right

Versienummer aanpassen naar versie 1.01.

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

De board per levels weer goed gezet.

Starten vanaf level 01:

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

Invinciblility uitzetten:

```

19B0 CDFB03 CALL #03FB ; check for kong beating chest and animate girl and her screams
19B3 CD0828 CALL #2808 ; check for collisions with hostile sprites [set to NOPS to make mario invincible to enemy sprites]
19B6 CD1D28 CALL #281D ; do stuff for hammers

```

Jungle versie maken: mogelijke title: JungleTwist.

De grafische elementen al voor een groot deel aangepast.

Nu nog de kleuren gaan aanpassen:

Proefondervindelijk blijkt dat de volgende blokken (in c-2j.br en c-2k.bpr) gerelateerd zijn aan de volgende schermen:

Nu vaststellen welke velden gebruikt worden voor pies en rivets.

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000	0F 06 0E 01 0F 00 01 00 0F 00 01 0F 0F 0F 01 0E
00000010	0F 01 00 01 0F 00 01 00 0F 01 0F 0E 0F 05 00 00
00000020	0F 05 00 01 0F 00 01 01 0F 0F 00 01 0F 0F 01 00
00000030	0F 00 0E 0F 0F 01 00 01 0F 02 00 0F 0F 00 01 0F
00000040	0F 06 0E 01 0F 00 01 00 0F 00 01 0F 0F 01 00 00
00000050	0F 01 00 00 0F 01 00 00 0F 01 00 00 0F 05 00 00
00000060	0F 05 00 01 0F 00 01 01 0F 0F 00 01 0F 0F 01 00
00000070	0F 00 0E 0F 0F 01 00 01 0F 02 00 0F 0F 00 01 0F
00000080	0F 06 0E 01 0F 00 01 00 0F 00 01 0F 0F 01 07 0E
00000090	0F 01 07 0E 0F 01 07 0E 0F 01 07 0E 0F 05 00 00
000000A0	0F 05 00 01 0F 00 01 01 0F 0F 00 01 0F 0F 01 00
000000B0	0F 00 0E 0F 0F 01 00 01 0F 02 00 0F 0F 00 01 0F
000000C0	0F 06 0E 01 0F 00 01 00 0F 00 01 0F 0F 0F 0E 00
000000D0	0F 0F 0E 00 0F 0F 0E 00 0F 0F 0E 00 0F 05 00 00
000000E0	0F 05 00 01 0F 00 01 01 0F 0F 00 01 0F 0F 01 00
000000F0	0F 00 0E 0F 01 00 01 0F 02 00 0F 0F 00 01 0F

Groen: Barrels, Springs en Intro

Rood: Pies en Progression

Blauw: Rivets en Titel

Voor de barrels geldt:

0F 01 07 0E : de 01 is de lichtrood, de 07 is de donkerrood, de 0E is de turquoise.

Dus de turquoise moet eigenlijk groen worden.

Palette kleurendefinities:

```

static void dkongPaletteInit()
{
    for (INT32 i = 0; i < 256; i++)
    {
        INT32 bit0 = (DrvColPROM[i+256] >> 1) & 1;
        INT32 bit1 = (DrvColPROM[i+256] >> 2) & 1;
        INT32 bit2 = (DrvColPROM[i+256] >> 3) & 1;
        INT32 r = 255 - (0x21 * bit0 + 0x47 * bit1 + 0x97 * bit2);

        bit0 = (DrvColPROM[i+0] >> 2) & 1;
        bit1 = (DrvColPROM[i+0] >> 3) & 1;
        bit2 = (DrvColPROM[i+256] >> 0) & 1;
        INT32 g = 255 - (0x21 * bit0 + 0x47 * bit1 + 0x97 * bit2);
    }
}

```

```

        bit0 = (DrvColPROM[i+0] >> 0) & 1;
        bit1 = (DrvColPROM[i+0] >> 1) & 1;
        INT32 b = 255 - (0x55 * bit0 + 0xaa * bit1);

        DrvPalette[i] = BurnHighCol(r, g, b, 0);
    }
}

```

Nu daarmee gaan stoeien.

Combinatie j/k: 03/03 geeft een middelbruine kleur.
 Combinatie j/k: 03/0B geeft een donkerbruine kleur.
 Combinatie j/k: 06/0E geeft een legergroene kleur.
 Combinatie j/k: 06/0F geeft een donkergroene kleur.
 Combinatie j/k: 00/03 geeft een middelgroene kleur.
 Combinatie j/k: 04/03 geeft een lichtergroene kleur.
 Combinatie j/k: 04/02 geeft een lichtergroene kleur.

Tijdelijke settings griders: j/k: 0F 03 03 04 / 0F 0B 03 02.

Nu zoeken naar waar de heldere groen op het titel screen gedefinieerd wordt: deductie doen...

Mooiere donkergroen: j/k: 0E BB

Settings voor de girders: j/k: 0F 03 03 0E / 0F 0B 03 0B.

Hetzelfde palette wordt ook voor elevators gebruikt en dat ziet er ook gaaf uit.

Nu nog voor de pies ook hetzelfde palette gaan gebruiken. Hoe kiezen we dat?

Palette voor pies:

```

; conveyors from #0CAF

0CDF 115D3B LD DE,#3B5D      ; load DE with start of table data for conveyors
0CE2 21867D LD HL,REG_PALETTE_A ; load HL with palette bank selector
0CE5 3601 LD (HL),#01          ; set palette bank selector
0CE7 23 INC HL                 ;
0CE8 3600 LD (HL),#00          ; clear palette bank selector
0CEA 3E09 LD A,#09             ; load A with conveyor music
0CEC 328960 LD (#6089),A       ; set music for conveyors
0CEF C3C60C JP #0CC6           ; jump back

```

Wordt gezet op A/B: 01/00 en moet worden 01/00:

0CE5: 3600
 0CE8: 3601

Dat gaat goed. Goede palette nu gekozen. Alleen de knipperende stukken, zijn het character van de rivets girders met het steen patroon. Gewone girder maken: grond.

Girder character wordt hier geladen:

```

263E: 3EB0 LD A,#B0           ; load A with girder character
264B   06B0 LD B,#B0           ; load B with girder character

```

#D0 maken. Gaat goed. Wellicht een andere graphic hiervoor gebruiken. Een vrij character kiezen. Groen met bruin maken met blaadjes.

Nu nog de oil can de kleuren geven van de elevator mounts. En dan de graphic aanpassen naar een soort van vuurpilaar. De elevator mounts hebben als kleurcode #0F.

Definitie van de oil cans (in girders en pies):

```
3E00 27 49 0C F0      ; oil can for girders  
3E04 7F 49 0C 88      ; oil can for conveyors ?
```

De twee 0C's in 0F veranderen. (of eventueel 8F). Ziet er nu ook goed uit.

Nu de rivets gaan aanpassen. Streven naar een soort maya tempel achtige structuur.

Tijdelijke settings rivets: j/k: 0F 0F 03 0E / 0F 0D 03 0B.

Alleen de blauw (2e positie: 0F / 0D) nog veranderen in een beige of grijze kleur.

Definitieve settings rivets: j/k: 0F 00 03 0E / 0F 09 03 0B.

Umbrella sprite vervangen door babana sprite. Zelfde kleur geven als de oil can: 0F.

; bonus items on conveyors

```
3E3C 53 73 0A A0      ; position of hat on pies is 53,A0  
3E40 8B 74 0A F0      ; position of purse on pies is 8B,F0  
3E44 DB 75 0A A0      ; umbrella on the pies is at DB,A0
```

; bonus items for elevators

```
3E48 5B 73 0A C8      ; hat at 5B,C8  
3E4C E3 74 0A 60      ; purse at E3,60  
3E50 1B 75 0A 80      ; umbrella on elevator is 80,1B
```

; bonus items for rivets

```
3E54 DB 73 0A C8      ; hat on rivets at DB,C8  
3E58 93 74 0A F0      ; purse on rivets at 93,F0  
3E5C 33 75 0A 50      ; umbrella on rivets at 33,50
```

De kleur is de 3e positie.

Testen met rivets ook dezelfde jungle als de andere boards. Zelfde palette geven:

j/k: 0F 03 03 0E / 0F 0B 03 0B en de girder character gelijk maken aan die van rivets.

Vaststellen welke palette voor de jumpman sprite: C8 t/m CB:

j/k: 0F 00 01 0F / 0F 0A 0F 0D. met 0F – roze – rood – blauw. Geprobeerd aan te passen, maar wordt niet mooi. Laat jumpman en Pauline maar gewoon hetzelfde.

Aanpassen van de tekst HOW HIGH CAN YOU GET ? naar WELCOME TO MY JUNGLE !!
27 15 1C 13 1F 1D 15 10 24 1F 10 1D 29 10 1A 25 1E 17 1C 15 10 36

Change the title to JUNGLETWIST.

In #3F18 staat de huidige titel: TWISTED

6E 76 24 27 19 23 24 15 14 3F

Maar daar staat direct (op #3F22) nog een stukje code achter. Dat is het stukje code voor het weergeven van de versie.

```
081C: C3223F JP #3F22 ; jump to additional code to display version  
3F22: 110103 LD DE,#0301 ; load task data for text "V1.00"  
3F25: CD9F30 CALL #309F ; insert task to draw text  
3F28: C31F08 JP #081F ; jump back
```

Dit kan nog naar achteren:

```
081C: C3273F JP #3F27 ; jump to additional code to display version  
3F27: 110103 LD DE,#0301 ; load task data for text "V1.00"  
3F2A: CD9F30 CALL #309F ; insert task to draw text  
3F2D: C31F08 JP #081F ; jump back
```

En versienummer naar 1.02 ophogen.

En dan de versie opnemen in stuk daarna:
#373E: 3D 76 26 01 2B 00 02 10 3F (V1.00)

Titel aanpassen in #3F18 naar JUNGLETWIST:

AE 76 1A 25 1E 17 1C 15 24 27 19 23 24 3F

Vaststellen kleuren high score screen en how high can you get screen.

Het wit van de tekst how high can you get staat op positie 5B van het kleurenpalette. Als je deze echter wijzigt dan heeft dat impact op de Goofy Kong. Dus gewoon zo laten!

De lichtblauwe tekst in het high score screen staat op positie 1B van het kleurenpalette. Deze veranderen in de donkergroene kleur: j/k = 0E / 0B.

Nu alleen nog het start level weer naar L=01 en de volgorde van de boards goedzetten.

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

Toch weer terug naar de originele graphics en blinking platforms board aanpassen

Aanpassen start level:

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.

Veld iets makkelijker maken door de middelste ladder op het tweede niveau gewoon heel i.p.v. gebroken te maken.

227E: 01 83 B8 83 D8 ; broken ladder center 2nd level

Aanpassen naar:

227E: 00 83 B8 83 D8 ; broken ladder center 2nd level

Kijken of het veld iets makkelijker wordt door één van de snappers te disablen.

De snapper wordt door de onderstaande code verplaatst (let niet op commentaar):

```
0B45: FE80      CP #80      ; time to move upper bird 2 yet?  
0B47: 380A      JR C #HHHH  ; no, skip next steps - jump relative to HHHH  
;  
0B49: 21C869    LD HL, #69C8  ; load HL with snapper 2 hw x position  
0B4C: 35        DEC (HL)   ; move bird one place to the right  
0B4D: 35        DEC (HL)   ; move bird one place to the right  
0B4E: 21F365    LD HL, #65F3  ; load HL with snapper 2 definition sprite x position  
0B51: 35        DEC (HL)   ; move bird one place to the right  
0B52: 35        DEC (HL)   ; move bird one place to the right
```

Dus beweging uit te zetten (snapper blijft wel actief, maar buiten beeld) door deze code allemaal door NOP's te vervangen:

0B45 t/m 0B52: NOP's

Klopt niet:

Dus gearceerde stuk NOP's maken.

Dat werkt goed. Nu maar één snapper. Deze kruist de bottom bird steeds op hetzelfde punt. Daarom wellicht de snapper sneller maken (of wellicht variabel qua snelheid?).

Snelheid aanpassen is extra DEC (#35) bij de snapper ervoor:

0B44-0B4C: 35 35 35 21 E3 65 35 35 35

Leuke snelheid want dan kan je ook rechtop springen en zelfs dezelfde kant op springen. Dit eens proberen met twee snappers...

OB44- : 35 35 35 21 E3 65 35 35 35 EE 80 38 0C 21 C8 69 35 35 35 21 E3 65 35 35 35 C9

Dezelfde code er achter terugzetten, maar ook met twee keer extra DEC. En dan de jump relative (JR) naar de RET (C9) aanpassen naar 0C (was 0A, maar twee extra DEC's tussen gevoegd).

De tweede start nu op een vreemd moment. Deze eerder laten beginnen. FE 80 aanpassen. Wellicht goede starttijd wanneer $\#80 * 2 / 3 = \#55$. Dus aanpassen naar FE 55.

De onderste vogel is nu vrij lastiq. Kijken als we deze eruit halen.

0B31: 21C069	LD HL, #69C0	; load HL with lower bird hw x position – label GGGG
0B34: 34	INC (HL)	; move bird one place to the right
0B35: 34	INC (HL)	; move bird one place to the right
0B36: 21D365	LD HL, #65D3	; load HL with lower bird definition sprite x position
0B39: 34	INC (HL)	; move bird one place to the right
0B3A: 34	INC (HL)	; move bird one place to the right

Klopt ook niet:

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000B00	5D	32	C5	69	32	C9	69	3A	A4	63	3C	32	A4	63	FE	02	J2Ãi2Éi: <c><2</c> cp.
00000B10	C0	3E	00	32	A4	63	3A	A1	65	21	B8	69	34	34	21	B3	À>.2c:;e!,i44!*
00000B20	65	34	34	FE	80	38	0A	21	BC	69	34	34	21	C3	65	34	e44p€8.!i44!Ãe4
00000B30	34	FE	20	38	0C	21	C0	69	34	34	00	21	D3	65	34	34	4b 8.!Ãi44!.!Oe44
00000B40	00	21	C4	69	35	35	35	21	E3	65	35	35	35	FE	55	38	!Ãi555!Ãe555pU8
00000B50	0C	21	C8	69	35	35	35	21	F3	65	35	35	35	C9	00	00	.!Èi555!Ãe555É..
00000B60	00	00	00	00	47	3E	01	32	B0	65	32	C0	65	32	D0	65G>.2°e2Ãe2Dè
00000B70	32	E0	65	32	F0	65	32	BA	69	32	BE	69	32	C2	69	3E	2àe2øe2°i2%i2Ãi>

Gearceerde stuk is het juiste. Daar zit in tegenstelling tot de listing nog een check met JR voor. Het lijkt erop dat de onderste bird pas na #20 gaat vliegen (FE 20).

Dus #0B31 t/m #0B40 NOP's maken.

Spawn de eerste fireball iets hoger?

```
2690: DD 21 00 64 LD    IX,#6400      ; load IX with start of fire #1
2694: DD 36 00 01 LD    (IX+#00),#01  ; set fire active
2698: DD 36 03 7F LD    (IX+#03),#7F  ; set fire X position
269C: DD 36 0E 7F LD    (IX+#0E),#7F  ; set fire X position #2
26A0: DD 36 05 D0 LD    (IX+#05),#D0  ; set fire Y position
26A4: DD 36 0F D0 LD    (IX+#0F),#D0  ; set fire Y position #2
```

De Y-positie veranderd naar B0 (2x). Gaat goed, maar wellicht weer iets te makkelijk.

Ander iets zou zijn om één fireball minder te laten spawnen. Dus ergens zit een max van vijf en die alleen voor pie-factory aanpassen naar vier.

De volgende code checkt het aantal firefoxes voordat een nieuwe geployed wordt:

```
; arrive here from #314E
317C 3AA163 LD    A,(#63A1)    ; \
317F FE05 CP    #05          ; \
3181 CA6A31 JP    Z,#316A    ; /
3184 3A2762 LD    A,(#6227)    ; \
3187 FE02 CP    #02          ; \
3189 C29531 JP    NZ,#3195    ; /
318C 3AA163 LD    A,(#63A1)    ; \
318F 4F LD    C,A          ; | on screen than the internal difficulty
3190 3A8063 LD    A,(#6380)    ; |
3193 B9 CP    C          ; |
3194 C8 RET   Z          ; /
3195 3AA063 LD    A,(#63A0)    ; \
3198 FE01 CP    #01          ; \
319A C26A31 JP    NZ,#316A    ; /
```

#318C t/m 3194 bepalen aantal firefoxes afhankelijk van de internal difficulty. Dit eerst maar eens aanpassen naar een vaste waarde.

```
318C: 3AA163 LD    A,(#63A1)    ; load A with # of fires active
318F: FE04 CP    #04          ;
3191: C8 RET   C          ; return if # of fires is four
3192-3194: NOP's
```

Dat werkt goed. Maar vier fireballs is iets beter te doen. Wellicht wel weer de eerste firefox lager laten spawnen en misschien de lower bird weer terug.

Voor het eindigen van het veld is het wellicht interessant om de firefoxes weer naar beneden te laten gaan. Nu vaak aan de linker kant vrij veel ruimte.

De volgende code is de check op de top girder en voorkomen dat de firefoxes weer naar beneden gaan.

```

; This subroutine returns to the higher subroutine (causing a ladder to NOT be taken) if a fireball is on the top girder and we are not on rivets.
; called from #33A4

33A1 3E07 LD A,#07 ; \ Return if immediately we are on rivets, fireballs do not get stuck on the top in this case
33A3 F7 RST #30 ; /

33A4 DD7E0F LD A,(IX+#0F) ; \ Return if Y-position is >= 59 (i.e., fireball is not on the top girder)
33A7 FE59 CP #59 ; |
33A9 D0 RET NC ; |

33AA 33 INC SP ; \ Else return to higher subroutine. This prevents fireballs from coming down on conveyors & girders once
33AB 33 INC SP ; | they reach the top level.
33AC C9 RET ; /

```

In #33A1 t/m 33A3 zit een uitzondering voor rivets.

De waarde #07 waarin in RST 30 gecheckt wordt betekend dat de fireballs niet naar beneden kunnen in 0111: 0 (rivets wel), 1 (elevator niet), 1 (conveyor niet), 1 (barrels niet). Dus eigenlijk nu checken op 0101 = #05.

33A1: 3E05 LD A,#05

Werkt wel, ze gaan ook weer naar beneden. Maar heel raar: ze gaan nu ook omhoog op de DK ladders en de ladder naar Pauline. Dat is niet de bedoeling.

Dit komt omdat hierdoor de fireball een dichtbijzijnde trap mag mounten. Maar dus ook een trap omhoog. Wellicht bij mounten trap omhoog checken op conveyors board en Y-waarde en dan niet mounten?

Stuk vrijmaken bij deel van de twee fireballs die niet meer gespawned worden:

```

2690: DD 21 00 64 LD IX,#6400 ; load IX with start of fire #1
2694: DD 36 00 01 LD (IX+#00),#01 ; set fire active
2698: DD 36 03 7F LD (IX+#03),#7F ; set fire X position
269C: DD 36 0E 7F LD (IX+#0E),#7F ; set fire X position #2
26A0: DD 36 05 D0 LD (IX+#05),#D0 ; set fire Y position
26A4: DD 36 0F D0 LD (IX+#0F),#D0 ; set fire Y position #2
;
26A8: C9 RET ; return
;
26A8: DD 36 20 01 LD (IX+#20),#01 ; set fire active
26AC: DD 36 23 1B LD (IX+#23),#1B ; set fire X position
26B0: DD 36 2E 1B LD (IX+#2E),#1B ; set fire X position #2
26B4: DD 36 25 D0 LD (IX+#25),#D0 ; set fire Y position
26B8: DD 36 2F D0 LD (IX+#2F),#D0 ; set fire Y position #2
;
26BC: DD 36 40 01 LD (IX+#40),#01 ; set fire active
26C0: DD 36 43 E4 LD (IX+#43),#E4 ; set fire X position
26C4: DD 36 4E E4 LD (IX+#4E),#E4 ; set fire X position #2
26C8: DD 36 45 D0 LD (IX+#45),#D0 ; set fire Y position
26CC: DD 36 4F D0 LD (IX+#4F),#D0 ; set fire Y position #2
;
26D0: C9 RET ; return

```

Dus #26A9 t/m #26D0 (en ook nog t/m #26DF) vrijmaken en testen of het dan nog werkt.

Ja dat werkt dus daar kan eventueel een extra check bijgeplaatst worden.

De onderstaande code wordt uitgevoerd wanneer er een trap naar beneden dichtbij is.

```

; Arrive because there is a ladder nearby to go up
3399 DD701F LD (IX+#1F),B ; Store B into +#1F = Y-position of top of ladder
339C DD360D08 LD (IX+#0D),#08 ; Else set direction to ascending ladder
33A0 C9 RET ; return

```

Dus hier een test tussenvoegen die test op het conveyors level en bovenste girder:

3399: C3A926 JP #26A9

En

```
26A9: 3A2762 LD A,(#6227) ; load screen nr
26AC: FE02 CP #02
26AE: C2BA26 JP NZ,#26BA ; skip next steps if not on conveyors: Jump to label AAAA
;
26B1: DD7E0F LD A,(IX+#0F) ; load Y-coordinate of fireball
26B4: FE59 CP #59
26B6: D2BA26 JP NC, #26BA ; skip next steps if Y-coordinate > 59
;
26B9: C9 RET ; return when on top girder on conveyors
;
26BA: DD701F LD (IX+#1F),B ; store B into +#1F = Y-position of top of ladder - AAAA
26BD: C39C33 JP #339C
```

Dit lijkt goed te werken. Nog wel testen of dit geen invloed heeft op de girders en rivets fireballs. Checken met invincibility.

Interessant om te spelen nu, soms de doortocht bovenin nog vrij eenvoudig. Wellicht de lower bird mee laten vliegen bovenin, zodat er daar drie vliegen?

Dit is de onderste bird die na 20 pas gaat vliegen. Is #0B31 t/m #0B40.

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
00000B00	5D 32 C5 69 32 C9 69 3A A4 63 3C 32 A4 63 FE 02]2Äi2Éi: mc<2mc .
00000B10	C0 3E 00 32 A4 63 3A A1 65 21 B8 69 34 34 21 B3	À>.2 mc :;e!,i44!*
00000B20	65 34 34 FE 80 38 0A 21 BC 69 34 34 21 C3 65 34	e44p€8.! i44!Äe4
00000B30	34 FE 20 38 0C 21 C0 69 34 34 00 21 D3 65 34 34	4 p 8.!Äi44.!Öe44
00000B40	00 21 C4 69 35 35 35 21 E3 65 35 35 35 FE 55 38	! Äi555!äe555pU8
00000B50	0C 21 C8 69 35 35 35 21 F3 65 35 35 35 C9 00 00	.!Èi555!öe555É..
00000B60	00 00 00 00 47 3E 01 32 B0 65 32 C0 65 32 D0 65G>.2°e2Äe2De
00000B70	32 E0 65 32 F0 65 32 BA 69 32 BE 69 32 C2 69 3E	2äe2øe2°i2% i2Äi >

Verplaatsen naar Y-coordinaat van bovenin?

```
0BA6: 3E64 LD A, #64 ; A := 64
0BA8: 32B565 LD (#65B5), A ; y-pos def upper bird 1
0BAB: 32C565 LD (#65C5), A ; y-pos def upper bird 2
0BAE: 32BB69 LD (#69BB), A ; y-pos hw upper bird 1
0BB1: 32BF69 LD (#69BF), A ; y-pos hw upper bird 2
;
0BB4: 3EA4 LD A, #A4 ; A := A4
0BB6: 32D565 LD (#65D5), A ; y-pos def lower bird
0BB9: 32C369 LD (#69C3), A ; y-pos hw lower bird
```

Dit kan dus door het laden van de nieuwe waarde in #0BB4 niet te doen. Dan krijgt de lower bird dezelfde y-waarde als de upper birds.

#0BB4 - #0BB5: NOP NOP

Klopt ook niet.

Het gearceerde gedeelte is het juiste:

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000B20	65	34	34	FE	80	38	0A	21	BC	69	34	34	21	C3	65	34	e44þ€8.!Ài44!Ãe4
00000B30	34	FE	20	38	0C	21	C0	69	34	34	00	21	D3	65	34	34	4þ 8.!Ài44!.Óe44
00000B40	00	21	C4	69	35	35	35	21	E3	65	35	35	35	FE	55	38	.!Ài555!âe555þU8
00000B50	0C	21	C8	69	35	35	35	21	F3	65	35	35	35	C9	00	00	.!Èi555!óe555É..
00000B60	00	00	00	00	47	3E	01	32	B0	65	32	C0	65	32	D0	65G>.2ºe2Àe2De
00000B70	32	E0	65	32	F0	65	32	BA	69	32	BE	69	32	C2	69	3E	2àe2àe2ºi2¾i2Ài>
00000B80	06	32	B9	65	32	C9	65	32	D9	65	3E	05	32	BA	65	32	.2¹e2Èe2Ùe>.2ºe2
00000B90	CA	65	32	DA	65	32	E9	65	32	F9	65	3E	06	32	EA	65	Èe2Ùe2éé2Ùe>.2èe
00000BA0	32	FA	65	3E	07	32	C6	69	32	CA	69	3E	64	32	B5	65	2Ùe>.2Æi2Èi>d2þe
00000BB0	32	C5	65	32	BB	69	32	BF	69	3E	A4	32	D5	65	32	C3	2Àe2»i2¿i>Àðe2À
00000BC0	69	3E	B0	32	C7	69	32	CB	69	3E	B4	32	E5	65	32	F5	i>²C¿i2Èi>'2àe2ò
00000BD0	65	78	C9	00	00	00	00	00	00	CD	1C	01	DF	CD	74	exÉ.....í..ßít	

Dus dat betekent dan:

#3BB9 - #3BBA: NOP NOP

Dit gaat goed. De onderste vogel vliegt nu als derde vogel boven.

Nu nog goed afstand van elkaar houden.

De loop is 240 lang, dus blokken van 80 maken, dus bird 2 na #50 en bird 3 na #A0 starten.

2e bird:

0B23: FE80 CP #80 ; time to move upper bird 2 yet?

Veranderen in:

0B23: FE50 CP #50 ; time to move upper bird 2 yet?

3e bird:

0B31: FE20 CP #20 ; time to move lower bird yet?

Veranderen in:

0B31: FEA0 CP #A0 ; time to move lower bird yet?

De onderlinge afstand klopt niet. Komt omdat de loop niet 240 is , maar 256.
Dus dan tweede en derde bird starten op #55 en #AA.
Dat gaat nu goed.

Wel vaak dat de fireboxes zich toch aan de rechterkant verzamelen. Wellicht mogelijk om ze toch te spawnen aan de kant waar jumpman staat. Dan in eerste instantie links en dan duurt het wellicht wat langer voordat ze aan de rechterkant zijn. Daarnaast kun je ze dan ook niet met geluk 'weghameren'.

Daartoe #348A t/m 348E herstellen:

```
3487 21AC3A LD HL,#3AAC ; load HL with start of table data
348A 3A0362 LD A,(#6203) ; \ Jump if Mario is on left side of the screen, in this case we spawn the fireball on the left
348D CB7F BIT 7,A ; |
348F CAA834 JP Z,#34A8 ; /
```

Nee dat is geen goed idee. Ze blijven nu wel heel erg links hangen. Komen niet snel naar rechts. Daardoor aan rechterkant makkelijke doorgang.

Terugzetten.

Zo laten. Versie aanpassen.

Versienummer aanpassen naar versie 1.03.

#373E: 3D 76 26 01 2B 00 **03** 10 3F (V1.03)

De boards per levels weer goedgezet.

Invincibility aan en de levels testen

Starten vanaf level 01:

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

Invincibility uitzetten:

```
19B0 CDFB03 CALL #03FB ; check for kong beating chest and animate girl and her screams  
19B3 CD0828 CALL #2808 ; check for collisions with hostile sprites [set to NOPs to make mario invincible to enemy sprites]  
19B6 CD1D28 CALL #281D ; do stuff for hammers
```

Weer een Jungle variant ervan maken

Daartoe eerst de palettes en sprite files in de rom zetten en dan de verschillende aanpassingen qua laden van palettes en keuze kleurcodes goedzetten. Daarna doorontwikkelen: kokosnoten ipv barrels, manchete ipv hamer, vuurpilaar anders, items alleen bananen, goofy kong aanpassen.

De sprite files en palette file in de zip-file gekopieerd.

De oil can is niet de goede kleur.

Definitie van de oil cans (in girders en pies):

```
3E00 27 49 0C F0 ; oil can for girders  
3E04 7F 49 0C 88 ; oil can for conveyors ?
```

De twee 0C's in 0F veranderen. (of eventueel 8F). Ziet er nu ook goed uit.

De items zijn niet de goede kleur.

Zelfde kleur geven als de oil can: 0F.

; bonus items on conveyors

```
3E3C 53 73 0A A0 ; position of hat on pies is 53,A0  
3E40 8B 74 0A F0 ; position of purse on pies is 8B,F0  
3E44 DB 75 0A A0 ; umbrella on the pies is at DB,A0
```

; bonus items for elevators

```
3E48 5B 73 0A C8 ; hat at 5B,C8  
3E4C E3 74 0A 60 ; purse at E3,60  
3E50 1B 75 0A 80 ; umbrella on elevator is 80,1B
```

; bonus items for rivets

```
3E54 DB 73 0A C8 ; hat on rivets at DB,C8  
3E58 93 74 0A F0 ; purse on rivets at 93,F0  
3E5C 33 75 0A 50 ; umbrella on rivets at 33,50
```

Lijkt nu allemaal goed. Alleen de palettes in pies is nog niet goed.

Palette voor pies:

```
; conveyors from #0CAF

0CDF 115D3B LD DE,#3B5D      ; load DE with start of table data for conveyors
0CE2 21867D LD HL,REG_PALETTE_A ; load HL with palette bank selector
0CE5 3601 LD (HL),#01        ; set palette bank selector
0CE7 23 INC HL              ;
0CE8 3600 LD (HL),#00        ; clear palette bank selector
0CEA 3E09 LD A,#09          ; load A with conveyor music
0CEC 328960 LD (#6089),A    ; set music for conveyors
0CEF C3C60C JP #0CC6        ; jump back
```

Wordt gezet op A/B: 01/00 en moet worden 01/00:

0CE5: 3600
0CE8: 3601

Dat gaat goed. Goede palette nu gekozen. Alleen de knipperende stukken, zijn het character van de rivets girders. Gewone girder maken: grond.

Girder character wordt hier geladen:

```
263E: 3EB0 LD A,#B0      ; load A with girder character
264B 06B0 LD B,#B0      ; load B with girder character
```

#D0 maken. Gaat goed. Wellicht een andere graphic hiervoor gebruiken.

De hartjes zijn zwart i.p.v. roze. Dat zijn de items van het conveyors field:

; bonus items on conveyors

```
3E3C 3B 76 09 D0      ; position of hat on pies
3E40 C4 76 09 D0      ; position of purse on pies
3E44 7F 76 09 90      ; umbrella on the on pies
```

Dat is nu ook goed.

Deze versie eventjes opslaan: v1.03.02.

De items bij elevators and rivets allemaal bananen maken.

; bonus items for elevators

```
3E48 5B 73 0A C8      ; hat at 5B,C8
3E4C E3 74 0A 60      ; purse at E3,60
3E50 1B 75 0A 80      ; umbrella on elevator is 80,1B
```

; bonus items for rivets

```
3E54 DB 73 0A C8      ; hat on rivets at DB,C8
3E58 93 74 0A F0      ; purse on rivets at 93,F0
3E5C 33 75 0A 50      ; umbrella on rivets at 33,50
```

Banaan is 75. De banaan ietsje kleiner maken. En hoger: 1 of 2 pixel's van de grond af.

Deze versie eventjes opslaan: v1.03.03.

Aanpassen van de tekst HOW HIGH CAN YOU GET ? naar WELCOME TO MY JUNGLE !!

27 15 1C 13 1F 1D 15 10 24 1F 10 1D 29 10 1A 25 1E 17 1C 15 10 36

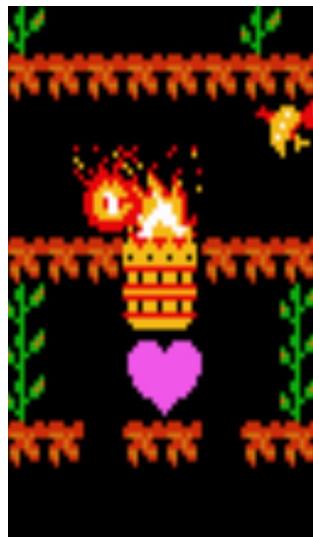
Titel aanpassen naar JUNGLE

10 10 1A 25 1E 17 1C 15 10 10

Deze versie eventjes opslaan: v1.03.04.

Oil can aangepast naar een soort van mand ipv pilaar.

Deze mand iets smaller gemaakt. Maar nu in pies veld niet helemaal in het midden.



Kijken of dit terug kan naar het midden.

```
3E00 27 49 0C F0      ; oil can for girders  
3E04 7F 49 0C 88      ; oil can for conveyors ?
```

X-waarde oil can voor conveyors is #3E04: 80. Ook getest met 81, maar dat is te ver.

Misschien ook het vuurtje 1 positie naar rechts verschuiven?

Dat kan door het aanpassen van de datatable hiervoor:

```
3DFA: 7F 40 01 78 02 00      ; initial data for conveyors to release a fire ?  
#3DFA: 80 40 01 70 02 00      ; initial data for conveyors to release a fire ?
```

80 is de nieuwe X-waarde en 70 was de al eerder aangepaste Y-waarde.

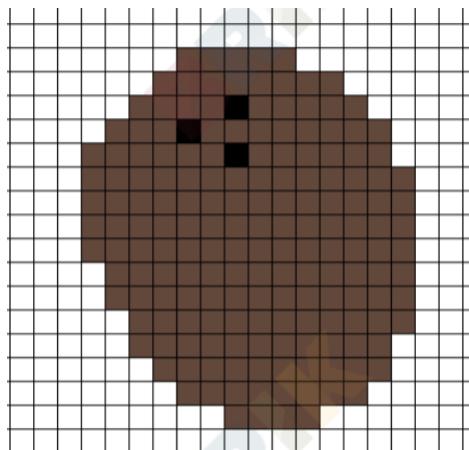


[Deze versie eventjes opslaan: v1.03.05.](#)

Bovenin is het pies screen nog wat makkelijk. Wellicht lastiger te maken door aan beide zijden een gat in de girder te maken. Dan moet je met springen op de vogels letten èn fireballs die weer eventueel naar beneden kunnen komen.

Maar nu eerst de barrels in kokosnoten veranderen.

Wellicht deze als uitgangspunt? En dan kijken hoe die rolt?



Kokosnoten gemaakt. Nu nog andere kleur maken. De Donkey Kong heeft de goede kleuren. Welke kleur wordt er aan die sprites gegeven? Lijkt kleurenpalet 08 te zijn.

Waar worden de barrel sprites geinitialiseerd? Daar de kleur aanpassen. Dat lijkt hier te gebeuren.

```

; IX holds 6700 +N*20 = start of barrel N info
; a barrel is being deployed

2CF6 DD360715 LD      (IX+#07),#15    ; set barrel sprite value to #15
2CFA DD36080B LD      (IX+#08),#0B    ; set barrel color to #0B
2CFE DD361500 LD      (IX+#15),#00    ; set +15 indicator to 0 = normal barrel, [1 = blue barrel]
2D02 3A8263 LD      A,(#6382)      ; load A with Crazy/Blue barrel indicator
2D05 07 RLCA          ; is this a blue barrel ?
2D06 D2152D JP      NC,#2D15      ; No blue barrel, then skip next 3 steps

; blue barrel

2D09 DD360719 LD      (IX+#07),#19    ; set sprite for blue barrel
2D0D DD36080C LD      (IX+#08),#0C    ; set sprite color to blue
2D11 DD361501 LD      (IX+#15),#01    ; set blue barrel indicator

```

De sprite color voor de normale barrel is 08 gemaakt en van de blue barrel 0B. Ziet er nu goed uit. Donkere kokosnoot als normale kokosnoot en lichte kokosnoot als speciale die het vuur ontsteekt en ook meer punten waard is wanneer deze kapot geslagen wordt.

[Deze versie eventjes opslaan: versie v1.03.06.](#)

De pile met barrels veranderen in een pile met kokosnoten. Sprite is aangepast. De kleur zelfde maken als de normale kokosnoot: 08.

```

; data used for the barrel pile next to kong
; called from #0FD7

3DDC 1E 18 0B 4B      ; first barrel
3DE0 14 18 0B 4B      ; second barrel
3DE4 1E 18 0B 3B      ; third barrel
3DE8 14 18 0B 3B      ; fourth barrel

```

Dit wordt nu (naar links verschoven, naar beneden verschoven en 2e afwijkende sprite en kleur):

```

1D 18 08 4C
12 1C 0B 4C
1D 18 08 3D
12 18 08 3D

```

[Deze versie eventjes opslaan: versie 1.03.07.](#)

De springs aanpassen. Aangepast in regendruppels. Weet nog niet of dat goed is. Zijn vrij groot. Meer waterbombs. Maar toch opslaan.

[Deze versie eventjes opslaan: versie 1.03.08.](#)

De springs nogmaals aanpassen. Aangepast in een springrups. De kleur daarvan nog aanpassen. De springs worden hier geïnitialiseerd.

```

; called from #102B and #1093

1186 21A211 LD HL,#11A2 ; load HL with start of data table
1189 110765 LD DE,#6507 ; load DE with destination
118C 010C0A LD BC,#0A0C ; set counters
118F CD2A12 CALL #122A ; copy

1192 DD210065 LD IX,#6500 ; load IX with ???
1196 218069 LD HL,#6980 ; load HL with sprite start (??)
1199 060A LD B,#0A ; B := #A
119B 111000 LD DE,#0010 ; load DE with offset
119E CDD311 CALL #11D3 ; copy

11A1 C9 RET ; return

```

; data table used above

11A2 3B 00 02 02

De #11A2 is de sprite waarde en de #1103 is de kleur van de springs.

Goed kleuren:

02 (rood/blauw), - aardig niet heel goed zichtbaar.
~~08 (creme/oranje), niet zo heel mooi, beetje flets.~~
~~09 (roze/oranje), beetje flets, maar witte ogen komen goed uit.~~
~~0C (lichtblauw/donkerblauw), erg contrasterend met het veld.~~
~~0D (geel/rood), aardig maar lijkt veel op elevator mounts en bananen.~~
~~0E (creme/blauw), beetje vreemd, rare mond door de kleuren.~~
0F (rood/zwart). - aardig, niet heel goed zichtbaar, ogen komen goed uit.

Vooralsnog heeft 02 de voorkeur.

Deze versie eventjes opslaan: versie v1.03.09.

Nu nog de titel en de versie veranderen.

Versienummer aanpassen naar versie 1.04.

#373E: 3D 76 26 01 2B 00 04 10 3F (V1.04)

Stukje code van 380D kan vrijgegeven worden. Table data voor de intro waarin Donkey Kong Pauline ontvoert. Maar dat wordt niet meer gebruikt.

Titel veranderen in TWISTED JUNGLE. Dat past niet op de oude plek. Daarom plaatsen in vrije stuk vanaf #380D. Stond in 3F18. Dat wordt hier aangeroepen:

En de verwijzing ernaar toe aanpassen:

3689: 18 3F

Dus nu dan:

3689: 0D 38

En:

380D: AE 76 24 27 19 23 24 15 14 10 1A 25 1E 17 1C 15 3F

Aangepast.

Deze versie eventjes opslaan: versie v.03.10.

Het HOW HIGH scherm aanpassen naar hetzelfde kleurenpalette als alle andere schermen. Waar wordt dat gezet. Hier:

```
0BEA 21867D LD HL,REG_PALETTE_A ; load HL with palette bank
0BED 3601 LD (HL),#01 ; set palette bank selector
0BEF 23 INC HL ; next palette bank
0BF0 3600 LD (HL),#00 ; clear palette bank selector
```

Moet 0,1 worden. Is nu goed. De goede kleur teksten.

#OBFF t/m #0C81 doet het tekenen van de meters en de Goofy Kongs. Dit is niet meer nodig. Eventjes voor nu allemaal NOP maken. Gaat goed. Er staat nu alleen nog: Welcome To My Jungle !

Nu de Kong uit het titelscherm overnemen.

```
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
```

Dat stuk overnemen vanaf 0BFF:

```
0BFF 21CF39 LD HL,#39CF ; load HL with table data for kong beating chest
0C02 CD4E00 CALL #004E ; update kong's sprites
0C05 210869 LD HL,#6908 ; load HL with start of kong sprite X pos
0C08 0E44 LD C,#44 ; load C with offset to add X
0C0A FF RST #38 ; draw kong in new position
0C0B 210B69 LD HL,#690B ; load HL with start of kong sprite Y pos
0C0E 0E78 LD C,#78 ; load C with offset to add Y
0C10 FF RST #38 ; draw kong
```

En daar dan ook de titel weergeven:

Tekst kun je zo weergeven:

```
080C 111E03 LD DE,#031E ; load task data for text "(C) 1981"
080F CD9F30 CALL #309F ; insert task to draw text
```

De titel staat in #380D en wordt gerefereerd in #3689. Dat is tekst nummer 1F.

```
0C11 111F03 LD DE,#031F ; load task data for title
0C14 CD9F30 CALL #309F ; insert task to draw text
```

Een girder en twee lianen tekenen.

```

0C6d: 02 EF A1 10 A1 ; top girder
0C72: 02 EF A1 10 A1 ; bottom girder
0C77: 00 53 18 53 54 ; kong's ladder (left)
0C7C: 00 63 18 63 54 ; kong's ladder (right)
0C81: AA              ; end of data code

```

Datatabel maken en dan aanroepen met:

```

0C17 116D0C LD DE,#0C6D ; load DE with start of table data
0C1A CDA70D CALL #0DA7 ; draw the screen

```

En over de datatabel heen springen.

```
0C6F C3820C JP #0C82 ; skip data table
```

Kijken toevoegen van heart sprite.

```

3F4B: 21206A LD HL,#6A20 ; load HL with heart sprite
3F4E: 36D6 LD (HL),#D6 ; set heart sprite x-position
3F50: 23 INC HL ; next
3F51: 3676 LD (HL),#76 ; set heart sprite
3F53: 23 INC HL ; next
3F54: 3609 LD (HL),#09 ; set heart sprite color
3F56: 23 INC HL ; next
3F57: 36C8 LD HL,#C8 ; set heart sprite y-position

```

4x opnemen met verschillende sprites.

Kleuren goedzetten:

Kokosnoot kleur: 08.

Snapper kleur: 07.

Rups kleur: 02.

Vogel kleur: 01.

Is nu helemaal klaar.

Deze versie eventjes opslaan: v1.03.11.

Het pies screen is wellicht nog te eenvoudig. Vaak een free pass aan de linkerkant. Wellicht de vogel terugbrengen? Of anders alleen in dit veld de fireballs niet voorkeur naar rechts laten hebben.

De vogel terugbrengen kan niet omdat deze al bovenin vliegt: zijn er drie.

Eventjes uitzoeken hoe het komt dat de fireballs voorkeur voor rechts hebben. Komt dat omdat ze sneller naar rechts dan naar links bewegen?

Lijkt hierdoor te komen:

```

321E CD0F33 CALL #330F ; Check if fireball should randomly reverse direction
3221 3A1860 LD A,(RngTimer1) ; \ Jump and do not climb any ladder with 75% probability, so a ladder is climbed with 25% probability.
3224 E603 AND #03 ; | Note that left moving fireballs always skip the ladder climbing check and instead jump to the end of
3226 C23332 JP NZ,#3233 ; / this subroutine without updating position.

3229 DD7E0D LD A,(IX+#0D) ; \ Jump to end of subroutine if fireball is moving left. This is reached with 25% probability so left-moving
322C A7 AND A ; | fireballs skip all movement with 25% probability, so their speed is randomized but averages 25% slower
322D CA5732 JP Z,#3257 ; / than the speed of right-moving fireballs.

```

Testen door 3229 t/m 322F NOP's te maken. Dan slaat hij de skip in 25% van de gevallen van links gaande fireballs niet over.

Dit werkt wel, maar nu hebben de fireballs meer een voorkeur voor links en heb je vrij vaak een vrije doorgang aan de rechterkant. Dus maar niet doen. Terugzetten naar opgeslagen versie.

Iets wat het ook nog weer lastiger kan maken is weer terug naar 5 fireballs. Zijn er nu 4. Even terugzoeken hoe dit op 4 gezet is.

Dat was deze aanpassing:

#318C t/m 3194 bepalen aantal fireboxes afhankelijk van de internal difficulty. Dit eerst maar eens aanpassen naar een vaste waarde.

```
318C: 3AA163 LD A,(#63A1) ; load A with # of fires active
318F: FE04 CP #04 ;
3191: C8 RET C ; return if # of fires is four
3192-3194: NOP's
```

Hier de waarde 05 inzetten? Ja dat gaat goed. Nu weer vijf fireballs. Daardoor veld iets lastiger. Maar dat is prima. Nog wel verder mee testen. Vooralsnog zo laten.

Deze versie eventjes opslaan: v1.03.12.

Bij finish van pies level springt Kong een aantal posities naar boven en gaat dan klimmen. Dat zou kunnen komen door de onderstaande code:

```
1662 218863 LD HL,#6388 ; load HL with end of level counter
1665 34 INC (HL) ; increase counter
1666 3E01 LD A,#01 ; A := 1 = code for girders
1668 F7 RST #30 ; if girders, continue below. else RET

1669 210B69 LD HL,#690B ; load HL with start of kong sprite
166C 0EFC LD C,#FC ; set movement for -4 pixels
166E FF RST #38 ; move kong
166F C9 RET ; return
```

Omdat het einde voor griders en pies hetzelfde moet zijn, moet de test op het screen uitgebreid worden:

```
1666 3E03 LD A,#03 ; A := 3 = code for girders and conveyors
```

Kijken of het einde voor alle schermen goed is. Checken met invincibility aan.

19B3: CD0828 : deze NOP maken.

Nee gaat niet goed. Gaat alleen maar extra omhoog. Blijkt dat girders hoger is dan conveyors. Wellicht de offset doen uit het volgende stuk:

```
1680 3E04 LD A,#04 ; A := 4 = 100 code for elevators
1682 F7 RST #30 ; only continue here if elevators, else RET

1683 210B69 LD HL,#690B ; load HL with start of Kong sprite
1686 0E04 LD C,#04 ; set to move by 4
1688 FF RST #38 ; move kong by +4
1689 C9 RET ; return
```

Dan hier de test uitbreiden met pies:

```
1680 3E06 LD A,#06 ; A := 6 = 110 code for elevators and conveyors
```

Nu werkt de finish animatie wel mooi in het pies screen.

Deze versie eventjes opslaan: v1.03.13.

Een versie maken met invincibility aan en juiste volgorde velden en startend bij level 1.

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

Deze versie eventjes opslaan: v1.03.14.

Getest: kill screen gaat goed. Level 1 t/m 6 doorloopt de screens goed.

Wellicht een idee om na de L=01 ook het screen aan te geven:

L=01-1 / L=01-2 / L=02-1 / L=02-2 / L=02-3 / L=03-1 / L=03-2 / L=03-3 / L=03-4 / ... / ...

Waar wordt de L=xx getekend? Dat gebeurt hier.

```
06D7 210375 LD HL,#7503 ; load HL with screen location for "L="
06DA 361C LD (HL),#1C ; draw "L"
06DC 21E374 LD HL,#74E3 ; next location
06DF 3634 LD (HL),#34 ; draw "="
06E1 3A2962 LD A,(#6229) ; load A with level #
06E4 fe64 CP #64 ; level < #64 (100 decimal) ?
06E6 3805 JR C,#06Ed ; yes, skip next 2 steps

06E8 3E63 LD A,#63 ; otherwise A := #63 (99 decimal)
06Ea 322962 LD (#6229),A ; store into level #

06Ed 010Aff LD BC,#ff0A ; B: = #FF, C := #0A (10 decimal)

06f0 04 INC b ; increment B
06f1 91 SUB c ; subtract 10 decimal
06f2 d2f006 JP NC,#06f0 ; not carry, loop again (counts tens)

06f5 81 ADD A,C ; add 10 back to A to get a number from 0 to 9
06f6 32A374 LD (#74A3),A ; draw level to screen (low byte)
06f9 78 LD A,b ; load a with b (number of tens)
06fa 32C374 LD (#74C3),A ; draw level to screen (high byte)
06fd c9 RET ; return
```

622E is het aantal Goofy Kongs dat getekend moet worden: dat is een indicatie voor het schermnummer binnen een level.

Aanpassing:

06FA C32038 JP #3820 ; jump to additional code – jump to label **AAAA**

en

3820 32C374 LD (#74C3),A ; draw level to screen (high byte) – label **AAAA**
;
3823 3E2C LD A,#2C ; load A with "-"
3825 328374 LD (#7483),A ; draw hyphen

```

;
3828 3A2E62 LD A,(#622E) ; load A with number of Goofy Kongs to draw
382B 3C INC A ; increment A
382C 326374 LD (#7463), A ; draw screen in level number
;
382F C9 RET ; return

```

Dit werkt nog niet goed. De variabele #622E wordt niet geupdate. Dat zat in het stuk bij het tekenen van de Goofy Kongs en dat is eruit gehaald.

Bij finish level girders, conveyors and elevators: #622E met 1 ophogen.
Bij finish level rivets #622E weer op 0 zetten.

Finish level girders, conveyors and elevators laatste step is code vanaf #178E.

```

17B0 3630 LD (HL),#30 ; set timer to #30
17B2 23 INC HL ; HL := GameMode2
17B3 3608 LD (HL),#08 ; set game mode2 to 8
17B5 C9 RET ; return

```

Aanpassen:

```

17B3 C33038 JP #3830 ; jump to additional code, jump to label AAAA
en

3830 3608 LD (HL),#08 ; set game mode2 to 8 – label AAAA
;
3832 212E62 LD HL, #622E ; load HL with variable
3835 34 INC (HL) ; increase variable
;
3836 C9 RET ; return

```

Finish level rivets is code vanaf #18C6.

```

1937 3E6f LD A,#6f ; else A := #6F
1939 32206A LD (#6A20),A ; store A into heart sprite X position
193C c9 RET ; return from sub

```

Aanpassen:

```

1939 C33738 JP #3837 ; jump to additional code, jump to label AAAA
en

3837 32206A LD (#6A20),A ; store A into heart sprite X position
;
383A 3E00 LD A,#00 ; A := 00
383C 322E62 LD (#622E),A ; store A in variable
;
383F C9 RET ; return

```

Werkt nu wel goed. Nog wel eventjes kijken. Bij L=01 rivets springt de teller vrij vroeg om. Bij alle andere levels gaat het goed. En bij start (L=00) staat er L=00-1 op het scherm.

Deze versie eventjes opslaan: versie v1.03.15.

Bij start van het spel staat er L=00-1 op het scherm. Dit mogelijk op te lossen door altijd een 0 te tonen wanneer het level 0 is.

Dit was toegevoegd:

```
06FA C32038 JP #3820 ; jump to additional code – jump to label AAAA  
en  
3820 32C374 LD (#74C3),A ; draw level to screen (high byte) – label AAAA  
;  
3823 3E2C LD A,#2C ; load A with “-”  
3825 328374 LD (#7483), A ; draw hyphen  
;  
3828 3A2E62 LD A,(#622E) ; load A with number of Goofy Kongs to draw  
382B 3C INC A ; increment A  
382C 326374 LD (#7463), A ; draw screen in level number  
;  
382F C9 RET ; return
```

Dit aanpassen naar:

```
3820 32C374 LD (#74C3),A ; draw level to screen (high byte) – label AAAA  
;  
3823 3E2C LD A,#2C ; load A with “-”  
3825 328374 LD (#7483), A ; draw hyphen  
;  
3828 3A2962 LD A,(#6229) ; load A with level number  
382B FE00 CP #00 ; is the level number 0?  
;  
382D C34038 JP #3840 ; jump to additional code, jump to label BBBB
```

en

```
3840 2804 JR Z, #3820 ; yes, skip next lines – label BBBB  
3842 3A2E62 LD A,(#622E) ; load A with number of Goofy Kongs to draw  
3845 3C INC A ; increment A  
3846 326374 LD (#7463), A ; draw screen in level number  
;  
3849 C9 RET ; return
```

Gaat nu goed. Bij begin L-00-0 en daarna wordt het goed opgehoogd binnen de levels. Ook bij two player game wordt dit onthouden en goed weergegeven.

[Deze versie eventjes opgeslagen: versie v1.03.16.](#)

Nu nog eventjes kijken naar of het level in de originele game bij L=01 rivets ook vroeg verspringt. Nee dat gebeurt niet. Hoe kan dat?

Hier wordt het level nummer opgehoogd:

```
194B 222A62 LD (#622A),HL ; store  
194E 322762 LD (#6227),A ; store A into screen number  
1951 212962 LD HL,#6229 ; load HL with level number address  
1954 34 INC (HL) ; increase #6229 by one  
1955 110005 LD DE,#0500 ; load task #5, parameter 0 ; adds bonus to player's score  
1958 CD9F30 CALL #309F ; insert task
```

Lijkt op code die ook voor niet-rivets doorlopen wordt, dus begrijp ik niet. Maar wat als we het resetten van het screen hier doen.

Dan 1939 terugzetten.

```
1939 32206A LD (#6A20),A ; store A into heart sprite X position
```

En dan het volgende aanpassen:

```
1955 CD3738 CALL 3837 ; jump to subroutine
```

en

```
3837 110005 LD DE,#0500 ; load task #5, parameter 0 ; adds bonus to player's score  
;  
383A 3E00 LD A,#00 ; A := 00  
383C 322E62 LD (#622E),A ; store A in variable  
;  
383F C9 RET ; return
```

Maakt niet uit. Gaat nog op dezelfde manier te vroeg. Vorige versie terugzetten en gaan testen.

Testen:

- Testen kill screen – spel geeft kill screen bij level L=22-1. Geeft ook sublevel goed aan.
- Testen spelen two players – gaat goed, (sub)levels onthouden.
- Test spelen one player t/m level 17 – gaat goed, (sub)levels goed.

Invincibility uitzetten:

```
19B3: CD0828
```

Deze versie eventjes opslaan: versie v1.03.17.

Deze versie is de final versie v1.04.

Testplaying wijst uit dat de pies te lastig is met 5 fireballs. Daarom aanpassen naar 4. Om te testen eventjes grof aangepast door #317F te veranderen in FE04 (maximum van 4 fireballs). Maar waarschijnlijk verandert dat ook het maximale aantal in andere velden. Later afhankelijk maken van screen.

Ja dat gaat wel beter. Soms nog heel afhankelijk van geluk. Dus misschien zelfs wel 3 maken. Daar eventjes mee testen. Nee drie is te makkelijk. Dus vier maken. Kijken waar je het stukje code kan tussen voegen:

```
317C 3AA163 LD A,(#63A1) ; \ Jump back and don't deploy fire if there are already 5 fires active  
317F FE05 CP #05 ; |  
3181 CA6A31 JP Z,#316A ; /
```

```
317C C35F3F JP #3F5F ; jump to additional code – jump to label AAAA
```

en

```
3F5F 3A2762 LD A,(#6227) ; load A with screen number – label AAAA  
3F62 FE02 CP #02 ; is screen number pie screen?  
3F64 3AA163 LD A,(#63A1) ; load A with number of active fireballs  
3F67 C27F31 JP NZ,#317F ; no pie screen, jump back and test for 5 fireballs  
3F6A FE04 CP #04 ; already 4 fireballs active?  
3F6C C38131 JP #3181 ; jump back
```

Op #3F5F nog een deel van de system message vrij te maken. Daar plaatsen.

Gaat goed. In pies maar 4 fireballs en in girders and rivets 5 fireballs.

Nu nog de hitbox van de snapjaws aanpassen. Is nu 5, maar aanpassen naar 4:

```
0B8A: 3E05      LD A, #05      ;A := 05
0B8C: 32BA65    LD (#65BA), A   ; height upper bird 1
0B8F: 32CA65    LD (#65CA), A   ; height upper bird 2
0B82: 32DA65    LD (#65DA), A   ; height lower bird
0B95: 32E965    LD (#65E9), A   ; width snapper 1
0B98: 32F965    LD (#65F9), A   ; width snapper 2
```

Aanpassen naar:

```
0B95 C3703F    JP  #3F70      ; jump to additional code - jump to label AAAA
en

3F70 3E04      LD A, #04      ; A := 04 - label AAAA
3F72 32E965    LD (#65E9), A   ; width snapper 1
3F75 C3980B    JP  #0B98      ; jump back
```

Gaat goed.

Nu deze wijzigingen doorvoeren op een verse kopie.

En testen:

Aanpassen start level:

```
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.

Versienummer aanpassen naar versie 1.05.

```
#373E: 3D 76 26 01 2B 00 05 10 3F (V1.05)
```

Jaartal ook aanpassen. Release in 2021. Staat op #3F00.

© 1981,2021 NINTENDO

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

Misschien nog zo maken dat maximale aantal fireballs by pies afhangt van level. Tot level 10 maximaal 4 en vanaf level 10 maximaal 5.

```
3F5F 3A2762    LD   A,(#6227)    ; load A with screen number - label AAAA
3F62 FE02      CP   #02        ; is screen number pie screen?
3F64 3AA163    LD   A,(#63A1)    ; load A with number of active fireballs
3F67 C27F31    JP   NZ,#317F    ; no pie screen, jump back and test for 5 fireballs
;
3F6A 4F         LD   C,A        ; save A to C
3F6B 3A2962    LD   A,(#6229)    ; load A with level number
3F6E FE0A      CP   #0A        ; is level 10 or higher
3F70 D27F31    JP   NC,#317F    ; yes, jump back and test for 5 fireballs
;
3F73 79         LD   A,C        ; restore C to A
3F74 FE04      CP   #04        ; already 4 fireballs active?
3F76 C38131    JP   #3181      ; jump back
```

En dan de aanpassing van de hitbox van de snapper verplaatsen:

```
0B95 C3793F      JP    #3F79      ; jump to additional code - jump to label AAAA  
en  
3F79 3E04        LD A, #04      ; A := 04 - label AAAA  
3F7B 32E965        LD (#65E9), A ; width snapper 1  
3F7E C3980B        JP    #0B98      ; jump back
```

Dat gaat goed zo.

Nu versie v1.04.01 opgeslagen.

Kijken of we het programma zo kunnen maken dat bij starten als eerste het titelscherm getoond wordt. Niet in de volgorde aanpassen. Dat is eerder geprobeerd, maar dat ging niet goed. Maar de fase wordt aangeduid met een variabele, misschien deze bij het starten anders initialiseren.

Wordt bepaald door de GameMode2: die wordt hier gezet:

```
01E2 3E01        LD    A,#01      ; A := 1  
01E4 32827D        LD    (REG_FLIPSCREEN),A ; store into flip screen setting  
01E7 320560        LD    (GameMode1),A ; store into game mode 1  
01EA 322762        LD    (#6227),A ; initialize screen to 1 (girders)  
01ED AF          XOR   A          ; A := 0  
01EE 320A60        LD    (GameMode2),A ; store into game mode 2
```

En hier is te zien dat introscherm overeenkomt met GameMode2 = 06:

```
0746 7E          LD    A,(HL)    ; else load A with game mode2  
0747 EF          RST   #28      ; jump based on A  
  
0748 79 07          0      ; #0779      ; clear screen, set color palettes, draw attract mode text and high score table,  
                           ; [continued] increase game mode2, clear sprites, ; draw "1UP" on screen , draws number of coins needed for play  
074A 63 07          1      ; #0763      ;  
074C 3C 12          2      ; #123C      ; set initial mario sprite position and draw remaining lives and level  
074E 77 19          3      ; #1977      ; set artificial input for demo play [change to #197A to enable playing in demo part 1/2]  
0750 7C 12          4      ; #127C      ; handle mario dying animations  
0752 C3 07          5      ; #07C3      ; clears the screen and sprites and increase game mode2  
0754 CB 07          6      ; #07CB      ; handle intro splash screen ?  
0756 4B 08          7      ; #084B      ; counts down a timer then resets game mode2 to 0
```

Eventueel aanpassen:

```
01ED C3813F      JP    #3F81      ; jump to additional code - jump to label AAAA  
01F0 00          NOP           ;  
en  
3F81 3E06        LD    A,#06      ; A := 6 - label AAAA  
3F83 320A60        LD    (GameMode2),A ; store into game mode 2  
3F86 C3F101        JP    #01F1      ; jump back
```

Ja dat werkt goed. Leuk !

Nu versie v1.04.02 opgeslagen.

Eventjes de velddefinities goedzetten en startveld.

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

Nu versie v1.04.03 opgeslagen.

Deze versie is de final versie v1.05.