

Donkey Kong Wizardry

Nieuw idee voor een nieuwe DK romhack. Drie nieuwe mechanismen:

1. Geen hamers maar ster die jumpman tijdelijke onoverwinnelijk maakt.
2. Spreuk 1: via P1 button: daarmee wordt jumpman tijdelijk onsterfelijk.
3. Spreuk 2: via P2 button: daarmee zet jumpman alles stil en kan eventueel ontsnappen.

En dan nieuwe en aangepaste velden maken die lastig zijn, zodat je soms de spreuken wel nodig hebt. Eventueel combineren met de ideeën voor DK randomizer?

Ander idee voor Barrels: als blue barrel dan checken aantal fireballs active. Als allemaal dan normale blue barrel. Anders een wild red barrel maken. Deze als wild laten gaan en bij de bounce random bepalen of de barrel daar kapot springt en er een fireball uitkomt.

Eerste idee voor Rivets: wanneer over verwijderde rivet springen dan wordt die teruggeplaatst. Kijken of dat een interessant mechanisme is en hoe je dan moet lopen om alle rivets te verwijderen.

Daarnaast lastiger maken door extra fireballs te introduceren op hogere levels?

Daarnaast leent het thema zich voor interessante sprites. Bijvoorbeeld magische items verzamelen en jumpman in blauwe mantel met punthoed (als dat kan). Ook nadenken over de wijze waarop de drie nieuwe mechanismen kenbaar gemaakt worden (outline sprites, twinkelende sterretjes, andere kleurenpalet).

Basis spel aanpassingen

Eerst de aanpassingen doorvoeren dat de climbing sequence verwijderd wordt en dat het spel start met het Donkey Kong logo.

Intro scherm, waarin Kong ladder beklmt, 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 code.

Ja dat gaat goed.

Dan starten met het titel scherm. 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 C3303F JP #3F30 ; jump to additional code – jump to label AAAA
01F0 00 NOP

```

En (in het stukje van de developers message die vrijgegeven is):

```

3F30 3E06 LD A,#06 ; A := 6 – label AAAA
3F32 320A60 LD (GameMode2),A ; store into game mode 2
3F35 C3F101 JP #01F1 ; jump back

```

Ja dat werkt goed.

Nu versie v0.01 opgeslagen.

Misschien ook alvast de titel WIZARDRY opnemen in intro scherm en de High Score. Daarmee de standaard aanpassingen al uitgevoerd. Ook eventjes het logo verwijderen dat in de code verstopt zit.
Dan heb je die geheugenruimte ook al vrij.
HIGH SCORE aanpassen in WIZARDRY

Op #36B2 staat de HIGH SCORE tekst:

#36B2: 80 76 18 19 17 18 10 23 13 1F 22 15 3F

Veranderen naar:

#36B2: 80 76 10 27 19 2A 11 22 14 22 29 10 3F

Is goed.

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:

© 1981-2021 NINTENDO

```
5C 77 49 4A 10 01 09 08 01 2C 02 00 02 01 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:

WIZARDRY

```
6E 76 27 19 2A 11 22 14 22 29 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
```

Gaat allemaal goed.

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 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: C3233F JP #3F23 ; jump to additional code to display version
3F23: 110103 LD DE,#0301 ; load task data for text "V1.00"
3F26: CD9F30 CALL #309F ; insert task to draw text
3F29: C31F08 JP #081F ; jump back

```

Stukje code van logo vrijgemaakt: #3FC8 t/m #3FFF.

Dit is een goede basis. Ook eventueel voor toekomstige romhacks.

Nu versie v0.02 opgeslagen.

Spreuken

Eerst het mechanisme voor de twee spreuken maken.

Idee is dat je altijd beide spreuken hebt en dat je die in een board slechts één keer kunt inzetten. Dit weergeven aan begin van board op scherm (SPELLS: <icon1> <icon2>). Waarbij icoontje verandert wanneer de spreuk gebruikt wordt.

Additionele code bij het checken of jumpman af gaat (de call op 19B3 naar de check collisions with hostile sprites).

Nu een call naar een stukje code waarin twee calls gedaan worden (voor elk mechanisme apart).

Dan elk mechanisme heeft een teller. Als de teller op 0 dan nog niet gebruikt.

Als de teller op 0 staat dan checken of de P1/P2 button ingedrukt wordt. Zo ja, dan teller verhogen, spreuk icoontje aanpassen en verder zonder check collisions, zo nee dan verder met check collisions. Als teller nog niet op maximum (vooralsnog eventjes 255), dan teller verhogen, animatie op basis van bit check en verder zonder check collisions en anders verder met check collisions.

Welke geheugenplekken te gebruiken voor de twee spell counters? Als de Kong climbing sequence verwijderd wordt, zijn #6385 (fasen binnen de climbing sequence), #638D (Kong bouncing counter) en #638E (Kong climbing counter) vrij.

Spell counter 1 (onsterfelijk door P1): #638D

Spell counter 2 (stilzetten door P2): #638E

Eerst eventjes testen of je tijdens het spel de P1 button kunt uitlezen. De code die dat doet is:

```
08F8 CDD508 CALL #08D5 ; draws press player buttons and loads A with IN2, masked by possible player numbers  
08FB FE04 CP #04 ; is the player 1 button pressed ?  
08FD CA0609 JP Z,#0906 ; yes, skip ahead  
  
0900 FE08 CP #08 ; is the player 2 button pressed ?  
0902 CA1909 JP Z,#0919 ; yes, skip ahead
```

Dus de call naar de collision detection aanpassen naar een call naar additionele code. Daar eventjes de test doen en zodra button gedrukt een muziekje doen. En daarna alsnog doorspringen naar de collision detection.

Aanpassen:

```
19B3 CD7B0A CALL #0A7B ; call additional code – call label AAAA
```

en:

```
0A7B CDD508 CALL #08D5 ; draws press player buttons, load in A – label AAAA  
0A7E FE04 CP #04 ; is the player 1 button pressed ?  
0A80 2005 JR NZ,#0A87 ; no, jump ahead – jump to label BBBB  
;  
0A82 3E03 LD A,#03 ; load sound duration of 3  
0A84 328360 LD (#6083),A ; play sound for bouncer  
;  
0A87 CD0828 CALL #2808 ; call collision detection – label BBBB  
0A8A C9 RET ; return
```

Dat gaat niet goed. De code op 08D5 doet veel meer dan alleen de input ophalen. In dat stuk code wordt de input In2 helemaal aan het einde opgehaald. Daar naartoe aanroepen.

```
0A7B CDF308 CALL #08F3 ; read input and load into A – label AAAA
```

Gaat nog niet goed. Komt omdat in dat stukje een CP met B gedaan wordt, maar B is nog niet gevuld. Dus onbetrouwbaar resultaat. De CP met B is nodig. Daarom alleen het commando voor het in A laden van IN2 opnemen.

```
0A7B 3A007D LD A,(IN2) ; load A with IN2 – label AAAA
```

Nu werkt het wel. Geluid van de spring wordt telkens gegeven als op de P1 button gedrukt wordt.

Ook meteen even testen voor de P2 button. Dat is dan een CP met #08 op 0A7E.

Nu versie v0.03, opgeslagen.

Nu zo maken dat op #0A7B twee calls komen: 1) checken P1 ingedrukt en afhankelijk daarvan wel/niet de collision detection uitvoeren en 2) checken P2 ingedrukt.

```

0A7B CD820A      CALL  #0A82      ; check P1 button pressed – call label AAAA
0A7E CDBBBB      CALL  #BBBB      ; check P2 button pressed – call label BBBB
0A81 C9          RET           ; return
;
0A82                         ; code for button 1 comes here – label AAAA

```

Vooralsnog de CALL naar BBBB nops laten zijn.

```

0A82 3A8D63      LD   A,(#638D)    ; load a with spellcounter1
0A85 FE00          CP  #00          ; A == 0?
0A87 2009          JR  NZ,0A92     ; no, spell is already activated – jump to label CCCC
;
0A89 47          LD   B,A          ; save A to B
0A8A 3A007D      LD   A,(IN2)      ; load A with IN2
0A8D FE04          CP  #04          ; is P1 button pressed?
0A8F 200D          JR  NZ,0A9E     ; no, do not activate spell – jump to label DDDD
;
0A91 78          LD   A,B          ; restore A from B
;
0A92 FEFF          CP  #FF          ; A == 255? – label CCCC
0A94 2808          JR  Z,0A9E     ; yes, spell is already finished – jump to label DDDD
;
; spell is active
;
0A96 3C          INC  A           ; increment A
0A97 328D63      LD   (#638D),A  ; store in spellcounter1
0A9A CDEEEE      CALL #EEEE      ; do visual changes for spell – call to label EEEE
0A9D C9          RET           ; return without checking collisions
;
; spell is not active
;
0A9E CD0828      CALL #2808     ; perform collision detection – label DDDD
0AA1 C9          RET           ; return

```

Visuele effecten nog eventjes weglaten. Dus op #0A9A vooralsnog 3 NOP's.

Lijkt helemaal goed te gaan. Na indrukken P1 een tijdje onsterfelijk. Daarna ga je toch weer af.

Alleen geen visuele weergave van hoe lang de onsterfelijkheid precies duurt. Dat gaan aanpassen. Wat voor visuele weergave doen? Eigenlijk een soort onzichtbaarheid. Met een bepaalde frequentie jumpman laten knipperen? Dat is lastig omdat je dan een lege sprite moet kiezen, maar daarna weer de oude terugzetten, terwijl dat allerlei sprites (stilstaan, lopen, springen, klimmen) kunnen zijn. Dan maar de kleur aanpassen.

```

0A9A CDA20A      CALL  0AA2      ; do visual changes for spell – call to label EEEE

```

en

```
0AA2 CB5F      BIT   3,A          ; test bit 3 of spellcounter – label EEEE
0AA4 3E02      LD    A, #02       ; load A with jumpman original sprite color
0AA6 2002      JR    NZ,FFFF     ; if bit set then skip ahead – jump to label FFFF
;
0AA8 3E03      LD    A,#03       ; load A with jumpman alternative color – label FFFF
;
0AAA 324E69    LD    (#694E),A  ; set color of jumpman sprite
;
0AAD C9        RET
```

Ja dat gaat goed. Alleen nog even een interessante kleur wijziging vinden.
08 is goed, 09 is goed, 0C is goed, 0E is goed.

En effect nog beter wanneer de snelheid van knipper iets lager:

Dus dan:

```
0AA2 CB67      BIT   4,A          ; test bit 4 of spellcounter – label EEEE
0AA8 3E0E      LD    A,#0E       ; load A with jumpman alternative color – label FFFF
```

Nu versie v0.04 opgeslagen.

Zou mooi zijn als bij activeren van de spell een geluid afgespeeld wordt. Dat toevoegen aan de visualisatiecode.

De code om een coin sound af te spelen:

```
0198 3E03      LD    A,#03       ; load sound duration
019A 328360    LD    (#6083),A  ; plays the coin insert sound
```

Nieuwe visualisatie en sound code:

```
0AA2 47        LD    B,A          ; save A to B – label EEEE
0AA3 FE01      CP    #01          ; A == 01 ?
0AA5 2006      JR    NZ,0AAD    ; no, skip ahead – jump to label GGGG
;
0AA7 3E03      LD    A,#03       ; load sound duration
0AA9 328360    LD    (#6083),A  ; plays the coin insert sound
0AAC 78        LD    B,A          ; restore A from B
;
0AAD CB67      BIT   4,A          ; test bit 4 of spellcounter – label GGGG
0AAF 3E02      LD    A, #02       ; load A with jumpman original sprite color
0AB1 2002      JR    NZ,0AB5    ; if bit set then skip ahead – jump to label FFFF
;
0AB3 3E0E      LD    A,#0E       ; load A with jumpman alternative color
```

```

;
0AB5 324E69      LD    (#694E),A      ; set color of jumpman sprite – label FFFF
;
0AB8 C9          RET

```

Dat werkt goed. Geeft extra effect aan de spell.

Nu versie v0.05 opgeslagen.

Toch nog even proberen met sound gedurende de gehele periode. Dus dan niet alleen als de spellcounter gelijk is aan 1. Maar bij test op bit 4 en dan in één van beide gevallen een AND doen zodat alleen de lagere bits overblijven en als die 1 zijn dan een geluid geven.

Nieuwe visualisatie en sound code:

```

;
0AA2 47          LD    B,A          ; save A to B
0AA3 CB67        BIT   4,A          ; test bit 4 of spellcounter – label EEEE
0AA5 3E02        LD    A, #02        ; load A with jumpman original color
0AA7 280C        JR    Z,0AB5       ; if bit not set then skip ahead – jump to label FFFF
;
0AA9 78          LD    A,B          ; restore A from B
0AAA E60F        AND   #0F          ; mask lower bits ?
0AAC 2005        JR    NZ,0AB3      ; if not zero skip next steps – jump to label GGGG
;
0AAE 3E03        LD    A,#03        ; load sound duration
0AB0 328360      LD    (#6083),A   ; plays the coin insert sound
;
0AB3 3E0E        LD    A,#0E        ; load A with jumpman alt color - label GGGG
;
0AB5 324E69      LD    (#694E),A   ; set color of jumpman sprite – label FFFF
;
0AB8 C9          RET

```

Dat gaat wel goed. Maar vrij langzaam. Wellicht leuker om het sneller te laten verlopen. Dus dan bij elke kleurverandering doen.

```

;
0AA2 47          LD    B,A          ; save A to B
0AA3 CB67        BIT   4,A          ; test bit 4 of spellcounter – label GGGG
0AA5 3E02        LD    A, #02        ; load A with jumpman original sprite color
0AA7 2002        JR    NZ,0AB5      ; if bit set then skip ahead – jump to label FFFF
;
0AA9 3E0E        LD    A,#0E        ; load A with jumpman alternative color
;
0AAB 324E69      LD    (#694E),A   ; set color of jumpman sprite – label FFFF
;
0AAE 78          LD    A,B          ; restore A from B

```

```

0AAF E60F      AND #0F          ; mask lower bits
0AB1 FE01      CP  #01          ; equal to #01 ?
0AB3 C0        RET NZ          ; no, return
;
0AB4 3E03      LD   A,#03        ; load sound duration
0AB6 328360    LD   (#6083),A  ; plays the coin insert sound
;
0AB9 C9        RET

```

Dat gaat nu goed. Veel beter effect en ook beter te voorspellen wanneer de spell weer afgelopen is. Dus zou houden.

Nu versie v0.06 opgeslagen.

Dan nu kijken voor de tweede spreek. De spell die geactiveerd wordt door P2 en die alles stil zet. In principe na indrukken van P2 alleen de teller laten lopen en verder niets doen.

We hadden al een stukje voorbereid voor een call naar afhandeling van P2:

```

0A7B CD820A      CALL #0A82      ; check P1 button pressed – call label AAAA
0A7E CDBBBB      CALL #BBBB      ; check P2 button pressed – call label BBBB
0A81 C9          RET           ; return

```

Aanpassen:

```
0A7E CDBAOA      CALL #0ABA      ; check P2 button pressed – call label BBBB
```

en:

```

0ABA 3A8E63      LD   A,(#638E)  ; load a with spellcounter2
0ABD FE00        CP  #00          ; A == 0?
0ABF 2008        JR   NZ,0AC9   ; no, spell is already activated – jump to label CCCC
;
0AC1 47          LD   B,A        ; save A to B
0AC2 3A007D      LD   A,(IN2)    ; load A with IN2
0AC5 FE08        CP  #08          ; is P2 button pressed?
0AC7 C0          RET NZ        ; no, do not activate spell – return
;
0AC8 78          LD   A,B        ; restore A from B
;
0AC9 FEFF        CP  #FF          ; A == 255? – label CCCC
0ACB C8          RET Z         ; yes, spell is already finished - return
;
; spell is active
;
0ACC 3C          INC  A          ; increment A
0ACD 328E63      LD   (#638E),A  ; store in spellcounter2
0AD0 CDEEEE      CALL #EEEE      ; do visual changes for spell – call to label EEEE

```

```
0AD3 C9          RET          ; return
```

Visuele effecten nog eventjes weglaten. Dus op #0A9A vooralsnog 3 NOP's.

Ja dat werkt goed. Je kunt eenmaal de P2 knop indrukken en dan wordt de teller langzaam opgehoogd totdat het maximum van 255 bereikt wordt.

Nu versie v0.07 opgeslagen.

Eens kijken of we iets eenvoudigs tijdelijk kunnen laten stoppen. Iets vanuit de main dat eenvoudig tegengehouden kan worden?

Bijvoorbeeld:

```
198C CDED30      CALL #30ED      ; update fires if needed
```

Wat gebeurt er als we deze niet uitvoeren wanneer de timer oploopt? Dan geen fireballs geupdate? Zal wel genuanceerdeerder liggen en lastiger worden. Maar eerst eens kijken gewoon.

```
198C CD383F      CALL #3F38      ; update fires if needed via P2 spell control
```

en

```
3F38 3A8E63      LD  A,(#638E)    ; load a with spellcounter2
3F3B FE00        CP  #00          ; A == 00 ?
3F3D 2805        JR  Z,#3F44     ; yes, spell is not active – jump to label AAAA
;
3F3F FEFF        CP  #FF          ; A == 255 ?
3F41 2801        JR  Z,#3F44     ; yes, spell is already used – jump to label AAAA
;
3F43 C9          RET           ; spell is active – return without update fires
;
3F44 CDED30      CALL #30ED      ; update fires if needed – label AAAA
3F47 C9          RET           ; return
```

Ja dat werkt. De fireball wordt gepauzeerd en gaat weer verder zodra de spellcounter het maximum bereikt heeft. Het is nog niet zo eenvoudig om over een stilstaande fireball heen te springen...

Nu versie v0.08 opgeslagen.

Kijken of we de barrels ook kunnen stoppen. Vanuit de main zijn er drie aanroepen die barrel afhandeling doen.

```
1983 CD721F      CALL #1F72      ; roll barrels
1986 CD8F2C      CALL #2C8F      ; roll barrels ?
1989 CD032C      CALL #2C03      ; do barrel deployment ?
```

Kijken of we ze kunnen afvangen met één stuk code.

```

1983 CD483F      CALL  #3F48      ; roll barrels via P2 spell control
1986 000000      NOP,NOP,NOP
1989 000000      NOP,NOP,NOP

```

en

```

3F48 3A8E63      LD   A,(#638E)    ; load a with spellcounter2
3F4B FE00        CP   #00          ; A == 00 ?
3F4D 2805        JR   Z,#3F54     ; yes, spell is not active – jump to label AAAA
;
3F4F FEFF        CP   #FF          ; A == 255 ?
3F51 2801        JR   Z,#3F54     ; yes, spell is already used – jump to label AAAA
;
3F53 C9          RET
;
3F54 CD721F      CALL #1F72      ; roll barrels – label AAAA
3F57 CD8F2C      CALL #2C8F      ; roll barrels
3F5A CD032C      CALL #2C03      ; do barrel deployment
3F5D C9          RET
;
```

Ja dat werkt. De barrels blijven stilstaan. Je kunt er dan overheen springen of een moeilijke combinatie ontwijken. Ook worden er geen nieuwe barrels gedeployed. Wild barrels blijven in de lucht hangen. Als ze vlak boven een girder hangen, dan kan je er overheen springen.

Nu versie v0.09 opgeslagen.

Er gaan meer van die testblokken komen. En die zijn grotendeels steeds hetzelfde.

Zou kunnen dat er één call gedaan wordt naar deze code en dan checken en dan aan het einde alle CALLS die afhankelijk van P2 zijn na elkaar doen. Maar dan veranderd deels de volgorde waarin de calls aangeroepen worden. Misschien kan het ook anders gedaan worden door wel afzonderlijke calls te gebruiken, maar het gemeenschappelijke teststuk te delen.

De testcode zo maken dat A=1 als P2 spell active and A=0 als P2 spell not active.

En dan vanuit de code een aanroep naar deze testcode en daarna de A conditioneel testen.

Testcode:

```

3F38 3A8E63      LD   A,(#638E)    ; load a with spellcounter2
3F3B FE00        CP   #00          ; A == 00 ?
3F3D 2807        JR   Z,#3F46     ; yes, spell is not active – jump to label AAAA
;
3F3F FEFF        CP   #FF          ; A == 255 ?
3F41 2803        JR   Z,#3F46     ; yes, spell is already used – jump to label AAAA
;
3F43 3E01        LD   A,#01      ; spell is active - load A with #01
3F45 C9          RET
;
```

;
3F46 3E00 LD A,#00 ; spell is not active – load A with #00 – label AAAA
3F48 C9 RET ; return

en

198C CD493F CALL #3F49 ; update fires if needed via P2 spell control

en

3F49 CD383F CALL #3F38 ; test P2 spell active
3F4C C0 RET NZ ; return if spell is active – do not update fires
3F4D CDED30 CALL #30ED ; update fires
3F50 C9 RET ; return

en

1983 CD513F CALL #3F51 ; roll barrels via P2 spell control

1986 000000 NOP,NOP,NOP

1989 000000 NOP,NOP,NOP

en

3F51 CD383F CALL #3F38 ; test P2 spell active
3F54 C0 RET NZ ; return if spell is active – do not roll barrels
3F55 CD721F CALL #1F72 ; roll barrels
3F58 CD8F2C CALL #2C8F ; roll barrels
3F5B CD032C CALL #2C03 ; do barrel deployment
3F5E C9 RET ; return

Ja dat werkt.

Nu versie v0.10 opgeslagen.

Dan nu de springs stoppen. Vanuit de main gebeurt dat hier:

198F CD042E CALL #2E04 ; update bouncers if on elevators

Aanpassen:

198F CD5F3F CALL #3F5F ; update bouncers if on elevators via P2 spell control

en

3F5F CD383F CALL #3F38 ; test P2 spell active
3F62 C0 RET NZ ; return if spell is active – do not update fires
3F63 CD042E CALL #2E04 ; update bouncers if on elevators
3F66 C9 RET ; return

Ja dat gaat goed. Ook de springs stoppen met bouncen.

Nu versie v0.11 opgeslagen.

En dan de elevators laten stoppen. Vanuit de main gebeurt dat hier:

```
19A7 CDFA26      CALL #26FA      ; do stuff for elevators
```

Aanpassen:

```
19A7 CD673F      CALL #3F67      ; do stuff for elevators
```

en

```
3F67 CD383F      CALL #3F38      ; test P2 spell active
3F6A C0          RET  NZ         ; return if spell is active – do not update fires
3F6B CDFA26      CALL #26FA      ; do stuff for elevators
3F6E C9          RET
```

Ja dat gaat goed. Zie nu wel dat de springs stoppen, maar als ze weer verder gaan is er een spring reset. Net alsof er een extra spring komt. Misschien daar nog een keer naar kijken.

Nu versie v0.12 opgeslagen.

Dan nu de conveyors stoppen. Daar zijn twee stukken voor:

```
199B CD0722      CALL #2207      ; do stuff for conveyors
19AA CDF225      CALL #25F2      ; handle conveyor directions
```

Aanpassen:

```
199B CD6F3F      CALL #3F6F      ; do stuff for conveyors
19AA CD773F      CALL #3F77      ; handle conveyor directions
```

en

```
3F6F CD383F      CALL #3F38      ; test P2 spell active
3F72 C0          RET  NZ         ; return if spell is active – do not update fires
3F73 CD0722      CALL #2207      ; do stuff for conveyors
3F76 C9          RET
;
3F77 CD383F      CALL #3F38      ; test P2 spell active
3F7A C0          RET  NZ         ; return if spell is active – do not update fires
3F7B CDF225      CALL #25F2      ; handle conveyor directions
3F7E C9          RET
```

Dit testen door te starten met level 3 en screen 2:

```
095E 01 6A 3A 01 00 00 00 ; #3A65 is start of table data for screens/levels
```

Aanpassen naar:

```
095E 03 6A 3A 01 00 00 00 ; #3A65 is start of table data for screens/levels
```

En dan in de tabel op #3A6A het eerste board 01 (barrels) vervangen door 02 (pies).

Gaat redelijk goed. Meestal stoppen de pies en DK met bewegen en kan jumpman gewoon over de lopende band lopen zonder sneller en/of langzamer te gaan. Dus voor gevoel is de lopende band ook uit. Maar soms gaan de pies juist sneller bewegen i.p.v. te stoppen. En in dat geval beweegt DK ook veel sneller en loopt uit het scherm en verschijnt aan de andere kant. De fireballs stoppen op dat moment wel. Geen idee hoe dat komt.

Het lijkt erop dat de eerste keer wel gewoon gestopt wordt, maar de volgende keren niet. Wellicht daar een verklaring in te vinden?

Het lijkt erop alsof er nog een entry is voor pie factory:

```
1992 CDEA24 CALL #24EA ; do stuff for pie factory
```

Aanpassen

```
1992 CD7F3F CALL #3F7F ; do stuff for pie factory
```

en

```
3F7F CD383F CALL #3F38 ; test P2 spell active  
3F82 C0 RET NZ ; return if spell is active – do not update fires  
3F83 CDEA24 CALL #24EA ; do stuff for pie factory  
3F86 C9 RET
```

Dit gaat al beter. De pies stoppen altijd en jumpman's snelheid op de conveyors wordt niet aangepast. Maar DK beweegt nog wel twee keer zo snel. Waar zit dat dan?

Dat lijkt hier te zitten:

```
19B0 CDFB03 CALL #03FB ; check for kong beating chest and animate girl
```

Aanpassen:

```
19B0 CD873F CALL #3F87 ; check for kong beating chest and animate girl
```

en

```
3F87 CD383F CALL #3F38 ; test P2 spell active  
3F8A C0 RET NZ ; return if spell is active – do not update fires  
3F8B CDFB03 CALL #03FB ; check for kong beating chest and animate girl
```

```
3F8E C9          RET
```

Ja dat gaat goed.

Nu versie v0.13 opgeslagen.

Nog kijken naar de spring reset. Bouncer deployment gebeurd hier:

```
19BF CDCB2F      CALL  #2FCB      ; for non-girder levels, checks for bonus timer  
                                ; changes. if the bonus counts down, sets a possible  
                                ; new fire to be released, sets a bouncer to be  
                                ; deployed, updates the bonus timer onscreen, and  
                                ; checks for bonus time running out
```

Doet dus veel meer. Maar op zich ook wel leuk als de bonus timer ook stil komt te staan. Gebeurd denk ik ook bij barrel board omdat daar de timer verlaagd wordt bij elke barrel die gedeployed wordt.

Aanpassen:

```
19BF CD8F3F      CALL  #3F8F      ; for non-girder levels, checks for bonus timer  
                                ; changes. if the bonus counts down, sets a possible  
                                ; new fire to be released, sets a bouncer to be  
                                ; deployed, updates the bonus timer onscreen, and  
                                ; checks for bonus time running out
```

en

```
3F8F CD383F      CALL  #3F38      ; test P2 spell active  
3F92 C0          RET   NZ        ; return if spell is active – do not update fires  
3F93 CDCB2F      CALL  #2FCB      ; for non-girder levels, checks for bonus timer  
                                ; changes. if the bonus counts down, sets a possible  
                                ; new fire to be released, sets a bouncer to be  
                                ; deployed, updates the bonus timer onscreen, and  
                                ; checks for bonus time running out
```

```
3F96 C9          RET
```

Even testen met de springs. Daarom level 3 eerste board 03 (springs) te maken.

Dat gaat nu goed.

Nu versie v0.14 opgeslagen.

Nog wel een keer in springs na spell dat op een trap niet meer het einde van de trap gezien wordt. Dus dan kan jumpman doorklommen tot buiten het scherm of zo hoog dat het gezien wordt als board complete.

Ik vertrouw het hierdoor toch niet helemaal. Daarnaast is het ook zo dat het stoppen van de elevators, de springs, de barrels en de conveyors ook wel een beetje het hele mechanisme verzieken.

Te overwegen om alleen de freeze van de fireballs (en firefoxes) te behouden. En daar dan wel een visualisatie aan toe te voegen.

[Terug naar versie v0.08.](#)

Animatie door bijvoorbeeld de fireballs een alternatieve kleur te geven? Of te laten knipperen van kleur net als dat jumpman knippert van kleur bij de P1 spell.

Als de hamer actief is worden de fireballs ook een andere kleur. Je zou kunnen uitzoeken hoe dat dan gedaan wordt. Maar wat doe je dan als freeze van fireballs èn hamer gebruikt. Daarom misschien maar niet doen.

Ander punt is dat bij einde board wanneer spell P1 actief, jumpman soms in de alternatieve kleur blijft staan. Dat wellicht goedzetten door bij einde level de kleur van jumpman te forceren.

Einde level zit hier:

```
; 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
```

Aanpassen:

```
1615 CD483F      CALL #3F48      ; clear sprites and reset jumpman color
```

en

```
3F48 CDBD30      CALL #30BD      ; clear sprites
3F4B 3E02         LD A,#02        ; load A with jumpman original color
3F4D 324E69      LD (#694E),A   ; set color of jumpman sprite
3F50 C9          RET           ; return
```

Ja dat gaat goed.

[Nu versie v0.15 opgeslagen.](#)

#3165 zet de kleur van de fireball wanneer de hamer actief is. Testen verschillende kleuren.

```

3158 3E01      LD      A,#01          ; \ Set fire color to #01 (normal) if hammer is not active, and #00 (blue) if hammer is active
315A DD7708      LD      (IX+#08),A    ; |
315D 3A1762      LD      A,(#6217)     ; |
3160 FE01      CP      #01          ; |
3162 C26A31      JP      NZ,#316A     ; |
3165 3E00      LD      A,#00          ; |
3167 DD7708      LD      (IX+#08),A    ; /

```

Kleur #0C ziet er wel uit als bevroren vuur...Kleur #0D is ook wel aardig.

Maar dat gaat hem niet worden. Daarom zo laten. Alleen freezeen en geen visualisatie.

Nu de spel status weergeven bovenin het scherm. Eerst even bedenken hoe we dit gaan weergeven en dan maken. Misschien tekst SP: X / X. En dan X is icoontje outline jumpman/fireball en wanneer gebruikt een minteken.

Stuk code invoegen als extra stap in de main. Dat kan op de drie NOP's op #19C2. Als er later behoefte aan nog meer stappen in de main dan kan dat later aangepast worden.

Stuk code om SP te tekenen (als eerste test, onder de L=):

```
19C2 CD513F      CALL   #3F51          ; display spell info
```

en

```

3F51 212475      LD      HL,#7524      ; load HL with screen location for spell info
3F54 3623        LD      (HL),#23      ; draw "S"
3F56 210475      LD      HL,#7504      ; next location
3F59 3620        LD      (HL),#20      ; draw "P"
3F5B C9          RET               ; return

```

Dit ziet er niet mooi uit. Omdat op die lijn alleen de kleuren van de girders (2 kleuren bordeau rood en lichtblauw) te gebruiken zijn. Daarom boven de L= zetten. Daar zijn de kleuren wit, geel en blauw mogelijk. Standaard kleur is wit, maar door definiëren letters in andere kleur kan je ook één van de andere kleuren gebruiken.

```

3F51 212275      LD      HL,#7522      ; load HL with screen location for spell info
3F54 3623        LD      (HL),#23      ; draw "S"
3F56 210275      LD      HL,#7502      ; next location
3F59 3620        LD      (HL),#20      ; draw "P"
3F56 21E274      LD      HL,#74E2      ; next location
3F59 3634        LD      (HL),#34      ; draw "="
3F5B C9          RET               ; return

```

Gaat goed. Maar bij einde level gaat er een wipe overheen waardoor de S verwijderd wordt. Dus beperken tot alleen de S:

3F51	210275	LD	HL,#7502	; load HL with screen location for spell info
3F54	3623	LD	(HL),#23	; draw "S"
3F56	21E274	LD	HL,#74E2	; next location
3F59	3634	LD	(HL),#34	; draw "="
3F5B	C9	RET		; return

En dan de afhankelijke karakters:

3F5B	21C274	LD	HL,#74C2	; next location
3F5E	3A8D63	LD	A,(#638D)	; load spellcounter1
3F61	FE00	CP	00	; A == 00?
3F63	3E35	LD	A,#35	; set character spell1 used
3F65	2002	JR	NZ,#3F67	; no, skip next step
3F67	3E01	LD	A,#01	; set character spell1 not used
3F69	77	LD	(HL),A	; draw character spell1
;				
3F6A	21A274	LD	HL,#74A2	; next location
3F6D	3635	LD	(HL),35	; draw "-"
;				
3F6F	218274	LD	HL,#7482	; next location
3F72	3A8E63	LD	A,(#638E)	; load spellcounter2
3F75	FE00	CP	00	; A == 00?
3F77	3E35	LD	A,#35	; set character spell2 used
3F79	2002	JR	NZ,#3F7D	; no, skip next step
3F7B	3E02	LD	A,#02	; set character spell2 not used
3F7D	77	LD	(HL),A	; draw character spell2
;				
3F7E	C9	RET		; return

Nieuwe characters

3F54	36BA	LD	(HL),#BA	; draw "S"
3F59	36BB	LD	(HL),#BB	; draw "="
3F63	3EBD	LD	A,#BD	; set character spell1 used
3F67	3EBE	LD	A,#BE	; set character spell1 not used
3F6D	36BC	LD	(HL),BC	; draw "/"
3F77	3EBD	LD	A,#BD	; set character spell2 used
3F7B	3EBF	LD	A,#BF	; set character spell2 not used

Sprites aangepast. Spell 1 hoofd jumpman en spell 2 fireball.

[Nu versie v0.16 opgeslagen.](#)

Gaat allemaal goed. Alleen aan het einde van het level blijft het spell dashboard staan (met de laatste status) totdat bij het volgende level het weer gevuld wordt. Daarom dan misschien helemaal wissen of dan -/- weergeven.

Wanneer een board klaar is, dan wordt de volgende code uitgevoerd:

```

; 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

```

Tegelijkertijd met het wissen van de sprites moet ook het spell dashboard gewist worden. Dus de aanroep in 1615 veranderen in een andere call en daar als eerste de call om de sprites te wissen en dan de additionele code om het spell dashboard te wissen.

Wissen zou kunnen door een tekst te plaatsen met 5 spaties. Er zijn nog een aantal tekst sloten vrij. Alternatief is om zelf een loop te maken die de 5 spaties naar video RAM schrijft. Het eerste is het makkelijkste. Het tweede wordt o.a. in de originele code gedaan bij #05E9. Het eerste doen. Als later nog meer teksten benodigd, dan kan altijd de code in #05E9 los aangeroepen worden.

Slot 6 is nog vrij.

De call om sprites te wissen is al eerder verplaatst naar additionele code:

```
1615 CD483F CALL #3F48 ; clear sprites and reset jumpman color
```

en

```
3F48 CDBD30 CALL #30BD ; clear sprites
3F4B 3E02 LD A,#02 ; load A with jumpman original color
3F4D 324E69 LD (#694E),A ; set color of jumpman sprite
3F50 C9 RET ; return
```

Aanpassen:

```
3F48 CD873F CALL #3F87 ; call additional code – call label AAAA
```

en

```
3F87 CDBD30 CALL #30BD ; clear sprites – label AAAA
3F8A 110603 LD DE,#0306 ; load task data to draw " " – task 3, slot 6
3F8D CD9F30 CALL #309F ; insert task to draw text
3F90 C9 RET ; return
```

en

```
3657 7F 3F ; position of text string
```

en

```
3F7F 02 75 10 10 10 10 10 3F ; pos pos space space space space space end
```

Ja dat gaat goed wanneer einde level. Maar het dashboard blijft ook staan wanneer jumpman afgaat. Dan worden de sprites blijkbaar ook gewist. Die call ook aanpassen nog.

Dat wordt hier gedaan:

```
12A3 CDBD30 CALL #30BD ; clear sprites ?
12A6 3E03 LD A,#03 ; load A with duration of sound
12A8 328860 LD (#6088),A ; play death sound
12AB C9 RET ; return
```

Dus de call op #12A3 aanpassen naar een call naar 3F87.

Aanpassen:

```
12A3 CD873F CALL #3F87 ; call additional code – call label AAAA
```

Ja dat gaat ook goed.

Nu versie v0.17 opgeslagen.

Dan nu het mechanisme van de rivets aanpassen. Als je over een rivet gat heen springt dat er weer een rivet gevuld wordt (wellicht met een andere kleur?). En dan checken: als eerste keer eroverheen dan alleen punten uitdelen. Daarna geen punten. Dus ook gaan bijhouden of het de eerste keer is of niet.

Eerst de code analyseren die het verwijderen van de rivets checkt en uitvoert.

De code voor het checken van de rivets start bij #1A33. De code van de daadwerkelijke check is:

```
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
```

Dus ergens in de rivets array wordt dat bijgehouden. Rivets array start bij #6292.

Debugging geeft aan dat acht geheugenplaatsen vanaf #6292 gevuld zijn met #01 en wanneer een rivet gewist wordt, dan wordt deze waarde in #00 veranderd. Hierboven wordt met AND A gecheckt of de waarde #00 is. Zo ja, dan is deze al een keer gewist.

Kijken hoe dan bijgehouden wordt hoeveel rivets er gewist zijn en wanneer ze allemaal gewist zijn.

De volgende code laat dat zien:

```
; 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
```

Dus #1A89 verlaagt het aantal rivets (staat initieel op 8) wat is opgeslagen in #6290. Geen idee waar de tussenliggende #6291 voor gebruikt wordt.

De code van #1A84 t/m #1AC0 zorgt voor de afhandeling wanneer rivet verwijderd dient te worden. Doet drie dingen: 1) de teller verlagen, 2) de rivet in video ram wissen en 3) bonus uitdelen en bonus geluid afspelen. Het grootste gedeelte is het wissen van het video ram (door er spaties in te schrijven) en dat wil je hergebruiken door rivet weer te tekenen afhankelijk van wissen dan wel terugzetten.

Maar de code is vrij lastig om aan te passen. Daarom wellicht toch dat stukje kopiëren.

Daarom op 1A81 springen naar additionele code die de test doet al/dan niet eerder gewist en dan springen naar code wanneer nog niet gewist en verder gaan met specifieke code wanneer al wel gewist: teller verhogen, rivet markeren als niet gewist (02 invullen) en rivet opnieuw tekenen.

```
1A81 C3C83F JP #3FC8 ; jump to additional code – jump to label AAAA
```

en

```
3FC8 7E LD A,(HL) ; get the status of the rivet mario is crossing
3FC9 A7 AND A ; has this rivet already been traversed?
3FCA C2841A JP NZ, 1A84 ; yes, jump to code handle rivets erase
;
3FCD 3602 LD (HL),#02 ; set this rivet as filled again
3FCF 219062 LD HL,#6290 ; load HL with address of number of rivets remaining
3FD2 34 INC (HL) ; increase number of rivets
;
3FD3 78 LD A,B ; A := B
3FD4 010500 LD BC,#0005 ; load BC with offset of 5
3FD7 1F RRA ; rotate right. carry? (is this rivet on right side?)
;
3FD8 212B01 LD HL,#012B ; load HL with alternate master offset for rivets
3FDB 3003 JR NC,#3EF0 ; no, skip ahead
3FDD 21CB02 LD HL,#02CB ; else load HL with master offset for rivets
;
3FE0 A7 AND A ; A == 0 ?
3FE1 2804 JR Z,#3FE7 ; yes, skip next 3 steps
;
3FE3 09 ADD HL,BC ; add offset to HL
3FE4 3D DEC A ; decrease A. zero?
```

```

3FE5 20FC        JR    NZ,#3FE3      ; no, loop again
;
3FE7 010074      LD    BC,#7400      ; start of video RAM is #7400
3FEA 09          ADD   HL,BC      ; add offset computed
3FEB 36B7        LD    (HL),#B7      ; insert the rivet
3FED 2D          DEC   L           ; next video memory
3FEE 36B8        LD    (HL),#B8      ; insert the top of the rivet
;
3FF0 C9          RET               ; return

```

Om dit te testen beginnen met rivets.

Gaat bijna goed. Alleen nieuwe rivet wordt aan de verkeerde kant neergezet. Omdraaien:

```

3FD8 21CB02      LD    HL,#02CB      ; load HL with alternate master offset for rivets
3FDB 3003        JR    NC,#3EF0      ; no, skip ahead
3FDD 212B01      LD    HL,#012B      ; else load HL with master offset for rivets

```

Gaat nu goed. Alleen kijken of we anders gekleurde rivet kunnen maken. Aangepaste rivet zit nu in characters 97 en 98. Daarom aanpassen:

```

3FEB 36B4        LD    (HL),#B4      ; insert the rivet
3FED 2D          DEC   L           ; next video memory
3FEE 36B8        LD    (HL),#B8      ; insert the top of the rivet

```

Dat gaat niet goed. Tekent wel de omgekeerde rivet maar daar valt jumpman doorheen als hij eroverheen loopt. Kijken voor andere character positie. Aangepast hierboven.

Nu versie v0.18 opgeslagen.

Alleen nog aanpassen dat alleen de eerste keer een rivet verwijderen punten oplevert.

Uitgebreid functioneel testen met L=05 rivets.

Aanpassen naar:

095E 05 73 3A 01 00 00 00 ; #3A65 is start of table data for screens/levels

En dan in de tabel op #3A73 het eerste board 01 (barrels) vervangen door 04 (rivets).

Alleen eerste keer punten voor verwijderen rivet. Bonus uitkeren gebeurt hier:

```

1AAA 3E01      LD     A,#01          ; A := 1
1AAC 324063    LD     (#6340),A    ; store into #6340 - usually 0, changes when mario picks up bonus item.
1AAF 324263    LD     (#6342),A    ; store into scoring indicator
1AB2 322562    LD     (#6225),A    ; store into bonus sound indicator
1AB5 3A1662    LD     A,(#6216)    ; load A with jump indicator
1AB8 A7        AND    A            ; is mario jumping ?
1AB9 CC951D    CALL   Z,#1D95    ; no, play the bonus sound

1ABC C9        RET              ; else return

```

Dus bij tweede keer rivet verwijderen niet de bonus indicatoren (#6340 en #6342) vullen, maar wel het bonus geluid afspelen.

Maar hoe weet je hier of je de eerste of de tweede keer over een rivet gegaan bent? Dat weet je niet. Alleen aan het begin van de hele afhandeling die begint op #1A84. Op dat moment heeft A de waarde van de rivet (01 als eerste keer, 02 als volgende keer) en wijst HL naar de geheugenplaats van de specifieke rivet in de rivets tabel.

Dus voor de jump naar afhandelen van de rivets erase naar een stukje additionele code springen:

Aanpassen:

```
3FCA C2913F      JP     NZ, 3F91    ; yes, jump handle rivets erase – jump to label BBBB
```

en

```

3F91 FE02      CP     #02          ; is rivet cleared earlier? – label BBBB
3F93 2806      JR     Z,#3F9B    ; yes, skip next two steps
3F95 324063    LD     (#6340),A  ; store into bonus indicator
3F98 324263    LD     (#6342),A  ; store into scoring indicator
3F9B C3841A    JP     #1A84    ; yes, jump to code handle rivets erase

```

en

```
1AAC t/m 1AB1      NOP's
```

Vreemd dit werkt niet goed. Wel de bonus en geluiden, maar de eerste keer wordt de rivet niet gewist, maar de keren daarop wel. Blijkt door de CP veroorzaakt te worden. Daardoor zijn de flags anders dan in de originele aanroep waardoor de positie, waar de rivet verwijderd dient te worden, verkeerd wordt berekend. Dit is op te lossen door er een extra AND A tussen te voegen.

Aanpassen:

```

3F9B A7        AND    A
3F9C C3841A    JP     #1A84    ; yes, jump to code handle rivets erase

```

Werkt nu wel.

Nu versie v0.19 opgeslagen.

Ingewikkelder maken door fireballs aan willekeurige kant te spawnen? Nu gebaseerd op de positie van jumpman. Maar dat random maken. Wordt hier gedaan:

```
34B9 3A2762 LD A,(#6227) ; \ Return if current screen is elevators (Can this ever happen?)  
34BC fe03 CP #03 ; |  
34Be c8 RET Z ; /  
  
34BF 3A0362 LD A,(#6203) ; \ Jump if bit 7 of Mario's X-position is set (i.e., Mario is on the right half of the screen)  
34C2 cb7f BIT 7,A ; |  
34C4 C2ED34 JP NZ,#34ED ; /  
  
34C7 21C43A LD HL,#3AC4 ; Load HL with start of table data for spawning fireball on right side
```

Aanpassen:

```
34BF 3A1860 LD A,(#6018) ; \ Jump if bit 1 of RNGTimer1 is set  
34C2 CB4F BIT 1,A ; |  
34C4 C2ED34 JP NZ,#34ED
```

Dat werkt wel, maar is niet heel handig. Komt omdat de spawn ook aan de kant van jumpman kan zijn en dan heb je het probleem dat dat ook de op plek kan zijn waar jumpman precies staat en dan ben je meteen af. Dus niet doen. Terug naar versie v0.19.

Wat ook zou kunnen is om de fireballs ook naar beneden te laten gaan als jumpman hoger dan ze is. Dan verspreiden ze meer. Kan interessant zijn. Dat gebeurt hier:

```
; 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 ; /
```

Aanpassen:

336B NOP maken: dan geen return als jumpman hoger dan fireball.

Dit lijkt wel een leuke aanpassing. Mede ook doordat de fireballs nu ook meer beneden blijven waardoor daar de rivets clearen wat lastiger wordt. En dan misschien nog combineren met een extra fireball. En misschien de kans vergroten dat ze een ladder nemen. Dan krijg je meer bewegingen omhoog en naar beneden.

Fireball informatie wordt vastgelegd in een array 5 blokken van #20 bytes vanaf #6400-#6495. Die informatie wordt steeds gebruikt om de fireball sprites bij te werken: #69D0.

In de debugger blijkt dat er na de array nog ruimte niet gebruikt wordt en ook de sprites erna niet gebruikt worden.

Dan dus op verschillende punten waar de 6400 als begin gebruikt wordt de loop aanpassen van 5 naar 6. Doen op basis van zoeken naar 6400:

2891	0606	LD	B,#06	; (collisions on barrels)
28B1	0606	LD	B,#06	; (collisions on pies)
28E1	0606	LD	B,#06	; (collisions on elevators)
2902	0608	LD	B,#08	; (collisions on rivets, including hidden squares DK)
2C67	0606	LD	B,#06	; (check fires for deployment on barrels)
3144	0606	LD	B,#06	; (fireball movement and color when hammering)
34F9	0606	LD	B,#06	; (update fires to sprites)
3EAA	0606	LD	B,#06	; (check jumping fires)

Lijkt erop dat bij rivets na de 6400 twee sloten gebruikt worden voor de hidden squares naast DK. Klopt dat, dan moeten die ook een slotpositie opgeschoven worden. Ja dat klopt: staan nu in #64A0 en #64C0!. Maar daarna ook nog ruimte. Dus #64A0 wordt #64C0

De hidden squares naast DK worden hier gedefinieerd:

```
; draws black squares next to kong???
115A 217E11 LD HL,#117E ; load HL with start of data table
115D 11A764 LD DE,#64A7 ; set destination sprites
1160 011C02 LD BC,#021C ; set counters B := 2, C := #1C
1163 CD2A12 CALL #122A ; copy

1166 DD21A064 LD IX,#64A0 ; load IX with address of black square sprite start
116A DD360001 LD (IX+#00),#01 ; store 1 into #64A0 = turn on first sprite
116E DD362001 LD (IX+#20),#01 ; store 1 into #64C0 = turn on second sprite

1172 215069 LD HL,#6950 ; load HL with ???
1175 0602 LD B,#02 ; set counter to 2
1177 112000 LD DE,#0020 ; set offset to #20
117A CDD311 CALL #11D3 ; draw items ???
```

Aanpassen:

115D 11C764 LD DE,#64C7 ; set destination sprites

en

1166 DD21C064 LD IX,#64C0 ; load IX with address of black square sprite start

Werkt niet goed. Bij debuggen blijkt dat de array gevuld wordt met een aantal waarden. Dat gebeurt hier:

```

; arrive here when rivets starts from #0FCC

1131 21F03D LD HL,#3DF0 ; load HL with start of table data
1134 110764 LD DE,#6407 ; load DE with destination ?
1137 011C05 LD BC,#051C ; set counters

113A CD2A12 CALL #122A ; copy fire location data to screen?

```

B bevat daar de counter voor het kopiëren van de initialisatie data. Deze ook verhogen naar 6.

Aanpassen:

```
1137 011C06 LD BC,#061C ; set counters
```

Nog steeds geen 6 fireballs en ook de squares bij DK kloppen niet. Je gaat nu pas af bij het midden van DK?

De zesde fireball wordt nu wel geïnitialiseerd. Maar het lijkt wel alsof de initialisatie van de squares nog verschoven zit. Ja die staat nog bij #64A0 en #64C0. Hoe komt dat?

Raar bij debuggen. Staat wel degelijk 64C7, maar in copy routine staat de DE op 64A7. Hoe kan dat?

De kopjeslag wordt in twee delen gedaan blijkbaar. Dit stukje doet ook iets (staat vlak voor het eerder aangepaste stuk):

```

114E 218211 LD HL,#1182 ; load HL with start of data table
1151 11A364 LD DE,#64A3 ; load DE with destination ?
1154 011E02 LD BC,#021E ; set counters
1157 CDEC11 CALL #11EC ; copy

```

Aanpassen:

```
1151 11C364 LD DE,#64C3 ; load DE with destination ?
```

Er komt nog steeds geen zesde fireball, maar nu zijn de hidden squares wel op de goede plekken en worden ze ook goed gecheckt: je gaat af wanneer je de zijkant van DK raakt.

De zesde fireball wordt niet gedeployed. Zit er dan toch nog ergens een check of limiet op 5? Daar naar gaan zoeken.

Gevonden. Er is inderdaad een plek waar het aantal fireballs gelimiteerd wordt op 5:

```

317C 3AA163 LD A,(#63A1) ; \ Jump back and don't deploy fire if there are already 5 fires active (Can this ever happen here?)
317F FE05 CP #05 ; |
3181 CA6A31 JP Z,#316A ; /
3184 3A2762 LD A,(#6227) ; \ Jump ahead if screen is not conveyors (i.e., the screen is rivets)
3187 FE02 CP #02 ; |
3189 C29531 JP NZ,#3195 ; /
318C 3AA163 LD A,(#63A1) ; \ Return if current count of # of fires == internal difficulty, on conveyors we never have more fireballs
318F 4F LD C,A ; | on screen than the internal difficulty
3190 3A8063 LD A,(#63B0) ; |
3193 B9 CP C ; |
3194 C8 RET Z ; /
3195 3AA063 LD A,(#63A0) ; \ Jump back and don't deploy fire if fire deployment flag is not set
3198 FE01 CP #01 ; |

```

Aanpassen:

317F FE06 CP #06

Nog steeds geen zesde fireball. Wordt deze nu wel in de array bijgewerkt of is hij er wel maar is de sprite niet zichtbaar?

De zesde fireball wordt wel actief gemaakt, maar is niet goed geïntialiseerd. Staan verder geen waarden in velden 03 (actual X), 05 (actual Y), 0D (direction), 0E (effective X), 0F (effective Y), 13 (bob index), 14 (ladder timer), 15 (animation timer), 16 (reverse counter).

Uitzoeken waar dat gezet wordt en waarom niet voor de zesde fireball.

Gevonden. Alle fireball movement wordt geregeld vanuit een loop. Daarin staat ook nog een limiet van 5 fireballs.

```

; Loop start
31BE 2AC863 LD HL,(#63C8) ; \ Move on to next fireball by incrementing address of fireball data array for current fireball by #20
31C1 012000 LD BC,#0020 ; |
31C4 09 ADD HL,BC ; |
31C5 22C863 LD (#63C8),HL ; /
31C8 7E LD A,(HL) ; \ Jump if fireball is not active
31C9 A7 AND A ; |
31CA CAD031 JP Z,#31D0 ; |

31CD CD0232 CALL #3202 ; Handle all movement for this fire

31D0 3AA263 LD A,(#63A2) ; \ Increment index of current fireball being processed
31D3 3C INC A ; |
31D4 32A263 LD (#63A2),A ; /
31D7 FE05 CP #05 ; \ Loop if index is less than 5
31D9 C2BE31 JP NZ,#31BE ; /

```

Aanpassen:

31D7 FE06 CP #06 ; \ loop if index is less than 6

Dat werkt. Nu wel een zesde fireball. Daarmee eens gaan testen qua moeilijkheid. En misschien ook nog wel eens testen met nog meer fireballs.

Nu versie v0.20 opgeslagen.

Interessant is nog wel dat de zesde fireball aan het einde van het level en doodgaan van jumpman niet gewist wordt. Dat nog wel aanpassen.

Dat gebeurt hier:

```
; arrive here from #1615 when rivets cleared
; clears all sprites for firefoxes, hammers and bonus items

30BD 215069 LD HL,#6950 ; load HL with start of hammers
30C0 0602 LD B,#02 ; B := 2
30C2 CDE430 CALL #30E4 ; clear hammers ?
30C5 2E80 LD L,#80 ; L := #80
30C7 060A LD B,#0A ; B := #A
30C9 CDE430 CALL #30E4 ; clear barrels ?
30CC 2EB8 LD L,#B8 ; L := #B8
30CE 060B LD B,#0B ; B := #B
30D0 CDE430 CALL #30E4 ; clear firefoxes ?
30D3 210C6A LD HL,#6A0C ; load HL with start of bonus items
30D6 0605 LD B,#05 ; B := 5
30D8 C3E430 JP #30E4 ; clear bonus items
```

Startpositie voor het wissen is #69B8 en de loop counter wordt op B gezet. Als het goed is staan de sprites op de posities vanaf #69D0. Geen idee hoe dit dan werkt.

Eerst checken of de sprites inderdaad vanaf #69D0 staan. Ja dat klopt: van #69D0 t/m #69E7.

Memory: Zilog Z80 ':maincpu' program space memory									
Debug Options		zilog z80 ':maincpu' program space me							
69d0									
69D0	DD	CE	01	A0	1B	4E	01	C7	A7
69E0	DC	4D	01	C6	B5	4E	01	EE	00

In de wisroutine wordt een add gedaan van 04. Dus het aantal #0B gaat niet over aaneengesloten posities, maar over aantal sprites dat gewist moet worden (wissen is in dit geval de X-positie wijzigen naar 0 zodat ze buiten het zichtbare scherm geplaatst worden). En dan klopt het dus: de laatste sprite die dan gewist wordt is die op #69B8 + #0B * 04 = #69E4: de vijfde fireball. Dus als de zesde ook wissen dan moeten er #0C sprites gewist worden.

Aanpassen:

```
30CE 060C LD B,#0C ; B := #0C (clear until sixth fireball sprite)
```

Ja dat gaat goed.

Nu versie v0.21 opgeslagen.

Met zes fireballs is het vaak nog te doen zonder de spells te gebruiken. Komt omdat de druk niet naar boven gaat, maar ze veel meer verdeeld zijn. Daarom testen met zeven fireballs.

Daartoe aanpassen:

2891	0607	LD	B,#07	; (collisions on barrels)
28B1	0607	LD	B,#07	; (collisions on pies)
28E1	0607	LD	B,#07	; (collisions on elevators)
2902	0609	LD	B,#09	; (collisions on rivets, including hidden squares DK)
2C67	0607	LD	B,#07	; (check fires for deployment on barrels)
3144	0607	LD	B,#07	; (fireball movement and color when hammering)
34F9	0607	LD	B,#07	; (update fires to sprites)
3EAA	0607	LD	B,#07	; (check jumping fires)
115D	11E764	LD	DE,#64E7	; (DK squares moved: original A7 – steps #20/fire)
1166	DD21E064	LD	IX,#64E0	; (DK squares moved: original A0 – steps #20/fire)
1137	011C07	LD	BC,#071C	; (fireball initialization)
1151	11E364	LD	DE,#64E3	; (DK squares moved: original A3 – steps #20/fire)
317F	FE07	CP	#07	; (fireball deployment limit)
31D7	FE07	CP	#07	; (fireball processing limit)
30CE	060D	LD	B,#0D	; (clear fireballs: original 0B – steps #01/fire)

Gaat goed alleen de eerste fireball heeft de eerste paar bewegingen de verkeerde spritewaarde. Dus de initialisatie daarvan gaat niet goed? Even met debugger bekijken.

Op de 6400 staat de initialisatie van de laatste square lijkt het wel. Dat is ook zo. Als je de squares nog één opschuift dan staat de laatste square initialisatie op 6410. Dus er gaat in de copy routines (2 stuks) iets mis met het berekenen van de offset:

```

114E 218211 LD HL,#1182 ; load HL with start of data table
1151 11A364 LD DE,#64A3 ; load DE with destination ?
1154 011E02 LD BC,#021E ; set counters
1157 CDEC11 CALL #11EC ; copy

; draws black squares next to kong???

115A 217E11 LD HL,#117E ; load HL with start of data table
115D 11A764 LD DE,#64A7 ; set destination sprites
1160 011C02 LD BC,#021C ; set counters B := 2, C := #1C
1163 CD2A12 CALL #122A ; copy

```

Vindt blijkbaar een rollover plaats van de counters van 64FF naar 6400. Dus alleen op te lossen als we de squares niet op 64E0 laten beginnen, maar op 6500. Kijken of dit dan niet de boel in de war gooit.

```

1151 110365 LD DE,#6503 ; (DK squares moved: original A3 – steps #20/fire)
115D 110765 LD DE,#6507 ; (DK squares moved: original A7 – steps #20/fire)
1166 DD210065 LD IX,#6500 ; (DK squares moved: original A0 – steps #20/fire)
2902 060A LD B,#0A ; (collisions on rivets, including hidden squares DK)

```

Gaat nu goed met 7 fireballs. Goede fireball sprites. DK hidden squares goed en clear verwijdert netjes alle sprites.

[Nu versie v0.22 opgeslagen.](#)

Toch even kijken hoe het speelt als de fireballs wel de neiging naar boven hebben.

```

; 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 ; /

```

336B weer herstellen naar D0 i.p.v. NOP.

Hiermee een tijdje testen. Is meer het originele fireball gedrag dat men kent. Nee toch niet zo leuk. Veel leuker wanneer ze veel meer verspreiden.

Toch nog mogelijk om nog een extra fireball toe te voegen? De squares zijn al opgeschoven en er is nog ruimte voor één extra fireball. Dan dus acht: mayem !

Daartoe aanpassen:

2891	0608	LD	B,#08	; (collisions on barrels)
28B1	0608	LD	B,#08	; (collisions on pies)
28E1	0608	LD	B,#08	; (collisions on elevators)
2C67	0608	LD	B,#08	; (check fires for deployment on barrels)
3144	0608	LD	B,#08	; (fireball movement and color when hammering)
34F9	0608	LD	B,#08	; (update fires to sprites)
3EAA	0608	LD	B,#08	; (check jumping fires)
1137	011C08	LD	BC,#081C	; (fireball initialization)
317F	FE08	CP	#08	; (fireball deployment limit)
31D7	FE08	CP	#08	; (fireball processing limit)
30CE	060E	LD	B,#0E	; (clear fireballs: original OB – steps #01/fire)
1151	110365	LD	DE,#6503	; (DK squares moved: original A3 – steps #20/fire)
115D	110765	LD	DE,#6507	; (DK squares moved: original A7 – steps #20/fire)
1166	DD210065	LD	IX,#6500	; (DK squares moved: original A0 – steps #20/fire)
2902	060A	LD	B,#0A	; (collisions on rivets, including hidden squares DK)

Ja dat gaat goed. De squares worden goed afgehandeld en de fireballs worden verwijderd wanneer level finish en/of jumpman dead.

Nu versie v0.23 opgeslagen.

Het board zodanig aanpassen dat de toren scheef staat, dus aan de rechterkant de verschuiving van de trappen hetzelfde maken als aan de linkerkant.

Dat houdt de volgende aanpassingen in:

Girder 1: einde rechts maken als girder 4:	3CEA: 05 D7 F8 10 F8 ; bottom level girder
Girder 2: einde rechts maken als girder 3:	3CE5: 05 DF D0 18 D0 ; level 2 girder
Girder 3: einde rechts maken als girder 2:	3CEO: 05 E7 A8 20 A8 ; level 3 girder
Girder 4: einde rechts maken als girder 1:	3CDB: 05 EF 80 28 80 ; level 4 girder
Bottom right ladder on X top right ladder:	3CCC: 00 CB D0 CB F8 ; bottom right ladder
Second right ladder on X third right ladder:	3CC2: 00 D3 A8 D3 D0 ; level 2 ladder right side
Third right ladder on X second right ladder:	3CAE: 00 DB 80 DB A8 ; level 3 ladder right side
Top right ladder on X bottom right ladder:	3CA4: 00 E3 58 E3 80 ; top right ladder

Dat is niet goed.



Girder 1: einde rechts maken als girder 4:
Girder 2: einde rechts maken als girder 3:
Girder 3: einde rechts maken als girder 2:
Girder 4: einde rechts maken als girder 1:
Bottom right ladder on X top right ladder:
Second right ladder on X third right ladder:
Third right ladder on X second right ladder:
Top right ladder on X bottom right ladder:

3CE5: 05 CF D0 18 D0 ; level 2 girder
3CE0: 05 D7 A8 20 A8 ; level 3 girder
3CDB: 05 DF 80 28 80 ; level 4 girder
3CD6: 05 EF 58 30 58 ; girder kong stands on
3CCC: 00 CB D0 CB F8 ; bottom right ladder
3CC2: 00 D3 A8 D3 D0 ; level 2 ladder right side
3CAE: 00 DB 80 DB A8 ; level 3 ladder right side
3CA4: 00 E3 58 E3 80 ; top right ladder

Dit klopt nog steeds niet:



De volgende girders moeten eentje langer worden:

Girder 1: einde rechts maken als girder 4: 3CE5: 05 D7 D0 18 D0 ; level 2 girder

Girder 2: einde rechts maken als girder 3: 3CEO: 05 DF A8 20 A8 ; level 3 girder

Girder 3: einde rechts maken als girder 2: 3CDB: 05 E7 80 28 80 ; level 4 girder

Is nu wel goed.



Alleen de symmetrie voelt een beetje raar. Ladders verplaatsen.

Bottom center ladder: 1 naar links: 3C90: 00 73 D0 73 F8 ; bottom center ladder

Level 2 ladder 3: 1 naar links: 3CBD: 00 9B A8 9B D0 ; level 2 ladder #3 of 4

Level 3 center ladder: 1 naar rechts: 3C8B: 00 83 80 83 A8 ; center ladder level 3

Top ladder 3: 3 naar rechts: 3C9F: 00 C3 58 C3 80 ; top right ladder (left side)

De top ladder 3 naar rechts is vreemd omdat je daardoor de bovenste rivet altijd als laatste moet doen.

Ziet nu zo uit:



Toch niet zo leuk zo. Hou wel van de symmetrie in het level.

[Terug naar versie v0.23.](#)

Of toch wel aan beide kanten spawnen: 1) als jumpman in een middengebied (waarbij niet te dicht bij spawn gedeelte), dan spawnen aan een random kant!, 2) bepalen hoogte jumpman en niet spawnen op de girder waar jumpman is.

Wordt hier gedaan:

```

34B9 3A2762 LD A,(#6227) ; \ Return if current screen is elevators (Can this ever happen?)
34BC fe03 CP #03 ; |
34Be c8 RET Z ; /
34BF 3A0362 LD A,(#6203) ; \ Jump if bit 7 of Mario's X-position is set (i.e., Mario is on the right half of the screen)
34C2 cb7f BIT 7,A ; |
34C4 C2ED34 JP NZ,#34ED ; /
34C7 21C43A LD HL,#3AC4 ; Load HL with start of table data for spawning fireball on right side

```

Aanpassen:

```

34BF 3A1860 LD A,(#6018) ; \ Jump if bit 1 of RNGTimer1 is set
34C2 CB4F BIT 1,A ; |
34C4 C2ED34 JP NZ,#34ED

```

Om eventjes te testen hoe dat werkt (los van te dichtbij spawnen). Heel lastig doordat de firefoxes nogal verspreid spawnen en er bijna geen doorkomen aan is.

De fireballs spawnen hier:

```
; 3AC4: EE F0 ; bottom, right  
; 3AC6: DB A0 ; middle, right  
; 3AC8: E6 C8 ; 2nd from bottom, right  
; 3ACA: D6 78 ; 2nd from top, right  
  
; 3AD4: 1B C8 ; 2nd from bottom, left  
; 3AD6: 23 A0 ; middle, left  
; 3AD8: 2B 78 ; 2nd from top, left  
; 3ADA: 12 F0 ; bottom, left
```

Als X-positie jumpman tussen #33 en #CE dan willekeurige kant kiezen
Anders positie bepalen op basis linkerkant/rechterkant.

Aanpassen naar:

```
34BF C3D40A      JP    #0AD4      ; jump to additional code – jump to label AAAA  
34C2 00 00 00 00 00 NOPx5
```

en

```
0AD4 3A0362      LD     A,(#6203)   ; load A with jumpman X-position – label AAAA  
;  
0AD7 FE33        CP     #33        ; X-position lower than #33?  
0AD9 DAC734      JP     C,#34ED    ; yes, spawn at left side  
0ADC FECE        CP     #CE        ; X-position higher than #33?  
0ADE D2ED34      JP     NC,#34C7    ; yes, spawn at right side  
;  
0AE1 3A1860      LD     A,(#6018)   ; load A with RNGtimer1  
0AE4 CB4F        BIT    1,A        ; is bit 1 set?  
0AE6 C2ED34      JP     NZ,#34ED    ; no, jump to code select spawn right side  
0AE9 C3C734      JP     #34C7      ; yes, jump to code select spawn left side
```

Gaat goed, maar wanneer je eerst naar links en links blijft dan spawnen ze alsnog allemaal aan de rechterkant.

Nee dit is allemaal niet echt een goed idee.

[Terug naar versie v0.23.](#)

Kijken of de titel in twee kleuren gemaakt kan worden.

Op #36B2 staat de HIGH SCORE tekst:

#36B2: 80 76 18 19 17 18 10 23 13 1F 22 15 3F

Veranderen naar:

#36B2: 80 76 10 96 97 98 99 9A 9B 9C 9D 10 3F

En op die posities de naam aangepast zodat deze gestreept rood en lichtblauw is.

Nu versie v0.24 opgeslagen.

De sprites aangepast voor jumpman, pauline en de hamer naar een wizards thema.

Nu versie v0.29 opgeslagen.

Nu kijken of de bonus items aangepast kunnen worden naar boeken in hetzelfde kleurenpalette als de hamers: wel een beetje oude look.

Daartoe wel de sprite color aanpassen:

```
; 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 sprite color voor de hamers is #07.

Dat is nu goed.

De oil can ook de kleur van de hamers geven.

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

Gaat nu allemaal goed.

Nu versie v0.30 opgeslagen.

Het terug pluggen van de rivets is niet leuk omdat het de mogelijke routes beperkt. Wellicht zo maken dat er twee save spots zijn en dat jumpman op één ervan start. En dat de fireballs meteen respawned worden op de 8 mogelijke plaatsen verspreid over het veld.

Trouwens een raar fenomeen. Wanneer doorgespeeld wordt is het tweede board het pie board. Wanneer je deze finished wordt één van de draaipunten gewist. Dat zijn sprites, maar waarschijnlijk overlapt dat met de sprites van de extra fireballs... Dat op te lossen door het wissen alleen te doen wanneer het rivets scherm is? In de andere velden niet tot 8 fireballs gaan: dat maakt het wel heel moeilijk.

De additionele code voor het terug pluggen van de rivets wordt aangeroepen op #1A81, #1A82 en #1A83. Die terugzetten naar de originele code:

```
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
```

Ja dat werkt. Dan ook de additionele code vrijgeven.

```
#3FC8-#3FF0 vrijgeven  
#3F91-#3F9E vrijgeven
```

1AAC-1AB1 herstellen omdat je nu geen bonus krijgt.

```
1AAC 324063 LD      (#6340),A    ; store into #6340  
1AAF 324263 LD      (#6342),A    ; store into scoring indicator
```

Ja dat is nu weer allemaal goed.

Nu versie v0.31 opgeslagen.

Aantal ideeën:

- In elevators zorgen dat er extra fireballs komen: in stuk waar naar toe springen en beginstuk.
- Wellicht P1 board afhankelijk maken: onsterfelijk in barrels en rivets, maar in pies stoppen van de lopende banden en in elevators stoppen van de elevators?
- Rode barrels die al eerder uiteenspatten tot fireballs?
- Een van de fireballs in rivets veranderen in een anders sprite die de rivets weer dichtgooit?
- Rode barrels die 2 fireballs spawnen? Dus veel lastiger te ontwijken?

Eerst maar eens testen met een randomization van het barrels board (dus randomizer combineren), maar dan met girders die vanuit het midden schuin aflopen. Dan is de afstand tussen de girders gelijk en kan er gekozen worden voor ladders van een veelvoud van vier. Eerst testen met een nieuwe rom en met een PS script die dit veld opbouwt. Aanpassingen voor rollen barrels en eventueel ook nog een trager omhoog rollen in die rom inbouwen en testen. Pas later samenvoegen met wat we nu al in de wizardry rom hebben zitten qua spells en extra fireballs. Eerst kijken of zo'n

barrels board leuk is en of het inderdaad een soort van signature is (omhoog rollen is magisch en de vorm van girders is onderscheidend en opvallend).

Eerst een barrels board maken met de vier middelste barrels en alle ladders (behalve de ladders naar Pauline en de twee DK ladders) verwijderd.

3AE4: 02 97 38 68 38 ; top girder where girl sits
3AE9: 02 9F 54 10 54 ; girder where kong sits
3AED: 02 DF 58 A0 55 ; 1st slanted girder at top right
~~3AF3: 02 EF 6D 20 79 ; 2nd slanted girder (has hammer at left side)~~
~~3AF8: 02 DF 9A 10 8E ; 3rd slanted girder~~
~~3AFD: 02 EF AF 20 BB ; 4th slanted girder~~
~~3B02: 02 DF DC 10 D0 ; 5th slanted girder (has hammer at right side)~~
3B07: 02 FF F0 80 F7 ; bottom slanted girder
3B0C: 02 7F F8 00 F8 ; bottom flat girder where mario starts
~~3B11: 00 CB 57 CB 6F ; short ladder at top right~~
~~3B16: 00 CB 99 CB B1 ; short ladder at center right~~
~~3B1B: 00 CB DB CB F3 ; short ladder at bottom right~~
3B20: 00 63 18 63 54 ; kong's ladder (right)
~~3B25: 01 63 D5 63 F8 ; bottom broken ladder~~
~~3B2A: 00 33 78 33 90 ; short ladder at left side under top hammer~~
~~3B2F: 00 33 BA 33 D2 ; short ladder at left side above oil can~~
3B34: 00 53 18 53 54 ; kong's ladder (left)
~~3B39: 01 53 92 53 B8 ; second broken ladder from bottom, on 3rd girder~~
~~3B3E: 00 5B 76 5B 92 ; longer ladder under the top left hammer~~
~~3B43: 00 73 B6 73 D6 ; longer ladder to left of bottom hammer~~
~~3B48: 00 83 95 83 B5 ; center longer ladder~~
3B4D: 00 93 38 93 54 ; ladder leading to girl
~~3B52: 01 BB 70 BB 98 ; third broken ladder on right side near top~~
~~3B57: 01 6B 54 6B 75 ; fourth broken ladder near kong~~
3B5C: AA ; AA code signals end of data

En deze als basis nemen voor het script zodat de randomizations gedaan kunnen worden.

Eerst de intro sequence eruit gooien.

De code is #0A76 t/m #0BD9. De aanroep daarnaartoe staat hier:

```
070E 63 0A          ; (6) #0A63      ; clears screen and sprites, check for intro screen to run
0710 76 0A          ; (7) #0A76      ; kong climbs ladders and scary music played
0712 DA 0B          ; (8) #0BDA      ; draw goofy kongs, how high can you get, play music
```

Dus uitzetten door deze aanroep in NOP's te veranderen: #0710: NOP, NOP.

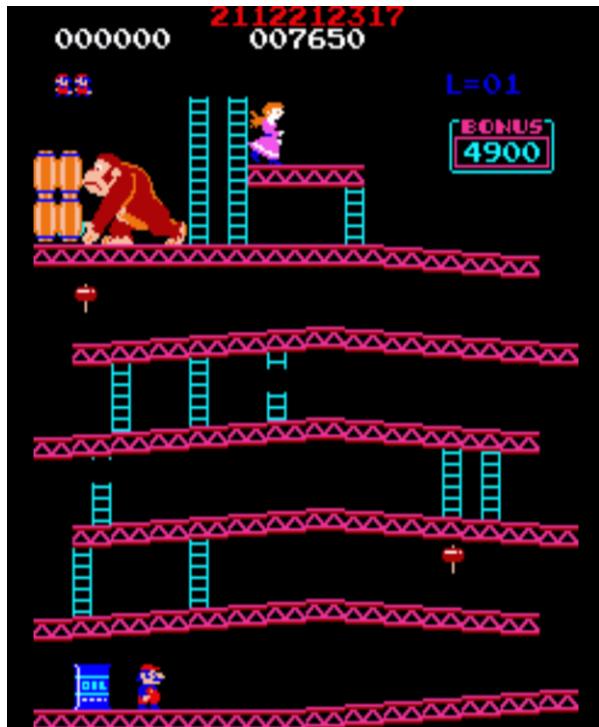
De gamemode moet aangepast worden. Dus de aanroep weer terugzetten en dan de code aanpassen:

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

0A7A C9 RET

En dan vrijgeven #0A7B t/m 0BD9 (doen we in dit prototype niet).

Met PowerShell script een board gemaakt met girders die in het midden het hoogste zijn.



Nu nog aanpassen dat de barrels, fireballs en jumpman aangepast worden voor de schuine girders.

In Barrelopaloza ook zo iets gedaan voor de omgedraaide girder waar de barrels omhoog rollen. Iets soortgelijks maken voor de vier girders. Daar gewoon afkijken.

De originele code die de aanpassing voor de schuine girders doet: #2333 t/m #236B. Een deel daarvan is het afvangen van de rechte girders boven en onder. Maar het volgende stukje doet de echte aanpassing:

```
2353 7D      LD     A,L  
2354 CB6F      BIT    5,A  
2356 CA5C23    JP     Z,#235C  
  
2359 90      SUB    B  
235A 6F      LD     L,A  
235B C9      RET          ; return  
  
235C 80      ADD    A,B          ; A := A + B  
235D C35A23    JP     #235A        ; loop back
```

Op basis van de waarde van bit 5 van L (bevat de Y-positie) (horizontale banden van het scherm), wordt een aanpassing naar boven of naar beneden gedaan.

Je kunt dit aanpassen door:

2353	C3303F	JP	#3F30	; jump to additional code – jump to label AAAA
en:				
3F30	7D	LD	A,L	; load A with Y-position – label AAAA
3F31	FE68	CP	#68	; check Y-position higher than #68?
3F33	3807	JR	C,#BBBB	; no, continue original code – jump to label BBBB
3F35	FEE0	CP	#E0	; check Y-position lower than #E0?
3F37	3003	JR	NC,#BBBB	; no, continue original code – jump to label BBBB
;				
3F39	C35923	JP	#2359	; roll downhill
;				
3F3C	CB6F	BIT	5,A	; original code – label BBBB
3F3E	CA5C23	JP	Z,#235C	
3F41	C35923	JP	#2359	

Aanpassen voor roll uphill:

3F39	C35C23	JP	#235C	; roll uphill
------	--------	----	-------	---------------

De nieuwe 4 girders liggen tussen:

- 71 – 78:
- 91 – 99:
- B1 – B8:
- D1 – D9:

Dus daarom hierboven checken op waarden tussen #68 en #E0.

Dat klopt allemaal. Lijkt erop dat nu alleen nog het onderscheid links rechts van de top van belang is. De top van de girders ligt op 8F. Dus dat vooralsnog als test nemen en dan rechts ervan een roll uphill doen en links ervan een roll downhill doen.

3F30	7D	LD	A,L	; load A with Y-position – label AAAA
3F31	FE68	CP	#68	; check Y-position higher than #68?
3F33	380E	JR	C,#BBBB	; no, continue original code – jump to label BBBB
3F35	FEE0	CP	#E0	; check Y-position lower than #E0?
3F37	300A	JR	NC,#BBBB	; no, continue original code – jump to label BBBB
;				
3F39	7C	LD	A,H	; load A with X-position
3F3A	FE8F	CP	#8F	; check X-position left of #8F?
3F3C	7D	LD	A,L	; load A with Y-position
3F3D	DA5923	JP	C,#2359	; yes, jump to roll downhill
3F40	C35C23	JP	#235C	; roll uphill
;				
3F43	CB6F	BIT	5,A	; original code – label BBBB
3F45	CA5C23	JP	Z,#235C	
3F48	C35923	JP	#2359	

Ja dat werkt goed. De barrels rollen goed nu. En zijn op verschillende plekken met succes te sturen.

De rollover gaat nog niet goed. Idee dat opkwam: is het mogelijk om de girder waar de barrels opvallen te laten doorlopen in het onzichtbare deel van het scherm? Gaat dat goed?

Girder array nu:

```
$girderarray = @C (2,0x8F,0x71,0x20,0x78),      # top girder - left half
(2,0xEF,0x77,0x90,0x72),      # top girder - right half
(2,0x8F,0x91,0x10,0x99),      # top girder - left half
(2,0xDF,0x96,0x90,0x92),      # top girder - right half
(2,0x8F,0xB1,0x20,0xB8),      # top girder - left half
(2,0xEF,0xB7,0x90,0xB2),      # top girder - right half
(2,0x8F,0xD1,0x10,0xD9),      # top girder - left half
(2,0xDF,0xD6,0x90,0xD2) )    # top girder - right half
```

En dan de 2e girder aan de linkerkant laten doorlopen, dus op derde regel, 2e coördinaat veranderen in 0x00,0x9A? Ja dat gaat goed. Nu wel rollover, barrel deels uit het zicht.

Hetzelfde doen met de andere girders.

De 3e girder aan de rechterkant laten doorlopen, dus op de zesde regel, de 1e coördinaat veranderen in 0xFF,0xB8.

De 4e girder aan de linkerkant laten doorlopen, dus op de zevende regel, de 2e coordinaat veranderen in 0x00, 0xDA.

Dat gaat goed. Alle rollovers gaan goed. Alleen nog wel kijken of jumpman nu ook niet uit het beeld kan lopen. Ik denk van niet omdat als dat het geval was, hij in de originele rom ook van de girders zou afvallen.

Girder array nu:

```
$girderarray = @C (2,0x8F,0x71,0x20,0x78),      # top girder - left half
(2,0xEF,0x77,0x90,0x72),      # top girder - right half
(2,0x8F,0x91,0x00,0x9A),      # third girder - left half
(2,0xDF,0x96,0x90,0x92),      # third girder - right half
(2,0x8F,0xB1,0x20,0xB8),      # second girder - left half
(2,0xFF,0xB8,0x90,0xB2),      # second girder - right half
(2,0x8F,0xD1,0x00,0xDA),      # bottom girder - left half
(2,0xDF,0xD6,0x90,0xD2) )    # bottom girder - right half
```

Vaste trappen vanaf de bottom girder en naar de top girder toevoegen.

Zijn toegevoegd. Alleen de top right ladder ging niet goed. Alleen goed te krijgen door deze twee korter te maken aan de bovenkant.

```
(0x01,0x63,0xD3,0x63,0xF8),      # bottom broken ladder
(0x00,0xDB,0xD6,0xDB,0xF2),      # bottom right ladder
(0x01,0x6B,0x54,0x6B,0x73),      # top broken ladder
(0x00,0xCB,0x59,0xCB,0x75) )    # top right ladder
```

Dat betekent dat nog het volgende aangepast moet worden:

- Top rechte girder moet twee segmenten korter worden.

02 7F 54 10 54 ; girder where kong sits

- Top schuine girder moet twee segmenten langer worden.
02 DF 5A 80 55 ; 1st slanted girder at top right
- Ladder naar Pauline moet twee langer worden aan de onderkant.
00 93 38 93 56 ; ladder leading to girl
- De speciale adjustment voor schuine top girder moet opgeschoven worden.

De code voor de speciale adjustment zit hier:

```

2366 7C      LD      A,H          ; A := H
2367 FE98      CP      #98          ; < #98 ?
2369 D8      RET      C          ; no, return

236A 7D      LD      A,L          ; A := L
236B C35C23    JP      #235C        ; loop back

```

Dus dan de check op de x-coördinaat ook twee segmenten naar links schuiven:

```
2367 FE78  CP      #78          ; < #78 ?
```

Lijkt nu goed te gaan. Alleen soms wordt een hele ladder als gebroken getekend. Hoe kan dat nou? Eventjes een keer testen met ladders op alle plaatsen.

Interessant idee. Kun je barrels een willekeurige kant op laten rollen wanneer ze van een ladder afkomen? Dat maakt het wel heel erg lastig. Twee richtingsverkeer. Lijkt wel heel erg lastig. Maar leuk om eens te proberen.

Testen met snelheid van de barrels. Afhandeling is verschillend voor naar links en naar rechts rollen:

```

1F93 DD7E01    LD      A,(IX+#01)    ; Load Crazy Barrel indicator
1F96 3D      DEC      A          ; is this a crazy barrel?
1F97 CAEC20    JP      Z,#20EC        ; Yes, jump ahead

1F9A DD7E02    LD      A,(IX+#02)    ; no load A with next indicator - determines the direction of the barrel
1F9D 1F      RRA          ; Is this barrel going down a ladder?
1F9E DAAC1F    JP      C,#1FAC        ; Yes, jump away to ladder sub.

1FA1 1F      RRA          ; Is this barrel moving right?
1FA2 DAE51F    JP      C,#1FE5        ; yes, jump away to move right sub.

1FA5 1F      RRA          ; is this barrel moving left?
1FA6 DAEF1F    JP      C,#1FEF        ; yes, jump to moving left sub

1FA9 C35320    JP      #2053        ; else jump ahead

```

Aanpassing voor naar rechts rollen:

Bedenk dat dit op de bottom en de third girder is en dan alleen aan de linkerkant van de girder.

Eerst testen op rechts: zo niet dan terug naar originele code en dan testen tussen #90 en #98 (third girder) of tussen #D0 en #D8 (bottom girder). Kijken of dit ook met een bit check gedaan kan worden:

```
72 - 77 - 0111 0010 - 0111 0111
92 - 96 - 1001 0010 - 1001 0110
B2 - B6 - 1011 0010 - 1011 0110
D2 - D6 - 1101 0010 - 1101 0110
```

Het onderscheid zit dus in bit 5. Als bit 5 nul is dan barrel langzamer laten rollen

Aanpassen:

1FA2 DA4B3F	JP	C,#3F4B	; yes, jump to additional code – jump to label AAAA
-------------	----	---------	--

en:

3F4B DD7E03	LD	A,(IX+#03)	; load A with the barrels X-position – label AAAA
3F4E FE88	CP	#88	; is Y-position at the right part of the girder ?
3F50 D2E51F	JP	NC,#1FE5	; yes, jump back to original code
;			
3F53 DD7E05	LD	A,(IX+#05)	; load A with the barrels Y-position – label AAAA
3F56 CB6F	BIT	5,A	; is bit 5 of X-position set?
3F58 C2E51F	JP	NZ,#1FE5	; yes, jump back to original code
;			
3F5B 3A1A60	LD	A,(FrameCounter)	; load A with frame counter ?
3F5E E603	AND	#01	; mask bits. random number between 0 and 1
3F60 CA8A1F	JP	Z,#1F8A	; if 0: return without processing barrel, do next
3F63 C3E51F	JP	#1FE5	; jump back to original code

En hetzelfde voor de barrels die naar links rollen. Maar dan omgedraaid.

Aanpassen:

1FA6 DA663F	JP	C,#3F66	; yes, jump to additional code – jump to label AAAA
-------------	----	---------	--

en:

3F66 DD7E03	LD	A,(IX+#03)	; load A with the barrels X-position – label AAAA
3F69 FE88	CP	#88	; is Y-position at the left part of the girder ?
3F6B DAEF1F	JP	C,#1FEF	; yes, jump back to original code
;			
3F6E DD7E05	LD	A,(IX+#05)	; load A with the barrels Y-position – label AAAA
3F71 CB6F	BIT	5,A	; is bit 5 of X-position set?
3F73 CAEF1F	JP	Z,#1FEF	; yes, jump back to original code
;			
3F76 3A1A60	LD	A,(FrameCounter)	; load A with frame counter ?
3F79 E603	AND	#01	; mask bits. random number between 0 and 1

3F7B CA8A1F	JP	Z,#1F8A	; if 0: return without processing barrel, do next
3F7E C3EF1F	JP	#1FEF	; jump back to original code

Gaat nu goed. Wel erg lastig.

Na een aantal velden gespeeld te hebben, dit terugdraaien. Is te ingewikkeld. Is leuker als er druk van de fireballs komt.

De eerste barrel is altijd een wild blue barrel. Deze rood maken. En dan bij het bouncen op een girder checken of het een red barrel is en dan op basis van een kans van 1-op-2 een fireball spawnen op de plek van de red barrel en deze weer blue maken. Als nog rood bij tweede girder dan sowieso een fireball spawnen en weer blue maken. De barrel rolt dan weer door en spawnt een tweede fireball bij de oil can.

Kleur van de blue barrel wordt hier gezet:

```

; 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

```

Maar dan eigenlijk checken of dit de eerste barrel is. Zo ja, dan de kleur op rood zetten. Checken eerste barrel zou moeten kunnen met checken van waarde van #6382:

```

; #6382 = 00 and turns to 80 when a blue barrel is about to be deployed.
; First blue barrel has this at 81 and then 02. changes to 1 for crazy barrel
; Bit 7 is set when barrel is blue
; Bit 0 is set when barrel is crazy
; bit 1 is set for the second barrel of the round which can't be crazy

```

Eerst eens debuggen met watchpoint op 2CFA: stopt dan bij eerste blue barrel. Dan kijken wat er in #6382 zit. Ja daar zit #81 in. En #80 bij een volgende blauwe barrel.

Aanpassen:

2D0D CD4B3F	CALL #3F4B	; jump to additional code – jump to label AAAA
2D10 00	NOP	

en:

```
3F4B 3A8263 LD A,(#6382) ; load barrel crazy and color status – label AAAA
3F4E FE81 CP #81 ; is this the first crazy/blue barrel?
3F50 2805 JR Z,#BBBB ; yes, set color to red – jump to label BBBB
;
3F52 DD36080C LD (IX+#08),#0C ; set sprite color to blue
3F56 C9 RET ; return
;
3F57 DD36080F LD (IX+#08),#0F ; set sprite color to red
3F5B C9 RET ; return
```

Ja dat gaat goed. De eerste blue crazy barrel wordt rood gemaakt. Dan nu nog bij het bouncen afvangen en springen naar code die checkt of er fireball gedeployed moet worden. En zo ja dan dat doen, barrel weer blue maken en geluid geven.

De afhandeling van het raken van de girders bij de eerste barrel gebeurt hier:

```
; arrive here when crazy barrel is onscreen
; called when barrel deployed or hits a girder on the way down
; called from #2149

22CB 3A4863 LD A,(#6348) ; load A with oil can status
22CE A7 AND A ; is the oil can lit ?
22CF CAE122 JP Z,#22E1 ; no , jump ahead

22D2 3A8063 LD A,(#6380) ; else load A with difficulty
22D5 3D DEC A ; decrement. will be between 0 and 4
22D6 EF RST #28 ; jump based on A

22D7 F6 22 ; #22F6
22D9 F6 22 ; #22F6
22DB 03 23 ; #2303
22DD 03 23 ; #2303
22DF 1A 23 ; #231A

; arrive here when oil can is not yet lit
; used for initial crazy barrel

22E1 3A2962 LD A,(#6229) ; load A with level #
22E4 47 LD B,A ; store into B
22E5 05 DEC B ; decrement B
22E6 3E01 LD A,#01 ; load A with 1
22E8 CAF922 JP Z,#22F9 ; if level was 1, then jump ahead

22EB 05 DEC B ; decrement B again
22EC 3EB1 LD A,#B1 ; load A with #B1 - for use with level 2 initial crazy barrel
22EE CAF922 JP Z,#22F9 ; if level 2, then jump ahead

22F1 3EE9 LD A,#E9 ; else load A with #E9 - for level 3 and up initial crazy barrel
22F3 C3F922 JP #22F9 ; jump ahead and store
```

Dus de fireball check tussenvoegen op #22E1 en dan als laatste voor het terug springen de oorspronkelijke operatie van #22E1 (laden level nr) doen.

Aanpassen:

22E1 C35C3F JP #3F5C ; jump to additional code – jump to label AAAA

en:

3F5C	DD7E08	LD	A,(IX+#08)	; load A with color of first crazy barrel – label AAAA
3F5F	FEOF	CP	#0F	; is it still a red barrel?
3F61	2019	JR	NZ,#BBBB	; no, do not deploy fireball – jump to label BBBB
;				
3F63	DD7E05	LD	A,(IX+#05)	; load Y-position of barrel
3F66	FE60	CP	#60	; is Y-position lower than #60?
3F68	3812	JR	C,#BBBB	; no, do not deploy fireball -jump to label BBBB
;				
3F6A	FEC0	CP	#C0	; is Y-position higher than #C0?
3F6C	3007	JR	NC,#CCCC	; yes, deploy fireball – jump to label CCCC
;				
3F6E	3A1A60	LD	A,(FrameCounter)	; load A with FrameCounter (random)
3F71	CB57	BIT	2,A	; is bit 2 set ?
3F73	2007	JR	NZ,#BBBB	; no, do not deploy fireball – jump to label BBBB
;				
3F75	DD36080C	LD	(IX+#08),#0C	; set sprite color to blue - label CCCC
3F79	CD7B0A	CALL	#0A7B	; deploy fireball
;				
3F7C	3A2962	LD	A,(#6229)	; load A with level # - original code – label BBBB
3F7F	C3E422	JP	#22E4	; jump back

en:

0A7B	DD7E05	LD	A,(IX+#05)	; load a with Y-position of barrel
0A7E	D603	SUB	#03	; adjust for fireball position
0A80	47	LD	B,A	; save to b
0A81	DD7E03	LD	A,(IX+#03)	; load a with X-position of barrel
;				
0A84	DDE5	PUSH	IX	
0A86	DD210064	LD	IX,#6400	; load IX with start of fire #1
0A8A	DD360001	LD	(IX+#00),#01	; set fire active
;				
0A8E	DD7703	LD	(IX+#03),A	; set fire X position
0A91	DD770E	LD	(IX+#0E),A	; set fire X position #2
0A93	78	LD	A,B	; load a with Y-position of barrel from b
0A95	DD7705	LD	(IX+#05),A	; set fire Y position
0A98	DD770F	LD	(IX+#0F),A	; set fire Y position #2
0A9B	DDE1	POP	IX	
0A9D	C9	RET		; return

Door intro sequence is vrijgeven #0A7B t/m 0BD9 (doen we in dit prototype nu wel).

Ja dat gaat goed. Spawned één extra fireball op een girder boven jumpman. Kijken of het ook werkt met de andere levels, waarbij de eerste barrel schuin gegooid wordt.

Level aanpassen:

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

De eerste waarde is het level.

Bij alle drie de mogelijke levels gaat het goed.

Alleen gaat er nog iets niet goed met de random bepaling. Steeds of de bovenste, of de onderste. Meer random maken? Ander bit test doen?

```
3F71 CB47 BIT 0,A ; is bit 0 set ?
```

Ook nog een geluid geven wanneer de fireball spawned wordt en verder testen met de bit 0. Alle girders een keer gebruikt? En hoe vaak ongeveer: aantal keren testen, random verdeling.

Aanpassing voor geluid toevoegen:

```
0A9D 3E03 LD A,#03 ; load sound duration of 3  
0A9F 328260 LD (#6082),A ; play boom sound using sound buffer  
;  
0AA2 C9 RET ; return
```

Ja dat gaat goed.

Nu versie v0.33 opgeslagen

Alleen soms later ook nog een rode barrel gooien, maar die wordt niet meer blauw. Hoe kan dat? Soms ook nog steeds een bepaalde trap die gebroken getekend wordt, terwijl het een normale trap is.

Dit oplossen en dan de PowerShell code gaan refactoren naar code die erg op assembly language lijkt zodat het eenvoudig om te zetten is. En daarna P2 functies eruit halen en het How High scherm eenvoudiger maken, waardoor er ruimte komt om andere aanpassingen te doen.

- ~~Vaste girders en ladders overzetten naar de default rom~~
- PowerShell code voor variabele ladders omzetten naar assembly pseudo
- Random generator in default rom gebaseerd op uitlezen code
- Variabele ladders opnemen in default rom
- Oplossen ongewenste rode barrels anders dan de eerste wild barrel
- Oplossen onterecht als broken weergegeven ladders

- P2 functies verwijderen: ruimte vrijmaken
- How High scherm aanpassen: ruimte vrijmaken
- Titel aanpassen en andere kleur geven
- De kleur van de items en de vuurbeker aanpassen
- De P2 spell opnieuw aanmaken (fireballs laten stoppen)

Eerst de girders en vaste ladders overzetten naar de rom.

```
$girderarray = @C (2,0x8F,0x71,0x20,0x78),      # top girder - left half
(2,0xEF,0x77,0x90,0x72),      # top girder - right half
(2,0x8F,0x91,0x00,0x9A),      # third girder - left half
(2,0xDF,0x96,0x90,0x92),      # third girder - right half
(2,0x8F,0xB1,0x20,0xB8),      # second girder - left half
(2,0xFF,0xB8,0x90,0xB2),      # second girder - right half
(2,0x8F,0xD1,0x00,0xDA),      # bottom girder - left half
(2,0xDF,0xD6,0x90,0xD2) )    # bottom girder - right half

$ladderarray = @C (0,0,0,0,0), (0,0,0,0,0),
(0,0,0,0,0), (0,0,0,0,0),
(0,0,0,0,0), (0,0,0,0,0),
(0,0,0,0,0), (0,0,0,0,0),
(0x01,0x63,0xD3,0x63,0xF8),      # bottom broken ladder
(0x00,0xDB,0xD6,0xDB,0xF2),      # bottom right ladder
(0x01,0x6B,0x54,0x6B,0x73),      # top broken ladder
(0x00,0xCB,0x59,0xCB,0x75) )      # top right ladder
```

Is gedaan: de definitie van de vaste onderdelen is nu:

3AE4:	02 97 38 68 38	; top girder where Pauline sits
3AE9:	02 7F 54 10 54	; girder where Kong sits
3AEE:	02 DF 5A 80 55	; 1st slanted girder at top right
3AF3:	02 FF F0 80 F7	; bottom slanted girder
3AF8:	02 7F F8 00 F8	; bottom flat girder where Jumpman starts
3AFD:	00 63 18 63 54	; Kong's ladder (right)
3B02:	00 53 18 53 54	; Kong's ladder (left)
3B07:	00 93 38 93 56	; ladder leading to Pauline
3B0C:	02 8F 71 20 78	; top mountain girder – left half
3B11:	02 EF 77 90 72	; top mountain girder – right half
3B16:	02 8F 91 00 9A	; third mountain girder – left half
3B1B:	02 DF 96 90 92	; third mountain girder – right half
3B20:	02 8F B1 20 B8	; second mountain girder – left half
3B25:	02 FF B8 90 B2	; second mountain girder – right half
3B2A:	02 8F D1 00 DA	; bottom mountain girder – left half
3B2F:	02 DF D6 90 D2	; bottom mountain girder – right half
3B34:	01 63 D3 63 F8	; bottom broken ladder
3B39:	00 DB D6 DB F2	; bottom right ladder
3B3E:	01 6B 54 6B 73	; top broken ladder
3B43:	00 CB 59 CB 75	; top right ladder
3B48:	AA	

Nu versie v0.34 opgeslagen

Nu kijken of de bonus items aangepast kunnen worden naar boeken in hetzelfde kleurenpalette als de hamers.

Daartoe wel de sprite color aanpassen:

```
; 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 sprite color voor de hamers is #07.

Dat is nu goed.

De oil can ook de kleur van de hamers geven.

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

Gaat nu allemaal goed.

Nu versie v0.35 opgeslagen.

Eventjes een PowerShell script gemaakt dat het aantal mogelijke combinaties uitrekent. Dat zijn per ladderniveau 220 combinaties. En dat dan op 3 niveau's: $220 * 220 * 220 = 10.648.000$ combinaties!

Nu de PowerShell gaan vereenvoudigen. Eerst eens kijken of de array met y-posities niet vereenvoudigd kan worden. Als de posities evenredig zijn, dan kan je het ook uitrekenen.

```
$ladder_ypos = @(0x78, 0x98, 0xB8, 0xD8)
```

Dat is dan 120, 152, 184, 216, dus verschillen: 32, 32, 32. Dus gelijke afstanden. Dus dat kan op basis van een offset en dan op basis van de niveau-teller een aantal keren 32 erbij/eraf. Dat betekent ook dat de ladders altijd eenzelfde lengte hebben: ook 32.

In assembly language doet de DJNZ een loop met b-register als counter, maar dan telt het af i.p.v. op. Dit nabootsen in de PowerShell loop. Dan dus vanaf beneden de niveau-teller laten lopen (hogere Y-waarde is lager op het scherm). Ook dit aanpassen in de PowerShell.

Hier voor gebruiken: Randomize Wizard v0.2.ps1.

Ja dat gaat goed.

Nu versie v0.36 opgeslagen.

Kijken of de PowerShell meer in lijn met assembly language gebracht kan worden. Dus gebaseerd op de beschikbare opcodes en gebruik makend van registers en memory addressen.

Twee loops in elkaar met memory addressen die de loop counters doen. Geen gebruik maken van DJNZ met B register omdat je dan B verder niet kunt gebruiken. En dan hebben we wel nodig denk ik zo.

Memory 1: #XXXX	- loop counter for ladder level (outer loop)
Memory 2: #XXXX	- loop counter for ladders (inner loop)
Memory 3: #XXXX	- ladder level on which to skip broken ladder

Hier voor gebruiken: Randomize Wizard v0.3.ps1

Ja dat gaat goed.

Kijken of het bepalen van het laddertype en het skippen van één van de gebroken ladders gecombineerd kan worden.

Kijken of de X-positie eenvoudiger bepaald kan worden (nu met twee offsets en deze combineren).

Ja dat gaat goed.

Hier voor gebruiken: Randomize Wizard v0.4.ps1

Nu versie v0.37 opgeslagen.

Kijken of de ladder array als een enkele dimensionale array gedefinieerd en gevuld kan worden. En dan met een teller erdoorheen gaan: soort HL als pointer. Hier is de HL dan de pointer in de array, maar straks in de echte code een pointer naar een reeks van geheugenplaatsen. Maar dan is de werking alvast overeenkomstig.

Ja dat gaat goed.

Een aantal variabelen op een low level manier gebruiken: registers en memory waarden opvragen.

Ja dat gaat goed.

Hier voor gebruiken: Randomize Wizard v0.5.ps1

Nu versie v0.38 opgeslagen.

Geheugenplaatsen:

Memory 1: #XXXX	- loop counter for ladder level (outer loop)
Memory 2: #XXXX	- loop counter for ladders (inner loop)
Memory 3: #XXXX	- ladder level on which to skip broken ladder
Memory 4: #XXXX	- ladder level Y-value
Memory 5: #XXXX	- 1st ladder offset
Memory 6: #XXXX	- 2nd ladder offset
Memory 7: #XXXX	- 3rd ladder offset

Vooralsnog voldoende geoptimaliseerd en aangepast dat we hieruit de assembly code zouden moeten kunnen reconstrueren. Eerst belangrijk om twee voorbereidende stappen te realiseren:

1. Code voor het kiezen van een random nummer tussen 0 en 11 op basis van de programma code zelf. Plek in het geheugen bijhouden op basis van een geheugenplaats die niet gewist wordt tijdens het spelen. Eerste plek op basis van random seed aan begin van spel vaststellen. Daarna steeds ophogen en wanneer aan einde van programma code dan weer terug naar het begin.
2. Code voor het genereren van drie verschillende random nummers tussen 0 en 11 en di opslaan in Memory 5 t/m memory 7.

Eerst geheugenplaatsen zoeken die het hele spel niet gereset worden. In de code wordt het volgende gereset:

```
; called from #0D5F
; clears memories from #6200 - 6227 and #6280 to 6B00
; [why are #6280 - #6280+40 cleared? they are set immediately after]
; computes initial timer
; initializes all sprites
```

H en L opslaan dus daar zijn twee geheugenplaatsen nodig. Daarvoor dan reserveren:

Memory: #6232	- Opslaan H van de pointer into code
Memory: #6233	- Opslaan L van de pointer into code

Eerder al code gemaakt voor het uitlezen van de code. Deze aanpassen aan deze geheugenplaatsen èn zodanig dat altijd een getal tussen 0 en 11 het resultaat is:

```
0AA3 3A3262      LD   A,(#6232)    ; get stored value for H – label BBBB
0AA6 67          LD   H,A        ; store in H
0AA7 3A3362      LD   A,(#6233)    ; get stored value for L
0AAA 3C          INC  A          ; store in L – INC A (3C)
0AAB 6F          LD   L,A        ; increase L – LD L,A (6F)
0AAC 2008        JR   NZ         ; no carry, skip next steps – jump to label CCCC
;
0AAE 24          INC  H          ; increase H
0AAF 7C          LD   A,H        ; load A with H
0AB0 FE40        CP   #40        ; H==#40 ?
```

```

0AB2 2002      JR   NZ           ; no, skip next steps – jump to label CCCC
;
0AB4 2601      LD   H,#01        ; reset H
;
0AB6 7E         LD   A,(HL)      ; load opcode from HL – label CCCC
0AB7 47         LD   B,A          ; store A in B
0AB8 7C         LD   A,H          ;
0AB9 323262    LD   (#6232),A   ; save H
0ABC 7D         LD   A,L          ;
0ABD 323362    LD   (#6233),A   ; save L
0AC0 78         LD   A,B          ; restore A from B
;
0AC1 C9         RET            ; return opcode as random number in A

```

En dan code in de main steeds opnieuw aanroepen, zonder er iets mee te doen en dan met de debugger de getallen bekijken (welke opcode en welk getal onder de twaalf). Daarna hele reeks noteren en kijken naar verdeling en kijken of de rollover bij het einde goed gaat. Daarna ook nog kijken naar het initieel vullen op basis van een seed. En hoe wordt daarmee omgegaan in het geval van een nieuwe game. Dus op welk moment de seed vaststellen en geen onbedoelde reset ergens.

Code om het random getal te laten zien invoegen in de main. Lege plek voor call op #19C2.

```

19C2 CDA30A      CALL  #0AA3        ; call test rng code – call to label XXXX

```

Dat gaat goed.

Nu versie v0.39 opgeslagen.

Nu zodanig maken dat de teruggegeven waarde in A tussen de 0 en 11 ligt.

Bepalen op bitgroepen uit de waarde:

Bit 0 – waarde tussen 0 en 1

Bit 1,2 – waarde tussen 0 en 3

Bit 3,4,5 – waarde tussen 0 en 7

Deze drie bij elkaar optellen geeft waarde tussen 0 en 11 (1+3+7).

```

0AC1 E601      AND  #01        ; keep bit 0 – value between 0 and 1
0AC3 4F         LD   C,A          ; save to C
;
0AC4 78         LD   A,B          ; restore original opcode value from B
0AC5 1F         RRA            ; rotate right
0AC6 E603      AND  #03        ; keep bits 0 and 1 – value between 0 and 3
0AC8 81         ADD  A,C          ; add C
0AC9 4F         LD   C,A          ; save to C
;
0ACA 78         LD   A,B          ; restore original opcode value from B again
0ACB 1F         RRA            ; rotate right

```

```

0ACC 1F          RRA           ; rotate right
0ACD 1F          RRA           ; rotate right
0ACE E607        AND  #07      ; keep bits 0, 1 and 2 – value between 0 and 7
0ADO 81          ADD  A,C      ; add C
;
0AD1 C9          RET           ; return

```

Watchpoints zetten op 0AC0 en 0ADO en checken.

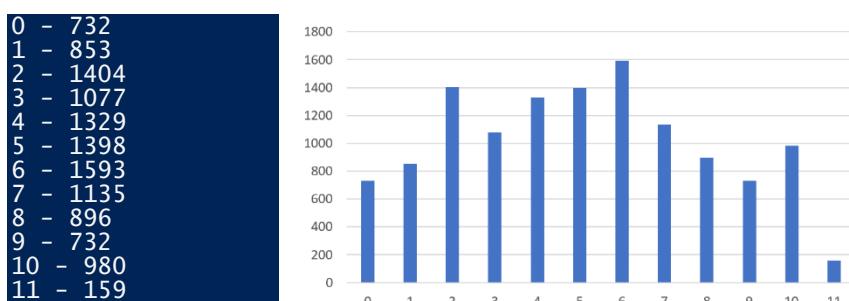
$32 \rightarrow 07 = 0011\ 0010 - 0 + 01 + 110 = 0 + 1 + 6 = 7.$
 $84 \rightarrow 02 = 1000\ 0100 - 0 + 10 + 000 = 0 + 2 = 2.$
 $7D \rightarrow 0A = 0111\ 1101 - 1 + 10 + 111 = 1 + 2 + 7 = A.$
 $C3 \rightarrow 02 = 1100\ 0011 - 1 + 01 + 000 = 1 + 1 + 0 = 2.$
 $02 \rightarrow 01 = 0000\ 0010 - 0 + 01 + 000 = 0 + 1 = 1.$
 $3A \rightarrow 08 = 0011\ 1010 - 0 + 01 + 111 = 0 + 1 + 7 = 8.$

Dat gaat allemaal zoals beoogd.

Nu versie v0.40 opgeslagen.

Eventjes met een PowerShell script checken wat de verdeling is van de code wanneer die omgezet wordt naar getallen tussen 0 en 11. Gebaseerd op drie rom-files. De '0' rom-file wordt niet meegenomen omdat daar een heel deel vrijgegeven code (met NOP's gevuld) is en dat geeft een vertekend beeld.

Resultaten:



Niet echt een hele lekkere verdeling. Hierdoor wellicht een te grote focus op de middelste ladders.

Kijken of je een betere verdeling kunt krijgen. Betere verdeling is liefste zo vlak mogelijk of in ieder geval aan de buitenkanten ook meer kans.

Onderzoeken:

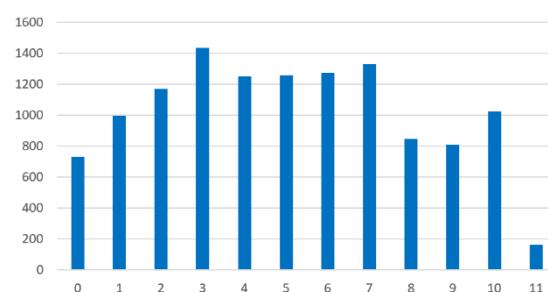
- Andere verdeling in bitgroepen?
- Gebaseerd op twee op-codes?
- Omwisselen van resultaten?

Eerst testen met andere verdeling van de bitgroepen:

Origineel : XX33 3221
Variant1: XX22 1333
Variant2: XX22 3331
Variant3: XX12 2333
Variant4: XX13 3322

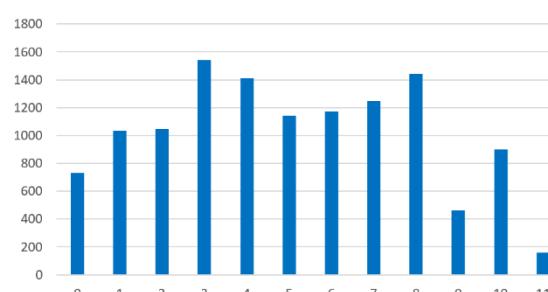
Variant1: XX22 1333

0 -	732
1 -	998
2 -	1171
3 -	1437
4 -	1252
5 -	1257
6 -	1274
7 -	1332
8 -	846
9 -	807
10 -	1023
11 -	159



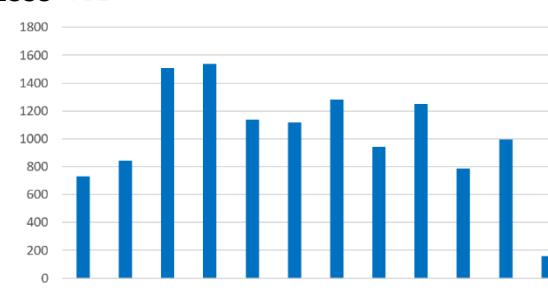
Variant2: XX22 3331

0 -	732
1 -	1035
2 -	1049
3 -	1540
4 -	1412
5 -	1144
6 -	1171
7 -	1245
8 -	1442
9 -	460
10 -	899
11 -	159



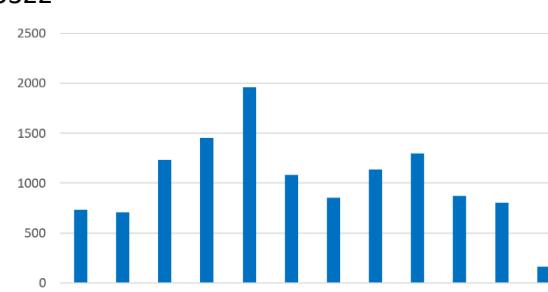
Variant3: XX12 2333

0 -	732
1 -	843
2 -	1508
3 -	1536
4 -	1138
5 -	1117
6 -	1282
7 -	941
8 -	1249
9 -	787
10 -	996
11 -	159



Variant4: XX13 3322

0 -	732
1 -	709
2 -	1232
3 -	1453
4 -	1960
5 -	1083
6 -	854
7 -	1135
8 -	1297
9 -	872
10 -	802
11 -	159

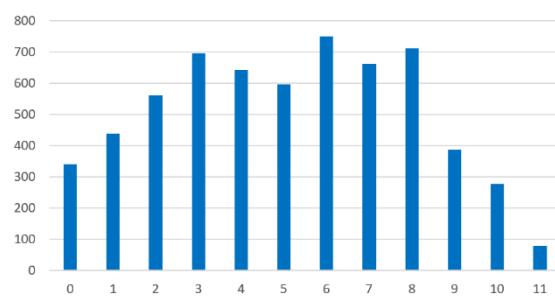


Hele verschillende resultaten qua verdeling. De meest interessante verdelingen zijn variant 1 en variant 2 en dat is allebei met de 333 op de lage bits. Maar in beiden ook nog steeds een hele lage waarde voor de 11.

Kijken wat er gebeurt als je twee opcodes pakt met dan de eerste opcode 0000 0333 en de tweede opcode 000 0221.

Variant 5:

0 -	340
1 -	439
2 -	562
3 -	697
4 -	643
5 -	596
6 -	751
7 -	662
8 -	713
9 -	387
10 -	276
11 -	78

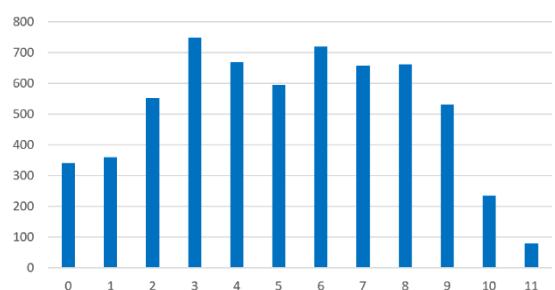


Dat is niet beter, maar juist slechter.

Kijken wat er gebeurt als je twee opcodes pakt met dan de eerste opcode 0000 0333 en de tweede opcode 000 0122.

Variant 6:

0 -	340
1 -	359
2 -	553
3 -	748
4 -	669
5 -	594
6 -	719
7 -	658
8 -	661
9 -	531
10 -	234
11 -	78



Dit is ook niet beter, maar juist slechter.

Ook nog een aantal andere combinaties geprobeerd, maar dat wordt niet echt beter.

Is er wellicht een mechanisme te verzinnen waardoor de 12 mogelijke posities eerlijker verdeeld worden. Beginnen met 3 willekeurige. De volgende 3 moeten anders zijn dan de vorige 3. De volgende drie moeten anders zijn dan de vorige 6 en de laatste drie moeten anders zijn dan de vorige 9. Maar dan moet je dat gaan bijhouden. Beetje zoals de boardselectie bij Barrelopaloza.

Kijken of het kan op basis van de RNGCounter1, FrameCounter en RNGCounter2. En dan meteen maar de drie gereserveerde adressen ermee vullen. Maar dan is dit altijd random (op basis van frames) en kan dus niet met een seed.

Dan dus een AND nemen met #0F levert getal tussen 0 en 15. Maar 12, 13, 14 en 15 niet nodig:
12 – 1100 / 13 – 1101 / 14 – 1110 / 15 – 1111.

Die mappen op een nieuw getal door de bits te flippen d.m.v. een AND:

RBGcounter1 - bovenste ladder level : AND 0111 (#F7): mappen op 4,5,6,7

FrameCounter – middelste ladder level: AND 1011 (#FB): mappen op 8,9,10,11

RNGCounter2 – onderste ladder level: AND 0011 (#F3): mappen op 0,1,2,3

Daardoor naar onderste level helemaal rechts en extra kans op ladder helemaal links. Daarna naar middelste level extra kans op ladder helemaal rechts en naar bovenste ladder een extra kans op een ladder in het midden. Kijken wat dat oplevert.

Nee dat kan niet. Want je hebt niet drie ladders nodig, maar je hebt 3x3 ladders nodig. Dus dat zijn er 9. Dat gaat niet hiermee, omdat de RNGCounter1, RNGCounter2 en FrameCounter niet wijzigen tijdens de uitvoer van de code.

Daarom gaan voor een methode waarin we wel de random waarden bepalen op basis van de code zoals we dat nu gemaakt hebben. Maar dan bij het bepalen van de drie ladder posities checken we deze waarden tegen een datastructuur waarin we de reeds gekozen posities aangegeven hebben. Als in gebruik dan de volgende positie nemen.

Maar allemaal heel ingewikkeld omdat het 3 keer moet, dus de code herbruikbaar voor verschillende geheugenranges. Hoe die onthouden met HL en IX erg lastig.

Ik denk dat ik hiermee ga stoppen. Of doorgaan met de huidige randomization en kijken wat dat oplevert qua random levels. Met het risico dat het niet lekker verdeeld en de moeite voor niets geweest is.

Eens kijken als we eens een stuk uitschrijven van de code van de huidige verdeling. Dus vanuit PowerShell een stuk laten draaien en dan de velden tekenen op basis van de resultaten. Wat is het resultaat hiervan?

0	1	2	3	4	5	6	7	8	9	10	11
			X		X					X	
			X	X			X				
		X	X		X						

0	1	2	3	4	5	6	7	8	9	10	11
X				X				X			
X				X	X						
			X	X		X					

0	1	2	3	4	5	6	7	8	9	10	11
X			X		X						
			X	X		X					
X	X	X									

0	1	2	3	4	5	6	7	8	9	10	11
					X		X			X	
			X	X		X					
X		X	X		X						

0	1	2	3	4	5	6	7	8	9	10	11
					X	X	X				
X		X	X								
			X	X	X						

0	1	2	3	4	5	6	7	8	9	10	11
				X				X		X	
			X		X			X			
		X			X					X	

0	1	2	3	4	5	6	7	8	9	10	11
			X			X			X		
X	X	X									
				X	X		X				

0	1	2	3	4	5	6	7	8	9	10	11
			X		X					X	
	X	X				X					
X	X						X				

Zomaar wat gepakt, maar dat levert best wel een gevarieerd resultaat op. Dus wellicht niet al te druk maken om de verdeling.

Dus nu kijken om de routine te maken die de drie geheugenplaatsen voor de unieke ladder posities te vullen. En dan draaien in de debugger en een aantal verschillende combinaties oopschrijven en kijken wat voor velden daar dan uitkomen.

Memory 1: #66A0	- loop counter for ladder level (outer loop)
Memory 2: #66A1	- loop counter for ladders (inner loop)
Memory 3: #66A2	- ladder level on which to skip broken ladder
Memory 4: #66A3	- ladder level Y-value
Memory 5: #66A4	- 1st ladder offset
Memory 6: #66A5	- 2nd ladder offset
Memory 7: #66A6	- 3rd ladder offset

Dus #66A4, #66A5 en #66A6 gebruiken.

Globaal:

Vul A met random waarde -> schrijf naar #66A4

Vul A met random waarde -> schrijf naar #66A5

Vul A met random waarde -> schrijf naar #66A6

Check #66A4 met #66A5

Check #66A4 met #66A6

Check #66A5 met #66A6

Anders drie verschillende waarden gevuld.

```

0AD2 DD21A466 LD IX,#66A4 ; set IX to index #66A4 – label AAAA
0AD6 CDA30A CALL #0AA3 ; get random number 0-11
0AD9 DD7700 LD (IX+#00),A ; store in #66A4
0ADC CDA30A CALL #0AA3 ; get random number 0-11
0ADF DD7701 LD (IX+#01),A ; store in #66A5
0AE2 CDA30A CALL #0AA3 ; get random number 0-11
0AE5 DD7702 LD (IX+#02),A ; store in #66A6
;
0AE8 DD7E00 LD A,(IX+#00) ; get #66A4
0AEB DD9601 SUB (IX+#01) ; subtract #66A5
0AEE 28E2 JR Z,#0AD2 ; if equal do again – jump to label AAAA
;
0AF0 DD7E00 LD A,(IX+#00) ; get #66A4
0AF3 DD9602 SUB (IX+#02) ; subtract #66A6
0AF6 28DA JR Z,#0AD2 ; if equal do again – jump to label AAAA
;
0AF8 DD7E01 LD A,(IX+#01) ; get #66A5
0AFB DD9602 SUB (IX+#02) ; subtract #66A6
0AFE 28D2 JR Z,#0AD2 ; if equal do again – jump to label AAAA
;
0B00 C9 RET

```

en:

```
19C2 CDD20A CALL #0AD2 ; call test rng code – call to label XXXX
```

Berekenen jumpafstanden JR Z operaties (inclusief beide opcodes)

30 – 1E inverse = E2

38 – 26 inverse = DA

46 – 2E inverse = D2

Lijkt goed te gaan. Zorgt er in ieder geval voor dat er altijd drie unieke waarden gekozen worden. Getallen voor zes schermen opgeschreven. Kijken wat dat oplevert.

0	1	2	3	4	5	6	7	8	9	10	11
X	X							X			
	X			X		X					
X			X	X							

0	1	2	3	4	5	6	7	8	9	10	11
						X	X		X		
		X		X			X				
		X		X						X	

0	1	2	3	4	5	6	7	8	9	10	11
		X	X	X							
	X						X	X			
		X		X			X				

0	1	2	3	4	5	6	7	8	9	10	11
			X	X		X					
				X		X	X				
		X			X			X			

0	1	2	3	4	5	6	7	8	9	10	11
	X	X			X						
X		X	X								
		X		X		X	X				

0	1	2	3	4	5	6	7	8	9	10	11
		X	X	X							
	X			X	X						
		X	X		X						

Hier mee doorgaan. En dan zien we wel hoe dit afloopt.

Wel nog de start bekijken. Anders krijg je steeds dezelfde reeks. En dat willen we niet. Dus initieel een random start voor de memory index vaststellen op basis van de dip switch instellingen en dan ook nog checken met een vaste waarde, of de roll-over aan het einde goed gaat. De default dip switch instellingen altijd een random waarde bepalen.

Maar dit kan altijd nog. Wellicht handiger om eerst het random opbouwen van de ladders toe te voegen. Dan kun je gaan kijken hoe de velden er uitzien en daarna de startwaarde aanpassen.

Nu versie v0.41 opgeslagen.

Nu dan kijken hoe de code in PowerShell omgezet kan worden in assembly code in de rom.

Als basis de Randomize Wizard v0.5.ps1 gebruiken.

Memory 1: #66A0	- loop counter for ladder level (outer loop)
Memory 2: #66A1	- loop counter for ladders (inner loop)
Memory 3: #66A2	- ladder level on which to skip broken ladder
Memory 4: #66A3	- ladder level Y-value
Memory 5: #66A4	- 1st ladder offset
Memory 6: #66A5	- 2nd ladder offset
Memory 7: #66A6	- 3rd ladder offset

De ladder definitie opbouwen in #66B0 en verder:

$$9 * 5 \text{ (ladders)} = 45 + 1 \text{ (afsluitcode)} = 46 = \#2E + \#66B0 = \#66DE.$$

HL register is de pointer naar de plek in de datastructuur.

```

0B01 3A1860      LD   A,RngTimer1    ; get random number for broken ladder skip
0B04 E603        AND  #03          ; random number between 0 and 3
0B06 FE03        CP   #03          ; A == 3 ?
0B08 2002        JR   NZ,#AAAA    ; no, skip next step
0B0A 3E01        LD   A,#01          ; A = 1
0B0C C601        ADD  A,#01          ; random number between 1 and 3
0B0E 32A266      LD   (#66A2),A    ; store in brokenladderskip
;
0B11 3E78        LD   A,#78          ; load A with start position top ladders
0B13 32A366      LD   (#66A3),A    ; store in ladderlevelypos
;
0B16 21B066      LD   HL,#66B0    ; load HL with start of datastructure
;
; Outer loop
;
0B19 3E03        LD   A,#03          ; repeat 3 times – values 3, 2 , 1
0B1B 32A066      LD   (#66A0),A    ; store in outerloopcounter
;
0B1E E5          PUSH  HL          ; save HL – label AAAA
0B1F DDE5        PUSH  IX          ; save IX
;
0B21 CDD20A      CALL  #0AD2    ; calculate 3 different random values and store
;
0B24 DDE1        POP   IX          ; restore IX
0B26 E1          POP   HL          ; restore HL
;
; Inner loop
;
0B27 3E03        LD   A,#03          ; repeat 3 times – values 3, 2 , 1
0B29 32A166      LD   (#66A1),A    ; store in innerloopcounter
;
0B2C 47          LD   B,A          ; store innerloopcounter in B – label BBBB
;
0B2D DD21A466    LD   IX,#66A4    ; load IX with 1stladderoffset
0B31 FE01        CP   #01          ; innerloopcounter = 01?
0B33 2808        JR   Z,#CCCC    ; yes, skip next steps – jump to label CCCC
;
0B35 DD23        INC   IX          ; load IX with 2ndladderoffset
0B37 FE02        CP   #02          ; innerloopcounter = 02?
0B39 2802        JR   Z,#CCCC    ; yes, skip next steps – jump to label CCCC
;
0B3B DD23        INC   IX          ; load IX with 3rdladderoffset
0B3D DD7E00      LD   A,(IX+#00)  ; load A with correct ladderoffset – label CCCC
0B40 57          LD   D,A          ; store correct ladderoffset in D
;
0B41 78          LD   A,B          ; restore innerloopcounter from B
0B42 FE01        CP   #01          ; innerloopcounter = 01 (last ladder)?

```

0B44	3AA066	LD	A,(#66A0)	; load A with outerloopcounter
0B47	47	LD	B,A	; store outerloopcounter in B
;				
0B48	200B	JR	NZ,DDDD	; no skip next steps – jump to label DDDD
;				
0B4A	3AA266	LD	A,(#66A2)	; load A with brokenladderskip
0B4D	90	SUB	B	; outerloopcounter = brokenladderskip?
0B4E	CA8B0B	JP	Z,#0B8B	; yes, skip this broken ladder – jump to label QQQQ
;				
0B51	3E01	LD	A,#01	; load A with 01 – laddertype broken
0B53	1802	JR	#EEEE	; skip next step – jump to label EEEE
;				
0B55	3E00	LD	A,#00	; load A with 00 – laddertype normal – label DDDD
;				
0B57	77	LD	(HL),A	; store laddertype in data structure – label EEEE
0B58	23	INC	HL	; set HL to next element
;				
0B59	7A	LD	A,D	; load A with ladderoffset
0B5A	3C	INC	A	; increment A
0B5B	CB27	SLA	A	; shift left 4 times – ladderoffset multiply #10
0B5D	CB27	SLA	A	; shift left
0B5F	CB27	SLA	A	; shift left
0B61	CB27	SLA	A	; shift left
0B63	4F	LD	C,A	; save multiplied ladderoffset in C
;				
0B64	78	LD	A,B	; restore outerloopcounter from B
0B65	FE02	CP	#02	; outerloopcounter = 2?
0B67	3E23	LD	A,#23	; set offset to left on segment
0B69	2002	JR	NZ,#FFFF	; no, skip next step – jump to label FFFF
0B6B	3E2B	LD	A,#2B	; set offset to right on segment
0B6D	81	ADD	A,C	; add multiplied ladderoffset – label FFFF
;				
0B6E	77	LD	(HL),A	; store xposition in data structure
0B6F	23	INC	HL	; set HL to next element
;				
0B70	47	LD	B,A	; store xposition in B
;				
0B71	7A	LD	A,D	; restore ladderoffset from D
0B72	FE07	CP	#07	; ladderoffset < 7
0B74	3AA366	LD	A,(#66A3)	; load A with ladderlevelypos
0B77	3004	JR	NC,#GGGG	; no, skip next steps – jump to label GGGG ;
;				
0B79	92	SUB	D	; A == A - D
0B7A	3D	DEC	A	; A == A - #01
0B7B	1803	JR	#HHHH	; skip next steps – jump to label HHHH
;				
0B7D	82	ADD	A,D	; A == A + D – label GGGG
0B7E	D60D	SUB	#0D	; A == A - #0D

```

;
0B80 77 LD (HL),A ; store ypositiontop in data structure – label HHHH
0B81 23 INC HL ; set HL to next element
;
0B82 C620 ADD #20 ; A == A + #20
0B84 4F LD C,A ; store ypositionbottom in C
;
0B85 78 LD A,B ; restore xposition
;
0B86 77 LD (HL),A ; store xposition in data structure
0B87 23 INC HL ; set HL to next element
;
0B88 79 LD A,C ; restore ypositionbottom
;
0B89 77 LD (HL),A ; store ypositionbottom in data structure
0B8A 23 INC HL ; set HL to next element
;
0B8B 3AA166 LD A,(#66A1) ; load A with innerloopcounter – label QQQQ
0B8E D601 SUB #01 ; decrease innerloopcounter
0B90 32A166 LD (#66A1),A ; store in innerloopcounter
0B93 C22C0B JP NZ,#0B2C ; repeat if not zero – jump to label BBBB
;
0B96 3AA366 LD A,(#66A3) ; load A with ladderlevelypos
0B99 C620 ADD A,#20 ; increase ladderlevelypos #20
0B9B 32A366 LD (#66A3),A ; store in ladderlevelypos
;
0B9E 3AA066 LD A,(#66A0) ; load A with outerloopcounter
0BA1 D601 SUB #01 ; decrease outerloopcounter
0BA3 32A066 LD (#66A0),A ; store in outerloopcounter
0BA6 C21E0B JP NZ,#0B1E ; repeat if not zero – jump to label AAAA
;
0BA9 3EAA LD A,#AA ; load A with #AA
0BAB 77 LD (HL),A ; store endmark in data structure
;
0BAC C9 RET ; return

```

En aanroep hiernaartoe invoegen na het tekenen van het normale barrels board.

Eerst testen met dummy aanroep zodat via de debugger gekeken kan worden of de code inderdaad via dubbele loop doorlopen wordt, of de waarden goed gevuld worden en de data structuur opgebouwd wordt.

```
19C2 000000 NOP,NOP,NOP ; three original NOP's
```

De schermen worden hier getekend:

```
; other screens return here

0CC6 CDA70D CALL #0DA7 ; draw the screen
```

Aanpassen:

```
0CC6 CDC70B CALL #0BC7 ; draw the screen
```

en:

```
0BC7 CDA70D CALL #0DA7 ; draw the screen
;
0BCA 3A2762 LD A,(#6227) ; load A with screen number
0BCD FE01 CP #01 ; is this the girders?
0BCF C0 RET NZ ; no, return
;
0BD0 CD010B CALL #0B01 ; build datastructure random ladders
0BD3 11B066 LD DE,#66B0 ; load DE with start of datastructure random ladders
0BD6 CDA70D CALL #0DA7 ; draw the random ladders
0BD9 C9 RET ; return
```

Is nu helemaal goed.

[Nu versie v0.42 opgeslagen.](#)

De ladders worden nu alleen nog maar getekend. Nu nog ervoor zorgen dat de ladders ook beklimmen kunnen worden.

```
245D CABAOB JP Z,#0BBA ; if girders, jump to additional code
```

en:

```
0BBA CD7124 CALL #2471 ; activate default ladders
;
0BBD 21B066 LD HL,#66B0 ; load HL with start of data structure random ladders
0BC0 DD210763 LD IX,#6307 ; #6307 is used for ladder positions?
0BC4 C37524 JP #2475 ; activate random ladders
```

Dit gaat niet goed omdat op dit moment de structuur op 66B0 alweer gewist is.

Kijken of je de ladders meteen na het vullen van de structuur kan activeren. En dan de default ladders pas bij de originele code.

```
0CC6 CDBDOB CALL #0BBD ; draw the screen
```

en:

```

0BBD CDA70D      CALL #0DA7      ; draw the screen
;
0BC0 3A2762       LD   A,(#6227)  ; load A with screen number
0BC3 FE01         CP   #01        ; is this the girders?
0BC5 C0           RET  NZ        ; no, return
;
0BC6 CD010B       CALL #0B01      ; build datastructure random ladders
0BC9 11B066       LD   DE,#66B0  ; load DE with start of datastructure random ladders
0BCC CDA70D       CALL #0DA7      ; draw the random ladders
;
0BCF 21B066       LD   HL,#66B0  ; load HL with start of data structure random ladders
0BD2 DD210063     LD   IX,#6300  ; #6300 is used for ladder positions?
0BD6 CD7524       CALL #2475      ; activate random ladders
0BD9 C9           RET           ; return

```

en dan even om te testen de default ladders niet laten toevoegen...

```
245D C8          RET  Z
```

Gaat ook niet goed omdat waarschijnlijk de structuur vanaf 6300 in de tussentijd weer gewist wordt.

Activeren na default ladders gaat niet omdat dan 66B0 structuur weer gewist is.

Activeren meteen na opbouwen 66B0 structuur wordt weer gereset.

Dus hoe krijg je dat dan goed? Schrijven in een stuk dat niet gewist wordt?

In het stuk code van #0F56 t/m #0F79 wordt een heleboel gewist #6280 t/m #6B00. Dus ook het stuk van #66B0 waar de data van de random trappen staat. Wellicht deze data niet wissen? Hele stuk van 66B0 t/m 6B00 is helemaal leeg. Eens kijken met debuggen van een standaard rom of hier tijdens het spelen wel data neergezet wordt. Zo niet, dan dit hele deel uitsluiten van de wisactie?

Eerste stuk vanaf #6B00 lijkt niet gebruikt te worden? En wordt niet gewist? Proberen daar de structuur aan te maken? En dan dat als basis voor tekenen en activeren?

Stuk vanaf #6B00 wordt niet gebruikt. Althans zo lijkt het wanneer de originele dkong.zip gebruikt wordt. Dus daar de datastructuur opbouwen.

Daartoe versie 0.42 terugzetten en aanpassen:

```
0B16 21006B       LD   HL,#6B00  ; load HL with start of datastructure
```

en:

```
0BD3 11006B       LD   DE,#6B00  ; load DE with start of datastructure random ladders
```

Dan moet het getekend worden, èn dan kijken in debugger of de datastructuur blijft bestaan.

Ja dat gaat goed.

Dan nu de code voor het activeren van de trappen opnieuw toepassen, maar dan op basis van de datastructuur op #6B00.

```
245D CABE0B      JP      Z,#0BBE      ; if girders, jump to additional code
```

en:

```
0BBE CD7124      CALL   #2471      ; activate default ladders  
;  
0BC1 21006B      LD      HL,#006B      ; load HL with start of data structure random ladders  
0BC0 DD210763      LD      IX,#6307      ; #6307 is used for ladder positions?  
0BC4 C37524      JP      #2475      ; activate random ladders
```

Ja dat gaat goed. Trappen te beklimmen en barrel board is speelbaar zo.

Nu versie 0.43 opgeslagen.

Nog doen:

- Set start randomizer seed at start game, based on FrameCounter
- Test with seed that overflows #3FFF
- Check occasional red barrel later on in the barrel board

Wanneer een nieuwe game gestart wordt, wordt het scherm gewist:

```
; arrive from #0701 when GameMode2 == 6  
; clears screen and sprites, check for intro screen to run  
  
0A63 DF      RST   #18      ; count down WaitTimerMSB and only continue here if == 0, else return to higher sub.  
0A64 CD7408      CALL  #0874      ; clears the screen and sprites  
0A67 210960      LD    HL,WaitTimerMSB ; load HL with timer  
0A6A 3601      LD    (HL),#01      ; set timer to 1  
0A6C 2C      INC   L       ; HL := GameMode2  
0A6D 34      INC   (HL)      ; increase game mode2 to 7  
0A6E 112C62      LD    DE,#622C      ; load DE with game start flag address  
0A71 1A      LD    A,(DE)      ; load A with game start flag  
0A72 A7      AND   A       ; is this game just beginning?  
0A73 C0      RET   NZ      ; yes, return  
  
0A74 34      INC   (HL)      ; else increase game mode2 to 8 - skip kong intro to begin  
0A75 C9      RET           ; return
```

Daar eventueel het random maken van de seed aan toevoegen?

Eerst testen met seed vlak onder #3FFF en dan kijken of de overflow goed gaat.

Aanpassen:

```
0A67 CD823F      CALL  #3F82      ; call to additional code
```

en:

```

3F82 213262      LD   HL,#6232      ; load HL with startaddress for random index
3F85 363F         LD   (HL),#3F      ; load random index H with #3F
3F87 23           INC  HL
3F88 36F0         LD   (HL),#F0      ; load random index L with #F0
;
3F8A 210960       LD   HL,WaitTimerMSB ; load HL with timer
3F8D C9           RET

```

Ja, de overflow gaat goed.

Nu zo maken dat er een willekeurige waarde ingevuld wordt:

```

3F82 213262      LD   HL,#6232      ; load HL with startaddress for random index
;
3F85 3A1860       LD   A,(RngTimer1) ; load A with random value (#00-#FF)
3F88 E63F         AND  #3F          ; A is random value (#00-#3F)
3F8A 77           LD   (HL),A      ; set random index H
3F8B 23           INC  HL
3F8C 3A1A60       LD   A,(FrameCounter) ; load A with random value (#00-#FF)
3F8F 77           LD   (HL),A      ; set random index L
;
3F90 210960       LD   HL,WaitTimerMSB ; load HL with timer
3F93 C9           RET

```

Ja dat gaat goed.

Nu versie v0.44 opgeslagen.

In plaats van d HIGH SCORE weer WIZARDRY tonen. Was al eerder gedaan, maar door nieuwe rom moet dat nogmaals:

Op #36B2 staat de HIGH SCORE tekst:

#36B2: 80 76 18 19 17 18 10 23 13 1F 22 15 3F

Veranderen naar:

#36B2: 80 76 10 27 19 2A 11 22 14 22 29 10 3F

Dat is goed. Nu nog blauw maken. Dat betekent dat dat deel van het scherm dezelfde kleurpalette moet krijgen als het deel waar de L= levelaanduiding staat.

Dit zit in het bestand v-5e.bpr:

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

Door het aanpassen van deze twee waarden wordt de titel bovenin donkerblauw. Dat past heel goed bij de kleuren van de jumpman wizard sprites.

Dan starten met het titel scherm. 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 C3943F JP #3F94 ; jump to additional code – jump to label AAAA
01F0 00 NOP

```

En (in het stukje van de developers message die vrijgegeven is):

```

3F94 3E06 LD A,#06 ; A := 6 – label AAAA
3F96 320A60 LD (GameMode2),A ; store into game mode 2
3F99 C3F101 JP #01F1 ; jump back

```

Dat gaat nu goed.

Nu versie v0.45 opgeslagen.

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.

Nu versie v0.46 opgeslagen.

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:

© 1981-2021 NINTENDO

5C 77 49 4A 10 01 09 08 01 2C 02 00 02 01 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:

WIZARDRY

6E 76 27 19 2A 11 22 14 22 29 3F

En de verwijzing ernaar 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

Gaat allemaal goed.

Nu versie v0.47 opgeslagen.

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 (V1.01)

en

#364D: 3E 37

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

```
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,#690B ; 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: C3233F JP #3F23 ; jump to additional code to display version
3F23: 110103 LD DE,#0301 ; load task data for text "V1.00"
3F26: CD9F30 CALL #309F ; insert task to draw text
3F29: C31F08 JP #081F ; jump back
```

Nu versie v0.48 opgeslagen.

Nu het How High scherm aanpassen naar één vaste DK en zonder de meter aanduidingen. En daarmee extra ruimte vrijmaken.

De tekst HOW HIGH CAN YOU GET ? veranderen in LET THE BARRELS ROLL !

Net zo lang: Aanpassen op #36CC:

5E 77 1C 15 24 10 24 18 15 10 12 11 22 22 15 1C 23 10 22 1F 1C 1C 10 38

Dat is goed. Nu ervoor zorgen dat de Donkey Kong maar één keer getekend wordt.

Eerst hele stuk verwijderen: #OBFD t/m #0C81.

Dan volgende toevoegen:

```
OBFD 21BC75 LD HL,#75BC ; load HL with screen location start for goofy kong
;
0C00 0E50 LD C,#50 ; C := #50 = start graphic for goofy kong
;
0C02 71 LD (HL),C ; draw part of goofy kong
0C03 0C INC C ; next graphic
0C04 2B DEC HL ; next screen location
0C05 71 LD (HL),C ; draw part of goofy kong
0C06 0C INC C ; next graphic
0C07 2B DEC HL ; next screen location
```

```

OC08 71      LD   (HL),C    ; draw part of goofy kong
OC09 0C      INC  C        ; next graphic
OC0A 2B      DEC  HL       ; next screen location
OC0B 71      LD   (HL),C    ; draw part of goofy kong
OC0C 79      LD   A,C       ; load A with graphic number
OC0D FE67      CP   #67      ; == #67 ? (are we done?)
OC0F CA1A0C  JP   Z,#0C1A  ; yes, skip next 4 steps
;

OC12 0C      INC  C        ; next C
OC13 112300  LD   DE,#0023  ; load DE with offset
OC16 19      ADD  HL,DE    ; add to screen location
OC17 C3020C  JP   #0C02    ; loop again
OC1A C3820C  JP   #0C82    ; continue after freed space

```

En dat is goed.

De tekst HOW HIGH CAN YOU GET ? veranderen in USE YOUR MAGIC SPELL !

Net zo lang: Aanpassen op #36CC:

5E 77 25 23 15 10 29 1F 25 22 10 1D 11 17 19 13 10 23 20 15 1C 1C 10 38

Nu versie v0.49 opgeslagen.

Niet zo heel veel ruimte vrijgemaakt, maar wel iets. Kijken hoever we komen met het weer inbrengen van de magic spell: alleen het freezeen van de fireballs.

Afhandeling indrukken knop P2 invoegen in een vrij slot van de main:

```
19C2 CD1D0C      CALL  #0C1D      ; call additional code – call label AAAA
```

en:

```

OC1D 3A8E63      LD   A,(#638E)  ; load a with spellcounter2
OC20 FE00      CP   #00      ; A == 0?
OC22 2008      JR   NZ,0C2C  ; no, spell is already activated – jump to label CCCC
;
OC24 47      LD   B,A      ; save A to B
OC25 3A007D  LD   A,(IN2)  ; load A with IN2
OC28 FE08      CP   #08      ; is P2 button pressed?
OC2A C0      RET  NZ      ; no, do not activate spell – return
;
OC2B 78      LD   A,B      ; restore A from B
;
OC2C FEFF      CP   #FF      ; A == 255? – label CCCC
OC2E C8      RET  Z       ; yes, spell is already finished - return
;
; spell is active

```

```

;
0C2F 3C           INC   A           ; increment A
0C30 328E63       LD    (#638E),A ; store in spellcounter2
0C33 C9           RET              ; return

```

Dit zou na indrukken P2 de spellcounter (#638E) moeten laten oplopen. Checken met de debugger. Ja dat gaat inderdaad goed. P2 kan eenmalig gebruikt worden en dan loopt de spellcounter op naar #FF. En elk spel opnieuw wordt de spellcounter weer op 0 gezet.

Dan ervoor zorgen dat de fireballs tijdelijk bevrioren worden na indrukken van P2.

```

198C CD340C       CALL  #0C34      ; update fires if needed via P2 spell control
en

0C34 3A8E63       LD    A,(#638E)  ; load a with spellcounter2
0C37 FE00         CP    #00        ; A == 00 ?
0C39 2805         JR    Z,#3F44   ; yes, spell is not active – jump to label AAAA
;
0C3B FEFF         CP    #FF        ; A == 255 ?
0C3D 2801         JR    Z,#3F44   ; yes, spell is already used – jump to label AAAA
;
0C3F C9           RET              ; spell is active – return without update fires
;
0C40 CDED30       CALL  #30ED      ; update fires if needed – label AAAA
0C43 C9           RET              ; return

```

Ja dat werkt.

Nu versie v0.50 opgeslagen.

Nu nog de grafische weergave van de gebruikte spell ook terugbrengen. Dat was eerst aparte code, maar dat integreren met de code die de P2 knop checkt.

```

0C1D 210275       LD    HL,#7502   ; load HL with screen location for spell info
0C20 3623         LD    (HL),#23   ; draw "S"
0C22 21E274       LD    HL,#74E2   ; next location
0C25 3634         LD    (HL),#34   ; draw "="
0C27 21C274       LD    HL,#74C2   ; next location
;
0C2A 3A8E63       LD    A,(#638E)  ; load a with spellcounter2
0C2D FE00         CP    #00        ; A == 0?
0C2F 200E         JR    NZ,0C3F   ; no, spell is already activated – jump to label CCCC
;
0C31 47           LD    B,A       ; save A to B
;
0C32 3E02         LD    A,#02      ; set character spell2 not used
0C34 77           LD    (HL),A   ; draw character spell2

```

```

;
0C35 3A007D      LD   A,(IN2)      ; load A with IN2
0C38 FE08        CP   #08         ; is P2 button pressed?
0C3A C0          RET  NZ          ; no, do not activate spell – return
;
0C3B 3E35        LD   A,#35       ; set character spell2 used
0C3D 77          LD   (HL),A     ; draw character spell2
;
0C3E 78          LD   A,B         ; restore A from B
;
0C3F FEFF        CP   #FF         ; A == 255? – label CCCC
0C41 C8          RET  Z          ; yes, spell is already finished - return
;
; spell is active
;
0C42 3C          INC  A          ; increment A
0C43 328E63      LD   (#638E),A  ; store in spellcounter2
0C46 C9          RET
;
```

Nieuwe characters:

```

0C32 3EBF        LD   A,#BF       ; set character spell2 not used
0C3B 3EBD        LD   A,#BD       ; set character spell2 used
;
```

Sprites aangepast. Spell 2 is a fireball.

En hierdoor de code voor het stopzetten van de fireballs verplaatst:

```
198C CD470C      CALL #0C47      ; update fires if needed via P2 spell control
```

en

```

0C47 3A8E63      LD   A,(#638E)  ; load a with spellcounter2
0C4A FE00        CP   #00         ; A == 00 ?
0C4C 2805        JR   Z,#0C53    ; yes, spell is not active – jump to label AAAA
;
0C4E FEFF        CP   #FF         ; A == 255 ?
0C50 2801        JR   Z,#0C53    ; yes, spell is already used – jump to label AAAA
;
0C52 C9          RET
;
0C53 CDED30      CALL #30ED    ; update fires if needed – label AAAA
0C56 C9          RET
;
```

Ja dat gaat nu goed.

Nu versie v0.51 opgeslagen.

De kleur van S= nog aanpassen. Is nu nog wit. Rood maken. In het bestand v-5e.bpr:

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

Nee dat gaat niet goed. Dan is de fireball ook ineens de verkeerde kleur.

Hoe is dat dan in de vorige versie gedaan? Die heeft het palette niet aangepast maar laat wel rode letters zien. Alsof dat letters in een andere kleur zijn. En dat klopt ook. De S heeft code #BA en = heeft code #BB.

Dus aanpassen:

OC20	36BA	LD	(HL),#BA	; draw "S"
OC25	36BB	LD	(HL),#BB	; draw "="

De spell duurt nog wel erg lang. Wellicht iets korter maken.

0C3F	FEC0	CP	#C0	; A == 192 ? – label CCCC
0C4E	FEC0	CP	#C0	; A == 192 ?

Nu versie v0.52 opgeslagen.

De extra fireball niet op de girder onder Kong (met de hamer links) laten vrijkomen. Want dan gaat de fireball toch vrij vaak naar boven en komt niet meer naar beneden.

Aanpassen

3F66	FE80	CP	#80	; is Y-position lower than #80?
3F68	3812	JR	C,#BBBB	; no, do not deploy fireball -jump to label BBBB

Nu versie v0.53 opgeslagen.

Alleen nog het dashboard weghalen wanneer het level beëindigd wordt of jumpman dood gaat.

Wanneer een board klaar is, dan wordt de volgende code uitgevoerd:

```
; 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
```

Tegelijkertijd met het wissen van de sprites moet ook het spell dashboard gewist worden. Dus de aanroep in 1615 veranderen in een andere call en daar als eerste de call om de sprites te wissen en dan de additionele code om het spell dashboard te wissen.

Wissen zou kunnen door een tekst te plaatsen met 5 spaties. Er zijn nog een aantal tekst sloten vrij. Slot 6 is nog vrij.

Aanpassen

```
1615 CD570C CALL #0C57 ; clear sprites and clear spell dashboard
```

en:

```
0C57 CDBD30 CALL #30BD ; clear sprites – label AAAA
0C5A 110603 LD DE,#0306 ; load task data to draw " " – task 3, slot 6
0C5D CD9F30 CALL #309F ; insert task to draw text
0C60 C9 RET ; return
```

en:

```
3657 61 0C ; position of text string
```

en:

```
0C61 02 75 10 10 10 10 10 3F ; pos pos space space space space space end
```

Ja dat gaat goed wanneer einde level. Maar het dashboard blijft ook staan wanneer jumpman afgaat. Dan worden de sprites blijkbaar ook gewist. Die call ook aanpassen nog.

Dat wordt hier gedaan:

```
12A3 CDBD30 CALL #30BD ; clear sprites ?
12A6 3E03 LD A,#03 ; load A with duration of sound
12A8 328860 LD (#6088),A ; play death sound
12AB C9 RET ; return
```

Dus de call op #12A3 aanpassen naar een call naar 3F87.

Aanpassen:

```
12A3 CD570C      CALL #0C57      ; call additional code – call label AAAA
```

Ja dat gaat ook goed.

Nu versie v0.54 opgeslagen.

Eerst wat opschoonwerk doen:

- Verwijderen player 2 functionaliteit:
 - Aantal credits in scherm
 - Drukken voor P2
 - Afhandeling player 2 en wisselen player 2
- Oplossen additionele red wild barrel
- Mogelijk probleem met broken ladder op bepaalde positie
- Oplossen wissen van Paulines hoed.

Paulines hoed wordt gewist doordat in de barrels stage de bovenste sport van de DK ladders gewist worden. Dat wordt blijkbaar gedaan door 3 blanke sprites te gebruiken.

```
; draw 3 black sprites above the top kongs ladder
; effect to erase the 2 girders at the top of kong's ladder

0FB5 21006A LD HL,#6A00      ; else load HL sprite RAM - used for blank space sprite
0FB8 3E4F LD A,#4F          ; A := #4F = X position of this sprite
0FBA 0603 LD B,#03          ; For B = 1 to 3

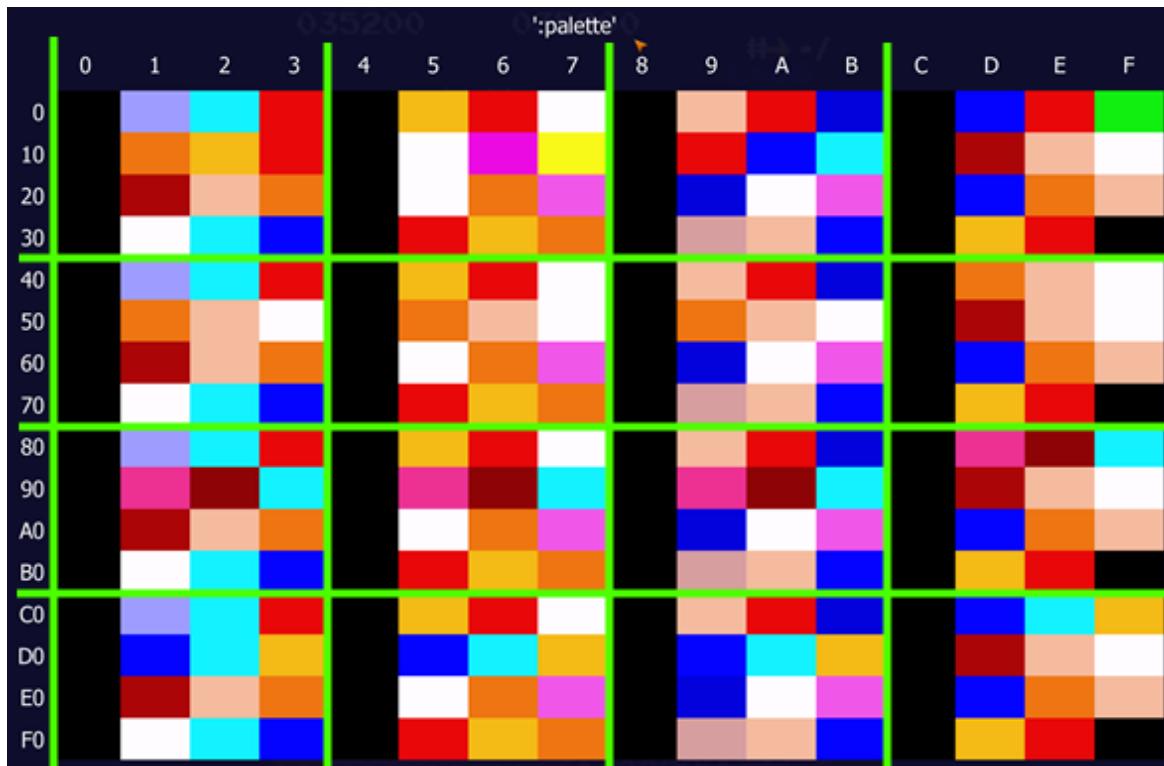
0FBC 77 LD (HL),A          ; set the sprite X position
0FBD 2C INC L              ; next address = sprite type
0FBE 363A LD (HL),#3A        ; set sprite type as blank square
0FC0 2C INC L              ; next address = sprite color
0FC1 360F LD (HL),#0F        ; set color to black
0FC3 2C INC L              ; next address = sprite Y position
0FC4 3618 LD (HL),#18        ; set sprite Y position to #18
0FC6 2C INC L              ; next memory
0FC7 C610 ADD A,#10         ; A := A + #10 to adjust for next X position
0FC9 10F1 DJNZ #0FBC        ; Next B

0FCB 79 LD A,C              ; load A with screen number
0FCC EF RST #28             ; jump depending on the screen
```

Als je dit terugbrengt tot 2 sprites (0FBA: 0602 LD B,#02) dan wordt de hoed van Pauline wel goed getekend, maar dan zie je DK ‘vervormen’ wanneer hij het beeld uit gaat met Pauline onder de arm.

Misschien interessanter om Pauline dezelfde hoed te geven als jumpman en haar jurk ook van sterren te voorzien. En dan het palette aan te passen zodat het een donkergrone jurk en hoed worden.

De sprites zijn aangepast. Nu nog het kleurenpalette aanpassen.



Dit zijn de vier kleurenpaletten voor de verschillende schermen. Je ziet de kleuren van Pauline op #25-#27, #65-#67, #A5-#A7 en #E5-#E7.

De kleurencombinatie van Pauline wordt verder nergens meer gebruikt. Wel eerst voor de bonus items, maar die zijn al van kleur veranderd.

De roze kleur staat op #27, #67, #A7 en #E7. Die allemaal veranderen in een andere kleur. Om te testen even de groene kleur van #0F kopiëren.

Dat moet in de files c-2j.bpr (#0E) en c-2k.bpr (#03). Deze waarden neerzetten in de #27, #67, #A7 en #E7 in de betreffende files.

Ja dat werkt, maar dat heeft alleen effect op de bovenkant (hoofd), maar niet op de onderkant (jurk). Dat is blijkbaar een andere spritekleur (wit, roze, blauw). Dat is de roze in #2B, #6B, #AB en #EB. Ook die aanpassen. Ja dat gaat goed.

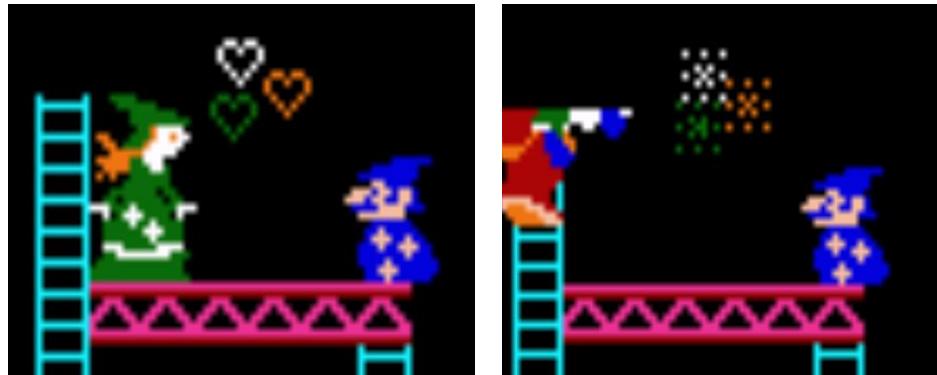
Alleen nog wel een donkerder kleur groen maken. Kan door in de c-2j.bpr #0E in #0F te veranderen.

Ja is gedaan en de sprites zijn ietwat aangepast.

Nu versie v0.55 opgeslagen.

Toch niet helemaal waar: de heart sprite is wel roze en gebruikt dus dezelfde sprite color. Oplossen door een ander symbool/symbolen maar dan in de nieuwe kleuren. Liefste iets wat met tovenaars te maken heeft. Of kleine hartjes in groen en wit door elkaar?

Drie hartjes in groen, wit en oranje en die dan uiteenspatten. Ziet er zo uit:



Nu versie v0.56 opgeslagen.

Testen additionele rode barrels die later in het spel gegeven worden. Nu net als laatste een red wild barrel gedeployed:



Memory: Zilog Z80 ':maincpu' program space memory																
Debug Options								zilog z80 ':maincpu' program space me								
6700	01	00	02	CB	00	90	19	15	0B	02	02	00	00	00	03	03
6710	01	00	00	23	05	00	00	AE	00	D1	00	00	00	00	00	#.....
6720	01	00	04	AB	00	6E	5F	95	0B	02	02	00	00	00	03	03
6730	FF	00	00	21	09	00	00	70	00	00	00	00	00	00	00	00
6740	01	00	04	C7	00	B0	4D	15	0B	02	02	00	00	00	03	01
6750	FF	00	00	23	09	00	00	AE	00	00	00	00	00	00	00	#.....
6760	01	00	04	AD	00	AE	4D	15	8B	02	02	00	00	00	03	03
6770	FF	00	00	23	09	00	00	DO	00	00	00	00	00	00	00	00
6780	01	00	04	3D	00	71	5F	95	0B	02	02	00	00	00	03	02
6790	FF	00	00	21	09	00	00	8D	00	00	00	00	00	00	00	00
67A0	01	00	02	C6	00	54	00	15	0B	02	02	00	00	00	00	02
67B0	00	00	00	00	00	00	6E	00	00	00	00	00	00	00	00	n.....
67C0	01	01	00	2C	02	5A	48	1B	0F	02	02	00	00	00	00	ZH.....
67D0	FF	36	00	00	13	01	00	00	4F	00	00	00	00	00	00	00
67E0	02	00	00	00	00	00	15	0B	02	02	00	00	00	00	00	00
67F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Memory: Zilog Z80 ':maincpu' program space memory																
Debug Options								zilog z80 ':maincpu' program space me								
6980	CB	15	0B	90	AB	95	0B	6E	C7	15	0B	B0	AD	15	8B	AE
6990	3D	95	0B	71	C6	15	0B	54	2C	1B	0F	5A	00	00	00	00
69A0	00	00	00	00	00	00	00	00	1E	18	0B	4B	14	18	0B	4B

X en Y posities zijn: #03 en #05: CB,90 – AB,6E – C7,B0 – AD,AE – 3D,71 – C6,54 – 2C,5A. De rode lijkt de derde van boven te zijn. Dat zou dan die van #6720 zijn?

Nog niet veel in te ontdekken.

Blijkbaar is positie #08 de kleur en de rode barrel is kleur #0F. Dan zou het die van #67C0 zijn.

Moet bewust op rood gezet worden, want wanneer een fireball spawned wordt, wordt de barrel al op blauw gezet. Dus kijken naar de voorwaarden voor het veranderen in een red barrel.

Wordt er een fireball spawned wanneer een extra red barrel gegooied wordt? Zou wel moeten. Maar welke fireball is dat dan? De code doet standaard fireball 1 (#6400) activeren. Dus als die al actief is, wordt deze 'verplaatst' naar de plek waar de barrel de girder raakt. Als deze niet meer actief is (na hamer), dan waarschijnlijk een normale spawn. Maar hoe testen? Het vindt namelijk niet zo vaak plaats. Spelen en dan eventueel save state maken wanneer het optreedt.

Of een extra check bij rood maken op eerste fase in barrel board. Maar hoe doe je dat? Ja dat kan door te checken of de oil can al ontstoken is. Bij de eerste blue barrel is dat niet het geval. Dus een dergelijke check tussenvoegen bij de code die van de eerste blue barrel een red barrel maakt.

Dit is de code die de eerste barrel rood maakt:

3F4B	3A8263	LD	A,(#6382)	; load barrel crazy and color status – label AAAA
3F4E	FE81	CP	#81	; is this the first crazy/blue barrel?
3F50	2805	JR	Z,#BBBB	; yes, set color to red – jump to label BBBB

```

;
3F52 DD36080C LD (IX+#08),#0C ; set sprite color to blue
3F56 C9 RET ; return
;
3F57 DD36080F LD (IX+#08),#0F ; set sprite color to red
3F5B C9 RET ; return

```

En dit is de check of oil can al aangestoken is.

```

2178 3A4863 LD A,(#6348) ; get status of the oil can fire
217B A7 AND A ; is the fire lit ?
217C CAB221 JP Z,#21B2 ; no, always take ladders before oil is lit

```

Dat als extra check toevoegen.

```

3F4B 3A8263 LD A,(#6382) ; load barrel crazy and color status
3F4E C3690C JP #0C69 ; jump to additional code – jump to label AAAA
3F51 00 NOP
3F4E FE81 CP #81 ; is this the first crazy/blue barrel?
3F50 2805 JR Z,#BBBB ; yes, set color to red – jump to label BBBB
;
3F52 DD36080C LD (IX+#08),#0C ; set sprite color to blue - label BBBB
3F56 C9 RET ; return
;
3F57 DD36080F LD (IX+#08),#0F ; set sprite color to red - label CCCC
3F5B C9 RET ; return

```

en:

```

0C69 FE81 CP #81 ; is this the first crazy/blue barrel? – label AAAA
0C6B C2523F JP NZ,#3F52 ; no, set color to blue – jump to label BBBB
;
0C6E 3A4863 LD A,(#6348) ; get status of the oil can fire
0C71 A7 AND A ; is the fire lit ?
0C72 C2523F JP NZ,#3F52 ; yes, set color to blue – jump to label BBBB
0C75 C3573F JP #3F57 ; no, set color to red – jump to label CCCC

```

Lijkt goed te gaan. Heel veel spelen en kijken of er nog later rode barrels komen. Nog niet gehad.

Nu versie v0.57 opgeslagen.

Nu kijken of we jumpman een baard kunnen geven. De sprite gebruikt nu maar twee kleuren en de rood wordt niet gebruikt. De rood omzetten naar wit en dan een baard tekenen.

In de palette files de #0A, #4A, #8A en #CA gelijk maken aan de kleur wit in #2A. Ja dat gaat goed. En dan de sprites tekenen met baard.

Ja dat is goed gelukt.

Nu versie v0.58 opgeslagen.

De 2 player functionaliteit verwijderen en daarmee vrije ruimte maken.

Eerste stuk is gebaseerd op gamemode1.

Als de game nu opgestart wordt, wordt in het high score scherm de optie getoond met keuze 1 player of 2 players. De teksten daarvan aanpassen. Schermen gekopieerd uit DUEL design.



De regel met 2 2 weghalen en het scherm met PUSH 1 OR 2 PLAYERS BUTTON niet tonen.

De regel met 2 2 wordt hier getekend:

```
07A3 ED5B2260 LD DE,(CoinsPerCredit)      ; D := CoinsPer2Credits; E := CoinsPerCredit
07A7 216C75 LD HL,#756C      ; load HL with screen RAM location
07AA CDAD07 CALL #07AD      ; run this sub below twice

07AD 73 LD (HL),E      ; draw to screen number of coins needed for 1 player game
07AE 23 INC HL      ;
07AF 23 INC HL      ; next screen location 2 rows down
07B0 72 LD (HL),D      ; draw to screen number of coins neeeded for 2 player game
07B1 7A LD A,D      ; A := D
07B2 D60A SUB #0A      ; subtract #A (10 decimal). result == 0 ?
07B4 C2BC07 JP NZ,#07BC      ; no, skip next 3 steps

07B7 77 LD (HL),A      ; else draw this zero to screen
07B8 3C INC A      ; increase A, A := 1 now
07B9 328E75 LD (#758E),A      ; draw 1 to screen in front of the zero, so it draws "10" credits needed for 2 players

07BC 110102 LD DE,#0201      ; D := 2, E := 1, used for next loop for 1 player and 2 players
07BF 218C76 LD HL,#768C      ; set screen location to draw for next loop if needed
07C2 C9 RET      ; return
```

Voldoende om de #07B0 NOP te maken?

Ja dat werkt.

De teksten voor 1 player en 2 player worden hier getoond:

```
08D5 0604 LD B,#04 ; B := 4 = 0100 binary
08D7 1E09 LD E,#09 ; E := 9 , code for "ONLY 1 PLAYER BUTTON"
08D9 3A0160 LD A,(NumCredits) ; load A with number of credits
08DC FE01 CP #01 ; == 1 ?
08DE CAE408 JP Z,#08E4 ; yes, skip next 2 steps

08E1 060C LD B,#0C ; B := #0C = 1100 binary
08E3 1C INC E ; E := #0A, code for "1 OR 2 PLAYERS BUTTON"

08E4 3A1A60 LD A,(FrameCounter) ; load A with # Timer constantly counts down from FF to 00
08E7 E607 AND #07 ; mask bits. zero ?
08E9 C2F308 JP NZ,#08F3 ; no, skip next 3 steps

08EC 7B LD A,E ; yes, load A with E for code of text to draw, for buttons to press to start
08ED CDE905 CALL #05E9 ; draw text to screen
08F0 CD1606 CALL #0616 ; draw credits on screen
```

De tweede tekst weghalen door op #08DE een JP te doen i.p.v. JP Z.

Ja dat werkt.

En dan nu nog ervoor zorgen dat er geen 2 player game opgestart kan worden.

Dat wordt hier afgehandeld:

```
; jump from #08B5 when GameMode2 == 1

08F8 CDD508 CALL #08D5 ; draws press player buttons and loads A with IN2, masked by possible player numbers
08FB FE04 CP #04 ; is the player 1 button pressed ?
08FD CA0609 JP Z,#0906 ; yes, skip ahead

0900 FE08 CP #08 ; is the player 2 button pressed ?
0902 CA1909 JP Z,#0919 ; yes, skip ahead

0905 C9 RET ; return to #00D2
```

Dus geen two -player meer starten wanneer #0900 t/m #0904 in NOPS veranderd worden.

Dat klopt. Nu geen two-player mode meer.

Nu versie v0.59 opgeslagen.

En dan nu ruimte vrijgeven op de plakken waar afhandeling van de two-player zit.

De two-player start kan vrijgegeven worden: #0919 t/m #0937.

Daarna wordt de keuze op verschillende plekken gemaakt. Steeds op basis van de variabelen TwoPlayerGame, PlayerTurnA en PlayerTurnB. Dus daar op zoeken en de aanpassingen doen.

Eerst kijken naar de TwoPlayerGame variabele.

Wordt hier gebruikt:

```
0331 3A0F60 LD A,(TwoPlayerGame) ; load A with # of players in game
0334 A7 AND A ; is this a 1 player game?
0335 C8 RET Z ; yes, return

0336 3A0D60 LD A,(PlayerTurnA) ; Load current player #
0339 EE01 XOR #01 ; change player from 1 to 2 or from 2 to 1
033B CD4703 CALL #0347 ; Loads HL with location for score (either player 1 or 2)

033E 3C INC A ; increase A, now it has the number of the player
033F 77 LD (HL),A ; draw player number on screen
0340 19 ADD HL,DE ; next column
0341 3625 LD (HL),#25 ; draw "U" on screen
0343 19 ADD HL,DE ; next column
0344 3620 LD (HL),#20 ; draw "P" on screen
0346 C9 RET ; return
```

Hier is niets te winnen.

Wordt hier gebruikt:

```
079B 3A0F60 LD A,(TwoPlayerGame) ; load A with number of players in game
079E FE01 CP #01 ; 2 player game?
07A0 CCEE09 CALL Z,#09EE ; yes, skip ahead to handle
```

Hiermee wordt de 2UP getekend:

```
09EE 3E02 LD A,#02 ; load A with "2"
09F0 32E074 LD (#74E0),A ; write to screen
09F3 3E25 LD A,#25 ; load A with "U"
09F5 32C074 LD (#74C0),A ; write to screen
09F8 3E20 LD A,#20 ; load A with "P"
09FA 32A074 LD (#74A0),A ; write to screen
09FD C9 RET ; return
```

Dat stuk kan vrijgegeven worden: #09EE t/m #09FD.

Code van #09D6 t/m #09FD alleen in two-player mode. Is deels al vrijgegeven.

En dan vrijgeven: #09D6 t/m #09FD.

Wordt hier gebruikt:

```

09BD 3A0F60 LD A,(TwoPlayerGame) ; load A with number of players
09C0 A7 AND A ; 1 player game?
09C1 210960 LD HL,WaitTimerMSB ; load HL with timer address
09C4 110A60 LD DE,GameMode2 ; load DE with game mode2 address
09C7 CAD009 JP Z,#09D0 ; if 1 player game, skip ahead

; 2 player game

09CA 3678 LD (HL),#78 ; store #78 into timer
09CC EB EX DE,HL ; DE <> HL. HL now has game mode2
09CD 3602 LD (HL),#02 ; GameMode2 := 2
09CF C9 RET ; return

; 1 player game

09D0 3601 LD (HL),#01 ; store 1 into timer
09D2 EB EX DE,HL ; DE <> HL. HL now has game mode2
09D3 3605 LD (HL),#05 ; GameMode2 := 5
09D5 C9 RET ; return

```

Dit aan te passen naar:

```

09BD 210960 LD HL,WaitTimerMSB ; load HL with timer address
09C0 110A60 LD DE,GameMode2 ; load DE with game mode2 address
09C3 3601 LD (HL),#01 ; store 1 into timer
09C5 EB EX DE,HL ; DE <> HL. HL now has game mode2
09C6 3605 LD (HL),#05 ; GameMode2 := 5
09C8 C9 RET ; return

```

En dan vrijgeven: #09C9 t/m #09D5.

Let op: voor two-player game wordt GameMode2 op 2 gezet. Checken of die dan alleen dan gebruikt wordt, want dan kan de hele afhandeling van GameMode2 = 2 komen te vervallen?

Wordt hier gebruikt:

```

1315 3A0F60 LD A,(TwoPlayerGame) ; load A with number of players
1318 A7 AND A ; 1 player game?
1319 2807 JR Z,#1322 ; yes, skip next 3 steps

```

Niets te winnen.

Wordt hier gebruikt:

```

1336 3A0F60 LD A,(TwoPlayerGame) ; load A with number of players
1339 A7 AND A ; 1 player game?
133A CA3F13 JP Z,#133F ; yes, skip next step

```

Niets te winnen.

Dan zijn alle TwoPlayerGame variabelen bekijken.

Eerst eens kijken naar de GameMode2 = 2 versus GameMode2 =5 zoals hierboven opgemerkt:

```
06FE 3A0A60 LD A,(GameMode2) ; load A with game mode2
0701 EF RST #28 ; jump based on what the game state is

0702 86 09 ; (0) #0986 ; game start = clears screen, clears sounds, sets screen flip if needed
0704 AB 09 ; (1) #09AB ; copy player data, set screen, set next game mode based on number of players
0706 D6 09 ; (2) #09D6 ; clears palettes, draws "PLAYER <I>", draws player2 score, draws "2UP" (2 player game only)
0708 FE 09 ; (3) #09FE ; copy player data into correct area (2 player game only)
070A 1B 0A ; (4) #0A1B ; clears palettes, draws "PLAYER <II>", update player2 score, draw "2UP" to screen (2 player game only)
070C 37 0A ; (5) #0A37 ; updates high score, player score, remaining lives, level, 1UP
070E 63 0A ; (6) #0A63 ; clears screen and sprites, check for intro screen to run
```

Lijkt erop dat een 2 player game GameMode2 via 2 -> 3 -> 4-> 5 loopt, terwijl een 1 player game meteen naar 5 gaat. Dus dan kun je de code voor GameMode2 =2 en 3 en 4 verwijderen.

Dat is vrijgeven van de code van: #09D6 t/m #0A36.

Ja dat gaat nog steeds goed.

Nu versie v0.60 opgeslagen.

Met de debugger kijken welke GameMode2 waardes doorlopen worden. Alle stukken voor niet gebruikte GameMode2 waarden kunnen weg (wel checken of hier geen verwijzingen naar staan: soms gedeelde stukken en wordt er naar een deel van een ander stuk code gesprongen).

GameMode2 geheugenplaats is: #600A daarop monitoren in de debugger.

0 – 7 (DK logo) – 0 – 1 (High score) – 2 (attract) – 3 (attract) – 4 (dood attract) – 5 – (terug naar 6)
0 - 1 (High Score credit) – 5 – 6 – 7 – 8 - 9 – A (How high) – B – C (spel) – D (dood) – E (dood) –
8 – 9 - A - B – C (spel) – D (dood) – E (dood) -
8 – 9 – A – B – C (spel) - D (dood) – E – 10 (Game Over) – 14 – 0 – 1 ... en opnieuw

En board halen:

E – 8 – 9 – A – B – C (spel) – 16 (DK + Pauline) – 8 – 9 -

En even invincibility aanzetten en dan kijken wat er gebeurd bij einde level (rivets halen) en wat er gebeurd bij game over maar dan entry in high score table.

C (spel rivets) – 16 (DK valt) – 8 – 9 – A – B – C (spel barrels) ...

C (spel) – D (dood) – E (dood) – 10 (Game Over) – 14 – 15 (initials) – 14 (registered) – 0 – 1 ...

Dan dus niet gebruikt: F, 11, 12, 13 en 17.

Kijken of die vrijgemaakt kunnen worden.

F = vrijgeven #1344 t/m #138E.
11 = vrijgeven #13A1 t/m #13A9.
12 = vrijgeven #13AA t/m #13BA.
13 = vrijgeven #13BB t/m #13C9.
17 = vrijgeven #196B t/m #1976.

Testen met een spel spelen met score invoeren. Gaat dat allemaal goed? Dan ruimte succesvol vrijgemaakt en dan met aanpassingen van rivets beginnen.

Ja dat gaat goed. Tot level 4 gespeeld en daarna afgegaan en initialen ingevoerd.

Nu versie v0.61 opgeslagen.

Kijken of de attract mode aangepast kan worden.

Nu blijft jumpman beneden omdat de trap een segment naar rechts verplaatst is.

Attract mode inputs staan hier:

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

21D1 80 FE      ; jump
21D3 01 C0      ; run right
21D5 04 50      ; up = climb ladder
21D7 02 10      ; run left
21D9 82 60      ; jump left
21DB 02 10      ; run left
21DD 82 CA      ; jump left
21DF 01 10      ; run right
21E1 81 FF      ; jump right (gets hammer)
21E3 02 38      ; run left
21E5 01 80      ; run right - mario dies falling over right edge
21E7 02 FF      ; run left
21E9 04 80      ; up
21EB 04 60      ; up
21ED 80         ; ?
```

Eerdere aanpassingen van de attract mode hebben laten zien dat de weergave is vreemd. De actie is het eerste getal en de lengte van de actie is het getal ervoor (tweede getal op de lijn ervoor).

Dus langer naar rechts lopen gaat dan niet echt (is al #FE lang). Daarom een extra run right tussenvoegen en boven ook aanpassen zodat de hamer goed gepakt wordt.

Startend #21D1:

```
80      - jump up
FE 01   - run right
18 01   - run right
90 04   - up (climb ladder)
78 02   - run left
10 82   - jump left
```

```
60 02 - run left
10 82 - jump left
E2 01 - run right
10 81 - jump right (gets hammer)
FF 02 - run left
38 01 - run right
80 02 - run left
FF 04 - up
80 80 - jump up – jump up
```

En het jaartal in de copyright notice moet naar 2022:

In #3F00 de copyright notice:

© 1981-2022 NINTENDO

5C 77 49 4A 10 01 09 08 01 2C 02 00 02 02 10 1E 19 1E 24 15 1E 14 1F 3F

Nu versie v0.62 opgeslagen.

Kijken of we bij rivets ervoor kunnen zorgen dat het spawnen niet links/rechts keuze is, maar een verticale keuze.

Dit zijn de definities, met twee tabellen voor links/rechts:

```
; 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?
```

Met de debugger kijken wat jumpmans Y-waarde (#6205) is op de verschillende girders van boven naar beneden:

#F0 – 1111 0000

#C8 – 1100 1000

#A0 - 1010 0000

#78 – 0111 1000

#50 (top level krijgt geen spawn).

Dus dat betekent steeds een verschil van #28.

Wat zou kunnen is dat deze waarden samengevoegd worden en dat er op basis van een random getal tussen 0 en 7 een combinatie gekozen wordt. En dan checken of de Y-waarde een bepaald verschil heeft en zo nee, dan de volgende waarde nemen.

De tabellen samenvoegen:

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: 1B C8 ; 2nd from bottom, left
3ACE: 23 A0 ; middle, left
3ADO: 2B 78 ; 2nd from top, left
3AD2: 12 F0 ; bottom, left

En dan de code:

```
34BF 0600 LD B,#00 ;\ Load BC with random offset in table
34C1 3A1960 LD A,(RngTimer2) ; |
34C4 E60E AND #0E ; |
34D6 4F LD C,A ;/
;
34C7 C3C909 JP #09C9 ; jump to additional code
```

en:

```
09C9 21C43A LD HL,#3AC4 ; Load HL with start of table data – label BBBB
;
09CC 09 ADD HL,BC ; add this result into HL to get offset into table
;
09CD 23 INC HL ; increase the offset to Y-value of fireball
09CE 7E LD A,(HL) ; load A with Y-value fireball
09CF 57 LD D,A ; store in D
;
09D0 3A0562 LD A,(#6205) ; load A with Y-value jumpman
09D3 92 SUB D ; subtract D
09D4 FE10 CP #10 ; A lower than #10?
09D6 3808 JR C,#09E0 ; yes, jump to label AAAA
09D8 FEFO CP #F0 ; A higher than #F0?
09DA 3004 JR NC,#09E0 ; yes, jump to label AAAA
;
09DC 2B DEC HL ; decrement the offset to X-value of fireball
09DD C3D334 JP #34D3 ; jump back to spawn fireball
```

```

;
09E0 79          LD   A,C           ; load A with C – label AAAA
09E1 C602        ADD  A,#02         ; add #02
09E3 FEOF        CP   #0F           ; A = 0F?
09E5 2002        JR   NZ,#09E9      ; no, skip next step
09E7 3E00        LD   A,#00         ; load A with first offset
09E9 4F          LD   C,A           ; load C with A
;
09EA 18DD        JR   #09C9         ; check next offset – jump to label BBBB

```

Dat lijkt goed te gaan.

Eventjes testen met level L=05. Dan sneller spawnen en wellicht lastiger.

Aanpassen naar:

095E 05 73 3A 01 00 00 00 ; #3A65 is start of table data for screens/levels

En dan in de tabel op #3A73 het eerste board 01 (barrels) vervangen door 04 (rivets).

Deze simpele wijziging maakt het wel veel ingewikkelder. Heb je soms best behoefte aan de spell om de fireballs stop te zetten.

Nu versie v0.63 opgeslagen.

Gaat echter niet altijd goed. Soms toch op dezelfde hoogte spawnen. Hoe kan dat? Nog vreemder is dat dit alleen gebeurd als je de hamer gebruikt, maar dan soms zelfs op midden op het veld spawnen. Precies op de x-waarde waar de hamer de fireball had weggeslagen. Lijkt wel of in dat geval de fireball de oude x- en y-waarden behoud. Hoe kan dat?

Komt doordat de berekening van de afstand voor de JR op #09EA niet goed was. Stond DC, maar moet DD zijn. Dan springt hij terug naar een RET en worden de fireball setting niet gezet (maar is nog wel active), dus een ‘spawn’ op de plek waar de fireball stond.

Is nu goed.

Nu versie v0.64 opgeslagen.

Wat kunnen we nu nog aanpassen?

- Rivets verschuiven? – zou kunnen. Net eventjes anders om te spelen.
- Eventueel additionele gaps? – nee dan de fireballs minder vrij bewegen.
- Eventueel veilige platforms toevoegen? Niet echt nodig.
- Mogelijk de stukken tussen de rivets als schuine girders? Lastig, maar wel leuk.
- Of de stukken tussen de girders 1 karakter verschuiven? Nee fireballs minder vrij bewegen.

Stukken tussen de rivets als schuine girders is wel lastig omdat daarmee de trappen ook weer niet uitlijnen op hele karakters en je dezelfde uitdagingen als het girders board krijgt qua plaatsing van de trappen. Maar dat is dan alleen voor de binnenste ladders. De buitenste lijnen nog steeds goed uit.

Maar als we de schuine girders dezelfde kant op schuin zetten dan is blijft de afstand hetzelfde, dus dan is dat ook geen probleem. Daarnaast is de code voor het aanpassen van de Y-waarde vrij eenvoudig (want voor alle schuine girders hetzelfde).

Daarnaast moeten de plekken voor het spawnen aangepast worden en wellicht de marge voor het niet spawnen op dezelfde girder iets opgehoogd worden.

Eerst eens kijken waar de functionaliteit zit voor het aanpassen van de Y-waarde voor schuine girders voor zowel jumpman als voor de fireballs.

Eerste het veld tekenen en dan pas kijken voor de functionele aanpassingen. Als het heel cool eruit ziet dan doorgaan, als twijfels dan stoppen.

Dit betekent dat elk van de drie girders die we schuin zetten opgesplitst worden in drie nieuwe (links, schuin, rechts). Dus van 3 naar 9 = 6 extra = 6x5=30 geheugenplaatsen. Eerst ruimte daarvoor maken door de definitie van het rivetsboard te verplaatsen achter de girders board definitie. Daarmee maken we de pies en springs kapot, maar dat herstellen we later wel weer.

Girders loopt nu van #3AE4 t/m #3B48.

Rivets verplaatsen van #3C8B t/m 3CEF naar #3B49 t/m #3BAD.
En dan de rest t/m 3CEF vrij maken. Daar dan later de nieuwe boarddefinities voor pies en springs aanmaken.

En dan de plekken waar de board definitie van rivets (#3C8B) gebruikt wordt, aanpassen naar de nieuwe plek (#3B49):

```
0CC3 11493B LD DE,#3B49 ; load DE with start of table data for rivets  
246E 21493B LD HL,#3B49 ; otherwise we're on rivets. load HL with table data for rivets
```

Ja dat gaat goed. Nu board aanpassen.

```
3B99: 05 D7 80 28 80 ; level 4 girder  
3B9E: 05 DF A8 20 A8 ; level 3 girder  
3BA3: 05 E7 D0 18 D0 ; level 2 girder  
3BA8: 05 EF F8 10 F8 ; bottom level girder  
3BAD: AA ; end code
```

Aanpassen naar:

```
3B99: 05 EF F8 10 F8 ; bottom level girder  
3B9E: 05 40 80 28 80 ; level 4 girder left  
3BA3: 05 D7 78 B8 78 ; level 4 girder right  
3BA8: 05 40 A8 20 A8 ; level 3 girder left  
3BAD: 05 DF A0 B8 A0 ; level 3 girder right  
3BB2: 05 40 D0 18 D0 ; level 2 girder left  
3BB7: 05 E7 C8 B8 C8 ; level 2 girder right
```

```
3BBC: 02 BF 78 40 80 ; level 4 slanted girder  
3BC1: AA ; end code
```

Getest, maar de mix van rivet girders en slanted girders gaat niet goed. De slanted girders moeten voor de ladders getekend worden (ander worden er happen uit de ladder gehaald en lijkt de ladder broken) en de rivet girders moeten na de ladders getekend worden (anders uiteinden in de stijl van de slanted girders). Zou nog wel kunnen door slimme volgorde te bepalen. Maar de bovenste ladders en onderste ladders lastig te plaatsen omdat daar de veelvoud van 4 pixels niet te realiseren is.

Dus hier mee stoppen.

Terug naar versie v0.64.

Wellicht anders maken door de vorm van de rivets iets te veranderen. En de kleur van de stangen aan weerszijden van DK donkerbaluw gemaakt.

Nu versie v0.65 opgeslagen.

In de tekst 'USE YOUR MAGIC SPELL !' het woord MAGIC in een andere kleur weergeven?

Aanpassen op #36CC:

```
5E 77 25 23 15 10 29 1F 25 22 10 1D 11 17 19 13 10 23 20 15 1C 1C 10 38
```

Eerste eens kijken welke kleuren er beschikbaar zijn door de letters van MAGIC te veranderen in de jumpman graphic (#FF) die alle kleuren bevat.

De M, G en de C in bruin en de A en de I in beige maken.

Letters gekopieerd naar #98 (M), #99 (A), #9A (G), #9B (I) en #9C (C).

En één positie naar links verschuiven (MAGIC recht onder Goofy Kong) en een extra uitroep teken erachter.

Aanpassen op #36CC:

```
7E 77 38 25 23 15 10 29 1F 25 22 10 98 99 9A 9B 9C 10 23 20 15 1C 1C 10 38
```

Misschien hele scherm anders maken: In het midden de sprite van jumpman vooraanzicht (sprite nr #78) tonen met daaronder de twee teksten:

```
!USE YOUR MAGIC SPELL!  
PRESS P2 TO ACTIVATE
```

Ja dat lijkt me een hele goede. Geeft duidelijk de nieuwe spell mogelijkheid weer.

Dus de Goofy Kong niet tekenen, maar in plaats daarvan de sprite en de teksten plaatsen.

De code voor het tekenen van de Goofy Kong en de jump naar het vervolg van het How High scherm staat van #0BFD t/m 0C1C. Met dan de jump naar vervolg op:

```
0C1A C3820C JP      #0C82      ; continue after freed space
```

Eventjes #0BFD t/m #0C19 vrijmaken door er NOP's van te maken en daardoor wordt de goofy Kong niet meer getekend.

De tekst -USE YOUR MAGIC SPELL- naar boven verplaatsen:

Aanpassen op #36CC:

```
72 77 2C 25 23 15 10 29 1F 25 22 10 98 99 9A 9B 9C 10 23 20 15 1C 1C 2C 3F
```

En dan de sprite tekenen. Is al eerder gedaan met DK Barrelopaloza.

0BFD	212069	LD	HL,#6920	; load HL with heart sprite
0C00	3680	LD	(HL),#80	; set sprite x-position
0C02	23	INC	HL	; next
0C03	3678	LD	(HL),#78	; set sprite number
0C05	23	INC	HL	; next
0C06	3602	LD	(HL),#02	; set sprite color
0C08	23	INC	HL	; next
0C09	3680	LD	HL,#80	; set sprite y-position

En dan de tekst PRESS P2 TO ACTIVATE maken:

```
55 77 20 22 15 23 23 10 20 02 10 24 1F 10 11 13 24 19 26 11 24 15 3F
```

Tekst A '1 OR 2 PLAYERS BUTTON' is vrijgekomen. Staat op #36FD. Daar deze tekst neerzetten.

Tekst ook laten tekenen:

```
0C0B 110A03 LD DE,#030A      ; load task data to draw "PRESS P2 TO ACTIVATE"
0C0E CD9F30 CALL #309F        ; insert task to draw text
```

En dan de P2 ook met donkerbruin en beige kleuren: karakters #94 (P) en #95 (2):

```
55 77 20 22 15 23 23 10 94 95 10 24 1F 10 11 13 24 19 26 11 24 15 3F
```

Ja dat ziet er goed uit. Maar ga het aanpassen: MAGIC met bruin en P2 met beige en niet meer met afwisselende kleuren per letter.

Nu versie v0.66 opgeslagen.

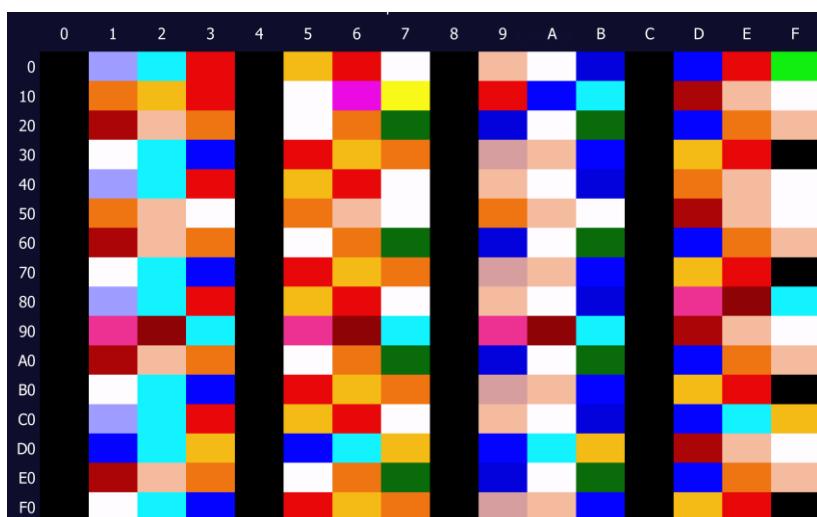
Het ziet er nu zo uit:



Ook de kleuren van het high score scherm aanpassen.



Dat kan niet door de definitie van de lijnen aan te passen, maar door de kleuren in het palette aan te passen. Dat is het eerste van de vier blokken van het palette (lijnen #0 t/m #30).



De groen is palette combinatie 03 (= vakje #0F)

De rood is palette combinatie 04 (= vakje #13)

De geel is palette combinatie 05 (= vakje #17)

De lichtblauw is palette combinatie 06 (= vakje #1B)

Die kleuren in de c-2j.bpr en de c-2k.bpr bestanden vervangen door de gewenste kleuren.

Door het veranderen van de rood van palette combinatie 06 verandert de rode 1UP boven in het scherm niet: dat is namelijk de rood van palette combinatie 00.

Kleur	c-2j.bpr	c-2k.bpr
#1B: Lichtblauw wordt donkerblauw van #0B:	0F	0D
#13: Rood wordt donkergroen van #27:	0F	03
#17: Geel wordt geeloranje van #26:	01	03

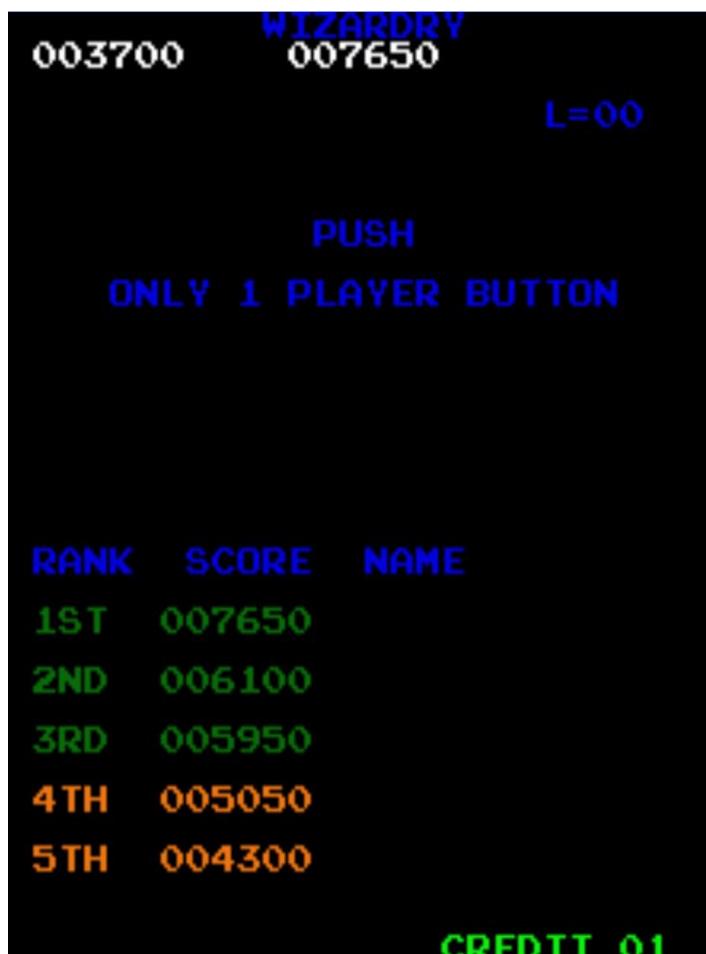
Aanpassingen c-2j.bpr:

```
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 00 0F 0F 0F 01 0E  
00000010 0F 01 00 0F 0F 00 01 01 0F 01 0F 0F 0F 05 00 00  
00000020 0F 05 00 01 0F 00 01 0F 0F 0F 00 0F 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 00 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 0F 0F 0F 00 0F 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 00 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 0F 0F 0F 00 0F 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 00 0F 0F 0E 00 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 0F 0F 0F 00 0F 0F 0F 01 00  
000000F0 0F 00 0E 0F 0F 01 00 01 0F 02 00 0F 0F 00 01 0F
```

Aanpassingen c-2k.bpr:

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

Ziet er nu zo uit:



Nu versie v0.67 opgeslagen.

Uit het testen blijkt dat het redelijk te doen is. Door de andere manier van spawnen is het wel zo dat star pattern vaak helemaal niet kan. Daardoor al interessant. De spell heb je soms hard nodig en soms niet. Wellicht een idee om het aantal fireballs naar 6 te verhogen waardoor de fireball druk nog meer verhoogd wordt, maar dan ook een aantal save zones te maken?

De eerdere aanpassingen voor zes fireballs:

2891	0606	LD	B,#06	; (collisions on barrels)
28B1	0606	LD	B,#06	; (collisions on pies)
28E1	0606	LD	B,#06	; (collisions on elevators)
2902	0608	LD	B,#08	; (collisions on rivets, including hidden squares DK)
2C67	0606	LD	B,#06	; (check fires for deployment on barrels)
3144	0606	LD	B,#06	; (fireball movement and color when hammering)
34F9	0606	LD	B,#06	; (update fires to sprites)
3EAA	0606	LD	B,#06	; (check jumping fires)
115D	11C764	LD	DE,#64C7	; (DK squares moved: original A7 – steps #20/fire)
1166	DD21C064	LD	IX,#64C0	; (DK squares moved: original A0 – steps #20/fire)

```

1137 011C06      LD    BC,#061C      ; (fireball initialization)
1151 11C364      LD    DE,#64C3      ; (DK squares moved: original A3 – steps #20/fire)
317F FE06        CP    #06          ; (fireball deployment limit)
31D7 FE06        CP    #06          ; (fireball processing limit)
30CE 060C        LD    B,#0C          ; (clear fireballs: original OB – steps #01/fire)

```

Ja dat gaat goed. En behoorlijk lastig. Daarom nog wel de savespots maken.

Nu versie v0.68 opgeslagen.

Savspost maken door op vier plekken een karakter te plaatsen. En dat bijvoorbeeld meteen na het plaatsen van de rivets.

Rivets worden zo getekend:

```

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

; start of table data for rivets used above
; these are addresses in video RAM for the rivets

0D17 CA 76      ; #76CA
0D19 CF 76      ; #76CF
0D1B D4 76      ; #76D4
0D1D D9 76      ; #76D9
0D1F 2A 75      ; #752A
0D21 2F 75      ; #752F
0D23 34 75      ; #7534
0D25 39 75      ; #7539

```

Dus de vier save platforms zo tekenen:

```

09F3 0604      LD    B,#04          ; for B = 1 to 4 save platforms to draw – label CCCC
09F5 21020A      LD    HL,#0A02      ; load HL with start of table data – label AAAA
;
09F8 3EB0      LD    A,#B0          ; load A with #B8 = graphic – label BBBB
09FA 5E        LD    E,(HL)        ; load E with high byte of video address
09FB 23        INC   HL            ; next HL

```

```

09FC 56          LD   D,(HL)      ; load D with low byte of video address
09FD 23          INC  HL         ; next HL
;
09FE 12          LD   (DE),A    ; draw save platform on screen
09FF 10F7        DJNZ #09F8     ; next B – jump to label BBBB
0A01 C9          RET           ; return
;
0A02 6C 74        ; #748B – save platform top right – label AAAA
0A04 8C 77        ; #776B - save platform top left
0A06 6C 74        ; #7490 – save platform 2nd right - zelfde als top even
0A08 8C 77        ; #7770 – save platform 2nd left – zelfde als top even

```

Rivets tekenen wordt hier aangeroepen:

```

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

```

De aanroep aanpassen:

```
0CCE CCEC09    CALL Z,#09EC    ; yes, call sub to draw the rivets
```

en sub:

```

09EC CD000D    CALL #0D00    ; call sub to draw the rivets
09EF CDF309    CALL #09F3    ; call sub to draw the saves – label CCCC
09F2 C9          RET         ; return

```

De save platforms voegen niet zo heel veel toe. De zes fireballs zijn gewoon heel erg lastig.
Teruggaan naar versie 0.67.

De tekst DESIGN BY PAUL GOES als naam bij de onderste default score ingeven.

DESIGN BY PAUL GOES

14 15 23 19 17 1E 10 12 29 10 20 11 25 1C 10 17 1F 15 23

```
35ED: 9C 77 05 24 18 10 10 14 15 23 19 17 1E 10 12 29 10 20 11 25 1C 10 17 1F 15 23 10 3F 00 00 43 00 FC 76
```

Nu versie v0.69 opgeslagen.

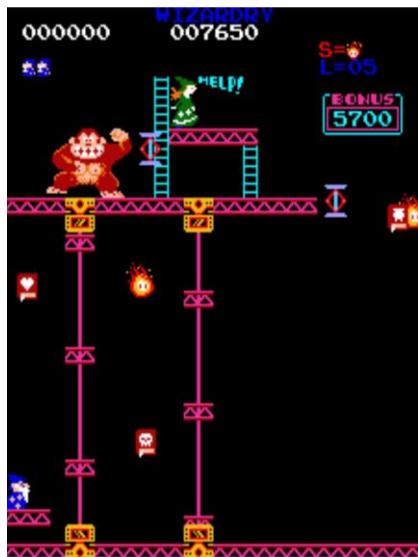
Nu het elevators board gaan aanpassen. Eerst alle girders en ladders verwijderen, behalve die helemaal links waar Jumpman start en die bij DK en Pauline. Daarna de rechter elevator gaan verplaatsen. Dan een structuur ertussen en kijken of daar ook random ladders tussen gezet kunnen worden. En dan eventueel de elevators variabel qua snelheid maken en de springs variabel met tussenafstand.

; table data for the elevators

; 165 bytes long

```
3BE5: 00 63 18 63 58 ; kong's ladder (right)
3BEA: 00 93 38 93 58 ; ladder leading up to girl
3BEF: 02 97 38 68 38 ; girder girl is on
3BF4: 02 B7 58 10 58 ; kong's girder
3BF9: 02 27 E8 10 E8 ; girder where mario starts
3BFE: 02 EF F8 10 F8 ; long bottom girder (mario dies if he gets that low)
3C03: AA           ; end code
```

Ja dat is goed nu. Alles is verwijderd. Ziet er nu zo uit:



Eiegnelijk kijken de linker 1 plek naar links. Dan elevators sneller laten lopen en dan vanaf beneden op de elevator en dan naar het middenstuk rechts springen. Maar als je dat niet haalt een escape naar links en dan met een lange ladder weer naar beneden en nogmaals proberen. En de andere elevator helemaal naar rechts. Maar kan die dan ook naar boven toe bewegen? Of anders elevators omdraaien?

Originele platforms links behouden, maar smaller met een ladder ertussen:

```
3BE5: 00 63 18 63 58 ; kong's ladder (right)
3BEA: 00 93 38 93 58 ; ladder leading up to girl
3BEF: 02 97 38 68 38 ; girder girl is on
3BF4: 02 B7 58 10 58 ; kong's girder
3BF9: 02 EF F8 10 F8 ; long bottom girder (mario dies if he gets that low)
3BFE: 02 1F E8 10 E8 ; lower girder left girder where jumpman starts
3C03: 02 1F 88 10 88 ; higher girder left
3C08: 00 1B 88 1B E8 ; ladder between left girders
3C0D: AA           ; end code
```

En hetzelfde aan de rechterkant voor een symmetrie.

3COD: 02 EF E8 E0 E8 ; lower girder right
3C12: 02 EF 88 E0 88 ; higher girder right
3C17: 00 E3 88 E3 E8 ; ladder between right girders
3C1C: AA ; end code

Nu nog de elevators verschuiven.

Elevators al eerder verschoven in DK Twisted Jungle. Beschreven in het document van Prototype 4.

Vooralsnog twee plaatsen bekend waar de elevator posities bepaald worden:

```
; data used above for top and bottom of elevator shafts

1121 37 45 0F 60          ; X = #37, color = #45, sprite = #F, Y = #60
1125 37 45 8F F7
1129 77 45 0F 60
112D 77 45 8F F7
```

en:

```
; used in elevators - called from #10CC

3E60: 44 03 08 04

; used in elevators, called from #11EC
; used for elevator sprites

3E64: 37 F4
3E66: 37 C0
3E68: 37 8C          ; elevators on left all have X value of 37

3E6A: 77 70
3E6C: 77 A4
3E6E: 77 D8          ; elevators on right all have X value of 77
```

De 37 en de 77 zijn de x-waarden voor respectievelijk de linker en de rechter elevator.

De linker verplaatsen naar 2F en de rechter naar CF?

Nu starten de platformen goed. Maar na verdwijnen worden ze op de oude posities getekend:

```
27BA DD360377 LD      (IX+#03),#77    ; set X position to right side of elevators
27BE DD360D04 LD      (IX+#0D),#04    ; set direction to down
```

en:

```
27F8 DD360337 LD      (IX+#03),#37    ; set X position to left side shaft
27FC DD3605F8 LD      (IX+#05),#F8    ; set Y position to bottom of shaft
2800 DD360D08 LD      (IX+#0D),#08    ; set direction to up
```

En de rode lijnen aanpassen:

Kabels worden hier getekend:

```

; called from #0CF2 for elevators only
; draws the elevator cables

0D27 210D77 LD HL,#770D ; load HL with screen RAM location
0D2A CD300D CALL #0D30 ; draw the left side elevator cable

0D2D 210D76 LD HL,#760D ; load HL with screen RAM location for right side cable

0D30 0611 LD B,#11 ; for B = 1 to #11

0D32 36FD LD (HL),#FD ; draw the cable to screen
0D34 23 INC HL ; next location
0D35 10FB DJNZ #0D32 ; Next B

0D37 110F00 LD DE,#000F ; load DE with offset [why here? should be before loop starts ?]
0D3A 19 ADD HL,DE ; add offset to location
0D3B 0611 LD B,#11 ; for B = 1 to #11

0D3D 36FC LD (HL),#FC ; draw cable to screen
0D3F 23 INC HL ; next location
0D40 10FB DJNZ #0D3D ; Next B

0D42 C9 RET ; return

```

De startpositie voor de rechter rode kabel is #760D, en voor de linker rode kabel is #770D. De #760D moet aangepast worden, maar naar welke waarde? Proefondervindelijk uitproberen. Links wordt dan #772D en rechts #74AD.

Ja dat gaat goed.

Nu versie v0.70 opgeslagen.

Even een girder toevoegen zodat je naar de andere elevator kunt lopen.

```

3C1C: 02 BF 90 47 90 ; long centre girder
3C1C: AA ; end code

```

En nu nog de grenzen van de elevator platforms voor het verplaatsen van jumpman aanpassen.

```

274F 3A0362 LD A,(#6203) ; load A with mario's X position. eg 37 for first, 75 for second
2752 FE2C CP #2C ; position < left edge of first elevator ?
2754 DA6627 JP C,#2766 ; yes, jump ahead

2757 FE43 CP #43 ; else is position < right edge of first elevator ?
2759 DA6F27 JP C,#276F ; yes, jump ahead for first elevator checks

275C FE6C CP #6C ; else is position < left edge of second elevator?
275E DA6627 JP C,#2766 ; yes, jump ahead

2761 FE83 CP #83 ; else is position < right edge of second elevator ?
2763 DA8727 JP C,#2787 ; yes, jump ahead for second elevator checks

```

Oude platform links = #37 en grenzen: #2C en #43
Oude platform rechts = #77 en grenzen: #6C en #83

#B verschil naar links en #C verschil naar rechts:

Nieuwe platform links = #2F : grenzen #24 en #3B
Nieuwe platform rechts = #CF : grenzen #C4 en #DB.

Gaat goed.

Nu versie v0.71 opgeslagen.

Eventjes interessant idee voor het pies scherm. Daar de conveyors vervangen door één lange die maar een kant op gaat en zonder pies. Alleen jumpman gaat sneller een kant op en moet daarbij over brandende ketels springen. Dan verder in het veld nog een constructie met girders die dicht op elkaar zitten zodat je bij springen naar een girder hoger gaat, maar terug gaat niet. En in die constructie dan een aantal fireballs. En dan de extractable ladders behouden en DK heen weer laten gaan. Geen fireballs die helemaal naar boven gaan. Maar wel een bonus level wanneer je een van de twee lange DK ladders beklmt. Bonus level nog bepalen wat: schieten, moeilijk springen, omgekeerde controls? En je haalt dan een music bonus. Dat weergeven bovenin het scherm. Wanneer bij einde rivets en music bonus gehaald een ander muziekje spelen en een bepaalde bonus (hoogte vaststellen nog) geven.

Nog niet helemaal zeker hoe de rechterkant maken. Liefste door middenstuk heen van linksboven naar rechts onder. En dan met de lift naar boven. Dus de richting van de lift omdraaien. En dan misschien de lift wel helemaal rechts maken (dus nog oopschuiven). Later bekijken.

Kijken hoe de springs eerder kunnen vallen. En daarna kijken of dat flexibel/random gemaakt kan worden.

De overgang van bouncer naar falling spring wordt hier gedaan:

```
2E44 23      INC   HL          ; next HL
2E45 DD8605  ADD   A,(IX+#05)    ; add item's Y position
2E48 DD7705  LD    (IX+#05),A    ; store into item's Y position

2E4B DD750E  LD    (IX+#0E),L
2E4E DD740F  LD    (IX+#0F),H    ; store the updated HL for next time
2E51 DD7E03  LD    A,(IX+#03)    ; load A with X position
2E54 FEB7    CP    #B7          ; < #B7 ?
2E56 DA6C2E  JP    C,#2E6C     ; no, skip ahead

2E59 79      LD    A,C          ; yes, A := C
2E5A FE7F    CP    #7F          ; == #7F (end code?)
2E5C C26C2E  JP    NZ,#2E6C     ; no, skip ahead

2E5F DD360D04 LD    (IX+#0D),#04  ; set +D to 4 (??)
2E63 AF      XOR   A          ; A := 0
2E64 328360  LD    (#6083),A    ; clear sound of bouncer
2E67 3E03    LD    A,#03        ; load sound duration of 3
2E69 328460  LD    (#6084),A    ; play sound for falling bouncer
```

Op #2E54 wordt een check gedaan op de X-waarde: zoalng kleiner dan B7, dan door blijven springen, maar wanneer groter dan overgaan op vallen (IX+#0D wordt dan op 4 gezet).

Als die waarde verlaagd wordt, gaat de bouncer waarschijnlijk eerder vallen. Testen bij welke waarden.

Testen met #77 voor een bounce eerder vallen.

Testen met #47 voor twee bounces eerder vallen.

Ja dat gaat in beide gevallen goed.

Nu kijken of je de bouncers bij deployen kunt kenmerken met kort/middel/lang en dan het punt van vallen daarop baseren. En wellicht dit doen middels de kleur van de spring? Dan kun je als speler zien of de spring kort, middel of lang is.

Hoe ziet de spring structuur er uit?

Datastructuur (IX) start bij 6500 met blokken van #10:

- IX+#00: Sprite active (#00 en #01)
- IX+#03: Sprite X-value
- IX+#05: Sprite Y-value
- IX+#0D: Sprite movement (bounce=#01, fall=#04)
- IX+#0E: Used for Y-offset calculation?
- IX+#0F: Used for Y-offset calculation?

Sprites (IY) starten bij #6980 met blokken van #04:

- IY+#00: Sprite X-value
- IY+#01: Sprite value (open=#3B, closed=#3C)
- IY+#02: Sprite color (#00)
- IY+#03: Sprite Y-value

Waar wordt de kleur gezet? Lijkt nergens te gebeuren. #IY+#02 wordt niet gebruikt. Staat op default van #00 en dat is de kleur. Dus we zouden kunnen experimenteren met het zetten van de spring color naar een willekeurige kleur. Om te testen met de verschillende kleuren.

```
; deploy new bouncer

2EAE AF      XOR    A      ; A := 0
2EAF 329663 LD     (#6396),A ; reset bouncer release flag
2EB2 DD360550 LD     (IX+#05),#50 ; set bouncer's Y position to #50
2EB6 DD360D01 LD     (IX+#00),#01 ; set value to sprite bouncing across, not down
2EBA CD5700 CALL   #0057 ; load A with random number
2EBD E60F AND    #0F ; mask bits, result is between 0 and #F
2EBF C6F8 ADD    A,#F8 ; add #F8 = result is now between #F8 and #07
2EC1 DD7703 LD     (IX+#03),A ; store A into initial X position for bouncer sprite
2EC4 DD360001 LD     (IX+#00),#01 ; set sprite as active
2EC8 21AA39 LD     HL,#39AA ; values #39 and #AA to be inserted below. #39AA is the start of table data for Y offsets to add for each movement
2ECB DD750E LD     (IX+#0E),L ;
2ECF DD740F LD     (IX+#0F),H ; store HL into +E and +F
2ED1 C3782E JP     #2E78 ; jump back
```

Dus hier ergens tussenvoegen:

2ED1	C3EC09	JP	#09EC	; jump to additional code – jump to label AAAA
------	--------	----	-------	---

en:

09EC	3E01	LD	A,#01	; load A with color – label AAAA
09EE	FD7702	LD	(IY+#02),A	; set sprite color
09F1	C3782E	JP	#2E78	; jump back

Kleur 01 = geel met wit - mooi
 Kleur 02 = beige met blauw
 Kleur 03 / 04 / 05 /06 = roze met lichtblauw
 Kleur 07 = donkerbruin met wit
 Kleur 08 = donkerbruin met lichtbruin
 Kleur 09 = wit met donkergroen
 Kleur 0A = donkerblauw met donkergroen – mooi
 Kleur 0B = donkerblauw met beige
 Kleur 0C = wit met donkerblauw – mooi
 Kleur 0D = donkerrood met lichtbruin
 Kleur 0E = beige met donkerblauw – mooi
 Kleur 0F = geel met zwart

Kleur 00, 01 en 0E voorlopig gebruiken.

De kleur zetten op basis van een random getal tussen 0 en F. Als 0-5 dan kleur 00. Als 6-A dan kleur 01 en anders kleur 0E.

Aanpassen:

```

09EC 3A1860      LD   A,(RngTimer1) ; load A with reandom number – label AAAA
09EF E60F        AND  #0F          ; random number between #00 and #0F
;
09F1 FE06        CP   #05          ; between #00 and #05?
09F3 3004        JR   NC,#09F9    ; no, skip next steps
09F5 3E00        LD   A,#00          ; load A with color
09F7 180A        JR   #0A03        ; jump ahead
;
09F9 FE0B        CP   #0A          ; between #06 and #0A?
09FB 3004        JR   NC,#0A01    ; no, skip next steps
09FD 3E01        LD   A,#01          ; load A with color
09FF 1802        JR   #0A03        ; jump ahead
;
0A01 3E0E        LD   A,#0E          ; load A with color
;
0A03 FD7702        LD   (IY+#02),A ; set sprite color – label BBBB
0A06 C3782E        JP   #2E78        ; jump back

```

Ja gaat goed.

Nu versie v0.72 opgeslagen.

En dan nu de beslissing overgang bounce naar fall laten afhangen van de kleur van de spring.

```

2E44 23      INC   HL          ; next HL
2E45 DD8605  ADD   A,(IX+#05)  ; add item's Y position
2E48 DD7705  LD    (IX+#05),A  ; store into item's Y position

2E4B DD750E  LD    (IX+#0E),L
2E4E DD740F  LD    (IX+#0F),H  ; store the updated HL for next time
2E51 DD7E03  LD    A,(IX+#03)  ; load A with X position
2E54 FEB7    CP    #B7        ; < #B7 ?
2E56 DA6C2E  JP    C,#2E6C   ; no, skip ahead

2E59 79      LD    A,C        ; yes, A := C
2E5A FE7F    CP    #7F        ; == #7F (end code?)
2E5C C26C2E  JP    NZ,#2E6C   ; no, skip ahead

2E5F DD360D04 LD    (IX+#0D),#04  ; set +D to 4 (??)
2E63 AF      XOR   A          ; A := 0
2E64 328360  LD    (#6083),A  ; clear sound of bouncer
2E67 3E03    LD    A,#03      ; load sound duration of 3
2E69 328460  LD    (#6084),A  ; play sound for falling bouncer

```

Aanpassen:

```

2E54 C3090A    JP    #0A09      ; jump to additional code – jump to label AAAA
2E57 0000      NOP, NOP

```

en:

```

0A09 FD7E02    LD    A,(IY+#02)  ; load A with color of sprite – label AAAA
;
0A0C FE00      CP    #00        ; sprite color = 00?
0A0E 2004      JR    NZ,#0A14   ; no, skip next steps
0A10 16B7      LD    D,#B7      ; load D with #B7
0A12 180A      JR    #0A1E      ; jump ahead – jump to label BBBB
;
0A14 FE01      CP    #01        ; sprite color = 01?
0A16 2004      JR    NZ,#0A1C   ; no, skip next steps
0A18 1677      LD    D,#77      ; load D with #77
0A1A 1802      JR    #0A1E      ; jump ahead – jump to label BBBB
;
0A1C 1647      LD    D,#47      ; load D with #47
;
0A1E DD7E03    LD    A,(IX+#03)  ; load A with X position bouncer - label BBBB
0A21 BA        CP    D          ; A < D?
0A22 DA6C2E    JP    C,#2E6C   ; no, continue bouncing
;
0A25 C3592E    JP    #2E59      ; yes, check falling

```

Ja dat gaat goed.

Nu versie v0.73 opgeslagen.

Gezien de plekken waar de springs vallen de rechter elevator opschuiven tot helemaal aan de buitenkant.

Nieuwe platform rechts = #E7 : grenzen #DC en #F3, rode lijn = 744D.

Oude platform links = #37 en grenzen: #2C en #43
Oude platform rechts = #77 en grenzen: #6C en #83

En dan ladder naar links en naar boven zodat je het board kunt finishen.

3C0D: 02 D7 88 C8 88 ; lower girder right
3C12: 02 EF 58 C8 58 ; higher girder right
3C17: 00 D3 58 D3 88 ; ladder between right girders
3C1C: 02 BF 90 47 90 ; long centre girder
3C21: AA ; end code

Dat ziet er nu goed uit.

Nu versie v0.74 opgeslagen.

En dan de elevator ook naar boven laten gaan.

Waar wordt dit gewisseld?

Elevator movement zit van #2797 t/m #2807.

Er zijn zes elevator platforms. Datastructuur vanaf 6400 met stappen van #10.

Movement indicator is IX+#0D als deze #04 is dan elevator omlaag (is de rechter) en als #08 dan elevator omhoog (is de linker).

Als een platform links bovenaan is, dan wordt deze overgezet naar rechts (bovenaan) en dan gaat deze naar beneden. Als helemaal beneden dan wordt de elevator inactive gemaakt. Inactive platform wordt daarna weer active gemaakt op de linker elevator onderaan.

Eigenlijk nu dan #08 is linker elevator en #04 is rechter elevator. Als bij linker bovenaan dan naar rechter onderaan (dus naast de X-waarde ook de Y-waarde aanpassen). Als bij rechter bovenaan dan inactief maken. En bij beiden de Y-waarde decreaseen (waardoor omhoog bewegen).

Linker elevator (was 'moving up'):

```

; elevator is moving up

27AE DD7E05 LD A,(IX+#05) ; load A with elevator Y position
27B1 3D DEC A ; decrement (move up)
27B2 DD7705 LD (IX+#05),A ; store result
27B5 FE60 CP #60 ; at top of elevator ?
27B7 C2C227 JP NZ,#27C2 ; no, skip next 2 steps

27BA DD360377 LD (IX+#03),#77 ; set X position to right side of elevators
27BE DD360D04 LD (IX+#0D),#04 ; set direction to down

27C2 DD19 ADD IX,DE ; add offset for next elevator
27C4 10DA DJNZ #27A0 ; next B
27C6 C9 RET ; return

```

Daar moet de Y-positie ook aangepast worden.

```

27BA C3280A JP #0A28 ; jump to additional code – jump to label AAAA
27BD 00 NOP

```

en:

```

0A28 DD3603E7 LD (IX+#03),#E7 ; set X-position to right elevator – label AAAA
0A2C DD3605F8 LD (IX+#05),#F8 ; set Y position to bottom of shaft
0A30 C3BE27 JP 27BE ; jump back

```

De direction moet wel aangepast worden, maar is eigenlijk omzetten naar rechter elevator.

Rechter elevator (was ‘moving down’):

```

; elevator is moving down

27C7 DD7E05 LD A,(IX+#05) ; load A with elevator Y position
27CA 3C INC A ; increase (move down)
27CB DD7705 LD (IX+#05),A ; store result
27CE FEF8 CP #F8 ; at bottom of shaft ?
27D0 C2C227 JP NZ,#27C2 ; no, loop for next

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

```

Daar omhoog bewegen en testen bereiken top:

```

27CA 3D DEC A ; decrease (move up)

```

en:

```

27CE FE60 CP #60 ; at top op shaft ?

```

Ja dat gaat goed.

Alleen jumpman beweegt op de rechter elevator nog naar beneden. Dat nog aanpassen:

```
; riding on second elevator

2787 3A0562 LD A,(#6205) ; load A with mario's Y position
278A FEE8 CP #E8 ; at bottom of elevator ? (death)
278C D27F27 JP NC,#277F ; yes, set death and return

278F 3C INC A ; else increment (move mario down)
2790 320562 LD (#6205),A ; store back into mario's Y position
2793 324F69 LD (#694F),A ; store into mario sprite Y value
2796 C9 RET ; return
```

Aanpassen:

```
278A FE71 CP #71 ; at top of elevator (death)?
278C DA7F27 JP C,#277F ; yes, set death and return
```

en:

```
278F 3D DEC A ; else decrement (move jumpman up)
```

Ja dat werkt.

Nu versie v0.75 opgeslagen.

Raar is nog wel dat eerst een gat ontstaat tussen de rechter elevator platforms, wat even later rechtgetrokken wordt en blijft. Komt door initiële posities van de elevator platforms denk ik.

Maar wellicht platform verwijderen (5 stuks gaan gebruiken?) en eventueel variabele afstanden introduceren? Geen idee hoe dat laatste gaat, lijkt goed te komen bij de deployment van nieuwe aan de linkerkant. Soort timer daarop blijkbaar.

5 stuks gebruiken kan door aanpassen:

```
109E 0605 LD B,#05 ; for B = 1 to 5
```

en:

```
2797 0605 LD B,#05 ; for B = 1 to 5 (for each elevator)
```

Niet zo heel spannend. Wellicht interessanter om het spawnen variabel te maken. Daardoor ontstaan gaps die doorschuiven naar de rechter.

Kijken waar het spawnen gedaan wordt. Bepalen waar besluit om nieuwe te deployen.

Dat lijkt hier te gebeuren: met interval van #34.

```
27F4 DD360001 LD (IX+#00),#01 ; make elevator active
27F8 DD360337 LD (IX+#03),#37 ; set X position to left side shaft
27FC DD3605F8 LD (IX+#05),#F8 ; set Y position to bottom of shaft
2800 DD360D08 LD (IX+#0D),#08 ; set direction to up
2804 3634 LD (HL),#34 ; reset elevator counter to #34

2806 35 DEC (HL) ; decrease elevator counter
2807 C9 RET ; return
```

Wat als we dit interval vergroten? Ja dan wordt de afstand ook groter.

Wellicht deze standaard afstand #34 vergroten met een random getal onder de 32?

Bij aanpassen opletten want er wordt vanuit #27DF naar de #2806 gesprongen. Als stuk verplaatsen dan ook die aanroep verplaatsen.

```
27DA 21A762 LD HL,#62A7 ; load HL with elevator counter address  
27DD 7E LD A,(HL) ; load A with elevator counter  
27DE A7 AND A ; == 0 ?  
27DF C20628 JP NZ,#2806 ; no, skip ahead, decrease counter and return
```

Aanpassen:

```
2804 C3AD0B JP #0BAD ; jump to additional code – jump to label AAAA  
2807 00 NOP
```

en:

```
0BAD 3A1960 LD A,(RngTimer2) ; load A with a random number – label AAAA  
0BB0 E63F AND #3F ; number between 0 en #3F  
0BB2 C601 ADD A,#01 ; add #01  
0BB4 77 LD (HL),A ; reset elevator counter  
;  
0BB5 35 DEC (HL) ; decrease elevator counter – label BBBB  
0BB6 C9 RET ; return
```

en:

```
27DF C2B50B JP NZ,#0BB5 ; no, jump ahead, jump to label BBBB
```

Gaat niet goed. Afstand lijkt variabel, maar ook wel heel regelmatig (dus niet echt random) en ook een afstand die veel te klein is: zou niet moeten kunnen. Klopte, eerste versie was via de CALL om random nummer te bepalen, maar die past de HL aan... Gaat nu wel goed.

Snelheid aanpassen gaat niet goed. Dan valt jumpman door de elevator platformen heen.

Nu versie v0.76 opgeslagen.

En dan nu de girders ertussen aanmaken. En veld verder aanpassen:

```
3BE5: 00 63 18 63 58 ; kong's ladder (right)  
3BEA: 00 93 38 93 58 ; ladder leading up to girl  
3BEF: 02 97 38 68 38 ; girder girl is on  
3BF4: 02 B7 58 10 58 ; kong's girder  
3BF9: 02 EF F8 10 F8 ; long bottom girder (mario dies if he gets that low)  
3BFE: 02 1F E8 10 E8 ; lower girder left girder where jumpman starts  
3C03: 02 1F 80 10 80 ; higher girder left  
3C08: 00 1B 80 1B E8 ; ladder between left girders  
3COD: 02 D8 68 C8 68 ; lower girder right  
3C12: 02 C7 60 B8 60 ; higher girder right  
3C17: 00 53 18 53 58 ; kong's ladder (left)  
;  
3C1C: 02 BF 80 47 80 ; long centre girder top  
3C21: 02 BF 98 5F 98 ; long centre girder top middle  
3C26: 02 BF B0 5F B0 ; long centre girder middle  
3C2B: 02 BF C8 5F C8 ; long centre girder bottom middle  
3C30: 02 CF E0 5F E0 ; long centre girder bottom  
;  
3C35: 00 73 80 73 98 ; 1st ladder between top and top middle  
3C3A: 00 A3 80 A3 98 ; 2nd ladder between top and top middle  
;
```

```

3C3F: 00 5B 98 5B B0      ; 1st ladder between top middle and middle
3C44: 00 8B 98 8B B0      ; 2nd ladder between top middle and middle
3C49: 00 B3 98 B3 B0      ; 3rd ladder between top middle and middle
;
3C4E: 00 73 B0 73 C8      ; 1st ladder between middle and bottom middle
3C53: 00 A3 B0 A3 C8      ; 1st ladder between middle and bottom middle
;
3C58: 00 5B C8 5B E0      ; 1st ladder between top middle and middle
3C5D: 00 8B C8 8B E0      ; 2nd ladder between top middle and middle
3C62: 00 B3 C8 B3 E0      ; 3rd ladder between top middle and middle
;
3C21: AA                  ; end code

```

Nu versie v0.77 opgeslagen.

Kijken of de elevator rechts hoger kan doorlopen. Dan kun je namelijk de trap daar weghalen en daardoor het middenplatform symmetrischer maken.

Rechter elevator (was 'moving down'):

```

; elevator is moving down

27C7 DD7E05 LD A,(IX+#05)      ; load A with elevator Y position
27CA 3C INC A                 ; increase (move down)
27CB DD7705 LD (IX+#05),A       ; store result
27CE FEF8 CP #F8               ; at bottom of shaft ?
27D0 C2C227 JP NZ,#27C2        ; no, loop for next

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

```

Daar omhoog bewegen en testen bereiken top:

```
27CA 3D DEC A                 ; decrease (move up)
```

en:

```
27CE FE48 CP #48               ; at top op shaft ?
```

En dan de mount ook #20 omhoog verplaatsen:

```
1129 77 45 0F 48
```

En de elevator cable ook doortrekkenb. Die worden hier getekend:

```

0D27 210D77 LD HL,#770D ; load HL with screen RAM location
0D2A CD300D CALL #0D30 ; draw the left side elevator cable

0D2D 210D76 LD HL,#760D ; load HL with screen RAM location for right side cable

0D30 0611 LD B,#11 ; for B = 1 to #11

0D32 36FD LD (HL),#FD ; draw the cable to screen
0D34 23 INC HL ; next location
0D35 10FB DJNZ #0D32 ; Next B

0D37 110F00 LD DE,#000F ; load DE with offset [why here? should be before loop starts ?]
0D3A 19 ADD HL,DE ; add offset to location
0D3B 0611 LD B,#11 ; for B = 1 to #11

0D3D 36FC LD (HL),#FC ; draw cable to screen
0D3F 23 INC HL ; next location
0D40 10FB DJNZ #0D3D ; Next B

0D42 C9 RET ; return

```

#0D2D was al aangepast naar starten vanaf #744D. Nu 3 posities omhoog: #744A. En dan langer tekenen (doet dat wel aan beide kanten dan. Dus tekenloop #14 lang i.p.v. #11. En dan de offset kleiner maken #000C. Aan de linkerkant wordt het bovenin zichtbaar. Daarom linker kabel omhoog doen: start nu op #772D. Nu 2 posities omhoog: #772B.

Gaat goed. Om te testen even de fireballs laten starten op de juiste balken en de bonus items positioneren.

Fireballs worden hier geïnitialiseerd:

```

; set up the 2 fireballs

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

; set up 2nd fireball

1101 DD362001 LD (IX+#20),#01 ; set fire active
1105 DD3623EB LD (IX+#23),#EB ; set fire X position
1109 DD362EEB LD (IX+#2E),#EB ; set fire X position
110D DD362560 LD (IX+#25),#60 ; set fire Y position
1111 DD362F60 LD (IX+#2F),#60 ; set fire Y position

```

Fireball 1: X = 78, Y = A8
 Fireball 2: X = 98, Y = D8

Bonus items plaatsen. Definitie staat hier:

```

; 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

```

Umbrella: X = 1B, Y = 78

Hat: X = 5B, Y = C0

Purse: X = BB, Y = 78

Op de rechter gaat jumpman nog steeds op de oude hoogte dood. Dat nog aanpassen naar de verhoging:

```
278A FE59      CP      #59      ; at top of elevator (death)?
```

Nu alleen nog invincibility weer uitzetten en dan gaan testen.

Nu versie v0.78 opgeslagen.

De sprong naar de rechter elevator is vrij lastig. Makkelijker maken door de girder een positie naar rechts te verlengen. Dan wordt het 'uitstekende' deel gelijk aan het uitstekende deel linksboven. Qua symmetrie fijn. Je kunt dan wel voorbij de vallende springs veilig gaan staan wachten zonder dat je geraakt kan worden door de springs. Maar natuurlijk nog wel het gevaar van de fireballs. En als je verder gaat staan, dan kun je niet meer terugtrekken als er een fireball aankomt. In de huidige situatie kun je overigens ook op het uiterste puntje gaan staan, zonder geraakt te worden.

```
3C30: 02 D7 E0 5F E0      ; long centre girder bottom
```

Ziet er nu zo uit:



Het veld is heel lastig als de fireballs heel snel naar boven gaan. Ze gaan niet zo snel meer naar beneden en afwachten is vervelend. Daar nog even mee testen. Wellicht op linker deel blijven

wachten totdat ze naar beneden gaan. Of wellicht kijken of er een extra girder boven in gemaakt kan worden waar de fireballs niet kunnen komen.

Je kunt links wachten, net iets naar beneden. En dan wachten totdat de fireballs naar beneden. En dan springen via elevator (wel lastig) en eventueel fireballs bevriezen. Hiermee testen. Ja dat gaat goed. Is niet makkelijk, maar kan wel.

Nu versie v0.79 opgeslagen.

En dan nu gaan kijken hoe we de pie factory kunnen gaan aanpassen.

Alle girders en trappen eruit halen. Alleen de standaard zaken bovenin behouden en de onderste lopende band, onderste girder en de linker trap daartussen.

Eerst de boel opschuiven? Nee, kan altijd nog. Maar nu opschuiven maakt eventuele aanpassingen lastig. Dus eerst aanpassen. Misschien houden we bij pies board ook nog wel ruimte over.

3B5D: 00 63 18 63 58 ; kong's ladder (right)
3B62: 00 53 18 53 58 ; kong's ladder (left)
3B67: 01 23 58 23 80 ; top broken ladder left side
3B6C: 01 DB 58 DB 80 ; top broken ladder right side
3B71: 00 2B D0 2B F8 ; bottom level ladder #1 of 4
3B76: 00 93 38 93 58 ; ladder leading to girl
3B7B: 02 97 38 68 38 ; girder where girl sits
3B80: 03 EF 58 10 58 ; top conveyor girder
3B85: 03 EF D0 10 D0 ; main lower conveyor girder (has hammer)
3B8A: 02 EF F8 10 F8 ; bottom level girder
3B8F: AA ; end code

Is goed.

Eerst de ketel naar beneden en dan kijken waar de conveyor moet komen. En dan kijken of er vier ketels kunnen komen.

De ketel (oil can) wordt hier gezet:

```
104F 21043E LD HL,#3E04 ; set up copy job from table in #3E04 = oil can sprite
1052 11FC69 LD DE,#69FC ; destination is #69FC = sprite
1055 010400 LD BC,#0004 ; four bytes to copy
1058 EDB0 LDIR ; draw oil can
```

Met data:

```
3E04 7F 49 0C 88 ; oil can for conveyors ?
```

Naar beneden brengen (kleur was al aangepast toen het een ketel gemaakt is):

```
3E04 7F 49 07 F0 ; oil can for conveyors
```

En dan moet het vuur dat erin brand ook naar beneden gebracht worden. Vuur wordt hier gezet:

```
1049 21FA3D LD HL, #3DFA ; load HL with start of table data  
104C CDFA11 CALL #11FA ; set fireball sprite
```

Lijkt gezet te worden in code van #11FA t/m #4162 met de volgende data:

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

Met de volgende betekenissen: X-pos / sprite nr / sprite color / Y-pos / hitbox width / hitbox height.

Naar beneden krijgen:

```
3DFA: 7F 40 01 E0 02 00
```

Ja dat is nu goed.

Nu kijken conveyor naar beneden, trap aanpassen en ook jumpman snelheid aanpassen aan conveyor. Dan kijken of het leuk is. Als leuk dan kijken hoe er extra ketels toegevoegd kunnen worden.

```
3B71: 00 2B E0 2B F8 ; bottom level ladder #1 of 4
```

```
3B85: 03 EF E0 10 E0 ; main lower conveyor girder (has hammer)
```

Ja is leuk, maar is zo laag niet leuk. Dus terug naar oorspronkelijke hoogte en ketel omhoog op van die kruisen.

```
3E04 7F 49 07 D8 ; oil can for conveyors
```

```
3DFA: 7F 40 01 C8 02 00
```

En dan de kruisen:

```
3B8F: 06 8F E0 70 E0 ; central patch of XXX's
```

Geprobeerd met één rij, maar dat is heel raar: jumpman sprite gaat achter de ketel langs en gaat voor de kruisen langs. Dan is de illusie dat de ketel op de kruisen staat weg. Dus niet doen. Misschien helemaal aan het einde van het ontwerp nog een leuke toepassing voor de kruisen, maar nu niet doen.

De conveyor één plek naar beneden doen. Wellicht extra ruimte later.

```
3B71: 00 2B D8 2B F8 ; bottom level ladder #1 of 4 D8?
```

```
3B85: 03 EF D8 10 D8 ; main lower conveyor girder (has hammer)
```

```
3E04 7F 49 07 E0 ; oil can for conveyors
```

```
3DFA: 7F 40 01 D0 02 00
```

Ja dat is goed nu. Nog wel de draaipunten ook één plek naar beneden verplaatsen.

; 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
```

Aanpassen:

```
3E34: 17 50 00 DC
3E38: E7 D0 00 DC
```

Ook nog de adjust van jumpmans snelheid op de onderste conveyor aanpassen:

```
2AE4 FEC8      CP      #C8          ; mario on lower pie tray ?
2AE6 CAF02A    JP      Z,#2AF0      ; yes, skip ahead
```

Aanpassen:

```
2AE4 FEC8      CP      #D0          ; mario on lower pie tray ?
```

Interessant:

```
25F5 CD0226    CALL    #2602      ; handle top conveyor and pulleys
25F8 CD2F26    CALL    #262F      ; handle middle conveyor and pulleys
25FB CD7926    CALL    #2679      ; handle lower conveyor and pulleys
25FE CDD32A    CALL    #2AD3      ; handle mario's different speeds when on a conveyor
2601 C9        RET     ; return
```

Door #25F5 t/m 25FD NOP's te maken, wellicht helemaal geen pies meer laten verschijnen. Kijken of dat zo is. Ja dat klopt, maar dan ook geen DK die nog beweegt en ook jumpman beweegt niet meer op de onderste. Ze lijken wel stil te staan. Kijken of dat in de eerste geregeld wordt. Die weer terugzetten. Ja nu beweegt DK weer. Maar de onderste twee conveyors doen niets. Ook de onderste weer terugzetten.

Ja dat gaat goed.

Nu alleen geen pies meer deployen. Deployment geldt voor alle conveyors, dus voor nu eventjes helemaal uitzetten. Als we besluiten geen pies meer te gebruiken, dan kijken hoe e.e.a. uit de code gehaald kan worden en vrijgegeven kan worden.

```

; look for a pie to deploy

2530 0606 LD B,#06 ; for B = 1 to 6 pies
2532 111000 LD DE,#0010 ; load DE with offset of #10 (16 decimal)
2535 DD21A065 LD IX,#65A0 ; load IX with start of pie sprites table

2539 DDCB0046 BIT 0,(IX+#00) ; is this pie already onscreen?
253D CA4525 JP Z,#2545 ; no, jump ahead and deploy this pie

2540 DD19 ADD IX,DE ; else load offset for next pie
2542 10F5 DJNZ #2539 ; next B

2544 C9 RET ; return [no room for more pies, 6 already onscreen]

```

Aanpassen:

```
253D 000000 NOP, NOP, NOP ; do not deploy pies anymore
```

Eventjes de hamer daar weghalen, want zo erg lastig testen.

Aanpassen:

```
3E10: 23 8D 23 AD ; for conveyors
```

Kijken of voor de onderste conveyor de richting niet aangepast kan worden, maar altijd naar rechts blijft.

Code vanaf #2679 voor de lower tray. Omdraaien via deze routine:

```

26DE CB7E BIT 7,(HL) ; is this direction moving right ?
26E0 CAE626 JP Z,#26E6 ; yes, skip next 2 steps

26E3 3602 LD (HL),#02 ; store 2 into (HL) - reverses the pie tray direction (now moving right)
26E5 C9 RET ; return

26E6 36FE LD (HL),#FE ; store #FE into (HL) - reverses the pie tray direction (now moving left)
26E8 C9 RET ; return

```

Dus moving right is door in HL #FE te stoppen.

Code die deze routine aanroeft:

```

2679 3A1A60 LD A,(FrameCounter) ; load A with this clock counts down from #FF to 00 over and over...
267C 0F RRCA ; rotate right. is there a carry?
267D DA8D26 JP C,#268D ; yes, skip ahead

2680 21A562 LD HL,#62A5 ; no, load HL with this counter
2683 35 DEC (HL) ; count it down. zero?
2684 C28D26 JP NZ,#268D ; no, skip ahead

2687 36FF LD (HL),#FF ; yes, reset counter to #FF
2689 2C INC L ; HL := #62A6 = master direction vector for lower level
268A CDDE26 CALL #26DE ; reverse direction of this tray

268D 21A662 LD HL,#62A6 ; load HL with master direction vector for lower level
2690 CDE926 CALL #26E9 ; load A with direction vector for this frame
2693 32A663 LD (#63A6),A ; store A into pie direction for lower level
2696 3A1A60 LD A,(FrameCounter) ; load A with this clock counts down from #FF to 00 over and over...
2699 E61F AND #1F ; mask bits. now between 0 and #1F
269B FE02 CP #02 ; == 2 ? (1/32 chance?)
269D C0 RET NZ ; no, return

269E 11F469 LD DE,#69F4 ; load DE with pulley sprite start
26A1 EB EX DE,HL ; DE <> HL
26A2 CDA626 CALL #26A6 ; call sub below to animate the pulleys [why? it should just continue here]
26A5 C9 RET ; return

```

Blijkbaar is #62A6 de richting van de onderste conveyor. Daar dan dus #FE in stoppen. Aanpassen:

```

268A 36FE LD (HL),#FE ; set direction to right
268C 00 NOP

```

Ja dat is op zich wel goed. Alleen de eerste ronde van DK naar rechts en naar links loopt de conveyor naar links. Dan in deze code en dan wordt het omgedraaid. Dus de initiele waarde aanpassen. Of een heel stuk van deze code veranderen in NOP's.

Eerst even veiligstellen.

Nu versie v0.80 opgeslagen.

#62A6 wordt initieel gevuld met FF op #0F78. Daar wordt een data tabgel gekopieerd naar geheugen vanaf #6280. Dus de #62A6 - #6280 = #26 (=38e) plek staat de initiële waarde.

```

;; values are copied into #6280 through #6280 + #40
;; 3D9C: 00 00 23 68
;; 3DA0: 01 11 00 00 00 10 DB 68 01 40 00 00 08 01 01 01
;; 3DB0: 01 01 01 01 01 01 00 00 00 00 00 00 80 01 C0 FF
;; 3DC0: 01 FF FF 34 C3 39 00 67 80 69 1A 01 00 00 00 00
;; 3DD0: 00 00 00 00 04 00 10 00 00 00 00 00 00 00 00 00

```

Dat lijkt #3DC1 te zijn. Die veranderen in #02?

Ja dat werkt. De onderste conveyor start naar rechts draaiend en blijft naar rechts draaien.

Dan kijken of er meerdere vuurketels weergegeven kunnen worden.

De ene wordt hier getekend:

```
1049 21FA3D LD HL,#3DFA ; load HL with start of table data
104C CDFA11 CALL #11FA ; set fireball sprite

104F 21043E LD HL,#3E04 ; set up copy job from table in #3E04 = oil can sprite
1052 11FC69 LD DE,#69FC ; destination is #69FC = sprite
1055 010400 LD BC,#0004 ; four bytes to copy
1058 EDB0 LDIR ; draw oil can
```

Met datababel:

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

En datababel:

```
3E04 7F 49 0C 88 ; oil can for conveyors ?
```

Kijken of dit uitgebreid kan worden. Door er een loop van te maken met extra datababel.

Maar welke sprites dan vrij: kijken met debugger.

Huidige sprite voor ketel (oil can) is #69FC en voor vuur is #6A28.

Het vuur wordt ook aan een datababel toegevoegd voor de collision detection. Dat is vanaf #66A0.

Huidige memory gebruik:

69F0	73	D0	00	84	17	52	00	DC	E7	D1	00	DC	7F	49	07	E0
6A00	4F	3A	0F	18	5F	3A	0F	18	6F	3A	0F	18	53	73	07	A0
6A10	8B	74	07	F0	DB	75	07	A0	23	1E	07	8D	23	1E	07	AD
6A20	00	00	00	00	00	00	00	00	7F	42	01	D0	00	00	00	00
6A30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
6A40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
6A50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
6A60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

66A0	01	00	00	7F	00	D0	00	40	01	02	00	00	00	00	00	00
66B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
66C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
66D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
66E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
66F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
6700	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
6710	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
6720	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Het vuur wordt geconfigureerd in routine vanaf #11FA. In die routine staat het adres van de sprite keihard: #6A28. Maar die wordt ook gebruikt voor de oil can van het barrels board. Kijken of je die toch nog kan hergebruiken door iets verder aan te roepen?

Ja dat moet kunnen:

```

11FA DD21A066 LD    IX,#66A0      ; load IX with sprite memory array for fire above the barrel
11FE 11286A LD    DE,#6A28      ; load DE with hardware sprite memory for same fire
1201 DD360001 LD    (IX+#00),#01  ; enable the sprite
1205 7E        LD    A,(HL)      ; load A with table data
1206 DD7703 LD    (IX+#03),A    ; store into sprite X position
1209 12        LD    (DE),A      ; store into sprite X position

```

Door IX en DE te vullen met de gewenste waarde en dan een CALL te doen naar #1201.

In totaal 3 vuurketels gaan gebruiken.

Sprites vuurketel: #6A2C, #6A30 en #6A34.
 Sprites vuur: #6A20, #6A24 en #6A28.
 Datatabel vuur: #66A0, #66B0 en #66C0.

Eerst ruimte vrijmaken. De middelste conveyors niet gebruiken en ook geen pies meer.

Op #25F8 zijn 3 NOP's aangemaakt voor afhandelen van de middelste conveyor. Die roepen de code aan op #262F. Die kan vrijgegeven worden: #262F t/m #2678.

In de main zitten drie stukken voor de pies:

1992 CALL 24EA – Deploy and handle pies – mag allemaal weg !
 199B CALL 2207 – Retractable ladder – houden !
 19AA CALL 25F2 – Handle conveyor directions and adjust jumpman speed – houden !

Het hele stuk van afhandeling van de pies kan vrijgegeven worden: #24EA t/m #25F1.

En de call op #1992 verwijderen: in NOP's veranderen.

Vrijgegeven. Nu testen met de boards die al klaar zijn. Of er niet per ongeluk delen van de verwijderde code gebruikt worden.

Ja dat gaat goed.

Nu versie v0.81 opgeslagen.

Datatabel voor de ketels:

24EA: 4F 49 07 E0 ; left oil can
 24EE: 7F 49 07 E0 ; middle oil can
 24F2: AF 49 07 E0 ; high oil can

Datatabel voor de fires:

24F6: 4F 40 01 78 02 00 ; left fire
 24FC: 7F 40 01 78 02 00 ; left fire

2502: AF 40 01 78 02 00 ; left fire

Volgende vrije is #2508.

Aanpassen tekenen vuur:

```
1049 21FA3D LD HL,#3DFA      ; load HL with start of table data
104C CDFA11 CALL #11FA       ; set fireball sprite

104F 21043E LD HL,#3E04      ; set up copy job from table in #3E04 = oil can sprite
1052 11FC69 LD DE,#69FC      ; destination is #69FC = sprite
1055 010400 LD BC,#0004      ; four bytes to copy
1058 EDB0 LDIR               ; draw oil can

1049 21F624 LD HL,#24F6      ; load HL with start of table data
104C CD0825 CALL #2508       ; set fireball sprites
```

en:

```
2508 DD21A066 LD IX,#66A0    ; load IX with sprite memory array for fire
250C 11206A LD DE,#6A20    ; load DE with hardware sprite memory for same fire
250F CD0112 CALL #1201      ; draw left fire
;
2512 23 INC HL             ; next table entry
2513 13 INC DE             ; next hardware sprite
;
2514 DD21B066 LD IX,#66B0    ; load IX with sprite memory array for fire
2518 CD0112 CALL #1201      ; draw middle fire
;
251B 23 INC HL             ; next table entry
251C 13 INC DE             ; next hardware sprite
;
251D DD21C066 LD IX,#66C0    ; load IX with sprite memory array for fire
2521 CD0112 CALL #1201      ; draw right fire
;
2524 C9 RET                ; return
```

Aanpassen tekenen ketels:

```
104F 21EA24 LD HL,#24EA      ; set up copy job from table in #24EA
1052 112C6A LD DE,#6A2C      ; destination is #6A2C – sprites
1055 010C00 LD BC,#000C      ; twelve bytes to copy
```

De drie ketels en de drie vuurtjes worden getekend. Alleen nog de volgende zaken die nog goedgezet moeten worden:

- Alleen rechter vuur beweegt en lijkt te branden, de andere staan stil

- Alleen linker vuur doet mee in collision detection
- Als het bonus item beneden gepakt wordt verdwijnt deze, maar ook de middelste ketel.
- Als over het linker vuur gesprongen wordt, verdwijnt de middelste ketel ook.

Nu versie v0.82 opgeslagen.

Collision detection oil can pies wordt hier gedaan:

```

28D0 0601 LD B,#01 ; B := 1
28D2 78 LD A,B ; A := 1
28D3 32B963 LD (#63B9),A ; store counter for use later
28D6 1E00 LD E,#00 ; E := 0
28D8 DD21A066 LD IX,#66A0 ; load IX with oil can address
28DC CD1329 CALL #2913 ; check for collision with oil can fire

```

Maar zijn er nu drie. Al voldoende om B op #03 te zetten en E op #10? Ja dat helpt.

Vuur laten bewegen. Nu beweegt alleen de rechter. Wordt hier gedaan:

```

; called from #03CE and #03EC above
; animates the oilcan fire

03F2 70 LD (HL),B ; store B into (HL) - set the oilcan fire sprite
03F3 3A1960 LD A,(RngTimer2) ; load A with random number
03F6 0F RRCA ; rotate right. carry ?
03F7 D8 RET C ; yes, return

03F8 04 INC B ; else increase B
03F9 70 LD (HL),B ; store B into (HL) - set the oilcan fire sprite with higher value
03FA C9 RET ; return

```

Maar dat niet alleen voor HL doen (dat is slechts één fire), maar voor alle drie.

Even kijken wat er in HL zit en welke dat dan is. Watchpoint in debugger op #03F2.

Daar zit 6A29 in? Dat klopt. Dat is de sprite value van de rechter fire sprite.

Maar dat dan ook doen voor de andere twee vuurtjes. En lekker in hetzelfde ritme laten bewegen.
Magisch vuur dat in sync brandt?

Ineens idee dat we nu het oilcan tekenen bij het barrelsboard ook beïnvloed hebben.
Even checken. Ja dat klopt. Hiermee stoppen. Wel leuk idee, maar al eens eerder gedaan. Daarnaast vrij lastig om te springen en ook nog eens barrels board beïnvloed.

[Terug naar versie v0.81.](#)

De ene vuurton houden en later nog twee hindernissen die op en neer gaan maken, die jumpman op de onderste lopende band moet ontwijken.

Nu eerst het middenstuk verder ontwikkelen. Eerst de twee draaipunten van de middelste conveyors weghalen en het spawnen van de fireballs niet meer doen.

Datatabel voor de draaipunten:

```
; 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
```

Het zijn die met de y-waarde van 84. Daar de X-waarde op 01 zetten (#3EC2 en #3E30).

Het deployen van de fireballs wordt hier gedaan:

```
; deploys fireball/firefoxes
; Arrive here from main routine at #1995

2DDB 3E0A      LD      A,#0A          ; A := binary 1010 = code for rivets and conveyors
2DDD F7        RST    #30            ; returns immediately on girders and elevators, else continue
```

Daar een andere testwaarde gebruiken. #0A = 1010. Alleen rivets nog maar doen: #08 = 1000.

Rivets doet het inderdaad nog. Maar het rare is dat er nog steeds fireballs op pies gespawned worden. Alsof er nog ergens een spawn gedaan wordt. Vreemd.

Checken of de code van #2DDE geraakt wordt. Zou niet moeten. En die wordt ook niet uitgevoerd. Maar waar worden de fireballs dan gedeployed?

Er is een additionele routine vanuit de main op #2FCB. Daar wordt voor niet-barrel boards op basis van een timer bepaalt dat er een fire gereleased kan worden door #03 in #62B9 te zetten. Ben benieuwd waar dit dan afgehandeld wordt.

Lijkt hier te zitten:

```

; called from #02DE

03A2 3E03 LD A,#03 ; A := 3 = 0011 binary
03A4 F7 RST #30 ; only continue if level is girders or conveyors, else RET

03A5 D7 RST #10 ; if mario is alive, continue, else RET

03A6 3A5063 LD A,(#6350) ; load A with 1 when an item has been hit with hammer
03A9 0F RRCA ; has an item been hit with the hammer ?
03AA D8 RET C ; yes, return, we don't do anything here while hammer hits occur

03AB 21B862 LD HL,#62B8 ; load HL with this counter
03AE 35 DEC (HL) ; decrease. at zero?
03AF C0 RET NZ ; no, return

03B0 3604 LD (HL),#04 ; yes, reset counter to 4
03B2 3AB962 LD A,(#62B9) ; load A with fire release indicator
03B5 0F RRCA ; roll right. carry? Is there a fire onscreen or is it time to release a new fire?
03B6 D0 RET NC ; no, return

```

Kijken wat er gebeurt als de check in #03A2 aangepast wordt naar #0001? Dan alleen dit doen in girders en niet meer in pies. Klopt nu geen spawn meer in pies.

Wel eventjes checken of fireballs spawn en hammering in alle velden goed gaat.

Ja dat lijkt allemaal goed te gaan. Nee dat gaat niet goed. Want nu is de animatie van de vuurketel verdwenen. Terugdraaien. Eventjes zo laten. Als er vijf fireballs pre-spawned zijn en er geen hamers meer zijn. Dan blijven ze en worden er geen respawned.

Nu extra middenveld maken. Eerst kijken een aantal lagen met girders en dat je van girder naar girder gaat als je springt en dan plekken maken waar je weer naar beneden kan. En dan fireballs spawnen op die girders. Als je daar voorbij bent dan nog kijken voor een vervolg deel met rollende barrels waarbij bij springen je kop stoot tegen de barrels die op de girder erboven rollen.

Girders maken op #3B8F:

```

3B8F: 02 EF B8 10 B8 ; 2nd level girder
3B94: 02 EF A8 10 A8 ; 3rd level girder
3B99: 02 EF 98 10 98 ; 4th level girder
3B9E: 00 E3 B8 E3 D8 ; ladder to 2nd level girder
3B9E: AA

```

Dit gaat niet goed. Afstand van een girder ertussen. Waarschijnlijk omdat jumpman bij het springen nu zijn hoofd stoot tegen de 4th level girder.

Proberen zonder afstand van een girder ertussen.

```

3B8F: 02 EF B8 10 B8 ; 2nd level girder
3B94: 02 EF B0 10 B0 ; 3rd level girder
3B99: 02 EF A8 10 A8 ; 4th level girder
3B9E: 00 E3 B8 E3 D8 ; ladder to 2nd level girder
3B9E: AA

```

Dit gaat ook niet goed. Het lukt niet om te springen. Wordt afgeremd door de girder erboven en zakt dan weer terug naar beneden. Dan misschien een soort pyramide maken?

Daarnaast gaat detrap niet goed. De uiteinden zijn normale girder, maar dat overschijft de conveyor girder van 3B85. Daarom die omdraaien:

3B85: 00 E3 B8 E3 D8 ; ladder to 2nd level girder

...

3B8F: 02 EF B8 10 B8 ; 2nd level girder

3B94: 02 DF B0 20 B0 ; 3rd level girder

3B99: 02 CF A8 30 A8 ; 4th level girder

3B9E: 03 EF D8 10 D8 ; main lower conveyor girder (has hammer)

3B9E: AA

Leuk allemaal, maar gaat niet werken. Wellicht toch als conveyors houden en dan bijvoorbeeld conveyor levels omdraaien dus de lange conveyor wisselen met de twee kleine en de vuurbeker. En dan misschien de ladders onderin random plaatsen? Dan heb je al snel druk van de fireballs en kun je de spell gebruiken.

[Terug naar versie v0.79.](#)



De twee conveyors met de vuurbeker één naar beneden: 80 -> A8

De lange conveyor twee omhoog: D0 -> 80

De losse drie girders één naar beneden: A8 -> D0

De patches met XXX één naar beneden: 90 -> B8, 98 -> C0, A0 -> C8.

3B5D: 06 8F B8 70 B8 ; central patch of XXX's
3B62: 06 8F C0 70 C0 ; central patch of XXX's
3B67: 06 8F C8 70 C8 ; central patch of XXX's
3B6C: 00 63 18 63 58 ; kong's ladder (right)
3B71: 00 63 80 63 A8 ; center ladder to left of oil can fire
3B76: 00 63 D0 63 F8 ; bottom level ladder #2 of 4
3B7B: 00 53 18 53 58 ; kong's ladder (left)
3B80: 00 53 A8 53 D0 ; ladder under the hat
3B85: 00 9B 80 9B A8 ; center ladder to right of oil can fire
3B8A: 00 9B D0 9B F8 ; bottom level ladder #3 of 4
3B8F: 01 23 58 23 80 ; top broken ladder left side
3B94: 01 DB 58 DB 80 ; top broken ladder right side
3B99: 00 2B 80 2B A8 ; ladder on left platform with hammer
3B9E: 00 D3 80 D3 A8 ; ladder on right platform with umbrella
3BA3: 00 A3 A8 A3 D0 ; ladder to right of bottom hammer
3BA8: 00 2B D0 2B F8 ; bottom level ladder #1 of 4
3BAD: 00 D3 D0 D3 F8 ; bottom level ladder #4 of 4
3BB2: 00 93 38 93 58 ; ladder leading to girl
3BB7: 02 97 38 68 38 ; girder where girl sits
3BBC: 03 EF 58 10 58 ; top conveyor girder
3BC1: 03 F7 A8 88 A8 ; top right conveyor next to oil can
3BC6: 03 77 A8 08 A8 ; top left conveyor next to oil can
3BCB: 02 A7 D0 50 D0 ; center ledge
3BD0: 02 E7 D0 B8 D0 ; right center ledge
3BD5: 02 3F D0 18 D0 ; left center ledge (has hammer)
3BDA: 03 EF 80 10 80 ; main lower conveyor girder (has hammer)
3BDF: 02 EF F8 10 F8 ; bottom level girder
3BE4: AA ; end code

Dat gaat goed. Dan de draaipunten aanpassen:

3E24: 17 50 00 5C
3E28: E7 D0 00 5C
3E2C: 8C 50 00 AC ; was 84 (girder 80) – now AC (girder A8)
3E30: 73 D0 00 AC ; was 84 (girder 80) – now AC (girder A8)
3E34: 17 50 00 84 ; was D4 (girder D0) – now 84 (girder 80)
3E38: E7 D0 00 84 ; was D4 (girder D0) – now 84 (girder 80)

Dat gaat goed.

Nu de vuurbeker en het vuur naar beneden.

Naar beneden brengen (kleur was al aangepast toen het een ketel gemaakt is):

3E04 7F 49 07 B0 ; oil can for conveyors

En dan moet het vuur dat erin brand ook naar beneden gebracht worden.

3DFA: 7F 40 01 A0 02 00

Dat gaat goed.

Het spawnen van de fireballs ook aanpassen.

```
; table data referenced in #
; 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
```

Alle waarden moeten #28 verhoogd worden.

```
80 7B 78 76 74 73 72 71 70 70 6F 6F 6F 70 70 71 72 73 74 75 76 77 78
A8 A3 A0 9E 9C 9B 9A 99 98 98 97 97 97 98 98 99 9A 9B 9C 9D 9E 9F A0
```

Dat gaat goed.

De hamers verplaatsen:

```
3E10: 23 B5 ; was 8D = bovenste hamer moet 28 omlaag = B5
3E12: 7B 64 ; was B4 = onderste hamer moet 2x28 omhoog = 64
```

Nu versie v0.83 opgeslagen.

Nu nog de pies op de goede hoogte spawnen en jumpman adjust op de goede hoogte doen.

Origineel worden twee Y-waarden gebruikt voor de deployment van pies. Dat zijn #7C (de bovenste twee korte conveyors) en #CC (de onderste lange conveyor)

De twee conveyors met de vuurbeker één naar beneden: 80 -> A8 : 7C -> A4

De lange conveyor twee omhoog: D0 -> 80 : CC -> 7C.

Aanpassen:

```
254A DD3605A4 LD (IX+#05),#A4 ; store #A4 into pie's Y position
```

```
2558 DD36057C LD (IX+#05),#7C ; store #7C into pie's Y position
```

```
25AF FEA4 CP #A4 ; is this the top level pie?
```

Dat gaat goed.

Nu nog jumpmans speed adjust aanpassen. Dat zit hier:

```
2AD3 3A0362 LD A,(#6203) ; load A with mario's X position
2AD6 47 LD B,A ; copy to B
2AD7 3A0562 LD A,(#6205) ; load A with mario's Y position
2ADA FE50 CP #50 ; is mario on upper level ?
2ADC CAEA2A JP Z,#2AEA ; yes, skip ahead

2ADF FE78 CP #78 ; mario on upper pie tray?
2AE1 CAF62A JP Z,#2AF6 ; yes, skip ahead

2AE4 FEC8 CP #C8 ; mario on lower pie tray ?
2AE6 CAF02A JP Z,#2AF0 ; yes, skip ahead
```

De twee conveyors met de vuurbeker één naar beneden: 80 -> A8 : 78 -> A0

De lange conveyor twee omhoog: D0 -> 80 : C8 -> 78.

Aanpassen:

```
2ADF FEA0 CP #A0 ; mario on upper pie tray?

2AE4 FE78 CP #78 ; mario on lower pie tray ?
```

Ja dat werkt goed.

Nu versie v0.84 opgeslagen.

Level is nog niet symmetrisch: wel maken.

Zorgen dat de fireballs aan random kant gespawned worden.

Wellicht ladders iets anders.

Kijken of het mogelijk is om te finishen bij Pauline.

En bekijken plaatsing hamers.

En bekijken plaatsing items.

En eventueel item (note) bij de ladder van DK?

Fireballs aan random kant spawnen. Dat gebeurt hier:

De fireballs random aan beide zijden laten spawnen. Is nu gebaseerd op de X-positie van jumpman:

```
348A 3A0362 LD A,(#6203) ; \ Jump if Mario is on left side of the screen,
348D CB7F BIT 7,A ; |
348F CAA834 JP Z,#34A8 ; /
```

Random kant spawnen van de fireballs:

```
348A 3A1960 LD A,(RngTimer2) ; Determine side to spawn fireball
```

348D CB47	BIT	0,A	; If timer value is even . . .
348F CAA834	JP	Z,#34A8	; . . . jump and spawn the fireball at left side

Ja dat werkt.

Nu kijken of je bij Pauline kunt eindigen.

De afwijkende ending voor pies wordt hier geregeld:

```

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

1E69 3A0362 LD A,(#6203) ; else load A with mario's X position
1E6C 17 RLA ; on left or right side of screen?

1E6D 3E00 LD A,#00 ; load A with #00. sprite for facing left
1E6F DA741E JP C,#1E74 ; if on left side, skip next step

1E72 3E80 LD A,#80 ; else load A with sprite facing right
1E74 324D69 LD (#694D),A ; set mario sprite
1E77 C3851E JP #1E85 ; jump ahead

```

De CP op #1E66 aanpassen naar #31. Dan nog wel kijken hoe je het einde doet als je een van DK's ladders op gaat. Maar eerst maar eens kijken wat er op de bovenste conveyor gebeurt.

Ja dat gaat goed en op de bovenste conveyor gaat jumpman mee naar links en naar rechts.

Alleen wanneer hij langs de retractable ladder gaat, gaat hij mee naar beneden en valt dan. Kijken hoe dat opgelost kan worden. Nee, dat is erg lastig op te lossen. Niet doen. Dus laten zoals het was. Dus weer terugzetten naar #51.

De onderste drie girders omzetten naar één girder en extra gebroken trappen maken.

```

3BCB: 02 EF D0 10 D0 ; center ledge
3BD0: 01 23 A8 23 D0 ; broken ladder left
3BD5: 01 DB A8 DB D0 ; broken ladder right

```

Maar nu de korte conveyors overschreven met girder uiteinde van de gebroken trappen. Daarom de definities daarvan omdraaien:

```

3BC1: 03 F7 A8 88 A8 ; top right conveyor next to oil can
3BC6: 03 77 A8 08 A8 ; top left conveyor next to oil can
...
3BD0: 01 23 A8 23 D0 ; broken ladder left
3BD5: 01 DB A8 DB D0 ; broken ladder right

```

Omdraaien naar:

3BC1: 01 23 A8 23 D0 ; broken ladder left

3BC6: 01 DB A8 DB D0 ; broken ladder right

...

3BD0: 03 F7 A8 88 A8 ; top right conveyor next to oil can

3BD5: 03 77 A8 08 A8 ; top left conveyor next to oil can

En dan nog symmetrisch maken: een ladder op een niet-symmetrische plek:

3BA3: 00 AB A8 AB D0 ; ladder to right of bottom hammer

Ja dat is nu goed. Nu een tijdje testen. Lijkt wel erg lastig.

De onderste hamer hangt nu precies voor de trap. Die nog opschuiven.

3E10: 3B B5 ; was 8D = bovenste hamer moet 28 omlaag = B5

Toch te snel opgegeven met eindigen bij Pauline. Toch doen en probleem met retractable ladder oplossen.

De CP op #1E66 aanpassen naar #31. Ja nu weer eindigen bij Pauline. Dat wel weer aanpassen zodat jumpman bij Pauline gezet wordt. De settings daarvoor eventjes bekijken met de debugger. De jumpman sprite is: #694C.

#694C (X-pos) = 93, #694D (sprite nr) = 00, #694E (color) = 02, #694F (Y pos) = 30.

De X-pos en de sprite nr moeten aangepast worden bij finish aan de linkerkant.

Dit is de code die dit afhandelt:

```
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

1E69 3A0362 LD A,(#6203) ; else load A with mario's X position
1E6C 17 RLA ; on left or right side of screen?

1E6D 3E00 LD A,#00 ; load A with #00. sprite for facing left
1E6F DA741E JP C,#1E74 ; if on left side, skip next step

1E72 3E80 LD A,#80 ; else load A with sprite facing right
1E74 324D69 LD (#694D),A ; set mario sprite
1E77 C3851E JP #1E85 ; jump ahead
```

Klopt niet. Comments suggereren dat #1E72 geskipt wordt wanneer aan de linker kant, maar volgens mij moet dat juist zijn dat dat stuk wel uitgevoerd wordt. Even testen met debugger. Ja hoor wanneer links dan wordt 1E72 wel uitgevoerd en rechts niet.

Dus #1E72 wordt alleen uitgevoerd wanneer aan de linkerkant. Dus op #1E6F een conditionele jump doen naar additionele code waar de juiste aanpassingen gedaan worden. Jump doen als links. Maar commenst kloppen niet , dus juist jumpen als NC.

Wellicht hier ook alvast vijf NOP's reserveren zodat daar later een waarde in een memory veld gezet kan worden zodat in rivets een alternatief muziekje gespeeld kan worden.

```
1E6F  D24413      JP    C,#1344      ; jump to additional code – jump to label AAAA  
1E72  3E00        LD    A,#00      ; else load A with sprite facing left
```

en:

```
1344  3E93        LD    A,#93      ; load A with new X position – label AAAA  
1346  324C69      LD    (#694C),A   ; set new X position  
;  
1349  0000        NOP, NOP  
134B  000000      NOP, NOP, NOP  
;  
134E  3E00        LD    A,#00      ; load A with sprite facing left  
1350  C3741E      JP    #1E74      ; jump back
```

Ja dat gaat nu allemaal goed.

Nu versie v0.85 opgeslagen.

Kijken oplossen retractable ladders. Lijkt erop dat de retractable ladders adjustment gedaan wordt op basis van X-waarde kleiner dan bepaalde waarde. Maar daardoor ook doen wanneer jumpman op de conveyor van DK staat.

Dat lijkt hier getest te worden:

```
2243  3A0562      LD    A, (#6205)    ; load mario's Y position  
2246  FE7A        CP    #7A          ; is mario on the top pie tray level or above?  
2248  D25722      JP    NC,#2257     ; no, skip ahead and return to higher sub
```

Maar kijken op welke waarde de Y-positie dan staat wanneer jumpman meerijdt op de bovenste conveyor (DK's) conveyor. Dus dit alleen doen als jumpman hoger is dan #7A. De onderkant van de retractable ladder zit op de girder van #78. Dus jumpman moet eerst een stukje klimmen. Maar op de conveyor van DK zit jumpman op Y-positie #50. Dus dan niet doen.

Aanpassen:

```
2246  C35313      JP    #1353      ; jump to additional code – jump to label AAAA
```

2249 0000 NOP. NOP

en:

```
1353 FE51      CP    #51      ; is jumpman at #50 or lower ? – label AAAA
1355 DA5722    JP    C,#2257  ; yes, do not process retractable ladder
;
1358 FE7A      CP    #7A      ; is jumpman at 7A or higher?
135A D25722    JP    NC,2257  ; yes, do not process retractable ladder
;
135D C34B22    JP    #224B    ; jump back
```

Dat gaat goed.

Nu versie v0.86 opgeslagen.

Ga je af door tegen Donkey Kong aan te lopen? Testen met invincibility uit.

Heel lastig. De twee gebroken ladders heel maken:

3BC1: 00 23 A8 23 D0 ; broken ladder left

3BC6: 00 DB A8 DB D0 ; broken ladder right

Je gaat dus niet af door tegen Donkey Kong aan te lopen. Kijken hoe dat opgelost is in Donkey Kong Reverse.

De X-positie van Kong wordt vastgelegd in variabele op #6910.

De volgende code handelt de collision detection af:

```
; called from main routine at #19B3
; checks for collisions with hostiles sprites

2808 FD210062 LD    IY,#6200    ; load IY with start of mario sprite
280C 3A0562 LD    A,(#6205)  ; load A with mario's Y position
280F 4F      LD    C,A       ; copy to C
2810 210704 LD    HL,#0407  ; H := 4, L := 7
2813 CD6F28 CALL  #286F    ; checks for collisions based on the screen. A := 1 if collision, otherwise zero
2816 A7      AND   A       ; was there a collision ?
2817 C8      RET    Z       ; no, return

; mario collided with hostile sprite

2818 3D      DEC   A       ; else A := 0
2819 320062 LD    (#6200),A  ; store into mario life indicator, mario is dead
281C C9      RET    ; return
```

Dit uitbreiden met een check wanneer in conveyors screen en de y-coordinaat van jumpman is tussen de conveyor en de girder van Pauline (dat is de hoogte van Donkey Kong) en de x-coordinaat van jumpman is tussen linker en rechterkant van Donkey Kong. Dit allemaal proefondervindelijk achterhalen welke coordinaten dat zijn.

Start y-coördinaat van jumpman naast Pauline is #31 en waarschijnlijk is de y-coördinaat van de conveyor dan #51.

Aanpassen:

```
2816 A7      AND   A          ; was there a collision ?
2817 CA6013  JP    Z,#1360   ; no, jump ahead - jump to label AAAA
281A C37F13  JP    #137F     ; yes, jump ahead, handle - jump to label BBBB
```

en:

; check jumpman collided with Donkey Kong in conveyors screen

```
1360 3A2762 LD    A,(#6227)  ; load A with screen number - label AAAA
1363 FE02   CP    #02       ; are we on the conveyors?
1365 C0     RET   NZ        ; no, return no collision

1366 3A0562 LD    A,(#6205)  ; load A with jumpman's y position
1369 FE31   CP    #31       ; jumpman higher than Donkey Kong ?
136B D8     RET   C         ; yes, return no collision
136C FE51   CP    #51       ; jumpman lower than Donkey Kong ?
136E D0     RET   NC        ; yes, no collision

136F 3A1069 LD    A,(#6910)  ; load A with Donkey Kong's x position
1372 D619   SUB   #19       ; calculate left side of Donkey Kong
1374 47     LD    B,A       ; copy to B, left side of Donkey Kong
1375 C630   ADD   A,#30     ; calculate right side of Donkey Kong
1377 4F     LD    C,A       ; copy to C, right side of Donkey Kong
1378 3A0362 LD    A,(#6203)  ; load A with x position of jumpman
137B B8     CP    B         ; jumpman left of Donkey Kong ?
137C D8     RET   C         ; yes, no collision
137D B9     CP    C         ; jumpman right of Donkey Kong ?
137E D0     RET   NC        ; yes, no collision
```

; jumpman collided with hostile sprite

```
137F 3E00   LD    A,#00     ; A := 0 - label BBBB
1381 320062 LD    (#6200),A ; store into jumpman life indicator, means dead
1384 C9     RET   
```

Interessant als dit werkt. Dan ook nog testen of de collisions met andere enemies (barrels, fireballs, springs nog goed werken).

Gaat goed. Maar de hitbox van DK is zo wel erg groot. Aan beide kanten kleiner maken.

```
1372 D613   SUB   #13       ; calculate left side of Donkey Kong
1375 C62A   ADD   A,#2A     ; calculate right side of Donkey Kong
```

Ja dat gaat goed.

Nu versie v0.87 opgeslagen.

Het veld is toch wel erg lastig. De bovenste lange conveyor met pies veranderen in een gewone girder.

Tekenen als gewone girder:

3BDA: 02 EF 80 10 80 ; main lower conveyor girder (has hammer)

En de draaipunten verwijderen:

3E24: 17 50 00 5C
3E28: E7 D0 00 5C
3E2C: 8C 50 00 AC
3E30: 73 D0 00 AC
3E34: 01 50 00 84
3E38: 01 E7 D0 00 84

Ja dat is goed.

En dan niet meer als conveyor met pies vullen:

Dat was de onderste conveyor. Afhandeling daarvan even stopzetten. Maar dat stopt wellicht ook de bewegingen van Donkey Kong.

```
25F2 3E02 LD A,#02 ; load A with 2 = 0010 binary
25F4 F7 RST #30 ; return if not conveyors

25F5 CD0226 CALL #2602 ; handle top conveyor and pulleys
25F8 CD2F26 CALL #262F ; handle middle conveyor and pulleys
25FB CD7926 CALL #2679 ; handle lower conveyor and pulleys
25FE CDD32A CALL #2AD3 ; handle mario's different speeds when on a conveyor
2601 C9 RET ; return
```

Even #25FB NOP's maken.

Ja dat gaat goed. Donkey Kong beweegt nog en jumpman wordt daar aangepast qua snelheid. Op de bovenste conveyor worden geen pies meer spawned en jumpman wordt niet aangepast qua snelheid. En op de twee korte worden wel pies spawned en wordt jumpman wel qua snelheid aangepast.

Dat betekent dat de code van #2679 vrijgegeven kan worden: #2679 t/m #26A5.

Daar nog wel eventjes goed mee testen, want niet makkelijk.

Nu versie v0.88 opgeslagen.

De bonus items nog aanpassen.

3E3C: 4B 73 07 78
3E40: 8B 74 07 F0
3E44: DB 75 07 50

Ja dat is goed.

Er komen niet zo heel veel pies op de kleine conveyors en de richting draait soms vreemd om.

De richting lijkt om te draaien wanneer DK door het midden gaat. Dat niet meer doen. Gewoon van buiten naar binnen laten gaan. En kijken of er iets meer spawned kunnen worden.

De code voor het draaien zit hier:

```
262F 21A362 LD HL, #62A3 ; load HL with address of master direction vector for middle conveyor
2632 3A0562 LD A, (#6205) ; load A with mario's Y position
2635 FEC0 CP #C0 ; is mario slightly above the lower conveyor?
2637 DA6F26 JP C, #266F ; yes, skip ahead. in this case the upper trays don't vary
```

Draaien gebeurt niet als jumpman boven #C0. Maar dat dus eigenlijk nooit doen. Dus de conditie kan er uit.

```
2637 C36F26 JP #266F ;jump ahead
```

Ja dat gaat goed.

En iets meer pies deployen op de twee kleine conveyors. Dat lijkt hier bepaald te worden:

```
; deploy a pie

2545 CD5700 CALL #0057 ; load A with a random number
2548 FE60 CP #60 ; < #60 ?
254A DD36057C LD (IX+#05), #7C ; store #7C into pie's Y position
254E DA5825 JP C, #2558 ; yes, skip next 3 steps

2551 3AA362 LD A, (#62A3) ; load A with master direction for middle conveyor
2554 3D DEC A ; is this tray moving outwards ?
2555 C26E25 JP NZ, #256E ; no, skip ahead

2558 DD3605CC LD (IX+#05), #CC ; store #CC into pie's Y position
255C 3AA662 LD A, (#62A6) ; load A with master direction vector for lower conveyor
255F 07 RLCA ; is this tray moving to the right ?
```

Dus de vergelijking met #60 bepaald op welke conveyor de pie gedeployed wordt. Kijken als we deze heel laag zetten wat er gebeurt. Bij #FE bijna geen pies. Bij #10 vrij veel pies. Daarom op #10 zetten.

Twee ladders weer broken maken.

3BC1: 01 23 A8 23 D0 ; broken ladder left
3BC6: 01 DB A8 DB D0 ; broken ladder right

De bovenste girder in drie delen splitsen.

De originele girder inkorten:

3BDA: 02 A7 80 50 80

De twee extra girders vooraan de definitie toevoegen:

3B53: 02 E7 80 BD 80

3B58: 02 3F 80 18 80

En dan de verwijzingen naar de definitie aanpassen van #3B5D naar #3B53:

0CDF 11_{533B} LD DE,#_{3B53} ; load DE with start of table data for conveyors

2461 21_{533B} ID HL,#_{3B53} ; load HL with start of table data for conveyors

Ja dat gaat goed.

Nu nog een bonus item iets opschuiven:

3E3C: 4B 73 07 78

3E40: 8B 74 07 F0

3E44: DB 75 07 50

Nu versie v0.89 opgeslagen.

Daarmee gaan testen. Lijkt vrij makkelijk zo.

De eerste testen lijken wel goed te gaan zo.

Wel zou het logisch zijn (en soms gewenst), als je ook de retractable ladder af kan gaan. Dit is al eerder geregeld voor DK Reverse. De retractable ladder functionaliteit zorgt alleen voor verschuiving van jumpman. Maar de onderliggende ladder is een broken ladder. Daar kan jumpman niet opgaan vanaf de bovenkant. Dat betekent dat de code hiervoor aangepast moet worden.

; else mario at top of ladder

```
1B31 7E      LD     A,(HL)      ; load A with broken ladder indicator
1B32 A7      AND    A          ; is this ladder broken?
1B33 C0      RET    NZ        ; yes, return. we can't go down broken ladders
```

Op regel 1B33 niet meteen een return doen als broken ladder, maar springen naar stuk code waar die test gedaan wordt (en daarvandaan dan eventueel returnen) als juiste ladder uitgeschoven is, niet returnen, maar jump naar #1B34.

Aanpassen:

1B31 C3A113 JP #13A1 ; jump to additional code – jump to label AAAA

en:

```
13A1 3A2762      LD     A,(#6227)    ; load A with screen number – label AAAA
13A4 FE02        CP     #02         ; are we on the conveyors?
13A6 2019        JR     NZ,#13C1    ; no, jump ahead – jump to label BBBB
;
13A8 3A0562      LD     A,(#6205)    ; load A with jumpman's Y position
13AB FE50        CP     #50         ; check jumpman at upper conveyor
13AD 2012        JR     NZ,#13C1    ; no, jump ahead – jump to label BBBB
;
13AF 3A0362      LD     A,(#6203)    ; load A with X position of jumpman
13B2 FE80        CP     #80         ; jumpman on left side of screen ?
```

```

13B4 3A8862      LD   A,(#6288)    ; load A with right retractable ladder status
13B7 3003        JR   NC,13BC    ; no, skip next step – jump to label CCCC
;
13B9 3A8062      LD   A,(#6280)    ; load A with left retractable ladder status
;
; check if ladder is all the way up
;
13BC FE00        CP   #00          ; ladder is all the way up? – label CCCC
13BE 2804        JR   Z,#13C4    ; yes, - take ladder – jump to label DDDD
13C0 C9          RET
;
13C1 7E          LD   A,(HL)      ; load A with broken ladder indicator – label BBBB
13C2 A7          AND  A
13C3 C0          RET  NZ         ; yes, return, we can't go down broken ladders
13C4 C3341B      JP   #1B34    ; jump back – label DDDD

```

Dat werkt.

Zie ineens dat bij de girders de onderste gebroken trap op hetzelfde segmentdeel zit als de ladders erboven. Dus dan staan ze recht boven elkaar. Dat moet aangepast worden.

Zie hier: de broken ladder zit links op het segment en de kleine hele ladder zit rechts op het segment. Dat klopt niet.

```

3B34: 01 63 D3 63 F8      ; bottom broken ladder
3B39: 00 DB D6 DB F2      ; bottom right ladder

```

Dus de bottom broken ladder moet naar het rechter deel van het segment:

```
3B34: 01 6B D3 6B F8      ; bottom broken ladder
```

Nu versie v0.90 opgeslagen.

Uitzoeken of de springs type keuze op basis van de lang/kort keuze: lijkt alsof de springs die het verste komen altijd lang zijn en degenen die dichtbij al vallen altijd kort zijn. Dat kan komen doordat dezelfde random waarde gebruikt wordt.

Het bepalen van de kleur van de springs (type) is gebaseerd op RngTimer1: #6018. Het bepalen van de lengte van de springs is gebaseerd op de call naar #0057. En dat is een combinatie van RngTimer1, RngTimer2 en FrameCounter. Dus dat zou variabel moeten zijn. Blijft toch makkelijk bovenin want als je een blauwe ziet kun je meteen gaan lopen naar de ladder en hoeft je niet te wachten totdat de spring voorbij is.

Toch nog eens goed kijken. Het varieert wel, maar minder dan ik zou verwachten. Kijken wat er gebeurt als we de kleur baseren op FrameCounter (#601A).

```
09EC 3A1A60      LD   A,(FrameCounter)  ; load A with random number
```

Dat varieert wel veel meer. Maar daardoor geen vaste ‘save’ zones meer op de trappen en op het uiteinde van de onderste girder en ook niet op de eerste trede van het trappetje bovenin. Bijna niet te doen zo. Ook met RngTimer2 is dat zo. Daarom op RngTimer1 houden.

Als Game Over dan wel de vuurketel, het vuur en de draaipunten weghalen.

Welke sprites zijn dat? Die moeten dan bij Game Over op Conveyors screen naar een hele lage X-waarde verplaatst worden (daarmee buiten scherm geplaatst en niet meer zichtbaar).

Vuurketel: #69FC - sprite 49
Vuur: #6A28 - sprite 43
Pulley1: #69EC - sprite 52
Pulley2: #69F0 - sprite D2

Kan dat kloppen? Checken met debugger. Ja dat klopt dat zijn de juiste sprite waarden.

Game Over tekst wordt hier weergegeven (normaal op twee plekken, maar nu nog maar enkel een one-player game):

```
1322 CD2618 CALL #1826 ; clear an area of the screen
1325 110003 LD DE,#0300 ; load task data for text #0 "GAME OVER"
1328 CD9F30 CALL #309F ; insert task to draw text
```

Aanpassen:

```
1322 CD1909 CALL #0919 ; call routine – call label AAAA
```

en:

```
0919 CD2618 CALL #1826 ; clear an area of the screen – label AAAA
;
091C 3A2762 LD A,(#6227) ; load A with screen number
091F FE02 CP #02 ; are we on the conveyors?
0921 C0 RET NZ ; no, return
;
0922 3E01 LD A,#01 ; load A with #01
0924 32FC69 LD (#69FC),A ; remove oil can sprite
0927 32286A LD (#6A28),A ; remove fire sprite
092A 32EC69 LD (#69EC),A ; remove pulley sprite
092D 32F069 LD (#69F0),A ; remove pulley sprite
0930 C9 RET ; return
```

Ja dat gaat goed zo. Nog wel eventjes ook testen in een ander scherm dan conveyors.

Hebben we alles vrijgegeven? Zo ja, dan niet veel ruimte meer. Kijken wat er eventueel nog mogelijk is.

#2679 t/m #26A5 - Is nog vrij. – 45 posities.

17 = vrijgeven #196B t/m #1976 – 10 posities.

#3C68 t/m #3C8A – is nog vrij – 36 posities.

#1385 t/m #138E – is nog vrij – 10 posities.

#0BB7 t/m #0BBD – is nog vrij – 7 posities.

Is allemaal niet veel ruimte. Eerst maar eens testen met wat we nu hebben. Als dat allemaal goed is zonder bugs, dan kijken wat we toevoegen. Muzieknotje met alternatieve muziek?

Weer starten met L=01 en dan testen vanaf begin.

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

Rivets is wel de grootste uitdaging. Wellicht toch nog wat veilige platformen maken?

Eventjes testen met rivets op L=05.

```
095E 05 73 3A 01 00 00 00
```

Extra platformen kunnen er nog na gedefinieerd worden. Want daar staat een data tabel voor .25 .50 .75 enz. Maar die wordt niet meer gebruikt.

```
3CEF 05 20 60 10 60 ; left side save zone  
3CF4 05 F7 60 D7 60 ; right side save zone  
3CF9 AA
```

En weer terug naar L=01.

Nu versie v0.91 opgeslagen.

Alles lijkt nu goed. Dan nu toch maar de sub level aanduiding toevoegen. Als later nog bugs op te lossen dan kan weer naar de vorige versie teruggegaan worden.

Sub level aanduiding is ook gedaan in DK Twisted Jungle. Daar kijken hoe gedaan.

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 C3683C JP #3C68 ; jump to additional code – jump to label AAAA
en

3C68 32C374 LD (#74C3),A ; draw level to screen (high byte) – label AAAA
;
3C6B 3E2C LD A,#2C ; load A with "-"
3C6D 328374 LD (#7483),A ; draw hyphen
;
3C70 3A2962 LD A,(#6229) ; load A with level number
3C73 FE00 CP #00 ; is the level number 0?
;
3C75 2804 JR Z, #3820 ; yes, skip next lines
3C77 3A2E62 LD A,(#622E) ; load A with number of Goofy Kongs to draw
3C7A 3C INC A ; increment A
3C7B 326374 LD (#7463),A ; draw screen in level number
;
3C7E C9 RET ; return

```

[23] #3C68 t/m #3C8A – is nog vrij – 36 posities.

Dit werkt nog niet omdat de variabele #622E 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 C3B70B JP #0BB7 ; jump to additional code, jump to label AAAA
en
0BB7 3608 LD (HL),#08 ; set game mode2 to 8 – label AAAA
;
0BB9 212E62 LD HL, #622E ; load HL with variable
0BBC 34 INC (HL) ; increase variable
;
0BBD C9 RET ; return
```

[7] #0BB7 t/m #0BBD – is nog vrij – 7 posities.

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 C38513 JP #1385 ; jump to additional code, jump to label AAAA
en
1385 32206A LD (#6A20),A ; store A into heart sprite X position – label AAAA
;
1388 3E00 LD A,#00 ; A := 00
138A 322E62 LD (#622E),A ; store A in variable
;
138D C9 RET ; return
```

[9] #1385 t/m #138E – is nog vrij – 10 posities.

Werkt nu wel goed.

Mogelijkheid tot tunen: de spell langer laten aanhouden zodat de fireballs iets langer stil staan.

Lengte wordt hier bepaald:

```
0C3F FEC0 CP #C0 ; A == 192 ? – label CCCC
0C4E FEC0 CP #C0 ; A == 192 ?
```

Even testen met #E0. Ja gaat goed. Iets meer voordeel van de spell en het stilstaan van de fireballs.

En misschien de rode barrel alleen doen vanaf L=05? Dan kan je eerst wennen aan de nieuwe girders en ook lastiger omdat daar de barrels nog niet zo goed stuurbare zijn.

De kleur van de eerste barrel wordt hier gezet:

```
0C69 FE81      CP    #81          ; is this the first crazy/blue barrel?  
0C6B C2523F    JP    NZ,#3F52    ; no, set color to blue  
;  
0C6E 3A4863    LD    A,(#6348)   ; get status of the oil can fire  
0C71 A7        AND   A          ; is the fire lit ?  
0C72 C2523F    JP    NZ,#3F52    ; yes, set color to blue  
0C75 C3573F    JP    #3F57      ; no, set color to red
```

Daar nog een extra check op het level (#6229) uitvoeren:

```
0C6E C37926    JP    #2679      ; jump to additional code – jump to label AAAA
```

en:

```
2679 3A2962    LD    A,(#6229)   ; load A with level nr  
267C FE05      CP    05          ; A < 5?  
267E DA523F    JP    C,#3F52    ; yes, jump back and set color to blue  
;  
2681 3A4863    LD    A,(#6348)   ; get status of the oil can fire  
2684 C3710C    JP    #0C71      ; jump back and process further
```

[14 posities] #2679 t/m #26A5 - Is nog vrij. – 45 posities.

Ja gaat goed. Geen red barrel meer bij L=01, 02, 03 en 04.

Nu versie v0.92 opgeslagen.

Nu even kijken wat er nodig is om het level en sublevel te registreren bij een nieuwe entry in de high score list. Niet heel veel ruimte over, maar wellicht net voldoende.

De tabelregel wordt initieel gevuld met veertien spaties:

```
; sets #61B7 through #61C4 to #10 (???)  
  
13EB 060E      LD    B,#0E      ; for B = 1 to #E  
  
13ED 3610      LD    (HL),#10    ; store #10 into memory at (HL)  
13EF 23        INC   HL          ; next HL  
13F0 10FB      DJNZ #13ED    ; next B
```

Nu dat niet doen. Eerst twee spaties, dan vier posities met het level en sub-level en dan nog acht spaties.

```
13EB 3610      LD    (HL),#10    ; store #10 into memory at (HL)  
13ED 23        INC   HL          ; next HL
```

13EE C38726	JP	#2687	; jump to additional code – jump to label AAAA
13F1 00	NOP		

en:

2687 3610	LD	(HL),#10	; store #10 into memory at (HL) – label AAAA
2689 23	INC	HL	; next HL
268A 23	INC	HL	
;			
268B 3A2962	LD	A,(#6229)	; load A with level nr
268E 010AFF	LD	BC,#FF0A	; B: = #FF, C := #0A (10 decimal)
;			
2691 04	INC	B	; increment B – label CCCC
2692 91	SUB	C	; subtract 10 decimal
2693 30FB	JR	NC,#1631	; not carry, loop again – jump to label CCCC
;			
2695 81	ADD	A,C	; add 10 back to A to get a number from 0 to 9
2696 77	LD	(HL),A	; store levelnr (singles) into memory at (HL)
2697 2B	DEC	HL	; next HL
;			
2698 78	LD	A,B	; load A with B (number of tens)
2699 77	LD	(HL),A	; store levelnr (tens) into memory at (HL)
269A 23	INC	HL	; next HL
269B 23	INC	HL	; next HL
;			
269C 362C	LD	(HL),#2C	; store hyphen into memory at (HL)
269E 23	INC	HL	; next HL
;			
269F 3A2E62	LD	A,(#622E)	; load A with board nr
26A2 3C	INC	A	; increment A
26A3 C37F3C	JP	#3C7F	; jump to additional code – jump to label DDDD

en:

3C7F 77	LD	(HL),A	; store board nr into memory at (HL)
3C80 23	INC	HL	; next HL
;			
3C81 0608	LD	B,#08	; for B = 1 to #08 – label BBBB
;			
3C83 3610	LD	(HL),#10	; store #10 into memory at (HL)
3C85 23	INC	HL	; next HL
3C86 10FB	DJNZ	#1640	; next B – jump back to label BBBB
;			
3C88 C3F213	JP	#13F2	; jump back

Ja dat gaat goed. Alleen wordt de naam er overheen geschreven. Dat nog aanpassen:

14CC 11F9FF	LD	DE,#FFF9	; load DE with offset of -8 (decimal)
14CF 19	ADD	HL,DE	; add offset
14D0 223A60	LD	(#603A),HL	; store result into ???

en

15D3 0605	LD	B,#05	; for B = 1 to #6 (11 decimal)
15D5 21E875	LD	HL,#75E8	; load HL with screen vram address
15D8 FD2A3A60	LD	IY,(#603A)	; load IY with ???
15DC 11E0FF	LD	DE,#FFE0	; load DE with offset of -#20

```
15DF 7E      LD    A,(HL)      ; load A with
15E0 FD7700  LD    (IY+#00),A  ; store
15E3 FD23    INC   IY          ; next
15E5 19      ADD   HL,DE      ; add offset
15E6 10F7    DJNZ #15DF      ; next B
```

Gaat nu allemaal goed. Alleen nog de header erboven aanpassen.

De kop moet dan ook waarschijnlijk aangepast worden:

```
367D: 9E 37      19      ; #379E "RANK SCORE NAME"
3790: 19 10 24 19 1D 15 10 10 30 03 00 31 10 3F 92 77  I.TIME.....
37A0: 22 11 1E 1B 10 10 23 13 1F 22 15 10 10 1E 11 1D  RANK..SCORE..NAM
37B0: 15 10 10 10 10 3F 72 77 29 1F 25 22 10 1E 11 1D  E.....YOUR.NAM
```

Dat past niet. Maar in de 'REGI TIME' ervoor zit extra ruimte. Maar dan past nog niet dus ook andere aanpassing: RANK aanpassen naar een #.

Dan dus de 3F op #379D één positie naar voren en de 9277 op #379E ook.

Dan begint RANK SCORE NAME op #379D, dus dan aanpassen:

367D 9D 37

En dan op de plaats van de N moet een hekje komen. Dus twee opschuiven in beeld: #7792 wordt dan #7752.

S C O R E L E V E L N A M E
#379D: 52 77 FA 10 10 23 13 1F 22 15 10 10 1C 15 26 15 1C 10 10 1E 11 1D 15 3F

Ja dat is goed. Alleen nog karakter voor # maken en invullen: #FA.

Ja dat gaat allemaal goed.

Invincibility weer uitzetten.

Nu versie v0.93 opgeslagen.

Dit wordt de finale versie v1.00.

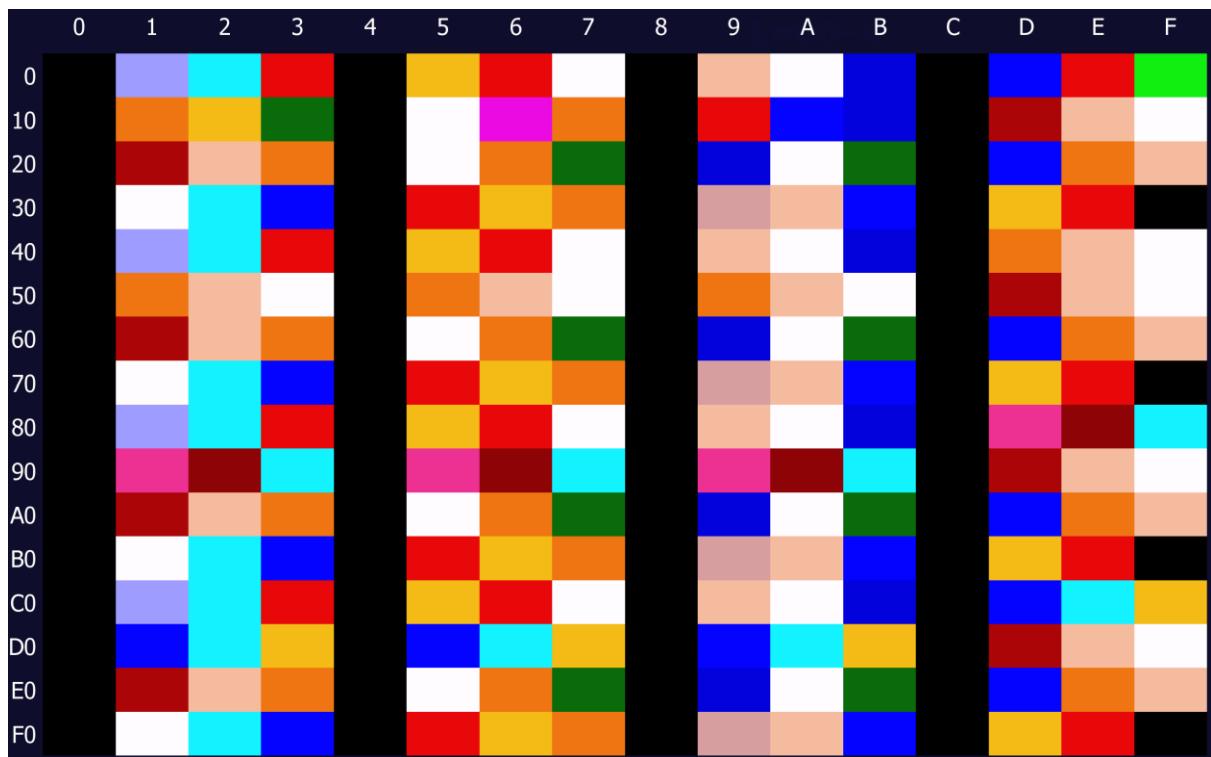
Eventueel kijken naar aanpassen kleuren en eventuele graphics om meer sfeer aan het thema te geven. Extra sprites met decoratie gaat niet omdat daar geen ruimte meer voor is in de code.

Eerst eens kijken naar de kleur van de girders en ladders in het barrels board en het elevators board.

Dit zijn de palettes:

```
;0 | 0  high score screen
;0 | 1  barrel and elevator stages
;1 | 0  pie factory stage
;1 | 1  rivet stage
```

Dit is het palette scherm:



De juiste kleurencombo's staan in #8D, #91, #95 en #99.

Maar welke is het dan? Kun je in het palette in de emulator ook aanpassingen doen? Dan kan je proefondervindelijk testen welke het is. Nee dat gaat niet.

Dan handmatig aanpassen: de bordeaux kleur vervangen door de blauw van #8B: #0F (J) en #0D (K).

Eerst in #8E (was J:#07 en K:#0F). Dat past alleen een deel aan. Misschien wel bij alle vier aanpassen. Dus dan ook in #92, #96 en #9A. Ja dat gaat goed. Wel gaaf effect door de kleuren van de girders zo.

De waarde van #8B in de roze kleur stoppen?

Paars = J:#07 en K:#0A

Blauw = J:#0F en K:#0D

Bordeauxrood = J:#07 en K:#0F

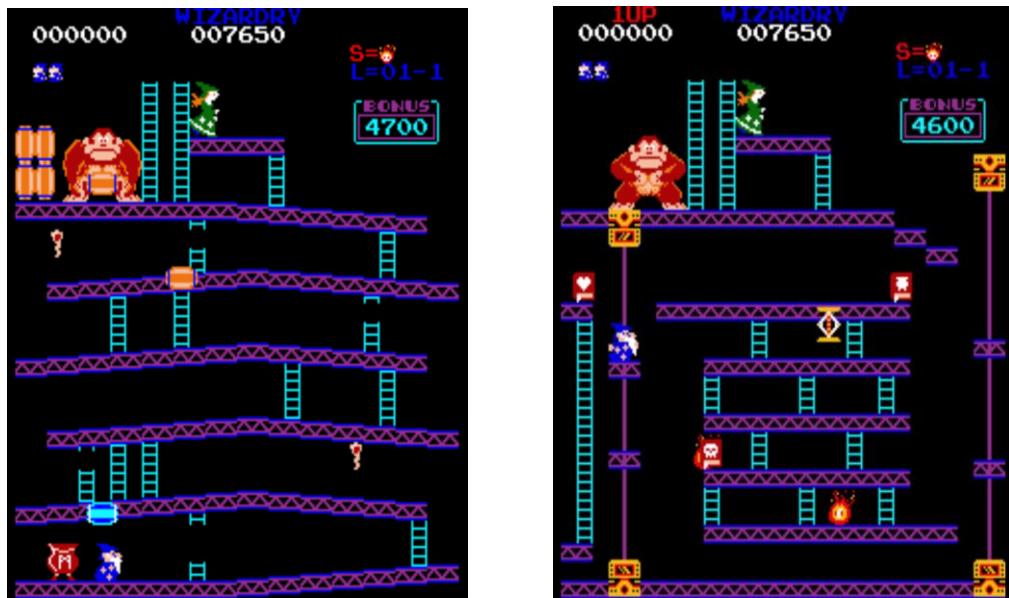
Lichtblauw = J:#0E en K:#00

De schuine stukken zijn nu paars: #8D, #91, #95 en #99

De accentstrepen bordeaurood maken: #8E, #92, #96 en #9A. Nee dat is niet zo mooi. Mooier als de accentstukken blauw zijn. Ja dat ziet er zeker beter uit.

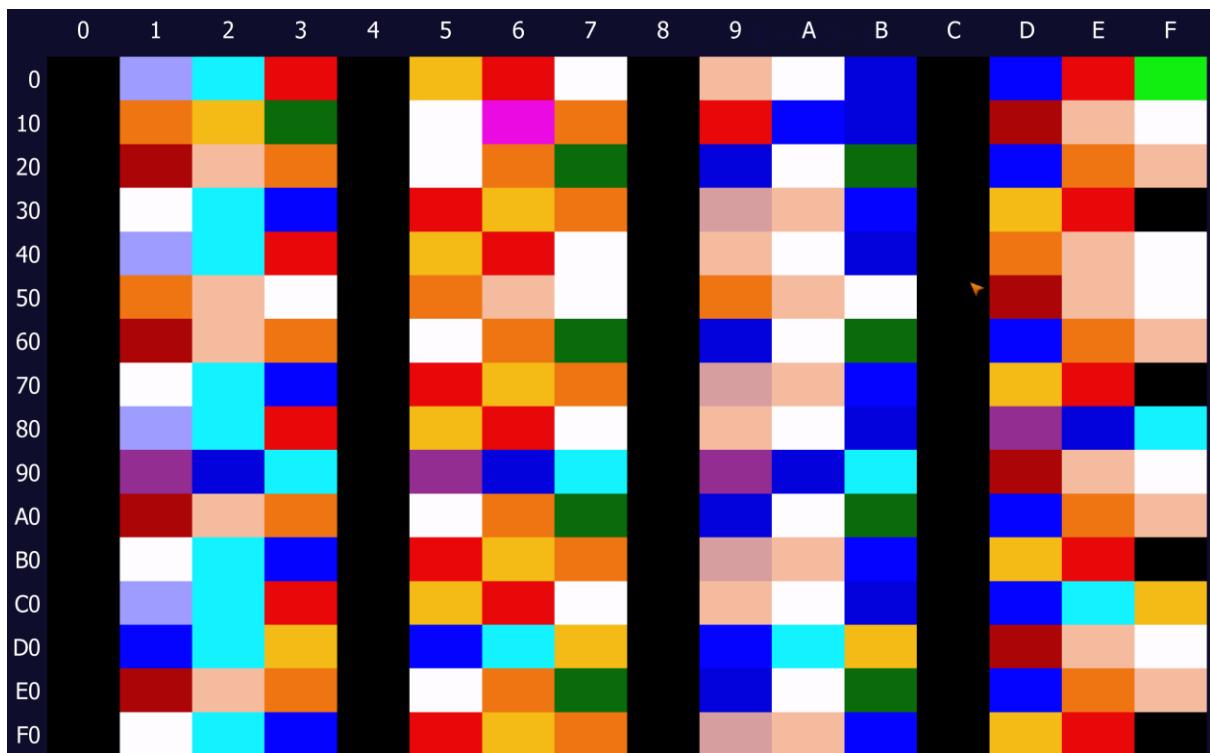
De sprites van de girders zodanig aanpassen dat de buitenkant blauw is en het schuine rasterwerk paars is.

Ziet er nu zo uit:



Nu versie v1.00-1 opgeslagen.

Dan nu kijken naar de kleuren van het rivets screen.



Dat lijken de kleuren in #CD, #D1, #D5 en #D9.

Iets met donkergoen maken?

Donkerblauw: J:#0F en K:#0C

Lichtblauw: J:#0E en K:#00

Donkergeel: J:#00 en K:#0B

Donkergroen: J:#0F en K:#03

Donkerbruin: J:#05 en K:#0F

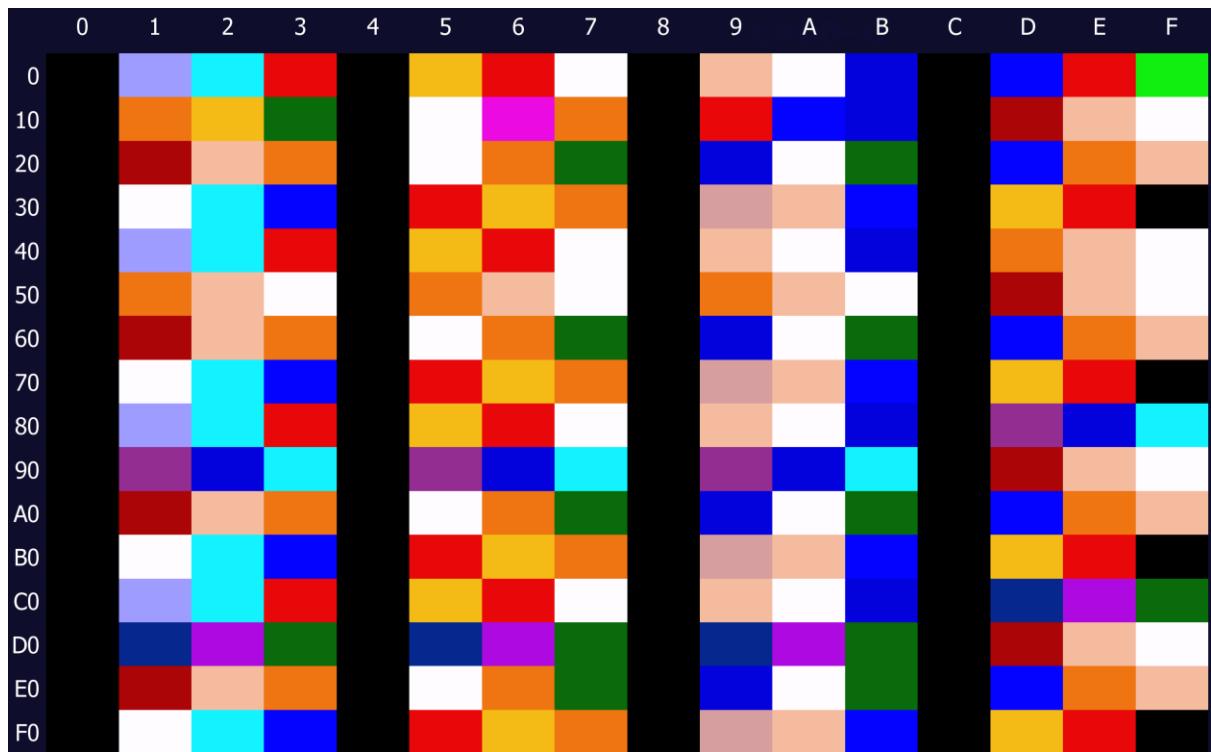
Aangepast: de donkergele trappen zijn groen geworden en de girders dezelfde kleuren paars en blauw als in de barrels en elevators.

Ziet er nu zo uit:



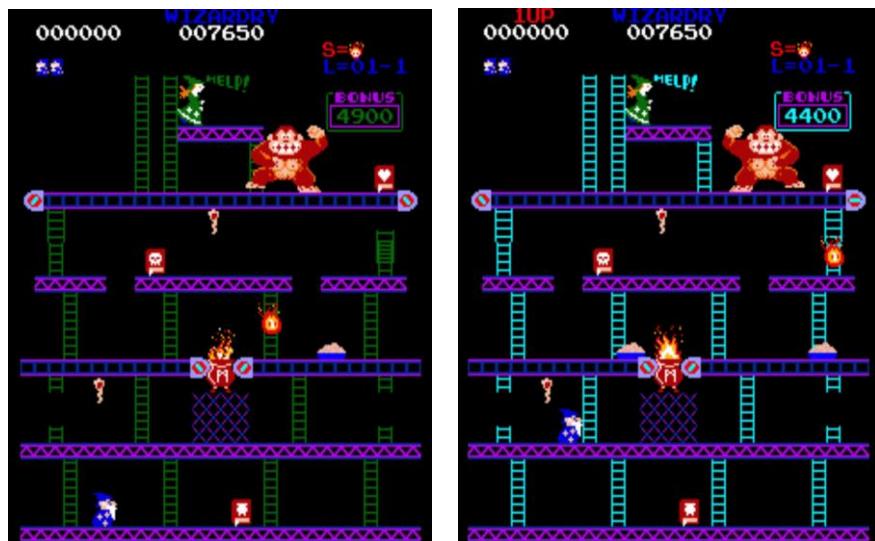
Nu versie v1.00-2 opgeslagen.

Nu dan nog de kleuren van het pies board aanpassen. Lastig kiezen.



Dat lijken de kleuren in #4D, #51, #55 en #59.

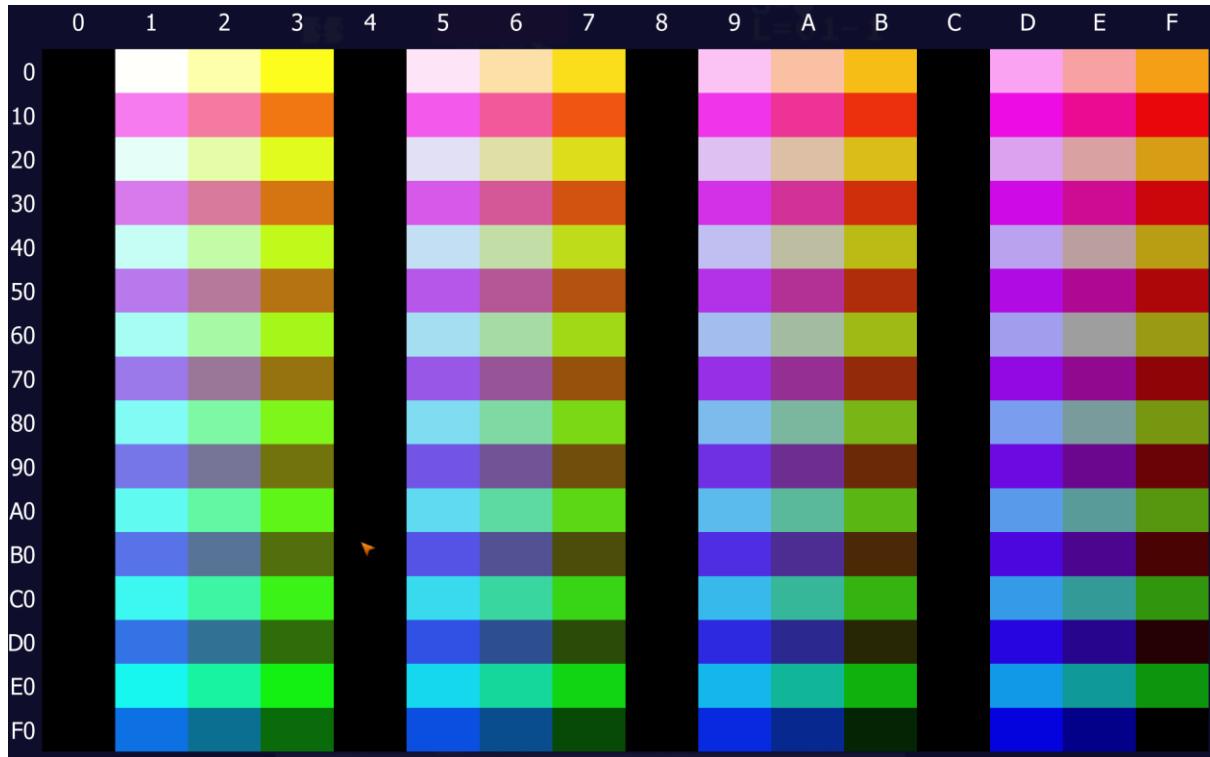
Ook daar maar dezelfde blauw, paars en groen toepassen. Andere kleuren trappen zijn allemaal niet zo mooi.



Maar met de lichtblauw kan ook. Maar dan zou je ze haast allemaal met hetzelfde kleuren palet moeten doen.

De gifgroen van de credits ook veranderen in donkergruen. Daarmee ook het selectiescherm met de letters ook donkergruen geworden. Wellicht nog wel de selectiemarker (sprite) een andere kleur geven. Geel bijvoorbeeld?

Kunnen we alle mogelijke kleuren weergeven door als test de palettes via een patroon te vullen?
Dan kijken of er wellicht nog interessante kleuren tussen zitten.



Alleen de zwarte kolommen worden niet weergegeven, maar geeft een interessant kleurenpalette zo.

Lichtblauw van de trappen is bijvoorbeeld 0E,00, maar een minder neon versie daarvan zou bijvoorbeeld 0E,09 zijn.

Daar dan de trappen van barrels en elevators mee vullen: #8E, #91, #95 en #99.

Neon groen van de credits is bijvoorbeeld ?,?, maar een minder neon versie daarvan zou bijvoorbeeld 0C,0B zijn.

Daar de trappen van pies mee vullen: #4D, #51, #55 en #59. Ja dat ziet er wel goed uit.
Ook toepassen op rivets: in #CD, #D1, #D5 en #D9.

Ja dat is allemaal subtiel mooi. Alleen nog kijken of voor pies level toch nog een afwijkende kleur trappen? Nee, dat niet. Gewoon de lichtere groen laten.

Ook de credits groen deze lichtere variant maken.

Aanpassen kleurcode voor de selectiebox bij invoeren van de naam.

Dat zit hier:

```

; sets the sprite to the square selector for initials entry
; called from #14D9 and #1533

15FA D5      PUSH   DE          ; save DE
15FB E5      PUSH   HL          ; save HL
15FC CB21    SLA    C          ;
15FE 210F36  LD     HL,#360F    ; start of table data
1601 09      ADD    HL,BC
1602 EB      EX    DE,HL
1603 217469  LD     HL,#6974
1606 1A      LD     A,(DE)     ; load A with table data
1607 13      INC    DE          ; next table entry
1608 77      LD     (HL),A     ; store
1609 23      INC    HL          ; next location
160A 3672    LD     (HL),#72
160C 23      INC    HL
160D 360C    LD     (HL),#0C
160F 23      INC    HL
1610 1A      LD     A,(DE)
1611 77      LD     (HL),A
1612 E1      POP    HL          ; restore HL
1613 D1      POP    DE          ; restore DE
1614 C9      RET

```

Waarschijnlijk zet instructie op #160D de kleur. Nu #0C. Kleur #01 gegeven: rood met oranje.

Kijken naar de pulley kleuren. Wellicht hetzelfde maken als de kleuren van de elevator shafts?

; 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

```

Kleur is nu #00. Aanpassen naar #01? Ja dat is mooi.

Wellicht de schuifladders ook aanpassen naar geel en/of rood?

```

; 2 ladder sprites for conveyors
; 46 = ladder

3E1C: 23 46 03 68          ; ladder at 23, 68
3E20: DB 46 03 68          ; ladder at DB, 68

```

Kleur naar #01 aanpassen.

Toch ook nog even proberen met groene trappen in barrels en elevators. Dan allemaal wel hetzelfde. #8F en de volgende drie allemaal naar 0C,0B aanpassen. Nee dat is het niet. Toch de lichte kleur blauw houden.

Ja, allemaal aangepast.

De versie aanpassen naar versie v1.01.

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

Nu versie v1.00-3 opgeslagen.

Dit wordt de finale versie v1.01.

Bij meerdere testplays blijkt dat bij level 3 (en hoger?) de onderste fireball bij elevators een freezer is en dat deze soms in een eindeloze loop rechts op de onderste girder van het middenstuk terecht komt. Daar kun je dan niet meer langs. Dat is unfair. Uitgebreider testen en aanpassen: 1) omdraaien 1 en 2 èn 2) de tweede fireball niet gebruiken (is een freezer) en i.p.v. daarvan het derde fireball slot gebruiken.

Eerst even starten op level 5 en als eerste elevator bord en invincible.

095E 05 73 3A 01 00 00 00 ; #3A65 is start of table data for screens/levels

Was 01 65 3A 01 00 00 00.

19B3 00 00 00 NOP, NOP, NOP

Was CD 08 28

Startveld level 5 aanpassen:

3A73 03 02 01 03 01 04

Was 01 02 01 03 01 04

Aanpassen fireballs:

Fireballs worden hier geïnitialiseerd:

```

; set up the 2 fireballs

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

; set up 2nd fireball

1101 DD362001 LD    (IX+#20),#01  ; set fire active
1105 DD3623EB LD    (IX+#23),#EB  ; set fire X position
1109 DD362EEB LD    (IX+#2E),#EB  ; set fire X position
110D DD362560 LD    (IX+#25),#60  ; set fire Y position
1111 DD362F60 LD    (IX+#2F),#60  ; set fire Y position

```

Fireball 1: X = 78, Y = A8

Fireball 2: X = 98, Y = D8

Nu:

DD 21 00 64

DD 36 00 01 Fire 1 active

DD 36 03 78 X

DD 36 0E 78 X

DD 36 05 A8 Y

DD 36 0F A8 Y

DD 36 20 01 Fire 2 active

DD 36 23 98 X

DD 36 2E 98 X

DD 36 25 D8 Y

DD 36 2F D8 Y

Aanpassen:

DD 36 00 01 Fire 1 active

DD 36 03 98 X

DD 36 0E 98 X

DD 36 05 D8 Y

DD 36 0F D8 Y

DD 36 40 01 Fire 3 active

DD 36 43 78 X

DD 36 4E 78 X

DD 36 45 A8 Y

DD 36 4F A8 Y

Ja dat gaat goed. Allebei nu geen freezer. Maar nu kijken of het niet moeilijker geworden is.

Invincibility weer aan:

19B3 CD 08 28

Bij testen ook een keer twee elevators over elkaar heen. Dan dus de spacing te dicht op elkaar:

```
0BAD 3A1960      LD    A,(RngTimer2) ; load A with a random number – label AAAA
0BB0 E63F        AND   #3F           ; number between 0 en #3F
0BB2 C601        ADD   A,#01         ; add #01
0BB4 77          LD    (HL),A       ; reset elevator counter
;
0BB5 35          DEC   (HL)         ; decrease elevator counter – label BBBB
0BB6 C9          RET
```

Kan dat komen wanneer random getal bijvoorbeeld 0 is en dan $+1 = 1$ pixel verschil. Wat als de ADD verhoogt wordt tot #10 (dan altijd afstand minimaal 8 pixels ertussen)? Ja dat gaat goed. Zelfs soms iets verder uit elkaar. Vooralsnog zo houden.

Het start veld level 5 weer aanpassen:

Startveld level 5 aanpassen:

3A73 01 02 01 03 01 04

Startlevel weer aanpassen:

095E 01 65 3A 01 00 00 00.

Versie verhogen naar 1.02.

#373E: 3D 76 26 01 2B 00 02 10 3F (V1.02)

Invincibility weer eventjes aan en de velden/levels checken.

Invincibility weer uit.

19B3 CD 08 28

Nu versie v1.01-1 opgeslagen.

Dit wordt de finale versie v1.02.

Bij het maken van Donkey Kong RNDMZR bleek dat high scores van levels met een tiental (dus 10 en hoger) niet goed afgehandeld worden. Er wordt niet gevraagd om de initialen en er wordt meteen een reeks nullen en spaties getoond met een jumpman linksboven en linksonder in het scherm (half half). Dit zit dus ook in deze versie van Donkey Kong Wizardy.

Het gaat dan om dit stuk code:

```

268B 3A2962      LD    A,(#6229)    ; load A with level nr
268E 010AFF      LD    BC,#FF0A     ; B: = #FF, C := #0A (10 decimal)
;
2691 04          INC   B           ; increment B – label CCCC
2692 91          SUB   C           ; subtract 10 decimal
2693 30FB        JR    NC,#1631    ; not carry, loop again – jump to label CCCC
;
2695 81          ADD   A,C         ; add 10 back to A to get a number from 0 to 9
2696 77          LD    (HL),A     ; store level nr (singles) into memory at (HL)
2697 2B          DEC   HL          ; next HL
;
2698 78          LD    A,B         ; load A with B (number of tens)
2699 77          LD    (HL),A     ; store levelnr (tens) into memory at (HL)

```

De JR op #2693 springt één teveel terug (dus naar 2690 i.p.v. 2691). Dit aanpassen door het te veranderen in 30FC.

Ja dat gaat nu wel goed.

De versie aanpassen naar versie v1.03.

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

Nu versie v1.02-1 opgeslagen.

Dit is de finale versie v1.03.