

Donkey Kong Super Mario Bros crossover

Idee is een romhack waarin allerlei spelementen en enemies van Super Mario Bros opgenomen worden.

Mogelijke aanpassingen:

- Werelden i.p.v. levels en dan 1-1 t/m 8-4.
- Na wereld 8-4 heb je het spel uitgespeeld.
- Score dan met kroontje markeren in high score list.
- Pijpen met planten die eruit komen.
- Goomba's waar je op kunt springen.
- Drie munten pakken voordat level afgerond kan worden.
- Vliegende vissen in boog van onderuit het scherm.
- Kanon met zwarte kogel die afgeschoten wordt.
- Liften in stijl van SMB met lava eronder.
- Bowser die hamers gooit?
- Ergens op een lastige plek een 1UP plaatsen in een vraagtekenblok.
- Einde van een board door springen naar vlaggenstok: hoogte bepaald bonus.
- Donkey Kong in twee boards vervangen door Bowser.
- Pauline in die boards vervangen voor Peach.
- Vallende blokken die weer omhoog gaan.
- Vraagteken blokken met een bonus.
- Sterren i.p.v. hamers en wanneer gepakt tijdelijke onsterfelijkheid.
- ~~Bovenin een elektriciteitspaal en wanneer vonken dan tijdelijk licht uit.~~
- ~~Random opbouw van de velden met vaste seed zodat voor iedereen hetzelfde.~~
- Evenwichts platformen aan katrollen.
- Blokken om muntjes te tanken en dus ook punten.
- SMB logo op de titelpagina.
- Ergens in een piesachtige wereld de Kong ladder klimbaar maken naar warpzone?

Eerst maar eens een gevoel krijgen door een begin te maken.

De barrels vervangen door draaiend schild met punten:



Dat in turaco aangemaakt.

Maar door de animatie van de tonnen draait deze de hele tijd op zijn kop.

Waar zit die animatie? Eerst uitzetten en daarna kijken of deze animatie ook gedaan kan worden door een aantal verschillende sprites na elkaar weer te geven.

Dat lijkt hier te zitten:

```
; else animate the barrel

1FD5 DD7E07 LD A,(IX+#07) ; yes, Load A with #07 indicator = sprite used
1FD8 EE01 XOR #01 ; toggle bit 1
1FDA DD7707 LD (IX+#07),A ; store back in #07 indicator = toggle sprite
1FDD 3E04 LD A,#04 ; A := 4

1FDF DD770F LD (IX+#0F),A ; store A into barrel #0F counter (from 4 to 1)
1FE2 C3BA21 JP #21BA ; jump ahead
```

Even testen door de toggle uit te zetten: 1FD8 en 1FD9 NOP maken.

Klopt ten dele: dit zet de sprite animatie van de wild barrels en de barrels op de ladders uit. Maar niet de animatie van de rollende barrels.

Even met de debugger kijken waar de sprite van de tweede barrel getoggled wordt. Dat is als het goed is de #6727 waarde. Daar een watchpoint op zetten.

Wordt gedaan in #23E6 en #23FD: toggle.

En aanpassen naar rollen vanaf trap: #21B2, #1FDA en #1FC0: change, rol, change.

```

23DE DD7E0F LD A,(IX+#0F) ; Load A with +#F property of barrel (counts from 4 to 1 over and over)
23E1 3D DEC A ; decrease by one. did counter go to zero?
23E2 C20324 JP NZ,#2403 ; if not, jump ahead, store new timer value and return

23E5 AF XOR A ; A := 0
23E6 DDCB0726 SLA (IX+#07) ; shift left the barrel sprite status, push bit 7 into carry flag
23EA 17 RLA ; rotate in carry flag into A
23EB DDCB0826 SLA (IX+#08) ; shift left the other barrel color, push bit 7 into carry flag
23EF 17 RLA ; rotate in carry flag into A
23F0 47 LD B,A ; copy result into B
23F1 3E03 LD A,#03 ; A := 3
23F3 B1 OR C ; bitwise OR with C
23F4 CD0930 CALL #3009 ; ???
23F7 1F RRA ;
23F8 DDCB081E RR (IX+#08) ; rotate right the barrel's color
23FC 1F RRA ;
23FD DDCB071E RR (IX+#07) ; Roll these values back
2401 3E04 LD A,#04 ; A := 4

2403 DD770F LD (IX+#0F),A ; store A into timer
2406 C9 RET ; return

```

Dit stuk checkt eerst de teller in #0F. Om de vier frames wordt de barrel sprite aangepast.

Maar het doet meer. Het doet ook iets met de barrel color. Weet van een eerdere romhack dat daar wat mee was, maar niet meer wat precies.

Je zou ook kunnen zeggen dat je dit hele stuk vervangt door een stuk dat alleen de sprite aanpast d.m.v. sequentieel een set van vier sprite te doorlopen. En die kunnen dan voor alle barrels (rollend, op trap en wild) gelden. Dat maakt het een stuk eenvoudiger.

Eerst de sprites aanmaken.

De vier sprites zijn aangemaakt in posities #15, #16, #17 en #18.

Volgens mij wordt in de kleur ook nog iets bepaald van de oriëntatie van de sprite. Maar dat is nu niet relevant. Dus even de kleur de kleur laten en de code vervangen door code die door de vier sprites heen loopt:

```

23E5 DD7E07 LD A,(IX+#07) ; load sprite value in A
23E8 3C INC A ; select next sprite
23E9 FE19 CP #19 ; sprite #19 selected?
23EB 2002 JR NZ,#AAAA ; no, skip next step – jump to label AAAA
;
23ED 3E15 LD A,#15 ; reset to sprite #15
;
23EF DD7707 LD (IX+#07),A ; save new sprite value - AAAA
;
23F2 3E04 LD A,#04 ; set new timer
;
23F4-2402 NOP's

```

```
;  
2403 DD770F LD (IX+#0F),A ; store A into timer  
2406 C9 RET ; return
```

Ja, dat gaat goed. Bij rollen zie je nu een draaiende beweging in de sprite. Gaat alleen nog niet overal goed. Hier nog verder naar kijken.

Nu versie v0.01 opgeslagen.

Kijken blauwe barrel. Die gaat nu niet goed. Dat komt omdat de blauwe barrel start met een andere sprite (#19) en daarvandaan gaat doorlopen en dus niet gereset wordt. Deze moet ook op #15 starten.

Dat gebeurt hier:

```
; 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
```

Dit override de eerder gezette sprite value van #15. Je kan deze dus weghalen of expliciet overriden met #15. Dat laatste maar doen.

```
2D09 DD360715 LD (IX+#07),#19 ; set sprite for blue barrel
```

Dat gaat nu goed.

Alleen wanneer de blue barrel een ladder neemt, dan toch weer een offset naar een waarde buiten de range. En soms wanneer een gewone barrel een ladder neemt zie je een barrel i.p.v. een schild. Geen idee hoe dat kan.

Kijken naar de sprite aanpassingen die bij een ladder gedaan worden. In principe hoeft dat nu helemaal niet meer.

In ieder geval de sprites kleiner gemaakt en zodanig aangepast dat de vertikale uitrekking minder wordt en de sprites omhoog geschoven zodat de schilden nu netjes op de girders rollen.

Ze draaien wel erg snel. Misschien visueel mooier als ze minder snel draaien.

```
23F2 3E0A LD A,#0A ; set new timer
```

Met #0A een leuke visuele draaisnelheid.

Nog steeds kijken naar de sprite aanpassingen die bij een ladder gedaan worden. In principe hoeft dat nu helemaal niet meer.

Dat gebeurt hier: #21B2, #1FDA en #1FC0: change, rol, change.

```
21B2 DD3407 INC (IX+#07) ; increase Barrel's deployment/animation status
21B5 DDCB02C6 SET 0,(IX+#02) ; set barrel to go down the ladder
21B9 C9 RET ; return
```

Aanpassen:

```
21B2 000000 3x NOP
```

```
1FD5 DD7E07 LD A,(IX+#07) ; yes, Load A with #07 indicator = sprite used
1FD8 EE01 XOR #01 ; toggle bit 1
1FDA DD7707 LD (IX+#07),A ; store back in #07 indicator = toggle sprite
1FDD 3E04 LD A,#04 ; A := 4

1FDF DD770F LD (IX+#0F),A ; store A into barrel #0F counter (from 4 to 1)
1FE2 C3BA21 JP #21BA ; jump ahead
```

Aanpassen:

```
1FDA 000000 3x NOP
```

```
1FB9 DD7E15 LD A,(IX+#15) ; load A with Barrel #15 indicator, zero = normal barrel, 1 = blue barrel
1FBC 07 RLCA ; roll left twice (multiply by 4)
1FBD 07 RLCA
1FBE C615 ADD A,#15 ; add #15
1FC0 DD7707 LD (IX+#07),A ; store into +7 indicator = sprite used
```

```
1FC0 000000 3x NOP
```

Ja dat gaat allemaal goed.

Dan alleen nog de stapel met schilden aanpassen. Die liggen nu nog raar uit elkaar.



Dat zit hier:

```
; data used for the barrel pile next to kong  
; called from #0FD7  
  
3DDC 1E 18 0B 4B      ; first barrel  
3DE0 14 18 0B 4B      ; second barrel  
3DE4 1E 18 0B 3B      ; third barrel  
3DE8 14 18 0B 3B      ; fourth barrel
```

Aangepast:

```
3DDC 1C 18 0B 4F      ; first spine  
3DE0 0E 18 0B 4F      ; second spine  
3DE4 1C 18 0B 43      ; third spine  
3DE8 0E 18 0B 43      ; fourth spine
```

Ziet er nu zo uit:



Nu versie v0.02 opgeslagen.

Dan nu kijken of de girders vervangen kunnen worden. Stenen structuur gaat denk ik niet goed met de hellende girders. Beter denk ik om de standaard grondblokken te gebruiken. Maar dan alleen de bovenkant, want de girders zijn maar 8 pixels dik.

Nee dat gaat niet goed. De grond is veel dikker en de blokken zijn 16x16. Dus de bovenkant vergt dat twee verschillende graphics naast elkaar. Voor de hellende is dat te lastig.

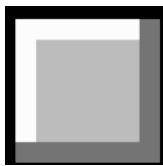
Daarom toch stenen gebruiken.

De graphics zijn aangepast.

Nu versie v0.03 opgeslagen.

Dan nu de kleuren van de achtergrond graphics aanpassen.

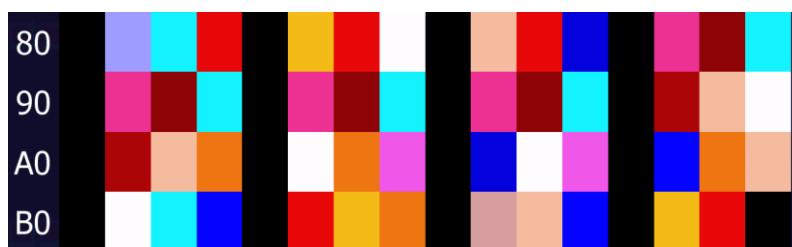
Voorbeeld uit Super Mario Bros:



Kleurcodes (2J/2K) via vergelijk met alle mogelijke DK kleuren:

- wit: #00 / #00
- lichtgrijs: #04 / #0A
- donkergrijs: #09 / #02

De roze moet wit worden, de bordeauxrood moet lichtgrijs worden en de blauw moet donkergrijs worden.



So the changes in the four ‘blocks’ will be:

C-2J: #00 / #04 / #09

C-2K: #00 / #0A / #02

Ja dat is zo goed.

Nu versie v0.04 opgeslagen.

Dan nu de kleuren van de schilden aanpassen naar rood en blauw.



Kleurcodes (2J / 2K) voor het rode schild:

- rood: #01 / #07
- wit: #00 / #00
- bruin: #00 / #0B

Kleur nu is:



De blauw moet wit worden, de roze moet bruin worden en de bruin moet rood worden.

Dat zijn de kleuren op #AD (blauw), #AE (bruin) en #AF (roze).

Kleurcodes (2J / 2K) voor het rode schild:

- #AD: #00 / #00 (wit)
- #AE: #01 / #07 (rood)
- #AF: #00 / #0B (bruin)

Om een blauw schild te maken, dezelfde codes gebruiken maar dan met blauw i.p.v. rood.

Blauwe schild zijn de kleuren op #B1, #B2 en #B3.

Kleurcodes (2J / 2K) voor het blauwe schild:

- #B1: #00 / #00 (wit)
- #B2: #0B / #08 (blauw)
- #B3: #00 / #0B (bruin)

Dat is nu goed.

Ziet er zo uit:



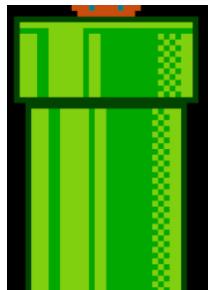
Nu versie v0.05 opgeslagen.

En dan de oil can veranderen naar een groene pijp en daar geen vuur op laten branden. Die sprites kunnen dan eventueel voor wat anders gebruikt gaan worden.

En dan de vuurballen aanpassen naar een andere enemy. En die laten bewegen. Wellicht richting afhankelijke weergave en animatie van de beweging.

En dan de hamer veranderen naar een onsterfelijkheidsster.

De oil can aanpassen naar een groene pijp:



De kleuren nu zijn:



Aanpassen naar de pijpkleuren (2J / 2K):

- donkergroene rand: #0F / #07
- groen: #0E / #0B
- lichtgroen: #08 / #07

De oil can gebruikt echter de kleurcode van het blauwe schild: #0C (#B1 / #B2 / #B3) in de tabel.

Daar dus een andere voor gaan gebruiken en de kleur overeenkomstig aanpassen.



De volgorde binnen het blok is: groen / lichtgroen / donkergroen.

Kleurcode #00 wordt niet voor een sprite gebruikt, alleen maar voor de bovenste regel (de rode kleur van de teksten). Deze gebruiken en later eventueel de kleur van de tekst aanpassen.

Dus dat is dan:

- #81: #0E / #0B (groen)
- #82: #08 / #07 (lichtgroen)
- #83: #0F / #07 (donkergroen)

En dan moet de oil can de kleur code #00 krijgen.

Dat gebeurt hier:

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

Ja dat is goed nu.

Nu versie v0.06 opgeslagen.

Eerst even de visuele zaken verder oppakken. Daarna pas de aanpassingen qua speelbaarheid.

Welke enemy uit SMB gaan gebruiken i.p.v. de fireballs? In ieder geval één pakken waar je meteen door dood gaat. Dus niet een Goomba want daar kun je normaal gesproken op springen, maar dat

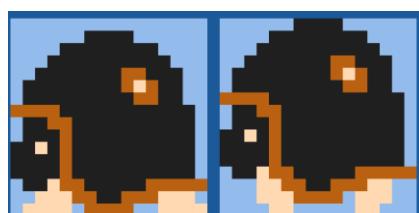
kan hier niet. Maar op zo'n beetje alles kun je springen. Daarom toch kiezen voor een Koopa. En dan later eventueel kijken of het mechanisme aangepast dient te worden.

Een koopa is wel erg lastig om ook te schalen. en dan is hij nog te groot voor een enkele sprite. Daarom gaan voor een buzzy beetle.

De sprite aangemaakt.



Nu de kleuren nog goedmaken.



Aanpassen naar de kleuren (2J / 2K):

- zwart - #0F / #0B
- bruin - #05 / #07
- roze - #00 / #06

Dat moet in de plaats van de fireball kleuren ingesteld worden. Dat is kleurcode #01 en dus #85, #86 en #87.



Dus aanpassen:

- #85: #00 / #06
- #86: #05 / #07
- #87: #0F / #0B

Ja, dat ziet er goed uit. Kijkt zelfs altijd in de richting waarin hij beweegt. Wel zo dat de score hierdoor ook aangepast is, dat moet nog goed gezet worden.

De buzzy beetle gaat nog wel op en neer en dat is in het origineel niet zo. Dat eruit halen.

Waar wordt dat op en neer bewegen gedaan. Met debugger kijken met watchpoints op #6405 en #640F. Zijn beiden Y-posities. Eentje voor de sprite en andere voor de logica, waarbij het op en neer bewegen alleen visueel is en de logica niet beïnvloed.

Zit op #3276 en #33D5. #33D5 lijkt de aanpassing van de Y-positie voor de schuine girders.

Dat zit mogelijk hier:

```
3261 DD7713 LD (IX+#13),A ; Store updated index into Y-position adjustment table
3264 1600 LD D,#00 ; \ Index the Y-position adjustment table using +#13 to get in A the amount to adjust the Y-position by to
3266 5F LD E,A ; | make the fireball bob up and down
3267 217A3A LD HL,#3A7A ; |
326A 19 ADD HL,DE ; |
326B 7E LD A,(HL) ; /
; 3A7A: FF 00 FF FF FE FE FE FE FE FE FE FE FF FF FF 00
```

Lijkt erop dat de datababel op #3A7A de offsets geeft. Eens kijken als die allemaal op nul gezet worden. Ja dat gaat goed.

Nu versie v0.07 opgeslagen.

Dan nu de hamer vervangen door een star.

Sprite aangemaakt. Alleen de hamer moet kleurcode #0B krijgen om de ster de goede kleuren te geven.

Dat zit hier:

```
3E08: 1E 07 ; 1E is the hammer sprite value. 07 is hammer color
3E0A: 03 09 ; ???
3E0C: 24 64 ; position of top hammer for girders. 24 is X, 64 is Y
3E0E: BB C0 ; bottom hammer for girders at BB, C0
```

Ja dat is nu goed.

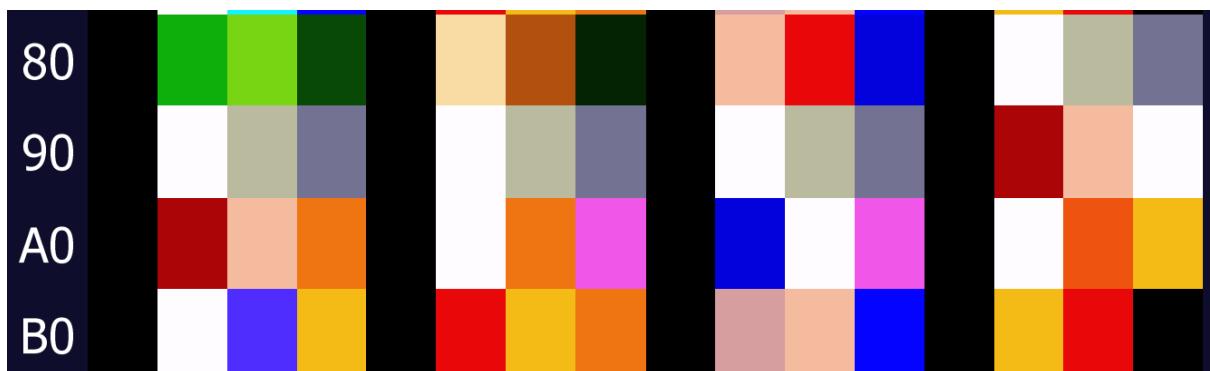
Nu versie v0.08 opgeslagen.

De verschillende sprite aanpassingen zijn nu gedaan.

Nu nog de volgende zaken aanpassen:

- 1UP / 2UP en High Score teksten in een passende kleur.
- High score tekst aanpassen naar titel.
- Score in een passende kleur.
- Level aanduiding naar code 1-1 t/m 8-4 in een passende kleur.
- Geen vuur boven de groene pijp. Animatie daarvan weg en vrijgeven.
- Hamer omzetten naar onsterfelijkheid a la pacman, jumpman knipperen en kortere tijd.

De kleuren zijn de derde kleur uit een blok.



De scores (de tweede lijn) in wit maken. Dus de hele tweede lijn omzetten naar kleurcode #07.

De 1UP en 2UP teksten (eerste lijn, zijkanten) in grijs maken. Dus die stukken omzetten naar kleurcode #03.

De High Score tekst (eerste lijn, midden) in geel maken. Dus dat stuk omzetten naar kleurcode #0B.

De level aanduiding tekst (vierde lijn rechter zijkant) in wit maken. Dus dat stuk omzetten naar kleurcode #07.

De High Score tekst (nu DUEL) aanpassen naar SMBros. Daartoe wellicht kleine letters nog toevoegen? Of SMBROS? Nee met kleine letters is leuker.

Kleine letters zitten in #A0, #A1 en #A2.

High Score zit hier:

```
36A0: 10 30 32 31 3F 94 76 20 1C 11 29 15 22 10 30 33 .<I>...PLAYER.<2  
36B0: 31 3F 80 76 18 19 17 18 10 23 13 1F 22 15 3F 9F >...HIGH.SCORE..  
36C0: 75 13 22 15 14 19 24 10 10 10 10 3F 5E 77 18 1F .CREDIT.....HO
```

Vervangen door: 80 76 10 10 23 1D 12 A0 A1 A2 10 10 3F.

Dat is nu allemaal goed.

Nu versie v0.09 opgeslagen.

Dan nu de level aanduiding aanpassen naar world aanduiding. L=01 vervangen door W1-1.

Dat stuk zit 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

```

Aanpassen dat zowel level als board weergegeven worden. Je hoeft niet te klooien met de tientallen want beiden blijven onder de 10 omdat we het spel eindigen na finishen van world 8-4.

```

06D7 11E0FF LD DE,#FFEO ; load DE with screen offset
06DA 21E374 LD HL,#74E3 ; load HL with screen location for "W"
06DD 3627 LD (HL),#27 ; draw "W"
;
06DF 3A2962 LD A,(#6229) ; load A with level #
06E2 19 ADD HL,DE ; load HL with next screen location
06E3 77 LD (HL),A ; draw level #
;
06E4 19 ADD HL,DE ; load HL with next screen location
06E5 362C LD (HL),#2C ; draw "-"
;
06E7 3A2762 LD A,(#6227) ; load A with board #
06EA 19 ADD HL,DE ; load HL with next screen location
06EB 77 LD (HL),A ; draw level #
;
06EC C9 RET ; return

```

Dat ziet er goed uit.

Nu versie v0.10 opgeslagen.

En dan nu kijken jumpman invincible maken en dan de hamer code vrijgeven wellicht. Dit is ook al gedaan in DK Pacman Crossover. Daar de code van als inspiratie gebruiken.

Ja puur ter inspiratie, want dat werkt daar wel net anders. Je pakt geen hamer maar een power pellet en de afhandeling daarvan is allemaal specifieke code.

Nu wel je de hamer pakken. Dan start de hamer periode. Daarin moet er niet gehamerd worden, maar moet de collision detectie uit en moet jumpman knipperen (en uiteindelijk het onsterfelijkheids muziekje te horen zijn).

De hamerafhandeling zit hier:

- In de main een aanroep naar #1E8C voor de hamer afhandeling (op #197D). Deze zorgt alleen voor de afhandeling van de hamer hit en dat is niet meer nodig. Dus hele stuk van #1E8C t/m #1F45 kan vrijgegeven worden. Ook de main aanroep op #197D kan vrijgegeven worden voor hergebruik.
- In de main een aanroep naar #2ED4 voor hamer grab afhandeling (op #1998). Deze aanpassen naar de nieuwe situatie. Wel de grab en einde houden, maar de animatie van hamer enz is niet nodig.
- In de main een aanroep naar #281D voor hammer afhandeling (op #19B6). Deze checkt of de hamer actief is en zo ja dan wordt er gecheckt op een hamer hit en dat is niet meer nodig. Dus hele stuk van #281D t/m #2852 kan vrijgegeven worden. Ook de main aanroep op #19B6 kan vrijgegeven worden voor hergebruik.

Geheugenplaats #6394 wordt gebruikt als hamer timer. Voor beide hamers wordt deze gezet en gaat dan teruglopen. Deze kan dus gecheckt worden om tijdelijk de invincibility te zetten.

Geheugenblok #6680 t/m #668F is datastructuur voor de eerste hamer.

Geheugenblok #6690 t/m #669F is datastructuur voor de tweede hamer.

Lastig stuk code. Aanpassingen:

- #2F97 t/m #2F9B: code check hammer grab: moet blijven.
- #2F9C t/m #2FB6: code when hammer is grabbed: moet blijven. Alleen aan het einde een JP doen naar de code voor het animeren van jumpman.
- #2FB7 t/m #2FCA: code om hamer kleur te bepalen: kan vrijgegeven worden.
- #2F05 t/m #2F96: is de code om de hameranimatie te doen afhankelijk van de richting waarop jumpman kijkt, op en neer beweging en sprite aanpassing. Dat is allemaal niet meer nodig. De betreffende hamer sprite uitzetten (door deze qua X-positie op 0 te zetten) en jumpman afhankelijk van de hamer timer laten knipperen.

Aanpassen:

```
2F05 EB EX DE,HL ; DE <> HL – load HL with hammer sprite address
2F06 3600 LD (HL),#00 ; make hammer invisible
;
; animate jumpman
;
2F08 0602 LD B,#02 ; load B with jumpman's normal color - BBBB
2F0A 3A9463 LD A,(#6394) ; load A with hammer timer

2F0D CB5F BIT 3,A ; is bit 3 on in A?
2F0F 2802 JR Z,AAAA ; no, skip next step – jump to AAAA
;
2F11 0603 LD B,#03 ; load B with jumpman's alternative color
;
2F13 78 LD A,B ; load A with color - AAAA
2F14 324E69 LD (#694E),A ; set jumpman color
;
2F17 C9 RET ; return
```

en:

```
2FB4 C3082F JP #2F08 ; jump to jumpman animation – jump to BBBB
```

Eerst maar eens testen zo.

En even invincibility aanzetten om goed te kunnen testen zonder dat je af gaat.

Er wordt geen hamer getoond, dus dat is goed. Maar jumpman knippert niet en ook loopt de hamer niet af. Je kunt daardoor niet klimmen en niet springen.

Debuggen toont aan dat de hammer timer (#6394) de hele tijd op nul staat en dus niet terugloopt. Wordt die dan wel gezet? Nee die wordt niet gezet. Daarnaast wordt de hamer pas weggehaald als jumpman helemaal geland is. terwijl dat best al eerder zou kunnen.

Waar wordt de #6394 normaal gezet? Zie ik in de code helemaal niet. Dat zou kunnen, maar dan start de waarde op #00 en wordt opgehoogd met 1 en daarna doorlopen tot weer op #00. Maar de code om te verhogen is verwijderd. Die weer inbrengen.

Hamer timer blijkt uit twee timers te bestaan: de #6394 (die rondloopt) en de #6395 (die tot 2 telt). Dus waarschijnlijk duurt de hamer 2 omlopen van #6394.

Aanpassen:

```
2F17 219463 LD HL,#6394 ; load HL with hammer timer
2F1A 34 INC (HL) ; increase. at zero?
2F1B C0 RET NZ ; no, return
;
```

```

; hammer is changing or ending
;
2F1C 219563      LD   HL,#6395    ; load HL with hammer length.
2F1F 34          INC  (HL)       ; increase
2F20 7E          LD   A,(HL)     ; get the value
2F21 FE02        CP   #02       ; is the hammer all used up?
2F23 C0          RET  NZ       ; no, return
;
; arrive here when hammer runs out
;
2F24 AF          XOR  A         ; A := 0
2F25 329563      LD   (#6395),A ; clear hammer length
2F28 321762      LD   (#6217),A ; store into hammer indicator
2F2B DD7701      LD   (IX+#01),A ; clear hammer active indicator
2F2E DD360000    LD   (IX+#00),#00 ; clear hammer active bit
;
2F32 3A8963      LD   A,(#6389)  ; load A with previous background music
2F35 328960      LD   (#6089),A ; set music with what it was before the hammer
;
2F38 C9          RET           ; return

```

Dat werkt nu wel goed.

Alleen eindigt nu op de alternatieve kleur en dat is niet gewenst. Dus twee kleuren omdraaien.

```

2F08 0603        LD   B,#03     ; load B with jumpman's alternative color - BBBB
2F0A 3A9463      LD   A,(#6394)  ; load A with hammer timer

2F0D CB5F        BIT  3,A      ; is bit 3 on in A?
2F0F 2802        JR   Z,AAAA   ; no, skip next step – jump to AAAA
;
2F11 0602        LD   B,#02     ; load B with jumpman's normal color

```

Dat gaat nu goed.

Jumpman kan nu echter niet springen en klimmen.

Hier zitten checks:

```

1AD1 3A1762      LD   A,(#6217)  ; load A with hammer check
1AD4 3D          DEC  A         ; is hammer active?
1AD5 CAE61A      JP   Z,#1AE6   ; yes, skip ahead

```

en:

```

1AFE 3A1762      LD   A,(#6217)  ; else load A with hammer check
1B01 3D          DEC  A         ; is the hammer active?
1B02 C8          RET  Z       ; yes, return

```

Zowel de conditionele jump en de conditionele return in NOP's veranderen. Juist niets doen wanneer de hamer actief is.

Ja dan kun je wel weer klimmen en springen. Maar dan gebeurt er ook weer iets geeks. Bij een sprong wordt namelijk de 'hamer' losgelaten en die blijft ergens hangen (zie je niet), maar kan dan ineens wel iets raken. Ook het knipperen van jumpman stopt dan. Dus er wordt getriggert dat de hamer active uitgezet wordt. Hoe kan dat?

Eerst even alleen het klimmen uitzetten. Dus de conditionele jump weer terugzetten. Dat gaat helemaal raar.

Blijkbaar gooit de hammer indicator (#6217) roet in het eten. Misschien die niet zetten. Hebben we zelf helemaal niet nodig.

Deze wordt gezet op #1B5D.

```
1B5A 3A1862 LD A,(#6218) ; load A with hammer grabbing indicator  
1B5D 321762 LD (#6217),A ; store into hammer indicator  
1B60 210762 LD HL,#6207 ; load HL with movement indicator address
```

Dus de #1B5D in 3xNOP veranderen.

Nee dat gaat niet goed. Dan blijft de ster hangen. Teruggedraaid.

Terug naar de situatie met de conditionele sprong en conditionele return verwijderd. Dit gaat in principe goed, maar de hamer wordt losgelaten. Waar wordt dat gedaan? Met debugger checken op de 6217 en 6218. Erg onduidelijk.

Even iets wilds proberen:

```
; arrive here when jump is complete  
  
1B5A 3A1862 LD A,(#6218) ; load A with hammer grabbing indicator  
1B5D 321762 LD (#6217),A ; store into hammer indicator
```

Dat wordt bij elke jump gedaan. Deze hier weghalen door ze beiden in NOP's te veranderen.

En dan hier:

```
; arrive here when hammer is grabbed  
  
2F9C DD360906 LD (IX+#09),#06 ; set width ?  
2FA0 DD360A03 LD (IX+#0A),#03 ; set height ?  
2FA4 3A0762 LD A,(#6207) ; load A with mario movement indicator	sprite value  
2FA7 07 RLCA ; rotate left the high bit into carry flag
```

de commando's voor de width en height verwijderen en daar de copy van 6218 naar 6217 invoegen.

Verdraaid. Dat werkt goed. Misschien nog even verder testen. Maar nu eerst eventjes veilig stellen.

Nu versie v0.11 opgeslagen.

Wanneer de hamer actief is, wordt ook de kleur van de buzzy beetles aangepast. Dat wil je ook niet.

Dat zit hier:

```
3151 3AA163 LD A,(#63A1) ; \ This fire slot is active. Increment count for # of fire
3154 3C INC A ; |
3155 32A163 LD (#63A1),A ; /
3158 3E01 LD A,#01 ; \ Set fire color to #01 (normal) if hammer is not 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 ; /
```

Het lijkt erop dat de aanpassing van de kleur wanneer de hamer active is gedaan wordt op #3167.
Dus die NOP maken.

Ja dat gaat goed.

Nu de onsterfelijkheid nogal lang. Wanneer het de helft van de tijd duurt:

```
2F21 FE01 CP #01 ; is the hammer all used up?
```

Dat is wel erg kort. Dus dat op 2 houden.

De andere timer korter maken door niet de hele ronde te doorlopen:

```
2F17 219463 LD HL,#6394 ; load HL with hammer timer
2F1A 34 INC (HL) ; increase. at zero?
2F1B C0 RET NZ ; no, return
```

Aanpassen:

```
2F17 C3392F JP #2F39 ; jump to additional code – jump to AAAA
2F1A 0000 NOP,NOP
```

en:

```
2F39 219463 LD HL,#6394 ; load HL with hammer timer - AAAA
2F3C 34 INC (HL) ; increase
2F3D 7E LD A,(HL) ; load A with hammer time
2F3E FED0 CP #B0 ; hammer time = #B0?
2F40 C0 RET NZ ; no, return
;
2F41 3600 LD (HL),#00 ; clear hammer timer
2F43 C31C2F JP #2F1C ; jump back
```

Dat gaat goed.

Dan nu de invincibility afhankelijk maken van de hammer active #6217.

Vanuit de main wordt voor de collision detection gesprongen naar #2808.
Daar dan een extra check inbouwen.

```
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.
2816 A7 AND A ; was there a collision ?
2817 C8 RET Z ; no, return
```

Aanpassen:

```
2808 C3462F JP #2F46 ; jump to additional code – jump to AAAA
280B 00 NOP
```

en:

```
2F46 3A1762 LD A,(#6217) ; load A with hammer active indicator
2F49 FE01 CP #01 ; is hammer active?
2F4B C8 RET Z ; yes, return
;
2F4C FD210062 LD IY,#6200 ; load IY with start of jumpman sprite
2F50 C30C28 JP #280C ; jump back
```

Ja, dat gaat goed.

Nu versie v0.12 opgeslagen.

Nu nog de overbodige stukken en de bijbehorende slots in de main vrijgeven.

- In de main een aanroep naar #1E8C voor de hamer afhandeling (op #197D). Deze zorgt alleen voor de afhandeling van de hamer hit en dat is niet meer nodig. Dus hele stuk van #1E8C t/m #1F45 kan vrijgegeven worden. Ook de main aanroep op #197D kan vrijgegeven worden voor hergebruik.
- In de main een aanroep naar #281D voor hammer afhandeling (op #19B6). Deze checkt of de hamer actief is en zo ja dan wordt er gecheckt op een hamer hit en dat is niet meer nodig. Dus hele stuk van #281D t/m #2852 kan vrijgegeven worden. Ook de main aanroep op #19B6 kan vrijgegeven worden voor hergebruik.

Eerst de aanroepen in de main verwijderen en testen of het nog goed werkt.

Ja dat werkt nog goed.

Dan de stukken vrijgeven.

Ja, dat werkt ook nog.

Nu versie v0.13 opgeslagen.

Dan nu het vuur boven de groene pijp niet laten aangaan en niet animeren en de onzichtbare sprite verwijderen. De code daarvoor vrijgeven.

Geheugenplaats #6438 wordt gebruikt om aan te geven of de oil can al brand. Dat moet je wel houden want daar hangen een aantal begin barrel eigenschappen van af.

Hier zit een aanroep om het vuur te ontsteken en/of te animeren:

```
02DA 77 LD (HL),A ; else store A into the memory, for next time
02DB CD7F03 CALL #037F ; check for updating of difficulty
02DE CDA203 CALL #03A2 ; check for releasing fires on girders and conveyors
02E1 18DA JR #02BD ; loop back to check for more tasks
```

Wat gebeurt er als deze aanroep verwijderd wordt?

Nee, dat gaat niet goed omdat er dan ook geen buzzy beetles meer gespawned worden. Dat is niet de bedoeling. Terugzetten.

```
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
```

De code op #104C zet de sprite aan voor het vuurtje. Eerst een blanco sprite en daarna een vuur dat geanimeerd wordt. Wat als je hem niet aanzet. Dan X=#00 en is hij niet zichtbaar. Animeren maakt dan niet uit denk ik.

Dus testen door de opdracht in NOP's te veranderen. Nee dat helpt niet. Terugzetten.

Wat ook zou kunnen is het bewust verplaatsen van het vuur buiten het scherm. Dan is het er wel en wordt het geanimeerd, maar zie je het niet.

```
3DF4: 27 70 01 E0 00 00 ; initial data for fires on girders ?
3DFA: 7F 40 01 78 02 00 ; initial data for conveyors to release a fire ?
```

Dan de #3DF4 aanpassen naar #00.

Ja dat werkt. Maar dan kun je niet echt code vrijgeven.

Nu versie v0.14 opgeslagen.

Nu het barrelboard aanpassen naar voorbeeld van 'Paulines items' board in DK Barrelopooza.

- Smallere girders en rechts extra platformen.
- ~~Extra platform achter groene buis (vroegere oil can)~~
- Barrel rollover aanpassen zodat de barrels goed rollen.
- Bonus items als muntjes boven de platformen rechts plaatsen.
- Ladder Pauline pas tonen als alle muntjes gepakt.
- Groene buis sprites over de platformen rechts plaatsen.
- Afhandeling van piranha planten die uit de buizen komen.

Eerst de girders korter maken:

3AE4: 02 97 38 68 38 ; top girder where girl sits
3AE9: 02 9F 54 10 54 ; girder where kong sits
3AED: 02 CF 57 A0 55 ; 1st slanted girder at top right
3AF3: 02 EF 6D 20 79 ; 2nd slanted girder (has hammer at left side)
3AF8: 02 CF 99 10 8E ; 3rd slanted girder
3AFD: 02 EF AF 20 BB ; 4th slanted girder
3B02: 02 CF DB 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

Ja dat ziet er goed uit.

Dan de extra platformen maken.

3B5C: 02 FF 51 E0 51 ; next to 1st slanted girder at top right
3B61: 02 FF 93 E0 93 ; next to 3rd slanted girder
3B66: 02 FF D5 E0 D5 ; next to 5th slanted girder (has hammer at right side)
3B6B: AA

Dat gaat wel dan nu over de definitie van de conveyoprs board heen. Maar dat geeft niet.

En extra platform maken achter groene pijp die de oil can vervangt.

3B6B: 02 30 E0 20 E0
3B70: AA

Nee dat gaat niet goed. Als op hoogte #E8, dan precies achter de bovenkant van de groene pijp. Maar als jumpman er dan van af springt gaat hij dood. Als op hoogte #E9/#EA/#EB/#EC/, dan kan hij er wel van afspringen, maar dan versprint het blauwe schild naar dat platform en gaat dan weer terug naar rechts. Dus dat kan niet.

Nu versie v0.15 opgeslagen.

Nu de barrel roll over aan de rechterkant aanpassen.

De check op de edges zit hier:

```
2017 DD7E03 LD A,(IX+#03) ; Load A with Barrels' X position
201A FE1C CP #1C ; have we arrived at left edge of girder?
201C DA2F20 JP C,#202F ; yes, jump ahead to handle

201F FEE4 CP #E4 ; else , have we arrived at right edge of girder?
2021 DABA21 JP C,#21BA ; no, jump way ahead - we're done, store values and try next barrel
```

De roll-over aan de rechterkant is #10 naar links verplaatst.

Aanpassen:

```
201F FED4 CP #D4 ; else, have we arrived at the right edge of girder ?
```

Ja dat gaat goed.

Soms toch lastig te springen naar platformen. De platformen 2 pixels naar beneden doen.

3B5C: 02 FF 51 E0 51 ; next to 1st slanted girder at top right
3B61: 02 FF 93 E0 93 ; next to 3rd slanted girder
3B66: 02 FF D5 E0 D5 ; next to 5th slanted girder (has hammer at right side)
3B6B: AA

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

En dan de bonus items plaatsen.

Kan tussengevoegd worden bij de aanroep van het plaatsen van de hamers voor het barrel board (op positie #1000):

```
1000 CDA41E CALL #1EA4 ; call additional code – call label AAAA
```

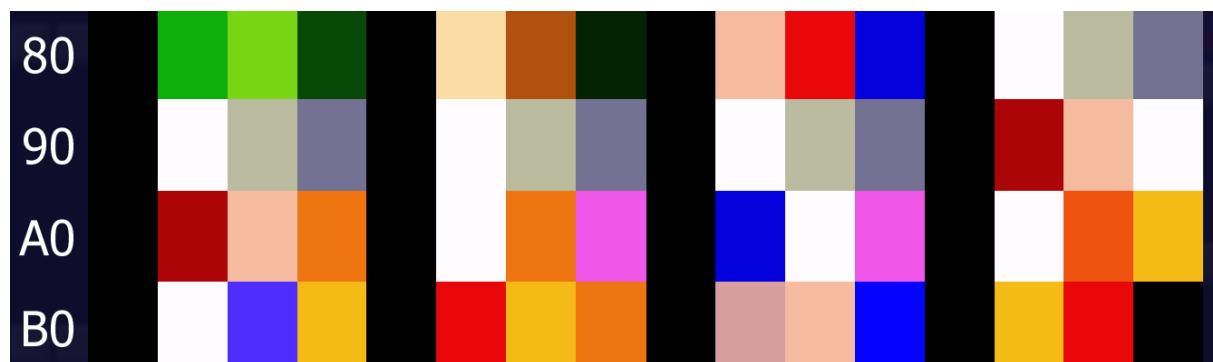
en

```
1E8C E6 73 09 4B ; position of coins on barrel board
1E90 E7 74 09 8D
1E94 E8 75 09 CF
;
1E98 00 00 00 00 ; reserved for position of green pipes on barrel board
1E9C 00 00 00 00
1EA0 00 00 00 00
;
1EA4 218C1E LD HL, #1E8C ; load start coin positions table
;
1EA7 110C6A LD DE,#6A0C ; load DE with sprite destination
1EAA 010C00 LD BC,#000C ; 3 items x 4 bytes = 12 bytes (#0C)
1EAD EDB0 LDIR ; draw bonus item sprites
```

```
;
1EAF 210C3E      LD      HL, #3EOC      ; start of table data for hammers
1EB2 C9          RET                 ; jump back
```

De coin sprite is aangemaakt op sprite waarde: #73 / #74 en #75. De oude bonus items. De coins zijn ten opzichte van elkaar 1 pixel verschoven omdat ze qua positie ook 1 verschoven moeten zijn. Kunnen niet op dezelfde X-positie staan vanwege de wijze van coderen. Dus dat heeft elkaar dan op en dan staan ze wel visueel onder elkaar.

Dat is nu goed. Nu alleen nog de kleur aanpassen. Is er nog een kleur slot dat niet gebruikt wordt?



De kleuren in slot #0B lijken het meest erop. Dus die gebruiken.

Dat is nu goed.

[Nu versie v0.17 opgeslagen.](#)

Nu zo maken dat de ladder naar Pauline niet getekend wordt en pas verschijnt als alle drie de coins gepakt zijn.

```
19C2  CD383F      CALL   #3F38      ; call additional main calls
```

en:

3F38	CD483F	CALL	#3F48	; call code to check for all three bonus items cleared
3F3B	CD623F	CALL	#3F62	; call code to handle addition ladder girders
3F3E	000000	NOP,NOP,NOP		; reserved for another call (used for piranha plants)
3F41	000000	NOP,NOP,NOP		; reserved for another call (used for lava)
3F44	000000	NOP,NOP,NOP		; reserved for another call (used for lava fireball)
3F47	C9	RET		; return

en:

3F48	3AB064	LD	A,(#64B0)	; read bonus-item pickup counter
3F4B	A7	AND	A	; are the three bonus items already picked?
3F4C	C0	RET	NZ	; yes, return
;				

```

3F4D 3A0C6A      LD   A,(#6A0C)    ; read X-position first bonus item
3F50 A7          AND  A           ; item is picked?
3F51 C0          RET  NZ         ; no, return
;
3F52 3A106A      LD   A,(#6A10)    ; read X-position second bonus item
3F55 A7          AND  A           ; item is picked?
3F56 C0          RET  NZ         ; no, return
;
3F57 3A146A      LD   A,(#6A14)    ; read X-position third bonus item
3F5A A7          AND  A           ; item is picked?
3F5B C0          RET  NZ         ; no, return
;
3F5C 3EF0          LD   A,#F0        ; load A with #F0
3F5E 32B064        LD   (#64B0),A  ; set bonus-item pickup counter to #F0
3F61 C9          RET             ; return

```

En alvast de trap naar Pauline weghalen.

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

En de rest van de definitie terugschuiven en laten aansluiten.

Nu met debugger testen of na drie items oppakken de #64B0 op waarde #F0 gezet wordt.

Ja dat gaat goed.

En dan nu de code schrijven voor het aanmaken van de ladder en het displayen van de sparkle sprite en het blinken ervan en het zetten van de bonus-pickup counter op #FF.

Gebruikt de heart sprite voor de sparkle (#6A20). Sparkle sprite nummer = #63 en blank sprite nummer = #64. Positie op de ladder knipperen: X=#AB, Y=#67. Color sparkle sprite = #09.

Aanmaken ladder wordt gedaan met call naar volgende code:

```

3FC8: 00 93 38 93 55 AA          ; test table data removed ladder
;
3FCE: 11C83F      LD   DE,#3FC8    ; load start table data
3FD1: CDA70D      CALL #0DA7     ; draw screen
3FD4: 21C83F      LD   HL,#3FC8    ; load HL with start of table data for ladder
3FD7: DD210C63    LD   IX,#630C    ; #6300 is start, with 11 ladders added: 630C
3FDB: CD7524      CALL #2475     ; draw ladder
3FDE: C9          RET             ; return

```

Wel nog aanpassen voor het aantal ladders dat er al is! Misschien al goed?

Aanpassen:

```

3F62 3A2762      LD   A,(#6227)    ; load A with screen number
3F65 3D          DEC  A           ; is this the girders?
3F66 C0          RET  NZ         ; if not return
;

```

```

3F67 3AB064      LD   A,(#64B0)    ; load A with bonus-item pickup counter
3F6A FE00         CP   #00          ; all bonus-items picked up?
3F6C C8           RET  Z           ; no, return
;
3F6D FEF0         CP   #F0          ; all bonus-items picked up?
3F6F 2813         JR   Z,#AAAA     ; yes, create ladder and start blinking – label AAAA
;
3F71 FE10         CP   #10          ; finished blinking?
3F73 C8           RET  Z           ; yes, return
;
3F74 3D           DEC  A           ; decrement A
3F75 32B064       LD   (#64B0),A  ; save back in bonus-item pickup counter
3F78 E610         AND  #10          ; bit 4 set?
3F7A 3E63         LD   A,#63        ; set sparkle sprite
3F7C 2802         JR   Z,XX        ; skip ahead
3F7E 3E64         LD   A,#64        ; reset sparkle sprite
3F80 32216A       LD   (#6A21),A  ; set sparkle sprite number
3F83 C9           RET
;
3F84 3D           DEC  A           ; decrement A – label AAAA
3F85 32B064       LD   (#64B0),A  ; save back in bonus-item pickup counter
;
3F88 CDCE3F       CALL #3FCE      ; create the ladder – label BBBB
3F8B 21206A       LD   HL,#6A20    ; load HL with heart sprite
3F8E 36AB         LD   (HL),#AB    ; set heart sprite x-position
3F90 23           INC  HL          ; next
3F91 3663         LD   (HL),#63    ; set sparkle sprite
3F93 23           INC  HL          ; next
3F94 3609         LD   (HL),#09    ; set heart sprite color
;
3F96 23           INC  HL          ; next
3F97 3667         LD   HL,#67        ; set heart sprite y-position
3F99 C9           RET

```

Gaat op zich goed. Alleen knippert niet en laat sparkle achter. Dus dat is nog niet goed.

Nu versie v0.18 opgeslagen.

Niet goed knipperen komt door tikfout waardoor conditionele return ineens een onconditionele return geworden is. Aangepast.

Knippert alleen nog op de verkeerde plek. Hierboven in de code op positie (#AB,#67). Aangepast naar positie (#94,#49).

Ja, dat gaat nu goed.

Nu versie v0.19 opgeslagen.

Dan nu groene pijpen plaatsen over de drie extra platformen.

Welke sprite slots daarvoor gebruiken? Vanaf #6A30 zou moeten kunnen.

Aanpassen:

```
1E98 E6 49 00 5B      ; reserved for position of green pipes on barrel board  
1E9C E6 49 00 9D  
1EA0 E6 49 00 DF
```

en:

```
1EB2 21981E      LD      HL, #1E98    ; load start coin positions table  
;  
1EB5 11306A      LD      DE,#6A30    ; load DE with sprite destination  
1EB8 010C00      LD      BC,#000C    ; 3 items x 4 bytes = 12 bytes (#0C)  
1EBB EDB0      LDIR      ; draw bonus item sprites  
;  
1EBD 210C3E      LD      HL, #3E0C    ; start of table data for hammers  
1EC0 C9          RET      ; jump back
```

Dat gaat goed, maar de bovenste wordt weggehaald wanneer je over een enemy springt. Wordt die gebruikt voor het weergeven van de score? Lijkt er op.

Daarom #6A40 gebruiken.

Ja, dat gaat nu goed.

[Nu versie v0.20 opgelagen.](#)

Nu kijken of er een piranha plant toegevoegd kan worden die uit de rechter groene pijpen tevoorschijn komt.

Die moet in de collisions voor het barrel board ook gecheckt gaan worden. Dat betekent dat er een standaard structuur gevuld moet worden. In de code kijken welke structuur dat minimaal moet zijn.

Blijkt steeds te starten met een IX met de start van de array van te controleren sprites en een DE met de offset en een B met het aantal sprites dat gecheckt moet worden. Dan wordt de code op #2913 aangeroepen.

Deze code checkt het volgende uit de definitie van de sprite:

- IX+#00: sprite active (#01) of niet (#00).
- IX+#03: sprite X-position
- IX+#05: sprite Y-position
- IX+#09: sprite hitbox X width
- IX+#0A: sprite hitbox Y height

Dus minimal deze waarden invullen. En een extra toevoegen als timer voor de afhandeling van de wachtperiode tussen verschijnen en de afhandeling van de animatie.

- IX+#0B: wait timer or animation timer

Maar welke IX daarvoor gebruiken? Een ongebruikt stuk geheugen of een stuk van een andere enemy gebruiken die niet in het barrel board gebruikt wordt.

Gecheckt. In het elevators board wordt de #6500 als start gebruikt voor de springs. Deze worden niet gebruikt in het barrel board. In debugger gecontroleerd en deze wordt inderdaad niet gebruikt.

Eerst stuk code maken die op basis van 4 toestanden (active status and wait timer).

Dit stukje code altijd aanroepen vanuit de main. We hadden al een extra stuk gemaakt:

Aanpassen:

3F3E CD7B0A CALL #0A7B ; call code to handle piranha plant - **AAAA**

en:

```
0A7B 3A2762        LD A,(#6227) ; load A with screen number - AAAA
0A7E 3D            DEC A        ; is this the girders?
0A7F C0            RET NZ      ; if not return
;
0A80 3A0065        LD A,(#6500) ; load A with piranha status
0A83 FE00          CP #00      ; is piranha plant active?
0A85 2017          JR NZ,#BBBB ; yes, jump forward – jump to BBBB
;
; piranha plant is not active
;
0A87 3A0B65        LD A,(#650B) ; load A with piranha wait timer
0A8A FE00          CP #00      ; timer expired?
0A8C 2805          JR Z,#CCCC ; yes, jump forward – jump to CCCC
;
0A8E 3D            DEC A      ; decrease wait timer
0A8F 320B65        LD (#650B),A ; store adjusted wait timer
0A92 C9            RET        ; done for now, return
;
0A93 3E01          LD A,#01     ; load A with #01 - CCCC
0A95 320065        LD (#6500),A ; make piranha plant active
0A98 3EF0          LD A,#F0     ; load A with #F0
0A9A 320B65        LD (#650B),A ; set piranha animation timer
0A9D C9            RET        ; done for now, return
;
; piranha plant is active
;
```

```

0A9E 3A0B65 LD A,(#650B) ; load A with piranha animation timer - BBBB
0AA1 FE00 CP #00 ; timer expired?
0AA3 200B JR NZ,#DDDD ; no, jump forward – jump to DDDD
;
0AA5 3E00 LD A,#00 ; load A with #01 - CCCC
0AA7 320065 LD (#6500),A ; make piranha plant inactive
0AAA 3E60 LD A,#60 ; load A with #60
0AAC 320B65 LD (#650B),A ; set piranha wait timer
0AAF C9 RET ; done for now, return
;
0AB0 FEF0 CP #F0 ; is this the start of the animation? - DDDD
0AB2 C2FB0A JP NZ,#0AFB ; no, jump forward – jump to EEEE
;
; Initialize piranha plant structure and piranha plant sprite
;
0AB5 3A0362 LD A,(#6203) ; load A with jumpman's X-position
0AB8 FEDB CP #DB ; jumpman is right of position #DB?
0ABA D0 RET NC ; yes, do not initialize, return
;
0ABB DD210065 LD IX,#6500 ; load IX with start piranha info
0ABF FD212C6A LD IY,#6A2C ; load IY with start piranha sprite
;
0AC3 3EE6 LD A,#E6 ; load A with #E6
0AC5 DD7703 LD (IX+#03),A ; set info X-position
0AC8 FD7700 LD (IY+#00),A ; set sprite X-position
;
0ACB FD360147 LD (IY+#01),#47 ; set sprite value – piranha mouth closed
0ACF FD360200 LD (IY+#02),#00 ; set sprite color
;
0AD3 DD360902 LD (IX+#09),#02 ; set info hitbox X width
0AD7 DD360A0A LD (IX+#0A),#0A ; set info hitbox y height
;
0ADB 3A1860 LD A,(RngTimer1) ; load A with random number
0ADE E603 AND #03 ; make it a random number between #00 and #03
0AE0 FE00 CP #00 ; A == #00?
0AE2 2002 JR NZ ; no, skip next step
0AE4 3E02 LD A,#02 ; make it a random number between #01 and #03
;
0AE6 47 LD B,A ; load B with A to start loop
0AE7 3E09 LD A,#09 ; load A with pipe start
;
0AE9 C642 ADD A,#42 ; add pipe offset - FFFF
0AEB 10FC DJNZ #FFFF ; loop until B is zero – loop to FFFF
;
0AED DD7705 LD (IX+#05),A ; set info Y-position
0AF0 FD7703 LD (IY+#03),A ; set sprite Y-position
;

```

```

0AF3 3A0B65      LD   A,(#650B)    ; load piranha animation timer
0AF6 3D          DEC  A           ; decrease piranha animation timer
0AF7 320B65      LD   (#650B),A  ; store adjusted piranha animation timer
;
0AFA C9          RET            ; done for now, return
;
; perform piranha plant animation
;
0AFB 3A0B65      LD   A,(#650B)    ; load piranha animation timer - EEEE
0AFE 3D          DEC  A           ; decrease piranha animation timer
0AFF 320B65      LD   (#650B),A  ; store adjusted piranha animation timer
;
0B02 C9          RET            ; done for now, return

```

Nu eerst testen met alleen plaatsen en dan checken collisions op verschillende hoogtes. Ook de hitbox misschien nog tunen. Als dat allemaal goed is, dan de animatie gaan toevoegen.

Het tonen van de sprites gaat nu goed. Weliswaar zonder animatie, maar werkt.

Nu versie v0.21 opgeslagen.

Nu kijken naar de hitbox. Daartoe de collision detection met de piranha plant toevoegen.

De laatste collision detection op het barrels board is de volgende code:

```

28A0 0601      LD   B,#01       ; B := 1
28A2 78         LD   A,B        ; A := 1
28A3 32B963    LD   (#63B9),A  ; store counter for use later
28A6 1E00      LD   E,#00       ; E := #00
28A8 DD21A066  LD   IX,#66A0    ; load IX with oil can fire location
28AC CD1329    CALL #2913     ; check for collision with oil can fire
28AF C9          RET            ; return

```

Aanpassen:

```
28AC CDC11E      CALL #1EC1      ; call additional code
```

en:

```

1EC1 CD1329      CALL #2913     ; check for collision with oil can fire
;
1EC4 0601      LD   B,#01       ; B := 1
1EC6 78         LD   A,B        ; A := 1
1EC7 32B963    LD   (#63B9),A  ; store counter for use later
1ECA 1E00      LD   E,#00       ; E := #00
1ECC DD210065  LD   IX,#6500    ; load IX with piranha plant location
1ED0 CD1329    CALL #2913     ; check for collision with piranha plant
1ED3 C9          RET            ; return

```

Ja, dat werkt. Maar zelfs in de bovenste positie springt jumpman er nog tegenaan van onderen af. Misschien de hitbox kleiner maken?

Met hitbox van #01 bij #01 gaat het goed in de bovenste positie. Maar tot hoe ver gaat het dan niet goed? Precies achter de pijp is #19 als start positie van de planten.

```
0AE7 3E19      LD    A,#19      ; load A with pipe start
```

In de onderste positie al heel snel na springen een hit. Dat is niet goed, want hoe ga je dit goed krijgen? Creatieve oplossing bedenken. Iets decoratiefs a la SMB waardoor je daar niet kunt komen. Maar de buzzy beetles wel.

Eerst maar de animatie goed maken.

```
;  
; perform piranha plant animation  
;  
0AFB 3A0B65      LD    A,(#650B)    ; load piranha animation timer - EEEE  
0AFE 3D          DEC   A          ; decrease piranha animation timer  
0AFF 320B65      LD    (#650B),A  ; store adjusted piranha animation timer  
;  
OB02 C9          RET              ; done for now, return
```

Aanpassen:

```
OB02 CB6F      BIT   6,A      ; test bit 6  
OB04 FE00      CP    #00      ; is it #00?  
OB06 3E47      LD    A,#47      ; load A with sprite value piranha mouth closed  
;  
OB08 2002      JR    NZ,#AAAA  ; no, skip next step  
OB0A 3E48      LD    A,#48      ; load A with sprite value piranha mouth open  
;  
OB0C 322D6A      LD    (#6A2D),A  ; set sprite value - animate plant mouth  
;  
OB0F C9          RET              ; done for now, return
```

Ja de bek gaat nu open en dicht.

Nu nog tevoorschijn laten komen en weer laten verdwijnen.

```
OB0F 3A0B65      LD    A,(#650B)    ; load A with piranha timer  
OB12 FEBO      CP    #B0      ; A > #B0?  
OB14 380B      JR    C,AAAA  ; no, jump forward - AAAA  
;  
OB16 3A0C65      LD    A,(#650C)    ; load A with piranha shift timer  
OB19 FEF8      CP    #F8      ; A == #F8?
```

```

OB1B 2027      JR   NZ,BBBB    ; no, jump forward - BBBB
;
OB1D 06FF      LD   B,#FF     ; load B with #FF (shift up)
OB1F 1814      JR   #CCCC    ; jump forward - CCCC
;
OB21 FE50      CP   #50      ; A<#50? - AAAA
OB23 D0        RET  NC      ; no, return - DDDD
OB24 00        NOP
;
OB25 3A0C65    LD   A,(#650C) ; load A with piranha shift timer
OB28 FEF8      CP   #F8      ; A == #F8?
OB2A 2018      JR   NZ,BBBB    ; no, jump forward - BBBB
;
OB2C 0601      LD   B,#01      ; load B with #01 (shift up)
OB2E 1805      JR   #CCCC    ; jump forward - CCCC
;
OB30 3A0C65    LD   A,(#650C) ; load A with piranha shift timer - DDDD
OB33 180F      JR   #BBBB    ; jump forward - BBBB
;
OB35 3A0565    LD   A,(#6505) ; load Y-position - CCCC
OB38 80        ADD  A,B      ; shift one position up or down
;
OB39 320565    LD   (#6505),A ; store Y-position in structure
OB3C 322F6A    LD   (#6A2F),A ; store Y-position in sprite
OB3F 3E00      LD   A,#00      ; load A with #00
OB41 320C65    LD   (#650C),A ; reset piranha shift timer
;
OB44 3D        DEC  A       ; decrease the piranha shift timer - BBBB
OB45 320C65    LD   (#650C),A ; store adjusted piranha shift timer
OB48 C9        RET
;
```

Gaat nog niet goed. Gaan debuggen wat er precies gebeurt.

Verschillende condities waren niet goed en shift timer werd aangepast in de uitgeschoven stand.

Allemaal aangepast en werkt nu. Moet nog wel getuned worden, maar eerst even veilig stellen.

Nu versie v0.22 opgeslagen.

Dan nu de tuning doen. De piranha plant op de juiste plek laten beginnen.

```
0AE7 3E19      LD   A,#19      ; load A with pipe start
```

De piranha plant moet iets sneller eruit komen. Dus de piranha shift timer aanpassen.

```
0B19 FEFC      CP   #FC      ; A == #FC?
0B28 FEFC      CP   #FC      ; A == #FC?
```

Ja, zo even laten.

Omhoog moet hij nog verder doorschuiven.

Dus de grens waarbij stoppen met omhoog schuiven moet naar beneden:

```
0B12 FEAB CP #AB ; A > #AB?
```

en dan te ver naar beneden doorschuiven.

Dus de grens waarbij beginnen met omlaag schuiven moet naar beneden:

```
0B21 FE42 CP #42 ; A < #42? - AAAA
```

Lastig. Gaat voor verschillende opvolgende keren dat de piranha plant verschijnt op een andere manier. De piranha timer en piranha shift timer worden niet gereset. Kan dat gedaan worden bij het initialiseren van de nieuwe piranha plant?

i.p.v. de volgende code een call naar additionele code doen:

```
0AF7 320B65 LD (#650B),A ; store adjusted piranha animation timer
```

Aanpassen:

```
0AF7 CD490B CALL #0B49 ; call additional code
```

en:

```
0B49 320B65 LD (#650B),A ; store adjusted piranha animation timer  
;  
0B4C 3E00 LD A,#00 ; load a with #00  
0B5E 320C65 LD (#650C),A ; reset piranha shift timer  
;  
0B51 C9 RET ; return
```

Ja, dat helpt.

Schuift iets te ver naar beneden:

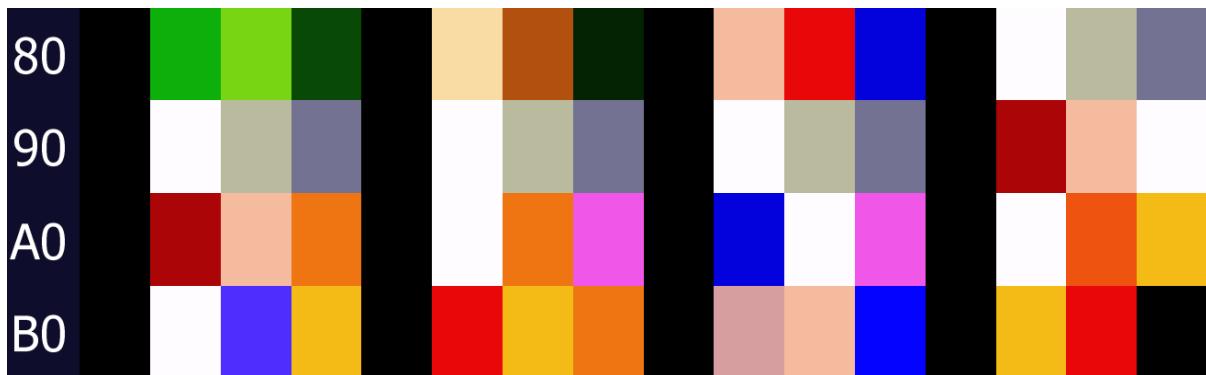
```
0B21 FE40 CP #40 ; A < #40? - AAAA
```

De sprite aanpassen omdat de blaadjes naast de pijp zichtbaar zijn. Dus smaller maken.

Ja, is nu goed.

Nu versie v0.23 opgeslagen.

Nu nog de kleuren van de piranha plant aanpassen.



De piranha plant is bruin / groen / wit. Die combinatie is er nog niet. Welke kleurcode wordt er in het barrel board nog niet gebruikt? Lijkt dat #0E en #0F nog niet gebruikt worden.

Kijken of we #0E kunnen gebruiken: dat is dan #B9/#BA/#BB.

```
0ACF FD36020E      LD      (IY+#02),#0E ; set sprite color
```



Kleurcodes:

- #B9 moet wit worden: #00 / #00
- #BA moet bruin worden: #01 / #03
- #BB moet groen worden: #0E / #0B

Kleuren zijn nu goed.

De wachttijd tot opnieuw verschijnen kan iets korter.

Dat wordt hier gedaan:

```
0AAA 3E30      LD      A,#30      ; load A with #30
0AAC 320B65     LD      (#650B),A ; set piranha wait timer
```

De munten liggen niet in het midden van de pijp. Optisch niet zo mooi. Iets naar links schuiven:

```
1E8C E5 73 0B 4B      ; position of coins on barrel board
1E90 E6 74 0B 8D
```

1E94 E7 75 0B CF

Ja, dat is nu ook goed.

Nu versie v0.24 opgeslagen.

Nu de hitbox groter (i.g. breder) maken en inbouwen check zodat niet van onderen de piranha plant te raken.

Hitbox aanpassen:

```
0AD3 DD360906 LD (IX+#09),#06 ; set info hitbox X width  
0AD7 DD360A01 LD (IX+#0A),#01 ; set info hitbox y height
```

Ja, dat is goed.

Alleen nu zo maken dat niet van onderen tegenaan te springen.

Ervan uitgaan dat de mario sprite lager is dan de piranha plant. Dan kun je dus checken op het verschil ertussen:

Jumpman Y-position: #6205

Piranha plant Y-position: #6505

Dus alleen collision checken als piranha hoger dan jumpman: dus alleen checken als piranha Y-position is lager dan jumpman y-position. Dus checken als piranha Y – jumpman Y = NC.

De code van hier voor:

```
1EC1 CD1329 CALL #2913 ; check for collision with oil can fire  
;  
1EC4 0601 LD B,#01 ; B := 1  
1EC6 78 LD A,B ; A := 1  
1EC7 32B963 LD (#63B9),A ; store counter for use later  
1ECA 1E00 LD E,#00 ; E := #00  
1ECC DD210065 LD IX,#6500 ; load IX with piranha plant location  
1ED0 CD1329 CALL #2913 ; check for collision with piranha plant  
1ED3 C9 RET ; return
```

Aanpassen:

```
1EC1 CD1329 CALL #2913 ; check for collision with oil can fire  
;  
1EC4 3A0562 LD A,(#6205) ; load A with jumpman Y-position  
1EC7 47 LD B,A ; store in B  
1EC8 3A0565 LD A,(#6505) ; load A with piranha plant Y-position  
1ECB 90 SUB B ; subtract jumpman Y-position  
1ECC D8 RET C ; return if jumpman has higher Y-position  
;
```

```

1ECD 0601      LD   B,#01      ; B := 1
1ECF 78        LD   A,B       ; A := 1
1ED0 32B963    LD   (#63B9),A ; store counter for use later
1ED3 1E00      LD   E,#00      ; E := #00
1ED5 DD210065  LD   IX,#6500  ; load IX with piranha plant location
1ED9 CD1329    CALL #2913    ; check for collision with piranha plant
1EDC C9        RET           ; return

```

Ja, dat werkt redelijk goed, maar niet helemaal. Bij testen blijkt dat jumpman toch soms af gaat. Uit meerdere testen, checken op Y-posities wanneer afgaan:

Y-pos piranha Y-pos-jumpman

9D	9D	even hoog
9A	9A	even hoog
9C	9B	jumpman 1 hoger dan piranha

Blijkbaar kan jumpman in 1 frame hoger komen dan de piranha plant terwijl ze niet gelijk zijn. Dit voorkomen door een offset te gebruiken? Offset van twee gebruiken.

Aanpassen:

```

1EC1 CD1329    CALL #2913    ; check for collision with oil can fire
;
1EC4 3A0562    LD   A,(#6205) ; load A with jumpman Y-position
1EC7 47        LD   B,A       ; store in B
1EC8 3A0565    LD   A,(#6505) ; load A with piranha plant Y-position
1ECB D605      SUB #05      ; offset by 5
1ECD 90        SUB B        ; subtract jumpman Y-position
1ECE D8        RET  C        ; return if jumpman has higher Y-position
;
1ECF 0601      LD   B,#01      ; B := 1
1ED1 78        LD   A,B       ; A := 1
1ED2 32B963    LD   (#63B9),A ; store counter for use later
1ED5 1E00      LD   E,#00      ; E := #00
1ED7 DD210065  LD   IX,#6500  ; load IX with piranha plant location
1EDB CD1329    CALL #2913    ; check for collision with piranha plant
1EDE C9        RET           ; return

```

Lijkt nu goed te gaan. Opslaan en dan meerdere malen testen.

Nu versie v0.25 opgeslagen.

Er gebeuren nu wel wat vreemde zaken. Als je iets te ver naar links naar de pijp springt, dan val je erlangs en ga je niet dood. Als je dan omhoog klimt en de ster pakt dan ga je dood wanneer je op de girder land. Hoe kan dit? Dat kan toch haast niet komen door de check en de offset die toegevoegd is? Lijkt me stug. Misschien ook met een eerdere versie testen.

Daarnaast lijkt het ook alsof DK meer treuzelt met het gooien van schilden. Sinds wanneer is dat dan?

Wat wellicht zou kunnen is dat de A en de B aangepast zijn door de check op de Y-positie. Als de check niet uitgevoerd wordt, wordt een return gedaan met aangepaste A en B. Dat kan voorkomen worden door de A en B tijdelijk op de stack op te slaan en terug te zetten voor de return. Wat ook zou kunnen is dat er in de logica met call en returns iets niet goed zit, waardoor er verkeerd teruggesprongen wordt. Dat laatste kan tot hele vreemde resultaten leiden.

De logica lijkt goed. Dan proberen met het veiligstellen van A en B op de stack.

Aanpassen:

```
1EC1 CD1329      CALL #2913      ; check for collision with oil can fire
;
1EC4 F5          PUSH AF        ; save A to stack for later
1EC5 C5          PUSH BC        ; save B to stack for later
1EC6 3A0562      LD  A,(#6205)   ; load A with jumpman Y-position
1EC9 47          LD  B,A        ; store in B
1ECA 3A0565      LD  A,(#6505)   ; load A with piranha plant Y-position
1ECD D605        SUB #05        ; offset by 5
1ECF 90          SUB B         ; subtract jumpman Y-position
1ED0 C1          POP BC        ; restore B from stack
1ED1 F1          POP AF        ; restore A from stack
1ED2 D8          RET C         ; return if jumpman has higher Y-position
;
1ED3 0601        LD  B,#01        ; B := 1
1ED5 78          LD  A,B        ; A := 1
1ED6 32B963      LD  (#63B9),A  ; store counter for use later
1ED9 1E00        LD  E,#00        ; E := #00
1EDB DD210065    LD  IX,#6500   ; load IX with piranha plant location
1EDF CD1329      CALL #2913    ; check for collision with piranha plant
1EE2 C9          RET           ; return
```

Dit werkt niet. De POP's zouden de carry flag niet beïnvloeden. Maar in de debugger blijft na de POP BC de carry flag wel staan, maar na de POP AF niet. Hoe kan dat?

Dan anders doen:

```
1EC1 CD1329      CALL #2913      ; check for collision with oil can fire
;
1EC4 F5          PUSH AF        ; save A to stack for later
1EC5 C5          PUSH BC        ; save B to stack for later
1EC6 3A0562      LD  A,(#6205)   ; load A with jumpman Y-position
1EC9 47          LD  B,A        ; store in B
1ECA 3A0565      LD  A,(#6505)   ; load A with piranha plant Y-position
1ECD D605        SUB #05        ; offset by 5
1ECF 90          SUB B         ; subtract jumpman Y-position
1ED0 3003        JR  NC         ; skip steps if jumpman has lower Y-position – AAAA
```

```

;
1ED2 C1          POP BC      ; restore B from stack
1ED3 F1          POP AF      ; restore A from stack
1ED4 C9          RET         ; return without collision check piranha plant
;
1ED5 C1          POP BC      ; restore B from stack - AAAA
1ED6 F1          POP AF      ; restore A from stack
;
1ED7 0601        LD  B,#01    ; B := 1
1ED9 78          LD  A,B     ; A := 1
1EDA 32B963      LD  (#63B9),A ; store counter for use later
1EDD 1E00        LD  E,#00    ; E := #00
1EDF DD210065    LD  IX,#6500 ; load IX with piranha plant location
1EE3 CD1329      CALL #2913   ; check for collision with piranha plant
1EE6 C9          RET         ; return

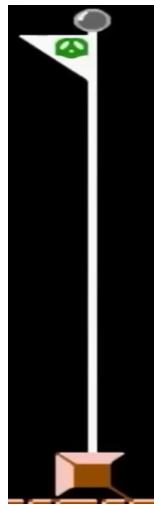
```

Het lijkt nu goed te gaan. Nu voor langere tijd testen.

[Nu versie v0.26 opgeslagen.](#)

En dan nu gaan kijken naar een vlaggenmast.

Die ziet er in het donker zo uit. Bij daglicht groen, met een groene vlag met witte opdruk.



Besluiten of we hem maken op basis van de achtergrond of op basis van sprites. Op basis van sprites heb je meer keuze qua kleuren (wellicht relevant bij nog te ontwerpen andere velden). De vlag moet sowieso een sprite worden als je die wilt laten schuiven (of in ons geval verplaatsen).

Eerst de sprites maken.

Sprite #65 = voet, sprite #66 = mast, #66 = top, #67 = vlag.

Dan ook nu gaan toevoegen dat dit getekend gaat worden. Dit kan op hetzelfde moment als de groene pijpen.

Dat was de volgende code:

```
1E98 E6 49 00 5B      ; reserved for position of green pipes on barrel board  
1E9C E6 49 00 9D  
1EA0 E6 49 00 DF
```

en:

```
1EB2 21981E      LD    HL, #1E98      ; load start coin positions table  
;  
1EB5 11306A      LD    DE,#6A30      ; load DE with sprite destination  
1EB8 010C00      LD    BC,#000C      ; 3 items x 4 bytes = 12 bytes (#0C)  
1EBB EDB0        LDIR             ; draw bonus item sprites  
;  
1EBD 210C3E      LD    HL, #3EOC      ; start of table data for hammers  
1EC0 C9          RET               ; jump back
```

In plaats van de 1EBD daar een additionele call tussenvoegen en daar dan de vlaggenmast tekenen.

Aanpassen:

```
1EBD CDE71E      CALL   #1EE7      ; call additional code
```

en:

```
1EE7 21F61E      LD    HL, #1EF6      ; load start flagpole positions table  
;  
1EEA 11506A      LD    DE,#6A50      ; load DE with sprite destination  
1EED 011400      LD    BC,#0014      ; 5 items x 4 bytes = 20 bytes (#14)  
1EF0 EDB0        LDIR             ; draw bonus item sprites  
;  
1EF2 210C3E      LD    HL, #3EOC      ; start of table data for hammers  
1EF5 C9          RET               ; jump back  
;  
1EF6 B4 65 03 40  ; vlaggenmast voet  
1EFA B4 66 03 30  ; vlaggenmast stok  
1EFE B4 66 03 20  ; vlaggenmast stok  
1F02 B4 67 03 10  ; vlaggenmast top  
1F06 AB 68 0E 20  ; vlag
```

Positionering doen. Wel lastig. Je kan niet testen met springen. Even de eindcheck aanpassen naar hoger, zodat je bij Pauline kan komen en kan gaan springen. Dat zit hier:

```

; check for end of level on girders and elevators

1E7A  FE31      CP      #31          ; are we on top level (rescued girl?)
1E7C  D0        RET     NC           ; no, return

1E7D  C36D1E    JP      #1E6D        ; level has been fished. jump to end of level routine.

```

Aanpassen:

```
1E7A  FE      21      CP      #21          ; are we on top level?
```

Ziet er nu zo uit:



Nu versie v0.27 opgeslagen.

Dan nu het platform van Pauline verder naar rechts uitbreiden en dan tevens de aanpassing voor schuine girders daar niet doen, zodat je er gewoon horizontaal kan lopen. en dan de check einde level op basis van springen naar stok en dan gaan meten welke hoogtes en daar dan bonussen aan koppelen.

Pauline platform:

```
3AE4: 02 97 38 68 38 ;
```

Verder naar rechts uitbreiden:

3AE4: 02 B7 38 68 38 ;

En dan de voet van de vlaggenstok verwijderen:

```
1EF6 B4 66 03 30 ; vlaggenmast stok  
1EFA B4 66 03 20 ; vlaggenmast stok  
1EFE B4 67 03 10 ; vlaggenmast top  
1F02 AB 68 0E 20 ; vlag
```

en index aanpassen:

```
1EED 011000 LD BC,#0010 ; 5 items x 4 bytes = 20 bytes (#14)
```

Ja dat is zo goed.

Ervoor zorgen dat de girder aanpassing voor de helling niet gedaan wordt op de girder van Pauline.

Helling aanpassing van jumpman wordt hier gedaan:

```
1Cdf 66 LD h,(HL) ; else load H with mario X position  
1Ce0 3A0562 LD A,(#6205) ; load A with mario Y position  
1Ce3 6f LD l,A ; copy to L. HL now has mario X,Y  
1Ce4 cd3323 CALL #2333 ; check for movement up/down a girder, might also change Y position ?  
1Ce7 7D LD A,l ; load A with new Y position  
1Ce8 320562 LD (#6205),A ; store into Y position
```

De call naar #2333 doet de daadwerkelijke aanpassing. Deze alleen doen als jumpman lager dan de girder van Pauline. Dus als de Y-position van jumpman groter dan #40.

Aanpassen:

```
1CE4 CD061F CALL #1F06 ; call addional code
```

en:

```
1F06 FE40 CP #40 ; jumpman is at Pauline's girder?  
1F08 D8 RET C ; yes, return  
;  
1F09 CD3323 CALL #2333 ; check for movement up/down girder  
1FOC C9 RET
```

Ja, dat gaat goed.

Nu versie v0.28 opgeslagen.

Nu einde niet door hoger dan iets te komen, maar checken hoger dan #40 èn rechts van de paal. En dan checken op hoe hoog en daar dan de animatie op aanpassen.

De check op finish level zit hier:

```
; check for end of level on girders and elevators

1E7A FE31      CP      #31          ; are we on top level (rescued girl?)
1E7C D0        RET     NC           ; no, return

1E7D C36D1E    JP      #1E6D        ; level has been fished. jump to end of level routine.
```

Aanpassen:

```
1E7A C30D1F    JP      #1F0D        ; jump to additional code
```

en:

```
1F0D FE31      CP      #31          ; are we on or above top girder?
1F0F D0        RET     NC           ; no, return
;
1F10 3A0362    LD      A,#6203      ; load A with jumpman's X-position
1F13 FEB4      CP      #B4          ; is jumpman touching the flagpole?
1F15 D8        RET     C            ; no, return
;
1F16 C37D1E    JP      #1E7D        ; jump back
```

Dat gaat goed.

Nu versie v0.29 opgeslagen.

Nu nog testen op hoe hoog en dan bonus toekennen. Eerst eens met debugger kijken hoe hoog we kunnen springen en dan op basis daarvan een verdeling van de bonus punten bedenken.

De hoogte varieert van #30 (laag) tot #22 (hoog).

Hoe wordt op rivets de bonus van de laatste rivet nog uitgekeerd? Stukje gevonden en hier toevoegen.

Aanpassen:

```
1F16 067E      LD      B,#7E        ; set sprite for 500 points
1F18 110500    LD      DE,#0005      ; set points at 500
1F1B CD9F30    CALL   #309F        ; award points
;
1F1E C37D1E    JP      #1E7D        ; jump back
```

Dit werkt wel. Je krijgt de 500 punten bonus in dit geval, maar de bonus sprite is niet te zien. Komt dat omdat de sprites allemaal gewist worden? Wat als we dat niet doen?

Dat lijkt hier te zitten:

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

1615 CD9F30 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
```

Dus even de code op #1615 NOP's maken. Ja, dat werkt, maar zie nog steeds de bonus indicatie niet.

Waar wordt bepaald waar deze bonus indicatie weergegeven dient te worden. Kan me herinneren dat hier ook iets mee gedaan is in DK Barrelopalooza met de steering bonus.

Daar wordt de bonus op een andere manier toegekend.

Aanpassen:

```
1F1B CD281E CALL #1E28 ; award points
```

Dat gaat wel goed, maar de bonus indicatie staat ietwat laag.

Uit de code op 1E28 wordt duidelijk dat de plaatsing relatief is t.o.v. jumpman's positie.

```
1E28 CD9F30 CALL #309F ; insert task to add score
1E2B 3A0562 LD A,(#6205) ; load A with Mario's Y position
1E2E C614 ADD A,#14 ; add #14
1E30 4F LD C,A ; store into C
1E31 3A0362 LD A,(#6203) ; load A with mario's X position

1E34 00 NOP
1E35 00 NOP ; [ what used to be here? was it LD B,#7B to set sprite for 100 pts? ]

; draw the bonus score on the screen

1E36 21306A LD HL,#6A30 ; load HL with scoring sprite start
1E39 77 LD (HL),A ; store X position
```

Daar verandering in brengen door eerst een eigen setup te doen en naar een plek ietsje verder te springen.

Aanpassen:

```
1F1B CD9F30 CALL #309F ; award points
1F1E 3A0562 LD A,(#6205) ; load A with jumpman's Y position
1F21 C604 ADD A,#04 ; add #04
1F23 4F LD C,A ; store in C
```

```

1F24 3EA0          LD   A,#A0        ; load A with #A0
1F26 CD361E        CALL #1E36      ; display scoring sprite
;
1F29 C37D1E        JP   #1E7D      ; jump back

```

Ja, dat gaat nu goed.

Alleen jammer dan jumpman achter de vlag weergegeven wordt. Aanpassen door voor de vlaggenmast spriteslots te nemen voor de jumpman sprite. Lijkt te kunnen startend vanaf #6930.

Aanpassen:

```
1EEA 113069        LD   DE,#6930    ; load DE with sprite destination
```

Ja dat gaat goed. Maar draait jumpman nog wel om. Als je de jumpman sprite niet wil aanpassen dan moet je springen naar #1E85.

Aanpassen:

```
1F29 C3851E        JP   #1E85      ; jump back
```

Ja, dat gaat goed.

Nu versie v0.30 opgeslagen.

Nu nog variabele punten toekennen.

Nu standaard 500 punten:

```

1F16 067E          LD   B,#7E        ; set sprite for 500 points
1F18 110500        LD   DE,#0005    ; set points at 500

```

De sprites zijn: #7B (100), #7C (200), #7D (300), #7E (500) en #7F (800).

Hoogte check:

- 22 - 800 punten – sprite #7F
- 23 - 500 punten – sprite #7E
- 24+ - 300 punten – sprite #7D

Hoe dit slim te programmeren? Of gewoon een lijst met conditionele sprongen maken? Dat laatste lijkt het meest zinvolle.

Aanpassen:

```

1F16 CD520B        CALL #0B52      - call additional code
1F19 0000          NOP, NOP

```

en:

```
0B52 3A0562      LD   A,(#6205)    ; load A with jumpman's Y-position  
;  
0B55 FE22        CP   #22         ; is it 22?  
0B57 2006        JR   NZ,#BBBB    ; no, skip forward  
0B59 067F        LD   B,#7F       ; set sprite for 800 points  
0B5B 110800      LD   DE,#0008    ; set points at 800  
0B5E C9          RET            ; return  
;  
0B5F FE23        CP   #23         ; is it 23?  
0B61 2006        JR   NZ,#BBBB    ; no, skip forward  
0B63 067E        LD   B,#7E       ; set sprite for 500 points  
0B65 110500      LD   DE,#0005    ; set points at 500  
0B68 C9          RET            ; return  
;  
0B69 067D        LD   B,#7D       ; set sprite for 300 points  
0B6B 110300      LD   DE,#0003    ; set points at 300  
0B6E C9          RET            ; return
```

Ja, dat werkt goed.

Normaal 500 punten of je moet wel heel slecht springen. De 800 punten is wel lastiger!

Weer even onsterfelijkheid uitzetten om de speelbaarheid te gaan testen.

Nu versie v0.31 opgeslagen.

De positie van de bovenste ster aanpassen door een girder lager en meer naar rechts. Daar eerder nodig doordat je de buzzy beetles niet kunt vernietigen en dus wanneer ze snel omhoog komen, lastig kunnen worden.

Dat zit hier:

```
3E08: 1E 07        ; 1E is the hammer sprite value. 07 is hammer color  
3E0A: 03 09        ; ???  
3E0C: 24 64        ; position of top hammer for girders. 24 is X, 64 is Y  
3E0E: BB C0        ; bottom hammer for girders at BB, C0
```

Aanpassen:

3E0C A2 7C

Nee dat is hem toch nog niet. Is wel heel erg makkelijk dan. Kan altijd nog.

Dan nu met het rivets board beginnen.

Starten met het rivets board.

```

3A65: 01 04          ; level 1
3A67: 01 03 04      ; level 2
3A6A: 01 02 03 04    ; level 3
3A6E: 01 02 01 03 04 ; level 4
3A73: 01 02 01 03 01 04 ; level 5 +
3A79: 7F            ; end code

```

Aanpassen: #3A65 naar 04.

En het palette van het girders board gebruiken.

```

0CB6 CD430D CALL #0D43      ; draws the blue vertical bars next to kong on rivets
0CB9 21867D LD HL,REG_PALETTE_A ; load HL with palette bank selector
0CBC 3601 LD (HL),#01      ; set palette bank selector
0CBE 3E0B LD A,#0B      ; load A with music code For rivets
0CC0 328960 LD (#6089),A ; set music
0CC3 118B3C LD DE,#3C8B ; load DE with start of table data for rivets

```

Voor girders is het palette (A/B): 0, 1 en dat wordt voor andere schermen overschreven. Dus niet overschreven als je het giders palette wil hebben.

Aanpassen:

0C8C 0000 NOP,NOP

De buitenste girders omzetten naar het type van griders.

```

3C8B: 00 7B 80 7B A8 ; center ladder level 3
3C90: 00 7B D0 7B F8 ; bottom center ladder
3C95: 00 33 58 33 80 ; top left ladder
3C9A: 00 53 58 53 80 ; top left ladder (right side)
3C9F: 00 AB 58 AB 80 ; top right ladder (left side)
3CA4: 00 CB 58 CB 80 ; top right ladder
3CA9: 00 2B 80 2B A8 ; level 3 ladder left side
3CAE: 00 D3 80 D3 A8 ; level 3 ladder right side
3CB3: 00 23 A8 23 D0 ; level 2 ladder left side
3CB8: 00 5B A8 5B D0 ; level 2 ladder #2 of 4
3CBD: 00 A3 A8 A3 D0 ; level 2 ladder #3 of 4
3CC2: 00 DB A8 DB D0 ; level 2 ladder right side
3CC7: 00 1B D0 1B F8 ; bottom left ladder
3CCC: 00 E3 D0 E3 F8 ; bottom right ladder
3CD1: 05 B7 30 48 30 ; girder above kong
3CD6: 05 CF 58 30 58 ; girder kong stands on
3CDB: 05 D7 80 28 80 ; level 4 girder
3CEO: 05 DF A8 20 A8 ; level 3 girder
3CE5: 05 E7 D0 18 D0 ; level 2 girder
3CEA: 05 EF F8 10 F8 ; bottom level girder

```

3CEF: AA ; end code

Oh dat kan niet zomaar, want de girders zijn geheel en niet in stukken. Makkelijkste is om ze allemaal om te zetten naar de girders variant (type = #02) en dan de middenstukken extra toe te voegen.

Maar dan eerst het stuk verplaatsen.

De definitie van girders eindigt op: #3B66. Dus die van rivets beginnen op #3B67.

3B67: 00 7B 80 7B A8 ; center ladder level 3
3B6C: 00 7B D0 7B F8 ; bottom center ladder
3B71: 00 33 58 33 80 ; top left ladder
3B76: 00 53 58 53 80 ; top left ladder (right side)
3B7B: 00 AB 58 AB 80 ; top right ladder (left side)
3B80: 00 CB 58 CB 80 ; top right ladder
3B85: 00 2B 80 2B A8 ; level 3 ladder left side
3B8A: 00 D3 80 D3 A8 ; level 3 ladder right side
3B8F: 00 23 A8 23 D0 ; level 2 ladder left side
3B94: 00 5B A8 5B D0 ; level 2 ladder #2 of 4
3B99: 00 A3 A8 A3 D0 ; level 2 ladder #3 of 4
3B9E: 00 DB A8 DB D0 ; level 2 ladder right side
3BA3: 00 1B D0 1B F8 ; bottom left ladder
3BA8: 00 E3 D0 E3 F8 ; bottom right ladder
3BAD: 02 B7 30 48 30 ; girder above kong
3BB2: 02 CF 58 30 58 ; girder kong stands on
3BB7: 02 D7 80 28 80 ; level 4 girder
3BBC: 02 DF A8 20 A8 ; level 3 girder
3BC1: 02 E7 D0 18 D0 ; level 2 girder
3BC6: 02 EF F8 10 F8 ; bottom level girder
3BCB: AA ; end code

En dan de table start aanpassen naar #3B67 op de plekken waar die gebruikt wordt:

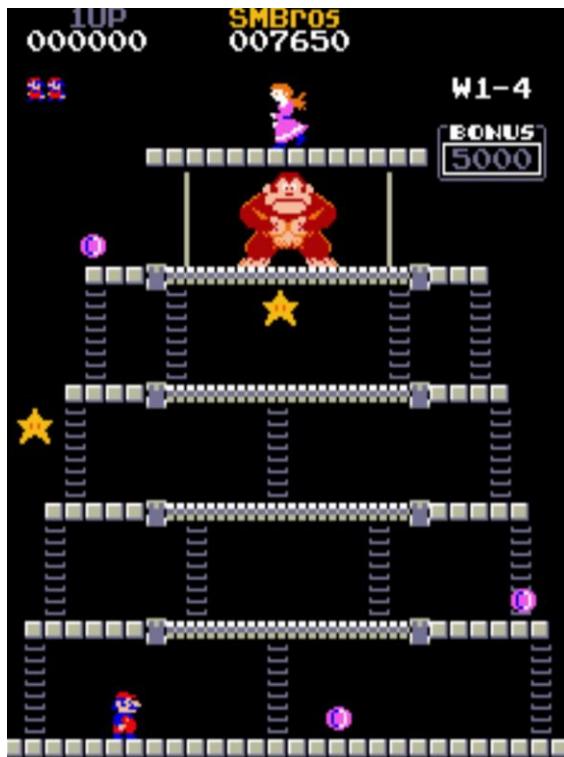
```
0CC3 118B3C LD DE,#3C8B ; load DE with start of table data for rivets  
246E 218B3C LD HL,#3C8B ; otherwise we're on rivets.
```

Dat werkt goed.

Nu de tussenliggende girders aanmaken met het oude type #05.

3BCB: 05 AF D0 50 D0 ; level 2 girder middle
3BDO: 05 AF A8 50 A8 ; level 3 girder middle
3BD5: 05 AF 80 50 80 ; level 4 girder middle
3BDA: 05 AF 58 50 58 ; kong girder middle
3BDF AA

Dat is goed. Ziet er nu zo uit:



Nu versie v0.32 opgeslagen.

Even de onderste girder in twee stukken splitsen zodat er lava tussen gemaakt kan worden. Het tweede extra stuk definiëren op de plek van de onderste middelste ladder.

Aanpassen:

3BC6: 02 **4F** F8 10 F8 ; bottom level girder left part

en:

3B6C: 02 EF F8 B0 F8 ; bottom level girder right part

3B6C: 00 7B D0 7B F8 ; bottom center ladder

Ja, dat ziet er nu goed uit.

Willen we bepaalde girders gaan verplaatsen? Of zo houden? Is wel erg hetzelfde als het origineel. Wel wat werk om girders (en rivets) en de afhandeling te verplaatsen.

Zo houden en dan:

- Lava sprites onderin en dood wanneer aanraken.
- Geen onsterfelijkheid.
- Leuke plaatsing van de bonus item muntjes.
- Goomba's i.p.v. firefoxes en erop springen maakt ze dood.

- Ze spawnen dan weer net als de firefoxes.
- Bowser i.p.v. Kong.
- Bowser gooit hamers afwisselend links en rechts.
- Finish pas na pakken bijl op onderste girder.
- Bij finish de midden girders wegschuiven en bowser in lava laten vallen.

Eerst starten met de lava sprites onderin en geen collision detection configureren. Collision detection op basis van extra stukje in de main die voor het rivets board de Y-positie van jumpman checkt en wanneer in de lava dan af.

De sprite voor de lava aangemaakt. Op de plek van de blauwe tonnen. Want die wordt toch niet meer gebruikt. Is graphic #19. En sprite de kleur van jumpman geven. Is kleurcode #02.

Sprites tekenen zonder collision detection kan door een data blok te maken en deze simpelweg met een LDIR te kopiëren naar het sprite geheugen. Deze sprites i.p.v. de barrels, dus die sprite slots kunnen gebruikt worden. Zijn de slots die starten op #6980. Even checken of die inderdaad in rivets board niet gebruikt worden. Nee die worden inderdaad niet gebruikt. Als het goed is zijn er 6 sprites naast elkaar nodig om de lava te tekenen.

Tussenvoegen bij het tekenen van de hamers op het rivets board.

Dat gebeurt hier:

```
113D 21143E LD HL,#3E14 ; load HL with start of table data for hammer locations
1140 CDA611 CALL #11A6 ; draw the hammers
```

Aanpassen:

```
1140 CD870B CALL #0B87 ; call additional code
```

Datablok:

```
0B6F 57 19 02 F6
0B73 67 19 02 F6
0B77 77 19 02 F6
0B7B 87 19 02 F6
0B7F 97 19 02 F6
0B83 A7 19 02 F6
```

en:

```
0B87 CDA611 CALL #11A6 ; draw the hammers
0B8A 216F0B LD HL,#0B6F ; start of table data of the lava
0B8D 118069 LD DE,#6980 ; sprites start
0B80 011800 LD BC,#0018 ; #18 bytes to copy
0B93 EDB0 LDIR ; copy
0B95 C9 RET ; return
```

Ja, dat ziet er goed uit.

Nu versie v0.33 opgeslagen.

Nu nog de detectie op het aanraken van de lava maken.

Jumpman's Y-positie op de onderste girder is #F0. Als deze groter is (dan is jumpman lager) dan jumpman laten afgaan.

Op #3F41 is nog een slot vrij om de collision detection with the lava tussen te voegen.

Aanpassen:

3F41 CD960B CALL #0B96 ; call additional code: collision detection with lava

en:

```
0B96 3A2762      LD A,(#6227) ; load A with screen number - AAAA
0B99 FE04      CP #04 ; is this the rivets?
0B9B C0      RET NZ ; if not return
;
0B9C 3A0562      LD A,(#6502) ; load A with Y-position jumpman
0B9F FEF1      CP #F1 ; is jumpman below lowest girder?
0BA1 D8      RET C ; no, return
;
0BA2 AF      XOR A ; A == #00
0BA3 320062      LD (#6200),A ; make mario dead
0BA6 C9      RET ; return
```

Ja, dat werkt goed, maar haalt wel de lava weg. En dat willen we eigenlijk niet. Dit is standaard na het finishen van het veld. Maar de lava willen we behouden. De groene pijpen bij barrels blijven ook staan. Is die ruimte hier ongebruikt? Dat zijn de sprite sloten beginnend bij #6A40. Die worden in het rivets board ook niet gebruikt.

Aanpassen:

0B8D 11406A LD DE,#6A40 ; sprites start

Ja, dat gaat nu allemaal goed.

Nu versie v0.34 opgeslagen.

Kijken of de firefoxes in Koopa's veranderd kunnen worden. En dan misschien ook in het barrels board aanpassen. Is toch leuker dan de Fuzzy Beetles.

Ja die is helemaal leuk. Nu nog de kleurcode #0E geven.

Dat zit hier.

```
3158 3E01      LD    A,#01          ; \ Set fire color to #01 (normal) if hammer is not 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   ; /
```

Ja, dat ziet er goed uit.

Nu ook zo maken voor het barrel screen?

Even terug naar barrel screen: aanpassen:

```
3A65: 01 04          ; level 1
3A67: 01 03 04        ; level 2
3A6A: 01 02 03 04      ; level 3
3A6E: 01 02 01 03 04    ; level 4
3A73: 01 02 01 03 01 04  ; level 5 +
3A79: 7F          ; end code
```

Ja dat ziet er veel leuker uit.

Nu versie v0.35 opgeslagen.

Nu de plaatsing van de sterren en de muntjes aanpassen. De bovenste ster gespiegeld aan de andere ster aan de rechterkant plaatsen.

Dat zit hier:

```
3E08: 1E 07          ; 1E is the hammer sprite value. 07 is hammer color
3E0A: 03 09          ; ???
3E0C: 24 64          ; position of top hammer for girders. 24 is X, 64 is Y
3E0E: BB C0          ; bottom hammer for girders at BB, C0
3E10: 23 8D 7B B4      ; for conveyors
3E14: 1B 8C 7C 64      ; for rivets
3E18: 4B 0E 04 02      ; ???
```

Aanpassen:

3E14 1B 8C E4 8C

Dan de plaatsing en de kleuren van de muntjes aanpassen.

Dat zit hier:

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

De kleur moet nu #0B zijn.

Aanpassen:

```
3E54 80 75 0B C8  
3E58 C3 75 0B 50  
3E5C 3B 75 0B 50
```

Het ziet er nu zo uit:



Nu versie v0.36 opgeslagen.

Idee is om uit de lava een vuurbal te laten schieten links en rechts naast het muntje. Als dat het heel moeilijk maakt dan kun je altijd het aantal koopa's terugbrengen naar maximaal vier stuks. Maar dat is dan tuning voor later.

Eerst even de sprite voor de bijl aan het einde toevoegen. Die zou eventueel als extra in de definitie van de lava toegevoegd kunnen worden. Moeten we wellicht wel iets gaan schuiven, maar wel het allermakkelijkste om te doen.

De sprite voor de switch (ziet eruit als een dubbelzijdige bijl) heeft sprite value #53.

Aanpassen:

Aanpassen:

```
1140 CD8B0B      CALL  #0B8B      ; call additional code
```

Datablok:

```
0B6F 57 19 02 F6  
0B73 67 19 02 F6  
0B77 77 19 02 F6  
0B7B 87 19 02 F6  
0B7F 97 19 02 F6  
0B83 A7 19 02 F6  
0B87 B7 53 03 E6
```

en:

```
0B8B CDA611      CALL  #11A6      ; draw the hammers  
0B8E 216F0B      LD    HL,#0B6F      ; start of table data of the lava  
0B91 11406A      LD    DE,#6A40      ; sprites start  
0B94 011C00      LD    BC,#001C      ; #1C bytes to copy  
0B97 EDB0        LDIR             ; copy  
0B99 C9          RET               ; return
```

en:

```
3F41 CD9A0B      CALL  #0B9A      ; call additional code: collision detection with lava
```

en:

```
0B9A 3A2762      LD    A,(#6227)    ; load A with screen number - AAAA  
0B9D FE04        CP    #04          ; is this the rivets?  
0B9F C0          RET   NZ           ; if not return  
;  
0BA0 3A0562      LD    A,(#6502)    ; load A with Y-position jumpman  
0BA3 FEF1        CP    #F1          ; is jumpman below lowest girder?  
0BA5 D8          RET   C            ; no, return  
;  
0BA6 AF          XOR   A            ; A == #00  
0BA7 320062      LD    (#6200),A    ; make mario dead  
0BAA C9          RET   ; return
```

Ja dat gaat goed. Ziet er goed uit.

Nu versie v0.37 opgeslagen.

Nu nog zo maken dat einde rivets pas na verwijderen alle rivets en activeren van de switch.

Check einde rivets zit hier:

```
1E80 3A9062 LD A,(#6290) ; load A with number of rivets left
1E83 A7 AND A ; all done with rivets ?
1E84 C0 RET NZ ; no, return

1E85 3E16 LD A,#16 ; else A := #16
1E87 320A60 LD (GameMode2),A ; store into game mode2
1E8A E1 POP HL ; pop stack to get higher address
1E8B C9 RET ; return to a higher level [returns to #00D2]
```

Aanpassen:

```
1E80 C3AB0B JP #0BAB ; jump to additional code
1E83 0000 NOP,NOP
```

en:

```
0BAB 3A9062 LD A,(#6290) ; load A with number of rivets left
0BAE A7 AND A ; all done with rivets?
0BAF C0 RET NZ ; no, return
;
0BB0 3A0362 LD A,(#6203) ; load A with jumpman X-position
0BB3 FEB9 CP #B9 ; is it #B9?
0BB5 C0 RET NZ ; no return
;
0BB6 3A0562 LD A,(#6205) ; load A with jumpman Y-position
0BB9 FEF0 CP #F0 ; is it #F0?
0BBC C0 RET NZ ; no return
;
0BBC C3851E JP #1E85 ; jump back, rivets finished
```

Ja, dat werkt goed.

Alleen bij het einde wordt Pauline op een nieuwe girder lager neergezet. Dit is echter die met de verdwijnen character. Moet eigenlijk het andere girder type zijn.

De data voor het aanpassen van het scherm zit hier:

```

; table data
; used when rivets are cleared

3A47: 05 AF F0 50 F0 AA
3A4D: 05 AF E8 50 E8 AA
3A53: 05 AF E0 50 E0 AA
3A59: 05 AF D8 50 D8 AA
3A5F: 05 B7 58 48 58 AA

```

Hoogstwaarschijnlijk is het de laatste. Aanpassen naar girder type #02.

Ja, dat klopt. Dat is nu goed. Later de hele animatie van het weghalen van de girders en de val van bowser aanpassen.

Nu versie v0.38 opgeslagen.

Dan nu code gaan maken voor het uit de lava schieten van vuurballen. Deze code is vergelijkbaar met de code voor het bewegen van de piranha plant.

Eerst de twee sprites toevoegen. Een voor omhoog vliegen en een voor omlaag vallen.

De sprite voor omhoog vliegen heeft sprite value #54 en de sprite voor omlaag vallen heeft sprite value #55.

De sprite moet bij einde van rivets wel weggehaald worden. Kan daarvoor het spriteslot van de pies gebruikt worden? Dan komt de vuurbal ook nog eens achter de lava vandaan. Dat is optisch wel een gewenst effect.

Pie sprites beginnen bij #69B8. Met debugger checken of deze inderdaad niet gebruikt wordt op rivets. Die wordt inderdaad niet gebruikt.

Ook een structuur kiezen voor het vastleggen van de data voor de vuurbal. Deze wordt gebruikt als IX met indexen voor het zetten van de waarden en voor het bepalen van de collisions. In het elevators board wordt de #6500 als start gebruikt voor de springs. Deze worden niet gebruikt in het barrel board. In debugger gecontroleerd en deze wordt inderdaad niet gebruikt.

Eerst stuk code maken die op basis van 4 toestanden (active status and wait timer).

Dit stukje code altijd aanroepen vanuit de main. We hadden al een extra stuk gemaakt:

Aanpassen:

```
3F44 CD0722      CALL  #2207      ; call code to handle piranha plant - AAAA
```

en:

```
2207 3A2762      LD    A,(#6227)   ; load A with screen number - AAAA
220A FE04        CP    #04        ; is this the rivets?
```

```

220C C0          RET   NZ      ; if not return
;
220D DD210065    LD    IX,#6500  ; load IX with start lava fireball info
2211 FD212C6A    LD    IY,#6A2C  ; load IY with start lava fireball sprite
;
2215 3A0065      LD    A,(#6500) ; load A with lava fireball status
2218 FE00         CP    #00      ; is lava fireball active?
221A 2045         JR    NZ,#BBBB  ; yes, jump forward – jump to BBBB
;
; lava fireball is not active
;
221C 3A0B65      LD    A,(#650B) ; load A with lava fireball wait timer
221F FE00         CP    #00      ; timer expired?
2221 2805         JR    Z,#CCCC  ; yes, jump forward – jump to CCCC
;
2223 3D          DEC   A       ; decrease wait timer
2224 320B65      LD    (#650B),A ; store adjusted wait timer
2227 C9          RET   ; done for now, return
;
; initialize and activate lava fireball
;
2228 3E01         LD    A,#01    ; load A with #01 - CCCC
222A DD7700      LD    (IX+#00),A ; set info lava fireball active
222D DD770C      LD    (IX+#0C),A ; set info lava fireball direction
;
2230 3E03         LD    A,#03    ; load A with #03
2232 DD770B      LD    (IX+#0B),A ; set info lava fireball speed timer
;
2235 3A1860      LD    A,(RngTimer1) ; load A with random number
2238 E601         AND   #01      ; make it a random number between #00 and #01
223A FE00         CP    #00      ; A == #00?
223C 3E60         LD    A,#60    ; load A with left position
223E 2002         JR    NZ      ; no, skip next step
2240 3EA0         LD    A,#A0    ; load A with right position
;
2242 DD7703      LD    (IX+#03),A ; set info X-position
2245 FD7700      LD    (IY+#00),A ; set sprite X-position
;
2248 3EF0         LD    A,#F0    ; load A with bottom position
224A DD7705      LD    (IX+#05),A ; set info Y-position
224D FD7703      LD    (IY+#03),A ; set sprite Y-position
;
2250 FD360154    LD    (IY+#01),#54 ; set sprite value – lava fireball up
2254 FD360200    LD    (IY+#02),#00 ; set sprite color
;
2258 DD360904    LD    (IX+#09),#04 ; set info hitbox X width
225C DD360A04    LD    (IX+#0A),#04 ; set info hitbox y height
;

```

```

2260 C9          RET          ; done for now, return
;
; lava fireball is active
;
2261 DD7E0B      LD   A,(IX+#0B)    ; load A with lava speed timer - BBBB
2264 3D          DEC  A           ; decrease speed timer
2265 FE00        CP   #00         ; did it run out?
2267 2804        JR   Z,EEEE       ; yes, jump forward - EEEE
;
2269 DD770B      LD   (IX+#0B),A  ; store decreased speed timer
226C C9          RET          ; done for now, return
;
; it is time to move the lava fireball
;
226D 3E03        LD   A,#03        ; reset the lava fireball speed timer - EEEE
226F DD770B      LD   (IX+#0B),A  ; store speed timer
;
2272 DD7EOC      LD   A,(IX+#0C)    ; load A with lava fireball direction
2275 FE01        CP   #01         ; is the fireball going up?
2277 2016        JR   NZ,#DDDD     ; no, jump forward – jump to DDDD
;
; move fireball going in the up direction
;
2279 DD7E05      LD   A,(IX+#05)    ; load A with lava fireball Y-position
227C 3D          DEC  A           ; move one position up
227D DD7705      LD   (IX+#05),A  ; store new lava fireball Y-position
2280 FD7703      LD   (IY+#03),A  ; store new lava fireball Y-position
;
2283 FEC0        CP   #C0         ; is lava fireball at the top?
2285 C0          RET  NZ          ; no, done for now, return
;
2286 DD360C02    LD   (IX+#0C),#02  ; set lava fireball direction down
228A FD360155    LD   (IY+#01),#55  ; set sprite value – lava fireball down
228E C9          RET          ; done for now, return
;
; move fireball going in the down direction
;
228F DD7E05      LD   A,(IX+#05)    ; load A with lava fireball Y-position - DDDD
2292 3C          INC  A           ; move one position down
2293 DD7705      LD   (IX+#05),A  ; store new lava fireball Y-position
2296 FD7703      LD   (IY+#03),A  ; store new lava fireball Y-position
;
2299 FEF0        CP   #F0         ; is lava fireball at the bottom?
229B C0          RET  NZ          ; no, done for now, return
;
229C DD360000    LD   (IX+#00),#00  ; set lava fireball inactive
22A0 DD360BB0    LD   (IX+#0B),#B0  ; set lava fireball wait timer
22A4 FD360164    LD   (IY+#01),#64  ; set sprite value – empty sprite

```

```
22A8 C9          RET           ; done for now, return
```

Dit is best een stuk code en daar is nu geen plaats voor. Eerst de ruimte van de afhandeling van pies vrijmaken. En dan kijken hoeveel er vrij komt. Er is nog best wel wat nodig voor de dingen die ik nog van plan ben met deze rom-hack. De vraag is of er voldoende ruimte vrij te maken is.

Vanuit de main worden aanroepen gedaan voor pie afhandeling naar #24EA, #2207 en #25F2.

#2207 t/m 22CA is voor afhandeling retractable ladders: kan vrijgegeven worden.

#24EA t/m #25F1 is voor afhandeling pies: kan vrijgegeven worden.

#25F2 t/m #26F9 is voor afhandeling pies: kan vrijgegeven worden.

Daarnaast kan eventueel later ook nog de code vrijgegeven worden voor afhandeling conveyors en elevators finished. Daar naar kijken bij het animeren van bowser vallen bij rivets. En de andere boards dan zelfde maken als barrels board.

En de drie aanroepen in de main vrijgeven:

```
1992 CDEA24      CALL     #24EA       ; do stuff for pie factory
199B CD0722      CALL     #2207       ; do stuff for conveyors
19AA CDF225      CALL     #25F2       ; handle conveyor directions
```

Ja, dat gaat goed.

Nu versie v0.39 opgeslagen.

De code ingevoegd. Gaat goed, maar kleur is niet goed, de sprite is ietwat groot en de snelheid is veel te langzaam.

Snelheid aanpassen:

```
2230 3E03        LD      A,#03       ; load A with #03
2232 DD770B      LD      (IX+#0B),A   ; set info lava fireball speed timer
```

en

```
226D 3E03        LD      A,#03       ; reset the lava fireball speed timer
226F DD770B      LD      (IX+#0B),A   ; store speed timer
```

Met snelheid terug naar 1 is de snelheid goed.

Moet kleurcode #0B krijgen.

```
2254 FD36020B    LD      (IY+#02),#0B  ; set sprite color
```

De kleuren zitten nog verkeerd. De geel en wit moeten omgedraaid. Aanpassen in sprite.

De posities wisselen netjes, maar zitten iets te ver naar buiten.

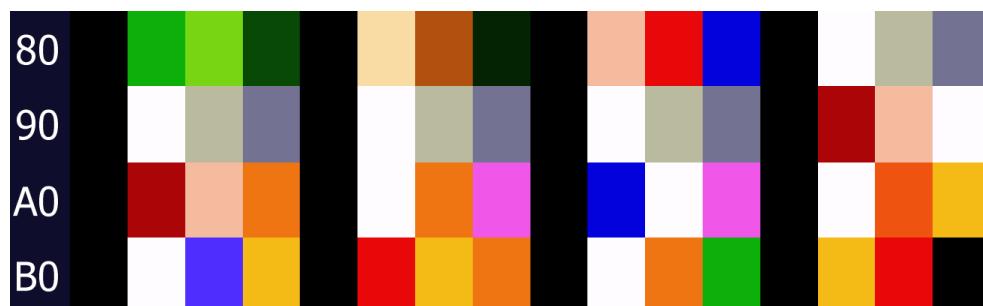
```
223C 3E68      LD    A,#68      ; load A with left position  
223E 2002      JR    NZ       ; no, skip next step  
2240 3E96      LD    A,#96      ; load A with right position
```

En de pauze tussen de vuurballen mag een stuk korter.

```
22A0 DD360B40   LD    (IX+#0B),#40 ; set lava fireball wait timer
```

De sprite moet iets platter gemaakt worden.

De kleur rood moet iets roder. Dat heeft ook effect op de muntjes en de ster, maar dat zal wel meevalen. Dat is kleurcode #0B.



Dus dat is positie #AE in de kleuren files. Dat is nu kleur #01/#07.

Testen met kleur #01/#0B. Dat ziet er een stuk beter uit.

De sprite wordt niet weggehaald bij finish level. Dat klopt. Verkeerde sprite gebruikt. Aanpassen:

```
2211 FD21B869   LD    IY,#69B8      ; load IY with start lava fireball sprite
```

De vuurballen beginnen en eindigen te hoog.

```
2248 3EF8      LD    A,#F8      ; load A with bottom position
```

```
2299 FEF8      CP    #F8      ; is lava fireball at the bottom?
```

Ja, dat gaat nu goed.

Sprite nog aanpassen. Niet tevreden over.

Origineel:



Eigen:



Nu:



Is nu allemaal goed.

Nu versie v0.40 opgeslagen.

Nu nog de collision detection toevoegen en dan met invincibility uit eens een aantal keren goed testen op speelbaarheid.

De laatste collision detection op het rivets board zit op #290F.

```
; jump here from #3E8C when on the rivets  
; check for collisions with firefoxes and squares next to kong
```

```
2901 E1 POP HL ; restore HL  
2902 0607 LD B,#07 ; B := 7  
2904 78 LD A,B ; A := 7  
2905 32B963 LD (#63B9),A ; store 7 into counter for use later  
2908 112000 LD DE,#0020 ; load DE with offset  
290B DD210064 LD IX,#6400 ; load IX with start of firefox arrays  
290F CD1329 CALL #2913 ; check for collisions with firefoxes/squares  
2912 C9 RET ; return
```

Aanpassen:

```
290F CDC11E CALL #1EC1 ; call additional code
```

en onderstaande code kan hergebruikt worden omdat dezelfde datastructuur op #6500 gebruikt is:

```
1EC1 CD1329 CALL #2913 ; check for collision with oil can fire  
;  
1EC4 0601 LD B,#01 ; B := 1  
1EC6 78 LD A,B ; A := 1  
1EC7 32B963 LD (#63B9),A ; store counter for use later  
1ECA 1E00 LD E,#00 ; E := #00  
1ECC DD210065 LD IX,#6500 ; load IX with piranha plant location  
1ED0 CD1329 CALL #2913 ; check for collision with piranha plant  
1ED3 C9 RET ; return
```

Testen door de invincibility uit te zetten.

Ja, dat gaat goed.

Wel nog even de hitbox breder en hoger maken:

```
2258 DD360906 LD (IX+#09),#06 ; set info hitbox X width  
225C DD360A05 LD (IX+#0A),#05 ; set info hitbox y height
```

Nu versie v0.41 opgeslagen.

Even gespeeld. Maar in world 3 heb je wel erg vaak dat er al een koopa snel omhoog komt. Daarom toch de ster verplaatsen:

```
3E0C A2 7C
```

Ja, dat is echt wel nodig. Helemaal goed zo.

De rivets is zo echt te moeilijk. Of een goede strategie vinden. Een tijdje spelen om te kijken. Anders wellicht positionering van de sterren aanpassen? Of wellicht het aantal Koopa's beperken tot slechts vier stuks?

Spreiden is wel een tactiek, maar dan kom je toch bovenin nog in de problemen. Eens proberen met de sterren omhoog.

```
3E14 1B 8C E4 8C
```

Aanpassen naar:

```
3E14 1B 6C E4 6C
```

Nog steeds vrij lastig. Nog wat meer testen.

De sterren weer terugzetten. En dan een aantal aanpassingen aan het gedrag van de fireballs zoals bij DK Pacman Crossover hier ook doorvoeren en dan vier Koopa's maximaal. En dan wellicht kijken of de vuurbal sneller kan (door steeds met twee te verplaatsen) en dan in dezelfde tijd twee keer zo hoog (en dus ook de volgende girder bestrijkt).

Maximaal 4 Koopa's.

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

Aanpassen:

```
317F FE04 CP #04
```

Ja, dat werkt.

Fireball moving left doet geen ladder mount check, daarom trager? Ja, ladder check met 25% kans, maar links bewegende springt dan naar einde en doet helemaal geen beweging. Zit hier:

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

Kijken wat er gebeurt als we dit stukje veranderen in NOP's. Ten eerste sneller en dus ook ladder af/op wanneer links bewegend. Maar wellicht nu teveel nadruk op linksaf gaan, want bij verlaten ladder altijd links af. Maar dat kan wellicht ook aangepast worden nog.

Aanpassen:

```
322D 000000 NOP,NOP,NOP
```

Ja, dat werkt.

De ladder omhoog gaat drie keer langzamer dan naar beneden. Dit aanpassen. Dat zit hier:

```
33F2 DD7E14 LD A,(IX+#14) ; \ Jump if it is not time to climb one pixel yet
33F5 A7 AND A ; |
33F6 C20134 JP NZ,#3401 ; /
33F9 DD361402 LD (IX+#14),#02 ; Reset ladder climb timer to 2
33FD DD350F DEC (IX+#0F) ; Decrement fireball's Y position, move up one pixel
3400 C9 RET ; return
```

Dus die check van #33F2 -#33F9 veranderen in NOP's. Dan niet overslaan 2 van de drie keer.

Nu ook ervoor zorgen dat fireball ook naar beneden mag wanneer lager dan jumpman. Want nu als jumpman boven in scherm dan blijven ze daar als gekken rondcirkelen.

Dat zit hier:

```
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 ; /
```

Dus de RET NC daar niet doen: altijd naar beneden gaan ongeacht van de Y-positie van jumpman. Dus #336B NOP maken.

Ja, dat gaat goed.

Zorgen dat ze niet meer freezing mode gaan. Dat ziet er namelijk nogal raar uit.

Freezer mode activeren wordt hier gedaan.

```
31B1 CDDD31 CALL #31DD ; Check if freezers should enter freezer mode  
en:
```

```
31DD 3A8063 LD A,(#6380) ; \ Return if internal difficulty is < 3, no freezers are allowed until difficulty 3.  
31E0 FE03 CP #03 ; |  
31E2 F8 RET M ; /  
  
31E3 CDF631 CALL #31F6 ; Check if we should enter freezer mode (25% probability every 256 frames of entering freezer mode)  
31E6 FE01 CP #01 ; \ Return if should not enter freezer mode  
31E8 C0 RET NZ ; /  
  
31E9 213964 LD HL,#6439 ; \ Set freezer indicator of 2nd fire to #02 to enable freezer mode  
31EC 3E02 LD A,#02 ; |  
31EE 77 LD (HL),A ; /  
  
31EF 217964 LD HL,#6479 ; \ Set freezer indicator of 4th fire to #02 to enable freezer mode  
31F2 3E02 LD A,#02 ; |  
31F4 77 LD (HL),A ; /  
31F5 C9 RET ; return
```

Deze routine wordt alleen vanaf #31B1 aangeroepen.

Testen geen freezers wanneer de call op #31B1 in 3 NOP's veranderd wordt.

De fireballs ook naar rechts laten gaan wanneer ze een trap verlaten. Dat afhankelijk maken van een random number.

Dat zit op twee plaatsen (1^e wanneer bovenaan ladder en 2^e wanneer onderaan ladder).

```
; Fireball at top of ladder  
337B DD360D00 LD (IX+#0D),#00 ; Set fireball direction to left  
337F DD7E19 LD A,(IX+#19) ; \ If freezer mode is engaged then set the freeze flag and return  
3382 FE02 CP #02 ; |  
3384 C0 RET NZ ; |  
3385 DD361D01 LD (IX+#1D),#01 ; |  
3389 C9 RET ; /
```

en

```
3394 DD360D00 LD (IX+#0D),#00 ; Fireball has reached the bottom, set the direction to left  
3398 C9 RET ; return
```

337B DD360D00 LD (IX+#0D),#00 ; Set fireball direction to left

3394 DD360D00 LD (IX+#0D),#00 ; Fireball has reached the bottom, set the direction to left

In beide gevallen de instructie aanpassen naar een call naar additional code en daar op basis van een random getal links of rechtsaf gaan en dan de IX+#0D daarmee aanpassen en returnen.

```
337B CDDF3F CALL #3FDF ; call additional code  
337E 00 NOP
```

en

```
3394 CDDF3F      CALL  #3FDF      ; call additional code
3397 00          NOP
```

en:

```
3FDF 3A1860      LD    A,(RngTimer1) ; load timer, a psuedo random number
3FE2 1F          RRA   ; roll right = 50% chance
3FE3 3805      JR    C,#AAAA      ; jump to label AAAA to go right
;
3FE5 DD360D00    LD    (IX+#0D),#00 ; Set fireball direction to left
3FE9 C9          RET   ; return
;
3FEA DD360D01    LD    (IX+#0D),#01 ; Set fireball direction to right – label AAAA
3FEE C9          RET   ; return
```

Dat gaat goed.

Nu versie v0.42 opgeslagen.

De fireball sneller omhoog laten gaan:

```
2279 DD7E05      LD    A,(IX+#05) ; load A with lava fireball Y-position
227C 3D          DEC   A          ; move one position up
```

Aanpassen:

```
2279 CDEF3F      CALL  #3FEF      ; call additional code
```

en:

```
3FEF DD7E05      LD    A,(IX+#05) ; load A with lava fireball Y-position
3FF2 3D          DEC   A          ; move one position up
3FF3 C9          RET   ; return
```

De fireball sneller omlaag laten gaan:

```
228F DD7E05      LD    A,(IX+#05) ; load A with lava fireball Y-position - DDDD
2292 3C          INC   A          ; move one position down
```

Aanpassen:

```
228F CDF43F      CALL  #3FF4      ; call additional code
```

en:

```
3FF4 DD7E05      LD    A,(IX+#05) ; load A with lava fireball Y-position
3FF7 3C          INC   A          ; move one position up
3FF8 C9          RET   ; return
```

En dan de vuurbal veel hoger laten gaan:

```
2283 FE98      CP      #98      ; is lava fireball at the top?
```

Gaat eigenlijk wel heel erg snel. Effect is eigenlijk wel weg.

En wanneer met de speed timer weer langzamer maken dan meteen weer halve snelheid en is het weer te langzaam. Daarnaast is zo hoog wel erg lastig.

Dus terug naar vorige versie.

We kunnen wel proberen met iets hogere vuurbal.

```
2283 FEA0      CP      #A0      ; is lava fireball at the top?
```

Is op zich wel leuk omdat dan ook de tweede girder niet veilig is. Maar toch niet doen voorlopig.

```
2283 FEB8      CP      #B8      ; is lava fireball at the top?
```

Dan nu kijken hoe de animatie zodanig gemaakt kan worden dat de girders openschuiven en niet beneden gestapeld worden. En dan verder laten vallen.

Hoe werkt dit globaal. Het einde wordt aangegeven door de Gamemode2 op #16 te zetten en terug te keren naar een ‘hoger level’.

In de overkoepelende main wordt op basis van Gamemode2 met waarde #16 de code op #1615 uitgevoerd.

Voor rivets wordt dan via een aantal conditionele sprongen gesprongen naar #1641:

```
1641 CDBD1D    CALL    #1DBD      ; check for bonus items and jumping scores, rivets
1644 3A8863    LD      A,(#6388)   ; load A with usually zero, counts from 1 to 5 when the level is complete

1647 EF        RST     #28      ; jump based on A

1648 B6 17      ; #17B6      ; 0
164A 69 30      ; #3069      ; 1
164C 39 18      ; #1839      ; 2
164E 6F 18      ; #186F      ; 3
1650 80 18      ; #1880      ; 4
1652 C6 18      ; #18C6      ; 5
```

Dus op basis van de waarde in #6388 worden verschillende fasen in de end of level animatie doorlopen.

Dit even testen met de debugger en stoppen en kijken wat er bij welke waarde gedaan wordt:

0. De girders vallen naar beneden

1. Er wordt even gewacht
2. Kong slaat op borst
3. Girder met Pauline zakt naar beneden
4. Kong valt naar beneden
5. Kong ligt beneden en draait met ogen, jumpman wordt verplaatst

Dus de aanpassing moet gedaan worden in het eerste deel (0). Dat roept de code op #17B6 aan. Maar dat doet allemaal dingen die maar éénmaal gedaan dienen te worden. Je kunt ook kijken in hoeverre Kong nog op zijn borst moet slaan. Nee Bowser valt gewoon naar beneden. Dus je zou het weghalen van de girders uit de code van (0) kunnen verwijderen en dan de code waarin Kong op zijn borst slaat aanpassen naar code op de girders weg te laten schuiven (van rechts naar links).

De code waar (0) naar verwijst staat op #17B6. Daar staan een heleboel regels met clear center video RAM. Geen idee wat welke doet. Proefondervindelijk vaststellen door steeds een deel te veranderen in NOP's.

Eerst de eerste:

```
17D3 21DA76 LD HL,#76DA ; load HL with center area of video ram
17D6 CD2618 CALL #1826 ; clear screen area
```

Dit zorgt ervoor dat de onderste girder niet verwijderd wordt.

Dan het tweede:

```
17D9 11473A LD DE,#3A47 ; load DE with start of table data
17DC CDA70D CALL #0DA7 ; draw the screen
```

Dit zorgt ervoor dat de onderste verplaatste girder niet getekend wordt.

Dus al deze blokken kunnen verwijderd worden:

```
17D3 21DA76 LD HL,#76DA ; load HL with center area of video ram
17D6 CD2618 CALL #1826 ; clear screen area
17D9 11473A LD DE,#3A47 ; load DE with start of table data
17DC CDA70D CALL #0DA7 ; draw the screen
17DF 21D576 LD HL,#76D5 ; load HL with center area of video ram
17E2 CD2618 CALL #1826 ; clear screen area
17E5 114D3A LD DE,#3A4D ; load DE with start of table data
17E8 CDA70D CALL #0DA7 ; draw the screen
17EB 21D076 LD HL,#76D0 ; load HL with center area of video ram
17EE CD2618 CALL #1826 ; clear screen area
17F1 11533A LD DE,#3A53 ; load DE with start of table data
17F4 CDA70D CALL #0DA7 ; draw the screen
17F7 21CB76 LD HL,#76CB ; load HL with center area of video ram
17FA CD2618 CALL #1826 ; clear screen area
17FD 11593A LD DE,#3A59 ; load DE with start of table data
1800 CDA70D CALL #0DA7 ; draw the screen
```

Eerst eens proberen met dit hele stuk veranderen in NOP's. Ja dat gaat goed, dan blijven de girders gewoon staan in deze fase en kunnen ze in een volgende fase weggeschoven worden.

Om de ruimte vrij te geven kan de code van #1803 t/m #1825 aangeschoven worden. Ja, dat werkt. Is wel niet zo'n heel groot stuk, maar toch. Aan het einde vaak behoeft aan alle ruimte die er maar is.

Nu versie v0.43 opgeslagen.

Dan kijken of dat stuk code waarin Kong op zijn borst slaat alleen in rivets gebruikt wordt. Zo ja, dan kan dat aangepast worden. Zo nee, dan moet er een nieuw stuk gemaakt worden.

Dat is de animatie (2) en die verwijst naar #1839. Nee die aanroep lijkt alleen vanuit de end level tabel van de rivets aangeroepen te worden.

De code daar werkt met een time op #6390. Eens kijken met de debugger hoe dat werkt. Watchpoint op code #1844 omdat daar alleen op bepaalde momenten van de timer iets gedaan wordt. Welke waarden heeft de timer dan? En hoeveel stappen? Als we dat weten, dan kunnen we dat wellicht gebruiken om de verschuifstappen te bepalen.

Begint op #88 en loopt met stappen omhoog tot #00. En bij elke stap wordt een animatie van Kong gedaan. Dat zijn dus 15 stappen (de laatste doet niks).

De girders bestaan uit 12 karakters. Dus we zouden met een paar stappen minder kunnen volstaan. Hoe kun je dan de girders met een slimme manier wissen?

Valt me nu op dat bij de eerder verwijderde blokken niet alleen de girders, maar ook de trappen verwijderd worden. Die staan er nu nog en zouden dan met wegschuiven van de girders ook langzaam van rechts naar links verdwijnen. Dat ziet er wel raar uit. Eigenlijk zou het dus zo gemaakt moeten worden dat de trappen al wel weggehaald zijn in het eerdere stuk.

Eens kijken of dat kan door het te wissen blok aan te passen, of dat deze wisactie ook op andere plekken gebruikt wordt. Ja, die wordt ook nog ergens anders gebruikt: bij Game Over en bij het wissen van het bovenste deel bij level end van rivets wanneer Pauline naar beneden geplaatst wordt. Maar we hebben niet meer de code nodig om de girders beneden in te tekenen, dus dat is misschien genoeg om een nieuwe routine te maken die een kleiner gebied wist.

Aanpassen:

```
17D3  21DB76      LD     HL,#76DB      ; load HL with center area of video ram
17D6  CD0E18      CALL   #180E      ; clear screen area
17D9  21D676      LD     HL,#76D6      ; load HL with center area of video ram
17DC  CD0E18      CALL   #180E      ; clear screen area
17DF  21D176      LD     HL,#76D1      ; load HL with center area of video ram
17E2  CD0E18      CALL   #180E      ; clear screen area
17E5  21CC76      LD     HL,#76CC      ; load HL with center area of video ram
17E8  CD0E18      CALL   #180E      ; clear screen area
;
17EB  215C38      LD     HL,#385C      ; load HL with start of kong graphic table data
```

```

17EE CD4E00    CALL #004E      ; update kong's sprites
17F1 210869    LD   HL,#6908    ; load HL with start of kong sprites
17F4 0E44      LD   C,#44      ; load offset of #44
17F6 FF        RST #38       ; move kong
17F7 210569    LD   HL,#6905    ; load HL with girl's sprite
17FA 3613      LD   (HL),#13    ; set girl's sprite
17FC 3E20      LD   A,#20      ; A := #20
17FE 320960    LD   (WaitTimerMSB),A ; set timer to #20
1801 3E80      LD   A,#80      ; A := #80
1803 329063    LD   (#6390),A  ; store into timer ????
1806 218863    LD   HL,#6388    ; load HL with end of level counter
1809 34        INC  (HL)      ; increase counter
180A 22C063    LD   (#63C0),HL ; store into ???
180D C9        RET           ; return
;
180E 11DCFF    LD   DE,#FFDC    ; load DE with offset for each column
1811 0E0E      LD   C,#0E      ; for C = 1 to #0E
1813 3E10      LD   A,#10      ; A := #10 (clear space on screen)
;
1815 0604      LD   B,#04      ; for B = 1 to 4
;
1817 77        LD   (HL),A    ; store A into (HL) - clears the screen element
1818 23        INC  HL       ; next HL
1819 10FC      DJNZ #182F    ; next B
;
181B 19        ADD  HL,DE    ; add offset to HL
181C 0D        DEC  C       ; next C
181D C21518    JP   NZ,#1815  ; loop until done
;
1820 C9        RET           ; return
;
1821-1824     4xNOP

```

Kijken of alleen de trappen nu weggehaald worden.

Ja dat gaat nu goed. Haalt alleen de trappen weg.

Nu versie v0.44 opgeslagen.

En dan nu weer kijken hoe we de girders kunnen weghalen. In principe kan eenzelfde soort routine gemaakt worden die een balk van één karakter breed van onder naar boven weghaalt. Dat betekent als ik goed tel dat er 16 karakters onder elkaar verwijderd dienen te worden (4 girders + 3 gaten van 4 karakters+4+12=16).

Alleen hoe zorg je ervoor dat je bij aanroep een verschoven startpositie krijgt. Op basis van de counter zou wellicht wel kunnen, maar dan in een register maar max 256 waarde. Maar je moet grotere waarden aankunnen en dat wordt wel heel erg lastig.

Daarom checken op beginwaarde, als die gezet, dan twee geheugenplaatsen met de start videolocatie vullen en steeds na een wisactie de nieuwe positie opslaan.

Twee geheugenplaatsen die gebruikt kunnen worden: #64B0 en #64B1.

Aanpassen:

```
1839 219063 LD HL,#6390 ; load HL with timer
183C 34 INC (HL) ; increase, at zero?
183D CA6218 JP Z,#1862 ; yes, animation finished, skip ahead - AAAA
;
1840 7E LD A,(HL) ; load A with the timer value
1841 E607 AND #07 ; mask bits, now between 0 and 7, at zero?
1843 C0 RET NZ ; no, done for now, return
;
1844 7E LD A,(HL) ; reload A with the timer value
1845 FE88 CP #88 ; start of the animation?
1847 2006 JR NZ,#184E ; no, skip ahead - BBBB
;
1849 214B75 LD HL,#754B ; load HL with start video location
184C 22B064 LD (#64B0),HL ; save video location
;
184F 2AB064 LD HL,(#64B0) ; load video location - BBBB
;
1852 3E10 LD A,#10 ; A := #10 (clear space on screen)
;
1854 0610 LD B,#10 ; for B = 1 to #10
;
1856 77 LD (HL),A ; store A into (HL) - clears the screen element
1857 23 INC HL ; next HL
1858 10FC DJNZ #XXXX ; next B
;
185A 111000 LD DE,#0010 ; load DE with offset
185D 19 ADD HL,DE ; add offset to HL
;
185E 22B064 LD (#64B0),HL ; save video location
1861 C9 RET ; done for now, return
;
1862 3E20 LD A,#20 ; A := #20 - AAAA
1864 320960 LD (WaitTimerMSB),A ; store into timer
1867 218863 LD HL,#6388 ; load HL with end of level counter
186A 34 INC (HL) ; increase counter
186B C9 RET ; return
```

Gaat goed maar loopt drie te ver door. Dus start aanpassen.

Nu start op #80, dat aanpassen naar #98 en dan de initialisatie op #A0.

Aanpassen:

```
1801 3E98        LD      A,#80        ; A := #98
```

en:

```
1845 FEA0        CP      #A0        ; start of the animation?
```

Ja, dat werkt nu goed.

Nu versie v0.45 opgeslagen.

Wel nu nog Kong verder naar beneden laten vallen en in de lava laten verdwijnen. Dan is ook de animatie van het rollen van de ogen niet meer benodigd. Maar wellicht iets anders doen dan? Ook is het geluid bij het vallen verdwenen. Dat terugbrengen?

Kan het zijn dat het weggevallen geluid door timing komt? Dat het vorige geluid nog niet afgespeeld is en het daardoor niet goed gaat.

Testen door de oude loopwaarden terug te zetten. Ja, dan is er wel geluid. Kunnen we misschien oplossen door meer te wissen (de rivetgaten ook meenemen)?

Aanpassen:

```
1801 3E88        LD      A,#88        ; A := #88
```

en:

```
1845 FE90        CP      #90        ; start of the animation?
```

en:

```
1849 212B75      LD      HL,#752B    ; load HL with start video location
```

Ja, dat gaat nu goed.

Kijken naar het verder laten vallen van Kong. Dat wordt hier gedaan:

```
1880 210B69      LD      HL,#690B    ; load HL with kong start sprite
1883 0E01        LD      C,#01      ; load C with 1 pixel to move
1885 FF          RST     #38      ; move kong
1886 3A1B69      LD      A,(#691B)  ; load A with ????
1889 FED0        CP      #D0      ; == #D0 ?
188B C0          RET      NZ      ; no, return
```

Als de waarde op 1889 aangepast wordt naar een groter getal, dan schuift hij verder naar beneden. Je kan zelfs een waarde als #10 invullen en dan schuift hij door en komt boven aan het scherm weer tevoorschijn.

Er zijn twee problemen mee. De sprites zijn aan de onderkant van de lava te zien omdat dat doorzichtig is (zwart is geen zwart). Misschien op te lossen door de lava sprite aan te passen zodat deze helemaal tot onderaan het zichtbare gedeelte afdekt. Moet kunnen. En dan de schuifroutine misschien zodanig aanpassen dat sprites met een bepaalde hele grote Y-waarde niet getoond worden?

Het vallen van DK gebeurd door in een loop de Y-positie van de sprites steeds met 1 te verhogen.

Dat aanpassen van de sprites wordt gedaan door een call naar RST #38:

```
0038 110400 LD DE,#0004 ; load offset of 4 to add
003B 060A LD B,#0A ; for B = 1 to #A (10 decimal)

003D 79 LD A,C ; Load A with C
003E 86 ADD A,(HL) ; Add the contents of HL into A
003F 77 LD (HL),A ; put back into HL, this increases the value in HL by C
0040 19 ADD HL,DE ; next HL to do will be 4 more than previous
0041 10FA DJNZ #003D ; next B

0043 C9 RET ; return
```

Die doet dus een verplaatsing van 10 opeenvolgende sprites met de afstand die in C staat. Afhankelijk van de waarde in HL kan dit gebruikt worden voor aanpassen van de X-positie of aanpassen van de Y-positie.

Waar wordt deze nog meer aangeroepen? O.a. voor het horizontaal verplaatsen van Kong op het pies board. Maar die hebben we niet meer en conveyor's mechanisme is verwijderd, dus daar zal het niet meer voor gebruikt worden. Daarnaast gebruikt voor klimmen van Kong op de ladder en springen naar links in het intrascherf, maar dat is ook verwijderd. Daarnaast kleine verplaatsingen her en der.

Maar dus allemaal niet in de uiterste gebieden met een hoge X- of Y-positie.

Kijken of de code aangepast kan worden dat wanneer de Y-waarde boven F8 komt dat dan de sprite onzichtbaar gemaakt wordt door een lege sprite als sprite value te zetten.

Aanpassen:

```
003D C3BF0B JP #0BBF ; jump to additional code - AAAA
```

en:

```
0BBF 79 LD A,C ; load A with C - AAAA
0BC0 86 ADD A,(HL) ; add the contents of HL into A
0BC1 77 LD (HL),A ; put back into HL
```

```

;
0BC2 FEF8      CP    #F8          ; is bottom of screen reached?
0BC4 2006      JR    NZ,#BBBB     ; no, jump forward - BBBB
;
0BC6 2B         DEC   HL           ; go two positions back
0BC7 2B         DEC   HL
;
0BC8 3664      LD    (HL),#64    ; set to empty sprite
;
0BCA 23         INC   HL           ; go two position forward
0BCB 23         INC   HL
;
0BCC C34000     JP    #0040      ; jump back - BBBB

```

Ja, dat gaat heel goed. Wordt nog wel een hoofd getekend met rollende ogen, maar dat is de volgende fase. Die nog aanpakken.

Nu versie v0.46 opgeslagen.

Na in de lava vallen dan wordt er een hoofd getoond met draaiende ogen. Dat hoeft niet meer en er wordt een dizzy sprite geanimeerd. Die zou heel leuk vlak boven de lava getoond kunnen worden.

Dat is de code op #1880.

```

188C 3E20      LD    A,#20        ; A := #20
188E 321969     LD    (#6919),A    ; store into kong's face sprite

```

Deze code niet doen: aanpassen naar NOP's.

Huh, dat gaat helemaal fout. Gezicht wordt inderdaad niet getekend, maar de animatie van deizziness is er niet meer en dan terug naar titelscherm? Hoe kan dat? Wanneer teruggezet, dan werkt het wel weer. Hoe kan dat nou?

Anders doen: bewust gewoon weer op de lege sprite zetten: #64.

Ook dat werkt niet. Zit er een check op de waarde om de ogen te laten rollen??? Dat ook aanpassen dan?

Wordt in de laatste afhandeling hier nog iets mee gedaan? Ja, dezelfde sprite wordt hier nog iets mee gedaan:

```

18D8 211969     LD    HL,#6919    ; load HL with ???
18DB 46         LD    B,(HL)     ; load B with this value
18DC CBA8       RES   5,B        ; clear bit 5 of B
18DE AF         XOR   A          ; A := 0
18DF CD0930     CALL  #3009     ; ???
18E2 F620       OR    #20        ; turn on bit 5
18E4 77         LD    (HL),A    ; store result

```

Waarschijnlijk verwacht die code een bepaalde waarde en loopt het stuk als je er iets anders in stopt.

Wat als we dat stuk ook helemaal NOP'en? Deze code wordt ook gebruikt voor Kong die de ladder opklimt bij beëindigen andere boards. Maar wat doet het daar dan?

Nee dat gaat goed. Daar doet het niets.

Wel de sprite voor de vonken een stuk naar beneden plaatsen.

Dat is deze code:

```
1894 367F LD (HL),#7F ; set sprite X value
1896 2C INC L ; next
1897 3639 LD (HL),#39 ; set sprite color
1899 2C INC L ; next
189A 3601 LD (HL),#01 ; set sprite value
189C 2C INC L ; next
189D 36D8 LD (HL),#D8 ; set sprite Y value
```

Aanpassen:

```
189D 36EC LD (HL),#EC ; set sprite Y value
```

Ja, dat is nu goed.

Alleen jumpman, het hartje en Pauline staan niet in het midden. Dit aanpassen zou het wel een mooier maken.

Daar zit ook nog logica achter aan welke kant jumpman de rivets finished, maar dat is nu niet meer relevant omdat hij dat nu alleen nog maar aan de rechterkant kan doen.

De heart sprite wordt hier gezet:

```
1923 21236A LD HL,#6A23 ; load HL with heart sprite
1926 3640 LD (HL),#40 ; set heart sprite Y position
1928 2B DEC HL ; decrement HL
1929 3609 LD (HL),#09 ; set heart sprite color
192B 2B DEC HL ; decrement HL
192C 3676 LD (HL),#76 ; set heart sprite
192E 2B DEC HL ; decrement HL
192F 368F LD (HL),#8f ; set heart sprite X position
```

Aanpassen:

```
192F 3680 LD (HL),#80 ; set heart sprite X position
```

Ja, die staat nu in het midden.

Dan Pauline verplaatsen.

Dat lijkt hier te zitten:

```
18AB 110400 LD DE,#0004 ; load counters
18AE 012802 LD BC,#0228 ; load counters
18B1 210369 LD HL,#6903 ; set sprite girl table data Y position
18B4 CD3D00 CALL #003D ; move the girl down
```

Wel een ietwat vreemde manier om te doen. Kan eenvoudiger. Eerst even testen of het inderdaad deze code is. Ja, dit is de code.

Aanpassen:

```
18AB 210069 LD HL,#6900 ; load HL with Pauline sprite X-position
18AE 3668 LD (HL),#68 ; set Pauline X-position
18B0 23 INC HL
18B1 23 INC HL
18B2 23 INC HL ; load HL with Pauline sprite Y-position
18B3 3650 LD (HL),#50 ; set Pauline Y-position
18B5 0000 NOP,NOP
```

Leuk geprobeerd, maar gaat niet goed. Pauline bestaat uit twee sprites, vandaar de manier van schuiven: twee sprites tegelijk. Moeten dus allebei aangepast worden. Toch iets meer code voor nodig.

```
18AB 3E68 LD A,#68
18AD 320069 LD (#6900),A ; set Pauline head X-position
18B0 320469 LD (#6904),A ; set Pauline body X-position
18B3 C3A922 JP #22A9 ; jump to additional code
18B6 00 NOP
```

en:

```
22A9 3E50 LD A,#43
22AB 320369 LD (#6903),A ; set Pauline head Y-position
22AE 3E40 LD A,#53
22B0 320769 LD (#6907),A ; set Pauline body Y-position
22B3 C3B718 JP #18B7
```

Ja, gaat goed. Alleen nog een aantal pixels te hoog. Aangepast bovenstaand. Nu wel goed.

Nu nog jumpman een stuk naar links verplaatsen. Ongeveer #08 naar links.

Dat lijkt hier te zitten:

```

18EE 3E50      LD    A,#50          ; A := #50
18F0 324F69    LD    (#694F),A    ; store into mario sprite Y value
18F3 3E00      LD    A,#00          ; A := 0
18F5 324D69    LD    (#694D),A    ; store into mario sprite value
18F8 3E9F      LD    A,#9F          ; A := #9F
18FA 324C69    LD    (#694C),A    ; set mario sprite X value at #9F
18FD 3A0362    LD    A,(#6203)     ; load A with mario X position
1900 FE80      CP    #80          ; < 80 ?
1902 D20F19    JP    NC,#190F     ; yes, skip next 4 steps

1905 3E80      LD    A,#80          ; A := #80
1907 324D69    LD    (#694D),A    ; store into mario sprite value
190A 3E5F      LD    A,#5F          ; A := #5F
190C 324C69    LD    (#694C),A    ; store into mario sprite X value

```

En dat doet meteen ook een plaatsing links of rechts. Maar dat is nu helemaal niet meer nodig.

Dat aanpassen en meteen de spritekleur weer terugzetten omdat deze bij onsterfelijkheid ook alleen de grijstinten kunnen zijn.

Aanpassen:

```

18F8 3E97      LD    A,#97          ; A := #97
18FA 324C69    LD    (#694C),A    ; set jumpman X position
18FD 3E02      LD    A,#02          ; A := #02
18FF 324E69    ID    (#694D),A    ; reset jumpman color
1902 t/m 190E   NOP's

```

Ja, dat gaat goed.

En de kleuraanpassing van jumpman ook.

Nu versie v.0.47 opgeslagen.

Nu eerst kijken dat bij einde rivets in wereld 8-4 de melding 'GAME COMPLETED' wordt getoond en daarna de speler zijn initialen kan ingeven. In de high score tabel ook de behaalde wereld aangeven en wanneer game completed dit doen met speciale aanduiding *-* of 8*4 i.p.v. 5-3.

Eerst de high score tabel uitbreiden met WORLD.

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 C31D28    JP   #281D    ; jump to additional code – jump to label AAAA
13F1 00        NOP

```

en:

```

281D 3610      LD   (HL),#10    ; store #10 into memory at (HL) – label AAAA
281F 23        INC  HL        ; next HL
2820 3610      LD   (HL),#10    ; store #10 into memory at (HL) – label AAAA
2822 23        INC  HL        ; next HL
;
2823 3A2962    LD   A,(#6229)  ; load A with level nr
2826 77        LD   (HL),A    ; store level nr into memory at (HL)
2827 23        INC  HL        ; next HL
;
2828 3A3062    LD   A,(#6230)  ; load A with game completed indication
282B FE01      CP   #01        ; game completed?
282D 3E7A      LD   A,#7A        ; load star character
282F 2802      JR   Z,#CCCC    ; yes, skip next step
2831 3E2C      LD   A,#2C        ; load hyphen
2833 77        LD   (HL),A    ; store character into memory at (HL)
2834 23        INC  HL        ; next HL
;
2835 3A2762    LD   A,(#6227)  ; load A with board nr
2838 77        LD   (HL),A    ; store board nr into memory at (HL)
2839 23        INC  HL        ; next HL
;
283A 0608      LD   B,#08        ; for B = 1 to #08 – label BBBB
;
283C 3610      LD   (HL),#10    ; store #10 into memory at (HL)
283E 23        INC  HL        ; next HL
283F 10FB      DJNZ #1640    ; next B – jump back to label BBBB
;
2841 C3F213    JP   #13F2    ; jump back

```

Geheugenplaats #6230 gebruiken om aan te geven of game completed behaald is. Zo ja, dan een ster i.p.v. een hyphen tekenen.

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.

Nu versie v0.48 opgeslagen.

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	W	O	R	L	D	N	A	M	E
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

```
#379D: 52 77 7B 10 10 23 13 1F 22 15 10 10 27 1F 22 1C 14 10 10 1E 11 1D 15 3F
```

Ja dat is goed. Alleen nog karakter voor # maken en invullen: #7B.

Nu versie v0.49 opgeslagen.

En dan in de initiële high score tabel de tekst DESIGN BY PAUL GOES opnemen.

Onderste regel: DESIGN BY PAUL GOES:

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

```

3565: 94 77 22 15 23 25 1C 24 10 10 10 10 10 00 00 10 10 2C 10 10 00 00 10 10 10 3F 00 50 76 00 F4 76
3587: 96 77 12 11 22 22 15 1C 23 10 10 10 00 00 00 00 10 10 10 00 00 00 00 10 10 3F 00 00 61 00 F6 76
35A9: 98 77 20 19 15 23 10 10 10 10 10 00 00 00 00 10 10 10 00 00 00 00 10 10 3F 00 50 59 00 F8 76
35CB: 9A 77 23 20 22 19 1E 17 23 10 10 10 00 00 00 00 10 10 10 00 00 00 00 10 10 3F 00 50 50 00 FA 76
35EE: 9C 77 10 10 10 14 15 23 19 17 1E 10 12 29 10 20 11 25 1C 10 17 1F 15 23 10 10 10 3F 00 00 43 00 FC 76

```

De DESIGN BY PAUL GOES staat te ver naar links. Moet zijn:

```
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 v0.50 opgeslagen.

En dan nu zodanig aanpassen dat bij finish rivets getest wordt op world 8-4. Als daaraan voldaan wordt, een melding GAME COMPLETED geven, de geheugenplaats #6230 op #01 zetten en juiste gamemode zetten om spel te beëindigen.

De GAME COMPLETED message in het vrije slot 6 op #3657:

```
3657 2C 1F           6           ; #1F2C "GAME COMPLETED"
```

En dan ergens de code aanmaken:

```
1F2C D2 76 90 91 92 93 10 94 95 92 96 97 93 98 93 99 3F  
G A M E     C O M P L E T E D
```

De GAME COMPLETED message plaatsen bij het verplaatsen van Pauline en Jumpman. Dat zit hier:

```
18F8 3E97      LD   A,#97      ; A := #97  
18FA 324C69      LD   (#694C),A    ; set jumpman X position  
18FD 3E02      LD   A,#02      ; A := #02  
18FF 324E69      ID   (#694D),A    ; reset jumpman color  
1902 t/m 190E      NOP's
```

Aanpassen:

```
1902 CD532F      CALL #2F53      ; jump to additional code
```

en:

```
2F53 3A2962      LD   A,(#6229)  ; load A with world  
2F56 FE08      CP   #08      ; is this world 8?  
2F58 C0          RET  NZ      ; no, return  
;  
2F59 3E01      LD   A,#01      ; load A with #01  
2F5B 323062      LD   (#6230),A  ; set game completed indication  
2F5E 110603      LD   DE,#0306  ; load DE with task code "GAME COMPLETED"  
2F61 CD9F30      CALL #309F    ; insert task  
;  
2F64 C9          RET
```

Ja, dat gaat goed. Wel voor nu even testen met world 2 i.p.v. world 87. Later nog aanpassen.

De kleur van de tekst is echter grijs en zou mooier zijn als het wit zou zijn. Dan moet je de benodigde letters in een andere kleur aanmaken.

Ja, dat ziet er nu goed uit.

Ook de GAME OVER tekst in het wit maken:

Die start op 368B:

```
368A:          96 76 17 11 1D      ..GAM
3690: 15 10 10 1F 26 15 22 3F 94 76 20 1C 11 29 15 22 E..OVER...PLAYER
```

Aanpassen:

```
368B 96 76 90 91 92 93 10 10 95 9A 93 9B 3F
```

Die is nu ook wit.

Nu versie v0.51 opgeslagen.

Nu nog aanpassen dat aan het einde van de animatie de gamemode zodanig gezet wordt dat het hetzelfde is als game over en dat de speler de mogelijkheid krijgt om eventueel zijn/haar initialen in te geven.

De GameMode2 bepaalt e.e.a.

De code op #12F2 (Gamemode2 = #0E) wordt gedaan op het moment dat jumpman dood gaat en dan wordt daar er een leven vanaf gehaald en als het aantal levens dan 0 is, dan wordt de tekst Game Over getoond en gekeken of de de speler zijn/haar initialen mag ingeven. Deze code houd ook nog eens rekening met de situatie van een 2-player game waarbij de andere speler eventueel nog kan/mag doorspelen.

Wat als we i.p.v. het tonen van de tekst, aan het einde van de rivet animatie het aantal levens op 1 zetten en dan de Gamemode2 op #0E zetten? Dan zou dit getriggert moeten worden. Geeft dan nog wel de melding Game Over, maar dat kan altijd nog conditioneel gemaakt worden.

De GameMode2 aan het einde wordt hier gezet:

```
1962 210960 LD    HL,WaitTimerMSB ; load HL with timer
1965 36E0     LD    (HL),#e0      ; set timer to #E0
1967 23       INC   HL        ; increase HL to GameMode2
1968 3608     LD    (HL),#08      ; set game mode2 to 8
196A C9       RET            ; return
```

Dus eigenlijk dat afhankelijk maken van de game completed indicatie (#6230).

Aanpassen:

```
1967 CD652F     CALL   #2F65      ; call additional code - AAAA
```

en:

```

2F65 23          INC   HL           ; increase HL to GameMode2 - AAAA
2F66 3608        LD    (HL),#08   ; set GameMode2 to #08
;
2F68 3A3062      LD    A,(#6230)  ; load A with game completed indication
2F6B FE01        CP    #01         ; game completed ?
2F6D C0          RET   NZ           ; no, return
;
2F6E 3E01        LD    A,#01       ; load A with #01
2F70 322862      LD    (#6228),A  ; set number of lives to #01
2F73 360E        LD    (HL),#0E   ; set GameMode2 to #0E
2F75 C9          RET   ; return

```

Ja, dat gaat goed. Je mag de initialen ingeven en die houd ook netjes rekening met de game completed en zet bij de score een sterretje.

Nu versie v0.52 opgeslagen.

Misschien wel enigszins raar dat hij ook nog GAME OVER eronder zet. Of niet en is dat juist heel duidelijk??? Op zich wel duidelijk. De GAME is ook inderdaad OVER.

Gaat nog niet goed. Bij spelen van een nieuw spel geeft hij ook GAME OVER bij eindigen van rivets. Dat is niet goed. Wordt de game completed indicator niet gereset? Of is de code niet goed? Testen door een nieuwe game te starten. Sowieso is de game completed indicator niet specifiek voor een speler, dus in een twee spelers spel gaat dat ook niet goed. De oplossing daarvoor is om er een single player game van te maken. Hebben we al eerder gedaan.

Even getest met een nieuwe game. Daar krijg je geen GAME OVER. Dus lijkt erop dat de game completed indicator niet gereset wordt? Ja, inderdaad. Deze wordt niet gereset. Maar andere zaken als levens, enz worden wel gereset. Waar wordt dat gedaan?

Dat wordt hier gedaan:

```

09AB 214060      LD    HL,P1NumLives   ; load HL with source data location
09AE 112862      LD    DE,#6228       ; load DE with destination data location. start with remaining lives
09B1 010800      LD    BC,#0008       ; byte counter set to 8
09B4 EDB0        LDIR            ; copy (HL) into (DE) from P1NumLives to P2NumLives into #6228 to #622F

```

Game completed indicator is #6230. Dus dat is 1 geheugenplaats verder in het blok dat gewist wordt.

Aanpassen:

```
09B1 010900      LD    BC,#0009       ; byte counter set to 9
```

Ja, dat gaat goed.

Nu versie v0.53 opgeslagen.

Vreemd is wel dat bij het game completed van world 2 (nu nog ingesteld) dit geregistreerd wordt als world 3*1. Dus die zijn al vooruit gezet voor het afhandelen. En ook nog even checken of de bonus punten er nog bijgeteld worden.

Ja dat zit vlak voor de aanpassing van de GameMode2 op lijn #1967. Maar dat hele blok is in ons geval niet nodig. Anders doen.

Herstellen:

1967	23	INC	HL	; increase HL to GameMode2
1968	3608	LD	(HL),#08	; set game mode2 to 8

Aanpassen:

193D	C3652F	JP	#2F65	; jump to additional code
------	--------	----	-------	---------------------------

en:

2F65	2A2A62	LD	HL,(#622A)	; load HL with contents of #622A
;				
2F68	3A3062	LD	A,(#6230)	; load A with game completed indication
2F6B	FE01	CP	#01	; game completed ?
2F6D	C24019	JP	NZ,#1940	; no, jump back
;				
2F70	3E01	LD	A,#01	; load A with #01
2F72	322862	LD	(#6228),A	; set number of lives to #01
;				
2F75	210960	LD	HL,WaitTimerMSB	; load HL with timer
2F78	36E0	LD	(HL),#E0	; set timer to #E0
2F7A	23	INC	HL	; increase HL to GameMode2
2F7B	360E	LD	(HL),#0E	; set game mode2 to #0E
2F7D	C9	RET		; return

Ja, gaat nu goed. De wereld wordt goed weergegeven bij completed game in de high score list. Ook de afhandeling gaat goed en bij een volgende game is de game completed indication weer gereset.

[Nu versie v0.54 opgeslagen.](#)

Er nu een one player game van maken.

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 even ervoor zorgen dat de GAME COMPLETED pas bij wereld 8 gegeven wordt.

```
2F56 FE08 CP #08 ; is this world 8?
```

En dan zo maken dat je in elke wereld steeds het barrel board en het rivets board speelt.

3A65: 01 04	; level 1
3A67: 01 04 01	; level 2
3A6A: 04 01 04 01	; level 3
3A6E: 04 01 04 01 04	; level 4
3A73: 01 04 01 04 01 04	; level 5 +
3A79: 7F	; end code

Ja, dat gaat goed. Game completed by world 8-4 en registratie met een ster.

De Koopa's bewegen wel erg snel op een trap op en neer. Dit iets langzamer maken? Kan dat?

Dat zit hier:

```
33E7 CD0934    CALL   #3409      ; Animate the fireball
33EA DD7E0D    LD      A,(IX+#0D)  ; \ Jump if fireball is moving down the ladder
33ED FE08      CP      #08       ; |
33EF C20534    JP      NZ,#3405   ; /
33F2 DD7E14    LD      A,(IX+#14)  ; \ Jump if it is not time to climb one pixel yet
33F5 A7        AND     A          ; |
33F6 C20134    JP      NZ,#3401   ; /
33F9 DD361402  LD      (IX+#14),#02 ; Reset ladder climb timer to 2
33FD DD350F    DEC     (IX+#0F)   ; Decrement fireball's Y position, move up one pixel
3400 C9        RET     .           ; return
3401 DD3514    DEC     (IX+#14)  ; Decrease ladder climb timer
3404 C9        RET     .           ; return
3405 DD340F    INC     (IX+#0F)  ; Increment fireball's Y position, move down one pixel
3408 C9        RET     .           ; return
```

Deze code is al aangepast door #33F2 t/m #33F8 NOP's te maken. Daardoor is de snelheid omhoog even snel als de snelheid naar beneden geworden.

Je zou dit kunnen aanpassen door terug te gaan naar de originele code en dan de eerste conditionele jump weghalen (voor omhoog en omlaag dezelfde code en in beide gevallen met een timer waarmee de snelheid bepaald kan worden).

Dus #33EA t/m #33F1 NOP's maken.

Nee dat gaat niet goed. Dan altijd omhoog. Altijd dezelfde code waarbij er een decrement gedaan wordt van de fireball's position.

Aanpassen:

```
33E7 CD0934    CALL   #3409      ; Animate the koopa
;
33EA DD7E14    LD      A,(IX+#14)  ; Load A with the climb timer
33ED A7        AND     A          ; is it time to move?
33EE 2804      JR      Z,#AAAA   ; yes, jump ahead - AAAA
;
33F0 DD3514    DEC     (IX+#14)  ; Decrease ladder climb timer
33F3 C9        RET     .           ; return
```

```

;
33F4 DD361402 LD (IX+#14),#02 ; Reset ladder climb timer to 2 - AAAA
;
33F8 DD7E0D LD A,(IX+#0D) ; load A with koopa direction
33FB FE08 CP #08 ; is it moving down?
33FD 2004 JR NZ,#3405 ; yes, jump ahead - BBBB
;
33FF DD350F DEC (IX+#0F) ; Decrement fireball's Y position, move up
3402 C9 RET ; return
;
3403 DD340F INC (IX+#0F) ; Increment fireball's Y position, move down - BBBB
3406 C9 RET ; return
;
3407 0000 NOP,NOP

```

Met reset van timer naar 2 is het te langzaam. Met een reset naar 1 is het goed.

```
33F4 DD361401 LD (IX+#14),#01 ; Reset ladder climb timer to 1
```

Nu versie v0.55 opgeslagen.

Nu even een tijdje in de ijskast qua ontwikkeling. Wel een speelbare versie, zodat je af en toe kunt spelen en testen of deze twee boards interessant zijn, speelbaar zijn en/of aanpassing nodig hebben.

Misschien een idee om de koopa's niet naar beneden te laten gaan wanneer ze op de 2e girder zitten. En wellicht de vuurbal random een stuk hoger laten gaan. Dus 50% met de huidige hoogte en 50% met een hoogte voorbij de volgende girder.

Het volgende was al aangepast t.a.v. het naar beneden gaan van de koopa's:

```

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 ; /

```

Dus de RET NC daar niet doen: altijd naar beneden gaan ongeacht van de Y-positie van jumpman.
Dus #336B NOP maken.

Maar eigenlijk dit aanpassen naar een vergelijking t.a.v. een bepaalde Y-waarde (i.p.v. een vergelijking met de Y-waarde van jumpman).

Aanpassen:

```

3363 3EC8 LD A,#C8 ; load A with threshold Y
3365 00 NOP
3366 47 LD B,A ; store in B

```

```

3367 DD7EOF      LD   A,(IX+#0F)    ; load A with koopa Y
336A 90          SUB  B             ; subtract the threshold Y
336B D0          RET   NC           ; koopa Y > treshold Y (lower) - no ladder, return

```

Nog wel even met de debugger checken welke treshold Y gebruikt moet worden.

Koopa op uiterste puntje eerste schuine girder bij barrels board is #D3.

Koopa op eerste girder bij rivets board is #C8.

Dus ook bij het barrel board niet naar beneden gaan.

De treshold Y #C8 maken.

Ja, dat gaat goed. Gaat niet meer naar beneden.

Dan nu kijken of de fireball soms hoger kan gaan.

De check op de top van de fireball zit op #2283 en #2285.

```

;
; move fireball going in the up direction
;
2279 DD7E05      LD   A,(IX+#05)    ; load A with lava fireball Y-position
227C 3D          DEC  A             ; move one position up
227D DD7705      LD   (IX+#05),A    ; store new lava fireball Y-position
2280 FD7703      LD   (IY+#03),A    ; store new lava fireball Y-position
;
2283 FEB8        CP   #B8          ; is lava fireball at the top?
2285 B8          RET  NZ           ; no, done for now, return
;
2286 DD360C02    LD   (IX+#0C),#02  ; set lava fireball direction down
228A FD360155    LD   (IY+#01),#55  ; set sprite value – lava fireball down
228E C9          RET   NZ           ; done for now, return

```

Dit aanpassen naar een jump naar additional code en daar eerst testen op #B8. Als #B8 dan random number. Als even dan return (doorgaan), als oneven dan jump naar #2286 (direction down zetten). Als niet #B8 dan testen op #A8. Als niet dan return (geen #B8 èn geen #A8 dus dan door omhoog). Als wel #A8 dan jump naar #2286 (direction down zetten).

Aanpassen:

```
2283 C3EA24      JP   #24EA        ; jump to additional code
```

en:

```

24EA FEB8        CP   #B8          ; is lava at low top?
24EC 2009        JR   NZ           ; no, jump forward - AAAA
;
24EE 3A1860      LD   A,(RngTimer1) ; load A with random number
24F1 E601        AND  #01          ; mask bits now random between 0 and 1
24F3 C0          RET  NZ           ; if zero, return , continue upward

```

```

24F4 C38622      JP    #2286      ; jump back, change direction downward
;
24F7 FE98        CP    #98       ; is lava at high top? - AAAA
24F9 C0          RET   NZ       ; no, continue upward
24FA C38622      JP    #2286      ; jump back, change direction downward

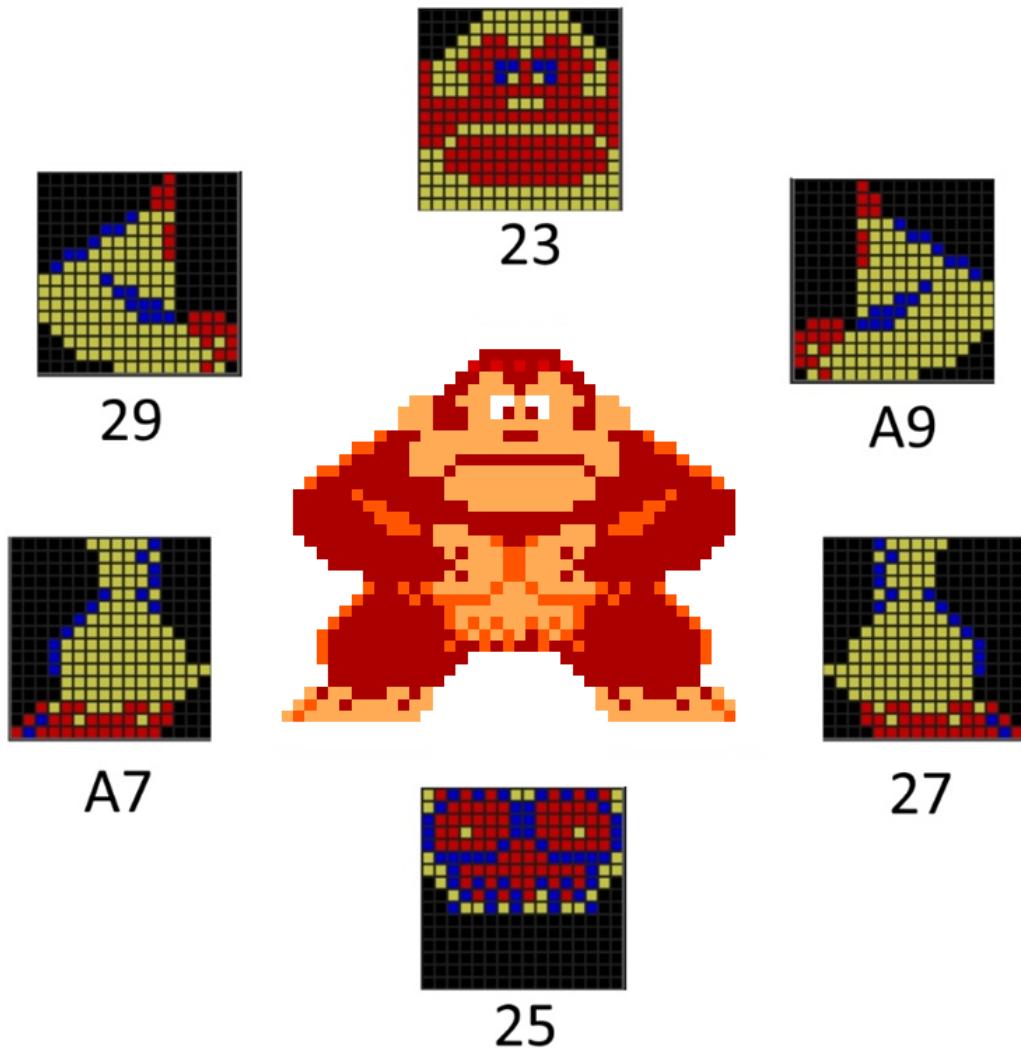
```

Ja, dat werkt.

Dat is ook wel leuk omdat je daarmee, ook op de tweede girder op de vuurballen moet letten. Maar niet altijd.

Nu versie v0.56 opgeslagen.

Nu Donkey Kong sprites vervangen door Bowser. Eerst kijken hoe DK opgebouwd is.



Hadden we al een keer bekijken voor Kwoka Kong.

En dan op basis hiervan de sprite voor Bowser gaan tekenen?

Het hoofd gedaan.

[Nu versie v0.57 opgeslagen.](#)

Het lijf gedaan.

[Nu versie v0.58 opgeslagen.](#)

Dan nu eerst de kleuren aanpassen.

Bowser oranje = #00 / #0F

Bowser roze = #00 / #0A

Bowser groen = #0E / #0B

De kleurcodes van DK zijn #07 (#9D/#9E/#9F) en #08 (#A1/#A2/#A3).

Aanpassen naar:

#9D en #A1: #00 / #0A (roze)

#9E en #A2: #00 / #0F (oranje)

#9F en #A3: #0E / #0B (groen)

[Nu versie v0.59 opgeslagen.](#)

Nu de armen doen.

De armen zitten te hoog: is sprite value #29 en #A9.

Aanpassen in de definitie:

385C: 47 27 08 50

3860: 2F A7 08 50

3864: 3B 25 08 50

3868: 00 70 08 48

386C: 3B 23 07 40

3870: 46 A9 08 44

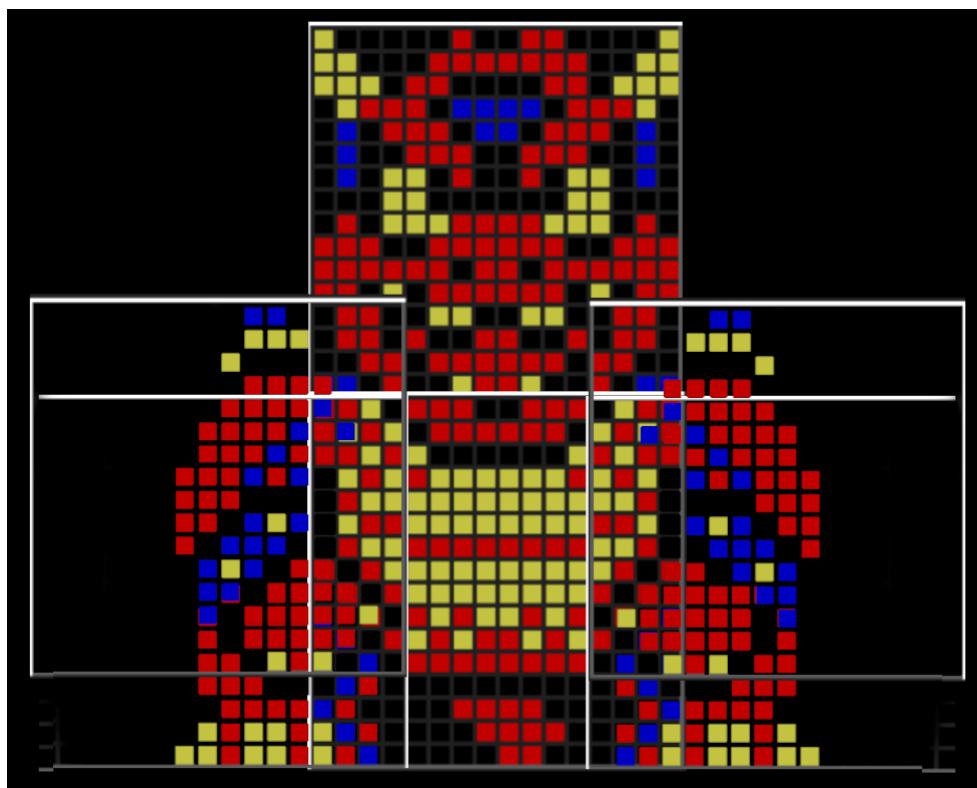
3874: 00 70 08 48

3878: 30 29 08 44

387C: 00 70 08 48

3880: 00 70 0A 48

Ze moeten 8 posities naar beneden. #3873 naar #3C en #387B naar #4C.



De armen en benen moeten 1 naar buiten toe.

```
385C: 47 27 08 50  
3860: 2F A7 08 50  
3864: 3B 25 08 50  
3868: 00 70 08 48  
386C: 3B 23 07 40  
3870: 46 A9 08 44  
3874: 00 70 08 48  
3878: 30 29 08 44  
387C: 00 70 08 48  
3880: 00 70 0A 48
```

#385C - Been rechts = #27 moet naar rechts = #48

#3860 - Been links = #A7 moet naar links = #2E

#3870 – Arm rechts = #A9 moet naar rechts = #47

#3878 – Arm links = #29 moet naar links = #2F



Ziet er nu zo uit. Helemaal goed.

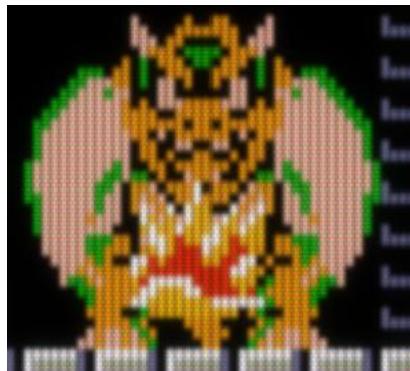
[Nu versie v0.60 opgeslagen.](#)

Nu nog de armen aanpassen wanneer een schild gepakt.

Daarna de animatie van armen omhoog en op borst slaan.

En bowser rechtopstaand naar beneden laten vallen bij einde rivets.

En dan de sprites maken van de werpende bowser.



Het bovenste deel (de schouder) is sprite #2A (rechterkant). Het onderste deel is sprite #2B (rechterkant). Eerst maar eens de #2A door een lege vervangen en de #2B door de eerder gemaakte arm maar dan gespiegeld.



De armen moeten 1 omhoog en 7 naar binnen.

; used to animate kong

395A: 47 27 08 4C

395E: 2F A7 08 4C

3962: 3B 25 08 4C

3966: 00 70 08 44

396A: 3B 23 07 3C

396E: 4B 2A 08 3C

3972: 4B 2B 08 4C

3976: 2B AA 08 3C

397A: 2B AB 08 4C

397E: 00 70 0A 44

#3972 - #2B is rechterkant – moet naar 4 naar links: #47

#397A - #AB is linkerkant – moet 4 naar rechts: #2F

Voor allebei een Y van #4B maken.

Is nu goed.

Sprite voor opgetrokken been maken:



Opgetrokken been (#56) moet 5 naar rechts, en de arm erboven moet naar beneden. En het been rechts moet ook verplaatst worden. En het gezicht moet ook aangepast worden. Eerst zelfde maken en daarna wellicht animatie.

```
; table data  
; kong is beating his chest
```

```
39CF: 47 27 08 50  
39D3: 2D 26 08 50  
39D7: 3B 25 08 50  
39DA: 00 70 08 48  
39DF: 3B 24 07 40  
39E3: 4B 28 08 40  
39E7: 00 70 08 48  
39EA: 30 29 08 44  
39EF: 00 70 08 48  
39F3: 00 70 0A 48
```

Linkervoet is #26.

De linkerarm is #29. in origineel: 2F 29 08 4C

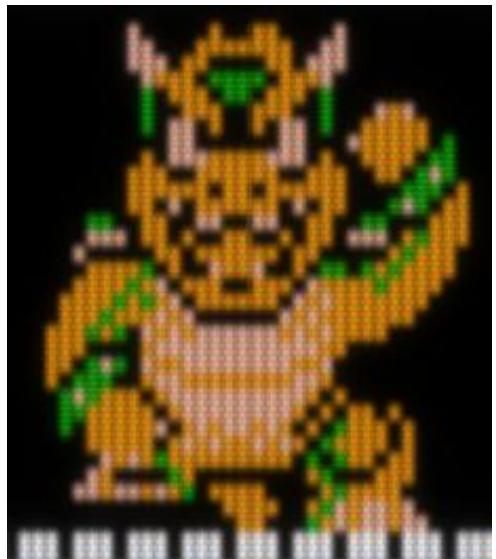
Het hoofd is #24 aanpassen naar #23



De opgeheven arm aanpassen: dat is sprite #28.



De opgeheven arm moet naar beneden en eentje naar links.

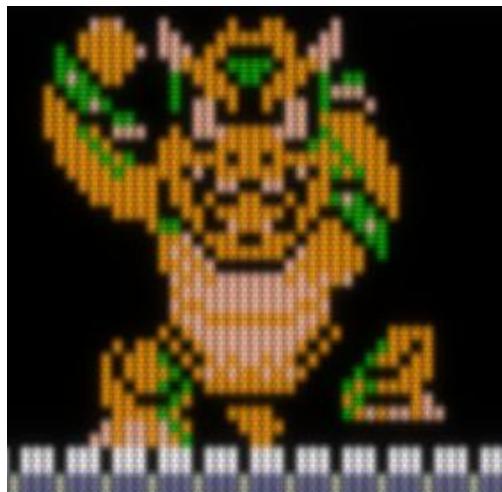


Ziet er zo goed uit.

[Nu versie v0.61 opgeslagen.](#)

Nu nog even het gezicht weer terugzetten en de sprite kopiëren en de mond licht maken.

Nu nog de andere aanpassen.



Dit is waarschijnlijk deze code:

```
; table data for animation of kong #28 bytes (40 decimal)
; used in #0445
; the kong is beating his chest with right leg lifted

39F7: 49 A6 08 50 2F A7 08 50 3B 25 08 50 00 70 08 48
3A07: 3B 24 07 40 46 A9 08 44 00 70 08 48 2B A8 08 40
3A17: 00 70 08 48 00 70 0A 48
```

39F7: 49 A6 08 50 – rechtervoet omhoog

39FB: 2F A7 08 50 – linkervoet omlaag

39FF: 3B 25 08 50 - lichaam

3A03: 00 70 08 48

3A07: 3B 24 07 40

3A0B: 46 A9 08 44 – rechterarm omlaag

3A0F: 00 70 08 48

3A13: 2B A8 08 40 – linkerarm omhoog

3A17: 00 70 08 48

3A1B: 00 70 0A 48

rechterarm moet 8 omlaag en 1 naar rechts.

De rechtervoet moet 5 naar links.

De linkerarm moet 5 omlaag 1 naar rechts.

En kijken of de staart kan wiebelen door ook het lichaam te spiegelen.



Zien er allebei goed uit zo.

[Nu versie v0.62 opgeslagen.](#)

Bowser is smaller dan Donkey Kong. Daardoor zijn de onzichtbare blokken op rivets niet goed. Daar moet de x waarde van de hitbox aangepast worden.

Dat lijkt hier te zitten:

```
; data used above for black space next to kong
117E 3F 0C 08 08          ; sprite code #3F (invisible square), color = #0C (black), size = 8x8 ???
1182 73 50 8D 50          ; 1st is at #73,#50 and the 2nd is at #8D,#50
```

3F 0C 08 72 50 8C 50 ervan gemaakt

Ja dat is nu goed.

Nu nog zo maken dat Bowser bij finish rivets rechtopstaand naar beneden valt.

Het omkeren lijkt hier te zitten:

```
1870 211F3A LD HL,#3A1F      ; start of table data for kong upside down
1873 CD4E00 CALL #004E       ; update kong's sprites
1876 3E03 LD A,#03           ; A := 3
1878 328460 LD (#6084),A    ; play falling sound
187B 218863 LD HL,#6388     ; load HL with end of level counter
187E 34 INC (HL)            ; increase
187F C9 RET                 ; return
```

Omkeren door de CALL op #1873 aan te passen naar 3xNOP.

Ja, dat werkt. Zou ook kunnen door de definitie aan te passen naar rechtopstaande, maar dan met beide handen omhoog...

Dan moet de CALL terug en dan combinatie van de staande bowser met beide armen vervangen door de arm omhoog definitie.

De staande browser definitie zit op #385C. Die eerst eens kopiëren naar #3A1F.

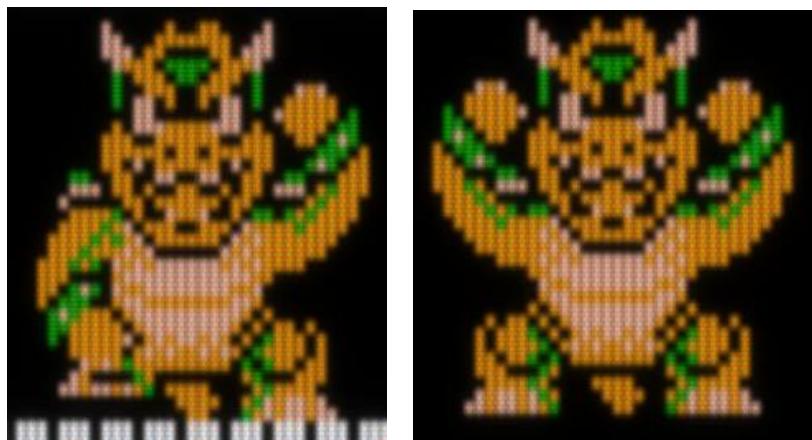
Dat gaat niet goed. Valt nu op de plaats waar Bowser in het barrels board staat. Dat klopt niet. Dan zouden alle X-posities een offset moeten krijgen. Even kijken hoeveel dan...

De X-pos van het lichaam in de barrels board definitie is: #3B, maar in de originele code van de vallende Bowser is dat #7F. Dus #7F - #3B opschuiven: #44.

Aanpassen:

```
47 27 08 50 - 8B  
2F A7 08 50 - 73  
3B 25 08 50 - 7F  
00 70 08 48  
3B 23 07 40 - 7F  
47 28 08 4C - 8B - rechterarm  
00 70 08 48  
2F 2A 08 4C - 73 - linkerarm  
00 70 08 48  
00 70 0A 48
```

Ja, valt nu wel op de juiste plek. Nu nog de armen aanpassen.

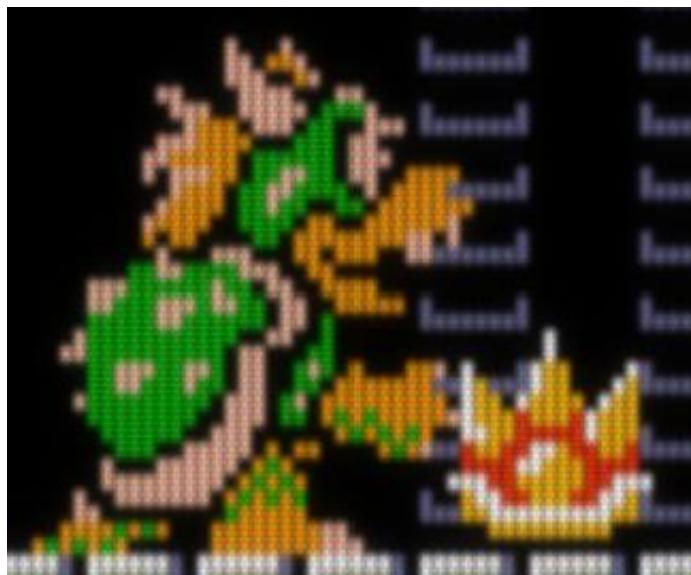


Is zo goed.

[Nu versie v0.63 opgeslagen.](#)

Nu kijken naar de sprites voor Bowser vanaf de zijkant terwijl hij een schild pakt en/of weggoooid.

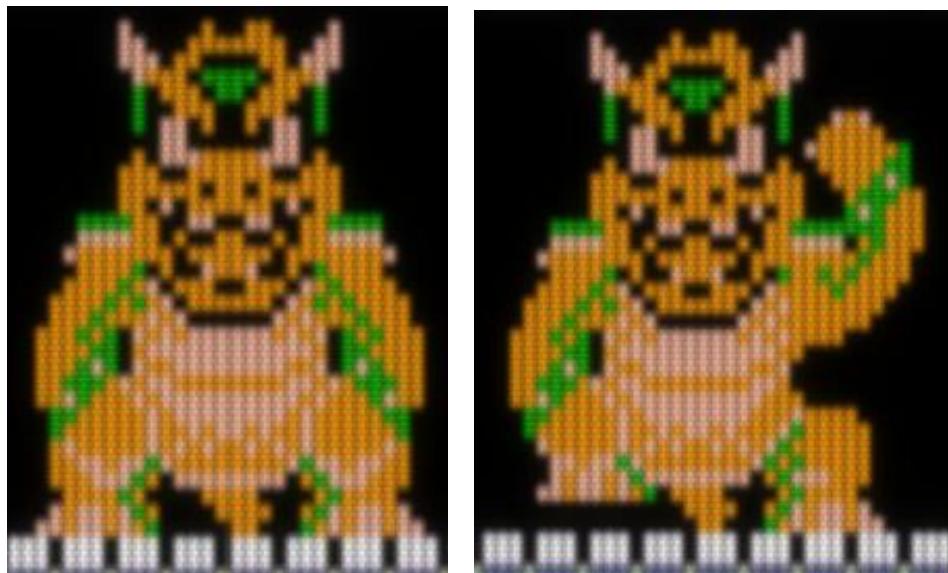
Het ziet er goed uit.



Nu versie v0.64 opgeslagen.

Eerste de Bowser minder handheld cartoony maken.

Ziet er nu zo uit:



Nu versie v0.65 opgeslagen.

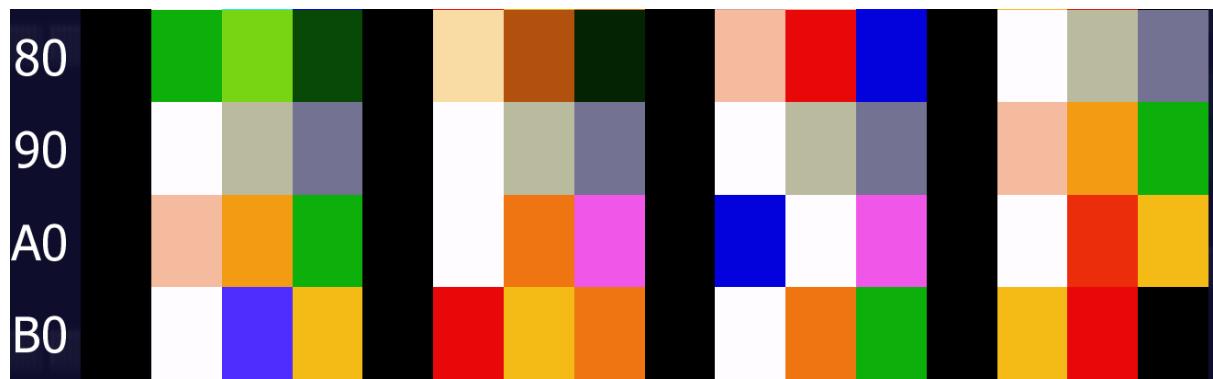
Nu de bowser van de zijkant minder handheld cartoony maken.

Ziet er nu zo uit:



Nu versie v0.66 opgeslagen.

De roze kleur van bowser echt wit maken.



Dat is kleur op #9D en op #A1. Die gelijk maken aan de kleur op #99. Nee dat is veel te wit. Ziet er beter uit met #02 / #01.

Ja, dat ziet er goed uit.

Nu versie v0.67 opgeslagen.

Dan nu de klimmende Bowser maken.

Was best lastig met allerlei tussenversies, maar ziet er nu zo uit:



Nu versie v0.68 opgeslagen.

En dan nu de kop kleuren weer goed zetten.

De scores en world waren in het wit en de titel was in het geel.

Ja, dat is nu goed.

Dan de top titel aanpassen naar BOWSER = #12 #1F #27 #23 #15 #22

High Score zit hier:

```
36A0:  10 30 32 31 3F 94 76 20 1C 11 29 15 22 10 30 33 .<I>...PLAYER.<2  
36B0:  31 3F 80 76 18 19 17 18 10 23 13 1F 22 15 3F 9F >...HIGH.SCORE..  
36C0:  75 13 22 15 14 19 24 10 10 10 10 3F 5E 77 18 1F .CREDIT.....HO
```

Vervangen door: 80 76 10 10 12 1F 27 23 15 22 10 10 3F.

Dat is nu goed.

De kleur van de bonus score popup aanpassen naar wit. Moet ook aangepast naar #01.

```

; draw the bonus score on the screen

1E36 21306A LD HL,#6A30 ; load HL with scoring sprite start
1E39 77 LD (HL),A ; store X position
1E3A 2C INC L ; next location
1E3B 70 LD (HL),B ; store sprite graphic
1E3C 2C INC L ; next
1E3D 3607 LD (HL),#07 ; store color code 7
1E3F 2C INC L ; next
1E40 71 LD (HL),C ; store Y position
1E41 3E05 LD A,#05 ; A := 5 = binary 0101
1E43 F7 RST #30 ; only allow continue on girders and elevators
1E44 218560 LD HL,#6085 ; load HL with bonus sound address
1E47 3603 LD (HL),#03 ; play bonus sound for 3 duration
1E49 C9 RET ; return

```

Hartje en gebroken hartje omzetten naar rood. Is nu kleur code #09.

```

1711 3676 LD (HL),#76 ; set heart sprite
1713 23 INC HL ; next
1714 3609 LD (HL),#09 ; set heart sprite color
1716 23 INC HL ; next

```

Kleur van jumpman #02 maken. En dan de sprite aanpassen naar de rode kleur.

```

1926 3640 LD (HL),#40 ; set heart sprite Y position
1928 2B DEC HL ; decrement HL
1929 3609 LD (HL),#09 ; set heart sprite color
192B 2B DEC HL ; decrement HL

```

Ja, dat is goed.

Dan de help roep van Pauline in beide velden wit maken. Is nu in barrels board grijs en in het rivets board blauw. De help is in de kleuren van de achtergrond. Is nu in het barrels board donkergrijs.

Door de sprite te veranderen van blauw naar rood, wordt het lichtgrijs.

Dat klopt voor het barrelboard, maar op het rivets board is het nu rood geworden. Hoe kan dat? Dat komt doordat de tekst op een regel hoger weergegeven wordt. In rivets board 1 positie naar beneden verplaatsen.

```

04BE 3E10 LD A,#10 ; A := #10 = code for clear space
04C0 212376 LD HL,#7623 ; load HL with video RAM for girl location
04C3 CD1405 CALL #0514 ; clear the "help" the girl yells on the left side
04C6 218375 LD HL,#7583 ; load HL with video RAM right of girl
04C9 CD1405 CALL #0514 ; clear the "help" the girl yells on the right side
04CC CB71 BIT 6,C ; check timer bit 6. zero?
04CE CA0905 JP Z,#0509 ; yes, skip ahead

```

```

04D9 3EDF      LD      A,#DF          ; else A := #DF
04DB 212376    LD      HL,#7623        ; load HL with video RAM for girl location|
04DE CD1405    CALL    #0514          ; draw "help" on the left side

04F1 3EEF      LD      A,#EF          ; A := #EF
04F3 218375    LD      HL,#7583        ; load HL with video RAM for girl location
04F6 CD1405    CALL    #0514          ; draw "help" on the right side

```

Alle vier de posities met 1 ophogen.

Ziet er niet zo heel leuk uit. Is te laag. terugzetten.

Aanpassen door de achtergrondkleur voor alle velden aan te passen. Heeft wellicht impact op de andere schermen. maar valt wel mee verwacht ik. Is midden van regel 4 die aangepast dient te worden.

Ja, is nu goed.

Nu versie v0.69 opgeslagen.

Nu het titelscherm, het how high scherm en het score scherm aanpassen. En daarna de attract screen besturing aanpassen met een sprong naar de pijp.

Nu het titelscherm aanpassen. Daarin de kleuren van Bowser goedzetten, en de titel BOWSER TAKEOVER toevoegen en blokjes van DONKEY KONG aanpassen naar girder blokjes en niet de rivet blokjes.

De DONKEY KONG wordt hier getekend:

```

; draws DONKEY KONG logo to screen

07F7 21083D    LD      HL,#3D08        ; load HL with start of table data

07FA 3EB0      LD      A,#B0          ; A := #B0 = code for girder on screen
07FC 46        LD      B,(HL)        ; get first data. this is used as a loop counter
07FD 23        INC     HL             ; next table entry
07FE 5E        LD      E,(HL)        ; load E with table data
07FF 23        INC     HL             ; next entry
0800 56        LD      D,(HL)        ; load D with table data. DE now has an address

```

Maak van character #B0 eens #D0.

Ja, dat is goed. Welk palette wordt er voor het titelscherm gebruikt?

Dat zit hier:

```

07DF 21867D LD HL,REG_PALETTE_A ; load palette bank
07E2 3600 LD (HL),#00 ; clear palette bank selector
07E4 79 LD A,C ; A := C
07E5 CB07 RLC A ; rotate left. carry bit set?
07E7 3002 JR NC,#07EB ; no, skip next step

07E9 3601 LD (HL),#01 ; set palette bank selector to 1

07EB 23 INC HL ; HL := REG_PALETTE_B = 2nd palette bank
07EC 3600 LD (HL),#00 ; clear the palette bank selector
07EE CB07 RLC A ; rotate left again. carry bit set ?
07F0 3002 JR NC,#07F4 ; no, skip next step

07F2 3601 LD (HL),#01 ; set palette bank selector to 1

07F4 328B63 LD (#638B),A ; store A into ???

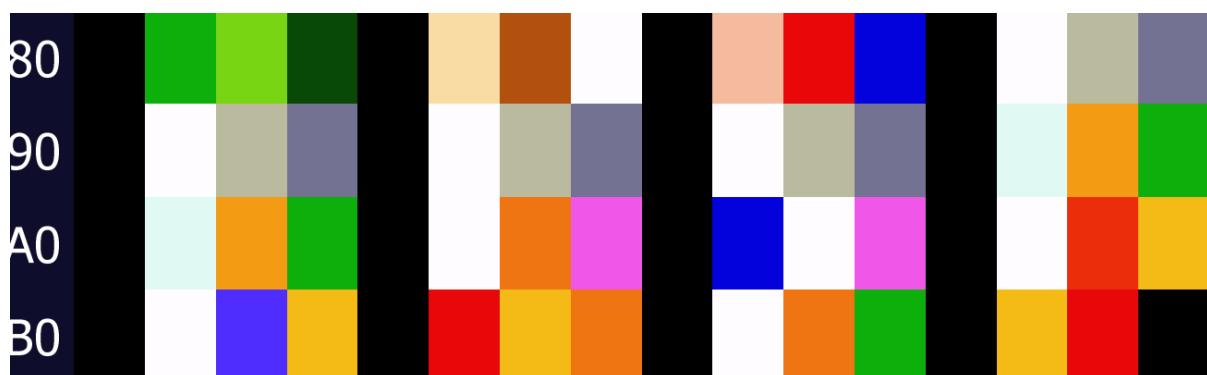
```

Knippert tussen palette 0/0 en 0/1.

Maar standaard dus palette 0/0. Dat is het eerste palette neem ik aan. Proberen door daar de kleuren van bowser aan te passen. Dat zijn kleurcode #09 en #0A. Dus kopiëren codes van #9C t/m #A3 naar #1C t/m #23.

Nee, dat klopt niet. Het lijkt eerder te knipperen tussen het barrels en het rivets scherm. Dus het rivets scherm aanpassen. Dat is het vierde palette. Dus kopiëren codes van #9C t/m #A3 naar #DC t/m #E3.

Ja, dat klopt wel. Nu nog de kleuren van het Donkey Kong logo hetzelfde maken zodat dat niet knippert.



Dat zijn vier blokken. Eén voor één aanpassen.

2J voor de aanpassingen:

```

0000000C0 OF 06 0E 01 0F 00 01 00 0F 00 01 0F 0F 0F 0E 00
0000000D0 OF 0F 0E 00 0F 0F 0E 00 0F 0F 0E 00 0F 02 00 0E
0000000E0 OF 02 00 0E 0F 00 01 01 0F 0F 00 01 0F 0F 01 00
0000000F0 OF 00 0E 0F 0F 0F 01 00 01 0F 02 00 0F 0F 00 01 0F

```

2K voor de aanpassingen:

```
0000000C0  0F 0C 00 0F 0F 0B 0F 00 0F 0A 0F 0D 0F 0C 00 0B  
0000000D0  0F 0C 00 0B 0F 0C 00 0B 0F 0C 00 0B 0F 01 0F 0B  
0000000E0  0F 01 0F 0B 0F 00 03 05 0F 0D 00 05 0F 0C 03 0A  
0000000F0  0F 00 00 0C 0F 0F 0B 03 0F 0E 0A 0C 0F 0B 0F 0F
```

De eerste twee (kleurcodes #03 en #04) zijn aangepast. Nu met het scherm overzicht de regels aanpassen. Zodat logo naar die verwijst en de rest niet. Allemaal #03 maken.

Kolommen #08, #09, #0A, #0B en #0C zijn DONKEY.

Kolommen #10, #11, #12, #13 en #14 zijn KONG.

In de andere schermen nog te spelen met de #03 en #04.

De kleur van de titel BOWSER bovenin klopt nog niet. Dat is de #0B kleur. Die ook kopiëren. Dus kopiëren #AC t/m #AF naar #EC t/m EF.

De kleuren zijn nu allemaal goed. Nu nog het knipperen langzamer maken.

```
07D3 3E60      LD      A,#60          ; else A := #60  
07D5 328A63    LD      (#638A),A     ; store into kong screen flash counter  
07D8 0E5F      LD      C,#5F          ; C := #5F
```

Twee keer zo groot, maar helpt niets. gewoon niet laten knipperen.

```
07DF 21867D    LD      HL,REG_PALETTE_A      ; load pallete bank  
07E2 3600      LD      (HL),#00        ; clear palette bank selector  
07E4 79        LD      A,C          ; A := C  
07E5 CB07      RLC     A          ; rotate left. carry bit set?  
07E7 3002      JR      NC,#07EB      ; no, skip next step  
  
07E9 3601      LD      (HL),#01      ; set pallete bank selector to 1  
  
07EB 23        INC     HL          ; HL := REG_PALETTE_B = 2nd pallete bank  
07EC 3600      LD      (HL),#00      ; clear the pallete bank selector  
07EE CB07      RLC     A          ; rotate left again. carry bit set ?  
07F0 3002      JR      NC,#07F4      ; no, skip next step  
  
07F2 3601      LD      (HL),#01      ; set pallete bank selector to 1  
  
07F4 328B63    LD      (#638B),A     ; store A into ???
```

In beide gevallen de PALETTE_A op #01 zetten.

Ja, dat werkt nu.

Kleuren zijn nu goed.

Nu versie v0.70 opgeslagen.

En dan nu de titel DUEL aanpassen naar BOWSER TAKEOVER.

De huidige titel staat in #3F18:

8E 76 10 14 10 25 10 15 10 1C 3F

Dat is het slot van de Nintendo of America op #3689.

De tekst van one or two players button wordt niet meer gebruikt. Die plek gebruiken. Dat is het blok van #36FD t/m #3714.

Daar dan BOWSER TAKEOVER aan toevoegen:

36FD EE 76 12 1F 27 23 15 22 10 24 11 1B 15 1F 26 15 22 3F

3689 FD 36

Dan het jaartal nog aanpassen. Dat maar alvast op 2026 zetten. Gaat niet meer in 2025 geredeleased worden.

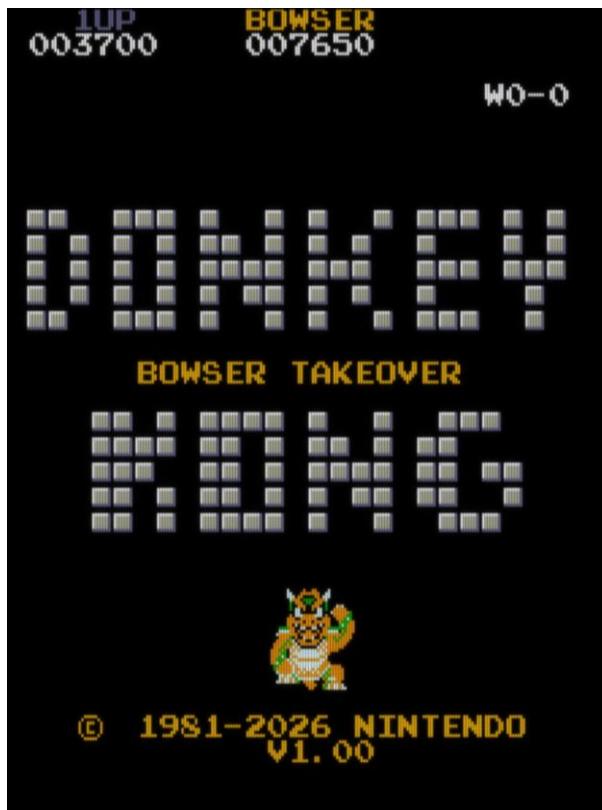
In #3F00 de copyright notice staat de copyright notice:

© 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

Aanpassen naar:

5C 77 49 4A 10 01 09 08 01 2C 02 00 02 06 10 1E 19 1E 24 15 1E 14 1F 3F



Het scherm ziet er nu zo uit. Helemaal goed.

Nu versie v0.71 opgeslagen.

Dan nu het how high scherm aanpassen. Statisch maken dat er geen stapel gebouwd wordt. Wel tekenen van bowser en twee koopa's en de tekst NOW YOU DEAL WITH ME ! en BWA HA HA !!!.

HOW HIGH CAN YOU GET is net zo lang als NOW YOU DEAL WITH ME.

Eerst een heel deel eruit halen: wissen van #0BFA t/m #0C81. Dan wordt de stapel DK's niet meer getekend.

Ja, dat gaat goed.

36CC: 5E 77 1E 1F 27 10 29 1F 25 10 14 15 11 1C 10 27 19 24 18 10 1D 15 10 38 10 3F

De How High tekst staat op #36CC.

Dat is nu goed. Misschien nog omhoog schuiven. Eerst de andere tekst maken.

0C72: 5C 77 12 27 11 10 18 11 10 18 11 10 38 38 38 3F

Tekst slot 8 is nog vrij. Die zit op #365B.

365B 72 0C

En dan aan de code van How high scherm toevoegen en dan als laatste een jump naar #0C82.

```

0BFA 110803 LD DE,#0308 ; load task data for text #8
0BFD CD9F30 CALL #309F ; insert task to draw text
;
0C00 C3820C JP #0C82 ; jump to remainder of how high screen

```

Dat gaat goed. Nu nog de teksten verschuiven en bowser ertussen tekenen.

```

0C00 215C38 LD HL,#385C ; start of table data of front bowser
0C03 CD4E00 CALL #004E ; draw bowser
;
0C06 210869 LD HL,#6908 ; load HL with bowser sprites X position
0C09 0E44 LD C,#44 ; set offset
0C0B FF RST #38 ; move bowser right
;
0C0C 210B69 LD HL,#690B ; load HL with bowser sprites Y position
0C0F 0E60 LD C,#60 ; set offset
0C11 FF RST #38 ; move bowser down
;
0C12 C3820C JP #0C82 ; jump to remainder of how high screen

```

Dan de teksten verplaatsen.

NOW YOU DEAL WITH ME ! moet een stuk omhoog.

36CC: 50 77 1E 1F 27 10 29 1F 25 10 14 15 11 1C 10 27 19 24 18 10 1D 15 10 38 10 3F

BWA HA HA !!! moet een stuk omhoog en naar rechts.

0C72: D9 76 12 27 11 10 18 11 10 18 11 10 38 38 38 3F

En dan de palette van het scherm aanpassen. Dat zit hier:

```

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

```

Allebei #01 maken. Ja, dat is nu goed.

Het scherm ziet er nu zo uit:



Nu versie v0.72 opgeslagen.

Nu nog het begin scherm met ONLY 1 PLAYER scherm aanpassen.

Eerst het palette op dat van rivets zetten. Nee, dat is niet mooi, dus naar die van barrels zetten. Dat zit hier:

```
08CE AF      XOR      A                ; A := 0
08CF 21867D    LD      HL,REG_PALETTE_A      ; load HL with palette bank
08D2 77      LD      (HL),A                ; clear palette bank selector
08D3 2C      INC     L                ; next palette bank
08D4 77      LD      (HL),A                ; clear palette bank selector
```

Aanpassen:

```
08CE C3150C      JP      #0C15      ; jump to additional code
08D1 0000      NOP, NOP
```

en:

```
0C15 3E00      LD      A,#00      ; A:=00
0C17 21867D    LD      HL,REG_PALETTE_A
0C1A 77      LD      (HL),A                ; set palette A
0C1B 3E01      LD      A,#01      ; A:=01
0C1D C3D308    JP      #08D3      ; jump back
```

Ja, dat gaat goed.

Maar voordat je credits erin doet, komt er ook nog een ander scherm langs:



Ook daar de kleur naar het barrels board zetten. En de 2 2 kunnen weg want er is geen 2 player mode meer.

De selectie van het palette zit hier:

```
0779 21867D LD HL,REG_PALETTE_A
077C 3600 LD (HL),#00 ; clear palette bank selector
077E 23 INC HL
077F 3600 LD (HL),#00 ; clear palette bank selector
0781 111B03 LD DE,#031B ; load task data for text "INSERT COIN"
0784 CD9F30 CALL #309F ; insert task to draw text
0787 1C INC E ; load task data for text "PLAYER COIN"
0788 CD9F30 CALL #309F ; insert task to draw text
```

Aanpassen:

```
077F 3601 LD (HL),#01 ; set palette bank selector
```

Dan nog de 2 2 weghalen. Dat zit hier:

```
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
```

Aanpassen:

```
07B0 00 NOP
```

Ja, dat is nu goed.

Nu versie v0.73 opgeslagen.

Nu nog het high score scherm aanpassen.

Palette voor het High Score entry scherm zit hier:

```
148D A7 AND A ; == 0 ?
148E C2DC14 JP NZ,#14DC ; no, skip ahead

1491 32867D LD (REG_PALETTE_A),A ; set palette bank selector|
1494 32877D LD (REG_PALETTE_B),A ; set palette bank selector
1497 3601 LD (HL),#01 ; set timer to 1
1499 213060 LD HL,HSCursorDelay ; load HL with HSCursorDelay
```

Aanpassen:

```
1494 C3200C JP #0C20 ; jump to additional code
```

en:

```
0C20 3E01 LD A,#01 ; A := #01
0C22 32877D LD (REG_PALETTE_B),A ; set palette bank selector
0C25 C39714 JP #1497 ; jump back
```

Ja, dat gaat goed.

En dan uiteindelijk nog tekst maken “BOWSER’S” en “EMBARRASSMENT LIST” met twee Bowser icon hoofden erbij.

Dat eigenlijk in twee schermen (maar wellicht dezelfde code die dat doet): het scherm zonder dat er credits ingevoerd zijn (met PUSH, ONLY 1 PLAYER BUTTON) en het scherm met credits ingevoerd (met INSERT COIN, PLAYER COIN, 1 1). Al die teksten weg en veranderen in de twee nieuwe teksten.

Gebruik de INSERT COIN tekst voor BOWSER’S:

37D2: A7 76 12 1F 27 23 15 22 3A 23 3F

Gebruik de ONLY 1 PLAYER BUTTON tekst voor EMBARRASSMENT LIST:

36E6: 29 77 15 1D 12 11 22 22 15 23 23 1D 15 1E 24 10 1C 19 23 24 3F

Op het scherm met de credits ingevuld, moet de PUSH tekst gewijzigd worden in de BOWSER’S

Dat zit hier:

```
08C1 110C03 LD DE,#030C ; load DE with task code to display "PUSH" onscreen  
08C4 CD9F30 CALL #309F ; insert task
```

Aanpassen naar task #9:

```
08C1 111B03 LD DE,#031B ; load DE with task code to display "BOWSER'S"
```

Ja, dat gaat goed. In ieder geval nog even goed positioneren.

EMBARRASSMENT LIST moet 3 naar beneden: #7729 -> #772C en 1 naar rechts: #772C -> #770C.
BOWSER’S moet 3 naar beneden: #76A7 -> #76AA en 2 naar rechts: #766A.

Dat is nu in dat scherm goed.

Nu in het scherm zonder credits ook nog goed zetten. Daar zaten twee teksten èn de kolommen met benodigde credits. Al deze code kan weg en vervangen worden door code die de twee teksten neerzet.

Dat zit in de code van #0779 t/m #07C2. Het stuk van #079B t/m #07BB tekent de benodigde credits voor 1 en twee player games. Daarin de tweede rij al weggehaald door #07B0 te veranderen in een NOP. Dit hele stuk NOP maken en kijken of de tekst dan weg is, maar de rest gewoon nog werkt.

Dan aanpassen #0787 t/m #078A NOP’s maken. Kan niet zo, want dat veronderstelt dat de twee teksten achter elkaar staan (door de INC E).

Maar in het vrijgemaakte stuk de EMBARRASSMENT LIST tekenen:

```
079B 110903      LD   DE,#0309      ; load task data for text "EMBARRESSMENT LIST"
079E CD9F30      CALL #309F      ; insert task to draw text
```

Ja, dat gaat nu goed.

Nu versie v0.74 opgeslagen.

Nu nog kijken of er twee Bowser logo's aan deze schermen toegevoegd kunnen worden.

Eerst maken met twee vierkanten en dan later Bowser logo daarin tekenen.

Sprite #69 gebruikt hiervoor.

Stuk code voor het weergeven.

Sprite slots #6980 en #6984 hiervoor gebruiken.

Aanpassen:

```
07A1 CDAF07      CALL #07AF      ; call additional code
07A4 C3BC07      JP  #07BC      ; jump forward
;
07A7 40 69 02 48 C0 69 02 48

07AF 118069      LD   DE,#6980      ; destination #6980
07B2 21A707      LD   HL,#07A7      ; source data at table
07B5 010800      LD   BC,#0008      ; byte counter at #08
07B8 EDB0        LDIR          ; copy
07BA C9          RET           ; return
```

Bij het scherm zonder credits is het zo goed. Nu nog toevoegen aan het scherm met credits.

```
08C1 110C03      LD   DE,#030C      ; load DE with task code to display "PUSH" onscreen
08C4 CD9F30      CALL #309F      ; insert task
08C7 210A60      LD   HL,GameMode2    ; load A with game mode2
```

Aanpassen:

```
08C4 C3280C      JP  #0C28      ; jump to additional code
```

en:

```
0C28 CD9F30      CALL #309F      ; insert task
0C2B CDAF07      CALL #07AF      ; draw two bowser logos
0C2E C3C708      JP  #08C7      ; jump back
```

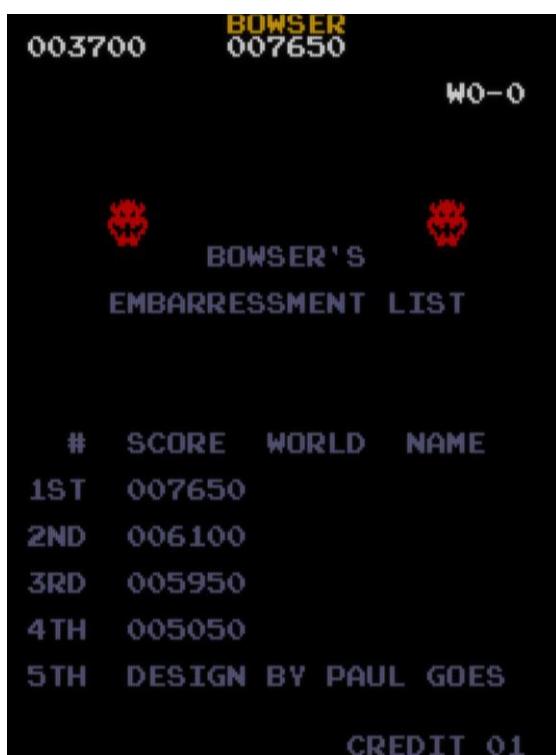
Ja, dat gaat nu goed.

Nu alleen nog de sprite aanpassen.

Voorbeeld:



Het scherm ziet er nu zo uit:



Nu versie v0.75 opgeslagen.

En dan nu de attract mode aansturing zodanig maken dat springen naar pijp en munt pakken en dan springen naar ster en onsterfelijk naar de volgende girder.

Deze sequence is het geworden (startend op #21D1):

80	- jump up
FE 01	- run right
60 04	- climb up (take ladder)
50 02	- run left
55 01	- run right
20 81	- jump right (to pipe)

```
20 01 - run right (get coin)
10 02 - run left
20 82 - jump left (to girder)
20 82 - jump left (grab star)
78 02 - run left
60 04 - climb up (take ladder)
18 01 - run right
60 04 - climb up (take ladder)
FF 01 - run right (fall off girder)
```

Erg leuk geworden en gaat goed.

Nu versie v0.76 opgeslagen.

Nu eerst het elevator board aanpassen. Basis datastructuur maken en dan de structuur en kleuren aanpassen. En kijken één elevator en de andere uit het zicht plaatsen. Dan onderaan de elevator lava en bovenaan scherpe punten. Dan de hamers laten gooien vanuit Bowser, daar sprite voor maken en sprite draaiende animatie geven. En dan de rest van het veld maken en de coins plaatsen. En dan de laatste trap neerzetten als alle coins gepakt en dan de vlag plaatsen en zorgen ook hier einde level als naar vlag gesprongen.

De basisstructuur laten beginnen op: #3BE0. Dan ook de verwijzingen ernaar aanpassen. Zat op #3BE5. Verwijzingen zijn:

```
0CFA 11E53B LD DE,#3BE5 ; load DE with start of table data for the elevators
2468 21E53B LD HL,#3BE5 ; load HL with start of table data for elevators
```

```
3BE0: 00 63 18 63 58 ; kong's ladder (right)
3BE5: 00 53 18 53 58 ; kong's ladder (left)
3BEA: 00 93 38 93 58 ; ladder leading up to girl
3BEF: 00 23 88 23 C0 ; long ladder on left side
3BF4: 00 1B C0 1B E8 ; bottom left ladder
3BF9: 02 97 38 68 38 ; girder girl is on
3BFE: 02 EF 58 10 58 ; kong's girder
3C03: 02 27 88 10 88 ; girder that holds the umbrella
3C08: 02 27 E8 10 E8 ; girder where mario starts
3C0D: 02 EF F8 10 F8 ; long bottom girder (mario dies if he gets that low)
3C12: AA ; end code
```

Testen met 03 als eerste board en starten vanaf level 2.

```
; table data use in code above - gets copied to Unk6041 to Unk6041+7
095E 01 65 3A 01 00 00 00 ; #3A65 is start of table data for screens/levels
```

Ja, deze basis ziet er goed uit.

De elevator die omlaag gaat (de rechter) opschuiven naar buiten beeld.

```
; 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
```

Aanpassen van de #77 naar #00. Werkt dat? Ja, in eerste instantie wel. Als de elevator echter opnieuw spawned, dan wordt deze toch nog op 77 geplaatst. Dat zit hier:

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

Daar ook aanpassen naar #00. Werkt nu goed.

De twee elevator kabels niet meer tekenen.

Dat is de code van #0D27 t/m #0D42 die vrijgegeven kan worden. De aanroep op #0CF2 moet dan wel NOP gemaakt worden.

```
0CF2 CD270D CALL #0D27 ; draw elevator cables
```

Nu versie v0.77 opgeslagen.

Nu nog het design aan de linkerkant aanpassen. Bredere girders en dan de onderste weg en die vervangen door een extra trap. Jumpman op de bodem laten starten (en daar niet meer dood laten gaan). Dan de elevators ook opschuiven en de koopa op de structuur links laten starten. En dan de elevator mounts aanpassen naar lava en spikes.

De girders aanpassen:

```
3BE0: 00 63 18 63 58 ; kong's ladder (right)
3BE5: 00 53 18 53 58 ; kong's ladder (left)
3BEA: 00 93 38 93 58 ; ladder leading up to girl
3BEF: 00 23 88 23 C0 ; long ladder on left side
3BF4: 00 1B C0 1B F8 ; bottom left ladder
3BF9: 02 97 38 68 38 ; girder girl is on
3BFE: 02 EF 58 10 58 ; kong's girder
3C03: 02 2F 88 10 88 ; girder that holds the umbrella
3C08: 00 2B C0 2B F8 ; additional bottom right ladder
3C0D: 02 2F C0 10 C0 ; additional bottom left girder
3C12: 02 EF F8 10 F8 ; long bottom girder (mario dies if he gets that low)
```

```
3C17: AA          ; end code
```

De linker structuur ziet er nu goed uit. Nu jumpman op de onderste girder laten beginnen. Dat zit hier:

```
123D 3A2762 LD A,(#6227)      ; load a with screen number
1240 fe03 CP #03             ; is this the elevators?
1242 0116E0 LD BC,#e016       ; B := #E0, C := #16. used for X,Y coordinates
1245 cA4B12 JP Z,#124B        ; if elevators skip next step

1248 013FF0 LD BC,#F03F       ; else load alternate coordinates for elevators

124B DD210062 LD IX,#6200      ; set IX to mario sprite array
```

Dus die hele check met alternatief voor elevators hoeft niet. Kan gewoon hetzelfde zijn als de andere boards. Nu zodanig aanpassen dat in beide gevallen dezelfde coordinaten geladen worden:

```
1248 0116E0 LD BC,#E016
```

Nee dat is niet goed. Nu de coordinaten voor de andere boards aangepast. De comments kloppen niet. Terugzetten en zo aanpassen:

```
1242 013FF0 LD BC,#F03F
```

Nee, start nu te ver naar rechts, dus moet naar links toe.

```
1242 012FF0 LD BC,#F02F
```

En dan ervoor zorgen dat jumpman niet dood gaat op de onderste girder. Dat zit hier:

```
; elevators only

26FD 3A0562 LD A,(#6205)      ; load A with mario's Y position
2700 FEF0 CP #F0              ; is mario too low ?
2702 D27F27 JP NC,#277F        ; yes, then mario dead
```

Dus hier de conditionele jump NOP's maken.

Dan de ene koopa bovenaan de linker structuur laten beginnen.

```

; 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

```

Van de eerste de X aanpassen naar #20. Ja, dat gaat goed.

Dan de elevator 8 naar rechts opschuiven.

```

; 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

```

Dus in de definitie van de eerste, de #37 aanpassen naar #3F.

En de respawn zit hier:

```

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

```

Dus daar ook de #37 aanpassen naar #3F. Dat is nu goed. Maar jumpman wordt niet goed aangepast als hij te ver naar rechts op de elevator loopt. En links kan hij ervanaf lopen en wordt toch nog naar boven geschoven. Dat zit hier:

```

; arrive here when mario riding on either elevator

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

```

Dus #37 is #3F geworden: +#08. #2C aanpassen naar #34 en #43 aanpassen naar #4B. Ja, dat werkt. Maar misschien beter om nog 8 verder te schuiven. Dus #37 -> #47 en #2C wordt #3C en #43 wordt #53. Even kijken hoe dat gaat. Ja, dat gaat nu goed.

Maar dan waarschijnlijk ook de tweede elevator nog actief v.w.b. jumpman aanpassen. Dus dat ook buiten beeld plaatsen: #6C aanpassen naar #02 en #83 aanpassen naar #04.

Nu versie v0.78 opgeslagen.

En dan nu de elevator mountpoints aanpassen naar lava en scherpe punten en dan overeenkomstig met de positionering van de lava de onderste girder splitsen in twee girders. Want anders loopt de koopa door de lava heen.

De elevator mountpoints worden hier geplaatst:

```

1115 117069 LD DE,#6970 ; destination #6970 (sprites used at top and bottom of elevators)
1118 212111 LD HL,#1121 ; source data at table below
111B 011000 LD BC,#0010 ; byte counter at #10
111E EDB0 LDIR ; copy
1120 C9 RET ; return

; 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

```

Dus de datatabel aanpassen naar sprites van lava (#19) en spikes.

```

1121 3F 45 03 66 ; spike links
1125 3F 19 0F F7 ; lava links
1129 4F 45 03 66 ; spike rechts
112D 4F 19 0F F7 ; lava rechts

```

Dan nu de onderste girder splitsen zodat deze niet meer zichtbaar is door de lava heen en de koopa er ook niet doorheen kan lopen.

```
3C12: 02 30 F8 10 F8 ; bottom girder left  
3C17: 02 EF F8 58 F8 ; bottom girder right  
3C1C: AA             ; end code
```

Ja dat is nu goed. Nu de elevator zelf qua sprite aanpassen.

[Nu versie v0.79 opgeslagen.](#)

En dan nu eerst het palette aanpassen en dan omzetten naar de kleuren van het barrels board. En dan de kleuren van de achtergrond gaan aanpassen.

Het palette #00 gebruiken. Het palette voor het elevators board wordt hier gezet:

```
0C9F 21867D LD    HL,REG_PALETTE_A      ; load HL with palette bank selector  
0CA2 3600   LD    (HL),#00      ; clear palette bank selector  
0CA4 23     INC   HL          ; next bank  
0CA5 3601   LD    (HL),#01      ; set palette bank selector  
0CA7 3A2762 LD    A,(#6227)    ; load A with screen number  
0CAA 3D     DEC   A          ; decrease by 1  
0CAB CAD40C JP    Z,#0CD4      ; if zero jump to #0Cd4 - we were on girders - continue on #0CC6  
  
0CAE 3D     DEC   A          ; if not decrease a again  
0CAF CADF0C JP    Z,#0CDF      ; if zero jump to #0CDF - we were on pie - continue on #0CC6  
  
0CB2 3D     DEC   A          ; if not decrease a again  
0CB3 CAF20C JP    Z,#0CF2      ; if zero jump to #0CF2 - we were on elevators - continue on #0CC6  
  
; else we are on rivets
```

Voor de code die hierop volgt wordt voor de rivets een override gedaan. Maar voor de andere boards is het allemaal hetzelfde palet.

Voor elevators wordt er naar deze code gesprongen:

```
; elevators from #0CB3  
  
0CF2 CD270D CALL  #0D27      ; draw elevator cables  
0CF5 3E0A   LD    A,#0A      ; A := #A  
0CF7 328960 LD    (#6089),A  ; set music for elevators  
0CFA 11E53B LD    DE,#3BE5  ; load DE with start of table data for the elevators  
0CFD C3C60C JP    #0CC6      ; jump back
```

Daar hebben we het tekenen van de kabels in NOP's veranderd. Daar kan dan nu de override van het palette ingevoerd worden. Aanpassen:

```
0CF2 3600   LD    (HL),#00      ; override palette  
0CF4 00     NOP
```

Ja, dat is nu goed.

Dan nu de kleuren van het barrelboard palette (#10) kopiëren naar palette #00.

Dat is ook goed. Dan nu de kleuren gaan aanpassen:



Wit moet kleur#06/#01 worden.

Lichtgrijs moet kleur #0D/#02 worden.

Donkergrijs moet kleur #0F/#0B worden.

Dat is goed. Probleem is wel dat de ladders niet goed te zien zijn. Wellicht is het mogelijk om een patroon te maken dat de zwart van de achtergrond gebruikt en de zwart (was donkergrijs) gebruiken voor een mooie kleur voor de ladders.

Wat gebeurt er wanneer je de girders de conveyor belt code geeft? Dan zou je die graphic kunnen gebruiken voor een nieuw design. Dat is girder code #03.

Dat gaat goed. Maar voor die girders worden de beide uiteinden niet getekend. Dat er dan uithalen, zodat deze wel gewoon getekend worden. Die uiteinden worden wel getekend maar daarvoor wordt een leeg character gebruikt. Standaard is #B1, maar aan de uiteinden wordt #B2 en #B3 gebruikt. Maar die zijn beiden leeg. oplossen door deze beiden ook naar #B1 te veranderen.

0EF3 3EB3 LD A, #B3

0F14 3EB2 LD A, #B2

Ja, dat gaat goed.

Nu kijken of je het element kunt tekenen op kleinere schaal en hoe dat overkomt. Nee, dat komt niet echt over. Misschien tekenen met dubbele breedte. Dan dus de code aanpassen dat om en om de karakters gekozen worden. Maar dat is wellicht wel lastig.

[Nu versie v0.80 opgeslagen.](#)

Kijken of de conveyors op de volgende manier getekend kunnen worden:



De twee sprites zijn dan:



De linker wordt #B1 en de rechter wordt #B2.

De code voor het tekenen van een conveyor aanpassen:

```
0EFO 2AAB63      LD   HL,(#63AB)    ; load HL with VRAM screen address
0EF3 012000      LD   BC,#0020    ; load BC with offset
;
0EF6 3EB2        LD   A,#B2      ; load A with right character #B2 - BBBB
0EF8 77          LD   (HL),A    ; draw character on screen
0EF9 09          ADD  HL,BC    ; add offset to HL, next screen address
;
0EFA 3AB163      LD   A,(#63B1)   ; load A with #63B1
0EFD D608        SUB  #08      ; subtract #08, done?
0EFF 3813        JR   C,#AAAA    ; yes, skip forward, jump to AAAA
;
0F01 32B163      LD   (#63B1),A  ; store updated value to #63B1
;
0F04 3EB1        LD   A,#B1      ; load A with left character #B1
0F06 77          LD   (HL),A    ; draw character on screen
0F07 09          ADD  HL,BC    ; add offset to HL, next screen address
;
0F08 3AB163      LD   A,(#63B1)   ; load A with #63B1
0F0B D608        SUB  #08      ; subtract #08, done?
0F0D 3805        JR   C,#AAAA    ; yes, skip forward, jump to AAAA
;
0F0F 32B163      LD   (#63B1),A  ; store updated value to #63B1
0F12 18E2        JR   #BBBB    ; loop back, jump to BBBB
;
0F14 13          INC  DE      ; next table entry - AAAA
0F15 C3A70D      JP   #0DA7    ; loop back and process next table entry
```

Ja, dat gaat goed. Alleen doen voor de onderste girder. Dat ziet er wel heel gaaf uit.

[Nu versie v0.81 opgeslagen.](#)

Dan nu de kleur van de elevators aanpassen. De kleur #01 nemen om te kijken. Die wordt nu nog nergens voor gebruikt.

Eerste kleur = #00 / #00

Tweede kleur = #02 / #0F

Derde kleur = #05 / #0B

En dan de elevator sprites ook aanpassen bij aanmaken:

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

```
3E60: 44 03 08 04
```

De #44 is de sprite value en de #03 is de sprite color.

Gaat niet helemaal goed. De eerste kleur wordt ook voor de wit in de kop gebruikt en ook voor de bonus score popups. Ervoor zorgen dat dat weer wit wordt. Dus omhusselen van de drie kleuren en de elevator sprite daarop aanpassen. Dat lijkt dan bijna op de oorspronkelijke kleur. Die terugzetten en dan de sprite aanpassen.

Dan de kleur van de munten aanpassen naar kleurcode #0B:

```
; 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
```

En de trappen iets minder donker maken.

De derde kleur aanpassen naar #0F / #0A.

[Nu versie v0.82 opgeslagen.](#)

Ziet er nu zo uit:



Dan nu eerst het hamer gooien gaan aanpassen. Want als dat goed gaat, dan kan op basis daarvan de rechter structuur opgebouwd gaan worden.

Waar wordt de spring geïnitieerd en hoe wordt daarbij de X-positie gezet?

```

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+#0D),#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

```

Wordt dus nu random waarde met een offset. Deze offset aanpassen zodat de spring vanaf Bowser vertrekt.

Aanpassen:

```

2EBD E607      AND     #07        ; mask bits, result is between #00 and #07
2EBF C638      ADD     A,#38      ; add offset

```

En iets hoger (vanuit z'n handen) laten starten.

Aanpassen:

```
2EB2 DD360550 LD *IX+#05),#48 ; set Y-position
```

Dat gaat goed.

Dan ervoor zorgen dat hij niet meer bounced, maar na de eerste boog verder gaat vallen.

Dat lijkt hier te zitten:

```
2E51 DD7E03 LD A,(IX+#03) ; load A with X position
2E54 FEB7 CP #B7 ; < #B7 ?
2E56 DA6C2E JP C,#2E6C ; no, skip ahead
```

Aanpassen:

```
2E54 FE60 CP #60 ; < #60 ?
```

Ja, dat gaat goed.

Maar het boogje is wel erg klein. Kan dit groter gemaakt worden, bijvoorbeeld door elke stap in de boog voor twee X-posities te doen? Dat zit hier:

```
2E31 DD3403 INC (IX+#03) ; no, increase X position
2E34 DD3403 INC (IX+#03) ; increase X position again
2E37 DD6E0E LD L,(IX+#0E)
2E3A DD660F LD H,(IX+#0F) ; load HL with table address for bouncer offsets of Y positions for each pixel across
2E3D 7E LD A,(HL) ; load table data
2E3E 4F LD C,A ; copy to C
2E3F FE7F CP #7F ; == #7F ? (end code ?)
2E41 CA9C2E JP Z,#2E9C ; yes, jump ahead, reset HL to #39AA, play bouncer sound, and continue at #2E4B|
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
```

Wellicht zou je de IX+#03 niet twee keer, maar wellicht drie keer kunnen verhogen. Maar dan gaat de hamer in het eerste stuk iets sneller.

Aanpassen:

```
2E34 C37E2F JP #2F7E ; jump to additional code
```

en:

```
2F7E DD3403 INC (IX+#03) ; increase X position 2nd time
2F81 DD3403 INC (IX+#04) ; increase X-position 3th time
2F84 C3372E JP #2E37 ; jump back
```

Dat is voor nu wel goed.

Dan nu de sprite aanpassen naar een hamer en kijken hoe de animatie gedaan kan worden. Welke stappen doorloopt het in het origineel bij het draaien?

Nieuwe hamers gemaakt in sprite slots #58, #59, #5A en #5B. Dan de code aanpassen dat de juiste hamer gekozen wordt n.a.v. timer die van 0 t/m F loopt. Dat zou kunnen met een right shift 2 wat neerkomt op delen door 4. En dan bij de #58 optellen geeft de sprite waarde.

Het wisselen tussen open en dichte spring gebeurt hier:

```
2E1C E60F      AND    #0F          ; mask out left 4 bits. result between 0 and F
2E1E C2292E    JP     NZ,#2E29      ; if not zero, jump ahead..

2E21 FD7E01    LD     A,(IY+#01)   ; load A with sprite value
2E24 EE07      XOR    #07          ; flip the right 3 bits
2E26 FD7701    LD     (IY+#01),A  ; store result = change the bouncer from open to closed
```

Aanpassen:

```
2E1E CB3F      SRL    A          ; shift right A – divide by 2
2120 CB3F      SRL    A          ; shift right A – divide by 2
2122 C658      ADD    A,#08      ; add #58 – value between #58 and #5B
2124 0000      NOP,NOP
```

Ja, dat gaat goed.

Nu nog de kleur van de hamer aanpassen. Dat is kleurcode #00 met de groen van de pijp. Die kun je aanpassen, omdat er in dit veld geen pijp komt.

Kleuren aangepast.

Nu is de kleur van de 1UP anders dan in de andere velden. Liever altijd hetzelfde houden. Is nu de kleur van de achtergrond, maar die is dus anders. Kiezen voor een kleur die altijd hetzelfde blijft.

Voor de 1UP aanpassen naar: kleurcode #0B = geel. Ja, dat is goed.

Nu versie v0.83 opgeslagen.

De elevators zijn zichtbaar boven de spikes. Zit daar nog een kier dan? De spikes één pixel omhoog doen?

```
1121 3F 45 03 65  ; spike links
1125 3F 19 0F F7  ; lava links
1129 4F 45 03 65  ; spike rechts
112D 4F 19 0F F7  ; lava rechts
```

Is nu goed.

Jumpman gaat iets te vroeg dood. Nog voordat hij de spikes raakt.

De spikes zit hier:

```
276F 3A0562 LD A,(#6205) ; load A with Mario's Y position
2772 FE71 CP #71 ; top of elevator ? (death)
2774 DA7F27 JP C,#277F ; yes, die
```

Aanpassen naar een check op #6E.

De check op de lava raken in rivets is deze:

```
0B9A 3A2762 LD A,(#6227) ; load A with screen number
0B9D FE04 CP #04 ; is this the rivets?
0B9F C0 RET NZ ; if not return
;
0BA0 3A0562 LD A,(#6502) ; load A with Y-position jumpman
0BA3 FEF1 CP #F1 ; is jumpman below lowest girder?
0BA5 D8 RET C ; no, return
;
0BA6 AF XOR A ; A == #00
0BA7 320062 LD (#6200),A ; make mario dead
0BA8 C9 RET ; return
```

Dit checkt op Y-positie jumpman lager dan de onderste girder. Dat zou je hier ook kunnen gebruiken. Omdat je op geen enkel veld daaronder kunt komen kun je de check op het board hier weghalen en dan werkt dit ook voor de lava in het elevators board.

Ja, dat klopt. Ook even checken met de andere boarden of dat nog goed gaat. Ja, dat gaat allemaal nog goed.

Dan nu de rechter structuur gaan maken. Drie girders tussenvoegen en ladders om er te komen.

```
3C12: 03 30 F8 10 F8 ; bottom girder left
3C17: 03 EF F8 58 F8 ; bottom girder right
3C1C: 02 EF D0 60 D0 ; first girder right
3C21: 02 EF A8 60 A8 ; second girder right
3C26: 02 EF 80 60 80 ; third girder right
3C2B: 00 73 D0 73 F8 ; left bottom ladder right structure
3C30: 00 73 80 73 A8 ; left top ladder right structure
3C35: 00 DB 80 DB A8 ; right top ladder right structure
3C3A: 00 C3 58 C3 80 ; ladder to bowsers girder
3C3F: AA ; end code
```

Ziet er goed uit. De laatste trap moet later weer weg en pas geplaatst worden wanneer de drie muntjes opgepakt zijn.

Nu nog de koopa positioneren.

```

; 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

```

Dat is de tweede koopa. Positioneren op #A8.

Positioneren van de muntjes zodat je ze kunt pakken.

```

; 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

```

Aanpassen:

```

3E48 E0 73 0B C8
3E4C E1 74 0B F0
3E50 20 75 0B 80

```

Is nu goed.

Nu versie v0.84 opgeslagen.

Even testen speelbaarheid (zonder de vallende blokken) op difficulty 5.

De koopa op de bovenste girder is erg lastig. Dat kun je oplossen door hem niet de ladder te laten nemen naar het bovenste niveau. Of hem de kans te geven om weer terug een niveau lager te gaan.

Het liefste zo maken dat de koopa niet omhoog gaat naar de bovenste girder. Dat is een stuk relaxter omdat je ook nog rekening moet houden met de hamers. En anders wel erg afhankelijk van geluk.

```

335C A7      AND     A          ; \ Jump if there is a ladder nearby to go up
335D CA9933  JP      Z,#3399   ; /

```

Deze code wordt met een kans van 25% uitgevoerd. Als springen naar #3399 dan wordt de trap omhoog genomen. Dus daar nog iets voor maken dat test op board #03 en als zo, dan return.

```
335D CAED06      JP      Z,#06ED      ; jump to additional code
```

en:

```
06ED 3A2762      LD      A,(#6227)    ; load A with screen number  
06F0 FE03        CP      #03          ; is this screen #03 ?  
06F2 C8          RET     Z            ; yes, return without taking ladder  
;  
06F3 C39933      JP      #3399        ; jump to original code
```

Dat werkt iets te goed. Nu neemt hij nooit meer een trap omhoog. Dat moet alleen gedaan worden als de Y-waarde boven een bepaalde waarde zit.

```
06ED 3A2762      LD      A,(#6227)    ; load A with screen number  
06F0 FE03        CP      #03          ; is this screen #03 ?  
06F2 2006        JR      NZ,#AAAA     ; no, jump forward - AAAA  
;  
06F4 DD7EOF      LD      A,(IX+#0F)   ; load Y-position koopa  
06F7 FE81        CP      #81          ; is the koopa on the 3rd girder?  
06F9 D8          RET     C            ; yes, return without taking ladder  
;  
06FA C39933      JP      #3399        ; jump to original code - AAAA
```

Ja, gaat goed. Gaat niet meer naar de girder met Bowser. Ook nog even testen met rivets en barrels. Daar nog wel overal heen. Ja, dat gaat allemaal goed.

Dit is best al uitdagend om te spelen. Erg leuk.

Nu versie v0.85 opgeslagen.

Zodanig aanpassen dat de girder van Pauline meer naar rechts loopt en dat de vlaggenmast en vlag getekend worden. De bovenste trap niet tekenen. Die pas tekenen wanneer de drie munten verzameld zijn. Indien geen hergebruiken tekenen vlaggenmast van barrels board, dan zo maken dat ook de vier vallende blokken ook meteen getekend worden.

De girder van Pauline langer maken:

3BF9: 02 B7 38 68 38 ; girder girl is on

Voor het barrels board wordt de volgende routine gebruikt om de vlaggenmast te tekenen:

```
1EE7 21F61E      LD      HL, #1EF6    ; load start flagpole positions table  
;  
1EEA 11506A      LD      DE, #6A50   ; load DE with sprite destination  
1EED 011000      LD      BC, #0010  ; 4 items x 4 bytes = 16 bytes (#10)
```

```

1EF0  EDB0          LDIR           ; draw bonus item sprites
;
1EF2  210C3E        LD   HL, #3EOC  ; start of table data for hammers
1EF5  C9            RET            ; jump back
;
1EF6  B4 66 03 30   ; vlaggenmast stok
1EEA  B4 66 03 20   ; vlaggenmast stok
1EFE  B4 67 03 10   ; vlaggenmast top
1F02  AB 68 0E 20   ; vlag

```

Maar daar zit een extra aanroep in die plaats moet maken voor de aanroep naar deze additionele code. En die moet nu niet uitgevoerd worden. Kan je doen door die uit deze routine te halen en dan twee extra stukken code voor barrels en elevators (en een derde voor conveyors).

Aanpassen:

```

1EF2  C9            RET            ; jump back
1EF3  000000        NOP,NOP,NOP

```

en:

```

0C31  CDE71E        CALL  #1EE7      ; call routine to draw flagpole and flag
0C34  210C3E        LD   HL,#3EOC  ; start of table data for hammers
0C37  C9            RET            ; return

```

en de originele aanroep aanpassen naar tussencode:

```

1EBD  CD310C        CALL  #0C31      ; call additional code

```

Ja, dat gaat nog goed voor het barrel board.

Nu dan ook tussenvoegen op het elevators board.

Bonus items worden hier geplaatst:

```

10DE  21483E        LD   HL,#3E48      ; source is data table for bonus items on elevators
10E1  110C6A        LD   DE,#6A0C      ; destination is RAM area for bonus items
10E4  010C00        LD   BC,#000C      ; counter set for #0C bytes
10E7  EDB0          LDIR           ; copy

```

Aanpassen:

```

10DE  CD380C        CALL  #0C38      ; call additional code

```

en:

0C38	CDE71E	CALL	#1EE7	; call routine to draw flagpole and flag
0C3B	000000	NOP, NOP, NOP		; reserved for call to draw thwomp
0C3E	21483E	LD	HL,#3E48	; source is data table for bonus items
0C41	C9	RET		; return

Ja, dat gaat nu goed. De vlaggenmast wordt netjes getekend. Ziet er goed uit.

Checken bonus items en tekenen ladder. Hoe werkt dat en hoe kan dat hergebruikt worden?

3F62	3A2762	LD	A,(#6227)	; load A with screen number
3F65	3D	DEC	A	; is this the girders?
3F66	C0	RET	NZ	; if not return
;				
3F67	3AB064	LD	A,(#64B0)	; load A with bonus-item pickup counter
3F6A	FE00	CP	#00	; all bonus-items picked up?
3F6C	C8	RET	Z	; no, return
;				
3F6D	FEFO	CP	#F0	; all bonus-items picked up?
3F6F	2813	JR	Z,#AAAA	; yes, create ladder and start blinking – label AAAA
;				
3F71	FE10	CP	#10	; finished blinking?
3F73	C8	RET	Z	; yes, return
;				
3F74	3D	DEC	A	; decrement A
3F75	32B064	LD	(#64B0),A	; save back in bonus-item pickup counter
3F78	E610	AND	#10	; bit 4 set?
3F7A	3E63	LD	A,#63	; set sparkle sprite
3F7C	2802	JR	Z,XX	; skip ahead
3F7E	3E64	LD	A,#64	; reset sparkle sprite
3F80	32216A	LD	(#6A21),A	; set sparkle sprite number
3F83	C9	RET		; return
;				
3F84	3D	DEC	A	; decrement A – label AAAA
3F85	32B064	LD	(#64B0),A	; save back in bonus-item pickup counter
;				
3F88	CDCE3F	CALL	#3FCE	; create the ladder – label BBBB
3F8B	21206A	LD	HL,#6A20	; load HL with heart sprite
3F8E	3694	LD	(HL),#94	; set heart sprite x-position
3F90	23	INC	HL	; next
3F91	3663	LD	(HL),#63	; set sparkle sprite
3F93	23	INC	HL	; next
3F94	3609	LD	(HL),#09	; set heart sprite color
3F96	23	INC	HL	; next
3F97	3649	LD	HL,#49	; set heart sprite y-position
3F99	C9	RET		

Aanpassen zodat deze code voor zowel barrels, pies en elevators gebruikt kan worden:

```
3F62 C3FD24      JP    #24FD      ; jump to additional code
3F65 0000        NOP,NOP
```

en:

```
24FD 3A2762      LD    A,(#6227)   ; load A with screen number
2500 FE04        CP    #04         ; is this rivets ?
2502 C8          RET   Z           ; if yes, no ladder check, return
2503 C3673F      JP    #3F67      ; jump back to perform ladder check
```

Het aanmaken van de ladder board afhankelijk maken:

```
3FC8: 00 93 38 93 55 AA      ; test table data removed ladder
;
3FCE: 11C83F      LD    DE,#3FC8   ; load start table data
3FD1: CDA70D      CALL  #0DA7     ; draw screen
3FD4: 21C83F      LD    HL,#3FC8   ; load HL with start of table data for ladder
3FD7: DD210C63    LD    IX,#630C   ; #6300 is start, with 11 ladders added: 630C
3FDB: CD7524      CALL  #2475     ; #2475
3FDE: C9          RET   ;
```

Aanpassen:

```
3FD7 CD0625      CALL  #2506      ; call additional code
```

en:

```
2506 3A2762      LD    A,(#6227)   ; load A with screen number
;
2509 FE01        CP    #01         ; is this the barrels board?
250B DD210C63    LD    IX,#630C   ; #6300 is start, with 11 ladders added: 630C
250F C8          RET   Z           ; yes, return
;
2510 FE02        CP    #02         ; is this the conveyors board?
2512 00000000    LD    IX,#630C   ; #6300 is start, with ? ladders added: 63??
2516 C8          RET   Z           ; yes, return
;
2517 00000000    LD    IX,#630C   ; #6300 is start, with ? ladders added: 63??
251B C9          RET   ; return
```

Ja, dat gaat goed. Maar de ladder die getekend wordt zit aan de onderkant een graphic te hoog. Dus de Bowser girder in barrels board zit hoger dan in het elevators board.

Dus misschien handiger om de code van #3FCE t/m #3FDE weer vrij te geven en daarin de afwijkende ladder definitie op te nemen. En dan een stuk nieuwe code op #2506 met board afhankelijke stukken waarin de ladders getekend worden. En daar dan naartoe springen i.p.v. naar #3FCE.

Aanpassen:

```
3FC8: 00 93 38 93 54 AA      ; table data removed ladder barrels board
3FCE: 00 93 38 93 58 AA      ; table data removed ladder other boards
```

En de rest vrijgeven.

En aanpassen:

Aanroep voor het plaatsen van een ladder:

```
3F88 CD0625      CALL #2506      ; create the ladder
```

en:

```
2506 3A2762      LD A,(#6227)    ; load A with screen number  
;  
2509 FE01        CP #01          ; is this the barrels board?  
250B 200F        JR NZ,#AAAA    ; no, skip forward - label AAAA  
;  
; barrels board  
;  
250D 11C83F      LD DE,#3FC8    ; load start table data  
2510 CDA70D      CALL #0DA7     ; draw screen  
2513 21C83F      LD HL,#3FC8    ; load HL with start of table data for ladder  
2516 DD210C63    LD IX,#630C    ; #6300 is start, with 11 ladders added: 630C  
251A 1820        JR #CCCC      ; jump forward - CCCC  
;  
; elevators board  
;  
251C FE03        CP #03          ; is this the elevators board? - AAAA  
251E 200F        JR NZ,#AAAA    ; no, skip forward - label BBBB  
;  
2520 11CE3F      LD DE,#3FCE    ; load start table data  
2523 CDA70D      CALL #0DA7     ; draw screen  
2526 21CE3F      LD HL,#3FCE    ; load HL with start of table data for ladder  
2529 DD210963    LD IX,#6309    ; #6300 is start, with 9 ladders added: 6309  
252D 180D        JR #CCCC      ; jump forward - CCCC  
;  
; conveyors board  
;  
252F 11CE3F      LD DE,#3FCE    ; load start table data - BBBB  
2532 CDA70D      CALL #0DA7     ; draw screen  
2535 21CE3F      LD HL,#3FCE    ; load HL with start of table data for ladder  
2538 DD210C63    LD IX,#630C    ; #6300 is start, with 11 ladders added: 630C  
;  
253C CD7524      CALL #2475      ; add ladder to data structure of ladders - CCCC  
253F C9          RET           ;
```

en:

3BEA: 00 93 38 93 58 ; ladder leading up to girl

verwijderen en de rest van de stastructuur opschuiven.

Ja, dat gaat goed. Zowel bij barrels als elevators wordt de ladder getekend en kan beklimmen worden.

Nu versie v0.86 opgeslagen.

En dan nu de vallende blokken maken. Eerst de sprite tekenen en daarna de vier sprites tekenen en de bijbehorende datastructuur aanmaken. En dan laag plaatsen en collision detection aanpassen.

Het is sprite nr #6A geworden.

#6500 met interval van #10 is de structuur voor de spring, maar die worden niet allemaal gebruikt. Maar daar is wel al collision detection voor.

#6980 met interval van #4 zijn de sprites voor de springs, maar die worden niet allemaal gebruikt.

Even checken met de debugger, hoeveel springs tegelijk op het scherm en eventueel dan hergebruiken van deze structuren en sprites om de vallen de blokken te definiëren.

Er worden er maar twee gebruikt. En er wordt inderdaad op 10 springs getest. Dus je zou voor de datastructuur kunnen beginnen met #6540 en voor de sprites met #6990.

Aan het begin wordt voor alle slots wel wat data ingevuld 6507: 3B 00 02 02. Dat lijkt de datababel op #11A2 te zijn. Die wordt gebruikt op #1186 en die code wordt aangeroepen vanaf #102B (conveyors screen) en #1093 (elevators screen). De #3B is de sprite value en de #02 en #02 zijn de hitbox.

Het lijkt erop dat je alleen maar de structuur in #6500 hoeft te vullen met de benodigde waarden en dan door subroutine in #11D3 aan te roepen kun je de corresponderende sprites vullen.

Dus de aanroep op #1093 aanpassen zodat nieuwe code aangeroepen wordt en daar de boel vullen en dan de sprites aanmaken.

```
1093 CD4825      CALL  #2548      ; call additional code
```

en:

```
2540 A8 E8 C8 E8 A8 C0 C8 C0      ; data definition X/Y positions of the four blocks
;
2548 CD8611      CALL  #1186      ; set up the springs first
;
254B DD214065    LD    IX,#6540    ; load IX with start data structure for blocks
254F 214025      LD    HL,#2540    ; load HL with start data definition for blocks
2552 111000      LD    #DE,#10    ; load DE with offset
2555 0604        LD    B,#04      ; loop counter: create 4 blocks
;
2557 DD360001    LD    (IX+#00),#01  ; set block active - AAAA
255B DD36076A    LD    (IX+#07),#6A  ; set block sprite value
255F DD360908    LD    (IX+#09),#08  ; set hitbox x = 8
2563 DD360A08    LD    (IX+#0A),#08  ; set hitbox y = 8
2567 7E          LD    A,(HL)     ; load X from data definition
2568 DD7703      LD    (IX+#03),A   ; set X-position
256B 23          INC   HL         ; next table entry
256C 7E          LD    A,(HL)     ; load Y from data definition
256D DD7705      LD    (IX+#05),A   ; set Y-position
```

```

2570 23           INC   HL          ; next table entry
2571 DD19           ADD   IX,DE      ; set IX to start next block
2573 10E2           DJNZ  #AAAA      ; loop until zero – jump back to AAAA
;
2575 DD214065       LD    IX,#6540    ; load IX with start data structure for blocks
2579 219069       LD    HL,#6990    ; load HL with start block sprites
2555 0604           LD    B,#04      ; loop counter: create 4 block sprites
257C CDD311       CALL  #11D3      ; create sprites
;
257F C9           RET               ; return

```

Gaat niet goed. Wordt overschreven door code op #2E31. Dat doet aanpassingen voor alle springs.

De counter aanpassen:

```
2E10 0604           LD    B,#04      ; for B = 1 to #04.
```

Gaat nu goed. De sprite nog wel even tweaken. Ziet er al wel goed uit. Maar punten aan de onderkant accentueren.



Even de positionering aanpassen:

```
2540 A8 DC C8 DC A8 B4 C8 B4           ; data definition X/Y positions of the four blocks
```



Nu versie v.087 opgeslagen.

En dan nu de thwomp blokken laten vallen en weer omhoog laten gaan. Sneller omlaag laten vallen dan omhoog bewegen. Dat variabel maken zodat dat getuned kan worden. Ze alle vier tegelijk laten vallen. Wanneer beneden een bonk geluid afspelen en even wachten totdat weer omhoog gaan.

Bovenste positie is: #DC/#B4, onderste positie is #F0/#C8, dus #14 verplaatsen naar beneden.

In de structuur positie IX+#0B gebruiken om de richting aan te geven en IX+#0C als een timer. Dat hoeft in principe alleen voor de eerste bijgehouden te worden omdat ze toch allemaal tegelijk gaan. Dus #654B is de richting (#00 is naar beneden, #01 is naar boven) en #654C is een timer.

#1992 is nog een vrij slot in de main om de additionele code aan te roepen.

Aanpassen:

1992 CD8225 CALL #2582 ; call additional code

en:

```
2582 3A2762        LD     A,(#6227)      ; load A with screen number
2585 FE03            CP     #03              ; is this the elevators ?
2587 C0              RET    NZ                ; no, return
;
2588 3A4565          LD     A,(#6545)      ; load A with Y position first block
258B FEDC            CP     #DC              ; is it in the top position?
258D 2011            JR     NZ,#AAAA        ; no, jump forward – label AAAA
;
; block in top position
;
258F 3A4C65          LD     A,(#654C)      ; load A with timer
2592 3D              DEC    A                ; decrement timer
2593 324C65          LD     (#654C),A     ; store adjusted timer
;
2596 FE50            CP     #C0              ; is it time to fall?
2598 C0              RET    NZ                ; no, return
;
2599 3E00            LD     A,#00            ; A := #00
259B 324B65          LD     (#654B),A     ; set direction down
259E 181C            JR     #BBBB            ; jump forward to move down – label BBBB
;
25A0 FEF0            CP     #F0              ; is it in the bottom position? - AAAA
25A2 2011            JR     NZ,#CCCC        ; no, jump forward – label CCCC
;
; block in bottom position
;
25A4 3A4C65          LD     A,(#654C)      ; load A with timer
```

```

25A7 3D          DEC   A           ; decrement timer
25A8 324C65      LD    (#654C),A ; store adjusted timer
;
25AB FEBO        CP    #D0         ; is it time to rise?
25AD C0          RET   NZ          ; no, return
;
25AE 3E01        LD    A,#01       ; A := #01
25B0 324B65      LD    (#654B),A ; set direction up
25B3 1822        JR    #DDDD       ; jump forward to move up – label EEEE
;
; block is moving
;
25B5 3A4B65      LD    A,(#654B)  ; load A with direction - CCCC
25B8 FE00        CP    #00         ; is it moving down?
25BA 2016        JR    NZ,CCCC    ; no, jump forward to move up – label DDDD
;
; block is moving down
;
25BC 0E01        LD    C,#01       ; load C with 1 (down = Y pos + 1) – BBBB
25BE CD420C      CALL  #0C42       ; move all four blocks down – call ZZZZ
;
25C1 3A4565      LD    A,(#6545)  ; load A with Y position first block
25C4 FEFO        CP    #F0         ; is it in the bottom position?
25C6 C0          RET   NZ          ; no, return
;
25C7 3E03        LD    A,#03       ; load sound duration of 3
25C9 328260      LD    (#6082),A ; play boom sound using sound buffer
25CC 3E00        LD    A,#00       ; A := #00
25CE 324C65      LD    (#654C),A ; reset timer
25D1 C9          RET   ; return
;
; block is moving up
;
25D2 3A1A60      LD    A,(#FrameCounter) ; load A with frame counter - DDDD
25D5 0F          RRCA ; is the frame counter odd?
25D6 D8          RET   C           ; yes, return, skip 50% for half speed going up
;
25D7 0EFF        LD    C,FF         ; load A with 1 (down = Y pos - 1) - EEEE
25D9 CD420C      CALL  #0C42       ; move all four blocks down – call ZZZZ
;
25DC 3A4565      LD    A,(#6545)  ; load A with Y position first block
25DF FEDC        CP    #DC         ; is it in the top position?
25E1 C0          RET   NZ          ; no, return
;
25E2 3E00        LD    A,#00       ; A := #00
25E4 324C65      LD    (#654C),A ; reset timer
25E7 C9          RET   ; return

```

en:

```
0C42 0604      LD   B,#04      ; move 4 sprites - ZZZZ
0C44 DD214565    LD   IX,#6545    ; load IX with Y-position of first block data
0C48 219369      LD   HL,#6993    ; load HL with Y-position of first block sprite
0C4B 111000      LD   DE,#0010    ; load DE with offset
;
0C4E 79          LD   A,C      ; load A with move offset – direction - YYYY
0C4F 86          ADD  A,(HL)    ; add current Y-position data to offset
0C50 77          LD   (HL),A    ; store updated Y-position in block sprite
0C51 DD7700      LD   (IX+#00),A ; store updated Y-position in block data
;
0C54 23          INC  HL       ; move HL to next block sprite
0C55 23          INC  HL
0C56 23          INC  HL
0C57 23          INC  HL
0C58 DD19        ADD  IX,DE    ; move IX to next block data
;
0C5A 10F2        DJNZ #YYYY     ; loop until done – jump back to YYYY
;
0C5C C9          RET           ; return
```

Dit werkt. De blokken bewegen op en neer met pauzes ertussen. Ze lijken wel even snel op als neer te bewegen en de pauzes lijken ook even lang. Daar nog naar kijken.

Nu versie v0.88 opgeslagen.

Met debugger gaan kijken naar tuning.

Gaat nu goed. Alleen beneden kan hij korter wachten. Is aangepast.

Nu wel erg voorspelbaar. Zo maken dat de wachttijd random wordt vastgesteld. De kortste wachttijd is #D0. Dit is niet lang genoeg om onder beiden door te lopen. En daar dan een random getal van aftrekken. Dat dan opslaan in #654D. En daarop checken.

Aanpassen:

```
2596 C3F425      JP   #25F4      ; jump to additional code
```

en:

```
25F4 47          LD   B,A       ; store timer in B
25F5 3A4D65      LD   A,(#654D)  ; load A with timer threshold
25F8 90          SUB  B       ; is timer equal to threshold ?
25F9 C0          RET   NZ      ; no, return
25FA C39925      JP   #2599    ; jump back
```

Aanpassen:

```
25E7 3A1960      LD   A,(RNGTimer2)      ; load A with random number
25EA E67F        AND  #7F            ; random number between #00 and #80
25EC 47          LD   B,A           ; store in B
;
25ED 3ED0        LD   A,#D0           ; load minimal time threshold
25EF 90          SUB  B             ; adjust with random number
25F0 324D65      LD   (#654D),A    ; store in first block data
;
25F3 C9          RET              ; return
```

Tussen C0 en 80 uitkomen.

Dat betekent max 64 eraf trekken: rnd getal met AND 0111 1111.

Ja, dat gaat nu goed. De blokken echter 4 posities naar rechts plaatsen.

```
2540 AC DC CC DC AC B4 CC B4      ; data definition X/Y positions of the four blocks
```

Nee, dat wordt te makkelijk. Terugzetten:

```
2540 A8 DC C8 DC A8 B4 C8 B4      ; data definition X/Y positions of the four blocks
```

Tunen van de interval. Ja, dat is nu goed.

Ook nog even het geluid van de vallende hamers weghalen. Alleen geluid van de blokken horen.

Dat zit hier:

```
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:

```
2E69 000000      NOP,NOP,NOP
```

Ja, dat gaat goed.

Ook het geluid wanneer Bowser op de borst slaat (lijkt meer op gooien). Eruit halen.

Dat zit hier:

; kong is beating his chest

```
0445 21F739 LD HL,#39F7 ; start of table data
0448 CD4E00 CALL #004E ; update kong's sprites
044B 3E03 LD A,#03 ; load sound duration of 3
044D 328260 LD (#6082),A ; play boom sound using sound buffer
```

Aanpassen:

```
044D 000000 NOP,NOP,NOP
```

Later dit veld nog wel een achtergrondmuziek geven. Anders wel erg stil.

Nu even versie maken met achter elkaar de drie velden spelen.

Nu versie v0.89 opgeslagen.



De snelheid van de elevators in world 1 is heel erg langzaam. Daarnaast is de beschikbare tijd in het elevators board wel heel krap en misschien in world 1 t/m 4 zelfs te kort. Wellicht geldt dit ook voor het rivets board. Nog checken en aanpassen.

De langzame elevators zitten hier:

```

270C C21A27    JP      NZ,#271A          ; if level <> 1 then jump ahead

; slow elevators for level 1, japanese rom only?

270F E603      AND     #03              ; mask bits of timer, now between 0 and 3
2711 FE01      CP      #01              ; == 1 ?
2713 CA1E27    JP      Z,#271E          ; yes, skip ahead and return

2716 DA2227    JP      C,#2722          ; if greater, then jump ahead and move elevators ?

2719 C9        RET                ; else return

271A 0F        RRCA               ; rotate right the timer
271B DA2227    JP      C,#2722          ; if carry jump ahead and move the elevators (50% of time)

```

De conditionele jump op 270C altijd doen.

Ja, dat is nu goed.

In het elevator board zit aan de onderkant iets niet goed. De ladder heeft niet de goede uiteinden waardoor de rechter onderkant verstoort wordt. Dit kun je verhelpen door de rechter onderkant als laatste te tekenen (in ieder geval na de ladder). Zou bijvoorbeeld kunnen door ze te verwisselen.

De onderste rechter girder verwisselen met de laatste trap uit de definitie:

3C12: 03 EF F8 58 F8 ; bottom girder left
 3C35: 00 C3 58 C3 80 ; ladder to bowser's girder

Aanpassen naar:

3C12: 00 C3 58 C3 80 ; ladder to bowser's girder
 3C35: 03 EF F8 58 F8 ; bottom girder left

Ja, dat ziet er goed uit.

Soms ga je in elevator bord af wanneer je naar de elevator springt en dan zie je de elevator wel achter je. Misschien pas af gaan wanneer lager in de lava. Dan is er nog de kans om op de elevator te springen.

De check op de lava raken is deze:

```

0B9A 3A2762      LD      A,(#6227)      ; load A with screen number
0B9D FE04        CP      #04            ; is this the rivets?
0B9F C0        RET     NZ              ; if not return
0B9F 00        NOP                ; changed so it works in all boards
;
0BA0 3A0562      LD      A,(#6502)      ; load A with Y-position jumpman
0BA3 FEF1        CP      #F1            ; is jumpman below lowest girder?
0BA5 D8        RET     C               ; no, return
;

```

```

0BA6 AF XOR A ; A == #00
0BA7 320062 LD (#6200),A ; make mario dead
0BA9 C9 RET ; return

```

Aanpassen check op 0BA3:

```

0BA3 FEF1 CP #F1 ; is jumpman below lowest girder?

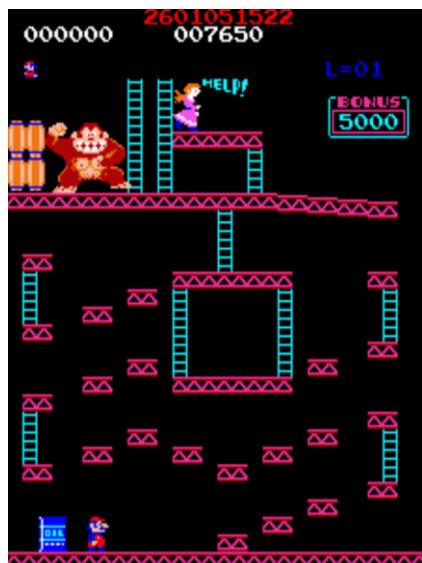
```

Niet lager dan #F6 want dan zie je jumpman onder de lava. Is niet mooi. Maar met waarde #F6 gaat het ook nog niet goed. Springt de elevator mis en landt dan alsnog lager in de lava. Dus is bij de normale DK waarschijnlijk ook zo. Maar valt daar minder op. Dus gewoon op #F1 laten staan, want dan bij aanraken lava ga je al af (en niet bij kopje onder gaan).

[Nu versie v0.90 opgeslagen.](#)

Dan nu met Powershell gaan werken aan een board dat vooral gebaseerd is op springen. Schuine stukken en springen en soort van palen waar je tussen kunt vallen. Alles opbouwen met Powershell en testen springbaarheid en later kijken hoe je het efficient in de romhack kan toevoegen.

Paaltjes maken gaat niet. Dat ziet het spel als losse girders en je kunt dan steeds op de volgende springen binnen de paal.



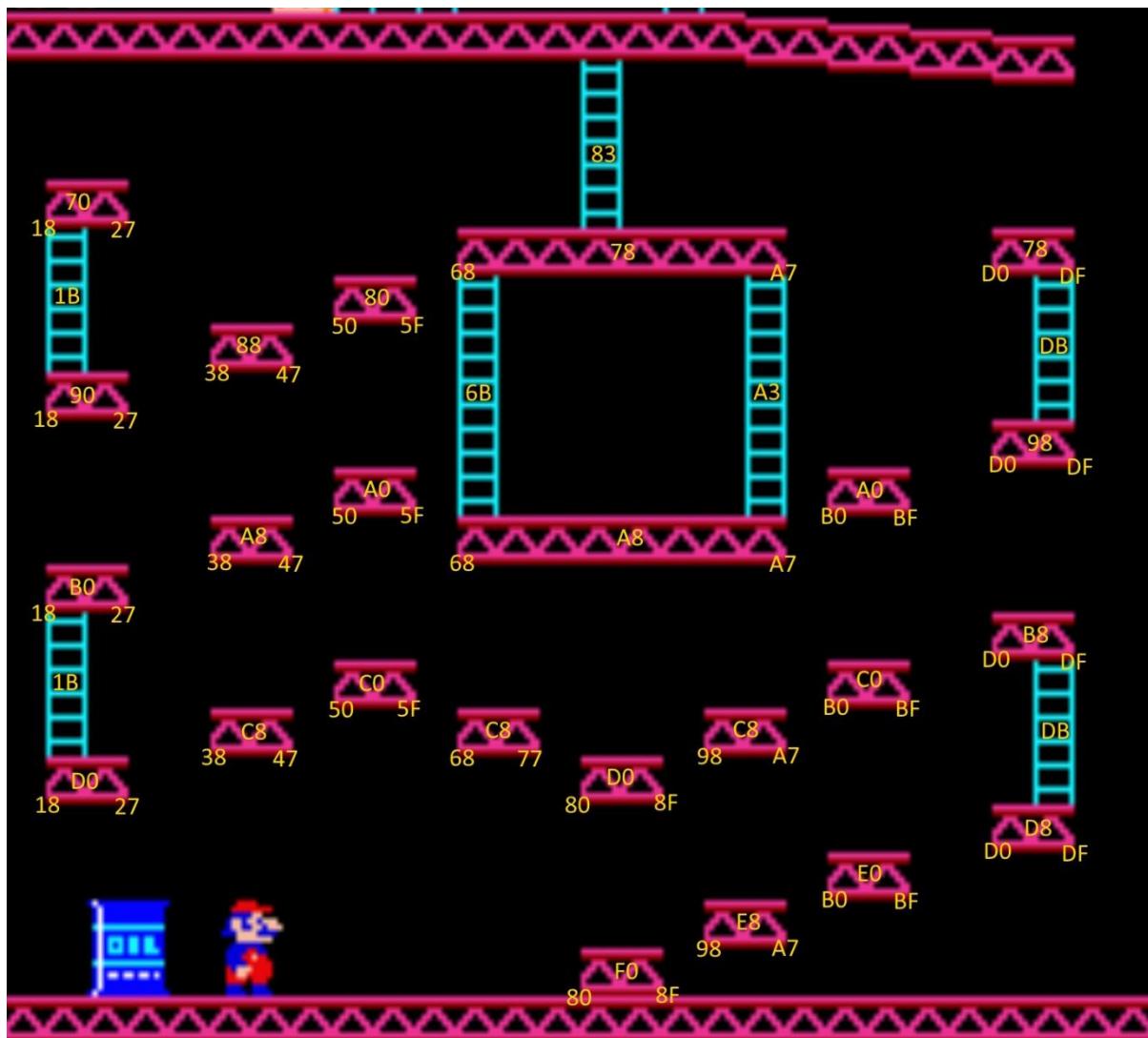
[Nu versie v0.91 opgeslagen.](#)

Dit is het uiteindelijke design. Hierin de munten plaatsen i.i.g. op de bovenste zijstructuren. In de middelste structuur een koopa laten rondlopen die niet naar de girder van Bowser mag gaan. Binnen in de middelste structuur 1 ster hangen (andere ster buiten beeld houden) en in world 3 en 6 in de middelste structuur een 1UP hangen (beide sterren buiten beeld houden). En dan vuurkogels als enemy op verschillende hoogten (nadenken over aantallen en random hoogten). En als het kan een vis die met een boog vanonder af vliegt. Onderaan het rechterdeel van het scherm lava tekenen.

Eerst even een plaat maken met de coördinaten van de verschillende elementen.

Daarna kijken hoe dit op een efficiënte manier opgebouwd kan worden.

```
@( (0x02, 0x8F, 0xF0, 0x80, 0xF0), (0x02, 0xA7, 0xE8, 0x98, 0xE8), (0x02, 0xBF, 0xE0, 0xB0, 0xE0),  
  (0x02, 0xDF, 0xD8, 0xD0, 0xD8), # staircase girders bottom  
  (0x00, 0xDB, 0xB8, 0xDB, 0xD8), # ladder bottom right  
  (0x02, 0xDF, 0xB8, 0xD0, 0xB8), (0x02, 0xBF, 0xC0, 0xB0, 0xC0), (0x02, 0xA7, 0xC8, 0x98, 0xC8),  
  (0x02, 0x8F, 0xD0, 0x80, 0xD0), (0x02, 0x77, 0xC8, 0x68, 0xC8), (0x02, 0x5F, 0xC0, 0x50, 0xC0),  
  (0x02, 0x47, 0xC8, 0x38, 0xC8), (0x02, 0x27, 0xD0, 0x18, 0xD0), # staircase girders 2nd level  
  (0x00, 0x1B, 0xB0, 0x1B, 0xD0), # ladder 2nd level left  
  (0x02, 0x27, 0xB0, 0x18, 0xB0), (0x02, 0x47, 0xA8, 0x38, 0xA8),  
  |(0x02, 0x5F, 0xA0, 0x50, 0xA0), # staricase girders 3rd level  
  (0x02, 0xA7, 0xA8, 0x68, 0xA8), (0x02, 0xA7, 0x78, 0x68, 0x78), # girders middle part  
  (0x00, 0x6B, 0x78, 0x6B, 0xA8), (0x00, 0xA3, 0x78, 0xA3, 0xA8), # ladders middle part  
  (0x02, 0x5F, 0x80, 0x50, 0x80), (0x02, 0x47, 0x88, 0x38, 0x88),  
  (0x02, 0x27, 0x90, 0x18, 0x90), (0x02, 0x27, 0x70, 0x18, 0x70), # staircase girders left top  
  (0x00, 0x1B, 0x70, 0x1B, 0x90), # ladder left top  
  (0x02, 0xBF, 0xA0, 0xB0, 0xA0), (0x02, 0xDF, 0x98, 0xD0, 0x98),  
  (0x02, 0xDF, 0x78, 0xD0, 0x78), # starcase girders right top  
  (0x00, 0xDB, 0x78, 0xDB, 0x98), # ladder right top  
  (0x00, 0x83, 0x54, 0x83, 0x78), # ladder to bowser level
```



Het handigste lijkt om de losse girder stukken in kolommen te tekenen. En dan per kolom een configurerbaar aantal girder stukken in een loop tekenen.

Eerst de vaste elementen tekenen en dan definities voor elke kolom maken en dan code om de kolommen te tekenen maken.

De vaste elementen:

3C3B: 00 63 18 63 58 ; kong's ladder (right)
3C40: 00 53 18 53 58 ; kong's ladder (left)
3C45: 02 B7 38 68 38 ; girder where girl sits
3C4A: 02 EF 58 10 58 ; kong's girder
3C4F: 03 EF F8 10 F8 ; bottom level girder
3C54: AA

En dan het adres voor de pies datastructure aanpassen:

0CDF 113B3C LD DE,#3C3B ; load DE with start of table data for conveyors

en:

2461 213B3C ID HL,#3C3B ; load HL with start of table data for conveyors

Ja, dat is goed.

Dan de ladders toevoegen:

3C54: 00 6B 78 6B A8 ; left ladder central structure
3C59: 00 A3 78 A3 A8 ; right ladder central structure
3C5E: 00 1B 70 1B 90 ; left top ladder
3C63: 00 DB 78 DB 98 ; right top ladder
3C68: 00 1B B0 1B D0 ; left bottom ladder
3C6D: 00 DB B8 DB D8 ; right bottom ladder
3C72: 00 83 58 83 78 ; central ladder to kong's girder
3C77: AA

Ja, dat is goed.

Dan de twee girders van het centrale deel toevoegen:

3C77: 02 A7 78 68 78 ; top girder central structure
3C7C: 02 A7 A8 68 A8 ; bottom girder central structure
3C81: AA

Ja, dat gaat goed.

En dan de kolommen eventjes als type patch opgeven met bovenste platform als data.

3C81: 06 27 70 18 70 ; column 1 – 4 platforms
3C86: 06 47 88 38 88 ; column 2 – 3 platforms
3C8B: 06 5F 80 50 80 ; column 3 – 3 platforms

```

3C90: 06 77 C8 68 C8 ; column 4 – 1 platform
3C95: 06 8F D0 80 D0 ; column 5 – 2 platforms
3C9A: 06 A7 C8 98 C8 ; column 6 – 2 platforms
3C9F: 06 BF A0 B0 A0 ; column 7 – 3 platforms
3CA4: 06 DF 78 D0 78 ; column 8 – 4 platforms
3CA9: AA

```

Ja, dat is nu allemaal goed.

Nu versie v0.92 opgeslagen.

En dan nu de code voor het tekenen van de patches met kruisen aanpassen naar het tekenen van een aantal platformen van 2 breed onder elkaar met een afstand van #20.

Om element van type 04, 05 en 06 te tekenen wordt de code van #0F2F gebruikt. Voor type 04 en 05 wordt eerst naar additionele code gesprongen om het graphic code aan te passen. Voor type 06 wordt deze gezet op #0F2D.

```

0F1E FE07 CP #07 ; <= 7 ?
0F20 F2CF0E JP P,#0ECF ; no, skip back and loop for next data item

0F23 FE04 CP #04 ; first data item == 4 ?
0F25 CA4C0F JP Z,#0F4C ; yes, skip ahead to handle

0F28 FE05 CP #05 ; first data item == 5 ?
0F2A CA510F JP Z,#0F51 ; yes, skip ahead to handle

; redraws screen when rivets has been completed

0F2D 3EFE LD A,#FE ; A := #FE

0F2F 32B563 LD (#63B5),A ; store into ???
0F32 2AAB63 LD HL,(#63AB) ; load HL with ???

0F35 3AB563 LD A,(#63B5) ; load A with ???
0F38 77 LD (HL),A ; store into ???
0F39 012000 LD BC,#0020 ; set offset to #20
0F3C 09 ADD HL,BC ; add offset for next
0F3D 3AB163 LD A,(#63B1) ; load A with ???
0F40 D608 SUB #08 ; subtract 8
0F42 32B163 LD (#63B1),A ; store result. done ?
0F45 D2350F JP NC,#0F35 ; no, loop again

0F48 13 INC DE ; else increase DE
0F49 C3A70D JP #0DA7 ; jump back

0F4C 3EE0 LD A,#E0 ; A := #E0
0F4E C32F0F JP #0F2F ; jump back
|
0F51 3EB0 LD A,#B0 ; A := #B0
0F53 C32F0F JP #0F2F ; jump back

```

Het stukje code van #0F2F t/m #0F4B tekent in een loop een girder van begin (rechts) naar einde (links) met steeds een offset van #20. Als het klaar is, wordt de DE verhoogt (index in de datababel met de te tekenen elementen) en wordt opnieuw gesprongen naar #0DA7.

Voor type 06 eigenlijk naar een heel stuk eigen code springen om de verschillende platformen te tekenen.

Aanpassen:

```
0F28 FE06      CP    #06          ; first data item == 6 ?
0F2A CAAAAAA   JP    Z,#AAAA      ; yes, skip ahead to handle – jump to AAAA
;
0F2D 3EB0      LD    A,#B0      ; A := #B0 – now the value for data item type 5
```

En even testen of dit inderdaad overal goed gaat door dummy afhandeling voor type 6 te maken:

```
3CAA 13        INC   DE
3CAB C3A70D    JP    #0DA7
```

Ja, dat gaat goed.

Dan nu de afhandeling van type 6 maken. Checken wat er in de geheugenplaatsen staat, wanneer de uitvoering bij #3CAA komt.

Dat is dan de eerste kolom:

3C81: 06 27 70 18 70 ; column 1 – 4 platforms

In memory staat dan:

```
63A0 00 00 00 00 00 00 00 00 00 00 00 00 6E 77 8E 77 00
63B0 00 0F 00 06 07 F0 00 00 00 00 00 00 00 00 00 00 00
```

En de pointer in de datastructuur staat op #3C86, maar dat is nà de INC DE. Dus bij begin van de routine staat hij nog op #3C85: dat is het laatste data element. Dus daar het aantal platformen in opslaan. In #63AB staat het VRAM adres van het eerste element en in #63B4 staat het type.

```
3CAA 1A        LD    A,(DE)      ; load number of platforms needed - AAAA
3CAB 47        LD    B,A        ; store in B
3CAC D5        PUSH  DE        ; save DE for later
;
3CAD 2AAB63    LD    HL,(#63AB)  ; load HL with start VRAM address
;
3CB0 36D0      LD    (HL),#D0   ; draw right part of platform - BBBB
3CB2 112000    LD    DE,#0020   ; set offset to 1 VRAM to the left
3CB5 19        ADD   HL,DE      ; set new VRAM address
3CB6 36D0      LD    (HL),#D0   ; draw left part of platform
3CB8 11E4FF    LD    DE,#FFE4   ; set offset to 1 VRAM right and 4 down
```

```

3CBB 19          ADD   HL,DE      ; set new VRAM address
3CBC 10F2        DJNZ  #3CB0      ; repeat until B=0, jump to BBBB
;
3CBE D1          POP   DE         ; restore saved DE
3CBF 13          INC   DE
3CC0 C3A70D      JP    #0DA7

```

en de definities aanpassen:

```

3C81: 06 27 70 00 04 ; column 1 – 4 platforms
3C86: 06 47 88 00 03 ; column 2 – 3 platforms
3C8B: 06 5F 80 00 03 ; column 3 – 3 platforms
3C90: 06 77 C8 00 01 ; column 4 – 1 platform
3C95: 06 8F D0 00 02 ; column 5 – 2 platforms
3C9A: 06 A7 C9 00 02 ; column 6 – 2 platforms
3C9F: 06 BF A0 00 03 ; column 7 – 3 platforms
3CA4: 06 DF 78 00 04 ; column 8 – 4 platforms
3CA9: AA

```

Ja, dat gaat goed. De platformen worden goed getekend.

Nog wel allerlei restzaken als draaipunten, oil can, koopa spawn, hamers, enz die nog verwijderd c.q. aangepast moeten worden.

Nu versie v0.93 opgeslagen.

Nu de onderste girder korter maken en rechts daarvan lava maken. Misschien kan daar code (of de ruimte) van het tekenen van de draaipunten voor gebruikt worden.

3C4F: 03 **68** F8 10 F8 ; bottom level girder

De lava zijn sprites met een breedte van 2 characters. Dan zijn er 8 sprites naast elkaar nodig.

De data voor het tekenen van de conveyor draaipunten zit hier: #3E24

```

; 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

; 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

```

Als je de twee ladders meeneemt is dat de data voor acht sprites. Wat als we die gewoon aanpassen naar de juiste waarden.

```
3E1C: 77 19 02 F6  
3E20: 77 19 02 F6  
3E24: 87 19 02 F6  
3E28: 97 19 02 F6  
3E2C: A7 19 02 F6  
3E30: B7 19 02 F6  
3E34: C7 19 02 F6  
3E38: D7 19 02 F6
```

Er zijn maar zeven sprites nodig. De eerste daarom maar gelijk gemaakt aan de tweede.

Ziet er goed uit.

Nu de oil can en het vuur en de twee ladder stukjes weghalen.

```
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  
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 t/m 1059 vrijgeven. Dit kan later gebruikt worden voor initialisatie van de vuurkogels en de vis en eventueel een koopa in het middenstuk. Ook hiervoor zit een stuk code voor initialisatie pies wat ook herbruikt kan worden.

Er worden nog stukjes trap getekend op de plek van de retractable ladders. Waar komt dat vandaan? O.a. op de eerste van de eerste kolom. Lijkt dit te zijn:

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

en:

```

; jump here from #0CD1
; a patch ?

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

; called from #3FA0 above

3FA6 3E02 LD A,#02 ; A := 2
3FA8 F7 RST #30 ; check to see if the level is pie factory. If not, RET to #3FA3 [then jump to #0D5F]

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

3FAE 3610 LD (HL),#10 ; clear the top of the ladder
3FB0 23 INC HL
3FB1 23 INC HL ; next address
3FB2 36C0 LD (HL),#C0 ; draw a ladder 2 rows down
3FB4 218C74 LD HL,#748C ; set HL for next loop - does the other side of the screen ; [sloppy? this instruction not needed on 2nd loop]
3FB7 10F5 DJNZ #3FAE ; Next B

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

3FBA: 00 00 00 00 00 00 ; unused

```

Eerst kijken door volgende aan te passen:

0CD1 C35F0D JP #0D5F

Ja, nu worden de ladder stukjes niet meer getekend.

Vrijgeven #3FA0 t/m 3FB9.

Ja, dat is nu goed.

Nu versie v0.94 opgeslagen.

Het spawnen van de koopa's niet meer doen. Eén koopa plaatsen op de bovenste girder van de midden structuur. Misschien een tweede koopa plaatsen beneden links van jumpman zodat hij meteen achterna gezeten wordt?

Fireball's in conveyors wordt hier gedaan:

```

32D2 CD7834 CALL #3478 ; Handle fireball movement while coming out of oilcan on conveyors
32D5 C9 RET ; return

```

De #32D2 aanpassen naar 3 NOP's en dan de code van #3478 t/m #34B8 vrijgeven.

Ja, dat gaat goed. Het spawnen in de andere velden gaat ook nog goed.

Eerst de muntjes goed plaatsen en kijken of na het pakken van alle drie de muntjes de ladder getekend wordt.

```

; bonus items on conveyors

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

```

Aanpassen:

```

3E3C D6 73 0A B0      ; coin bottom right
3E40 1F 74 0A 68      ; coin top left
3E44 D8 75 0A 70      ; coin top right

```

Ja, dat werkt en ook de ladder naar Pauline wordt getekend.

Alleen level is al klaar wanneer jumpman de bovenste girder opgaat. Dat moet niet. Level moet pas klaar zijn wanneer naar de vlag gesprongen wordt.

Die check zit hier:

```

; checks for end of level ?

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

1E5f 1F rra                 ; else rotate right with carry|
1E60 3A0562 LD A,(#6205)       ; load A with y position of mario
1E63 dA7A1E JP c,#1E7A         ; skip ahead on girders and elevators

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

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

; check for end of level on girders and elevators

1E7A FE31 CP #31              ; are we on top level (rescued girl?)
1E7C D0 RET NC                ; no, return

```

De aparte code voor conveyors board hoeft helemaal niet.

Aanpassen:

```
1E66 C37A1E      JP      #1E7A      ; jump forward
```

en

1E69 t/m 1E79 kan dan vrijgegeven worden.

Ja, dat gaat goed. En springen naar vlaggenmast gaat ook goed om level te eindigen. De vlaggenmast wordt echter nog niet getekend.

Dan nu de sterren weghalen. Dit level vooralsnog zonder sterren spelen. Daarom ze weghalen door ze buiten beeld te plaatsen.

De positie van de twee sterren wordt hier bepaald:

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

Aanpassen:

```
3E10: 00 8D 00 B4
```

Dat gaat goed: de sterren zijn nu weg.

Nu versie v0.95 opgeslagen.

Nog twee dingen gaan niet goed. Na springen naar vlaggenmast wordt wel de bonus getoond (afhankelijk van hoogte van raken vlaggenmast), maar het level wordt niet beëindigd. En jumpman kan voorbij de Kong girders lopen en langs bowser bewegen.

Bij elevators board kan jumpman niet verder naar links lopen bij de Kong girders. Zijn x-positie is dan #6B. Kan niet echt vinden waar dat gecheckt wordt. Het enige wat erop lijkt is de code:

```
; A is even

300E 0E93      LD      C,#93      ; C := #93
3010 0F        RRCA
3011 0F        RRCA      ; roll right twice
3012 D21730    JP      NC,#3017    ; no carry, skip next step

3015 0E6C      LD      C,#6C      ; C := #6C

3017 07        RLCA      ; roll left
3018 DA3130    JP      C,#3031    ; if carry, skip ahead

301B 79        LD      A,C      ; A := C
301C E6F0      AND     #F0      ; mask bits, 4 lowest bits set to zero
301E 4F        LD      C,A      ; store back into C
301F C33130    JP      #3031    ; skip ahead
```

Deze begint op #3009 en wordt inderdaad aangeroepen vanuit de code die uitgevoerd wordt wanneer de joystick naar links bewogen wordt. Kijken wat er gebeurt wanneer deze waarde aangepast wordt. Kun je dan in girders board en elevators board verder naar links bewegen?

Nee, dat kan niet. Dus dat is het niet. Deze waarde terugzetten.

Het is ook zo dat bij een jump tegen de kong ladder jumpman ook terugspringt zoals bij een wall jump. Dit verder uitzoeken.

Het lijkt hier te zitten:

```
2429 1C      INC   E          ; DE := #0001
242A FEEA      CP    #EA        ; is Mario's position > #EA ?
242C D0      RET   NC         ; yes, return

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

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

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

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

Kan kloppen, want de check op #6C wordt alleen gedaan bij girders of elevators. Even checken door hier de #6C aan te passen.

Ja, dat is de goede plek. Dus hier de check op het board aanpassen. Voor allemaal doen behalve voor rivets. Alleen dat past daar niet, dus uitstapje naar additionele code nodig.

Aanpassen:

```
242E C3C33C      JP    #3CC3        ; jump to additional code
2431 0000      NOP,NOP
```

en:

```
3CC3 3A2762    LD    A,(#6227)  ; load A with screen number
3CC6 FE04      CP    #04        ; is it rivets?
3CC8 C8      RET   Z          ; yes, return
3CC9 C33324    JP    #2433      ; no, jump back
```

Ja, dat gaat goed nu.

Dan kijken waarom het einde niet goed gaat. Blijft hangen na toekennen van de bonus.

Eerst de vlaggenmast eens tekenen. De code hiervoor zit op #1EE7.

Tussenvoegen in lege stuk van initialisatie conveyors:

```
1049 CDE71E      CALL    #1EE7      ; draw flagpole and flag
```

De vlaggenmast wordt nu goed getekend. Maar einde level werkt niet. Bowser gaat niet omhoog maar de 1UP knippert wel.

Maar ook gezien dat de lava rechts niet ver genoeg doorloopt.

```
3E1C: E7 19 02 F6  
3E20: 77 19 02 F6  
3E24: 87 19 02 F6  
3E28: 97 19 02 F6  
3E2C: A7 19 02 F6  
3E30: B7 19 02 F6  
3E34: C7 19 02 F6  
3E38: D7 19 02 F6
```

Ja, loopt nu wel goed door. Was dus wel goed geteld met 8 sprites!

End of level check zit hier (is al wel aangepast):

```

; checks for end of level ?

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

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

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

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

; check for end of level on girders and elevators

1E7A FE31 CP #31 ; are we on top level (rescued girl?)
1E7C D0 RET NC ; no, return

1E7D C36D1E JP #1E6D ; level has been fished. jump to end of level routine.

```

#1E7A is aangepast naar code die end of level checkt na springen naar vlaggenmast. Maar die wordt nu alleen gedaan voor girders en elevators.

Verder is de #1E66 vervangen door een JP #1E7A. Dus dat zou betekenen dat voor de conveyors ook de end check gedaan zou moeten worden. Met debugger kijken wat er gebeurt daar.

De code wordt gewoon goed doorlopen. Dan wordt op #1E85 de Gamemode2 (#600A) aangepast en wordt er naar de routine hoger gesprongen (#00D2), maar dan wordt het blijkbaar niet goed afgehandeld. Kan dat komen door een probleem met de stack? Op #00D2 wordt van alles gePOPt, maar als dat scheef gaat, dan werkt het niet. Onderzoeken wat er bij barrels en elevators gebeurt en dat vergelijken met wat er bij pies gebeurt.

Checken SP net voor de call en na het terugkeren en wat er van de verschillende registers weggeschreven wordt en weer ingelezen wordt. Dat moet gelijk zijn.

Watchpoint op #06FE – net na de call playing a game

Watchpoint op #00D2 – net na de call playing a game

Watchpoint op #00D9 – net na de POps

En dan registreren SP en waardes die daar staan.

06FE:

PC	0701
SP	6BEE
AF	0000
BC	0000
DE	00D1
HL	06FE
IX	76C1
IY	FFFF
AF2	0000
BC2	0000

SP wijst naar 6BEE.

6BA0	00 00
6BB0	00 00
6BC0	00 00
6BD0	00 00
6BE0	00 00
6BF0	FF FF C1 76 F2 60 E0 FF 04 93 14 03 50 03 CD 02 00 IY 00 IX 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6C00	00 00
6C10	00 00
6C20	00 00

Lastig te interpreteren.

De 00D2 is als HL op de stack gepushed en dan de RST en die heeft als return address de #00CA. Aan het einde van een level dan wordt er een POP HL gedaan. Dat zorgt ervoor dat de stack pointer op dat punt niet naar #00CA wijst, maar naar de #00D2.

```
1E85 3E16      LD      A,#16          ; else A := #16
1E87 320A60    LD      (GameMode2),A   ; store into game mode2
1E8A E1        POP    HL              ; pop stack to get higher address
1E8B C9        RET               ; return to a higher level [returns to #00D2]
```

Dus eigenlijk even kijken waar de SP op #1E8A naar wijst. En dan na de POP moet de SP wijzen naar de #00D2. Anders wordt er niet goed teruggesprongen. Dat is het probleem niet. Er wordt wel degelijk teruggesprongen naar #00D2.

En anders terug naar een eerdere versie en dan kijken of het daar wel werkt. Bijvoorbeeld met alleen een nieuw veld met één lange ladder naar boven en verder geen additionele code. Als het dan wel werkt, dan langzaam de rest van de code toevoegen en blijven testen.

Wat wellicht ook zou kunnen, is dat er ergens iets niet goed gaat met return adressen (bijvoorbeeld JP en terug met RET o.i.d.). Maar dan zou je verwachten dat de POP hierboven niet goed werkt, maar dat is niet zo.

En anders even kijken afhandeling van de conveyors qua klimmen van Kong e.d. Waar zit dat en hoe komt het programma daar? En komt het daar nu dan wel/niet?

Dat is op #1615 wanneer de Gamemode2 op 16 staat en eerst de sprites gewist worden. De uitvoering lijkt daar te komen want de sprites worden inderdaad gewist. En er klinkt ook het einde level muziekje. Dus daarvandaan debuggen en per stap kijken wat er gebeurt en waar er iets onverwachts gebeurt.

Even gekken, maar uiteraard duidelijk nu. Bij pies board is de einde level animatie heel anders. Doet iets met de conveyors en schuift DK van zijn willekeurige positie op de Kong girder terug naar links en gaat dan klimmen. Maar dat veronderstelt waarschijnlijk allerlei ingestelde zaken die nu wellicht niet goed staan. Dus eigenlijk voor dit board dezelfde end of level doen als voor girders en elevators. Daar is een selectie voor: deze aanpassen. Als dat werkt dan kan het hele gebeuren van end-of-level van conveyors vrijgegeven worden (daar waar niet gedeeld gebruikt).

De end of level afhandeling wordt hier gebaseerd op het board:

```
; 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
```

Aanpassen:

```
161B C3CC3C      JP #3CCC      ; jump to additional code
161E 00          NOP
```

en:

```
3CCC FE04        CP #04        ; is this the rivets?
3CCE CA4116      JP Z,#1641    ; yes, jump to code for rivets
3CD1 C31F16      JP #161F      ; no, jump to code for barrels, pies and elevators
```

Ja, dat gaat nu bij allemaal goed.

Nu versie v0.96 opgeslagen.

Eventueel stukken voor conveyors board finish vrijgeven:

#162F t/m #1640 en #16A3 t/m #1707.

Hiermee is er wel weer wat extra ruimte vrijgekomen. Kan eventueel voor extra enemies gebruikt worden of voor het plaatsen van bowser tekst in grotere letters op het titelscherm.

Wel checken of het allemaal nog werkt. Ja het werkt nog allemaal. Wel dat bowser verspringt bij het pies board vlak voor het klimmen van de ladders.

Met twee schermafdrukken gecontroleerd. Maar Bowser staat in het pies board en het elevators board op dezelfde plek. Met de debugger checken waar de sprites van bowser verschoven worden.

De kong sprites beginnen bij #6908. Niet eenvoudig te checken.

Wat wel te zien is, is dat de bowser figuur een kwartslag draait, zodat hij van opzij te zien is en kijkend naar rechts. Daarna begint het klimmen. Maar in het pies board staat hij in deze zijwaartse stand een aantal pixels te hoog.

Dat klopt. Bij het tekenen van de zijwaartse bowser wordt een verschuiving van een aantal pixels naar boven (negatief) gedaan voor het girders board.

Dat zit hier:

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

1669 210B69 LD HL,#6908 ; load HL with start of kong sprite
166C 0EFC LD C,#FC ; set movement for -4 pixels
166E FF RST #38 ; move kong
166F C9 RET ; return
```

Nee, dat is het niet. Want doet het inderdaad alleen voor het girders board.

Mogelijk zit het net iets verder. Daar wordt een verschuiving van vier pixels naar beneden (positief) gedaan voor het elevators board

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

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

Dit aanpassen:

De RST #30 werkt op basis van een binary:

barrels = 0001
pies = 0010
elevators = 0100
rivets = 1000

Dus als je wilt dat hij naast elevators (0100 = #04) ook pies (0010 = #02) doet, dan moet je de binary 0110 gebruiken en dat is #06!

Aanpassen:

```
1680 3E06          LD    A,#06          ; A := 6 = 0110 code for pies and elevators
```

Ja, dat gaat nu goed.

Dan nu kijken om een koopa te spawnen in de middenstructuur. Kan op dezelfde manier zoals dat bij de elevators gedaan wordt.

```
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
```

Toevoegen:

```
3F9A DD210064 LD    IX,#6400      ; load IX with start of koopa
3F9E DD360001 LD    (IX+#00),#01  ; set koopa active
3FA2 DD360358 LD    (IX+#03),#80  ; set koopa X position
3FA6 DD360E58 LD    (IX+#0E),#80  ; set koopa X position #2
3FAA DD360580 LD    (IX+#05),#70  ; set koopa Y position
3FAE DD360F80 LD    (IX+#0F),#70  ; set koopa Y position #2
3FB2 C9           RET             ; return
```

en aanroepen vanuit het vrijgegeven stukje initialisatie van het pies board:

```
104C CD9A3F        CALL   #3F9A      ; initialize koopa
```

Ja, de koopa wordt nu getekend. En hij gaat in de middenstructuur bewegen. Alleen gaat hij ook naar de kong girder. en dat willen we niet.

Dat is voor het elevators board al gedaan met onderstaande code:

```
06ED 3A2762 LD    A,(#6227)    ; load A with screen number
06F0 FE03  CP    #03          ; is this screen #03 ?
06F2 2006  JR    NZ,#AAAA    ; no, jump forward - AAAA
;
06F4 DD7EOF LD    A,(IX+#0F)  ; load Y-position koopa
06F7 FE81  CP    #81          ; is the koopa on the 3rd girder?
06F9 D8     RET   C           ; yes, return without taking ladder
;
06FA C39933 JP    #3399      ; jump to original code - AAAA
```

De koopa neemt de ladder niet als hij zelf op een positie zit hoger dan #81. In het pies board zit hij op de bovenste girder met een Y-positie van #70. Dat geeft bij een CP met #81 een carry en dus een return en neemt de trap dan niet.

Alleen moet deze code ook gaan gelden voor het elevators board:

06F0	C3B33F	JP	#3FB3	; jump to additional code
06F3	00		NOP	

en:

3FB3	FE03	CP	#03	; is this the elevators ?
3FB5	CAF406	JP	Z,#06F4	; jump back: check koopa position
3FB8	FE02	CP	#02	; is this the conveyors ?
3FBA	CAF406	JP	Z,#06F4	; jump back: check koopa position
3FBD	C3FA06	JP	#06FA	; jump back: do not check koopa position

Gaat inderdaad niet naar het bovenste niveau. Ook nog even de correcte werking bij de andere boards testen. Ja, die gaan ook goed.

Nu versie v0.97 opgeslagen.

Nu het palette aanpassen.

Willen we eventueel in de hogere werelden nog het licht laten knipperen als een soort van Bowser actie? Dan moeten we het palette van het elevators board toepassen. En dan kunnen we het palette van het pies board gebruiken als donker palette. Nee, niet doen want de boards hebben al twee verschillende paletten en dan loop je de kans dat sprites andere kleuren krijgen.

Dan kijken naar een bruin palette.

Eerst het palette van het elevators board naar het pies board kopiëren zodat de sprites goed zijn en dan de kleuren van de achtergrond graphics aanpassen naar de bruine variant.

Sprites zijn nu goed.

Dan nu de kleuren aanpassen:



#00/#0A - #07/#07

De ladders even blauw laten en de platformen dit design geven:



Links is sprite #B3 en rechts is sprite #B4.

Om deze te gaan gebruiken is de volgende aanpassing nodig:

3CBO 36B4 LD (HL),#B4 ; draw right part of platform - BBBB

en:

3CB6 36B3 LD (HL),#B3 ; draw left part of platform

Ja, dat ziet er goed uit.

Eventueel de blokjes girders aanpassen naar grond girders:

3C3B: 00 63 18 63 58 ; kong's ladder (right)

3C40: 00 53 18 53 58 ; kong's ladder (left)

3C45: 03 B7 38 68 38 ; girder where girl sits

3C4A: 03 EF 58 10 58 ; kong's girder

3C4F: 03 EF F8 10 F8 ; bottom level girder

3C54: AA

3C77: 03 A7 78 68 78 ; top girder central structure

3C7C: 03 A7 A8 68 A8 ; bottom girder central structure

Alleen de ladder naar het kong level gaat niet goed. Die moet getekend worden voordat de girders getekend worden.

Dat is deze:

3C72: 00 83 58 83 78 ; central ladder to kong's girder

Dus die omwisselen met kongs girder: #3C4A <-> 3C72.

Ja, dat is nu goed.

Alleen de trap naar Pauline wordt altijd pas later getekend en die loopt dus niet. Daarom Kong's girder en Paulines girder weer terug naar de blokjes girder.

De muntjes hebben nog de verkeerde kleur: moet kleurcode #0B worden.

3E3C D6 73 0B B0 ; coin bottom right

3E40 1F 74 0B 68 ; coin top left

3E44 D8 75 0B 70 ; coin top right

Ja, dat is nu goed.

Nu, alleen nog de kleur van de trappen aanpassen.

#02/#0F geeft een bruine kleur die ook in een vraagtekenblok zit. Kan interessant zijn als we nog een vraagtekenblok willen toevoegen.

#06/#0E geeft een grijze kleur.

Bij alle velden de trappen met dezelfde donkergrijze kleur gemaakt. Ziet er goed uit en betekent ook dat de bonus time altijd in dezezelfde kleur weergegeven wordt.

Ziet er nu zo uit:



Nu versie v0.98 opgeslagen.

Dan nu de horizontale vuurkogels maken. Als het goed is worden de pies nog wel geïnitialiseerd. Dat zou mooi zijn, want dan kun je ze een vuurkogel sprite geven en ze dan in een extra main aanroep gaan laten bewegen.

Eerst checken of de pies inderdaad geïnitialiseerd worden. De datastructuur voor de 6 pies begint bij #65A0:

65A0	00	00	00	00	00	00	00	4B	0E	04	02	00	00	00	00	00
65B0	00	00	00	00	00	00	00	4B	0E	04	02	00	00	00	00	00
65C0	00	00	00	00	00	00	00	4B	0E	04	02	00	00	00	00	00
65D0	00	00	00	00	00	00	00	4B	0E	04	02	00	00	00	00	00
65E0	00	00	00	00	00	00	00	4B	0E	04	02	00	00	00	00	00
65F0	00	00	00	00	00	00	00	4B	0E	04	02	00	00	00	00	00
6600	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

IX+#07 = graphic

IX+#08 = color

IX+#09 = hitbox width

IX+#0A = hitbox height

Dus wordt voor een deel al gevuld.

Andere velden die belangrijk zijn:

IX+#00 = active status

IX+#03 = X-position

IX+#05 = Y-position

Kunnen we de eerste laten vliegen? Gewoon laten rondlopen over het scherm? Om te testen op welke hoogtes het interessant is, welke richtingen interessant zijn en welke snelheid van de vuurkogel interessant is.

Aanroepen vanuit de main.

Op #199B is nog een vrij slot voor een aanroep vanuit de main:

```
199B CDFD25      CALL  #25FD      ; call additional code
```

en:

```
25FD DD21A065    LD   IX,#65A0    ; load IX with start first fire bullet
2601 DD7E03      LD   A,(IX+#03)  ; load A with X-position fire bullet
;
2604 3C          INC  A          ; move one position
2605 000000      NOP,NOP,NOP
;
2608 DD7703      LD   (IX+#03),A  ; store x-position
260B DD360001    LD   (IX+#00),#01 ; activate
260F DD3605A0    LD   (IX+#05),#A0 ; store y-position
;
2613 C9          RET
```

Gaat niet goed. Zie geen sprite. Zie wel dat de waarden in de datastructuur aangepast worden, maar de sprites (die beginnen bij #69B8) worden niet overeenkomstig bijgewerkt. Daarom code uitbreiden en tevens een check op pies board toevoegen.

```
25FD 3E02        LD   A,#02      ; check level for pies
25FF F7          RST  #30      ; if not pies: return
;
2600 DD21A065    LD   IX,#65A0    ; load IX with start data for first fire bullet
2604 FD21B869    LD   IY,#69B8    ; load IY with start sprite first fire bullet
;
2608 DD7E03      LD   A,(IX+#03)  ; load A with X-position fire bullet
260B 3C          INC  A          ; move one position
260C 000000      NOP,NOP,NOP
;
260F DD360001    LD   (IX+#00),#01 ; activate
```

```

2613 DD7703      LD   (IX+#03),A    ; store x-position in data structure
2616 FD7700      LD   (IY+#00),A    ; store x-position in sprite
2619 DD3605A0    LD   (IX+#05),#A0  ; store y-position in data structure
261D FD3603A0    LD   (IY+#03),#A0  ; store y-position in sprite
;
2621 C9          RET

```

Ja, gaat nu wel goed. Wel de X-positie met 2 i.p.v. 1 verhogen voor een acceptabele snelheid. Met 1 is te langzaam. Met 3 is te snel.

De sprite van de pie aanpassen naar een vuurkogel. Die wordt ook nog geanimeerd.

De kleuren aanpassen kleur #01 gebruiken.

De kleurcode is nu #0E en wordt hier gedefinieerd:

```
3E18: 4B 0E 04 02      ; ???
```

Aanpassen:

```
3E18: 4B 01 04 02
```

En dan de kleurcode #01 van het laatste blok kopiëren naar het tweede blok.

Dat is nu goed.

Nu de vuurkogel iets beter vormgeven. Ja dat is al beter.

En dan de vuurkogel bij elke verplaatsing verticaal spiegelen.

De gewone heeft karakter #4B en de gespiegelde heeft karakter #4C.

Code aanpassen om het effect te zien:

```

2621 DD7E07      LD   A,(IX+#07)  ; load character
;
2624 FE4B        CP   #4B        ; is it 4B?
2626 3E4C        LD   A,#4C
2628 2802        JR   Z,AAAA    ; yes, skip next step
262A 3E4B        LD   A,#4B
;
262C DD7707      LD   (IX+#07),A  ; set new character
262F FD7701      LD   (IY+#01),A  ; set new character
;
2632 C9          RET

```

Het effect is gaaf, maar gaat veel te snel. Dus niet bij elke stap doen. Dus daar is een timer voor nodig. Of afhankelijk van de positie maken.

Aanpassen:

```

2621 E603      AND #07      ; make A value between #00 and #07.
2623 FE00      CP #00      ; A == #00?
2625 C0        RET NZ      ; no, return
;
2626 DD7E07    LD A,(IX+#07) ; load character
;
2629 FE4B      CP #4B      ; is it 4B?
262B 3E4C      LD A,#4C
262D 2802      JR Z,AAAA   ; yes, skip next step
262F 3E4B      LD A,#4B
;
2631 DD7707    LD (IX+#07),A ; set new character
2634 FD7701    LD (IY+#01),A ; set new character
;
2637 C9        RET

```

Ja, dat ziet er goed uit.

De hitbox height is te klein. Altijd raken in de Y-richting:

3E18: 4B 01 04 **03**

Nu testen met verschillende hoogtes en eventueel richtingen. En beoordelen of het zinvol is om te variëren in de hoogte.

```

2619 DD3605C8    LD (IX+#05),#C8 ; store y-position in data structure
261D FD3603C8    LD (IY+#03),#C8 ; store y-position in sprite

```

De #C8 positie is helemaal goed. De #88 positie is ook goed. Maakt het bovenin een stuk lastiger. Dus code nu zo maken dat de IX en IY twee keer goedgezet wordt en er een routine aangeroepen wordt die de bullet afhandelt met een extra stuk eraan toegevoegd dat wanneer active, er eerst een variabele wachttijd gedaan wordt, dan het bewegen gestart wordt en wanneer rechts buiten scherm dat dan gedeactiveerd wordt en de timer gereset wordt.

Nu versie v0.99 opgeslagen.

Aanpassen:

```

25FD 3E02      LD A,#02      ; check level for pies
25FF F7        RST #30      ; if not pies: return
;
2600 DD21A065    LD IX,#65A0   ; load IX with start data for bottom fire bullet
2604 FD21B869    LD IY,#69B8   ; load IY with start sprite bottom fire bullet
2608 06C8        LD B,#C8      ; load B with Y-position
260A 3A1860    LD A,RngTimer1 ; load A with random number
260D CD2126    CALL #2621    ; process bottom fire bullet - AAAA
;

```

```

2610 DD21B065 LD IX,#65B0 ; load IX with start data for top fire bullet
2614 FD21BC69 LD IY,#69BC ; load IY with start sprite top fire bullet
2618 0688 LD B,#88 ; load B with Y-position
261A 3A1960 LD A,RngTimer2 ; load A with random number
261D CD2126 CALL #2621 ; process top fire bullet - AAAA
;
2620 C9 RET ; return

```

en:

```

2621 4F LD C,A ; save random number in C - AAAA
2622 DD7E00 LD A,(IX+#00) ; load status of the fire bullet
2625 FE01 CP #01 ; is it active?
2627 2815 JR Z,#BBBB ; yes, jump forward - BBBB
;
; fire bullet is inactive
;
2629 DD7E0B LD A,(IX+#0B) ; load inactivity timer
262C 3D DEC A ; decrease inactivity timer
262D DD770B LD (IX+#0B),A ; save adjusted inactivity timer
;
2630 FEF0 CP #F0 ; time to activate bullet ?
2632 C0 RET NZ ; no, don't activate and return
;
2633 79 LD A,C ; load A with saved random number from C
2634 E610 AND #10 ; make it a number between 00 and 16
2636 DD770B LD (IX+#0B),A ; store new random inactivity timer
;
2639 DD360001 LD (IX+#00),#01 ; make fire bullet active
263D C9 RET ; return
;
; fire bullet is active
;
263E 3A1A60 LD A,(FrameCounter) ; load A with framecounter - BBBB
2641 E607 AND #07 ; A is a number between 0 and 7
2643 FE00 CP #00 ; A == #00?
2645 2011 JR NZ,#CCCC ; no, do not change character - CCCC
;
2647 DD7E07 LD A,(IX+#07) ; load current character
;
264A FE4B CP #4B ; is it 4B?
264C 3E4C LD A,#4C
264E 2802 JR Z,DDDD ; yes, skip next step - DDDD
2650 3E4B LD A,#4B
;
2652 DD7707 LD (IX+#07),A ; set new character - DDDD
2655 FD7701 LD (IY+#01),A ; set new character
;

```

```

2658 78          LD   A,B          ; load A with Y-position - CCCC
2659 DD7705      LD   (IX+#05),A    ; store y-position in data structure
265C FD7703      LD   (IY+#03),A    ; store y-position in sprite
;
266F DD7E03      LD   A,(IX+#03)   ; load A with current X-position
2662 3C          INC  A           ; set to next position
2663 3C          INC  A           ; set to next position
2664 DD7703      LD   (IX+#03),A    ; store x-position in data structure
2667 FD7700      LD   (IY+#00),A    ; store x-position in sprite
;
266A FE00          CP   #00          ; is the bullet back at the left side?
266C C0          RET  NZ          ; no, done for now, return
;
266D DD360000    LD   (IX+#00),#00  ; make fire bullet inactive
2671 C9          RET                    ; return

```

Dat werkt. In één keer. En gaaf om te spelen. Niet eenvoudig.

Nu versie v0.100 opgeslagen.

Nu een aantal keren spelen om te testen ook met world 5 voor de hoogste difficulty waardoor de koopa sneller beweegt.

Voor het effect zou het ook gaaf zijn wanneer bij activeren van de fire bullet een geluid klinkt (zelfde geluid als bij de vallende blokken in elevator board).

Aanpassen:

```
2633 C37226      JP   #2672        ; jump to additional code - AAAA
```

en:

```

2672 3E03          LD   A,#03        ; load sound duration of 3 - AAAA
2674 328260        LD   (#6082),A    ; play boom sound using sound buffer
;
2677 79          LD   A,C          ; load A with saved random number from C
2678 E610          AND  #10         ; make it a number between 00 and 16
267A C33626        JP   #2636        ; jump back
;

```

Ja, dat werkt. En dat geeft wel een heel gaaf extra effect aan de fire bullets.

Nu versie v0.101 opgeslagen.

Dan nu kijken of we een 1up ergens kunnen plaatsen. Of wellicht laten rondvliegen. Plaatsen is lastig denk ik, want je wilt het wel een beetje een uitdaging geven. Dan als snel te eenvoudig of veel te moeilijk.

Wat wel leuk lijkt om het in het rechterbovendeel van het elevators board rond te laten bewegen. En dat dan in world 3 en 6.

De 1 UP is een groene paddenstoel die wanneer je hem pakt de tekst 1UP aangeeft.

Eerst de paddenstoel sprite maken. Heeft sprite graphic #6B.

Een stuk code maken dat vanuit de main aangeroepen wordt. #19AA is nog een vrij slot vanuit de main. Als spriteslot dezelfde gebruiken als voor de groene pijpen in het barrelsboard die start bij #6A30.

```
19AA CD7D26      CALL #267D      ; call routine to handle extra lives -AAAA
```

en:

```
267D 3E02      LD A,#03      ; A := #04 (0100 binary) - AAAA
267F F7        RST #30       ; if not elevators: return
;
2680 3A2962    LD A,(#6229)   ; load A with world number
2683 FE03      CP #03       ; is this world 3?
2685 2805      JR Z,#BBBB    ; yes, jump ahead - BBBB
2687 FE06      CP #06       ; is this world 6?
2689 2801      JR Z,#BBBB    ; yes, jump ahead - BBBB
;
268B C9        RET          ; return
;
268C DD21306A  LD IX,#6A30   ; load IX with start address 1UP sprite - BBBB
2690 DD360080  LD (IX+#00),#80  ; set sprite X-position
2694 DD36016B  LD (IX+#01),#6B  ; set sprite character
2698 DD360207  LD (IX+#02),#07  ; set sprite color
269C DD360380  LD (IX+#03),#90  ; set sprite Y-position
;
26A0 C9        RET          ; return
```

Eerst even testen met world 3 en elevators.

```
095E 01 65 3A 01 00 00 00
```

Ja, dat gaat goed. Alleen in elevators in world 3 en 6. Sprite ziet er goed uit.

[Nu versie 0.102 opgeslagen.](#)

Dan nu zo maken dat de sprite over de onderste girder met de koopa gaat bewegen van links naar rechts. En dan NOP's i.p.v. call voor de check van het oppakken van de 1UP.

Geheugenplaats voor vastleggen status 1UP: #64C0:

- #00 - starting board
- #01 - 1UP moving, not grabbed
- #02 - 1UP grabbed, displaying 1UP
- #03 - 1UP grabbed, no processing anymore

Aanpassen:

```

268C DD21346A LD IX,#6A34 ; load IX with start address 1UP sprite
2690 3AC064 LD A,(#64C0) ; load A with 1UP status
2693 EF RST #28 ; jump based on A
;
2694 9C26 ; A == #00 – jump to AAAA
2696 B226 ; A == #01 – jump to BBBB
2698 A316 ; A == #02 – jump to CCCC
269A C016 ; A == #03 – jump to DDDD
;
; 1 UP status = #00 (starting board)
;
269C DD360080 LD (IX+#00),#80 ; set sprite X-position
26A0 DD36016B LD (IX+#01),#6B ; set sprite character
26A4 DD360207 LD (IX+#02),#07 ; set sprite color
26A8 DD360380 LD (IX+#03),#A0 ; set sprite Y-position
;
26AC 3E01 LD A,#01 ; A := #01
26AE 32C064 LD (#64C0),A ; set 1UP status to moving
;
26B1 C9 RET
;
; 1 UP status = #01 (moving, not grabbed yet)
;
26B2 3AC164 LD A,(#64C1) ; load A with 1UP direction
26B5 FE00 CP #00 ; is the direction #00 (moving right)?
26B7 DD7E00 LD A,(IX+#00) ; load A with sprite X-position
26BA 200F JR NZ,#AAAA ; no, jump forward - AAAA
;
26BC 3C INC A ; increment A, move right
26BD DD7700 LD (IX+#00),A ; store adjusted sprite X-position
26C0 FEB8 CP #D0 ; rightmost position reached ?
26C2 2014 JR NZ,#BBBB ; no, jump forward - BBBB
26C4 3E01 LD A,#01 ; A := #01
26C6 32C164 LD (#64C1),A ; set 1UP direction to left
26C9 180D JR #BBBB ; jump forward - BBBB
;
26CB 3D DEC A ; decrement A, move left - AAAA
26CC DD7700 LD (IX+#00),A ; store adjusted sprite X-position
26CF FE80 CP #7E ; leftmost position reached ?
26D1 2005 JR NZ,#BBBB ; no, jump forward - BBBB
26D3 3E00 LD A,#00 ; A := #00

```

```

26D5 32C164      LD   (#64C1),A    ; set 1UP direction to right
;
26D8 3A0562      LD   A,(#6205)   ; load A with jumpman Y-position - BBBB
26DB FEA0        CP   #A0        ; is jumpman at correct Y-position?
26DD C0          RET  NZ        ; no, ready for now, return
;
26DE 3A0362      LD   A,(#6203)   ; load A with jumpman X-position
26E1 47          LD   B,A        ; save to B
26E2 DD7E00      LD   A,(IX+#00)  ; load A with sprite X-position
26E5 90          SUB  B          ; subtract B
26E6 FE02        CP   #02        ; result smaller than #02?
26E8 D0          RET  NC        ; no, ready for now, return
;
26E9 3E02        LD   A,#02       ; A := #02
26EB 32C064      LD   (#64C0),A  ; set 1UP status to grabbed
;
26EE 3E01        LD   A,#01       ; 
26F0 212862      LD   HL,#6228   ; load HL number of lives remaining
26F3 34          INC  (HL)      ; increase number of lives
26F4 C3B806      JP   #06B8     ; update the top of screen and return from there
;
26F7 C9          RET             ; return

```

Hier nu geen ruimte meer.

```

;
; 1 UP status = #02 (grabbed)
;
16A3 3AC264      LD   A,(#64C2)   ; load A with display timer
16A6 3D          DEC  A          ; decrease A
16A7 32C264      LD   (#64C2),A  ; save adjusted timer
16AA FEA0        CP   #A0        ; timer expired?
16AC 2809        JR   Z,#AAAA   ; yes, jump forward - AAAA
;
16AE DD36016C    LD   (IX+#01),#6C ; set sprite to 1UP graphic
16B2 DD360201    LD   (IX+#02),#01 ; set sprite to item scoring color
16B6 C9          RET             ; return
;
16B7 DD360000    LD   (IX+#00),#00 ; set sprite off screen - AAAA
16BB 3E03        LD   A,#03       ; A := #03
16BD 32C064      LD   (#64C0),A  ; set 1UP status to grabbed, done processing
16C0 C9          RET             ; return

```

Ja, dat gaat allemaal goed. Het extra leven wordt toegekend.

Nu versie v0.104 opgeslagen.

De sprite is alleen in het midden doorzichtig en daar zie je de hamers en koopa doorheen. Dit verhelpen door een sprite die erachter staat te gebruiken.

De sprite #6940 is de laagste vrije sprite. Die proberen.

```
268C DD214069 LD IX,#6940 ; load IX with start address 1UP sprite
```

Ja, dat gaat goed.

Het toe kennen van een extra leven bij 7000 eruit halen.

Dat zit hier:

```
0373 3E01 LD A,#01 ; A := 1
0375 322D62 LD (#622D),A ; store into extra life indicator
0378 212862 LD HL,#6228 ; load HL with address of number of lives remaining
037B 34 INC (HL) ; increase
037C C3B806 JP #06B8 ; skip ahead and update # of lives on the screen
```

Geen verhoging door #037B in NOP te veranderen.

Dat klopt inderdaad. Gaat goed. Nee, gaat niet goed. Bij begin van spel in het titelscherm wordt deze code ook aangeroepen en dan gaat het vreemd: geeft continue geluid, geeft 6 levens in beeld en geeft verkeerde karakters in de wereld aanduiding.

Anders doen:

```
036E 212160 LD HL,ExtraLifeThreshold ; load HL with score needed for extra life
0371 BE CP (HL) ; compare player's score to high score. is it greater?
0372 D8 RET C ; no, return

0373 3E01 LD A,#01 ; A := 1
0375 322D62 LD (#622D),A ; store into extra life indicator
0378 212862 LD HL,#6228 ; load HL with address of number of lives remaining
037B 34 INC (HL) ; increase
037C C3B806 JP #06B8 ; skip ahead and update # of lives on the screen
```

Gewoon de RET C op 0372 aanpassen. Daarmee is de conditie altijd waar en zou je zeggen dat het extra leven ook nooit toegekend wordt.

Gaat in ieder geval goed met het titelscherm. En het extra leven wordt ook niet toegekend.

De boardvolgorde aanpassen:

```

178E DF      RST    #18          ; count down timer and only continue here if zero, else RET
178F 2A2A62  LD     HL,(#622A)   ; load HL with address for this screen/level
1792 23      INC    HL           ; next screen
1793 7E      LD     A,(HL)      ; load A with the screen for next
1794 FE7F    CP     #7F          ; at end ?
1796 C29D17  JP     NZ,#179D    ; no, skip next 2 steps

1799 21733A  LD     HL,#3A73    ; load HL with table for screens/levels for level 5+
179C 7E      LD     A,(HL)      ; load A with the screen

179D 222A62  LD     (#622A),HL  ; store screen address lookup for next time
17A0 322762  LD     (#6227),A   ; store A into screen number

```

Met als datababel:

```

3A65: 01 04          ; level 1
3A67: 01 03 04       ; level 2
3A6A: 01 02 03 04    ; level 3
3A6E: 01 02 01 03 04  ; level 4
3A73: 01 02 01 03 01 04 ; level 5 +
3A79: 7F             ; end code

```

Eigenlijk heel simpel. Loopt door de hele tabel heen en springt dan bij 7F terug naar 3A73: de beginwaarde van eerste screen bij level 5+

Dus nu aanpassen:

```

3A65: 01 02 03 04      ; level 1+
3A69: 7F

```

en:

```
1799 21653A LD HL,#3A65 ; load HL with table for screens/levels for level 1+
```

Voor rivets beëindigen is er soortgelijke code:

Ja dus:

```

193D 2A2A62  LD     HL,(#622A)   ; load HL with contents of #622A. this is a pointer to the levels/screens data
1940 23      INC    HL           ; increase HL. = next level
1941 7E      LD     A,(HL)      ; load A with contents of HL = the screen we are going to play next
1942 fe7f    CP     #7f          ; is this the end code ?
1944 c24B19  JP     NZ,#194B    ; no, skip next 2 steps

1947 21733A  LD     HL,#3A73    ; yes, load HL with #3A73 = start of table data for screens/levels for level 5+
194A 7E      LD     A,(HL)      ; load A with screen number from table

```

Aanpassen:

```
1947 21653A LD HL,#3A65 ; yes, load HL table for screens/levels for level 1+
```

Testen door onsterfelijkheid uit te zetten.

Ja, dat gaat goed.

Nu versie v0.105 opgeslagen.

Nog doen:

- Geluidje afspelen bij pakken 1UP.
- Wat te doen met de muntjes op het rivets board? Niet logisch. De hamer pas laten zien als alledrie de muntjes gepakt? Of veranderen in iets anders?
- Muziek aanpassen.
- BOWSER nintendo kleuren font op titelscherm.
- Eventueel ornamenten voor de sier.

Play the bonus sound after grabbing the 1UP.

```
26E9 3E02      LD   A,#02      ; A := #02
26EB 32C064    LD   (#64C0),A ; set 1UP status to grabbed
;
26EE 3E01      LD   A,#01
26F0 212862    LD   HL,#6228  ; load HL number of lives remaining
26F3 34        INC  (HL)     ; increase number of lives
26F4 C3B806    JP   #06B8    ; update the top of screen and return from there
;
26F7 C9        RET
;
```

Aanpassen:

```
26EB C3B622    JP   #22B6    ; jump to additional code - AAAA
```

en:

```
22B6 32C064    LD   (#64C0),A ; set 1UP status to grabbed - AAAA
;
22B9 CD9D1D    CALL #1D9D    ; call routine to play bonus sound
22BC C3EE26    JP   #26EE    ; jump back
```

Ja, dat gaat goed.

Nu zo maken dat de bijl in rivets pas getoond wordt wanneer alle bonus items gepakt zijn.

De code nu voor het tekenen van de trap naar Pauline is gedeeld voor barrels, pies en elevators (rivets wordt uitgesloten). Zo'n zelfde code maken voor rivets, maar dan niet de trap neerzetten en het blinken, maar alleen de bijl sprite in beeld zetten (en initieel dus buiten beeld zetten).

Deze code aanroepen vanuit de main:

```
19B6 CD651E      CALL #1E65      ; call routine to check coins on rivets
```

en:

```
1E65 3A2762      LD A,(#6227)   ; load A with screen number
1E68 FE04        CP #04        ; is this the rivets?
1E6A C0          RET NZ        ; if not return
;
1E6B 3AB064      LD A,(#64B0)   ; load A with bonus-item pickup counter
1E6E FE00        CP #00        ; all bonus-items picked up?
1E70 C8          RET Z         ; no, return
;
1E71 3EB7        LD A,#B7      ; load A with axx X-position
1E73 32586A      LD (#6A58),A  ; make axx visible
1E76 C9          RET           ; return
```

De bijl is de laatste sprite in dit blok:

```
0B6F 57 19 02 F6 - #6A40
0B73 67 19 02 F6 - #6A44
0B77 77 19 02 F6 - #6A48
0B7B 87 19 02 F6 - #6A4C
0B7F 97 19 02 F6 - #6A50
0B83 A7 19 02 F6 - #6A54
0B87 B7 53 03 E6 - #6A58
```

Aanpassen:

```
0B87 00 53 03 E6
```

Ja, dat gaat goed. De bijl wordt getoond nadat de drie muntjes gepakt zijn. De controle op het eindigen van de rivets gaat alleen op basis van alle rivets unplugged en daarna op de positie beneden. Dat is ongeacht collision met de bijl of niet. Daar ook een conditie toevoegen dat alle muntjes gepakt moeten zijn.

De huidige code die het level einde regelt:

```
0BAB 3A9062      LD A,(#6290)   ; load A with number of rivets left
0BAE A7          AND A         ; all done with rivets?
0BAF C0          RET NZ        ; no, return
;
0BB0 3A0362      LD A,(#6203)   ; load A with jumpman X-position
0BB3 FEB9        CP #B9        ; is it #B9?
0BB5 C0          RET NZ        ; no return
;
0BB6 3A0562      LD A,(#6205)   ; load A with jumpman Y-position
0BB9 FEF0        CP #FO        ; is it #FO?
0BBC C0          RET NZ        ; no return
```

;
0BBC C3851E JP #1E85 ; jump back, rivets finished

Aanpassen:

0BBC C3C116 JP #16C1 ; jump to additional code - AAAA

en:

16C1 3AB064 LD A,(#64B0) ; load A with bonus-item pickup counter - AAAA
16C4 FE00 CP #00 ; all bonus-items picked up?
16C6 C8 RET Z ; no, return
;
16C7 C3851E JP #1E85 ; jump back, rivets finished

Ja dat gaat goed. Nu niet meer wanneer alle rivets gepakt, maar nog niet alle muntjes dat je zonder bijl sprite zichtbaar toch het level kon eindigen op de plek van de bijl.

De vuurbal die geschoten wordt bij rivets ook van een explosie geluid voorzien voor een extra effect.

Aanpassen:

2235 C3CA16 JP #16CA ; jump to additional code - AAAA

en

16CA 3E03 LD A,#03 ; load sound duration of 3 - AAAA
16CC 328260 LD (#6082),A ; play boom sound using sound buffer
;
16CF 3A1860 LD A,(RngTimer1) ; load A with random number
;
16D2 C33822 JP #2238 ; jump back

De bijl bij het einde van rivets weghalen. Wordt namelijk niet met de andere sprites weggehaald. Staat nu voor jumpman. Veel leuker als jumpman zichtbaar is en staat te kijken wanneer bowser in de lava valt, maar dat is nu niet te zien.

Aanpassen:

16C7 C3D516 JP #16D5 ; jump to additional code - AAAA

en:

16D5 3E00 LD A,#00 ; load A with axx X-position
16D7 32586A LD (#6A58),A ; make axx invisible
;
16DA C3851E JP #1E85 ; jump back, rivets finished

Ja, dat is nu allemaal goed.

Nu versie v1.06 opgeslagen.

Misschien nog het palette aanpassen zodat de koopa's en de muntjes meekleuren met de achtergrond? Even bekijken en dan keuze maken. Nee, niet doen.

Op de titelpagina de naam BOWSER'S met sprites in gekleurde Nintendo font neerzetten.

De data daarvoor (allemaal in dezelfde kleur nog):

```
3478: 40 3F 01 80 ; B
347C: 50 4F 07 80 ; O
3480: 60 5F 01 80 ; W
3484: 70 6F 01 80 ; S
3488: 80 6E 05 80 ; E
348C: 90 5C 07 80 ; R
3490: A0 3C 05 80 ; '
3494: B0 6F 01 80 ; S
```

En dan ergens code toevoegen die naar spritegeheugens kopieert tijdens het opbouwen van het titelscherm.

Het DONKEY KONG logo wordt hier getekend:

```
; draws DONKEY KONG logo to screen

07F7 21083D LD HL,#3D08 ; load HL with start of table data

07FA 3EB0 LD A,#B0 ; A := #B0 = code for girder on screen
07FC 46 LD B,(HL) ; get first data. this is used as a loop counter
07FD 23 INC HL ; next table entry
07FE 5E LD E,(HL) ; load E with table data
07FF 23 INC HL ; next entry
0800 56 LD D,(HL) ; load D with table data. DE now has an address
```

Daar dan het tekenen aan toevoegen. Aanpassen:

```
07F7 C39834 JP #3498 ; jump to additional code
```

en:

```
3498 217834 LD HL,#3478 ; load HL with table data for BOWSER'S sprites
349B 118069 LD DE,#6980 ; load DE with sprites start address
349E 012000 LD BC,0020 ; #20 (= 32) bytes to copy
34A1 EDB0 LDIR ; copy
;
```

```
34A3 21083D      LD      HL,#3D08      ; load HL with start of table data  
34A6 C3FA07      JP      #07FA
```

Logo wordt in de goede kleuren getekend.

Nog wel positionering doen en letters logo wellicht iets dichter op elkaar.

Nu versie v0.107 opgeslagen.



De KONG moet een regel naar beneden.

BOWSER TAKEOVER aanpassen naar TAKEOVER en één regel naar beneden.

BOWSER's sprites moeten 3 omhoog

De DONKEY moet 1 regel omhoog.

KONG 1 regel naar beneden, is oorspronkelijke +2:

```
3D59: 05 31 77 05 11 77 02 F2 76 02 D1 76 02 D4 76      ; K  
3D68: 05 91 76 05 71 76 01 51 76 01 55 76 05 31 76      ; O  
3D77: 05 F1 75 02 D2 75 02 B3 75 05 91 75              ; N  
3D83: 03 52 75 05 31 75 01 11 75 01 15 75            ; G (part 1)  
3D8F: 01 F1 74 01 F3 74 01 F5 74 02 D3 74            ; G (part 2)
```

TAKEOVER 1 regel naar beneden:

```
          B O W S E R      T A K E O V E R  
36FD  EE 76 12 1F 27 23 15 22 10 24 11 1B 15 1F 26 15 22 3F  
36FD  EF 76 10 10 10 24 11 1B 15 1F 26 15 22 10 10 10 10 3F
```

BOWSER'S drie regels omhoog:

```
3478: 48 3F 01 6C ; B  
347C: 58 4F 07 6C ; O  
3480: 68 5F 01 6C ; W  
3484: 78 6F 01 6C ; S  
3488: 88 6E 05 6C ; E  
348C: 98 5C 07 6C ; R  
3490: A4 3C 05 6C ; '  
3494: AC 6F 01 6C ; S
```

DONKEY 1 regel naar boven:

```
3D08: 05 87 77 01 67 77 01 6B 77 03 48 77 ; D  
3D14: 05 07 77 01 E7 76 01 EB 76 05 C7 76 ; O  
3D20: 05 87 76 02 68 76 02 49 76 05 27 76 ; N  
3D2C: 05 E7 75 01 C9 75 03 A8 75 01 87 75 01 8B 75 ; K  
3D3B: 05 47 75 01 27 75 01 29 75 ; E (part 1)  
3D44: 01 2B 75 01 07 75 01 09 75 01 0B 75 ; E (part 2)  
3D50: 03 C7 74 03 A9 74 03 87 74 ; Y
```

Wel nog een regelkleur aanpassen: regel #07 van #06 naar #03.

Ziet er nu goed uit:



De B staat echter op de plek van het lege blok dat gebruikt wordt naast Bowser op rivets. Daarom verplaatsen: nu sprite number #4A.

Aanpassen:

3478: 48 4A 01 6C ; B

Nu versie v0.108 opgeslagen.

Nog wel checken of de rest allemaal goed gaat.

Ja, dat gaat goed.

Nu de muziekjes gaan maken:

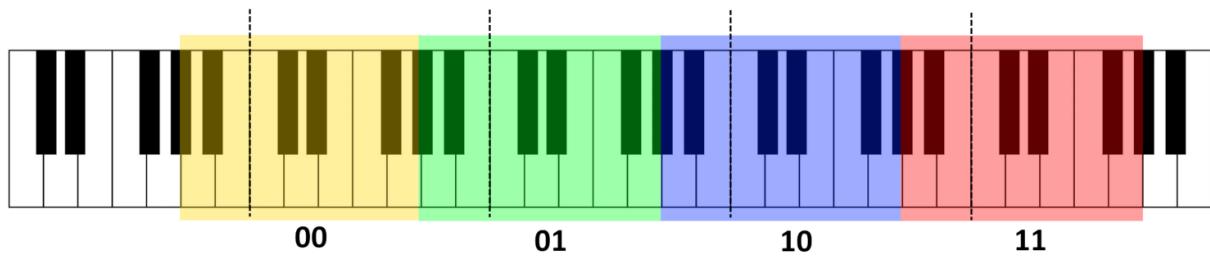
Uitleg van de patternopbouw:

```
; Format:  
; * If the value is $00, end the pattern.  
; * Else, if bit 7 is clear, the byte is the duration for all following notes until another duration code is read.  
; * Else, for channel A:  
;   * Bits 0-3 set the note (A, A#, B, C ...)  
;   * Bits 4-5 set the octave (higher values choose lower octaves)  
;   * Bit 6 indicates the next byte is a note to play on channel B.  
;  
; Channel B notes follow the same format, except bits 6-7 are ignored.  
;  
; This format means it is impossible to play a note on channel B without also playing one on channel A,  
; and the channels cannot have separate durations.
```

Notentabel:

0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011
A	A#	B	C	C#	D	D#	E	F	F#	G	G#
0	1	2	3	4	5	6	7	8	9	A	B

Octaven:



Eerst individueel maken op de plek van het How High muziekje. Dan kun je ze allemaal maken/testen en dan later in één keer de hele datastructuur samenstellen.

De sound zit in de s_3i_b.bin.

Het How High muziekje begint op #03EB.

Voorbeeld uitschrijven:

10 0001 0000 – duration voor de rest van de notes
E8 1110 1000 – Channel A – Octave 10 – Note 1000 - F
A4 1010 0100 – Channel B – Octave 10 – Note 0100 – C#
E8 1110 1000 – Channel A – Octave 10 – Note 1000 - F
00 0000 0000 - End

Beginnen met het onsterfelijkheidsmuziekje wanneer je een ster gepakt hebt:

"Invincible"

Super Mario Bros.

Composition by Koji Kondo

Arrangement by The Deku Trombonist

$\text{♩} = 150$

Piano

f

De D gaat niet goed. Dus transponeren naar als eerste het tweede deel:

14 – Duration

E7 – 1110 0111 – A – Octave 10 – E
83 – 1000 0011 – B – Octave 00 – C
A7 – 1010 0111 – A – Octave 10 – E
E7 – 1110 0111 – A – Octave 10 – E
8A – 1000 1010 – B – Octave 00 – G

0A – Duration

A3 – 1010 0011 – A – Octave 10 - C

14 – Duration

E7 – 1110 0111 – A – Octave 10 – E
93 – 1001 0011 – B – Octave 01 – C
A7 – 1010 0111 – A – Octave 10 – E

0A – Duration

A3 – 1010 0011 – A – Octave 10 – C
E7 – 1110 0111 – A – Octave 10 – E
8A – 1000 1010 – B – Octave 00 – G
A3 – 1010 0011 – A – Octave 10 – C

14 – Duration

E7 – 1110 0111 – A – Octave 10 – E
93 – 1001 0011 – B – Octave 01 – C

En het tweede deel 1 naar beneden

E6 – 1110 0110 – A – Octave 10 – D#
 82 – 1000 0010 – B – Octave 00 – B
 A6 – 1010 0110 – A – Octave 10 – D#
 E6 – 1110 0110 – A – Octave 10 – D#
 89 – 1000 1001 – B – Octave 00 – F#
0A – Duration
 A2 – 1010 0010 – A – Octave 10 – B
14 – Duration
 E6 – 1110 0110 – A – Octave 10 – D#
 92 – 1001 0010 – B – Octave 01 – B
 A6 – 1010 0110 – A – Octave 10 – D#
0A – Duration
 A2 – 1010 0010 – A – Octave 10 – B
 E6 – 1110 0110 – A – Octave 10 – D#
 89 – 1000 1001 – B – Octave 00 – F#
 A2 – 1010 0010 – A – Octave 10 – B
14 – Duration
 E6 – 1110 0110 – A – Octave 10 – D#
 92 – 1001 0010 – B – Octave 01 – B
00 – End

Ja, dat is nu goed.

Het hele stuk in hex dump om later in geheel op te nemen:

14 E7 83 A7 E7 8A 0A A3 14 E7 93 A7 0A A3 E7 8A A3 14 E7 93 E6 82 A6 E6 89 0A A2 14 E6 92 A6 0A
A2 E6 89 A2 14 E6 92 00

Nu het muziekje maken wanneer jumpman dood gaat:

"Lost a Life"

Super Mario Bros.

Composition by Koji Kondo

Arrangement by The Deku Trombonist

$\text{♩} = 100$

Piano

De D is een probleem dus hele melodie 1 naar beneden transponeren.

10 – Duration
 E1 – 1110 0001 – A – Octave 10 – A#
 89 – 1000 1001 – B – Octave 00 – F#

96 – 1001 0110 – A – Octave 01 – D#

87 – 1000 0111 – A – Octave 00 – E

60 - Duration

97 – 1001 0111 – A – Octave 01 – E

10 - Duration

89 – 1000 1010 – A – Octave 00 – F#

99 – 1001 1010 – A – Octave 01 – F#

86 – 1000 0110 – A – Octave 00 – D#

96 – 1001 0110 – A – Octave 01 – D#

87 – 1000 0111 – A – Octave 00 – E

60 – Duration

97 – 1001 0111 – A – Octave 01 – E

10 - Duration

83 – 1000 0011 – A – Octave 00 – C

93 – 1001 0011 – A – Octave 01 – C

80 – 1000 0000 – A – Octave 00 – A

90 – 1001 0000 – A – Octave 01 – A

81 – 1000 0001 – A – Octave 00 – A#

60 - Duration

91 – 1001 0001 – A – Octave 01 – A#

10 - Duration

83 – 1000 0011 – A – Octave 00 – C

93 – 1001 0011 – A – Octave 01 – C

80 – 1000 0000 – A – Octave 00 – A

90 – 1001 0000 – A – Octave 01 – A

81 – 1000 0001 – A – Octave 00 – A#

60 – Duration

91 – 1001 0001 – A – Octave 01 – A#

00 - End

Het hele stuk in hex dump om later in geheel op te nemen:

10 89 99 86 96 87 60 97 10 89 99 86 96 87 60 97 10 83 93 80 90 81 60 91 10 83 93 80 90 81 60 91 00

Nu een alternatieve background music Fortress:

"Fortress Theme"

Super Mario Bros. 3

Composed by Koji Kondo

Arranged by Latios212

$\text{♩} = 114$

Piano

Duration – 10

83 – 1000 0011 – A – Octave 00 – C

8A – 1000 1010 – A – Octave 00 – G

Duration – 60

89 – 1000 1001 – A – Octave 00 – F#

Duration – 10

88 – 1000 1000 – A – Octave 00 – F

87 – 1000 0111 – A – Octave 00 – E

86 – 1000 0110 – A – Octave 00 – D#

Duration – 60

82 – 1000 0010 – A – Octave 00 – B#

End - 00

Het hele stuk in hex dump om later in geheel op te nemen:

15 83 8A 7F 89 10 88 87 86 60 82 00

Dan nu einde level muziekje maken:

"Course Clear"

Super Mario Bros.

Composition by Koji Kondo

Arrangement by The Deku Trombonist

Piano

$\text{♩} = 150$

3

Bevat een D dus kijken naar getransponeerde versie. Eerst even uitschrijven als noten en dan kijken hoeveel schuiven zodat er geen D meer gebruikt wordt.

Duration – 10

91 – 1001 0001 – A – Octave 01 – A#
96 – 1001 0110 – A – Octave 01 – D#
9A – 1001 1010 – A – Octave 01 – G
E1 – 1110 0001 – A – Octave 02 – A#
86 – 1000 0110 – B – Octave 00 – D#
E6 – 1110 0110 – A – Octave 02 – D#
8A – 1000 1010 – B – Octave 00 – G
EA – 1110 1010 – A – Octave 02 – G
91 – 1001 0001 – B – Octave 01 – A#

Duration – 30

F1 – 1111 0001 – A – Octave 03 – A#
9A – 1001 1010 – B – Octave 01 – G
EA – 1110 1010 – A – Octave 02 – G
96 – 1001 0110 – B – Octave 01 – D#

Duration – 10

92 – 1001 0010 – A – Octave 01 – B
96 – 1001 0110 – A – Octave 01 – D#
99 – 1001 1001 – A – Octave 01 – F#
E2 – 1110 0010 – A – Octave 02 – B
86 – 1000 0110 – B – Octave 00 – D#
E6 – 1110 0110 – A – Octave 02 – D#
89 – 1000 1001 – B – Octave 00 – F#

E9 – 1110 1001 – A – Octave 02 – F#

92 – 1001 0010 – B – Octave 01 – B

Duration – 30

F2 – 1111 0010 – A – Octave 03 – B

99 – 1001 1001 – B – Octave 01 – F#

EA – 1110 1010 – A – Octave 02 – G

96 – 1001 0110 – B – Octave 01 – D#

Duration – 10

94 – 1001 0100 – A – Octave 01 – C#

98 – 1001 1000 – A – Octave 01 – F

9B – 1001 1011 – A – Octave 01 – G#

E4 – 1110 0100 – A – Octave 02 – C#

88 – 1000 1000 – B – Octave 00 – F

D8 – 1101 1000 – A – Octave 02 – F

8B – 1000 1011 – B – Octave 00 – G#

EB – 1110 1011 – A – Octave 02 – G#

94 – 1001 0100 – B – Octave 01 – C#

Duration – 30

F4 – 1111 0100 – A – Octave 03 – C#

99 – 1001 1001 – B – Octave 01 – F#

Duration – 10

F4 – 1111 0100 – A – Octave 03 – C#

98 – 1001 1000 – B – Octave 01 – F

F4 – 1111 0100 – A – Octave 03 – C#

98 – 1001 1000 – B – Octave 01 – F

F4 – 1111 0100 – A – Octave 03 – C#

98 – 1001 1000 – B – Octave 01 – F

Duration – 60

F6 – 1111 0110 – A – Octave 03 – D#

96 – 1001 0110 – B – Octave 01 – D#

End - 00

Het hele stuk in hex dump om later in geheel op te nemen:

0D 91 96 9A E1 86 E6 8A EA 91 27 F1 9A EA 96 0D 92 96 99 E2 86 E6 89 E9 92 27 F2 99 EA 96 0D 94
98 9B E4 88 D8 8B EB 94 27 F4 99 0D F4 98 F4 98 4E F6 96 00

Dan nu einde world (level) muziekje maken:

"World Clear"

Super Mario Bros.

Composition by Koji Kondo

Arrangement by The Deku Trombonist



Komt geen D in voor dus geen transpose nodig (toch wel aan einde, maar geen probleem)

Duration – 10

E3 – 1110 0011 – A – Octave 02 – C
83 – 1000 0011 – B – Octave 00 – C
9A – 1001 1010 – A – Octave 01 – G
97 – 1001 0111 – A – Octave 01 – E
E3 – 1110 0011 – A – Octave 02 – C
83 – 1000 0011 – B – Octave 00 – C
9A – 1001 10101 – A – Octave 01 – G
97 – 1001 0111 – A – Octave 01 – E

Duration – 30

D7 – 1101 0111 – A – Octave 01 – E
83 – 1000 0011 – B – Octave 00 – C
D7 – 1101 0111 – A – Octave 01 – E
83 – 1000 0011 – B – Octave 00 – C

Duration - 10

E4 – 1110 0100 – A – Octave 02 – C#
84 – 1000 0100 – B – Octave 00 – C#
9B – 1001 1011 – A – Octave 01 – G#
98 – 1001 1000 – A – Octave 01 – F
E4 – 1110 0100 – A – Octave 02 – C#
84 – 1000 0100 – B – Octave 00 – C#
9B – 1001 1011 – A – Octave 01 – G#
98 – 1001 1000 – A – Octave 01 – F

Duration – 30

D8 – 1101 1000 – A – Octave 01 – F
84 – 1000 0100 – B – Octave 00 – C#
D8 – 1101 1000 – A – Octave 01 – F
84 – 1000 0100 – B – Octave 00 – C#

Duration – 10

E6 – 1110 0110 – A – Octave 02 – D#
91 – 1001 0001 – B – Octave 01 – A#
E1 – 1110 0001 – A – Octave 02 – A#
8A – 1000 1010 – B – Octave 00 – G
DA – 1101 1010 – A – Octave 01 – G
86 – 1000 0110 – B – Octave 00 – D#
E6 – 1110 0110 – A – Octave 02 – D#
91 – 1001 0001 – B – Octave 01 – A#
E1 – 1110 0001 – A – Octave 02 – A#
8A – 1000 1010 – B – Octave 00 – G
DA – 1101 1010 – A – Octave 01 – G
86 – 1000 0110 – B – Octave 00 – D#

Duration – 30

E6 – 1110 0110 – A – Octave 02 – D#
91 – 1001 0001 – B – Octave 01 – A#

Duration – 10

D8 – 1110 1000 – A – Octave 02 – F
93 – 1001 0011 – B – Octave 01 – C
E8 – 1110 1000 – A – Octave 02 – F
93 – 1001 0011 – B – Octave 01 – C
E8 – 1110 1000 – A – Octave 02 – F
93 – 1001 0011 – B – Octave 01 – C

Duration – 60

AA – 1010 1010 – B – Octave 02 – G
92 – 1001 0010 – B – Octave 01 – B

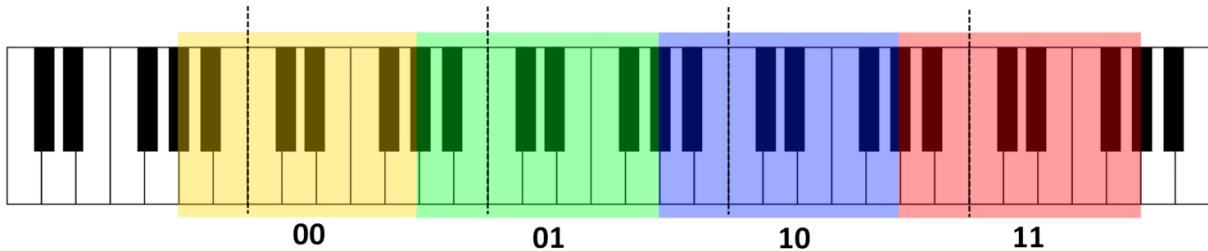
End - 00

Het hele stuk in hex dump om later in geheel op te nemen:

10 E3 83 9A 97 E3 83 9A 97 30 D7 83 D7 83 10 E4 84 9B 98 E4 84 9B 98 30 D8 84 D8 84 10 E6 91 E1
8A DA 86 E6 91 E1 8A DA 86 30 E6 91 10 D8 93 E8 93 E8 93 60 AA 92 00

Notentabel:

0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011
A	A#	B	C	C#	D	D#	E	F	F#	G	G#
0	1	2	3	4	5	6	7	8	9	A	B



"Invincible"

Super Mario Bros.

Composition by Koji Kondo

Arrangement by The Deku Trombonist

$\text{♩} = 150$

Piano

14 – Duration

E2 – 1110 0010 – A - Octave 10 – B

84 – 1000 0100 – B – Octave 00 – C#

A2 – 1010 0010 – A – Octave 10 – B

E2 – 1110 0010 – A – Octave 10 – B

8B – 1000 1011 – B – Octave 00 – G#

0A – Duration

94 – 1001 0100 – A – Octave 01 – C#

14 – Duration

E2 – 1110 0010 – A – Octave 10 – B

94 – 1001 0100 – B – Octave 01 – C#

A2 – 1010 0010 – A – Octave 10 – B

0A – Duration

94 – 1001 0100 – A – Octave 01 – C#

E2 – 1110 0010 – A – Octave 10 – B

8B – 1000 1011 – B – Octave 00 – G#

94 – 1001 0100 – A – Octave 01 – C#

14 – Duration

E2 – 1110 0010 – A – Octave 10 – B

94 – 1001 0100 – B – Octave 01 – C#

E1 – 1110 0001 – A - Octave 10 – A#

82 – 1000 0010 – B – Octave 00 – B

A1 – 1010 0001 – A – Octave 10 – A#

E1 – 1110 0001 – A – Octave 10 – A#

89 – 1000 1001 – B – Octave 00 – F#

0A – Duration
92 – 1001 0010 – A – Octave 01 – B
14 – Duration
E1 – 1110 0001 – A – Octave 10 – A#
92 – 1001 0010 – B – Octave 01 – B
A1 – 1010 0001 – A – Octave 10 – A#
0A – Duration
92 – 1001 0010 – A – Octave 01 – B
E1 – 1110 0001 – A – Octave 10 – A#
89 – 1000 1001 – B – Octave 00 – F#
92 – 1001 0010 – A – Octave 01 – B
14 – Duration
E1 – 1110 0001 – A – Octave 10 – A#
92 – 1001 0010 – B – Octave 01 – B

14 E2 84 A2 E2 8B 0A 94 14 E2 94 A2 0A 94 E2 8B 94 14 E2 94 E1 82 A1 E1 89 0A 92 14 E1 92 A1 0A
92 E1 89 92 14 E1 92 00

Dan nu de boel bijeenvoegen.

;
; Pattern #01 - Barrels board BGM – Fortress Theme
;
301: 15 83 8A 7F 89 10 88 87 86 60 82 00

;
; Pattern #02 - Pies board BGM – Original Pies
;
30D: 16 84 0B 84 84 00

;
; Pattern #03 – Elevators board BGM – Underground Theme
;
313: 10 89 99 86 96 87 60 97 10 89 99 86 96 87 60 97 10 83 93 80 90 81 60 91 10 83 93 80 90 81 60
91 00

;
; Pattern #04 – Rivets board BGM – Original Rivets
;
334: 18 84 0C 80 80 00

Ja, dat gaat goed voor barrels, pies en rivets. Voor elevators nog muziek activeren. Maar dat moet nog uitgezocht worden. De melodie zit er echter al wel in.

50A op 3 zetten is de oplossing? Ja ! Werkt nu wel.

;
; Pattern #05 – Running out of time - Original

```
;  
33A: 10 B8 B4 B0 00  
  
;  
; Pattern #06 – Hammer time – Invincibility Theme  
;  
33F: 14 E2 84 A2 E2 8B 0A 94 14 E2 94 A2 0A 94 E2 8B 94 14 E2 94 E1 82 A1 E1 89 0A 92 14 E1 92 A1  
0A 92 E1 89 92 14 E1 92 00
```

Deze gaat niet goed. Gaat nu al beter, maar geluid is anders dan verwacht?

```
;  
; Pattern #07 - Empty  
;  
367: 00  
  
;  
; Pattern #08 – Scored Points - Original  
;  
368: 09 B0 B2 B4 12 BA B6 00  
  
;  
; Pattern #09 – Death music – Lost a Live theme  
;  
370: 10 E1 89 20 A7 10 E7 89 15 E7 89 E6 8B E4 91 10 E2 92 DA 8A D7 87 D2 82 00  
  
;  
; Pattern #0A – DK Climbing ladders intro – Empty  
;  
389: 00  
  
;  
; Pattern #0B - Empty  
;  
38A: 00  
  
;  
; Pattern #0C: Unused cutscene - Empty  
;  
38B: 00  
  
;  
; Pattern #0D: Empty  
;  
38C: 00  
  
;  
; Pattern #0E: Empty
```

```

;
38D: 00

;

; Pattern #0F: Unused cutscene – Empty
;

38E: 00

;

; Pattern #10: How High Can You Get - Original
;

38F: 1B E2 88 09 A4 12 82 A8 12 88 A4 E2 82 A4 12 E0 80 09 82 83 86 88 8A 8C 24 D0 88 00

;

; Pattern #11: Rescued Pauline – Rivets ending (odd levels) – World Clear theme
;

3AB: 10 E3 83 9A 97 E3 83 9A 97 30 D7 83 D7 83 10 E4 84 9B 98 E4 84 9B 98 30 D8 84 D8 84 10 E6 91
E1 8A DA 86 E6 91 E1 8A DA 86 30 E6 91 10 D8 93 E8 93 E8 93 60 AA 92 00

```

Moet ook voor de even levels gekozen gaan worden: kan door code aan te passen.

```

;

; Pattern #12: Rescued Pauline – unused - empty
;

3E2: 00

;

; Pattern #13: Rescued Pauline – Rivets ending (even levels) – mappen op pattern 11 – Empty
;

3E3: 00

;

; Pattern #14: Non-rivets ending – Course Clear theme
;

3E4: 0D 91 96 9A E1 86 E6 8A EA 91 27 F1 9A EA 96 0D 92 96 99 E2 86 E6 89 E9 92 27 F2 99 EA 96 0D
94 98 9B E4 88 D8 8B EB 94 27 F4 99 0D F4 98 F4 98 F4 98 4E F6 96 00

;

; Pattern #15: DK is about to fall - Original
;

41A: 0A E0 98 E0 98 3C E0 98 12 88 84 88 84 88 84 88 84 88 84 00

```

430 – 4F7: allemaal aanvullen met FF

Is nu allemaal goed. De muziekjes zitten er in. Er is nog wel tuning qua timing e.d. nodig.

[Nu versie v0.109 opgeslagen.](#)

De onsterfelijkheidsmuziek gaat niet goed omdat de background muziek er doorheen gespeeld wordt. Kun je deze tijdelijk uitzetten en daarna weer aanzetten? Zit in origineel ook ergens (bij de originele hammer music?).

De muziek veilig stellen zit hier:

```
2FAE 3A8960 LD A,(#6089) ; load A with background music value  
2FB1 328963 LD (#6389),A ; save so it can be restored when hammer runs out.  
2FB4 C37C2F JP #2F7C ; return to program
```

De muziek weer terugzetten zit hier:

```
2F76 3A8963 LD A,(#6389) ; load A with previous background music  
2F79 328960 LD (#6089),A ; set music with what it was before the hammer was grabbed
```

Dat tussenvoegen bij de routines voor de afhandeling van de onsterfelijkheid.

Dat is het niet, want dat wordt al gezet (hammer grab, originele code) en weer teruggezet (einde invincibility, eigen code).

Even als test de originele hammer sound terugzetten:

```
12 90 09 90 90 12 90 90 94 90 94 90 94 09 94 94 12 94 94 98 94 98 94 00
```

Dat is pattern #6 die nu begint bij 33F. En de #367-einde opschuiven. Die gaat wel goed. Dus het ligt echt aan het pattern. Wat als we de originele over de gehele nieuwe heen kopiëren en dan langzaam van begin melodie gaan opbouwen?

Na terugzetten van de oude melodie over de nieuwe heen, gaat het goed.

```
14 E2 84 A2 E2 8B 0A 94 14 E2 94 A2 0A 94 E2 8B 94 14 E2 94 E1 82 A1 E1 89 0A 92 14 E1 92 A1 0A  
92 E1 89 92 14 E1 92 00
```

"Invincible"

Super Mario Bros.

Composition by Koji Kondo

Arrangement by The Deku Trombonist

$\text{♩} = 150$

Piano

The sheet music shows a piano arrangement of the 'Invincible' theme. The top staff is in treble clef and the bottom staff is in bass clef. Both staves are in 4/4 time. The dynamic is marked as forte (f). The music consists of two measures of eighth-note patterns followed by a repeat sign and another two measures of similar patterns.

De invincible erover heen gaat niet goed. Klinkt gewoon niet. Misschien een eenvoudiger muziekje maken ala het origineel maar wel met het gevoel van de invincible muziek van SMB.

C# en A# ritmisch herhalen en tweede deel C en A ritmisch herhalen:

14
C# 94 – 1001 0100 – X – Octave 01 – X
C# 94 – 1001 0100 – X – Octave 01 – X
C# 94 – 1001 0100 – X – Octave 01 – X
0A
A 90 – 1001 0001 – X – Octave 01 – X
14
C# 94 – 1001 0100 – X – Octave 01 – X
C# 94 – 1001 0100 – X – Octave 01 – X
0A
A 90 – 1001 0001 – X – Octave 01 – X
C# 94 – 1001 0100 – X – Octave 01 – X
A 90 – 1001 0001 – X – Octave 01 – X
14
C# 94 – 1001 0010 – X – Octave 01 – X

Totaal wordt dan:

14 94 94 94 0A 91 14 94 94 0A 91 94 91 14 94 93 93 93 0A 8B 14 93 93 0A 8B 93 8B 14 93 00

Iets hoger maken, +3 doen:

14 97 97 97 0A 94 14 97 97 0A 94 97 94 14 97 96 96 96 0A 92 14 96 96 0A 92 96 92 14 96 00

Het rare is dat als het invincible muziekje bij i.p.v. How High ingevoerd wordt, het wel gewoon goed klinkt.

Dat klopt. Tijdens het spelen, kun je alleen channel A gebruiken. Channel B is dan gereserveerd voor geluidseffecten.

14 94 94 94 0A 91 14 94 94 0A 91 94 91 14 94 93 93 93 0A 8B 14 93 93 0A 8B 93 8B 14 93 00

14 97 97 97 0A 94 14 97 97 0A 94 97 94 14 97 96 96 96 0A 92 14 96 96 0A 92 96 92 14 96 00

14 98 98 98 0A 95 14 98 98 0A 95 98 95 14 98 97 97 97 0A 93 14 97 97 0A 93 97 93 14 97 00

14 99 99 99 0A 96 14 99 99 0A 96 99 96 14 99 98 98 98 0A 94 14 98 98 0A 94 98 94 14 98 00

Voorlopig de geel gemarkeerde versie gaan gebruiken:

;
; Pattern #06 – Hammer time – Invincibility Theme
;
33F: 14 98 98 98 0A 95 14 98 98 0A 95 98 95 14 98 97 97 97 0A 93 14 97 97 0A 93 97 93 14 97 00

En dan de rest opschuiven terug naar 35D:

Pattern #07: 35D
Pattern #08: 35E
Pattern #09: 366
Pattern #0A: 37F
Pattern #0B: 380
Pattern #0C: 381
Pattern #0D: 382
Pattern #0E: 383
Pattern #0F: 384
Pattern #10: 385
Pattern #11: 3A1
Pattern #12: 3D8
Pattern #13: 3D9
Pattern #14: 3DA
Pattern #15: 410

Gaat niet goed. Gewoon het normale hamer geluid houden.

33F: 12 90 09 90 90 12 90 90 94 90 94 90 94 09 94 94 12 94 94 98 94 98 94 00

Aangepast. Dat is gewoon veel beter.

Nu versie v0.110 opgeslagen.

Geluid bij beëindigen barrels, pie en elevator boards aanpassen. Is nu de Course Clear melodie, maar die is veel te lang. Alleen het eerste stukje nemen en dan het Na-Na-Na-Na geluid afspelen.

Zit nu op:

3D4: 0D 91 96 9A E1 86 E6 8A EA 91 27 F1 9A EA 96 0D 92 96 99 E2 86 E6 89 E9 92 27 F2 99 EA 96 0D 94 98 9B E4 88 D8 8B EB 94 27 F4 99 0D F4 98 F4 98 F4 98 4E F6 96 00

Alleen eerste stuk behouden:

3D4: 0D 91 96 9A E1 86 E6 8A EA 91 27 F1 9A EA 96 00

En de rest erachter opschuiven.

Ja, dat gaat goed.

Nu versie v0.111 opgeslagen.

En dan nu kijken naar de melodie wanneer jumpman dood gaat. Tijdens het draaien een geluid en daarna pas de Death theme. Kijken of dat draai geluid ertussenuit kan.

Dat draaigeluid is een FX uit de sound chip. Doet eerst twee noten spelen en dan een aantal malen een geluid in een loop en na aantal keren wordt daaruit gesprongen naar het afspelen van de melodie. Misschien op te lossen door meteen naar de melodie te springen.

```
00F: 04 3C jmp $03C ; jump to melodic part
```

Ja, dat werkt. Tijdens het draaien al de melodie van Lost a Life.

Alleen de melodie is niet helemaal goed.

Een noot aangepast en de laatste paar noten langzamer:

```
10 E1 89 20 A7 10 E7 89 15 E7 89 E6 8B E4 92 18 E2 92 DA 8A D7 87 D2 82 00
```

Is nu goed.

Nu versie v0.112 opgeslagen.

Nu het How High Can You Get melodietje aan het einde langzamer maken:

```
1B E2 88 09 A4 12 82 A8 12 88 A4 E2 82 A4 12 E0 80 10 82 83 86 88 8A 8C 24 D0 88 00
```

En de tone naan het einde naar beneden afbuigen:

```
1B E2 88 09 A4 12 82 A8 12 88 A4 E2 82 A4 12 E0 80 10 82 83 84 86 88 8A 24 D0 84 00
```

Ja dat klinkt leuk. Alleen is iets langer dus loopt door nadat het How High scherm gewist wordt.

Die tijd iets langer maken.

Wellicht dat dat hier zit:

```
0C82 110703 LD DE,#0307 ; load task data for text #7 "HOW HIGH CAN YOU GET?"  
0C85 CD9F30 CALL #309F ; insert task to draw text  
0C88 210960 LD HL,WaitTimerMSB ; load HL with timer to wait  
0C8B 36A0 LD (HL),#A0 ; set timer for #A0 units  
0C8D 23 INC HL ; HL := GameMode2  
0C8E 34 INC (HL) ;  
0C8F 34 INC (HL) ; increase game mode twice - starts game  
0C90 C9 RET ; return
```

Aanpassen:

```
0C8B 36C0 LD (HL),#C0 ; set adjusted timer
```

Gaat goed.

Nu versie v0.113 opgeslagen.

De lava naar beneden verplaatsen. Nu op rivets nog dat bowser achter de lava langs valt en aan de onderkant nog zichtbaar is. Lijkt nu aan de onderkant uitgelijnd, maar dat is niet zo, doordat de blokjes juist niet helemaal tot onderaan zijn. De blokjes bestaan uit twee lege pixels aan de onderkant, dus de lava moet 2 pixels naar beneden verschoven worden.

Omdat we de lava in het pies board en het elevators board niet willen aanpassen, een nieuw lava blok maken: i.p.v. #19 is de nieuwe nu: #71.

De lava sprites in rivets worden gedefinieerd door dit data blok:

```
0B6F 57 19 02 F6  
0B73 67 19 02 F6  
0B77 77 19 02 F6  
0B7B 87 19 02 F6  
0B7F 97 19 02 F6  
0B83 A7 19 02 F6
```

Aanpassen:

```
0B6F 57 71 0F F8  
0B73 67 71 0F F8  
0B77 77 71 0F F8  
0B7B 87 71 0F F8  
0B7F 97 71 0F F8  
0B83 A7 71 0F F8
```

Dit gaat bij allemaal goed. Nog eventjes alle velden doorspelen om te kijken of de nu gekozen lege sprite om de lava in te definiëren niet toevallig ergens gebruikt wordt als onzichtbaar blok. Is helemaal goed.

Nu versie v0.114 opgeslagen.

De ending van rivets aanpassen. Het geluid van het weghalen van de girders loopt net te lang door. Daarna valt bowser vrij langzaam: versnellen. Het valgeluid aanpassen en daarna een keuze maken voor een eind melodie.

Eerst kijken hoe het geratel bij het weghalen van de girders korter gemaakt kan worden.

Lijkt pattern 15 te zijn:

```
;  
; Pattern #15: DK is about to fall - Original  
;  
3E4: 0A E0 98 E0 98 3C E0 98 12 88 84 88 84 88 84 88 84 88 84 84 00
```

Even kijken door langzamer te maken of dat inderdaad de juiste is. Ja dat is het zeker.

Korter maken kan door sneller af te spelen of door noten te verwijderen. Sneller klinkt nogal raar. Daarom korter maken. Hoeft niets op te schuiven, want dit is het laatste pattern.

Mooiste klinkt het wanneer er 4 afgehaald worden:

```
3E4: 0A E0 98 E0 98 3C E0 98 12 88 84 88 84 88 84 88 84 00
```

Dat gaat nu goed.

Bowser sneller laten vallen.

Dat lijkt hier te zitten:

```
1880 210B69 LD HL,#690B ; load HL with kong start sprite  
1883 0E01 LD C,#01 ; load C with 1 pixel to move  
1885 FF RST #38 ; move kong  
1886 3A1B69 LD A,(#691B) ; load A with ???  
1889 FED0 CP #D0 ; == #D0 ?  
188B C0 RET NZ ; no, return
```

Twee keer zo snel vallen, door met twee pixels te verplaatsen?

Ja, dat is eigenlijk veel mooier. Dan sluit het ook aan het valgeluid. Ongeveer net zo lang.

Alleen de sprites worden niet gewist, waardoor de sprites bovenaan het scherm zichtbaar worden.

Dat wordt hier gedaan:

```
0BC2 FEF8 CP #F8 ; is bottom of screen reached?  
0BC4 2006 JR NZ,#BBBB ; no, jump forward - BBBB
```

Aanpassen:

Misschien nog iets lager en niet op één waarde, maar wanneer boven een waarde:

```
0BC2 FEF8 CP #F8 ; is bottom of screen reached?  
0BC4 3806 JR C,#BBBB ; no, jump forward - BBBB
```

Ja, dat is nu goed.

Nu versie v0.115 opgeslagen.

Nu het melodietje aan het einde van het level aanpassen. Misschien Course Clear theme i.p.v. World Clear Theme?

```
;  
; Pattern #11: Rescued Pauline – Rivets ending (odd levels) – World Clear theme  
;  
39B: 10 E3 83 9A 97 E3 83 9A 97 30 D7 83 D7 83 10 E4 84 9B 98 E4 84 9B 98 30 D8 84 D8 84 10 E6 91  
E1 8A DA 86 E6 91 E1 8A DA 86 30 E6 91 10 D8 93 E8 93 E8 93 60 AA 92 00  
  
;  
; Pattern #14: Non-rivets ending – Course Clear theme  
;  
3D4: 0D 91 96 9A E1 86 E6 8A EA 91 27 F1 9A EA 96 0D 92 96 99 E2 86 E6 89 E9 92 27 F2 99 EA 96 0D  
94 98 9B E4 88 D8 8B EB 94 27 F4 99 0D F4 98 F4 98 F4 98 4E F6 96 00
```

Pattern 14 is korter gemaakt zodat alleen het eerste stuk te horen is. Leuk als bij rivets clear het hele melodietje gespeeld wordt. En dan kun je eventueel bij einde rivets world 8 (game completed) alsnog een ander melodietje spelen.

```
;  
; Pattern #11: Rescued Pauline – Rivets ending (odd levels) – World Clear theme  
;  
39B: 0D 91 96 9A E1 86 E6 8A EA 91 27 F1 9A EA 96 0D 92 96 99 E2 86 E6 89 E9 92 27 F2 99 EA 96 0D  
94 98 9B E4 88 D8 8B EB 94 27 F4 99 0D F4 98 F4 98 F4 98 4E F6 96 96 00
```

En dan de rest één terugschuiven omdat deze melodie 1 korter is dan de eerdere.

Kijken of het overal nog goed gaat door een world door te spelen.

[Nu versie v0.116 opgeslagen.](#)

Nu nog in even velden geen einde melodietje. De conditie hiervoor zodanig aanpassen dat een alternatief melodietje alleen gespeeld wordt bij wereld 8. Dat is namelijk game completed. En dan daar een interessante melodie voor maken.

De conditie zit hier:

```
1913 218A60 LD HL,#608A ; load HL with sound address  
1916 360C LD (HL),#0C ; play sound for rivets cleared  
1918 3A2962 LD A,(#6229) ; load A with level #  
191B 0F RRCA ; roll a right . is this an odd level ?  
191C 3802 JR c,#1920 ; Yes, skip next step  
  
191E 3605 LD (HL),#05 ; else play sound for even numbered rivets  
  
1920 23 INC HL ; HL := #608B = sound duration  
1921 3603 LD (HL),#03 ; set duration to 3
```

Aanpassen:

191B C3D43C JP #3CD4 ; jump to additional code - **AAAA**

en:

3CD4 FE08 CP #08 ; is this world 8? – **AAAA**
3CD6 C22019 JP NZ,#1920 ; no, use standard melody, jump back
;
3CD9 3605 LD (HL),#05 ; use game completed melody
3CDB C32019 JP #1920 ; jump back

Testen met rivets op verschillende levels.

Hiermee moeten alle even en oneven werelden het standaard melodietje gebruiken. En wereld 8 een nu nog leeg melodietje gebruiken. Dat klopt.

Nu versie v0.117 opgeslagen.

En dan nu een melodietje voor het einde maken.

"Ending"
Super Mario Bros.

Composition by Koji Kondo
Arrangement by The Deku Trombonist

♩ = 150

Piano



Deze gebruikt ook een D. Dus transponeren: 1 omhoog (kan zolang er in het origineel geen C# zit en dat is zo).

05 – Duration

G	Channel A	G#	Octave 01	1101-1011	DB
B	Channel B	C	Octave 00	1000-0011	83
F#	Channel A	G	Octave 01	1001-1010	9A
G	Channel A	G#	Octave 01	1001-1011	9B

20 - Duration

E	Channel A	F	Octave 01	1101-1000	D8
C	Channel B	C#	Octave 00	1000-0010	84

10 - Duration

F	Channel A	F#	Octave 01	1101-1001	D9
D	Channel B	D#	Octave 00	1000-0110	86
F#	Channel A	G	Octave 01	1101-1010	DA
D#	Channel B	E	Octave 00	1000-0111	87
G	Channel A	G#	Octave 01	1101-1011	DB
E	Channel B	F	Octave 00	1000-1000	88
C	Channel A	C#	Octave 00	1000-0100	84
C	Channel A	C#	Octave 02	1110-0100	E4
A	Channel B	A#	Octave 01	1001-0001	91
0A - Duration					
E	Channel A	F	Octave 02	1110-1000	E8
G	Channel B	G#	Octave 00	1000-1011	8B
05 - Duration					
E	Channel A	F	Octave 02	1010-1000	A8

10 - Duration

D	Channel A	D#	Octave 02	1110-0110	E6
F	Channel B	F#	Octave 00	1000-1001	89
E	Channel A	F	Octave 02	1110-1000	E8
E	Channel B	F	Octave 00	1000-1000	88
F	Channel A	F#	Octave 02	1110-1001	E9
D	Channel B	D#	Octave 00	1000-0110	86
B	Channel A	C	Octave 02	1101-0011	D3
F	Channel B	F#	Octave 00	1000-1001	89
D	Channel A	D#	Octave 02	1110-0110	E6
G	Channel B	G#	Octave 01	1001-1011	9B
G	Channel A	G#	Octave 00	1000-1011	8B
30 - Duration					
C	Channel A	C#	Octave 02	1110-0100	E4
C	Channel B	C#	Octave 00	1000-0100	84

Even op de plaats van How High melodie plaatsen op te testen.

37F: 14 DB 83 9A 9B D8 84 D9 86 DA 87 DB 88 84 E4 91 E8 8B A8 E6 89 E8 88 E9 86 D3 89 E6 9B 8B
E4 84 00

En dan de durations gaan toevoegen.

37F: 0C DB 83 9A 9B 48 D8 84 24 D9 86 DA 87 DB 88 84 E4 91 18 E8 8B 0C A8 24 E6 89 E8 88 E9 86
D3 89 E6 9B 8B 6C E3 83 00

Is nu goed.

Dan nu gaan tussenvoegen.

Verwijst naar melodie #05. Dat mapt op regel 505 en die verwijst naar #12. Dat mapt op regel 512 en die verwijst naar pattern #13.

Pattern 13 is bij mij nog leeg. Die moet gevuld worden.

Pattern 14 begint met 0D 91 96 9A E1 86 E6 8A EA. Die begint op #3D3.

Dit kopieren tot de reeks FF'en.

0D 91 96 9A E1 86 E6 8A EA 91 27 F1 9A EA 96 00 0A E0 98 E0 98 3C E0 98 12 88 84 88 84 88
84 00 FF FF

Pattern 13 is nu 00 en begint dus op #3D2. Daar de nieuwe melodie naar kopiëren.

En daar de het restant tot de FF achter plaatsen.

Ja, dat geeft bij wereld 8 het juiste melodietje.

Dan ook nog even testen met wereld 7.

Dat gaat ook goed.

Nu versie v0.118 opgeslagen.

Nu kijken of bij het verplaatsen van de bijl ook een sparkling sprite gestart kan worden. Dat zou heel mooi zijn. Nu wordt op dat moment de bijl verplaatst zodat deze zichtbaar wordt. Je zou op dat moment ook een geheugenplaats kunnen vullen en in een extra routine vanuit de main, daarop reageren en de animatie starten.

Wanneer alle drie de muntjes gepakt zijn, dan wordt geheugenplaats #64B0 op #F0 gezet. Dat kan gebruikt worden om het knipperen te doen.

Als extra routine vanuit de main aanroepen: op #197D is nog een slot vrij.

Aanpassen:

197D CDDD16 CALL #16DD ; process blinking after displaying axx

en:

```
16DD 3A2762        LD A,(#6227) ; load A with screen number
16E0 FE04           CP #04      ; is this the rivets?
16E2 C0             RET NZ     ; if not return
;
16E3 3AB064        LD A,(#64B0) ; load A with bonus-item pickup counter
16E6 FE00           CP #00      ; all bonus-items picked up?
16E8 C8             RET Z      ; no, return
;
```

16E9	FEF0	CP	#F0	; all bonus-items picked up?
16EB	CA5D0C	JP	Z,#0C5D	; yes, start blinking – label AAAA
;				
16EE	FE10	CP	#10	; finished blinking?
16F0	C8	RET	Z	; yes, return
;				
16F1	3D	DEC	A	; decrement A
16F2	32B064	LD	(#64B0),A	; save back in bonus-item pickup counter
16F5	E610	AND	#10	; bit 4 set?
16F7	3E63	LD	A,#63	; set sparkle sprite
16F9	2802	JR	Z,XX	; skip ahead
16FB	3E64	LD	A,#64	; reset sparkle sprite
16FD	32216A	LD	(#6A21),A	; set sparkle sprite number
1700	C9	RET		; return

en:

0C5D	3D	DEC	A	; decrement A – label AAAA
0C5E	32B064	LD	(#64B0),A	; save back in bonus-item pickup counter
;				
0C61	21206A	LD	HL,#6A20	; load HL with heart sprite
0C64	36B7	LD	(HL),#B7	; set heart sprite x-position
0C66	23	INC	HL	; next
0C67	3663	LD	(HL),#63	; set sparkle sprite
0C69	23	INC	HL	; next
0C6A	360B	LD	(HL),#0B	; set heart sprite color
0C6C	23	INC	HL	; next
0C6D	36EC	LD	HL,#EC	; set heart sprite y-position
0C6F	C9	RET		

Is goed. Alleen de sparkle sprite zit achter de bijl.

Heart sprite = #6A20 en bijl sprite = #6A58. Heart sprite is lager dus die wordt eerder getekend.
Kijken naar gebruikte sprites in rivets veld en dan een sprite slot kiezen dat hoger is dan #6A58. Slot #6A5C lijkt vrij te zijn.

Aanpassen:

0C61	215C6A	LD	HL,#6A5C	; load HL with heart sprite
------	--------	----	----------	-----------------------------

en:

16FD	325D6A	LD	(#6A5D),A	; set sparkle sprite number
------	--------	----	-----------	-----------------------------

Ja, dat gaat goed.

Nu versie v0.119 opgeslagen.

Er is niet veel ruimte meer beschikbaar. Die bewaren voor oplossen van eventuele problemen die bij het testen mogelijk nog naar voren komen.

Testen spelen wereld acht en uitspelen met verschillende scores en juiste weergave in de score tabel.

De kleur van de sparkle sprite is in elk board anders. Standaard maken:

3F94 360F LD (HL),#0F ; set heart sprite color

Checken of de bonus time wel opgeteld wordt bij game completed. Nee dat wordt niet gedaan.
Kijken of dit alsnog toegevoegd kan worden op het punt van het tonen van GAME COMPLETED.

Lijkt zo bijgewerkt te kunnen worden:

```
1955 110005 LD DE,#0500 ; load task #5, parameter 0 ; adds bonus to player's score
1958 CD9F30 CALL #309F ; insert task
```

GAME COMPLETED wordt hier gezet:

```
2F53 3A2962 LD A,(#6229) ; load A with world
2F56 FE08 CP #08 ; is this world 8?
2F58 C0 RET NZ ; no, return
;
2F59 3E01 LD A,#01 ; load A with #01
2F5B 323062 LD (#6230),A ; set game completed indication
2F5E 110603 LD DE,#0306 ; load DE with task code "GAME COMPLETED"
2F61 CD9F30 CALL #309F ; insert task
;
2F64 C9 RET ; return
```

Aanpassen:

```
2F61 C34428 JP #2844 ; jump to additional code
2F64 00 NOP
```

en:

```
2844 CD9F30 CALL #309F ; insert task
2847 110005 LD DE,#0500 ; load task #5, parameter 0 ; adds bonus to score
284A CD9F30 CALL #309F ; insert task
284D C9 RET ; return
```

Ja, dat gaat nu wel goed.

Nu versie v0.120 opgeslagen.

Andere testen:

Checken time out geluid en hele stapel schilden weg op barrels board. – gaat goed.

Noteren speelduur verschillende boarden verschillende werelden en speelbaarheid.

	World 1	World 2	World 3	World 4	World 5
Barrels	5000	6000	7000	8000	8000
Pies	5000	6000	7000	8000	8000
Elevators	5000	6000	7000	8000	8000
Rivets	5000	6000	7000	8000	8000

Bij de elevators nog wel eens in de problemen aan de linkerkant. Wellicht een idee om aan de linkerkant de twee trappen beneden en de ene trap boven om te draaien zodat er beneden één trap is en boven twee trappen? Dan kun je wellicht een opening zoeken door rondjes te cirkelen.

Dit zijn de ladders aan de linkerkant:

3BEA: 00 23 88 23 C0 ; long ladder on left side

3BEF: 00 1B C0 1B F8 ; bottom left ladder

3C03: 00 2B C0 2B F8 ; additional bottom right ladder

Y-waarden van de eerste omdraaien met die van de tweede en derde:

3BEA: 00 23 C0 23 F8 ; long ladder on left side

3BEF: 00 1B 88 1B C0 ; bottom left ladder

3C03: 00 2B 88 2B C0 ; additional bottom right ladder

Ja, dat ziet er goed uit en werkt prima. Iets meer kans om de koopa's te ontwijken. Zo houden.

Starten met world 1 en boardvolgorde 1 – 2 – 3 – 4.

Versie v0.121 opgeslagen.

Dat wordt de finale versie.

Versie v1.00 opgeslagen.

Embarrassment list tekst aanpassen:

36E6: 29 77 15 1D 12 11 22 22 11 23 23 1D 15 1E 24 10 1C 19 23 24 3F