

Donkey Kong – Barrel Party

Doelstelling is een groot aantal gevarieerde barrel velden.

Crazy Kong deel 1

Het pies veld aanpassen naar een Crazy Kong veld.

Eerst spel spelen met pies op level 5.

Aanpassen start level:

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

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

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

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

En pies instellen als eerste veld van level 5.

Dan definitie van het veld aanpassen naar definitie van barrels wat daarna aangepast kan worden naar kleinere girders.

Oude definities van pies (#3B5D-3BE4), elevators (#3BE5-3C8A)) and rivets (#3C8B-3CEF) alvast verwijderen.

De definitie van barrels erin plaatsen: #3AE4-3B5C naar #3B5D kopiëren.

Het veld wordt getoond, maar verder ook nog alle elementen van het oude pies veld.



- Jumpman loopt niet mee omhoog de schuine girder op.
- De conveyor belts worden nog getoond.
- De pies worden getoond en bewegen door het scherm.
- De retractable ladders worden getoond en bewegen.
- De items worden weergegeven en kunnen gepakt worden.
- De oil can staat nog bovenin en spawned fireballs.
- Donkey Kong beweegt ook alsof hij op de conveyor staat.

Eerst dit eens allemaal uit zetten, dan zorgen dat de trappen bekomen kunnen worden en dat de elementen van barrels inbrengen:

- Juiste palette van barrels
- Oil can beneden plaatsen (eigenlijk vanaf nu altijd doen).
- Donkey Kong bovenin plaatsen en stil laten staan.
- Donkey Kong barrels laten gooien.
- Jumpman schuin de barrels op laten lopen.

Todat het hele veld als een normaal barrels veld werkt.

Dan pas gaan werken aan het aanpassen van de girders.

Heel veel uit de main is niet meer benodigd. Beetje bij beetje weghalen:

```
1995 CDDB2D     CALL      #2DDB          ; deploy fireball/firefoxes for conveyors and rivets
```

Is niet meer benodigd. Eerst de call verwijderen (die komt daarmee vrij): 1995: 3x NOP.

Huh? Er worden nog steeds fireballs gespawned.

Eerst iets anders aanpassen:

```
199B CD0722     CALL      #2207          ; do stuff for conveyors
```

Deze kan verwijderd worden. Heeft te maken met de retractable ladders en de pies. Code loopt van #2207 t/m #22CA. Deze kan vrijgegeven worden.

Dit werkt. De retractable ladders bewegen niet meer en er schuiven ook geen pies meer langs.

Dan het volgende aanpassen:

```
19AA CDF225     CALL      #25F2          ; handle conveyor directions, adjust Mario's speed
```

Deze kan verwijderd worden. Heeft te maken met de conveyors. Code loopt van #25F2 t/m 26F9 en van #2AD3 t/m #2B1B. Deze kan vrijgegeven worden.

Dit werkt. Donkey Kong staat nu stil (maar nog wel te laag) en de conveyor pulleys draaien niet meer.

Dit stukje code lijkt ervoor te zorgen dat Donkey Kong bij girders een stukje omhoog geschoven wordt:

```
0D7A 3A2762 LD A,(#6227) ; load a with screen number
0D7D fe04 CP #04 ; is this rivets screen?
0D7F 280A JR Z,#0D8b ; if yes, jump ahead a bit

0D81 0F RRCA ; no, roll right twice
0D82 0F RRCA ; is this the conveyors or the elevators ?
0D83 d8 RET c ; yes, return

; else this is girders, kong needs to be moved

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

Dat kan verklaren waarom in het pies veld Donkey Kong lager staat. Aangezien alle velden barrels worden, moeten beide testen (rivets en conveyors/elevators) verwijderd worden. Altijd het stukje moven. Dat kan door #0D7A-0D83 in NOP's te veranderen.

Dit werkt bij het begin tekenen van Donkey Kong. Alleen tijdens het op de borst slaan, schuift de boel weer naar beneden. Dus waarschijnlijk ook nog ergens ander gereeld. Maar hele animatie van barrel deployment moet ook nog. Dus later naar kijken.

```
0CD1 C3A03F JP #3FA0 ; fix retractable ladders for pie factory and
returns to #0D5F.
```

Dit fixt de retractable ladders, maar dat is niet meer nodig. Kan vervangen worden door JP #0D5F. Dus wordt dan C35F0D.

De conveyor uiteinden (pulleys) niet meer tekenen. Onderdeel van het stuk code dat de setup doet voor conveyors. Dat is de code van #101F t/m #1086. Wat gebeurt er als we dat niet uitvoeren? Dus op #101F een RET plaatsen?

Dat werkt heel goed. Geen pulleys meer, geen oilcan meer bovenin scherm, geen retractable ladders meer. Alleen uit de (niet getoonde oilcan) komen nog wel fireballs tevoorschijn, maar met de verkeerde sprite waarde: een jumpman met verkeerde kleurpalette.

Eigenlijk moet in plaats van deze setup altijd de barrels setup uitgevoerd worden. In ieder geval kan dus het stuk code van #101F t/m #1086 dan vrijgegeven worden.

De #101F wordt aangeroepen vanuit hier:

```

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

; jump table data

0FCD  00 00          ; unused
0FCF  D7 0F          ; #0FD7 for girders
0FD1  1F 10          ; #101F for conveyors
0FD3  87 10          ; #1087 for elevators
0FD5  31 11          ; #1131 for rivets

```

Dus dat hoeft eigenlijk niet meer. In principe altijd girders setup aanroepen en eventueel later daar nog aparte setups achteraan die bijvoorbeeld additionele dingen regelen. Dus vooralsnog als volgt aanpassen: #0FCB t/m #0FD5 aanpassen naar NOP's. Daarna loopt het door in het stuk dat de girder setup doet waaruit met een RET gesprongen wordt.

In ieder geval is de stapel met barrels bij Donkey Kong nu aanwezig, staat de oilcan op de juiste plaats en worden de hamers getoond. Dat laatste moet wellicht nog verwijderd worden en in een speciale setup ondergebracht worden. Maar nu kan alle code voor conveyors, elevators en rivets vrijgegeven worden. Dat kan later in een soort van jumptable per veld een specifieke setup doen (indien nodig).

Dus vrijgeven: #101F t/m #1086 (conveyors), #1087 t/m #112D (elevators) en #1131 t/m 117D (rivets).

Eventjes kijken waar het palette en de muziek voor de verschillende screens gezet wordt.

De keuze wordt hier geregeld:

```

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

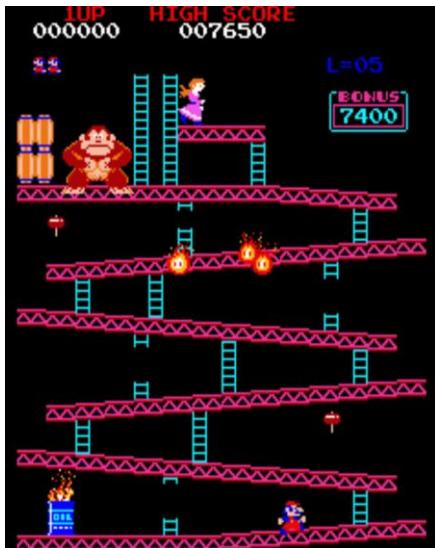
0CB2  3D       DEC    A            ; if not decrease a again
0CB3  CAF20C   JP     Z,#0CF2      ; if zero jump to #0CF2 - we were on elevators - continue on #0CC6

; else we are on rivets

```

Altijd nu de setup voor girders doen. Dus dat is de code op #0CD4. Dat kan door de conditionele JP Z simpel in een JP te veranderen: #0CAB wordt dan C3D40C. Later kan hier eventueel nog wel weer een keuze gemaakt worden voor het opzetten van bijvoorbeeld het palette voor Into The Dark veld.

Het veld ziet er nu zo uit:



Wat nog aangepast moet worden:

- Donkey Kong moet naar boven geplaatst worden.
- Fireballs spawnen op de barrels manier: door een blauwe barrel die de oil can bereikt.
- Jumpman moet schuin met de girders meelopen.

Dus eigenlijk door de code lopen en overal waar keuze op basis van het screen zitten, deze grotendeels verwijderen: altijd girders logica uitvoeren.

Barrel deployment zit hier:

```
; called from main routine at #1989

2C03 3E01      LD      A,#01          ; \ Return if screen is not barrels
2C05 F7        RST     #30          ; /
2C06 D7        RST     #10          ; Return if Mario is not alive

2C07 3A9363    LD      A,(#6393)      ; \ Return if we are already in the process of deploying a barrel, no need to deploy another one
2C0A 0F        RRCA           ; |
2C0B D8        RET      c          ; /
```

Barrel deployment check altijd doen. Dus 2C05 check niet uitvoeren: NOP maken.

Check conveyors:

```
03FB 3A2762    LD      A,(#6227)      ; load A with screen number
03FE FE02      CP      #02          ; are we on the conveyors?
0400 C21304    JP      NZ,#0413      ; no, skip ahead
```

Zorgt voor specifieke uitvoer voor conveyors. Is niet nodig. De conditionele JP Z in #0400 aanpassen in een gewone JP: #0400 C31304.

De volgende code doet iets met Kong die op zijn borst slaat en verschuiven van 4 posities.

```

; 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

0450 3A2762 LD A,(#6227) ; load A with screen number
0453 0F RRCA ; is this the girders or the elevators ?
0454 D27804 JP NC,#0478 ; no, skip ahead

0457 0F RRCA ; else is this the rivets ?
0458 DA8604 JP C,#0486 ; yes, skip ahead

; else pie factory

045B 210B69 LD HL,#690B ; load HL with start of Kong sprite data
045E 0EFC LD C,#FC ; C := #FC. used in sub below to move kong by -4
0460 FF RST #38 ; move kong
0461 C38604 JP #0486 ; skip ahead

```

Maar volgens mij klopt het commentaar niet. Volgens mij voor girders moet hij altijd het stuk code van #045B t/m #0463 uitvoeren. Dus dat kan afgedwongen worden door de check in #0450 te vervangen door een JP naar #045B: #0450 C35B04. Nee dat klopt niet. Nu springt Donkey Kong tijdens het slaan op borst naar het midden van het scherm. Teruggedraaid.

Maar misschien is de LD A,(#6227) nog wel relevant. Daarom de conditionele JP NC in #0454 aanpassen naar een JP maar #0458: #0454 C35B04. Dit werkt wel goed. Nu blijft Donkey Kong op de goede hoogte staan wanneer hij op zijn borst slaat.

Dan nu nog kijken waar jumpman's positie aangepast wordt aan de schuine girders. Dat lijkt hier te gebeuren:

```

1Cd8 3A2762 LD A,(#6227) ; load A with screen number
1Cdb 3D DEC a ; are we on the girders?
1Cdc c2Eb1C JP NZ,#1Ceb ; no, skip ahead

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

1CEB 210F62 LD HL,#620F ; load HL with address of movement indicator
1CEE 35 DEC (HL) ; decrease movement indicator
1CEF C3A61D JP #1DA6 ; jump ahead to update mario sprite and RET

```

Het stuk code van #1CDF t/m #1CEA altijd uitvoeren. Daartoe de check ervoor eruit halen. Dat kan door #1CD8 t/m #1CDE in NOP's te veranderen.

Dat werkt nu goed. Jumpman's Y-positie wordt aangepast aan de schuine girders. Hij kan ook de ladders opgaan en kan de hamers pakken. Het enige dat nog niet goed gaat is dat de kleine ladder rechts bovenin niet beklimmen kan worden. Heeft waarschijnlijk nog met logica van het pies scherm te maken: retractable ladders alleen te beklimmen als uitgeschoven.

Nu versie 0.03 opgeslagen.

De barrels worden niet gedeployed en de fireballs worden nog steeds in het midden gespawned. Dit heeft toch nog met screen checks te maken.

```
19BF CDCB2F     CALL    #2FCB           ; for non-girder levels, ....
```

Deze is nu niet meer nodig. Dus deze call vrijgeven. En die komt dan vrij voor eventuele andere calls.

#19BF: 3x NOP

Dat werkt. De fireballs worden niet meer midden in het scherm gereleased. Ook kan jumpman nu ineens wel de ladder rechts bovenin beklimmen. Het veld is echter klaar als die trap beklimmen is. Dat komt omdat in pies jumpman het veld al finished als hij op de girder van Kong komt en niet pas wanneer hij op de girder van Pauline komt.

Deze end of level code staat van #1E57 t/m #1E8B. Deze code aanpassen naar het volgende:

```
1E57 3A0562 LD A, (#6205)      ; load A with y position of mario
1E5A FE31 CP #31                ; are we on top level (rescued girl?)
1E5C D0 RET NC                 ; no, return
;
1E5D C36D1E JP #1E6D          ; else level has been finished.
```

En #1E60 t/m #1E6C NOP's maken.

Dat gaat nu goed. Jumpman kan op Kongs girder lopen en het veld eindigt pas wanneer hij op de girder met Pauline komt. Er is nog wel iets geeks. Jumpman kan helemaal naar links doorlopen. Gaat niet dood door aanraken Kong. Moet stoppen bij de twee trappen naar boven.

Nu versie 0.04 opgeslagen.

De volgende zaken zijn nog niet goed:

- De barrels worden nog niet gegooid.
- Jumpman kan voorbij de twee lange ladders naar Donkey Kong komen.
- Bij einde level wordt Donkey Kong weer te laag weergegeven.
- Bij einde level blijft het spel hangen.

Vanuit de main game routine zijn er drie acties die schijnbaar iets doen met barrel roll/deployment:

```

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

```

In de eerste call wordt iets gedaan met het laten rollen van barrels:

```

; called from main routine at #1983
; used to roll barrels

1F72 3A2762    LD      A,(#6227)      ; load a with screen number
1F75 3D        DEC     a              ; is this the girders ?
1F76 C0        RET     NZ            ; no, return

; yes, we are on girders
; this subroutine checks the barrels, if any are rolling it does something, otherwise returns

```

De check op het screen number moet hier dus verwijderd worden. Dat kan door de RET NZ te vervangen door een NOP, dan wordt er niet meer weggesprongen: #1F76: NOP.

Helpt nog niet. Worden nog steeds geen barrels gerold.

En in het tweede stuk:

```

; called from main routine #1986

2C8F 3E01    LD      A,#01          ; A := 1 = code for girders
2C91 F7        RST     #30          ; if screen is girders, continue. else RET
2C92 D7        RST     #10          ; if mario is alive, continue. else RET
2C93 3A9363    LD      A,(#6393)      ; load A with barrel deployment indicator
2C96 0F        RRCA           ; is a barrel being deployed ?
2C97 DA152D    JP      C,#2D15        ; yes, skip ahead

```

De check op het screen number moet hier dus ook verwijderd worden. Dat kan door de RST #30 te vervangen door een NOP, dan wordt er niet meer weggesprongen: #2C91: NOP.

Er worden nu wel barrels gerold. Alleen ga je er niet door af en kun je ze niet kapotslaan. Dat is vreemd. Ook worden de fireballs nog op de verkeerde plaats gespawned (bovenin het veld). Daarnaast valt op de jumpman niet goed kan springen op de plekken waar de girders dicht bij elkaar zitten. Lijkt erop dat hij zijn hoofd stoot.

En in het derde stuk:

```

; called from main routine at #1989

2C03 3E01 LD A,#01 ; \ Return if screen is not barrels
2C05 F7 RST #30 ; /
2C06 D7 RST #10 ; Return if Mario is not alive

2C07 3A9363 LD A,(#6393) ; \ Return if we are already in the process of deploying a barrel, no need to deploy another one
2C0A 0F RRCA ; |
2C0B D8 RET c ; /

```

De check op het screen number moet hier dus ook verwijderd worden. Dat kan door de RST #30 te vervangen door een NOP, dan wordt er niet meer weggesprongen: #2C05: NOP. Dat is al gedaan. In een eerder stuk.

Toch nog eens kijken naar alle screen # testen: #6227 en RST #30.

Eerst de RST #30's nalopen:

```

03A2 3E03 LD A,#03 ; A := 3 = 0011 binary
03A4 F7 RST #30 ; only continue if level is girders or conveyors, else RET

03A5 D7 RST #10 ; if mario is alive, continue, else RET

```

De code hierna heeft met hamer hit detectie te maken. Aanpassen door de RST #30 te vervangen door een NOP.

Hamer slaat de barrels nog niet stuk, maar wellicht de aanroepende code nog niet doorlopen.

Volgende RST #30:

```

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

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

```

Dit heeft te maken met einde level detectie en Donkey Kong goed plaatsen. Die zakt nu nog erdoorheen. Aanpassen door de RST #30 te vervangen door een NOP.

Donkey Kong zakt er nog steeds door. Dat komt omdat het einde level ook apart geregeld wordt voor elk screen. Daar de keuze ook aanpassen dat het einde van girders gekozen wordt.

Dat hele stuk staat van #1615 t/m #196A. Voor elk screen is er een tabel met CALL's die afhankelijk van de stap uitgevoerd worden. Alleen die van girders houden en de uitwerkingsstappen die niet bij girders horen vrijgeven.

```

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

                                                ; handle for girders or elevators, they are same here

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

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

```

Dit betekent dat de testen ertussen uit kunnen.

```

1618 3A8863 LD A, (#6388)   ; load A with this counter usually zero
161B EF RST #28             ; jump based on A

161C 54 16                  ; #1654      ; 0
161E 70 16                  ; #1670      ; 1
1620 8A 16                  ; #168A      ; 2
1622 32 17                  ; #1732      ; 3
1624 57 17                  ; #1757      ; 4
1626 8E 17                  ; #178E      ; 5

```

Dat werkt goed. Nu wel einde van het level, Donkey Kong zakt niet meer, pakt Pauline en klimt met Pauline omhoog.

Nu versie 0.05 opgeslagen.

Dan de niet gebruikte stukken vrijgeven. Dat zijn:

```
#1628 t/m 1653
#16A3 t/m 1707
#17B7 t/m 193C
```

Vrijgegeven. Gaat goed.

Nu versie v0.06 opgeslagen.

Volgende RST #30: nr 3 is al wel: valt in zojuist vrijgemaakte stuk

Volgende RST #30: nr 4 is al wel: valt in zojuist vrijgemaakte stuk

Volgende RST #30: nr5:

```

1E41 3E05      LD      A,#05          ; A := 5 = binary 0101
1E43 F7        RST     #30          ; only allow continue on girders and elevators, others do RET here
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

```

Dit heeft te maken met bonus sounds (wanneer hamer of springen over barrels?). Aanpassen door de RST #30 te vervangen door een NOP.

Heeft nog niet direct effect. Wellicht moet de detectie van het event nog aangepast worden.

Volgende RST #30: nr 6: Is al aangepast. Dit stuk wordt vanuit de main al niet meer aangeroepen en is al vrijgegeven om te hergebruiken.

Volgende RST #30: nr 7:

```

; called from main routine at #1992
; copies pie buffer to pie sprites

24EA 3E02      LD      A,#02          ; check level for conveyors
24EC F7        RST     #30          ; if not conveyors, RET, else continue
24ED CD2325    CALL    #2523      ; check for deployment of new pies

```

Deze wordt aangeroepen vanuit de main routine:

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

Deze aanroep in NOP's veranderen. En dan kan het stuk van #24EA t/m #25F1 vrijgegeven worden.

Nu versie v0.07 opgeslagen.

Volgende RST #30: nr 8: Is al aangepast. Dit stuk wordt vanuit de main al niet meer aangeroepen en is al vrijgegeven om te hergebruiken.

Volgende RST #30: nr 9:

```

; arrive here from main routine at #19A7

26FA 3E04      LD      A,#04          ; A := 4 = 0100 binary
26FC F7        RST     #30          ; only continue here if elevators, else RET

; 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

```

Deze wordt aangeroepen vanuit de main routine:

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

Deze aanroep in NOP's veranderen. Het stuk niet vrijgeven omdat we uiteindelijk wellicht nog wel een level willen met elevators. En dan heb je dit wellicht weer nodig (maar dan met een ander screen number als test).

Volgende RST #30: nr 10:

```
; arrive here when jumping at top of jump, check for hammer grab
```

```
2954  3E0B      LD      A,#0B          ; A := #0B = 1011 binary
2956  F7        RST     #30          ; if level is elevators RET from this sub now. no hammers on elevators.
2957  CD7429    CALL    #2974          ; load A with 1 if hammer is grabbed, 0 if no grab
295A  321862    LD      (#6218),A    ; store into hammer grabbing indicator
```

Aanpassen door de RST #30 te vervangen door een NOP. Vanuit elevators wordt hieruit gesprongen, maar dat moet niet: dit altijd uitvoeren.

Volgende RST #30: nr 11:

```
; called from #2B23 during a jump
```

```
29AF  3E04      LD      A,#04          ; A := 4 = 0100
29B1  F7        RST     #30          ; only continue here if we are on the elevators, else RET

29B2  FD210062  LD      IY,#6200        ; load IY with mario's array
29B6  3A0562    LD      A,(#6205)       ; load A with mario's Y position
29B9  4F        LD      C,A          ; copy to C
29BA  210804    LD      HL,#0408        ; H := 4, L := 8
29BD  CD222A    CALL    #2A22          ; check for collision with elevators
29C0  A7        AND     A          ; was there a collision?
29C1  CA202A    JP      Z,#2A20        ; no, load B with #00 and return
```

Aanpassen door de RST #30 te vervangen door een RET. Dit alleen voor elevators relevant en nu dus niet. Meteen een return doen.

Volgende RST #30: nr 12: Deze is al aangepast. Vervangen door NOP zodat dit altijd uitgevoerd wordt.

Volgende RST #30: nr 13: Deze is al aangepast. Vervangen door NOP zodat dit altijd uitgevoerd wordt.

Volgende RST #30: nr 14: Dit stuk code wordt vanuit de main al niet meer aangeroepen. Maar deze code kan nog wel vrijgegeven worden. Vrijgeven #2DDB t/m #2E03.

Volgende RST #30: nr 15:

```

; called from main routine at #198F
; called during the elevators. used to move the bouncers ????

2E04 3E04      LD     A,#04          ; A := 4 (0100 binary) to check for elevators screen
2E06 F7        RST    #30          ; if not elevators it will return to program

2E07 D7        RST    #10          ; if mario is alive, continue, else RET

```

Aanpassen door de RST #30 te vervangen door een RET. Dit alleen voor elevators relevant en nu dus niet. Meteen een return doen.

Nu versie v0.08 opgeslagen.

Volgende RST #30: nr 16:

```

; arrive from main routine at #1998
; checks for hammer grabs etc ?

2ED4 3E0B      LD     A,#0B          ; B = # 1011 binary
2ED6 F7        RST    #30          ; continue here on girders, conveyors, rivets only. elevators RET from this sub, it has no hammers.
2ED7 D7        RST    #10          ; continue here only if mario is alive, otherwise RET from this sub

```

Aanpassen door de RST #30 te vervangen door een NOP. Voor elevators wordt hieruit gesprongen, maar dat moet niet. Altijd doen.

Volgende RST #30: nr 17: Dit stuk code wordt vanuit de main al niet meer aangeroepen. Maar deze code kan nog wel vrijgegeven worden. Vrijgeven #2FCB t/m #2FEF.

Volgende RST #30: nr 18:

```

; This subroutine returns to the higher subroutine (causing a ladder to NOT be taken) if a fireball is on the top girder and we are not on rivets.
; called from #334A

33A1 3E07      LD     A,#07          ; \ Return if immediately we are on rivets, fireballs do not get stuck on the top in this case
33A3 F7        RST    #30          ; /
33A4 DD7E0F      LD     A,(IX+#0F)    ; \ Return if Y-position is >= 59 (i.e., fireball is not on the top girder)
33A7 FE59      CP     #59          ; |
33A9 D0        RET    NC          ; /

```

Aanpassen door de RST #30 te vervangen door een NOP. Voor rivets wordt hieruit gesprongen, maar dat moet niet. Altijd doen: Fireballs mogen de top-girder niet verlaten.

Volgende RST #30: nr 19: Deze is al aangepast aan de aanroepende kant. Hier komt de code niet meer.

Nu versie v0.09 opgeslagen.

Nog steeds de volgende zaken herstellen:

- Jumpman springen bij lage stukken, ‘stoot’ zijn hoofd.
- Geen punten voor springen over barrels.
- Geen punten voor hameren van barrels.

- Fireballs spawnen op de verkeerde plaats.

De checks voor het jumpen over items (barrels en fireballs):

```
; called from #286B
; a patch ?

3E88 3A2762 LD A,(#6227) ; load A with screen number
3E8B E5 PUSH HL ; save HL
3E8C EF RST #28 ; jump to new location based on screen number

; data for above:

3E8D 00 00 ; unused
3E8F 99 3E ; #3E99 - girders
3E91 B0 28 ; #28B0 - pie
3E93 E0 28 ; #28E0 - elevator
3E95 01 29 ; #2901 - rivets
3E97 00 00 ; unused

; checks for jumps over items on girders

3E99 E1 POP HL ; restore HL
3E9A AF XOR A ; A := 0
```

Deze check eigenlijk niet doen en altijd springen naar 3E99. Dat kan door #3E88 t/m #3E98 te vervangen door NOP's. Dan hele check weg en altijd de girders check uitvoeren.

Dit gaat goed. Nu worden de jumps over de barrels wel gezien en worden punten toegekend. Alleen nog wel te hoge score ervoor gerekend: bijna alle barrels 800 punten.

Het volgende stuk gaat ovefr het spawnen van de fireballs op de juiste plek:

```
32BD 3A2762 LD A,(#6227) ; \ Jump if we are currently on barrels
32C0 FE01 CP #01 ; |
32C2 CACE32 JP Z,#32CE ; /

32C5 FE02 CP #02 ; \ Jump if we are on conveyors
32C7 CAD232 JP Z,#32D2 ; /

32CA CDB934 CALL #34B9 ; Spawn fireball in proper location on rivets
32CD C9 RET ; return

32CE CD2C34 CALL #342C ; Handle fireball movement while coming out of oilcan on barrels
32D1 C9 RET ; return

32D2 CD7834 CALL #3478 ; Handle fireball movement while coming out of oilcan on conveyors
32D5 C9 RET ; return
```

Dus hier altijd het stuk uitvoeren voor het girders screen. Dit kan door de conditionele JP Z in #32C2 te veranderen in een JP: #32C2: C3CE32. Dit gaat goed. De fireballs spawnen nu wel goed uit de oil can beneden.

Maar de firefoxen worden niet aangepast aan de schuine girders en zakken erin of gaan er juist boven zweven. Dat is nog niet goed. Daarnaast lijkt de punten scoring voor de barrels eerst wel goed te gaan, maar op een gegeven moment niet meer. Het vreemde is tevens dat jumpman niet af gaat door een barrel, maar wel door een fireball. Allemaal kleine punten die toch nog rechtgezet moeten worden. Eerst nu maar eens kijken naar de plekken waar het screen nummer #6227 gebruikt wordt om keuzen te maken. Deze allemaal aanpassen net als de RST #30's.

#6227 1: Commentaar.

#6227 2: In de RST #30: laten staan.

#6227 3: Zetten van de waarde: laten staan.

#6227 4: Is al eerder aangepast.

#6227 5: Is al eerder aangepast.

#6227 6:

```
048D 3A2762 LD A, (#6227) ; load A with screen number  
0490 FE04 CP #04 ; are we on the rivets level?  
0492 CABE04 JP Z,#04BE ; yes, jump ahead to handle
```

Voor de rivets verder springen. Heeft te maken met de Help scream van Pauline en die is voor Rivets anders omdat ze omdraait naar de kant van jumpman en de Help dan aan de andere kant weergegeven moet worden. Aanpassen dat deze jump niet gedaan wordt: altijd de girders code doen: #0492: 3x NOP. En het stuk met de Rivets specifieke code kan vrijgegeven worden: #04BE t/m #0513.

#6227 7: Zetten van de waarde: laten staan.

#6227 8: Zetten van de waarde: laten staan.

#6227 9: Zetten van de waarde: laten staan.

#6227 10: Is al eerder aangepast.

#6227 11:

```

; other screens return here

0CC6 CDA70D    CALL    #0DA7          ; draw the screen

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

```

Deze call om de rivets te tekenen niet doen. Dus #0CCE: 3x NOP.

#6227 12: Is al eerder aangepast.

#6227 13: Commentaar.

#6227 14:

```

0FAD 3A2762    LD      A,(#6227)      ; load A with screen number
0FB0 4F        LD      C,A          ; copy to C, used at #0FCB
0FB1 CB57      BIT     2,A          ; is this the rivets ?
0FB3 2016      JR      NZ,#0FCB      ; yes, skip ahead [would be better to jump to #1131, or JR to #0FCC]

```

De call om te skippen voor rivets niet doen. Dus #0FB3: 2x NOP.

De waarde van de #6227 is ook naar C gekopieerd en wordt later gebruikt voor een jumptable. Maar dat is al eerder aangepast.

#6227 15:

```

; set initial mario sprite position and draw remaining lives and level

123C DF        RST    #18          ; count down WaitTimerMSB and only continue when 0
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

```

De alternatieve startcoordinaten van jumpman voor elevators niet doen. Dit kan later wel handig zijn voor een screen waarbij jumpman ergens bovenin begint. Maar vooralsnog niet. Conditionele skip niet doen (wel vreemd: commentaar lijkt niet te kloppen: skip step if elevators, terwijl de step de alternatieve coordinate voor elevators zou zetten). #1245: 3x NOP.

Nu versie v0.10 opgeslagen.

#6227 16: Is al eerder aangepast.

#6227 17: Zetten van de waarde: laten staan.

#6227 18: Zetten van de waarde: laten staan.

#6227 19: Is al eerder aangepast.

#6227 20:

```
; arrive here when walking over a rivet, not jumping. from #1AB9, or from #1C70

1D95 322562 LD (#6225),A ; store A into bonus sound indicator. A is zero so this clears the indicator
1D98 3A2762 LD A,(#6227) ; load A with screen number
1D9B 3D DEC A ; is this the girders?
1D9C C8 RET Z ; yes , then return, we don't play this sound for the girders
```

Voor de girders een RET doen. De conditionele RET Z vervangen door een RET. #1D9C: C9.

#6227 21: Is al eerder aangepast.

#6227 22: Is al eerder aangepast.

#6227 23:

```
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
```

Als girders dan doorgaan. Dus de RET NC veranderen in NOP's. De code erna altijd uitvoeren. #2432: NOP.

#6227 24:

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

Nu altijd girders dus de skip ahead altijd doen. Dus conditionele JP Z vervangen door een JP. #245D: C37124.

#6227 25: Commentaar.

#6227 26:

```

286F 3A2762 LD A,(#6227) ; load A with screen number
2872 E5 PUSH HL ; save HL

2873 EF RST #28 ; jump to address below depending on screen:

2874 00 00 ; unused
2876 80 28 ; #2880 - girders
2878 B0 28 ; #28B0 - conveyors
287A E0 28 ; #28E0 - elevators
287C 01 29 ; #2901 - rivets
287E 00 00 ; unused

```

Alleen de code voor girders doen. De RST #28 helemaal niet doen. Gewoon gaan naar #2880. Dat kan door #2873 t/m 287F in NOP's te veranderen. De andere stukken kunnen vrijgegeven worden: #28B0 t/m #28DF, #28E0 t/m #2900 en #2901 t/m #2912.

#6227 27:

```

; arrive here when a jump is in progress
; called from #2B20 above

2B29 3A2762 LD A,(#6227) ; load A with screen number
2B2C 3D DEC A ; are we on the girders?
2B2D C2532B JP NZ,#2B53 ; No, skip ahead

```

Nu altijd girders dus de skip ahead nooit doen. Dus de conditionele JP NZ vervangen door NOP's. #2B2D: 3x NOP.

#6227 28: Zit in een stuk dat vanuit de main niet meer aangeroepen wordt en al vrijgegeven is.

#6227 29:

```

3184 3A2762 LD A,(#6227) ; \ Jump ahead if screen is not conveyors (i.e., the screen is rivets)
3187 FE02 CP #02 ; |
3189 C29531 JP NZ,#3195 ; /

```

Nu altijd girders. Dus dat is niet conveyors. Dus altijd verder springen. Dus de conditionele JP NZ vervangen door een JP. #3189 C39531.

Nu versie v0.11 opgeslagen.

#6227 30:

```

32BD 3A2762 LD A,(#6227) ; \ Jump if we are currently on barrels
32C0 FE01 CP #01 ; |
32C2 CACE32 JP Z,#32CE ; /
32C5 FE02 CP #02 ; \ Jump if we are on conveyors
32C7 CAD232 JP Z,#32D2 ; /

```

Nu altijd girders. Dus de conditionele JP Z, moet altijd uitgevoerd worden. Veranderen in een JP. #32C2: C3CE32. Is al eerder gedaan blijkbaar.

#6227 31:

```

33C3 3A2762 LD A,(#6227) ; \ Return if we are not on barrels
33C6 fe01 CP #01 ; |
33C8 c0 RET NZ ; /
33C9 DD660E LD H,(IX+#0E) ; Load H with fireball X-position
33CC DD6E0F LD L,(IX+#0F) ; Load L with fireball Y-position
33CF DD460D LD B,(IX+#0D) ; Load B with fireball direction
33D2 CD3323 CALL #2333 ; Check for fireball moving up/down a slanted girder ?
33D5 DD750F LD (IX+#0F),L ; Store adjusted Y-position
33D8 C9 RET ; return

```

Nu altijd girders. Dus geen return doen. Dus de conditionele RET NZ veranderen in een NOP. #33C8: NOP.

#6227 32: Dit stukje code wordt aangeroepen vanuit 32CA. Maar die aanroep wordt niet meer uitgevoerd omdat daar door de aanpassing in #6227 30 overheen gesprongen wordt.

#6227 33:

```

3E88 3A2762 LD A,(#6227) ; load A with screen number
3E8B E5 PUSH HL ; save HL
3E8C EF RST #28 ; jump to new location based on screen number

; data for above:

3E8D 00 00 ; unused
3E8F 99 3E ; #3E99 - girders
3E91 B0 28 ; #28B0 - pie
3E93 E0 28 ; #28E0 - elevator
3E95 01 29 ; #2901 - rivets
3E97 00 00 ; unused

; checks for jumps over items on girders

```

Alleen de code voor girders doen. De RST #28 helemaal niet doen. Gewoon gaan naar #2880. Dat kan door #3E8C t/m 3E98 in NOP's te veranderen. De andere stukken kunnen vrijgegeven worden: #28B0 t/m #28DF, #28E0 t/m #2900 en #2901 t/m #2912. Die waren al vrijgegeven bij #2667 26. Dit lijkt een patch op de eerdere code. Door deze patch wordt voor girders naar een ander stuk code gesprongen, maar voor de andere screens naar de oorspronkelijke code.

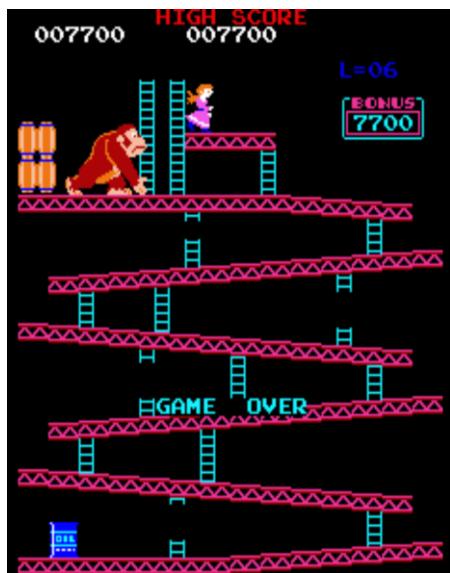
Dit is blijkbaar al eerder aangepast. Gaat om het springen over items. Daar zit nog wel een probleem. Punten worden niet goed geteld.

Er is inderdaad een probleem. De eerdere aanpassing heeft ook #3E88 t/m 3E8B in NOP's veranderd. Daardoor wordt er geen PUSH HL gedaan, maar daarna wel een POP HL. Dat is niet goed. Daarom #3E88 t/m 3E8B terugzetten naar orginele code.

Lijkt nu helemaal goed te werken. Een flink aantal keren testen door te spelen. Als geen problemen meer, dan kan dit level naar Crazy Kong variant aangepast gaan worden.

Nu versie v0.12 opgeslagen.

Er is nog een klein probleem met de GAME OVER melding. Deze wordt in een kleinere box weergegeven:



Dit klopt. Wissen van deel van het scherm en plaatsen van de GAME OVER tekst wordt hier gedaan:

```
1322 CD2618    CALL    #1826          ; clear an area of the screen
1325 110003    LD      DE,#0300        ; load task data for text #0 "GAME OVER"
1328 CD9F30    CALL    #309F          ; insert task to draw text
```

Het stukje dat het scherm wist is #1826 t/m #1838:

```

; called from several places with HL preloaded with a video RAM address
; used to clear sections of the rivets screen when it is completed

1826 11DBFF LD DE,#FFDB ; load DE with offset for each column
1829 0E0E LD C,#0E ; for C = 1 to #0E
182B 3E10 LD A,#10 ; A := #10 (clear space on screen)

182D 0605 LD B,#05 ; for B = 1 to 5

182F 77 LD (HL),A ; store A into (HL) - clears the screen element
1830 23 INC HL ; next HL
1831 10FC DJNZ #182F ; next B

1833 19 ADD HL,DE ; add offset to HL
1834 0D DEC C ; next C
1835 C22D18 JP NZ,#182D ; loop until done

1838 C9 RET ; return

```

Omdat dit een stukje voor rivets leek, is dit vrijgegeven. Dit moet teruggezet worden. Zonder gat over te laten zou dat kunnen op #17B6:

```

17B6 11DBFF LD DE,#FFDB ; load DE with offset for each column
17B9 0E0E LD C,#0E ; for C = 1 to #0E
17BB 3E10 LD A,#10 ; A := #10 (clear space on screen)

17BD 0605 LD B,#05 ; for B = 1 to 5

17BF 77 LD (HL),A ; store A into (HL) - clears the screen element
17C0 23 INC HL ; next HL
17C1 10FC DJNZ #17BF ; next B

17C3 19 ADD HL,DE ; add offset to HL
17C4 0D DEC C ; next C
17C5 C2BD17 JP NZ,#17BD ; loop until done

17C8 C9 RET ; return

```

En de aanroep aanpassen:

```
1322 CDB617 CALL #17B6 ; clear an area of the screen
```

Dit werkt nu wel goed. Ook even testen de werking met twee spelers. Werkt niet goed voor player 2, daar wordt niets gewist lijkt het wel.

In #1373 staat nog een aanroep naar het wissen van een stuk scherm.

```
1373 CD2618 CALL #1826 ; clear an area of the screen
```

Ook deze aanpassen:

```
1373 CDB617 CALL #17B6 ; clear an area of the screen
```

Nu is dat goed.

Nu versie v0.13 opgeslagen.

Screen 02: Crazy Kong 1

Voor dit scherm moeten de girders ingekort worden:



De definitie voor scherm 2 staat vanaf #3B5D. Dit is de definitie van de girders (van het orginele barrels screen).

```
3AE4: 02 97 38 68 38 ; top girder where girl sits  
3AE9: 02 9F 54 10 54 ; girder where kong sits  
3AED: 02 DF 58 A0 55 ; 1st slanted girder at top right  
3AF3: 02 EF 6D 20 79 ; 2nd slanted girder (has hammer at left side)  
3AF8: 02 DF 9A 10 8E ; 3rd slanted girder  
3AFD: 02 EF AF 20 BB ; 4th slanted girder  
3B02: 02 DF DC 10 D0 ; 5th slanted girder (has hammer at right side)  
3B07: 02 FF F0 80 F7 ; bottom slanted girder  
3B0C: 02 7F F8 00 F8 ; bottom flat girder where mario starts
```

Eerste aanpassing lijkt erop dat hij deze definitie niet pakt.

Klopt hier moeten we eerst aandacht aan besteden. Gaat om de setup van het screen. Is nu zodanig dat altijd de setup van screen 1 gedaan wordt (originele barrels). Dit wordt geregeld in het stuk van #0C92 t/m #0D5E eigenlijk zo maken dat er eerst een generieke setup gedaan wordt en dan op basis van een RST #30 conditioneel een specifieke setup voor elk screen. Daar dan rekening houden met 15 verschillende screens.

; arrive here from #0776 during attract mode

```
0C92 CD7408 CALL #0874 ; clears the screen and sprites
0C95 AF XOR A ; A := 0
0C96 328C63 LD (#638C),A ; reset onscreen timer
0C99 110105 LD DE,#0501 ; load DE with task #5
0C9C CD9F30 CALL #309F ; insert task
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 3E08 LD A,#08 ; A := 8 = music code for girders
0CA9 328960 LD (#6089),A ; set music for girders
;
0CAC 3A2762 LD A,(#6227) ; load A with screen number
0CAF EF RST #28 ; jump depending on the screen
;
; jump table data
;
OCB0 00 00 ; unused
OCB2 E0 0C ; #0CE0 for screen 1 Barrels Original
OCB4 14 0D ; #0D14 for screen 2 Into The Dark
OCB6 E6 0C ; #0CE6 for screen 3 Crazy Kong 2
OCB8 E0 0C ; #0CE0 for screen 4 Wild Barrels
OCBA EC 0C ; #0CEC for screen 5 Reverse
OCBC F2 0C ; #0CF2 for screen 6 Both Blinking Hammers
OCBE FD 0C ; #0CFD for screen 7 Use Long Ladders
OCC0 E0 0C ; #0CE0 for screen 8 Random Slow Barrels
OCC2 28 0D ; #0D28 for screen 9 Uneven spaced girders
OCC4 3A0D ; #0D3A for screen 10 Pauline's presents
OCC6 45 0D ; #0D45 for screen 11 Springs
OCC8 4B 0D ; #0D4B for screen 12 Anti gravity
OCCA 00 00 ; #XXXX for screen 13
OCCC 00 00 ; #XXXX for screen 14
OCCE 00 00 ; #XXXX for screen 15
;
0CD0 CDA70D CALL #0DA7 ; draw the screen
;
0CD3 C35F0D JP #0D5F ; continue setup
;
Stukje vrijhouden en daarna de specifieke code voor orginelle girders (screen 1) en Crazy Kong 1
(screen 2). Later de andere daarvan toevoegen.
;
OCE0 11E43A LD DE,#3AE4 ; Load DE with start of table data for screen 1
OCE3 C3D00C JP #0CD0 ; jump back
;
```

```

0CE6 115D3B LD DE,#3B5D ; Load DE with start of table data for screen 2
0CE9 C3D00C JP #0CD0 ; jump back
;

```

En de rest t/m #0D5E vrijgeven.

Additionele schermen hierboven toevoegen in de jump.

Nu weer verder met het aanpassen van screen 2: Crazy Kong 1:

De aanpassingen zijn als volgt:

```

3B5D: 02 97 38 68 38 ; top girder where girl sits
3B62: 02 9F 54 10 54 ; girder where kong sits
3B67: 02 CF 57 A0 55 ; 1st slanted girder at top right
3B6C: 02 EF 6D 30 80 ; 2nd slanted girder (has hammer at left side)
3B71: 02 CF 99 10 8E ; 3rd slanted girder
3B76: 02 EF AF 30 BC ; 4th slanted girder
3B7B: 02 CF DB 10 D0 ; 5th slanted girder (has hammer at right side)
3B80: 02 FF F0 80 F7 ; bottom slanted girder
3B85: 02 7F F8 00 F8 ; bottom flat girder where mario starts

```

Het veld ziet er nu als volgt uit:



Nu versie v0.14 opgeslagen.

Nu de barrels eerder een roll-over laten doen omdat de randen van de girders opgeschoven zijn. En dat dan maar meteen zo maken dat dat voor screen 2 en screen 3 gedaan wordt.

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

```

Dit aanpassen naar

```

2017 C3BE04 JP #04BE ; check barrel left edge, jump to label AAAA
;
201A C3E104 JP #04E1 ; check barrel right edge, jump to label DDDD – label CCCC
;
201D – 2023 NOP's
;
; Check barrel roll-over left edge
;
04BE 3A2762 LD A,(#6227) ; load A with screen number – label AAAA
04C1 FE02 CP #02 ; is this screen 2 ?
04C3 CAD604 JP Z, #04D6 ; yes, jump to label BBBB
04C6 FE03 CP #03 ; is this screen 3 ?
04C8 CAD604 JP Z, #04D6 ; yes, jump to label BBBB
;
; Check barrel left edge most screens
;
04CB DD7E03 LD A,(IX+#03) ; load A with Barrel's X position
04CE FE1C CP #1C
04D0 DA2F20 JP C, #202F ; yes, jump ahead to handle
04D3 C31A20 JP #201A ; continue check right edge, jump to label CCCC
;
; Check barrel left edge screens #2 and #3
;
04D6 DD7E03 LD A,(IX+#03) ; load A with Barrel's X position – label BBBB
04D9 FE2C CP #2C
04DB DA2F20 JP C, #202F ; yes, jump ahead to handle
04DE C31A20 JP #201A ; continue check right edge, jump to label CCCC
;
; Check barrel roll-over right edge
;
04E1 3A2762 LD A,(#6227) ; load A with screen number – label DDDD
04E4 FE02 CP #02 ; is this screen 2 ?
04E6 CAF904 JP Z, #04F9 ; yes, jump to label EEEE
04E9 FE03 CP #03 ; is this screen 3 ?
04EB CAF904 JP Z, #04F9 ; yes, jump to label EEEE
;
; Check barrel right edge most screens
;

```

```

04EE DD7E03 LD A,(IX+#03) ; load A with Barrel's X position
04F1 FEE4 CP #E5
04F3 D22420 JP NC, #2024 ; yes, jump ahead to handle
04F6 C3BA21 JP #21BA ; continue, jump way ahead
;
; Check barrel right edge screens #2 and #3
;
04F9 DD7E03 LD A,(IX+#03) ; load A with Barrel's X position – label EEEE
04FC FED4 CP #D5
04FE D22420 JP NC, #2024 ; yes, jump ahead to handle
0501 C3BA21 JP #21BA ; continue, jump way ahead

```

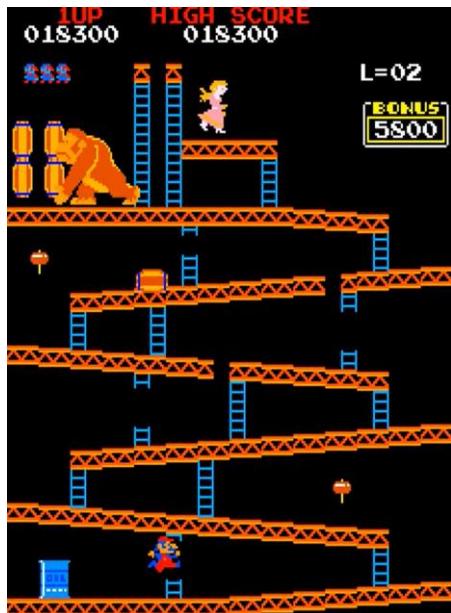
Dit werkt goed. Allen het pakken van de hamer linksboven is lastig. Hoe gaat dat in Crazy Kong? Anders de hamer verplaatsen. In de Crazy Kong val je ook met de hamer naar beneden. Daar lukt het helemaal niet om de hamer te pakken en weer terug te bouncen. Gaat met mijn variant wel. Maar wel erg lastig.

Nu versie v0.15 opgeslagen.

Het bovenste stukje girder laten staan? Net als in Crazy Kong? Nee, is niet mooi. Gewoon laten zoals het is.

Screen 03: Crazy Kong 2

Dit is hetzelfde veld als veld 2 maar dan met twee gaten in de girders.



Dus veld 3 in de definitie opnemen.

0CB6 E6 0C ; #0CE6 for Crazy Kong 2

Is dezelfde definitie als veld 2. Kijken of het hiermee werkt.

Veld drie werkt. Nu nog de twee gaten maken. Dat zou kunnen ergens in de code die ook de bovenste trapdelen verwijderd. Dat gebeurt in het stuk code van #0FB5 t/m #101E. Er is daar een stuk vrij met NOP's. Daar een CALL opnemen naar extra code waarin dit goedgezet kan worden.

0FCB CD2010 CALL #1020 ; additional code to erase griders for screen 3

en

```
1020 3A2762 LD A,(#6227) ; load A with screen number
1023 FE03 CP #03 ; is this screen 3 ?
1025 C0 RET NZ ; no, return
;
1026 212E75 LD HL,#752E ; video memory address to clear
1029 3E10 LD A,#10 ; load a with space character
102B 77 LD (HL),A ; clear memory address
;
102C 211276 LD HL,#7612 ; video memory address to clear
102F 3E32 LD A,#10 ; load a with I character
1031 77 LD (HL),A ; clear memory address
1032 23 INC HL ; next memopry address, one lower
1033 77 LD (HL),A ; clear memory address
;
1034 C9 RET
```

Dit werkt helemaal goed.

Nu versie v0.16 opgeslagen.

Screen 04: Wild barrels

Dit is hetzelfde veld als veld 1, maar dan met heel veel wild barrels.

Dus veld 4 in de definitie opnemen.

0CB8 E0 OC ; #0CEO for Wild Barrels

Is dezelfde definitie als veld 1. Kijken of het hiermee werkt.

Werkt nog niet helemaal goed. Er worden namelijk rivets gewist en dat geeft oof nog punten. Dat moet natuurlijk niet. Dat eerst oplossen.

Checken op rivets wordt aangeroepen vanuit de main:

199E CD331A CALL #1A33 ; check for and handle running over rivets

Deze aanroep niet meer doen door te veranderen in 3x NOP.

Dan kan de code die dit doet ook vrijgegeven worden: #1A33 t/m # 1AC2.

Dan zorgen voor meer wild barrels. Dit wordt hier geregeld:

```
2C41 CD5700 CALL #0057 ; else load A with a random number

;; hack to increase crazy barrels
;; 2C41 3E 00 LD A, #00
;; 2C43 00 NOP

;; hack to increase crazy barrels:
;; 2C44 E600 AND #00 ; mask all 4 bits to zero
;;
;; 2C44 E60F AND #0F ; mask out left 4 bits to zero. A becomes a number between 0 and #F
2C46 C2862C JP NZ,#2C86 ; If result is not zero, deploy a normal barrel. this routine sets #6382 to 0,
; loads A with 3 and returns to #2C4F
```

Bij #2C41 springen naar additionele code die dit afhankelijk maakt van screen nummer.

```
2C41 C3331A JP #1A33 ; jump to additional code
2C44-2C48 NOP's
```

En

```
1A33 3A2762 LD A,(#6227) ; load A with screen number
1A36 FE04 CP #04 ; is this screen 4 ?
1A38 CA461A JP Z, #1A46 ; yes jump forward, jump to label AAAA
;
1A3B CD5700 CALL #0057 ; load A with random number
1A3E E60F AND #0F ; mask out left 4 bits to zero
1A40 C2862C JP NZ, #2C86 ; if result not zero, deploy normal barrel
1A43 C3492C JP 2C49 ; else deploy crazy barrel
;
1A46 CD5700 CALL #0057 ; load A with random number – label AAAA
1A49 E603 AND #03 ; mask out left 6 bits to zero
1A4B C2862C JP NZ, #2C86 ; if result not zero, deploy normal barrel
1A4E C3492C JP 2C49 ; else deploy crazy barrel
```

Dat werkt nu goed.

Nu versie v0.17 opgeslagen.

Screen 05: Reverse naar hartje

Dit is een nieuw veld waarin de trap naar Pauline is weggelegd.

Deze definitie neerzetten op: #3BD6 t/m #3C49. Kopie van de definitie van het originele veld, waar de ladder naar Pauline weggehaald is.

Veld 5 in de definitie opnemen.

```
0CBA EC 0C ; #0CEC for screen5 Reverse
```

En dan de table data for screen 5 laden:

```
0CEC 11D63B LD DE,#3BD6 ; Load DE with start of table data for screen 5  
0CEF C3D00C JP #0CD0 ; jump back
```

Het veld wordt goed weergegeven. De trap naar Pauline is niet zichtbaar. Nu eerst de heart sprite daar beneden weergeven.

Dat zou kunnen ergens in de code die ook de bovenste trapdelen verwijderd. Dat gebeurt in het stuk code van #0FB5 t/m #101E. Er is daar een stuk vrij met NOP's. Daar een JP opnemen naar extra code waarin dit goedgezet kan worden.

```
0FCE CD3510 CALL #1035 ; additional code to display heart in screen 5
```

en

```
1035 3A2762 LD A,(#6227) ; load A with screen number  
1038 FE05 CP #05 ; is this screen 5 ?  
103A C0 RET NZ ; no, return  
;  
103B 21206A LD HL,#6A20 ; load HL with heart sprite  
103E 36D6 LD (HL),#D6 ; set heart sprite x-position  
1040 23 INC HL ; next  
1041 3676 LD (HL),#76 ; set heart sprite  
1043 23 INC HL ; next  
1044 3609 LD (HL),#09 ; set heart sprite color  
1046 23 INC HL ; next  
1047 36C8 LD HL,#C8 ; set heart sprite y-position  
;  
1049 C9 RET
```

Jumpman bovenin het scherm laten starten:

Initiële coordinaten van Jumpman worden hier gezet:

```
; set initial mario sprite position and draw remaining lives and level  
  
123C DF RST #18 ; count down WaitTimerMSB and only continue when 0  
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
```

Dit aanpassen naar:

```
#1240 FE05 CP #05 ; is this screen 5?  
#1242 01984C LD BC,#4C98 ; Coordinates jumpman for screen 5, X=98 Y=4C.
```

Dit werkt goed jumpman start nu bovenin het scherm.

Nu versie v0.18 opgeslagen.

De code voor de afhandeling van het hartje pakken van DK Twisted hier invoegen. Die geldt voor twee keer een hartje, dus kan eenvoudiger. Deze code aanpassen.

```
17C9: 3A2762 LD A,(#6227) ; load A with screen number
17CC: FE05 CP #05 ; is this screen 5 ?
17CE: C0 RET NZ ; no, return
;
17CF: 3A 9B 63 LD A,(#639B) ; load switch status variable in A
17D2: E6 04 AND #04 ; check bit 3 set: blink sparkle
17D4: CA1818 JP Z,#1818 ; if not set then do not blink – jump label AAAA
;
17D7: 21216A LD HL,#6A21 ; load HL with heart sprite
17DA: 3AA065 LD A,(#65A0) ; load A with blink counter
17DD: 3C INC A ; increment blink counter
17DE: 32A065 LD (#65A0), A ; store blink counter
;
17E1: 47 LD B,A
17E2: FE 16 CP #16
17E4: CA 06 18 JP Z, #1806 ; sprite off – jump label HHHH
;
17E7: 78 LD A,B
17E8: FE 32 CP #32
17EA: CA 09 18 JP Z, #1809 ; sprite on – jump label IIII
;
17ED: 78 LD A,B
17EE: FE 48 CP #48
17F0: CA 06 18 JP Z, #1806 ; sprite off – jump label HHHH
;
17F3: 78 LD A,B
17F4: FE 64 CP #64
17F6: CA 09 18 JP Z, #1809 ; sprite on – jump label IIII
;
17F9: 78 LD A,B
17FA: FE 80 CP #80
17FC: CA 06 18 JP Z, #1806 ; sprite off – jump label HHHH
;
17FF: 78 LD A,B
1800: FE 96 CP #96
1802: CA 0C 18 JP Z, #180C ; stop blinking – jump label GGGG – nu DDDD
;
1805: C9 RET ; no action continue blinking
;
1806: 3664 LD (HL),#64 ; set blank sprite – label HHHH
1808: C9 RET ; continue blinking
;
1809: 3663 LD (HL),#63 ; set sparkle sprite – jump label III
180B: C9 RET ; continue blinking
;
255A: 3A 9B 63 LD A,(#639B) ; load switch status variable in A – label GGGG
255D: E602 AND #02 ; check bit 2 set
255F: C27625 JP NZ, #2576 ; jump if switch 2 activated – jump label DDDD
;
2562: CD F0 24 CALL #24F0 ; draw lower ladder
2565: 21206A LD HL,#6A20 ; load HL with heart sprite
2568: 36E3 LD (HL),#E3 ; set heart sprite x position
256A: 23 INC HL ; next
```

```

256B: 3676 LD (HL),#76 ; set heart sprite
256D: 23 INC HL ; next
256E: 3609 LD (HL),#09 ; set heart sprite color
2570: 23 INC HL ; next
2571: 3648 LD HL,#48 ; set heart sprite y-position
2573: C37925 JP #2579 ; skip next jump label FFFF
;
180C: CD F0 24 CALL #24F0 ; draw upper ladder - label DDDD
;
180F: 3A9B63 LD A,(#639B) ; load switch status variable in A - label FFFF
1812: E6 FB AND #FB ; reset bit 3 , stop blinking sparkle
1814: 32 9B 63 LD (#639B), A ; write back to switch status variable
1817: C9 RET ; return
;
1818: 3A 9B 63 LD A,(#639B) ; load switch status variable in A - label AAAA
181B: E6 01 AND #01 ; check bit 1 set
2587: C2 B7 25 JP NZ,#25B7 ; jump to next switch - jump label FFFF
181D: C0 RET NZ ; heart is already grabbed
;
181E: 3A 03 62 LD A,(#6203) ; load A with jumpman's x-position
1821: FE D6 CP #D6 ; check x-position = #D6
1823: C0 RET NZ ; return if x-position is not correct
1824: 3A 05 62 LD A,(#6205) ; load A with jumpman's y-position
1827: FE D4 CP #D4 ; check y-position = #D4
1829: C0 RET NZ ; return if y-position is not correct
182A: 3A 9B 63 LD A,(#639B) ; load switch status variable in A
182D: F6 05 OR #05 ; set bit 1 and bit 3, switch 1 activated, blink
182F: 32 9B 63 LD (#639B), A ; write back to switch status variable
1832: 3E 03 LD A,#03 ; load sound duration of 3x
1834: 32 85 60 LD (#6085),A ; play sound for item pickup
;
1837: 21206A LD HL,#6A20 ; load HL with heart sprite
183A: 3693 LD (HL),#93 ; set heart sprite x-position
183C: 23 INC HL ; next
183D: 3663 LD (HL),#63 ; set sparkle sprite
183F: 23 INC HL ; next
1840: 360C LD (HL),#0C ; set heart sprite color
1842: 23 INC HL ; next
1843: 364A LD HL,#4A ; set heart sprite y-position
;
1845: 3E 00 LD A,#00 ; A:=0
1847: 32A065 LD (#65A0), A ; reset blink counter
184A: C9 RET ; return
;
25B7: 3A 9B 63 LD A,(#639B) ; load switch status variable in A - label FFFF
25BA: E6 02 AND #02 ; check bit 2 set
25BC: C0 RET NZ ; return if switch already activated
25BD: 3A 03 62 LD A,(#6203) ; load A with jumpman's x-position
25C0: FE E3 CP #E3 ; check x position = #E3
25C2: C0 RET NZ ; return if x position is not correct
25C3: 3A 05 62 LD A,(#6205) ; load A with jumpman's y-position
25C6: FE 49 CP #49 ; check y position = #49
25C8: C0 RET NZ ; return if y position is not correct
25C9: 3A 9B 63 LD A,(#639B) ; load switch status variable in A
25CC: F6 06 OR #06 ; set bit 2 and 3, switch 2 activated, blink
25CE: 32 9B 63 LD (#639B), A ; write back to switch status variable
25D1: 3E 03 LD A,#03 ; load sound duration of 3
25D3: 32 85 60 LD (#6085),A ; play sound for item pickup
;
25D6: 21206A LD HL,#6A20 ; load HL with heart sprite
25D9: 3693 LD (HL),#93 ; set heart sprite x-position
25DB: 23 INC HL ; next
25DC: 3663 LD (HL),#63 ; set sparkle sprite

```

```

25DE: 23      INC HL          ; next
25DF: 3609    LD (HL),#0C     ; set heart sprite color
25E1: 23      INC HL          ; next
25E2: 364A    LD HL,#4A      ; set heart sprite y position
;
25E4: 3E 00    LD A,#00        ; A:=0
25E6: 32A065   LD (#65A0),A    ; reset blink counter
25E9: C9      RET             ; return

```

Ladder tekenen wordt zo gedaan:

```

24EA: 00 93 38 93 54 AA      ; test table data removed ladder
24F0: 11EA24    LD DE,#24EA    ; load start table data
24F3: CDA70D    CALL #0DA7    ; draw screen
24F6: 21EA24    LD   HL,#24EA  ; load HL with start of table data for ladder
24F9: DD210C63   LD   IX,#630C  ; #6300 is start, with 11 ladders added: 630C
24FD: CD7524    CALL #2475    ;
2500: C9      RET             ; return

```

En dan dit als test aanroepen vanaf de #1049 (was de RET van de initial setup waarin het heart getekend wordt):

```

1049 CDF024  CALL #24F0    ; draw ladder
104C C9      RET             ; return

```

Deze test is goed. Ladder naar Pauline wordt getekend en kan beklimmen worden. Weer terugzetten naar alleen de C9.

De switchvariabele en blink counter moet nog wel initieel gezet worden:

```

1035 3A2762   LD   A,(#6227)  ; load A with screen number
1038 FE05     CP   #05       ; is this screen 5 ?
103A C0       RET  NZ       ; no, return
;
103B 21206A   LD   HL,#6A20    ; load HL with heart sprite
103E 36D6     LD   (HL),#D6    ; set heart sprite x-position
1040 23       INC  HL          ; next
1041 3676     LD   (HL),#76    ; set heart sprite
1043 23       INC  HL          ; next
1044 3609     LD   (HL),#09    ; set heart sprite color
1046 23       INC  HL          ; next
1047 36D4     LD   HL,#C8      ; set heart sprite y-position
;
1049: 3E 00    LD A,#00        ; A:=0
104B: 32A065   LD (#65A0),A    ; reset blink counter
104E: 329B63   LD (#639B),A    ; reset switch variable
;
1051 C9      RET             ; return

```

En het geheel moet vanuit de main aangeroepen worden. #1992 is de eerste vrij plek in de main. Daar tussenvoegen.

```
1992: CDC917  CALL #17C9      ; do stuff for screen 5, blink and add top ladder
```

Gaat niet fout, maar kan hartje niet pakken. Kan komen door positie? Check positie gelijk maken aan positie hartje: (D6,C8). Werkt nog steeds niet.

Kijken of het blinken wel werkt. Van de #639B moeten dan de bit 1 en bit 3 gezet: Dat is 5. Het blinken werkt dan wel.

De kleur van de sparkle is niet goed. Nog aanpassen. Moest 0C ipv 09 zijn.

Werkt nu helemaal goed.

Nu versie v0.19 opgeslagen.

Misschien nog aanpassen. Tweede barrel gewoon doorgaan en barrels onder jumpman niet laten verdwijnen. Wellicht daardoor iets lastiger. Nu vrijenvoudig vooruit te rennen.

Daartoe is dit een interessant stukje:

```
2172 78 LD A,B ; yes, load A with B which has the value of the ladder from the check ??  
2173 D605 SUB #05 ; subtract 5  
2175 DD7717 LD (IX+#17),A ; store into +17 to indicate which ladder we might be going down ???  
2178 3A4863 LD A,(#6348) ; get status of the oil can fire  
217B A7 AND A ; is the fire lit ?  
217C CAB221 JP Z,#21B2 ; no, always take ladders before oil is lit  
  
217F 3A0562 LD A,(#6205) ; else load A with mario's Y position + 5  
2182 D604 SUB #04 ; subtract 4  
2184 BA CP D ; is the barrel already below mario ?  
2185 D8 RET C ; yes, return without taking ladder
```

Even testen. #217B t/m #217E NOP's zorgt ervoor dat de tweede barrel; gewoon rechtdoor gaat.

Ander interessant stukje:

```
20A9 210562 LD HL,#6205 ; load HL with mario's Y position address  
20AC DD7E05 LD A,(IX+#05) ; load A with +5 = barrel's Y position  
20AF D616 SUB #16 ; subtract #16  
20B1 BE CP (HL) ; compare to mario Y position. is the barrel below mario?  
20B2 D2C320 JP NC,#20C3 ; yes, skip next 5 steps  
  
20B5 DD7E10 LD A,(IX+#10) ; load A with +10 = rolling over edge/direction indicator  
20B8 A7 AND A ; A == 0 ? is this barrel is rolling right?  
20B9 C2E120 JP NZ,#20E1 ; no, skip ahead and set alternate values, continue at #20C3  
  
20BC DD7711 LD (IX+#11),A ; else set +11 (???) to zero  
20BF DD3610FF LD (IX+#10),#FF ; set +10 = rolling over edge indicator to #FF for rolling left
```

#20B2 t/m 20B4 NOP's zorgt ervoor dat barrels onder jumpman niet van de rand verdwijnen.

Jumpman verder naar links laten starten:

```
#1242 01704C LD BC,#4C70 ; Coordinates jumpman for screen 5, X=70 Y=4C.
```

De test wijst uit dat het veld daardoor veel leuker wordt. Dus dit conditioneel voor scherm 5 opnemen. Na terugzetten van vorige onaangepaste versie.

```
2178 C35210 JP #1052 ; jump to additional code  
217B 00 00 00 00 NOP NOP NOP NOP
```

en

```
1052 3A2762 LD A,(#6227) ; load A with screen number  
1055 FE05 CP #05 ; is this screen 5 ?  
1057 CA7F21 JP Z, #217F ; yes, jump back  
;  
105A 3A4863 LD A,(#6348) ; no, get status of the oil can fire  
105D A7 AND A ; is the fire lit ?  
105E CAB221 JP Z,#21B2 ; no, always take ladders before oil is lit  
;  
1061 C37F21 JP #217F ; yes, jump back
```

en

```
20B1 C36401 JP #1064 ; jump to additional code  
20B4 00 NOP
```

en

```
1064 F5 PUSH AF  
1065 3A2762 LD A,(#6227) ; load A with screen number  
1068 FE05 CP #05 ; is this screen 5 ?  
106A F1 POP AF  
106B CAB520 JP Z, #20B5 ; yes, jump back  
;  
106E BE CP (HL) ; compare to mario Y position. is the barrel below mario?  
106F D2C320 JP NC,#20C3 ; yes, jump back and skip steps  
;  
1072 C3B520 JP #20B5
```

De laatste aanpassing gaat niet goed. Anders doen.

```
20A9 C36410 JP #1064
```

en

```
1064 3A2762 LD A,(#6227) ; load A with screen number  
1067 FE05 CP #05 ; is this screen 5 ?  
1069 C27210 JP NZ, #1072 ; no, jump forward, jump to label AAAA  
;  
106C 210562 LD HL,#6205 ; load HL with mario's Y position address  
106F C3B520 JP #20B5
```

```

;
1072 210562 LD HL,#6205 ; load HL with mario's Y position address – label AAAA
1075 C3AC20 JP #20AC

```

Nu gaat het wel goed.

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

Screen 06: Both blinking hammers used

Dit is hetzelfde veld als veld 5, Het veld met de missende trap naar Pauline. Maar een extra variabele vullen is nodig, dus een eigen call maken.

Dus veld 6 in de definitie opnemen.

```
0CFC F2 0C ; #0CF2 for screen 6 Both Blinking Hammers
```

En dan de table data for screen 5 laden:

```
0CF2 11D63B LD DE,#3BD6 ; Load DE with start of table data for screen 5
0CF5 C3D00C JP #0CD0 ; jump back
```

Dit werkt goed. Veld wordt weergegeven, zonder de trap naar Pauline.

Hammer grabs worden hier afgehandeld:

```

2965 FE01 CP #01 ; was lower hammer on girders & conveyors, or upper hammer on rivets, grabbed?
2967 CA6F29 JP Z,#296F ; yes, skip next 2 steps

296A DD360101 LD (IX+#01),#01 ; set 1st hammer active
296E C9 RET ; return

296F DD361101 LD (IX+#11),#01 ; set 2nd hammer active
2973 C9 RET ; return

```

Hammer counter variabele 65A0 gebruiken. Deze initieel zetten. En daarna in de bovenstaande code conditioneel voor scherm deze variabele verhogen voor beide hamers. Daarna stuk code maken dat checkt op het aantal hamers en als dat 2 is laten blinken en trap laten verschijnen.

Aanpassing t.b.v. zetten van de variabele 65A0:

```

0CF2 3E00 LD A,#00 ; A := 0
0CF4 32A065 LD #65A0,A ; set hamer counter to 0
;
0CF7 11D63B LD DE,#3BD6 ; Load DE with start of table data for screen 5
0CFA C3D00C JP #0CD0 ; jump back

```

Werkt nog steeds, maar niet veel aan te zien.

Aanpassing om de hammer counter op te hogen:

```
2965 C37810  JP    #1078      ; jump to additional code, jump to label AAAA  
2968 00 00    NOP NOP
```

en

```
1078 F5      PUSH AF      ; save A for later – label AAAA  
1079 3A2762   LD   A,(#6227)  ; load A with screen number  
107C FE06    CP   #06      ; is this screen 6 ?  
107E C28510   JP   NZ, #1085  ; no, jump forward, jump to label BBBB  
1081 21A065   LD   HL, #65A0  ; load HL with hammer counter  
1084 34      INC  (HL)     ; increment hammer counter  
1085 F1      POP  AF      ; restore A – label BBBB  
;  
1086 FE01    CP   #01      ; was lower hammer or upper hammer grabbed?  
1088 CA6F29   JP   Z, #296F  ; yes, goto 2nd hammer  
108B C36A29   JP   #296A  ; no, goto 1st hammer
```

Werkt nog steeds. De hamers zijn ook te gebruiken.

Nu maken dat de trap verschijnt als de hammer counter op 2 staat. En dan de hammer counter ophogen naar 3 zodat dit niet steeds blijft gebeuren.

#1995 is vrije plek in de main waar extra tests tussengevoegd kunnen worden.

```
1995 CD8E10  CALL  #108E      ; call routine with eight possible tests
```

Daarvandaan springen naar stuk code waar ruimte is voor meerdere tests:

```
108E 3A2762   LD   A,(#6227)  ; load A with screen number  
1091 FE06    CP   #06      ; is this screen 6 ?  
1093 CCAAAA   CALL Z, #AAAAA  ; yes, call subroutine – check two hammers grabbed  
;  
1096 3A2762   LD   A,(#6227)  ; load A with screen number  
1099 FE06    CP   #06      ; is this screen 6 ?  
109B CC0011   CALL Z, #1100  ; yes, call subroutine – make hammer blink  
;  
109E 3A2762   LD   A,(#6227)  ; load A with screen number  
10A1 FEXX    CP   #02      ; is this screen 2 ?  
10A3 CCBA16   CALL Z, #16BA  ; yes, call subroutine  
;  
10A6 3A2762   LD   A,(#6227)  ; load A with screen number  
10A9 FE0A    CP   #0A      ; is this screen 10 ?  
10AB CC3B22   CALL Z, #223B  ; yes, call subroutine  
;  
10AE 3A2762   LD   A,(#6227)  ; load A with screen number  
10B1 FEXX    CP   #0B      ; is this screen 11 ?
```

```

10B3 CCDB2D    CALL Z, #2DDB      ; yes, call subroutine
;
10B6 3A2762    LD   A,(#6227)    ; load A with screen number
10B9 FEXX      CP   #XX        ; is this screen X ?
10BB CCAAAA    CALL Z, #AAAAA    ; yes, call subroutine
;
10BE 3A2762    LD   A,(#6227)    ; load A with screen number
10C1 FEXX      CP   #XX        ; is this screen X ?
10C3 CCAAAA    CALL Z, #AAAAA    ; yes, call subroutine
;
10C6 3A2762    LD   A,(#6227)    ; load A with screen number
10C9 FEXX      CP   #XX        ; is this screen X ?
10CB CCAAAA    CALL Z, #AAAAA    ; yes, call subroutine
;
10CE C9        RET             ; return to main

```

En

```
10D0 C9        RET             ; return without doing anything
```

Eventjes testen. Dit gaat goed. Nu nog op 10D0 de code toevoegen die de check op 2 hamers doet en dan een periode de blink sprite laat zien en dan de trap tekent.

```

10D0 3AA065    LD A, (#65A0)    ; load A with hammer counter
;
10D3 FE02      CP #02        ; compare A with 02
10D5 D8        RET C         ; A < 2 ? yes: return, do nothing
10D6 CAE210    JP Z, #10E2    ; A = 2 ? yes: set sparkle sprite – jump to label AAAA
;
10D9 FE30      CP #30        ; compare A with 30
10DB DAF010    JP C,#10F0    ; A < 30 ? yes: hammer counter++ – jump to label BBBB
10DE CAF510    JP Z,#10F5    ; A = 30 ? yes: create ladder – jump to label CCCC
;
10E1 C9        RET             ; A > 30 ! return
;
10E2: 21206A    LD HL,#6A20    ; load HL with heart sprite – label AAAA
10E5: 3693      LD (HL),#93    ; set heart sprite x-position
10E7: 23        INC HL        ; next
10E8: 3663      LD (HL),#63    ; set sparkle sprite
10EA: 23        INC HL        ; next
10EB: 360C      LD (HL),#0C    ; set heart sprite color
10ED: 23        INC HL        ; next
10EE: 364A      LD HL,#4A    ; set heart sprite y-position
;
10F0 21A065    LD HL,#65A0    ; load HL with hammer counter – label BBBB
10F3 34        INC (HL)       ; increment hammer counter
10F4 C9        RET             ; return
;
```

```

10F5  21216A LD HL,#6A21 ; load HL with heart sprite – label CCCC
10F8  3664    LD (HL),#64 ; set blank sprite
;
10FA  CDF024 CALL #24F0 ; draw ladder
10FD  C3F010 JP #10F0 ; increment hammer counter and return – jump to label BBBB

```

Gaat niet goed: trap verschijnt niet. Hammer counter ipv level laten zien?

```

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 #

```

Dus 06E1 laden met #65A0: 3AA065!

De hammer counter blijft bij grijpen van een hamer gewoon 0? Hoe kan dat? Het ophogen van de hammer counter werkt niet goed. Aangepast, nu wel goed. Alleen sparkle sprite blijft veel te lang staan en wordt na de ladder plaatsing ergens anders getekend.

Werkt nu goed. De trap verschijnt na pakken van de tweede hamer (volgorde maakt niet uit) en

Nu versie v0.22 opgeslagen.

Nu nog de hammers blinking maken. Code overnemen uit DK On The Run.

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

Vanuit de main springen naar onderstaande code:

```

1096 3A2762 LD A,(#6227) ; load A with screen number
1099 FE06 CP #06 ; is this screen X ?
109B CC0011 CALL Z, #1100 ; yes, call subroutine – call label AAAA

```

en

```

0A7B: 3A2762 LD A,(#6227) ; load A with screen number
0A7E: FE03 CP #03 ; are we on the elevators?
0A80: C8 RET Z ; yes, return
;
1100: 3A8563 LD A,(#6385) ; load A with blink counter – label AAAA
1103: 3C INC A ; increment blink counter
1104: 328563 LD (#6385), A ; store blink counter
;
1107: FE50 CP #50 ; is blink counter #50?
1109: CA1A11 JP Z, #111A ; yes: make hammer invisible – jump to label BBBB
;
110C: FEA0 CP #A0 ; is blink counter #A0?

```

```

110E: CA2311  JP Z, #1123      ; yes: make hammer visible – jump to label CCCC
;
1111: FEC0      CP #C0          ; is blink counter greater than #C0?
1113: D8        RET C          ; no return
1114: 3E00      LD A, #00        ; load A with #00
1116: 328563    LD (#6385), A   ; store blink counter
1119: C9        RET             ; return
;
111A: 3E07      LD A, #07        ; load A with value empty sprite – label BBBB
111C: 32196A    LD (#6A19), A   ; make first hammer invisible
111F: 321D6A    LD (#6A1D), A   ; make second hammer invisible
1122: C9        RET             ; return
;
1123: 3E1E      LD A, #1E        ; load A with hammer sprite – label CCCC
1125: 32196A    LD (#6A19), A   ; make first hammer visible
1128: 321D6A    LD (#6A1D), A   ; make second hammer visible
112B: C9        RET             ; return

```

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

Nu versie 0.23 opgeslagen.

Nu nog zo maken dat de hamers alleen gepakt kunnen worden wanneer de hamer zichtbaar is.

```

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

```

Dat dan vervangen:

```
2954 C32C11  JP #112C      ; jump to additional code – jump to label AAAA
```

en

```

112C: 3E0B  LD A,#0B    ; A := #0B = 1011 binary – label AAAA
112E: F7    RST #30     ; if level is elevators RET from this sub now.
112C: 3A196A LD A,(#6A19) ; load A with sprite value first hammer
112F: FE07   CP #07      ; check if sprite value is #07: invisible hammer
1131: C8     RET Z       ; return if invisible: no grab
1132: C35729 JP #2957    ; jump back to code and continue check hammer grab

```

De hamer kan nu alleen gegrepen worden als deze zichtbaar is. Interval zichtbaar kleiner maken zodat het moeilijker wordt: betere timing van de grab benodigd.

```

1107: FE50      CP #30          ; is blink counter #30?
110C: FEB0      CP #B0          ; is blink counter #B0?

```

Nu versie 0.24 opgeslagen.

Werkt nog niet goed. Zonder de hamers te pakken kan je al wel omhoog. Dat komt omdat de trappen apart gedefinieerd worden, los van het tekenen van het scherm. Voor alle velden met afwijkende trappen (dus nu screen 5 en screen 6 waarin de bovenste trap ontbreekt, moet bij het initieel definiëren van de trappen, het aangepaste veld gebruikt worden. Voor velden met dezelfde trappen, maakt het niet uit.

De definitie van de trappen per screen gebeurt hier:

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

2460 3D DEC A ; else is this the conveyors?
2461 215D3B LD HL,#3B5D ; load HL with start of table data for conveyors
2464 CA7124 JP Z,#2471 ; if conveyors, skip ahead

2467 3D DEC A ; else is this the elevators?
2468 21E53B LD HL,#3BE5 ; load HL with start of table data for elevators
246b CA7124 JP Z,#2471 ; if elevators, skip ahead

246E 218B3C LD HL,#3C8B ; otherwise we're on rivets. load HL with table data for rivets

2471 DD210063 LD IX,#6300 ; #6300 is used for ladder positions?
```

Dit is eerder aangepast door #245D een onconditionele jump te maken: C37124. Dus stuk code van #2460 t/m #2470 is in principe vrij. Hier de testen op screen 5 en 6 toevoegen met misschien ruimte voor meer afwijkende screens.

```
2456 3A2762 LD A,(#6227) ; load A with screen number
2459 21D63B LD HL,#3BD6 ; load HL with start of table data screen 5 and 6
245C FE05 CP #05 ; is this screen 5 ?
245E CA7124 JP Z,#2471 ; yes, define ladders
;
2461 FE06 CP #06 ; is this screen 6 ?
2463 CA7124 JP Z,#2471 ; yes, define ladders
;
2466 21E43A LD HL,#3AE4 ; load HL with start of table data of regular screen
;
2469 – 2470 NOP's
```

Ja dat werkt goed. Je kunt de onzichtbare trap aan het begin niet opklimmen. Pas na beide hamers gepakt te hebben kan dat.

Deze wijziging ook testen met screen 5. Dat is goed.

Wel vreemd dat in screen 5 de hamers nu ook blinking hammers geworden zijn. En je kunt ze alleen maar grijpen als ze zichtbaar zijn. Even kijken hoe dat met de ander levels is. Ja, bij alle levels nu blinking hammers die alleen te grijpen zijn als ze zichtbaar zijn.

Nu versie v0.25 opgeslagen.

Probleem met blinking hammers in andere screens komt door verkeerde opcode CALL Z,# is niet CD maar CC. Aangepast en nu alleen in screen 6 blinking hammers.

Nu versie v0.26 opgeslagen.

Screen 07: Remix reversed veld

Was :Dit is een nieuw veld waarin een girder naar boven en een girder naar beneden is geplaatst.
Maar: werkt niet. Daarom aangepast naar Remix reversed.

Deze definitie neerzetten op: #3BD6 t/m #3C49. Kopie van de definitie van het originele veld, met de nodige aanpassingen.

Veld 7 in de definitie opnemen.

0CBE FD 0C ; #0CFD for screen 7 uneven spaced girders

En dan de table data for screen 5 laden:

0CFD 114A3C LD DE,#3C4A ; Load DE with start of table data for screen 7

0CEF C3D00C JP #0CD0 ; jump back

De aanpassing aan de giders:

3C5E: 02 DF 9A 10 8E ; 3rd slanted girder

3C63: 02 EF AF 20 BB ; 4th slanted girder

De 3e moet omhoog (Y-waarde -8) en de 4e moet naar beneden (Y-waarde +8):

3C5E: 02 DF 92 10 86 ; 3rd slanted girder

3C63: 02 EF B7 20 C3 ; 4th slanted girder

De giders zijn aangepast, maar de barrels rollen nu na de eerste girder weer naar rechts omhoog. Dat is niet goed. Dit voorlopig eventjes laten zitten. Is lastig om dit goed te krijgen.

Daarom gaan voor een ander veld: het omgekeerde veld van Remix.

Omdat we het veld aanpassen (ook qua ladders, is het nodig dat we de ladderdefinitie ook opnemen).

Kan er nog tussen:

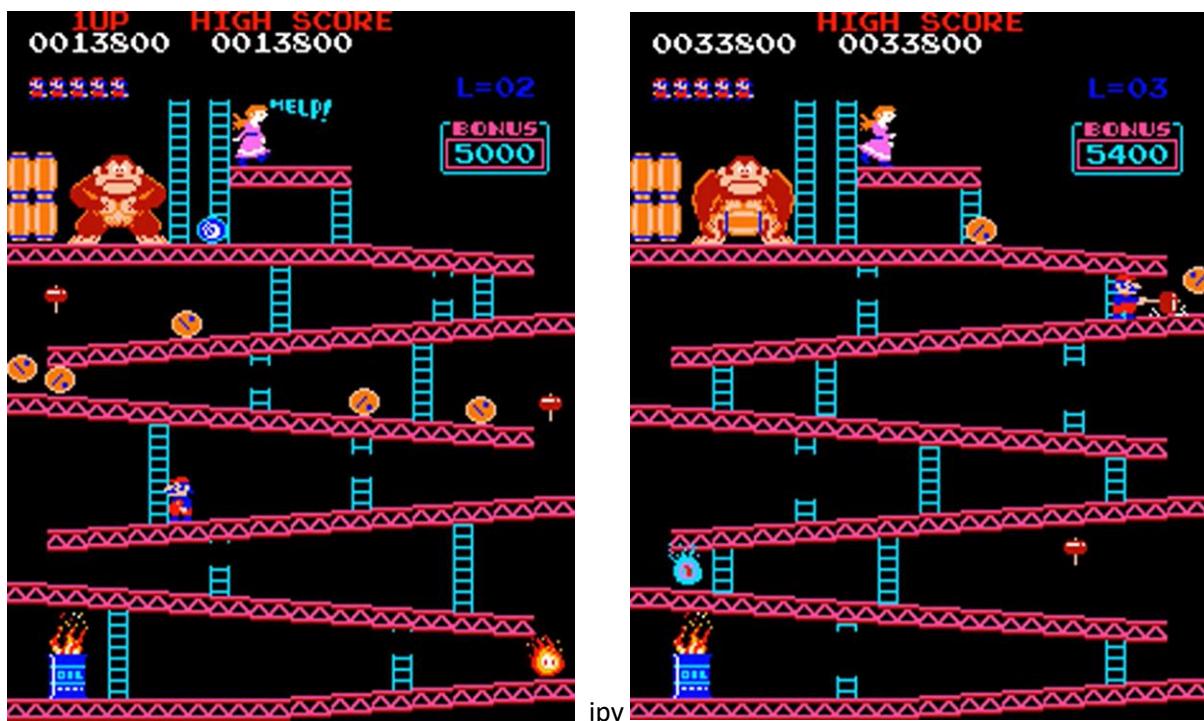
2456 3A2762 LD A,(#6227) ; load A with screen number

```

2459 21D63B LD HL,#3BD6 ; load HL with start of table data screen 5 and 6
245C FE05 CP #05 ; is this screen 5 ?
245E CA7124 JP Z,#2471 ; yes, define ladders
;
2461 FE06 CP #06 ; is this screen 6 ?
2463 CA7124 JP Z,#2471 ; yes, define ladders
;
2466 214A3C LD HL,#3C4A ; load HL with start of table data screen 7
2469 FE07 CP #07 ; is this screen 7 ?
246B CA7124 JP Z,#2471 ; yes, define ladders
;
246E 21E43A LD HL,#3AE4 ; load HL with start of table data of regular screen

```

Deze aangepaste girders terugzetten. Is gedaan. Het veld ziet er zo uit:



Dit is de lijst van trappen om eenvoudiger te kunnen aanpassen.

- 3B11: 00 CB 57 CB 6F ; short ladder at top right
- 3B16: 00 CB 99 CB B1 ; short ladder at center right
- 3B1B: 00 CB DB CB F3 ; short ladder at bottom right
- 3B20: 00 63 18 63 54 ; kong's ladder (right)
- 3B25: 01 63 D5 63 F8 ; bottom broken ladder
- 3B2A: 00 33 78 33 90 ; short ladder at left side under top hammer
- 3B2F: 00 33 BA 33 D2 ; short ladder at left side above oil can
- 3B34: 00 53 18 53 54 ; kong's ladder (left)
- 3B39: 01 53 92 53 B8 ; second broken ladder from bottom, on 3rd girder
- 3B3E: 00 5B 76 5B 92 ; longer ladder under the top left hammer

3B43: 00 73 B6 73 D6 ; longer ladder to left of bottom hammer

3B48: 00 83 95 83 B5 ; center longer ladder

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

3B52: 01 BB 70 BB 98 ; third broken ladder on right side near top

3B57: 01 6B 54 6B 75 ; fourth broken ladder near kong

Even alle trappen heel maken: dan geen probleem met teveel broken ladders.

Aanpassen bottom broken ladder: naar links en heel maken:

3C8B: 01 63 D5 63 F8 -> 00 3B D2 3B F8 : trap goed, alleen klimt door naar boven.

3C8B: 01 63 D5 63 F8 -> 00 43 D3 43 F8 : trap goed, alleen klimt door naar boven.

3C8B: 01 63 D5 63 F8 -> 00 4B D3 4B F8 : trap goed, alleen klimt door naar boven.

3C8B: 01 63 D5 63 F8 -> 00 53 D4 53 F8 : trap nu goed

Aanpassen short ladder at bottom right: broken maken:

3C81: 00 CB DB CB F3 -> 01 CB DB CB F3 : trap wel gebroken, maar ziet eruit als heel.

3C81: 00 CB DB CB F3 -> 01 BB DA BB F4 : trap wel gebroken, maar ziet eruit als heel.

3C81: 00 CB DB CB F3 -> 01 AB D9 AB F5 : trap wel gebroken, maar ziet eruit als heel.

3C81: 00 CB DB CB F3 -> 01 8B D7 8B F7 : trap nu goed.

Aanpassen longer ladder to left of bottom hammer: naar rechts verplaatsen.

3CA9: 00 73 B6 73 D6 -> 00 BB B2 BB DA : trap nu goed.

Aanpassen short ladder at left side above oil can: broken maken en naar rechts.

3C95: 00 33 BA 33 D2 -> 01 63 B7 63 D5 : trap nu goed.

Aanpassen second broken ladder from bottom, on 3rd girder: heel maken en naar links verplaatsen.

3C9F: 01 53 92 53 B8 -> 00 4B 91 4B B9 : trap nu goed:

Aanpassen short ladder at center right: broken maken en naar links.

3C7C: 00 CB 99 CB B1 -> 01 8B 95 8B B5 : trap nu goed

Aanpassen third broken ladder on right side near top: heel maken en naar links

3CB8: 01 BB 70 BB 98 -> 00 B3 70 B3 98 : trap nu goed

Aanpassen longer ladder under the top left hammer: broken maken en naar rechts

3CA4: 00 5B 76 5B 92 -> 01 73 74 73 94

Aanpassen fourth broken ladder near kong: heel maken en naar rechts

3CBD: 01 6B 54 6B 75 -> 00 7B 54 7B 74

Aanpassen center longer ladder: heel laten en helemaal naar links

-> 00 23 8F 23 BB

Aanpassen short ladder at top right: heel laten en naar rechts: nee gaat niet: terugzetten

3C77: 00 CB 57 CB 6F -> 00 DB 58 DB 6E

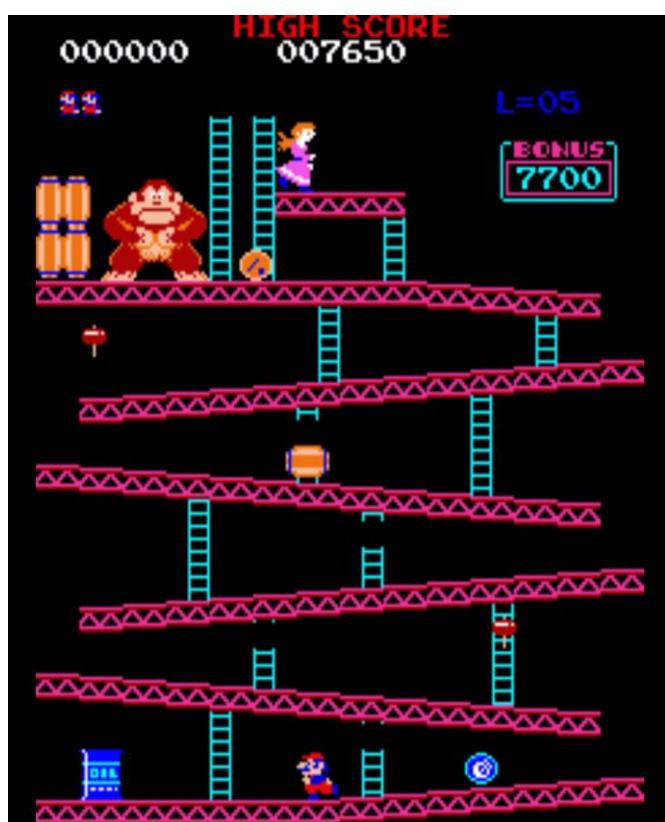
Weghalen center longer ladder

3CAE: 00 83 95 83 B5

Weghalen short ladder at left side under top hammer

3C90: 00 33 78 33 90 ;

Veld ziet er nu zo uit:



Nu versie 0.27 opgeslagen.

De hamers op andere plekken hangen en jumpman rechtsonder laten starten.

De hamerdefinitie wordt hier geladen:

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

3E0C is de hamer definitie van girders en 3E14 de hamer definitie van rivets.

Aanpassen:

```
1000 C3030D JP #0D03 ; jump to additional code – jump to label AAAAA
```

en

```
0D03 3A2762 LD A,(#6227) ; load A with screen number – label AAAAA  
0D06 FE07 CP #07 ; is this screen 7 ?  
0D08 210C3E LD HL,#3E0C ; load HL with table data for hammers regular screen  
0D0B C20310 JP NZ, #1003 ; not screen 7, jump back  
0D0E 21143E LD HL,#3E14 ; load HL with table data for hammers screen 7  
0D11 C30310 JP #1003 ; screen 7, jump back
```

en

3E14: 1B 8C 7C 64 aangepast naar 5B 9E A4 7D: hamers nu goed.

Nu versie 0.28 opgeslagen.

Wellicht dit scherm ook maken met gaten in de girders en smallere girders? Smaller girders niet doen. Gaten is wel een idee.

Hoe de gaten in screen 3 gemaakt? Kopie maken en juist toevoegen.

Call kan nog toegevoegd worden in het stuk code wat nog NOP's zijn, waar ook de bovenste girder stukken gewist worden.

```
0FD1 CD3511 CALL #1135 ; additional code to erase griders for screen 3
```

en

```
1135 3A2762 LD A,(#6227) ; load A with screen number  
1138 FE07 CP #07 ; is this screen 7 ?  
113A C0 RET NZ ; no, return  
;
```

```

113B 211676 LD HL,#7616 ; video memory address to clear
113E 3E10 LD A,#10 ; load a with space character
1140 77 LD (HL),A ; clear memory address
1141 23 INC HL ; next memory address, one lower
1142 77 LD (HL),A ; clear memory address

;
1143 217276 LD HL,#7672 ; video memory address to clear
1146 3E10 LD A,#10 ; load a with I character
1148 77 LD (HL),A ; clear memory address
1149 23 INC HL ; next memory address, one lower
114A 77 LD (HL),A ; clear memory address
;
114B C9 RET

```

Werkt goed.

Nu versie v0.29 opgeslagen.

Jumpman op een positie meer naar rechts laten beginnen. Dat is al eerder gedaan bij screen 5.

Initiële coördinaten van Jumpman worden hier gezet:

```

; set initial mario sprite position and draw remaining lives and level

123C DF RST #18 ; count down WaitTimerMSB and only continue when 0
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

```

Dit aanpassen naar:

```

#1240 FE05 CP #05 ; is this screen 5?
#1242 01984C LD BC,#4C98 ; Coordinates jumpman for screen 5, X=98 Y=4C.

```

Toen aangepast door screen 5 als elevators te definieren. Nu twee schermen die een alternatieve positie willen. Dit aanpassen naar:

```

1240 C34C11 JP #114C ; jump to additional code – jump to label AAAA
1243-124A NOP

```

en

```

114C FE05      CP    #05          ; is this screen 5? – label AAAA
114E 01984C    LD    BC,#4C98     ; coordinates jumpman for screen 5, X=98 Y=4C
1151 CA4B12    JP    Z,#124B     ; jump back
;
1154 FE07      CP    #07          ; is this screen 7?
1156 01E3E9    LD    BC,#E9E3     ; coordinates jumpman for screen 7, X=E3 Y= E9
1159 CA4B12    JP    Z,#124B     ; jump back
;
115C 013FF0    LD    BC,#F03F     ; coordinates jumpman for regular screens, X=3F Y=F0
115F C34B12    JP    #124B       ; jump back

```

Jumpman start nu inderdaad op de juiste plek aan de rechterkant.

Nu versie v0.30 opgeslagen.

Jumpman kijkt echter naar rechts. Het is beter als hij bij de start naar links kijkt. Waar wordt de startsprite voor jumpman geladen?

Dat wordt hier gedaan:

```

1252 DD360001 LD    (IX+#00),#01   ; turn on sprite
1256 DD7103  LD    (IX+#03),C      ; store X position
1259 71      LD    (HL),C        ; store X position
125A 2C      INC   L             ; next
125B DD360780 LD    (IX+#07),#80   ; store sprite graphic
125F 3680  LD    (HL),#80       ; store sprite graphic
1261 2C      INC   L             ; next
1262 DD360802 LD    (IX+#08),#02   ; store sprite color
1266 3602  LD    (HL),#02       ; store sprite color
1268 2C      INC   L             ; next
1269 DD7005  LD    (IX+#05),B     ; store Y position
126C 70      LD    (HL),B        ; store Y position
126D DD360F01 LD    (IX+#0F),#01   ; turn this on (????)

```

Dit aanpassen naar:

```
125B C36211  JP    #1162       ; jump to additional code – jump to label AAAA
```

en

```

1162 3A2762  LD    A,(#6227)    ; load A with screen number – label AAAA
1165 FE07    CP    #07          ; is this screen 7 ?
1167 2809  JR    Z,#1172     ; yes, jump to label BBBB
;
1169 DD360780 LD    (IX+#07),#80 ; store sprite graphic - jumpman facing right
116D 3680  LD    (HL),#80       ; store sprite graphic
116F C36112 JP    #1261       ; jump back

```

```

;
1172 DD360700 LD (IX+#07),#00 ; store sprite graphic - jumpman facing left – label BBBB
1176 3600 LD (HL),#00 ; store sprite graphic
1178 C36112 JP #1261 ; jump back

```

Dat gaat goed, jumpman kijkt nu aan het begin naar links.

Nu versie v0.31 opgeslagen.

Even testen of de andere screens het nog goed doen. Ja die doen het allemaal nog goed.

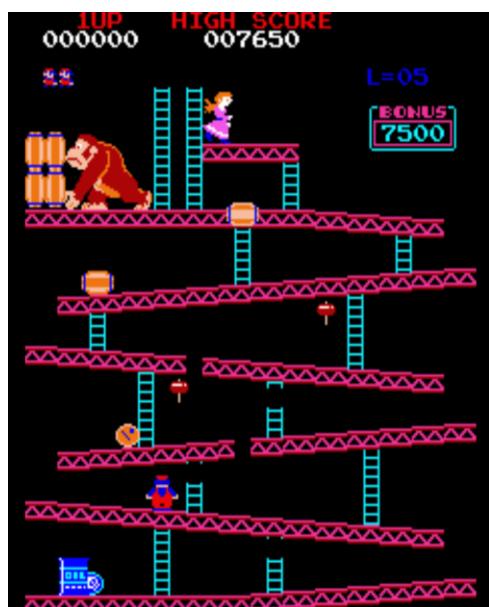
Het level is nog wel vrij eenvoudig lijkt het. Misschien lastiger als de bovenste kleine ladder opschuift.

Aanpassen longer ladder under the top left hammer: broken maken en naar rechts

3C9F: 00 5B 76 5B 92 -> 01 73 74 73 94

Dus die weer naar links verschuiven: dan lijkt hij eventjes heel, maar om te testen.

3C9F: 01 33 79 33 90



De gaten ook opschuiven zodat ze boven elkaar liggen:

```

1135 3A2762 LD A,(#6227) ; load A with screen number
1138 FE07 CP #07 ; is this screen 7 ?
113A C0 RET NZ ; no, return
;
113B 211676 LD HL,#7616 ; video memory address to clear
113E 3E10 LD A,#10 ; load a with space character
1140 77 LD (HL),A ; clear memory address

```

```

1141 23      INC  HL          ; next memory address, one lower
1142 77      LD   (HL),A    ; clear memory address

;

1143 211276  LD   HL,#7612  ; video memory address to clear
1146 3E10      LD   A,#10    ; load a with I character
1148 77      LD   (HL),A    ; clear memory address
1149 23      INC  HL          ; next memory address, one lower
114A 77      LD   (HL),A    ; clear memory address
;
114B C9      RET

```

Ook eens proberen met L=01. Dan eerste barrel recht naar beneden: spawned de fireball dan al?
Nee, daar kan je nog net voor langs.

Aanpassen start level:

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

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

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

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

Kan de lange trap beneden ook links naast de oil can getekend worden en dan beklimmen worden?

Aanpassen bottom broken ladder: naar links en heel maken:

3C8B: 01 63 D5 63 F8 -> 00 53 D4 53 F8 : trap nu goed

Aanpassen naar:

3C88: 00 1B D0 1B F8.

Dat gaat goed. Is wel erg leuk detail. Voorlopig zo laten.

Veld is zo goed. Alleen nog de kleine gebroken trap tekenen met alleen de bovenkant of de onderkant.

Nu versie v0.32 opgeslagen.

Het tekenen van de onderkant van de ladder niet doen. Dat gebeurt hier:

```

; this is a broken ladder. draw bottom part of ladder

0E3B 2D      DEC   L          ; decrease HL
0E3C 36C0    LD    (HL),#C0  ; set HL to #C0 - draws bottom part of broken ladder to screen
0E3E 2C      INC   L          ; increase HL

0E3F 3AB063  LD    A,(#63B0) ; load A with ???
0E42 FE00    CP    #00        ; == 0 ?
0E44 CA4B0E  JP    Z,#0E4B  ; yes, skip next 3 steps

0E47 C6E0    ADD   A,#E0    ; add #E0
0E49 2C      INC   L          ; next HL
0E4A 77      LD    (HL),A  ; store into ???

0E4B 13      INC   DE        ; next table entry
0E4C C3A70D  JP    #0DA7    ; loop again

```

En dit stukje code uit DK On The Run die ook checkt op de linker korte ladder x-positie:

```

OC4E: 3A2762  LD   A,(#6227) ; load a with screen number – label BBBB
OC51: 3D      DEC  A        ; is this the barrels?
OC52: C23B0E  JP   NZ,#0E3B ; no, draw normal broken ladder
OC55: D5      PUSH DE     ; store DE
OC56: 1B      DEC  DE     ; decrement DE, DE points to data-item with x-position
OC57: 1A      LD   A, (DE) ; A = x-position ladder
OC58: D1      POP  DE     ; restore DE
OC59: FE33    CP   #33    ; check if short ladder at left side
OC5B: CA3FOE  JP   Z, #0E3F ; yes, draw short broken ladder
OC5E: C33B0E  JP   #0E3B    ; no, draw normal broken ladder

```

Aanpassen:

```

0E3B C3A316  JP   #16A3    ; jump to additional code – jump to label AAAA
0E3E 00      NOP

```

en

```

16A3 3A2762  LD   A,(#6227) ; load A with screen number – label AAAA
16A6 FE07    CP   #07    ; is this screen 7 ?
16A8 2009    JR   NZ,#16B3 ; no, skip next steps – jump to label BBBB
;
16AA: D5      PUSH DE     ; store DE
16AB: 1B      DEC  DE     ; decrement DE, DE points to data-item with x-position
16AC: 1A      LD   A, (DE) ; A = x-position ladder
16AD: D1      POP  DE     ; restore DE
16AE: FE33    CP   #33    ; check if short ladder at left side
16B0: CA3FOE  JP   Z,#0E3F ; yes it is, jump back without drawing bottom part
;

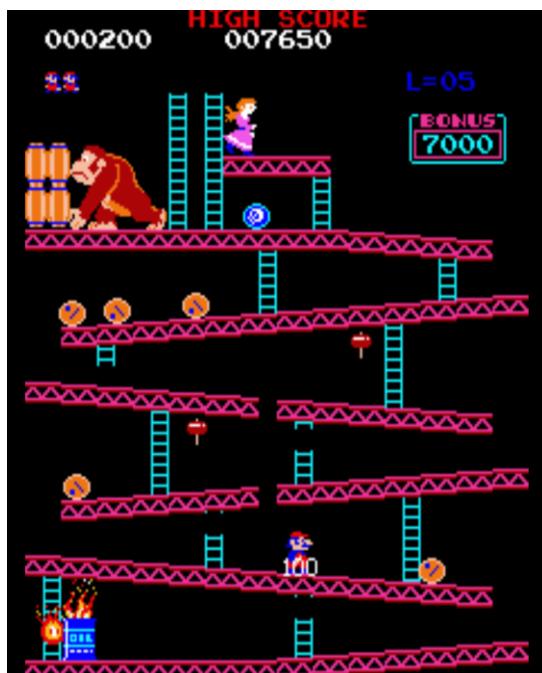
```

```

;
16B3 2D      DEC   L          ; decrease HL – label BBBB
16B4 36C0      LD    (HL),#C0   ; set HL to #C0 - draws bottom part of broken ladder
16B6 2C      INC   L          ; increase HL
16B7 C33F0E    JP    #0E3F    ; jump back

```

Dit gaat goed. Kleine ladder links wordt gebroken weergegeven. Het veld ziet er nu als volgt uit:



Nu versie v0.33 opgeslagen.

Gaat ook goed met de andere screens. Dus dat is prima.

Aanpassen screen 02 naar Into The Dark

Screen 2 definitie daarom naar een nieuw stukje code laten wijzen:

```
0CB4 14 0D      ; #0D14 for screen 2 Into The Dark
```

```
0D14 11E43A  LD  DE,#3AE4    ; Load DE with start of table data for screen 1
0D17 C3D00C  JP  #0CD0    ; jump back
```

Om eventjes te testen of dit inderdaad het originele scherm tekent. Dat gebeurt inderdaad. Alleen de barrels rollen weer terug. Dat komt door de code die de barrels eerder over de rand laat gaan in screen 2 en screen 3. Zodanig aanpassen dat dat alleen nog maar in screen 3 gebeurt.

```
04BE 3A2762  LD  A,(#6227)    ; load A with screen number – label AAAA
04C1 FE02  CP  #02        ; is this screen 2 ?
04C3 CAD604  JP  Z, #04D6    ; yes, jump to label BBBB
```

```
04C6 FE03      CP #03          ; is this screen 3 ?  
04C8 CAD604    JP Z, #04D6      ; yes, jump to label BBBB
```

en

```
04E1 3A2762    LD A,(#6227)    ; load A with screen number – label DDDD  
04E4 FE02      CP #02          ; is this screen 2 ?  
04E6 CAF904    JP Z, #04F9      ; yes, jump to label EEEE  
04E9 FE03      CP #03          ; is this screen 3 ?  
04EB CAF904    JP Z, #04F9      ; yes, jump to label EEEE
```

De doorgehaalde stukken NOPs maken zorgt ervoor dat we weer een normaal screen 2 hebben.

De palette's nu in gebruik zijn 'high score screen' 0 0 en 'barrel' 0 1

```
; Palettes:  
; A | B  
; -----  
; 0 | 0  high score screen  
; 0 | 1  barrel and elevator stages  
; 1 | 0  pie factory stage  
; 1 | 1  rivet stage
```

De andere twee gebruiken voor Into The Dark. 1 0 als de blauwe variant en 1 1 als de donkere variant.

Bij starten van screen 2 palette zetten op 1 0. Dat eerst testen.

```
0D14 11E43A  LD DE,#3AE4      ; Load DE with start of table data for screen 1  
;  
0D17 21867D  LD HL,REG_PALETTE_A ; load HL with palette bank selector  
0D1A 3601    LD (HL),#01      ; clear palette bank selector  
0D1C 23      INC HL          ; next bank  
0D1D 3600    LD (HL),#00      ; set palette bank selector  
;  
0D1F C3D00C  JP #0CD0        ; jump back
```

Dat werkt goed. En na veld wordt palette weer goedgezet voor het volgende scherm.

Nu nog de code toevoegen om het palette tijdens het spelen te laten uitgaan en de flashes te geven.

Voor de flashes worden twee timers gebruikt deze aan het begin initialiseren:

```
0D14 11E43A  LD DE,#3AE4      ; Load DE with start of table data for screen 1  
;  
0D17 21867D  LD HL,REG_PALETTE_A ; load HL with palette bank selector  
0D1A 3601    LD (HL),#01      ; clear palette bank selector
```

```

0D1C 23      INC   HL          ; next bank
0D1D 3600    LD    (HL),#00    ; set palette bank selector
;
0D1F: 328563 LD (#6385), A    ; store flash timer 1
0D22: 328D63 LD (#638D), A    ; store flash timer 2
;
0D25 C3D00C  JP   #0CD0      ; jump back

```

De code voor het flashen halen uit DK into The Dark:

In ieder geval aanroepen vanuit:

```

109E 3A2762  LD   A,(#6227)  ; load A with screen number
10A1 FE02    CP   #02        ; is this screen 2 ?
10A3 CCBA16  CALL  Z,#16BA   ; yes, call subroutine
10A6 C9      RET             ; return

```

en

```

16BA: 3A8D63 LD A,(#638D)    ; load A with flash timer 2
16BD: FE00    CP #00        ; is flash time 2 timer #00?
16BF  CAC716  JP Z, #16C7    ; yes, flash timer 2 ran out – jump to label GGGG
;
16C2: 3D      DEC A         ; decrement flash timer 2
16C3: 328D63 LD (#638D),A   ; store flash timer 2
16C6: C9      RET            ; return
;
16C7: 3A8563 LD A,(#6385)    ; load A with flash timer 1 - Label GGGG
16CA: 3C      INC A         ; increment flash timer 1
16CB 328563 LD (#6385), A   ; store flash timer 1
;
16CE: FED8    CP #D8        ; is flash timer 1 #D8?
16D0: CA4B18  JP Z, #184B    ; yes: make screen dark – jump to routine – label AAAA
;
16D3: FEC8    CP #C8        ; is flash timer 1 #C8?
16D5: CA4B18  JP Z, #184B    ; yes: make screen dark – jump to routine – label AAAA
;
16D8: FEC0    CP #C0        ; is flash timer 1 #C0?
16DA: CA5718  JP Z, #1857    ; yes: make screen visible - jump to label HHHH
;
16DD: FED0    CP #D0        ; is flash timer #D0?
16DF: CA5718  JP Z, #1857    ; yes: make screen visible – jump to label HHHH
;
16E2: FEF0    CP #F0        ; is flash timer greater than #F0?
16E4: D8      RET C         ; no return
;
16E5: 3E00    LD  A, #00      ; load A with #00
16E7: 328563 LD (#6385), A   ; store flash timer 1

```

```

;
16EA: 3A2962 LD A,#80 ; A = 80
16ED: 328D63 LD (#638D),A ; store flash timer 2
;
16F0: C9 RET ; return

```

En deze twee routines:

```

184B: 21877D LD HL,REG_PALETTE_B ; – label AAAA
184E: 3601 LD (HL),#01 ; set palette to dark
1850: C9 RET ; return
;
1851: 21877D LD HL,REG_PALETTE_B ; – label BBBB
1854: 3600 LD (HL),#00 ; set palette to visible
1856: C9 RET ; return

```

En deze routine:

```

1857: 3E03 LD A,#03 ; load sound duration of 3 – label HHHH
1859: 328260 LD (#6082),A ; play boom sound using sound buffer
185C: CA5118 JP Z, #1851 ; yes: make screen visible – jump to label BBBB

```

Dit gaat nu goed. Het andere palette wordt gekozen en de flashes worden gegeven naar het visible palette.

Nu versie v0.34 opgeslagen.

Bij behalen level en bij dood gaan moet het juiste palette weer gekozen worden. Daarnaast wanneer Donkey Kong op zijn borst slaat geen geluid maken.

Dit is het laatste stukje van de code wanneer een screen gehaald:

```

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]

```

Dit vervangen door;

```

1E85 C3F116 JP #16F1 ; jump to additonal code
1E88-1E8B NOP's

```

en

```

16F1 3A2762 LD A,(#6227) ; load A with screen number
16F3 FE02 CP #02 ; is this screen 2 ?
16F5 2003 JR NZ, #16FA ; no, skip next step
;

```

```

16F7: CD5118 CALL #1851      ; make screen visible
;
16FA: 3E16 LD A,#16      ; else A := #16
16FC: 320A60 LD (GameMode2),A ; store into game mode2
16FF: E1 POP HL      ; pop stack to get higher address
1700: C9 RET      ; return to a higher level [returns to #00D2]

```

Dit is een stukje van de code wanneer jumpman dood gaat:

```

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

```

Dit vervangen door:

```

12A3 CD5F18 CALL #185F      ; jump to additional code
12A6 3E03 LD A,#03      ; load A with duration of sound
12A8 328860 LD (#6088),A ; play death sound
12AB C9 RET      ; return

```

en

```

185F 3A2762 LD A,(#6227) ; load A with screen number
1862 FE02 CP #02      ; is this screen 2 ?
1864 2003 JR NZ, #16FA ; no, skip next step
;
1866: CD5118 CALL #1851      ; make screen visible
1869: CDBD30 CALL #30BD      ; clear sprites ?
186C: C9 RET      ; return

```

Nu het geluid uitzetten wanneer Donkey Kong op zijn borst slaat. Dat gebeurt 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

```

Dit aanpassen door:

```

044B C36D18 JP #186D      ; jump to additional code
044E 00 00 NOP NOP

```

en

```

186D 3A2762 LD A,(#6227) ; load A with screen number

```

```

1862 FE02      CP  #02          ; is this screen 2 ?
1864 C25004    JP  NZ, #0450    ; no, jump back, do not play sound
;
1867 3E03      LD  A,#03        ; load sound duration of 3
1869 328260    LD  (#6082),A   ; play boom sound using sound buffer
186C C35004    JP  #0450        ; jump back

```

Gaat nu allemaal goed.

Nu versie v0.35 opgeslagen.

En dan nu de palettes goedzetten.

Nu vaststellen welke velden gebruikt worden voor pies en rivets.

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

Groen: Barrels, Springs en Intro

Rood: Pies en Progression

Blauw: Rivets en Titel

Rood = Lights on variant en Blauw = Lights off variant.

Lights on: 2j: OF OF OF OE en 2k: OF OD OA OC.

Lights off: 2j: OF OF OF OE en 2k: OF OF OF OC.

Dat gaat goed. Echter nog wel uitdaging. De trap is echter nog lichtblauw. Die kleur blijft in het palette, want daarmee wordt o.a. de bonus timer getekend.

Nu versie v0.36 opgeslagen.

In Into The Dark worden de trappen met dezelfde kleur als de girders getekend. Hoe is dat gedaan? Heel simpel: door alle sprites met trapdelen aan te passen. Dat is nu niet mogelijk. Als je echter het blauw van de trappen ook zwart maakt, dan zie je de bonus timer ook niet meer!!! Maar eventjes

kijken wat we kunnen doen. Bonus timer heeft wel geluid. Dus dat zou je kunnen gebruiken om te weten hoeveel tijd je nog hebt (en tijdens de flitsen zichtbaar).

Dan palette aanpassen:

Lights off: 2j: OF OF OF OF en 2k: OF OF OF OF.

Dat werkt goed. Veld nog iets lastiger maken door de period tussen de flitsen langer te maken.

```
16EA: 3E8000 LD A,#80 ; A = 80
16ED: 328D63 LD (#638D), A ; store flash timer 2
```

Flits nu met een periode van 6 seconden. Voor nu wel lang genoeg.

Probleem is dat nu ook het How High scherm qua palette aangepast is. Misschien beter om lights on het standaard barrels palette te laten en alleen het licht uit te doen door naar rivets te schakelen.

Dat kan door het palette niet te wijzigen aan het begin:

```
;  
0D17 21867D LD HL,REG_PALETTE_A ; load HL with palette bank selector  
0D1A 3601 LD (HL),#01 ; clear palette bank selector  
0D1C 23 INC HL ; next bank  
0D1D 3600 LD (HL),#00 ; set palette bank selector  
;  
0D1F: 328563 LD (#6385), A ; store flash timer 1  
0D22: 328D63 LD (#638D), A ; store flash timer 2  
;  
0D25 C3D00C JP #0CD0 ; jump back
```

En door het switchen te doen naar het palette van rivets:

```
184B: 21867D LD HL,REG_PALETTE_A ; – label AAAA  
184E: 3601 LD (HL),#01 ; set palette to dark  
1850: C9 RET ; return  
;  
1851: 21867D LD HL,REG_PALETTE_A ; – label BBBB  
1854: 3600 LD (HL),#00 ; set palette to visible  
1856: C9 RET ; return
```

En dan het palette van het How High scherm weer te herstellen: herstellen van de rode definitie:

2j: OF 01 00 00 en 2k: OF 03 0A 00.

Gaat nu goed. Screen 2 is klaar.

Nu versie v0.37 opgeslagen.

In screen 3 Crazy Kong de bovenste twee trapdelen niet wissen

Het wissen van de trapdelen gebeurd hier:

```
0A98 3E10 LD A,#10 ; A := #10
0A9A 32A376 LD (#76A3),A ; erase a graphic near top of screen
0A9D 326376 LD (#7663),A ; erase a graphic near top of screen
```

Nee is niet de goede. Dit is van de intro waarbij Kong met Pauline naar boven klimt.

Het juiste stuk is:

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

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

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

Aanpassen:

```
0FB5 C37D18 JP #187D ; jump to additional code
```

en

```
187D 3A2762 LD A,(#6227) ; load A with screen number
1880 FE03 CP #03 ; is this screen 3 ?
1882 CACB0F JP Z,#0FCB ; no, jump back, do not erase top of ladder
;
1885 21006A LD HL,#6A00 ; yes, load HL sprite RAM – used for blank space sprite
1888 C3B80F JP #0FB8 ; jump back, do erase top of ladder
```

Dit gaat goed. Bij screen 3 blijven de twee ander gekleurde ladder toppen aanwezig.

Nu versie v0.38 opgeslagen.

Variabele speed testen.

```

1F72 3A2762 LD A, (#6227) ; load a with screen number
1F75 3D DEC a ; is this the girders ?
1F76 c0 RET NZ ; no, return

```

Dit is de check voor de barrel roll processing.

Aanpassen (eventjes voor alle screens) als test:

```
1F72 C38B18 JP #188B ; jump to additional code
```

en

```

188B 3A1A60 LD A,(FrameCounter) ; load A with frame counter ?
188E E603 AND #01 ; mask bits. random number between 0 and 1
1890 C8 RET Z ; if 0: return without processing barrel
1891 C3771F JP #1F77 ; continue processing barrel

```

Dat werkt wel. Bij een AND met 01 heel langzaam, en steeds sneller met 3, 5, 7 en F.

Kijken of dit met twee counters kan: #6385 is de counter die de snelheid bepaalt en #638D is de counter die telt tot de snelheid en dan één keer return geeft en de snelheid met 1 ophoogt. Totdat de snelheid een bepaalde waarde heeft en dan resetten.

Teller #6385 initialiseren met #00 en teller #638D initialiseren met #02 (1e byte is ophogen/verlagen, 2e byte is speed beginnend met 2).

Teller ophogen

Als teller is niet gelijk aan 2e byte van speed

Gewoon barrel processen

Anders:

Als 1e byte speed is 0 (speed ophogen)

2e byte van speed ophogen

Als speed is gelijk aan F

Zet 1e byte speed op 1 (speed verlagen)

Anders

2e byte van speed verlagen

Als speed is gelijk aan 2

Zet 1e byte speed op 0 (speed verhogen)

Zet teller weer op nul

Return zonder barrel processen

Voor nu eventjes laten rusten. Eerst een ander idee uitwerken. Veld maken waarin sommige barrels een lagere snelheid hebben dan andere barrels.

Screen 08 Random slow barrels

Het idee is om tijdens de deployment van een nieuwe barrel random barrels te kiezen die een afwijkende kleur krijgen (bv rood) en die langzamer rollen dan de andere barrels. Dit kan best voor interessante gameplay zorgen. Misschien zelfs leuker dan de variabele snelheids barrels.

Misschien kan onderstaande geheugenplek gebruikt worden:

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

En dan een ander bit gebruiken om een red barrel te identificeren?

Of wellicht mogelijk om de kleur gewoon op rood te zetten en daar dan op te checken?

Kleur zetten van de barrels gebeurd hier:

```
; IX holds 6700 +N*20 = start of barrel N info  
; a barrel is being deployed  
  
2CF6 DD360715 LD (IX+#07),#15 ; set barrel sprite value to #15  
2CFA DD36080B LD (IX+#08),#0B ; set barrel color to #0B  
2CFE DD361500 LD (IX+#15),#00 ; set +15 indicator to 0 = normal barrel, [1 = blue barrel]  
2D02 3A8263 LD A,(#6382) ; load A with Crazy/Blue barrel indicator  
2D05 07 RLCA ; is this a blue barrel ?  
2D06 D2152D JP NC,#2D15 ; No blue barrel, then skip next 3 steps  
  
; blue barrel  
  
2D09 DD360719 LD (IX+#07),#19 ; set sprite for blue barrel  
2D0D DD36080C LD (IX+#08),#0C ; set sprite color to blue  
2D11 DD361501 LD (IX+#15),#01 ; set blue barrel indicator
```

Dat betekent dat bij afhandeling van de barrel de IX+#08 gelezen moet kunnen worden.

```

; yes, we are on girders
; this subroutine checks the barrels, if any are rolling it does something, otherwise returns

1F77 DD210067 LD IX,#6700 ; load IX with start of barrel array
1F7B 218069 LD HL,#6980 ; load HL with start of sprites used for barrels
1F7E 112000 LD DE,#0020 ; load DE with offset of #20. used for checking next barrel
1F81 060A LD B,#0A ; for B = 1 to #0A ( do for each barrel)

1F83 DD7E00 LD A,(IX+#00) ; Load A with Barrel indicator (0 = no barrel, 2 = being deployed, 1=rolling)
1F86 3D DEC A ; Is this barrel rolling ?
1F87 CA931F JP Z,#1F93 ; Yes, jump ahead

1F8A 2C INC L ; otherwise increase L by 4
1F8B 2C INC L
1F8C 2C INC L
1F8D 2C INC L
1F8E DD19 ADD IX,DE ; Add offset to check for next barrel
1F90 10F1 DJNZ #1F83 ; Next B

1F92 C9 RET ; return

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

```

Ja dat moet kunnen. Afhandeling van een rolling barrel start op plek 1F93. Daar dan test tussenvoegen op screen 8 en als niet screen 8 dan terug naar de processing van de rolling barrel. Als wel screen 8 dan checken Framecounter en dan met een AND om de zoveel frames een return doen en de processing van de rolling barrel onderbreken en anders terug naar de processing van de rolling barrel.

Eerst barrels andere kleur geven.

```

2CFA C38B18 JP #188B ; jump to additional code – jump to label AAAA
2CFD 00 NOP

```

en

```

188B DD36080B LD (IX+#08),#0B ; set barrel color to #0B – label AAAA
188F 3A2762 LD A,(#6227) ; load A with screen number
1892 FE08 CP #08 ; is this screen 8 ?
1894 C2FE2C JP NZ, #2CFE ; no, jump back
;
1897 3A1860 LD A,(RngTimer1) ; load A with random number ?
189A E603 AND #0F ; mask bits. random number between 0 and F
189C C2FE2C JP NZ, #2CFE ; A is not 00, jump back
;
189F DD360807 LD (IX+#08),#07 ; set barrel color to #07
18A3 C3FE2C JP #2CFE ; jump back

```

Om dit te testen wel een screen 8 maken en daar starten.

Definitie van screen 8 is het normale barrels screen:

```
0CC0 E0 0C ; #OCE0 for screen 8 Random Slow Barrels
```

Dit werkt. Donkey Kong gooit random ook anders gekleurde barrels.

Nu nog de check inbouwen voor het langzamer laten rollen van de barrels.

```
1F93 C3A618 JP #18A6 ; jump to additional code – jump to label AAAA
```

en

```
18A6 3A2762 LD A,(#6227) ; load A with screen number – label AAAA
18A9 FE08 CP #08 ; is this screen 8 ?
18AB 200F JR NZ, 18BC ; no, skip next steps – jump to label BBBB
;
18AD DD7E08 LD A,(IX+#08) ; Load A with Barrel color
18B0 FE07 CP #07 ; is this a slow barrel?
18B2 2008 JR NZ,18BC ; no, skip next steps – jump to label BBBB
;
18B4 3A1A60 LD A,(FrameCounter) ; load A with frame counter ?
18B7 E603 AND #03 ; mask bits. random number between 0 and 3
18B9 CA8A1F JP Z,#1F8A ; if 0: return without processing barrel, do next
;
18BC DD7E01 LD A,(IX+#01) ; Load Crazy Barrel indicator – label BBBB
18BF C3961F JP #1F96 ; jump back and process barrel
```

Werkt wel. De anders gekleurde barrels gaan langzamer, maar die gaan ook wel schokkerig. Kan dat anders door een teller te introduceren die dan aan het begin van de barrel processing opgehoogd wordt en dan daar de AND op doen. Dan zeker dat 1 op 4 weggelaten wordt.

Het barrelboard begint met:

```
1F72 3A2762 LD A,(#6227) ; load a with screen number
1F75 3D DEC a ; is this the girders ?
1F76 c0 RET NZ ; no, return

; yes, we are on girders
; this subroutine checks the barrels, if any are rolling it does something, otherwise returns
```

Daarin is de RET NZ vervangen door een NOP: de code erna altijd uitvoeren. Maar in principe kan dit allemaal weg.

```
1F72 C3C218 JP #18C2 ; jump to additional code – jump to label AAAA
```

en

```

18C2 3A2762 LD A,(#6227) ; load A with screen number – label AAAA
18C5 FE08 CP #08 ; is this screen 8 ?
18C7 C2771F JP NZ, #1F77 ; no, jump back
;
18CA 218563 LD HL,#6385 ; load HL with address of counter
18CD 34 INC (HL) ; increment counter
18CE C3771F JP #1F77 ; jump back

```

en

```
18B4 3A8563 LD A, (#6385) ; load A with counter ?
```

Dat is net zo schokkerig als de framecounter gebruiken. Hoe kan dat? Komt hij niet altijd hier? Wat ook vreemd is, is dat trappen meestal langzaam, maar soms snel genomen worden. Hoe kan dat?

Debugging laat zien dat de counter soms een sprong maakt zonder dat de counter uitgelezen wordt. Hoe kan dat? Debugging laat zien dat soms de counter meerdere malen na elkaar verhoogd wordt (bovenstaande code), zonder dat hij uitgelezen wordt.

Nu versie 0.39 opgeslagen.

Kijken of we het op een alternatieve manier voor elkaar kunnen krijgen.

De teller ophogen doen in een apart stuk, los van de barrel processing.

Daarom terugzetten #1F72 t/m #1F76.

Daarna vanuit de main het ophogen van de counter aanroepen.

```

10A6 3A2762 LD A,(#6227) ; load A with screen number
10A9 FE08 CP #08 ; is this screen 8 ?
10AB CCC218 CALL Z, #18C2 ; yes, call subroutine

```

En de code die de teller ophoogt:

```

18C2 218563 LD HL,#6385 ; load HL with address of counter
18C5 34 INC (HL) ; increment counter
18C6 C9 RET ; return

```

Maakt geen verschil blijft schokkerig. En dat komt volgens mij niet door het overslaan van een deel van de stappen. Want het staat nu op 01, dus de helft van de stappen moet overgeslagen worden, maar je ziet beweging van meerdere stappen en dan een schok.

Eventjes testen hoe het eruit ziet als we de skip niet doen.

```

18B4 3A1A60 LD A,(FrameCounter) ; load A with frame counter ?
18B7 E603 AND #03 ; mask bits. random number between 0 and 3

```

```
18B9 CA8A1F    JP      Z,#1F8A          ; if 0: return without processing barrel, do next  
;  
18BC DD7E01    LD      A,(IX+#01)       ; Load Crazy Barrel indicator – label BBBB  
18BF C3961F    JP      #1F96          ; jump back and process barrel
```

18B9 NOP NOP NOP maken

Dan gaat het goed, maar zie je ook wat schokken (net als bij de gewone barrels die dat ook hebben).
Alleen lijkt vergroot als slow.

Ander ding te proberen. De test terug, maar omgekeerd: JP NZ i.p.v. JP Z.

18B9 C28A1F JP NZ,#1f8A ; if 1; return without processing barrel, do next

Maakt niets uit.

Tijdens het debuggen blijkt dat er een periode is, waarin de barrels wel behandeld worden, maar er geen barrel met de kleur 07 is (kolom IX+#08).

Kleuren 8B, 0C, 0B, 87, 0B, 8B: heel vreemd waar komt die 8 bij sommige vandaan? Wordt de kleur ergens tijdelijk veranderd? En zodra weer wel in de loop de check, dan staat er weer wel een met kleur 07 (die in barrel definitie van 6760). Lijkt wel of de byte ergens voor gebruikt wordt.

Ja, de code doet iets met de kleur: propt er een bit 7 in. Geen idee waarom. Daarom wellicht beter om te checken op het tweede byte. Dus eerste een AND met #0F doen voor je de kleur checkt.

Dit is de code voor het checken van de kleur:

```
18AD DD7E08 LD A,(IX+#08) ; Load A with Barrel color  
18B0 FE07 CP #07 ; is this a slow barrel?  
18B2 2008 JR NZ, #18BC ; no, skip next steps – jump to label BBBB  
;
```

Bug fix maken:

```
18B0 C3C718 JP #18C7 ; jump to additional code – jump to label AAAA  
18B3 NOP
```

en

```
18C7 E60F AND #0F ; get low byte  
18C9 FE07 CP #07 ; is this a slow barrel?  
18CB C2BC18 JP NZ, #18BC ; no, jump back, do not slow barrel  
18CE C3B418 JP #18B4 ; yes, jump back, do slow barrel
```

Nu werkt het wel goed. Maar de AND met 01 is wel heel erg langzaam. Aanpassen:

```
18B7 E603 AND #03 ; mask bits. random number between 0 and 3
```

Dit is prima. Hiermee testen qua speelbaarheid.

[Nu versie v0.40 opgeslagen.](#)

Er gebeuren vreemde dingen bij het herstarten. Soms 2 credits en dan springt naar begin met 0 credits. Komt dat omdat we de 6385 gebruiken?

Teruggaan naar versie met FrameCounter.

```
18B4 3A1A60 LD A,(FrameCounter) ; load A with frame counter ?
```

En dan is de code om de #6385 op te hogen ook niet meer nodig.

```
1F72 3A2762 LD A,(#6227) ; load a with screen number  
1F75 3D DEC a ; is this the girders ?  
1F76 c0 RET NZ ; no, return
```

En de andere code om het op te hogen:

```
10A6 3A2762 LD A,(#6227) ; load A with screen number  
10A9 FE08 CP #08 ; is this screen 8 ?
```

~~10AB CCC218 CALL Z, #18C2 ; yes, call subroutine~~

En de code die de teller ophoogt:

```
18C2 218563 LD HL,#6385 ; load HL with address of counter  
18C5 34 INC (HL) ; increment counter  
18C6 C9 RET ; return
```

Werkt leuk, maar nog meer variatie door sommige barrels nog langzamer te laten gaan met een andere kleur.

Opnieuw maken, met FrameCounter, met random keuze AND 0F voor kleur 07 en als deze dan gekozen dan nog random keuze AND 03 voor kleur 01. En dan checken kleur 07: als dat dan speed aanpassen met AND 03 op FrameCounter en anders checken kleur 01: als dat dan speed aanpassen met AND 01 op FrameCounter.

Terug naar versie v0.38.

Juiste veld tekenen:

```
0CC0 E0 0C ; #0CE0 for screen 8 Random Slow Barrels
```

Werkt.

Dan nu de kleuren van sommige barrels random aanpassen:

```
2CFA C38B18 JP #188B ; jump to additional code – jump to label AAAA  
2CFD 00 NOP
```

en

```
188B DD36080B LD (IX+#08),#0B ; set barrel color to #0B – label AAAA  
188F 3A2762 LD A,(#6227) ; load A with screen number  
1892 FE08 CP #08 ; is this screen 8 ?  
1894 C2FE2C JP NZ, #2CFE ; no, jump back  
;  
1897 3A1A60 LD A,(FrameCounter) ; load A with random number ?  
189A E60F AND #03 ; mask bits. random number between 0 and F  
189C C2FE2C JP NZ, #2CFE ; A is not 00, jump back  
;  
189F DD360807 LD (IX+#08),#07 ; set barrel color to #07  
;  
18A3 3A1860 LD A,(RngTimer1) ; load A with random number ?  
18A6 E603 AND #03 ; mask bits. Random number between 0 and 3  
18A8 C2FE2C JP NZ, #2CFE ; A is not 00, jump back  
;  
18AB DD360801 LD (IX+#08),#01 ; set barrel color to #01  
18AF C3FE2C JP #2CFE ; jump back
```

Dat gaat goed een aantal barrels met kleur 07 en soms een barrel met kleur 01.

Nu nog de check inbouwen voor het langzamer laten rollen van de barrels.

```
1F93 C3B218      JP    #18B2      ; jump to additional code – jump to label AAAA
```

en

```
18B2 3A2762      LD    A,(#6227)      ; load A with screen number – label AAAA
18B5 FE08        CP    #08          ; is this screen 8 ?
18B7 201E        JR    NZ, 18C8      ; no, skip next steps – jump to label CCCC
;
18B9 DD7E08      LD    A,(IX+#08)    ; Load A with Barrel color
18BC E60F        AND   #0F          ; get low byte
18BE FE07        CP    #07          ; is this a slow barrel?
18C0 200A        JR    NZ,18CC      ; no, skip next steps – jump to label BBBB
;
18C2 3A1A60      LD    A,(FrameCounter) ; load A with frame counter ?
18C5 E603        AND   #03          ; mask bits. random number between 0 and 3
18C7 CA8A1F      JP    Z,#1F8A      ; if 0: return without processing barrel, do next
18CA 180C        JR    #18C8      ; skip next steps – jump to label CCCC
;
18CC FE01        CP    #01          ; is this a super slow barrel? – label BBBB
18CE 2008        JR    NZ,18C8      ; no, skip next steps – jump to label CCCC
;
18D0 3A1A60      LD    A,(FrameCounter) ; load A with frame counter ?
18D3 E601        AND   #01          ; mask bits. random number between 0 and 1
18D5 CA8A1F      JP    Z,#1F8A      ; if 0: return without processing barrel, do next
;
18D8 DD7E01      LD    A,(IX+#01)    ; Load Crazy Barrel indicator – label CCCC
18DB C3961F      JP    #1F96      ; jump back and process barrel
```

Is goed zo. Mee gaan testen.

[Nu versie v0.41 opgeslagen.](#)

Leuk detail zou zijn als de barrel pile ook de andere kleuren zou weergeven.

De barrel pile wordt hier gedefinieerd: de derde waarde is de kleur van de barrel.

```
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
```

En deze definitie wordt hier gebruikt om de barrel pile te tekenen:

```

; arrive here when playing girders

0FD7 21DC3D LD HL, #3DDC ; source - has the information about the barrel pile at #3DDC
0FDA 11A869 LD DE, #69A8 ; destination = sprites
0FDD 011000 LD BC, #0010 ; counter is #10
0FE0 EDB0 LDIR ; draws the barrels pile next to kong

```

Ergens een alternatieve barrel pile definiëren en dan de aanroep aanpassen dat voor screen 8 de alternatieve barrel pile definitie geladen wordt.

```

18DE 1E 18 0B 4B ; first barrel
18E2 14 18 01 4B ; second barrel
18E6 1E 18 07 3B ; third barrel
18EA 14 18 0B 3B ; fourth barrel

```

en

```
0FD7 C3EE18 JP #18EE ; jump to additional code – jump to label AAAA
```

en

```

18EE 21DC3D LD HL, #3DDC ; load HL with information original barrel pile
18F1 3A2762 LD A, (#6227) ; load A with screen number – label AAAA
18F4 FE08 CP #08 ; is this screen 8 ?
18F6 C2DA0F JP NZ, #0FDA ; no, return
18F9 21DE18 LD HL, #18DE ; load HL with information alternative barrel pile
18FC C3DA0F JP #0FDA ; return

```

Ja ziet er goed uit dat detail.

Nu versie v0.42 opgeslagen.

Screen #09 Girders niet gelijkelijk verspreiden

Zelfde aanpassing als eerder, maar dan met andere waarden.

Deze definitie neerzetten op: #2501 t/m #2579. Kopie van de definitie van het originele veld, met de nodige aanpassingen.

Veld 9 in de definitie opnemen.

```
0CC2 28 0D ; #0D28 for screen 9 Uneven spaced girders
```

En dan de table data for screen 9 laden:

```

0D28 110125 LD DE, #2501 ; Load DE with start of table data for screen 9
0D2B C3D00C JP #0CD0 ; jump back

```

De aanpassing aan de girders:

2515: 02 DF 9A 10 8E ; 3rd slanted girder

251A: 02 EF AF 20 BB ; 4th slanted girder

De 3e moet omhoog (Y-waarde -8) en de 4e moet naar beneden (Y-waarde +8):

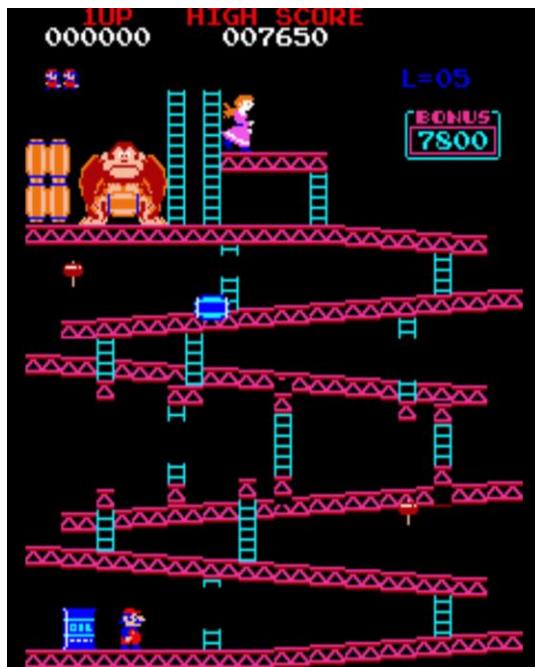
2515: 02 DF 92 10 86 ; 3rd slanted girder

251A: 02 EF B7 20 C3 ; 4th slanted girder

De giders zijn aangepast, maar de barrels rollen nu na de eerste girder weer naar rechts omhoog.
Dat is niet goed.

De trappen moeten nog goedgezet worden. Omdat ze aangepast zijn, moeten ze ook apart nog gedefinieerd worden. Zie blz 40 voor code die dit ook al doet voor andere schermen.

Het scherm ziet er nu zo uit:



De groene ladders moeten aangepast:

252E: 00 CB 57 CB 6F ; short ladder at top right

2533: 00 CB 99 CB B1 ; short ladder at center right: 00 CB 91 CB B9

2538: 00 CB DB CB F3 ; short ladder at bottom right

253D: 00 63 18 63 54 ; kong's ladder (right)

2542: 01 63 D5 63 F8 ; bottom broken ladder

2547: 00 33 78 33 90 ; short ladder at left side under top hammer: 00 33 78 33 88

254C: 00 33 BA 33 D2 ; short ladder at left side above oil can: 00 33 C2 33 D2

2551: 00 53 18 53 54 ; kong's ladder (left)

2556: 01 53 92 53 B8 ; second broken ladder from bottom, on 3rd girder: 01 53 8A 53 C0

255B: 00 5B 76 5B 92 ; longer ladder under the top left hammer: 00 5B 76 5B 8A

2560: 00 73 B6 73 D6 ; longer ladder to left of bottom hammer: 00 73 BE 73 D6
 2565: 00 83 95 83 B5 ; center longer ladder: 00 83 8D 83 BD
 256A: 00 93 38 93 54 ; ladder leading to girl
 256F: 01 BB 70 BB 98 ; third broken ladder on right side near top: 01 BB 70 BB 90
 2574: 01 6B 54 6B 75 ; fourth broken ladder near kong
 2579: AA ; AA code signals end of data

Het veld wordt nu getekend. Nu nog het juiste adres van het veld opgeven bij het definiëren van de trappen.

Nu versie v0.43 opgeslagen.

Code voor de ladder definitie.

```

2456 3A2762 LD A,(#6227) ; load A with screen number
2459 21D63B LD HL,#3BD6 ; load HL with start of table data screen 5 and 6
245C FE05 CP #05 ; is this screen 5 ?
245E CA7124 JP Z,#2471 ; yes, define ladders
;
2461 FE06 CP #06 ; is this screen 6 ?
2463 CA7124 JP Z,#2471 ; yes, define ladders
;
2466 214A3C LD HL,#3C4A ; load HL with start of table data screen 7
2469 FE07 CP #07 ; is this screen 7 ?
246B CA7124 JP Z,#2471 ; yes, define ladders
;
246E 21E43A LD HL,#3AE4 ; load HL with start of table data of regular screen
;
2471 – code voor het definiëren van de ladders.

```

Geen ruimte voor een extra ladder. Aanpassen:

```
246E C3FF18 JP #18FF ; jump to additional code – jump to label AAAA
```

en

```

18FF 210125 LD HL,#2501 ; load HL with start of table data screen 9
1902 FE09 CP #09 ; is this screen 9 ?
1904 CA7124 JP Z,#2471 ; yes, define ladders
;
1907 21E43A LD HL,#3AE4 ; load HL with start of table data of regular screen
190A C37124 JP #2471 ; jump back and define ladders

```

Jumpman kan goed over de girders lopen en de trappen beklimmen.

Als barrels via een ladder op de nieuwe girders komen dan rollen ze goed.

Alleen als ze op een nieuwe girder via de rand komen, dan gaat het niet goed. Lijkt een of andere index te zijn die niet goed gaat (negatief/positief). Gaat ook niet goed voor jumpman en de fireballs. Ook die gaan soms verkeerd.

De bijstelling voor de schuine girders wordt gedaan van #2333 t/m #236D. Deze code wordt aangeroepen voor de barrels, jumpman en de fireballs. Deze code wordt aangeroepen met in HL de x- en y-coordinaat (H=X en L=Y) en in B de richting (FF = links, 01 = rechts).

Er wordt daar inderdaad iets gedaan met de L waarde (de U-coordinaat). Er wordt gecheckt op bit 5: dat is de 16. Dus om de 16 (Y-coordinaat) een bepaald resultaat. Omdat nu de girders verplaatst zijn klopt dat niet meer. Het stuk houd ook rekening met de rechte stukken bovenin en onderin.

Dit is het stukje met de check op bit 5 van de L:

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

De twee verplaatste girders zitten nu op:

2515: 02 DF 92 10 86 ; 3rd slanted girder - van (10, 86) naar (DF, 92)
251A: 02 EF B7 20 C3 ; 4th slanted girder - van (20, C3) naar (EF, B7)

Het gaat onder andere fout bovenin. Kijken wat de waarde van jumpman Y-coördinaat daar is met de debugger. Y-coordinaat is #6205.

Helemaal links op de 3e girder (7E), helemaal rechts (8A).

Helemaal links op de 4e girder (BB), helemaal rechts (AF).

Testen met een aanpassing: de bit 5 vervangen door jump naar additionele code. Daar testen screen 9, zo niet dan de bit 5 doen en correct terug springen. Anders op screen 9. Dan testen Y kleiner dan 7E: dan bit 5 doen, dan testen Y groter dan BB: dan bit 5 doen. Dan testen Y kleiner dan 9F dan een resultaat van de bit 5 doen en anders het andere resultaat van de bit 5.

2353 C3511A JP #1A51 ; jump to additional code: jump to label AAAA

en

```
1A51 3A2762  LD     A,(#6227)      ; load A with screen number – label AAAA  
1A54 FE09    CP     #09          ; is this screen 9 ?  
1A56 2011    JR     NZ, #1A69      ; no continue original code – jump to label BBBB  
;  
1A58 7D      LD     A,L          ; load A with L = Y-position
```

```

1A59 FE7E      CP    #7E          ; check if Y-position is lower than #7E
1A5B 380C      JR    C, #1A69    ; yes: continue original code – jump to label BBBB
;
1A5D FEBC      CP    #BC          ; check if Y-position is higher than #BC
1A5F 3008      JR    NC, #1A69    ; yes: continue original code – jump to label BBBB
;
; Y-position in area of moved girders
;
1A61 FE9C      CP    #9C          ; check if Y-position is lower than #9C
1A63 DA5C23    JP    C, #235C    ; yes: perform no bit 5 – jump back 1
1A66 C35923    JP    #2359      ; no: perform no bit 5 – jump back 2
;
1A69 7D        LD    A,L        ; original code that performs the bit 5 – label BBBB
1A6A CB6F      BIT   5,A
1A6C CA5C23    JP    Z,#235C    ; jump back 1
1A6F C35923    JP    #2359      ; jump back 2

```

Gaat nu goed voor de bovenste girder. De onderste doet nog wel raar.

Nu versie v0.44 opgeslagen.

De onderste girder doet raar. Als bal over de rand, dan het eerste blokje rolt naar rechts en springt dan ineens terug naar girder erboven en rolt dan omhoog. Daarnaast gaat de fireball bovenin nog steeds zweven. Wordt de hamer gedropt als je van de girder springt. En vallen de barrels tussen de twee verschoven girders buiten beeld. Nogal wat te fixen dus.

Het rollen gaat op zich goed onderin. Alleen die rare sprong naar boven. Hoe groot is die sprong?

De girder waar van gesprongen wordt: 02 DF DC 10 D0

De girder waar naar gesprongen wordt: 02 EF B7 20 C3

Dat is een sprong van D0 naar C3 = -13 = -D = F3

De echte Y-coordinaat van de barrel daar beneden is CB.

Misschien ook eventjes testen met de onderste girder weer naar de originele plek terug. Dan heb je niet dat de barrels ernaast vallen en misschien het probleem met de girder jump. En anders de girder op een plek daartussen zetten. Proefondervindelijk testen.

Korte test met weer terugzetten van de onderste girder, leert dat ze nog steeds aan de rechterkant eraf vallen. Maar doordat de trappen niet aangepast zijn, komen ze via de trappen op de oude positie en rollen dan naar links. Daar nemen ze ronding nu wel goed (op dezelfde hoogte als eerst), terwijl ze op dezelfde hoogte rollen. Lijkt dus dat het rare gedrag veroorzaakt wordt doordat er een girder er vlak boven zit. Toch eens zoeken of er iets is dat daarnaar kijkt: naar aanwezigheid girder boven jumpman, firefox of barrel wat veroorzaakt dat er een offset naar boven komt.

De sprong omhoog van de barrel, komt door instructie op 2A7F. Geen idee wat die code doet, maar daar wordt wel aan de Y-coordinaat gesleuteld. Als je met debugger vertraagd afspeeld, lijkt het erop dat hier het stuiteren van de barrels geanimeerd wordt.

Vlak daarvoor staat er ook nog iets met het opvragen van het video character op die positie. Dat kan interessant zijn. Vermoeden is, dat het daar iets mee te maken heeft.

```

2A2F DD7E03 LD A,(IX+#03) ; load A with Barrel's X position
2A32 67 LD H,A ; Store into H
2A33 DD7E05 LD A,(IX+#05) ; load A with Barrel's Y position
2A36 C604 ADD A,#04 ; Add 4
2A38 6F LD L,A ; Store in L
2A39 E5 PUSH HL ; Save HL to stack
2A3A CDF02F CALL #2FF0 ; convert HL into VRAM memory address
2A3D D1 POP DE ; load DE with HL = barrel position X,Y
2A3E 7E LD A,(HL) ; load A with the graphic at this location
2A3F FEB0 CP #B0 ; < #B0 ?
2A41 DA7B2A JP C,#2A7B ; yes, skip ahead, clear A to 0 and return - nothing to do.

2A44 E60F AND #0F ; mask bits. now between 0 and #F
2A46 FE08 CP #08 ; < 8 ?
2A48 D27B2A JP NC,#2A7B ; no, skip ahead, clear A to 0 and return - nothing to do.

2A4B 7E LD A,(HL) ; load A with graphic at this location
2A4C FEC0 CP #C0 ; == girder with ladder on bottom going up ?
2A4E CA7B2A JP Z,#2A7B ; yes, clear A to 0 and return - nothing to do.

2A51 DA692A JP C,#2A69 ; < this value ? if so, skip ahead

2A54 FED0 CP #D0 ; > ladder graphic with girder under going up and out ?
2A56 DA6E2A JP C,#2A6E ; yes, skip ahead to handle

2A59 FEE0 CP #E0 ; > girder graphic going up and out ?
2A5B DA632A JP C,#2A63 ; yes, skip next 2 steps

2A5E FEF0 CP #F0 ; > girder graphic in several vertical phases coming up from bottom ?
2A60 DA6E2A JP C,#2A6E ; yes, skip ahead to handle

```

De x- en y-coördinaten van de barrel op het moment dat het fout gaat, waren (20,CA) en wordt dan (20,BE). Door dit stuk. Kun je dit handmatig doorlopen?

Wat wellicht kan is deze check op als die video characters niet doen voor screen 9 èn de Y-coordinaat is gelijk aan de girder die hij moet doorlopen?

Aanpassen:

```
2A2F C3721A JP #1A72 ; jump to additional code – jump to label AAAA
```

en

```

1A72 3A2762 LD A,(#6227) ; load A with screen number – label AAAA
1A75 FE09 CP #09 ; is this screen 9 ?
1A77 2006 JR NZ, #BBBB ; no continue original code – jump to label BBBB
;
```

```

1A79 DD7E05 LD A,(IX+#05) ; load A with barrel's Y position
1A7C FECA CP #CB ; is Y-position CB ?
1A7E C8 RET Z ; yes, skip this code: return
;
1A7F DD7E03 LD A,(IX+#03) ; load A with Barrel's X position – label BBBB
1A82 C3322A JP #2A32 ; jump back

```

Ja dit werkt nu wel goed.

[Nu versie 0.45 opgeslagen.](#)

Nu testen met invincibility uit. Hoe moeilijk zijn die lange ladders eigenlijk? Eventueel een extra ladder toevoegen om steer down te kunnen doen? Nee, gaat eigenlijk wel goed met level 05. Ook eventjes testen met andere levels (waar de stuurbaarheid minder is). Maar als je hamers gebruikt op level 1 dan komt de stuurbuurheid ook al snel.

Voor mijn gevoel kun je heel laat springen. Klopt dat? Vergelijken met andere levels. Level 01 en level 03 zijn ook te doen. Misschien wel te makkelijk?

Daarnaast nog aanpassen: fireball die te hoog loopt bovenin, barrels van scherm af en hamer's positie.

Getest met de hamer, maar dat is niet goed. Daarom de hamer op de girder lager neerhangen, helemaal rechts. En de hamer benede ook iets naar rechts. Daartoe een nieuwe hamer definitie laden. Dat is al voor een ander screen gedaan: screen 7. Voor screen 7 wordt de hamerdefinitie van rivets gebruikt:

```
1000 C3030D JP #0D03 ; jump to additional code – jump to label AAAA
```

en

```

0D03 3A2762 LD A,(#6227) ; load A with screen number – label AAAA
0D06 FE07 CP #07 ; is this screen 7 ?
0D08 210C3E LD HL,#3E0C ; load HL with table data for hammers regular screen
0D0B C20310 JP NZ, #1003 ; not screen 7, jump back
0D0E 21143E LD HL,#3E14 ; load HL with table data for hammers screen 7
0D11 C30310 JP #1003 ; screen 7, jump back

```

en

3E14: 1B 8C 7C 64 aangepast naar 5B 9E A4 7D: hamers nu goed.

Aanpassen:

```

0D03 3A2762 LD A,(#6227) ; load A with screen number – label AAAA
0D06 FE07 CP #07 ; is this screen 7 ?
0D08 21143E LD HL,#3E14 ; load HL with table data for hammers screen 7
0D0B CA0310 JP Z, #1003 ; yes screen 7, jump back

```

```
;  
0D0E FE09 CP #09 ; is this screen 9 ?  
0D10 C32E0D JP #0D2E ; jump to additional code – jump to label BBBB
```

en

```
0D2E 21103E LD HL,#3E10 ; load HL with table data for hammers screen 9 – label BBBB  
0D31 CA0310 JP Z, #1003 ; yes screen 9, jump back  
;  
0D34 210C3E LD HL,#3E0C ; load HL with table data for hammers regular screen  
0D37 C30310 JP #1003 ; jump back
```

En (hamer definitie hergebruikt van conveyors):

3E10 DB 7A DB C4

Dit werkt goed.

Nu versie v0.46 opgeslagen.

Zorgen dat de barrel op de 3e schuine girder eerder aan de rechterkant er vanaf valt:

```
2515: 02 DF 92 10 86 ; 3rd slanted girder  
251A: 02 EF B7 20 C3 ; 4th slanted girder
```

Even testen wat de barrel coördinaten daar zijn: Met debugger: (D0,8D). Testen op Y waarde en X-waarde (deze of de oude één blok naar links verschoven). Als daaraan voldaan dan de roll over doen.

Stukje code met aanpassingen van smaller girders is screen 2 en 3, wat daarna weer aangepast is nadat screen 2 verandert is naar Into The Dark, waardoor de testen op screen 2 weer verwijderd zijn.

```
2017 C3BE04 JP #04BE ; check barrel left edge, jump to label AAAA  
;  
201A C3E104 JP #04E1 ; check barrel right edge, jump to label DDDD – label CCCC  
;  
201D – 2023 NOP's  
;  
; Check barrel roll-over left edge  
;  
04BE 3A2762 LD A,(#6227) ; load A with screen number – label AAAA  
04C1 FE02 CP #02 ; is this screen 2 ?  
04C3 CAD604 JP Z, #04D6 ; yes, jump to label BBBB  
04C6 FE03 CP #03 ; is this screen 3 ?  
04C8 CAD604 JP Z, #04D6 ; yes, jump to label BBBB  
;  
; Check barrel left edge most screens  
;  
04CB DD7E03 LD A,(IX+#03) ; load A with Barrel's X position
```

```

04CE FE1C      CP  #1C
04D0 DA2F20    JP  C, #202F      ; yes, jump ahead to handle
04D3 C31A20    JP  #201A      ; continue check right edge, jump to label CCCC
;
; Check barrel left edge screens #2 and #3
;
04D6 DD7E03    LD  A,(IX+#03)    ; load A with Barrel's X position – label BBBB
04D9 FE2C      CP  #2C
04DB DA2F20    JP  C, #202F      ; yes, jump ahead to handle
04DE C31A20    JP  #201A      ; continue check right edge, jump to label CCCC
;
; Check barrel roll-over right edge
;
04E1 3A2762    LD  A,(#6227)    ; load A with screen number – label DDDD
04E4 FE02      CP  #02      ; is this screen 2 ?
04E6 CAF904    JP  Z, #04F9      ; yes, jump to label EEEE
04E9 FE03      CP  #03      ; is this screen 3 ?
04EB CAF904    JP  Z, #04F9      ; yes, jump to label EEEE
;
; Check barrel right edge most screens
;
04EE DD7E03    LD  A,(IX+#03)    ; load A with Barrel's X position
04F1 FEE4      CP  #E4
04F3 D22420    JP  NC, #2024      ; yes, jump ahead to handle
04F6 C3BA21    JP  #21BA      ; continue, jump way ahead
;
; Check barrel right edge screens #2 and #3
;
04F9 DD7E03    LD  A,(IX+#03)    ; load A with Barrel's X position – label EEEE
04FC FED4      CP  #D5
04FE D22420    JP  NC, #2024      ; yes, jump ahead to handle
0501 C3BA21    JP  #21BA      ; continue, jump way ahead

```

Aanpassen:

```

04E4 FE09      CP  #09      ; is this screen 9 ?
04E6 CA851A    JP  Z, #1A85      ; yes, jump to label FFFF
en

;
; Check barrel right edge screen #9
;
1A85 DD7E05    LD  A,(IX+#05)    ; load A with Barrel's Y position – label FFFF
1A88 FE8D      CP  #8D      ; Y-position is 8D ?
1A8A 200A      JR  NZ,#1A96      ; no, check for other roll over – jump to label GGGG

```

```

1A8C DD7E03 LD A,(IX+#03) ; load A with Barrel's X position
1A8F FED0 CP #D0 ; X-position > D0 ?
1A91 3803 JR C,#1A96 ; no, check for other roll over – jump to label GGGG
;
1A93 C32420 JP #2024 ; yes, jump ahead to handle right roll over on girder 3
;
1A96 DD7E03 LD A,(IX+#03) ; load A with Barrel's X position – label GGGG
1A99 FEE4 CP #E4
1A9B D22420 JP NC,#2024 ; yes, jump ahead to handle
1A9E C3BA21 JP #21BA ; continue, jump way ahead

```

Dit doet nu wel wat, maar gaat niet goed. In plaats van roll over, rolt de barrel terug omhoog. Dat kan komen omdat de girder nog niet ingekort is. Dat eerst doen.

2515: 02 DF 92 10 86 ; 3rd slanted girder aanpassen naar: 02 CF 91 10 86

Ja dat gaat goed.

De bovenste hamer nu niet te pakken. Daarom hamers naar links verplaatsen.

3E10 DB 7A DB C4 aangepast naar

Nu versie v0.47 opgeslagen.

Gaat toch nog niet helemaal goed. Barrels die aan de linkerkant op de onderste verplaatste girder (onder jumpman) uit beeld rollen komen rechts weer tevoorschijn. Blijven daar half buiten beeld ronddraaien. Totdat jumpman van links naar langs de grote trap in het midden loopt. Heel vreemd. Was dat ook al zo met de versie zonder ingekorte girder? Heeft dit iets te maken met logica die bepaalt of een barrel onder jumpman niet verder naar beneden mag rollen? Hoe werkt dat? Is dat nog gebaseerd op een even verdeling van de girders?

Eerst maar eens kijken naar de firefoxes bovenin. Komt wellicht doordat de y-waarde van de firefox steeds een klein beetje varieert om het animatie effect te vergroten. Dus wellicht op te lossen door de grenswaarde van de aanpassing iets bij te stellen: bovenkant grenswaarde iets naar beneden (dus Y-waarde groter)?

Dat is deze code:

```

1A59 FE7E CP #7E ; check if Y-position is lower than #7E
1A5B 380C JR C,#1A69 ; yes: continue original code – jump to label BBBB

```

OPgelost door de CP #7E te veranderen in CP #7C.

Nu versie v0.48 opgeslagen.

De barrels die niet goed van het scherm afrollen. Goed nadenken waarom dat is. Maar code die daar iets mee doet is:

```

; normal barrel traversed edge

20A9 210562 LD HL,#6205 ; load HL with mario's Y position address
20AC DD7E05 LD A,(IX+#05) ; load A with +5 = barrel's Y position
20AF D616 SUB #16 ; subtract #16
20B1 BE CP (HL) ; compare to mario Y position. is the barrel below mario?
20B2 D2C320 JP NC,#20C3 ; yes, skip next 5 steps

20B5 DD7E10 LD A,(IX+#10) ; load A with +10 = rolling over edge/direction indicator
20B8 A7 AND A ; A == 0 ? is this barrel is rolling right?
20B9 C2E120 JP NZ,#20E1 ; no, skip ahead and set alternate values, continue at #20C3

```

Wellicht te verhelpen door er meer dan 16 af te trekken. Proefondervindelijk testen.

Dat beïnvloed wel wanneer de barrels na de hoge val naar rechts van de girder af rollen. Maar ze verzamelen nog steeds aan de rechterkant.

Het lijkt erop dat het eraf rollen te ver door loopt. Ze lopen aan de linkerkant eraf, de x-positie wordt steeds lager en dan wellicht negatief. Waardoor ze aan de andere kant terecht komen? Waar wordt dat bepaald?

Lijkt erop dat aan de linkerkant als lager dan jumpman de barrel verder naar links rolt. Dan een extra test invoeren: als X=10 (buiten beeld, dan barrel vrijgeven).

```

04C1 FE09 CP #09 ; is this screen 9 ?
04C3 CAA11A JP Z, #1AA1 ; yes, jump to label HHHH

```

en

```

1AA1 DD7E03 LD A,(IX+#03) ; load A with Barrel's X position – label BBBB
1AA4 FE10 CP #10
1AA6 DA7920 JP C, #2079 ; yes, jump ahead to handle
1AA9 C3CB04 JP #04CB ; continue check left edge, jump to label CCCC

```

Gaat nu goed.

Nu versie v0.49 opgeslagen.

Nu gaan testen met de andere screens. Zijn die nog goed? En daarna met invincibility off om te kijken of de moeilijkheid goed is.

Screen 10 – Pauline's presents

Deze definitie neerzetten op: #257A t/m #????. Kopie van de definitie van het originele veld, met de nodige aanpassingen.

Veld 10 in de definitie opnemen.

```
OCC2 28 0D ; #0D28 for screen 10 Uneven spaced girders
```

En dan de table data for screen 9 laden:

```
0D3A 117A25 LD DE,#257A ; Load DE with start of table data for screen 9  
0D3D C3D00C JP #0CD0 ; jump back
```

Drie girders korter maken:

```
257A: 02 97 38 68 38 ; top girder where girl sits  
257F: 02 9F 54 10 54 ; girder where kong sits  
2584: 02 CF 57 A0 55 ; 1st slanted girder at top right  
2589: 02 EF 6D 20 79 ; 2nd slanted girder (has hammer at left side)  
258E: 02 CF 99 10 8E ; 3rd slanted girder  
2593: 02 EF AF 20 BB ; 4th slanted girder  
2598: 02 CF DB 10 D0 ; 5th slanted girder (has hammer at right side)  
259D: 02 FF F0 80 F7 ; bottom slanted girder  
25A2: 02 7F F8 00 F8 ; bottom flat girder where mario starts
```

En de roll over aan de rechterkant aanpassen. Lastig want de overgebleven vrije plekken van screen 2 zijn inmiddels alweer gebruikt door screen 9.

Dus daar iets tussen plaatsen.

Aanpassing screen 9 was:

```
;  
; Check barrel roll-over right edge  
;  
04E1 3A2762 LD A,(#6227) ; load A with screen number – label DDDD  
04E4 FE09 CP #09 ; is this screen 9 ?  
04E6 CA851A JP Z, #1A85 ; yes, jump to label FFFF
```

en

```
;  
; Check barrel right edge screen #9  
;  
1A85 DD7E05 LD A,(IX+#05) ; load A with Barrel's Y position – label FFFF  
1A88 FE8D CP #8D ; Y-position is 8D ?  
1A8A 200A JR NZ,#1A96 ; no, check for other roll over – jump to label GGGG
```

Nu aanpassen naar:

```
04E4 C30722 JP #2207 ; jump to additional code  
04E7 00 00 NOP, NOP
```

en

```

2207 FE09      CP  #09          ; is this screen 9 ?
2209 CA851A    JP  Z, #1A85     ; yes, jump to label FFFF
;
220C FE0A      CP  #0A          ; is this screen 10?
220E CAF904    JP  Z, #04F9     ; yes, perform barrel rollover more left
2211 C3E904    JP  #04E9       ; no, check for screen 3

```

Dat werkt goed. Ook nog even testen of screen 1, 3 en 9 nog steeds goed gaan.

Ja die gaan ook allemaal nog goed.

Nu versie v0.50 opgeslagen.

De drie extra smallle girders aan de rechterkant toevoegen:

```

25F2: 02 FF 59 E0 59 ; next to 1st slanted girder at top right
25F7: 02 FF 9B E0 9B ; next to 3rd slanted girder
25FC: 02 FF DD E0 DD ; next to 5th slanted girder (has hammer at right side)
2601: AA

```

Nu nog de items plaatsen.

Kan met de volgende code:

```

2214 3A2762    LD   A,(#6227)   ; load A with screen number – label DDDD
2217 FE0A      CP  #0A          ; is this screen 10 ?
2219 C0        RET  NZ         ; no, return
;
221A 213C3E    LD   HL,#3E3C     ; load HL with table data for bonus items on conveyors
221D 110C6A    LD   DE,#6A0C     ; load DE with sprite destination
2220 010C00    LD   BC,#000C     ; 3 items x 4 bytes = 12 bytes (#0C)
2223 EDB0      LDIR           ; draw bonus item sprites
2225 C9        RET             ; return

```

En aanroepen vanuit 0D7A: daar een stukje code vrij door verwijderen van specifiek stukje code voor rivets screen:

```
0D7A CD1422 CALL #2214      ; call additional routine to place the bonus items
```

De bonus items worden geplaatst, maar je kunt ze nog niet pakken.

```

3E3C 53 73 0A A0      ; position of hat
3E40 8B 74 0A F0      ; position of purse
3E44 DB 75 0A A0      ; position of umbrella

```

Nu versie v0.51 opgeslagen.

Posities aanpassen zodat de items op de juiste plek staan:

```

3E3C E5 73 0A D5      ; position of hat
3E40 8B 74 0A F0      ; position of purse
3E44 DB 75 0A A0      ; position of umbrella

```

Nu versie v0.52 opgeslagen.

De ladder naar Pauline verwijderen.

Nu versie v0.53 opgeslagen.

Aanpassingen om het aantal items te tellen en bij oppakken van alle drie, de ladder naar Pauline weer laten verschijnen. Zelfde als screen 6 met de knipperende hamers.

De item counter (#65A0)ophogen:

```

0D3A 3E00   LD A,#00    ; A := 0
0D3C 32A065 LD #65A0,A ; set item counter to 0
;
0D3F 117A25 LD DE,#257A ; Load DE with start of table data for screen 9
0D42 C3D00C JP #0CD0    ; jump back

```

En de item counter ophogen als item gepakt. Dat gebeurd hier:

```

; bonus item has been picked up

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

```

Aanpassen:

```

19FA C32622 JP #2226    ; jump to additional code – jump to label AAAA
19FD 00      NOP

```

en

```

2226 E5      PUSH HL     ; save HL for later
2227 3A2762  LD A,(#6227) ; load A with screen number – label AAAA
222A FE0A    CP #0A      ; is this screen 10 ?
222C C23322  JP NZ, #2233 ; no, jump forward, jump to label BBBB
222F 21A065  LD HL,#65A0 ; load HL with item counter
2232 34      INC (HL)    ; increment item counter
;
2233 E1      POP HL      ; restore HL - label BBBB
2234 2D      DEC L      ; decrease L
2235 224363  LD (#6343),HL ; store into this temp memory. read from at #1E18

```

```
2238 C3FE19    JP    #19FE          ; jump back
```

De code om de item counter te checken en bij 3 opgepakte items de ladder naar Pauline weer te tekenen ook overnemen van screen 6.

Aanroepen vanuit (is nog vrije positie hiervoor):

```
10A6 3A2762    LD   A,(#6227)    ; load A with screen number  
10A9 FE0A      CP   #0A        ; is this screen 10 ?  
10AB CC3B22    CALL Z,#223B     ; yes, call subroutine – call label AAAA  
10AE C9        RET
```

en

```
223B 3AA065    LD A, (#65A0)    ; load A with item counter  
;  
223E FE03      CP #03        ; compare A with 03  
2240 D8        RET C        ; A < 3 ? yes: return, do nothing  
2241 CA4D22    JP Z,#224D     ; A = 3 ? yes: set sparkle sprite – jump to label AAAA  
;  
2244 FE30      CP #30        ; compare A with 30  
2246 DA5B22    JP C,#225B     ; A < 30 ? yes: item counter++ – jump to label BBBB  
2249 CA6022    JP Z,#2260     ; A = 30 ? yes: create ladder – jump to label CCCC  
;  
224C C9        RET          ; A > 30 ! return  
;  
224D: 21206A    LD HL,#6A20    ; load HL with heart sprite – label AAAA  
2250: 3693      LD (HL),#93    ; set heart sprite x-position  
2252: 23        INC HL       ; next  
2253: 3663      LD (HL),#63    ; set sparkle sprite  
2255: 23        INC HL       ; next  
2256: 360C      LD (HL),#0C    ; set heart sprite color  
2258: 23        INC HL       ; next  
2259: 364A      LD HL,#4A     ; set heart sprite y-position  
;  
225B 21A065    LD HL,#65A0    ; load HL with item counter – label BBBB  
225E 34        INC (HL)     ; increment item counter  
225F C9        RET          ; return  
;  
2260 21216A    LD HL,#6A21    ; load HL with heart sprite – label CCCC  
2263 3664      LD (HL),#64    ; set blank sprite  
;  
2265 CDF024    CALL #24F0     ; draw ladder  
2268 C35B22    JP #225B     ; increment item counter and return – jump to label BBBB
```

Werkt goed. De ladder wordt naar het pakken van het derde item getekend. Maar net als eerder moet de ladder niet gedefinieerd worden aan het begin. Want je kunt de trap naar Pauline gewoon beklimmen ook wanneer deze niet zichtbaar is.

Code voor de ladder definitie.

```
2456 3A2762 LD A,(#6227) ; load A with screen number
2459 21D63B LD HL,#3BD6 ; load HL with start of table data screen 5 and 6
245C FE05 CP #05 ; is this screen 5 ?
245E CA7124 JP Z,#2471 ; yes, define ladders
;
2461 FE06 CP #06 ; is this screen 6 ?
2463 CA7124 JP Z,#2471 ; yes, define ladders
;
2466 214A3C LD HL,#3C4A ; load HL with start of table data screen 7
2469 FE07 CP #07 ; is this screen 7 ?
246B CA7124 JP Z,#2471 ; yes, define ladders
;
246E 21E43A LD HL,#3AE4 ; load HL with start of table data of regular screen
;
2471 – code voor het definiëren van de ladders.
```

Al eerder aangepast:

```
246E C3FF18 JP #18FF ; jump to additional code – jump to label AAAA
```

en

```
18FF 210125 LD HL,#2501 ; load HL with start of table data screen 9
1902 FE09 CP #09 ; is this screen 9 ?
1904 CA7124 JP Z,#2471 ; yes, define ladders
;
1907 21E43A LD HL,#3AE4 ; load HL with start of table data of regular screen
190A C37124 JP #2471 ; jump back and define ladders
```

Aanpassen:

```
1907 217A25 LD HL,#257A ; load HL with start of table data of screen 10
190A FE0A CP #0A ; is this screen 10 ?
190C CA7124 JP Z,#2471 ; yes, define ladders
;
190F 21E43A LD HL,#3AE4 ; load HL with start of table data of regular screen
1912 C37124 JP #2471 ; jump back and define ladders
```

Gaat nu goed.

Nu versie v0.54 opgeslagen.

Nu nog de drie items laten verschijnen als Pauline bereikt wordt (wanneer heart sprite getekend) en weer laten verdwijnen als Kong Pauline pakt (wanneer broken heart sprite getekend).

Daarvoor de item definitie van elevators gebruiken:

; 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 at 80,1B
```

```
; called from #1654 and #16A3
; clears all sounds, draws heart sprite, redraws girl sprite, clear "help", play end of level sound

1708 CD1C01 CALL #011C      ; clear all sounds
170B 21206A LD HL,#6A20      ; load HL with heart sprite
170E 3680 LD (HL),#80      ; set heart sprite X position
1710 23 INC HL             ; next
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
1717 3620 LD (HL),#20      ; set heart sprite Y position
1719 210569 LD HL,#6905      ; load HL with girl's sprite
171C 3613 LD (HL),#13      ; set girl's sprite
171E 21C475 LD HL,#75C4      ; load HL with VRAM screen address
1721 112000 LD DE,#0020      ; DE := #20
1724 3E10 LD A,#10          ; A := #10
1726 CD1405 CALL #0514      ; clear "help" that the girl yells
1729 218A60 LD HL,#608A      ; load sound address
172C 3607 LD (HL),#07      ; play sound for end of level
172E 23 INC HL             ; HL now has sound duration
172F 3603 LD (HL),#03      ; set duration to 3
```

Aanpassen:

```
1726 C36B22 JP #226B      ; jump to additional code – jump to label AAAA
```

en

```
226B CD1405 CALL #0514      ; clear "help" that the girl yells – label AAAA
226E 3A2762 LD A,(#6227)    ; load A with screen number
2271 FE0A CP #0A            ; is this screen 10 ?
2273 C22917 JP NZ,#1729      ; no, jump back
;
2276 21483E LD HL,#3E48      ; load HL with table data for bonus items on elevators
2279 110C6A LD DE,#6A0C      ; load DE with sprite destination
227C 010C00 LD BC,#000C      ; 3 items x 4 bytes = 12 bytes (#0C)
227F EDB0 LDIR             ; draw bonus item sprites
2281 C32917 JP #1729      ; jump back
```

De items worden wel getekend, maar verdwijnen vrijwel meteen weer.

Debugging toont aan dat dit op 30E5 gebeurt:

```
30D3 210C6A LD HL,#6A0C ; load HL with start of bonus items
30D6 0605 LD B,#05 ; B := 5
30D8 C3E430 JP #30E4 ; clear bonus items

; called from #12DF
; clears mario and elevators from the screen

30DB 214C69 LD HL,#694C ; load address for mario sprite X position
30DE 3600 LD (HL),#00 ; clear this memory = move mario off screen
30E0 2E58 LD L,#58 ; HL := #6958 = elevator sprite start
30E2 0606 LD B,#06 ; for B = 1 to 6

30E4 7D LD A,L ; load A with low byte addr

30E5 3600 LD (HL),#00 ; clear this sprite position to zero = move off screen
30E7 C604 ADD A,#04 ; add 4 for next sprite
30E9 6F LD L,A ; store into HL
30EA 10F9 DJNZ #30E5 ; next B

30EC C9 RET ; return
```

Testen door #30D8 in een RET te veranderen. Ja dat gaat goed, de bonus items blijven staan. Die dan nog weer verwijderen door bij het zetten van de heart sprite met het gebroken hart alsnog de bonus items te wissen: door een CALL naar #30D3 te doen.

Dat gebeurt hier:

```
174E 21216A LD HL,#6A21 ; load HL with heart sprite
1751 34 INC (HL) ; change heart to broken
1752 218863 LD HL,#6388 ; load HL with end of level counter
1755 34 INC (HL) ; increase counter
1756 C9 RET ; return
```

Aanpassen:

```
174E C38422 JP #2284 ; jump to additional code – jump to label AAAA
1751 00 NOP
```

en

```
2284 21216A LD HL,#6A21 ; load HL with heart sprite – label AAAA
2287 34 INC (HL) ; change heart to broken
;
2288 210C6A LD HL,#6A0C ; load HL with start of bonus items
```

```
228B 0605 LD B,#05 ; B := 5
228D CDE430 CALL #30E4 ; erase bonus items
2290 C35217 JP #1752 ; jump back
```

De hamers blijven nu staan. Die zitten waarschijnlijk na de bonus items? Klopt dat qua adressen?

Dan geen RET op #30D8, maar beter om

```
30D3 21186A LD HL,#6A18 ; load HL with start of hammers
30D6 0602 LD B,#02 ; B := 2
30D8 C3E430 JP #30E4 ; clear hammers
```

Nu versie v0.55 opgeslagen.

Alleen nog de purse en hat positioneren en een andere sprite voor de umbrella en die ook positioneren.

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

Bijna goed. Alleen Pauline sprite wordt ook aangepast op dat moment. Dat niet doen. De originele houden. Dat klopt bij einde scherm wordt Pauline sprite 13 gezet, terwijl 11 veel beter past bij het weergeven van de items.

```
1719 210569 LD HL,#6905 ; load HL with girl's sprite
171C 3613 LD (HL),#13 ; set girl's sprite
```

Aanpassen door:

```
1719 C39322 JP #2293 ; jump to additional code
```

en

```
2293 210569 LD HL,#6905 ; load HL with pauline's sprite
2296 3A2762 LD A,(#6227) ; load A with screen number
2299 FE0A CP #0A ; is this screen 10 ?
229B C21C17 JP NZ, #171C ; no, jump back
229E 3611 LD (HL),#11 ; set pauline's sprite
22A0 C31E17 JP #171E ; jump back
```

Gaat nu helemaal goed.

Nu versie v0.56 opgeslagen.

Leuk is als Pauline vier verschillende teksten random roept. Maar daarvoor wellicht het roepen (en wissen) aanpassen naar vier karakters. De teksten: Help!, Go!Go!, SaveMe!, Hurry!

De vier teksten zijn aangemaakt in de graphics file. Ze zijn nu vier karakters lang ipv drie:

```
#AC-#AF : Help !
#BC-#BF : SaveMe!
#CC-#CF : Hurry!
#DC-#DF : Go!Go!
```

Het tekenen van de tekst wordt door de onderstaande code gedaan. Deze lijkt generiek, maar wordt alleen aangeroepen voor het tekenen van de Help! tekst.

```
;  
; this sub gets called a lot  
; HL is preloaded with an address of video RAM ?  
; DE is preloaded with an offset to add  
; A is preloaded with a value to write  
; writes A into HL, A-1 into HL+DE, A-2 into HL+2DE  
;  
  
0514 0603 LD B,#03 ; for B = 1 to 3  
  
0516 77 LD (HL),A ; store A into memory  
0517 19 ADD HL,DE ; next memory  
0518 3D DEC A ; decrease A  
0519 10FB DJNZ #0516 ; next B  
  
051B C9 RET ; return
```

Omdat het nu vier karakters lang is, moet het volgende aangepast worden:

```
0514 0604 LD B,#04 ; for B = 1 to 4
```

Omdat er vanaf achteren getekend wordt, moeten de RAM video adressen voor de aanroep met #20 verlaagd worden.

Dit is de code voor het weergeven en wissen van de Help! tekst tijdens het spelen van het screen:

```

0495 79      LD     A,C          ; load A with the timer
0496 A7      AND    A          ; == 0 ?
0497 CAA104   JP     Z,#04A1      ; yes, skip next 3 steps

049A 3EEF      LD     A,#EF        ; else A := #EF
049C CB71      BIT    6,C          ; is bit 6 of the timer set ?
049E C2A304   JP     NZ,#04A3      ; no, skip next step

04A1 3E10      LD     A,#10        ; A := #10

04A3 21C475   LD     HL,#75C4      ; load HL with address of a location in video RAM where girl yells "HELP"
04A6 CD1405   CALL   #0514        ; update girl yelling "HELP"
04A9 3A0569   LD     A,(#6905)    ; load A with girl's sprite

```

Deze moet aangepast worden:

```

049A C3A322   JP     #22A3        ; jump to additional code – jump to label AAAA
049D 00000000 NOP, NOP, NOP, NOP

```

en

```

22A3 3A1A60   LD     A,(FrameCounter) ; load A with frame counter ? – label AAAA
22A6 E603     AND    #03          ; mask low 2 bits: number between 0 and 3
22A8 060A     LD     B, #0A        ; load B with 0A
22AA 80      ADD    A,B          ; add B to A: add #0A to A
22AB OF      RRCA           ; 4 time rotate left: swap high and low bytes
22AC OF      RRCA           ;
22AD OF      RRCA           ;
22AE OF      RRCA           ;
22AF F60F     OR     #0F          ; and #0F
22B1 CB71     BIT    6,C          ; is bit 6 of the timer set ?
22B3 C2A304   JP     NZ,#04A3      ; no, jump back skipping steps
22B6 C3A104   JP     #04A1        ; jump back

```

en

```

04A3 21A475   LD     HL,#75A4      ; load HL with adjusted VRAM screen address

```

Dit is de code voor het wissen van de Help! tekst aan het einde van het screen:

```

171E 21C475   LD     HL,#75C4      ; load HL with VRAM screen address
1721 112000   LD     DE,#0020      ; DE := #20
1724 3E10      LD     A,#10        ; A := #10
1726 CD1405   CALL   #0514        ; clear "help" that the girl yells

```

Deze moet aangepast worden naar:

```

171E 21A475   LD     HL,#75A4      ; load HL with adjusted VRAM screen address

```

Dit werkt wel, maar in de periode dat de tekst aanstaat, wordt deze steeds opnieuw aangezet. Maar omdat er dan een andere Framecounter geld, wordt er een andere tekst geplaatst. Dus alle vier de teksten worden heel snel cyclisch achter elkaar getoond.

Hoe kan dit aangepast worden?

```

0495 79      LD    A,C          ; load A with the timer
0496 A7      AND   A          ; == 0 ?
0497 CAA104   JP    Z,#04A1     ; yes, skip next 3 steps

049A 3EEF      LD    A,#EF        ; else A := #EF
049C CB71      BIT   6,C          ; is bit 6 of the timer set ?
049E C2A304   JP    NZ,#04A3     ; no, skip next step

04A1 3E10      LD    A,#10        ; A := #10

04A3 21C475   LD    HL,#75C4      ; load HL with address of a location in video RAM where girl yells "HELP"
04A6 CD1405   CALL  #0514        ; update girl yelling "HELP"
04A9 3A0569   LD    A,(#6905)    ; load A with girl's sprite

```

Mischien kan dat door na de check op bit 6 niet terug te springen naar #04A3, maar door terug te springen naar 04A9. Daartoe het volgende aanpassen:

```
22B3 C2A904   JP    NZ,#04A9      ; no, jump back skipping steps
```

Nee, dat gaat niet goed. Dan wordt er niets getoond.

Nog eens goed kijken wat deze code doet met de timer en de bit 6.

Als timer = 0 dan wordt de tekst gewist.

Als timer bit 6 is gezet dan wordt de tekst getekend.

Anders wordt de tekst gewist.

De timer loopt cyclisch van 0 to 7F (01111111). Bit 6 test op de 2e positie van links.

Dus tonen vanaf: 01000000 = #40 to 01111111 = 7F.

Dus de code kan anders:

```

22A3 FE40      CP    #40          ; A = #40 ? - label AAAA
22A5 CAB022    JP    Z,#22B0      ; yes, display help text – jump to label BBBB
;
22A8 FE7F      CP    #7F          ; A = #7F ?
22AA CA410A    JP    Z,#0A41      ; yes, jump back and remove help text
;
22AD C3A904    JP    #04A9        ; jump back without changing help text
;
22B0 3A1A60    LD    A,(FrameCounter) ; load A with frame counter ? – label BBBB
22B3 E603      AND   #03          ; mask low 2 bits: number between 0 and 3
22B5 060A      LD    B, #0A          ; load B with 0A
22B7 80        ADD   A,B          ; add B to A: add #0A to A
22B8 0F        RRCA           ; 4 time rotate left: swap high and low bytes

```

```
22B9 0F      RRCA          ;  
22BA 0F      RRCA          ;  
22BB 0F      RRCA          ;  
22BC F60F    OR   #0F      ; and #0F  
22BE C3A304  JP   #04A3      ; jump back and display help text
```

Gaat op zich qua weergeven en timing goed. Alleen ze roept vrijwel altijd SaveMe!, maar tijdens het hameren roept ze Help!.

Misschien is de Framecounter niet random genoeg in dit geval. Eventjes testen met RngTimer1: daartoe aanpassen #1A60 naar #1860. Ook nog niet. Beste resultaat met RngTimer2: #1960.

```
22B0 3A1960  LD   A,(RngTimer2) ; load A with random timer 2 ? – label BBBB
```

De SaveMe! tekst aanpassen naar Jump!

Nu versie v0.57 opgeslagen.

Er valt nog iets anders op. Nog eens naar kijken. Lijkt erop dat de items van beneden naar boven gepakt moeten worden. Dus de volgorde is niet vrij? Namelijk de middelste overgeslagen, maar dan kan ik de bovenste niet pakken. Terug naar de middelste en pas als die gepakt is, is de bovenste te pakken. Vreemd. Testen of dit ook het geval is, wanneer de onderste overgeslagen wordt.

Dit komt waarschijnlijk omdat de items dezelfde X-waarde hebben. 1e item checkt X-waarde: dan die processen en kijken naar Y. Als die niet klopt niet verder kijken. Dus dit krijg je alleen goed als ze 1 positie ten opzichte van elkaar verschoven worden.

De hat 1 naar rechts en de purse 2 naar rechts.

```
3E3C E6 73 0A D5      ; position of hat  
3E40 E7 74 0A 93      ; position of purse  
3E44 E5 75 0A 51      ; position of umbrella
```

Gaat nu wel goed. Items in willekeurige volgorde te pakken.

Nu versie v0.58 opgeslagen.

Aanpassen nog: 1) knipperende trap bij reverse veld (screen 5) en 2) extra gaten bij blinking hammers (screen 6).

De knipperende trap bij reverse veld is al aangepast. De tekst hier is verloren gegaan. Wellicht niet goed gesaved? In ieder geval dkong.zip wel opgeslagen.

Nu versie v0.59 opgeslagen.

Nu nog de extra gaten bij de blinking hammers in screen 6:

```
0FD4 CD2619  CALL #1926      ; additional code to erase griders for screen 6
```

en

```
1915 3A2762 LD A,(#6227) ; load A with screen number
1918 FE06 CP #06 ; is this screen 6 ?
191A C0 RET NZ ; no, return
;
191B 211676 LD HL,#7612 ; video memory address to clear
191E 3E10 LD A,#10 ; load a with space character
1920 77 LD (HL),A ; clear memory address
1921 23 INC HL ; next memory address, one lower
1922 77 LD (HL),A ; clear memory address

;
1923 217276 LD HL,#7672 ; video memory address to clear
1926 3E10 LD A,#10 ; load a with I character
1928 77 LD (HL),A ; clear memory address
1929 23 INC HL ; next memory address, one lower
192A 77 LD (HL),A ; clear memory address
192B C9 RET
```

Op OFD4 was de laatste plek. Als nu nog een screen met een dergelijke definitie, dan op OFD4 springen naar een apart stuk waar de dan deze CALL en eventuele additionele CALL's kunt doen en dan terug kunt springen.

Er gebeuren rare dingen nu. Bij dood en starten nieuw spel, start soms niet, rare kleuren, weergeven rare sprites, enz. Kijken vanaf welke versie dat is. Versie v0.59 heeft het probleem ook. Versie v0.58 heeft het probleem ook.

Het lijkt er op dat dit alleen gebeurd wanneer screen 6 gespeeld wordt. Klopt dat? Ja daar lijkt het wel op. Dit moeten we later eens goed onderzoeken waarom dit gebeurd. Dit gebeurde al voor de aanpassing van zojuist. Dus daar ligt het niet aan. Dus eerst een keer het probleem tackelen en daarna nog kijken of er nog gaten in de girders gewenst zijn.

Lijkt te gebeuren door de code die hier aangeroepen wordt:

```
1096 3A2762 LD A,(#6227) ; load A with screen number
1099 FE06 CP #06 ; is this screen 6 ?
109B CC0011 CALL Z, #1100 ; yes, call subroutine – make hammer blink
```

Als deze namelijk allemaal NOP gemaakt dan treedt het probleem niet meer op.

In de aangeroepen code wordt #6385 gebruikt als counter. Maar die wordt ook in het introscherm gebruikt. Dat hebben we nog steeds, dus wellicht staat deze counter dan niet goed. Maar deze wordt ook in andere screens gebruikt.

Bijvoorbeeld in screen 8 met de slow en super slow barrels. Daar zou het probleem dus ook kunnen optreden. Nee dat lijkt goed te gaan. Kan kloppen omdat in later stadium deze teller er weer

uitgehaald is en teruggegaan is naar eenenvoudiger mechanisme met checken van de FrameCounter.

Bijvoorbeeld in screen 2 van Into The Dark waarbij het gebruikt wordt voor de flash timer. Daar zou het probleem dus ook kunnen optreden. En daar treedt het ook op.

Voor nu eventjes negeren en de schermen afmaken. Aan het einde het intro screen weghalen zodat dit geen probleem meer is.

Dus eventjes terug naar versie v0.59 en opnieuw de gaten in screen 6 maken en ermee testen wat de beste of interessantste combinatie is.

Is gedaan. Screen 6 heeft nu twee gaten dicht bij elkaar. Ziet er goed uit en is interessant om te spelen.

Nu versie v0.60 opgeslagen.

Screen 11 – springs moving vertically

Het normale screen daarvoor gebruiken. Voor nu daarom springen naar de code die het normale screen tekent. Wellicht later additionele setup benodigd, die dan nog toevoegen.

0CC6 E0 0C ; #OCE0 for screen 11 Springs

6500 is het startadres voor de springs data en #6980 is het startadres voor de springs sprites.

Zie #2E04 voor verschillende indexen, zodat duidelijk is wat de verschillende velden in de datastructuur betekenen.

#2E04 t/m 2ED3 is code voor bouncers voor het elevators screen. Deze code was nog bewaard, maar is niet meer nodig. Die vrijgeven en gaan gebruiken voor nieuwe spring code.

Start van nieuwe code kan vanaf #2DDB.

Dus voor screen 11 een aanroep naar #2DDB.

```
10AE 3A2762 LD A,(#6227) ; load A with screen number
10B1 FE0B CP #0B ; is this screen 11 ?
10B3 CCDB2D CALL Z, #2DDB ; yes, call subroutine
10B6 C9 RET ; return
```

Dus de sprite waarden:

#6980 – sprite X value
#6981 – sprite number
#6982 – sprite color
#6983 – sprite Y value

Eerst een stukje code om de sprite te tekenen:

```
2DDB 218069 LD HL,#6980 ; load HL with spring sprite
2DDE 36A7 LD (HL),#A7 ; set spring sprite X position
2DE0 23 INC HL ; next
2DE1 363B LD (HL),#3B ; set spring sprite
2DE3 23 INC HL ; next
2DE4 3609 LD (HL),#00 ; set spring sprite color
2DE6 23 INC HL ; next
2DE7 3680 LD (HL),#80 ; set spring sprite Y position
```

Gaat goed. Nu de sprite laten bewegen.

```
2DDB 218069 LD HL,#6980 ; load HL with spring sprite
2DDE 36A7 LD (HL),#80 ; set spring sprite X position
2DE0 23 INC HL ; next
2DE1 363B LD (HL),#3B ; set spring sprite
2DE3 23 INC HL ; next
2DE4 3600 LD (HL),#00 ; set spring sprite color
2DE6 23 INC HL ; next
2DE7 34 INC (HL) ; increase spring sprite Y position
```

Gaat raar schokkerig. Geen idee hoe dat komt. Eventjes anders doen:

```
2DDB 218069 LD HL,#6980 ; load HL with spring sprite
2DDE 36A7 LD (HL),#80 ; set spring sprite X position
2DE0 23 INC HL ; next
2DE1 363B LD (HL),#3B ; set spring sprite
2DE3 23 INC HL ; next
2DE4 3600 LD (HL),#00 ; set spring sprite color
2DE7 23 INC HL ; next
2DE8 7E LD A,(HL)
2DE9 3C INC A
2DEA 77 LD (HL),A
```

Gaat net zo raar schokkerig. Wordt de Y op één of andere manier nog meer aangepast. Debuggen met watchpoint op #6983.

Grappig. Het schokkerige is voornamelijk daar waar de spring bij een girder zit. Het lijkt erop dat de spring als een barrel gezien wordt (lijkt beetje op afremmen wanneer een wild barrel de girder raakt).

Lijkt te kloppen. Als je in de code kijkt dan is #6890 ook het begin van de sprites die voor de barrels gebruikt worden. Betykt dit dan ook dat de eerste blue barrel nu vreemd gedrag vertoont? Ja dat klopt. Die wordt niet gegooid.

Dus ander sprite geheugen gebruiken. Gebruiken de pies een ander deel? Lijkt er wel op: beginnend bij #69B8. Eventjes testen daarmee. Ja gaat nu wel goed.

De snelheid is vrij langzaam. Als 3x INC wordt gedaan, dan interessante snelheid. Maar uiteindelijk zo maken dat de snelheid nog getuned kan worden.

Eventjes kijken of de springs datastructure niet ook hetzelfde is als voor de barrels. Anders daarvoor ook de pies gebruiken? Misschien sowieso handig, omdat je die kan kapotslaan. Maar misschien wil je dat wel niet? Hoe handel je dat af dan. Maar dat zien we later dan wel.

#6500 – spring status (00 = inactive, 01 = active)

#6503 – spring X position

#6505 – spring Y position

#650D – spring horizontal or vertical direction

Ook deze zetten. En kijken of er collisions komen.

```
2DDB 218069 LD HL,#6980 ; load HL with spring sprite
2DDE 36A7 LD (HL),#80 ; set spring sprite X position
2DE0 23 INC HL ; next
2DE1 363B LD (HL),#3B ; set spring sprite
2DE3 23 INC HL ; next
2DE4 3600 LD (HL),#00 ; set spring sprite color
2DE6 23 INC HL ; next
2DE7 34 INC (HL) ; increase spring sprite Y position
2DE8 34 INC (HL) ; increase spring sprite Y position
2DE9 34 INC (HL) ; increase spring sprite Y position
;
2DEA DD210065 LD IX,#6500 ; load IX with start spring table
2DEE DD360001 LD (IX+#00),#01 ; set spring active
2DF2 DD360380 LD (IX+#03),#80 ; set spring X position
2DF6 7E LD A,(HL)
2DF7 DD7705 LD (IX+#05),A ; set spring Y-position
2DFA C9 RET ; return
```

De table wordt wel gevuld (zichtbaar in debugger), maar je gaat niet dood wanneer je door de spring geraakt wordt. Staat misschien de collision detection daarvoor uit?

Ja dat klopt. De screen check van #2873 t/m #287F is uitgezet en alleen de check voor girders wordt gedaan door middel van de code van #2880 t/m #28AF. Daar zit een check op collision with springs niet tussen. Die voor de C9 op #28AF toevoegen:

```
28AF DD210065 LD IX,#6500 ; load IX with start of addresses for springs
28B3 CD1329 CALL #2913 ; check for collisions with springs
28B6 C9
```

Dit gaat niet goed. Gaat steeds langzamer spelen. Wat checkt hij allemaal. Hoeft eigenlijk alleen de eerste spring. B bepaalt het aantal iteraties (aantal springs).

```

28AF 3A2762 LD A,(#6227) ; load A with screen number
28B2 FEOB CP #0B ; is this screen 11 ?
28B4 C0 RET NZ ; no, return
;
28B5 0601 LD B,#01 ; B := #01
28B7 DD210065 LD IX,#6500 ; load IX with start of addresses for springs
28BB CD1329 CALL #2913 ; check for collisions with springs
28BE C9 RET ; return

```

Gaat nog niet goed. Je wordt niet geraakt door de spring. Maar dat komt misschien omdat de hitbox niet gedefinieerd is (en leeg is width and height van 0).

Aanpassen:

```

2DEA DD210065 LD IX,#6500 ; load IX with start spring table
2DEE DD360001 LD (IX+#00),#01 ; set spring active
2DF2 DD360380 LD (IX+#03),#80 ; set spring X position
2DF6 DD360908 LD (IX+#09),#08 ; set spring size width 08
2DFA DD360A08 LD (IX+#0A),#08 ; set spring size height 08
2DFE 7E LD A,(HL)
2DFF DD7705 LD (IX+#05),A ; set spring Y-position
2E02 C9 RET ; return

```

Gaat ook nog niet goed. Misschien nog parameters goedzetten voor de aanroep.

Aanpassen:

```

28AF 3A2762 LD A,(#6227) ; load A with screen number
28B2 FEOB CP #0B ; is this screen 11 ?
28B4 C0 RET NZ ; no, return
;
28B5 0601 LD B,#01 ; B := 1
28B7 78 LD A,B ; A := 1
28B8 32B963 LD (#63B9),A ; store counter for use later
28BB 1E00 LD E,#00 ; E := #00
28BD DD210065 LD IX,#6500 ; load IX with spring data location
28C1 CD1329 CALL #2913 ; check for collision with spring
28C4 C9 RET ; return

```

Gaat nog steeds niet goed. Eventjes goed kijken hoe dit precies werkt. Anders wellicht pies gebruiken. Die ook in Twisted Jungle gebruikt.

De originele springs worden initieel gevuld met IX+#07=3B, IX+#09=02 en IX+#0A=02. Eens testen met deze width en heights. Maakt geen verschil.

Wel steeds af op dezelfde X-positie. Klopt dat wel: de X positie van de sprite en de X-positie van de data wel hetzelfde? Even debuggen: X spring sprite = #6980 en X spring data = #6503.

Rare is dat 6503 80 is (dat is midden van scherm, dus dat klopt niet). Moet A7 zijn. Daar eens mee testen.

2DF2 DD3603A7 LD (IX+#03),#A7 ; set spring X position

Ja dat was het, is nu wel goed. Nu nog testen met width en height, voorlopig 4 en 4 aanhouden.

Je kunt dus de spring met de hamer kapot slaan. Interessant. Maar doordat de code nu steeds de spring weer actief zet, sla je hem wel 4 a 5 keer kapot. Dat zal straks anders worden. Dan de spring alleen bij deployment actief zetten.

Nu de spring variabel deployen. Eventueel kunnen #6501 en #6502 gebruikt worden als counters.

Is de spring active?

Nee

Is de deploy timer (6501) gelijk aan 0?

Ja

- Maak de spring active
- Zet de sprite op grote spring
- Zet de verschillende waarden goed.
- Bepaal alvast een nieuwe deploy timer
- Start geluid vallende spring
- Spring naar bewegen van de spring – label A

Is de deploy timer (6501) kleiner dan 20?

Ja

- Zet de sprite op kleine spring

Decrement de deploy timer (6501)

Return

Ja

- Increment de Y-waarde van de spring (3x)

Is de Y-waarde > F8?

Ja

- Maak de spring inactive
- Zet de sprite op grote spring
- Zet de Y-waarde op beginwaarde
- Return

Return

De nieuwe code moet vanaf #2DDB, Daar hoeft geen check voor het screen nummer aan het begin want hier wordt alleen naartoe gesprongen als het screen 11 is.

```

2DDB DD210065 LD IX,#6500 ; load IX with start spring table
;
2DDF DD7E00 LD A,(IX+#00) ; load spring status to A
2DE2 FE00 CP #00 ; is spring active ?
2DE4 C2422E JP NZ,#2E42 ; yes, process active spring – jump to label AAAA
;
2DE7 DD7E01 LD A,(IX+#01) ; load deployment timer to A
2DEA FE00 CP #00 ; is deployment timer 0 ?
2DEC C2232E JP NZ,#2E23 ; no, do not make active – jump to label BBBB
;
2DEF DD360001 LD (IX+#00),#01 ; set spring active
2DF3 DD3603A7 LD (IX+#03),#A7 ; set spring X position to #A7
2DF7 DD360520 LD (IX+#05),#20 ; set spring Y position to #20
2DFB DD360904 LD (IX+#09),#04 ; set spring size width #04
2DFF DD360A04 LD (IX+#0A),#04 ; set spring size height #04
;
2E03 21B869 LD HL,#69B8 ; load HL with spring sprite
2E06 36A7 LD (HL),#A7 ; set spring sprite X position to #A7
2E08 23 INC HL ; next
2E09 363B LD (HL),#3B ; set spring sprite number to open spring
2E0B 23 INC HL ; next
2E0C 3600 LD (HL),#00 ; set spring sprite color
2E0E 23 INC HL ; next
2EOF 3620 LD (HL),#20 ; set spring sprite Y position to #20
;
2E11 CD5700 CALL #0057 ; load A with a random number
2E14 E67F AND #7F ; A is now number between #00 and #7F
2E16 C620 ADD A,#20 ; A is now number between #20 and #9F
2E18 DD7701 LD (IX+#01),A ; store A in deployment timer
;
2E1B 3E03 LD A,#03 ; load sound duration of 3
2E1D 328460 LD (#6084),A ; play sound for falling spring
;
2E20 C3422E JP #2E42 ; process active spring – jump to label AAAA
;
2E23 FE20 CP #20 ; is deployment timer less than 20 ? – label BBBB
2E25 D2342E JP NC,#2E34 ; no, do not display the closed spring – jump to label CCCC
;
2E28 21B969 LD HL,#69B9 ; load HL with spring sprite number
2E2B 363C LD (HL),#3C ; set spring sprite number to closed spring
2E2D 23 INC HL ; next
2E2E 23 INC HL ; next
2E2F 3620 LD (HL),#20 ; adjust spring sprite Y position to #20 for short spring
2E31 C33D2E JP #2E3D ; skip next steps – jump to label DDDD
;
2E34 21B969 LD HL,#69B9 ; load HL with spring sprite – label CCCC
2E37 363B LD (HL),#3B ; set spring sprite number to open spring
2E39 23 INC HL ; next

```

```

2E3A 23      INC   HL          ; next
2E3B 3620    LD    (HL),#20    ; adjust spring sprite Y position to #20 for long spring
;
2E3D 3D      DEC   A          ; A := A - 1 - label DDDD
2E3E DD7701    LD    (IX+#01),A  ; store A in deployment timer
2E41 C9      RET          ; return
;
2E42 DD7E05    LD    A,(IX+#05)  ; load spring Y-position to A - label AAAA
2E45 3C      INC   A          ; increment A
2E46 3C      INC   A          ; increment A
2E47 3C      INC   A          ; increment A
2E48 FEF8      CP    #F8        ; is A greater than #F8 ?
2E4A D2552E    JP    NC,2E55    ; yes, make spring inactive - jump to label EEEE
;
2E4D DD7705    LD    (IX+#05),A  ; store A in Y-position of spring
2E50 21BB69    LD    HL,#69BB    ; load HL with spring sprite Y-position
2E53 77      LD    (HL),A      ; store A in Y-position of spring sprite
2E54 C9      RET          ; return
;
2E55 DD360000  LD    (IX+#00),#00  ; set spring inactive - label EEEE
2E59 C9      RET          ; return

```

Nu versie v0.61 opgeslagen.

Gaat nog niet helemaal goed. Spring wordt heel even kort (bij 20?) en dan weer lang en valt daarna. Zou eigenlijk vanaf 20 tot 0 klein moeten zijn. Klopt: opcode is niet de juiste JP NC moet D2 zijn.

Gaat nu wel goed.

Alleen nog kijken bij hameren. Dan wordt de sprite inactive, maar niet getoond. En verder positioneren van de long en short spring zodat hij naar boven optrekt. Dan nog een elevator endpoint sprite tekenen bij opbouwen veld, waaraan de spring komt te hangen. En daarna ook nog veld aanpassen dat je gedwongen wordt op elke girder de spring te passeren.

De closed spring zit qua bovenkant in de sprite 4 posities lager. Dus deze 4 posities hoger tekenen. Dus dan op #1C ipv #20.

```
2E2F 361C      LD    (HL),#1C      ; adjust spring sprite Y position to #1C for short spring
```

Eventjes invincibility uitzetten. En kijken hoe het veld aanpassen. Gewoon de middelste ladder weglaten.

```
0CC6 E0 0C      ; #0CE0 for screen 11 Springs
```

Aanpassen en naar een eigen stukje code laten wijzen.

```
0CC6 45 0D      ; #0D45 for screen 11 Springs
```

en

```
0D45 115A2E LD DE,#2E5A ; Load DE with start of table data for screen 11
0D48 C3D00C JP #0CD0 ; jump back
```

Ja dat is goed. Alleen de onzichtbare ladder kan nog wel beklimmen worden. Dat nog aanpassen.

Al eerder gedaan

```
1907 217A25 LD HL,#257A ; load HL with start of table data of screen 10
190A FEOA CP #0A ; is this screen 10 ?
190C CA7124 JP Z,#2471 ; yes, define ladders
;
190F 21E43A LD HL,#3AE4 ; load HL with start of table data of regular screen
1912 C37124 JP #2471 ; jump back and define ladders
```

Aanpassen naar:

```
1907 217A25 LD HL,#257A ; load HL with start of table data of screen 10
190A FEOA CP #0A ; is this screen 10 ?
190C CA7124 JP Z,#2471 ; yes, define ladders
;
190F 215A2E LD HL,#2E5A ; load HL with start of table data of screen 11
1912 FEOA CP #0B ; is this screen 11 ?
1914 CA7124 JP Z,#2471 ; yes, define ladders
;
1917 21E43A LD HL,#3AE4 ; load HL with start of table data of regular screen
191A C37124 JP #2471 ; jump back and define ladders
```

Nu nog even tunen.

Wellicht de kort spring minder lang tonen. Nu op #20. Daar wellicht #18 van maken. Nee nog beter #10.

En ze sneller laten komen. De AND doen met #1F en de ADD doen met #18. Gaat goed. Nog eens testen met invincibility aan.

Eerst kijken naar het hameren. Na het hameren wordt de sprite niet goed getoond. Die staat gewoon uit (gezet op een lege sprite neem ik aan). Dat kan het probleem niet zijn, want ik zet altijd de sprite. Gekeken in de code. De code geeft de sprite een X-waarde van 0. Dat is buiten het zichtbare scherm. Dus in de code de X-waarde goedzetten.

```
2E3D 3D DEC A ; A := A – 1 – label DDDD
2E3E DD7701 LD (IX+#01),A ; store A in deployment timer
2E41 C9 RET ; return
```

Aanpassen:

```
2E3D C3FD25      JP    #25FD          ; jump to additional code – jump to label FFFF
2E40 0000      NOP, NOP
```

en

```
25FD 21B869      LD    HL,#69B8      ; load HL with spring sprite – label FFFF
2600 36A7      LD    (HL),#A7      ; set spring sprite X position to #A7
2602 3D      DEC   A             ; A := A – 1 – label DDDD
2603 DD7701      LD    (IX+#01),A    ; store A in deployment timer
2606 C9      RET
```

Ja dat hameren gaat nu goed.

Nu versie v0.62 opgeslagen.

Nog wel checken. Het hameren in andere velden gaat niet goed lijkt het. In ieder geval in veld 6 de bovenste hamer deed heel raar. Ook checken. In screen 1, 3, 5, 7 en 9 gooit Kong geen barrels meer?

Is weer goed als op #28AF een C9 gedaan wordt.

```
28AF 3A2762      LD    A,(#6227)    ; load A with screen number
28B2 FE0B      CP    #0B          ; is this screen 11 ?
28B4 C0      RET   NZ          ; no, return
;
28B5 0601      LD    B,#01        ; B := 1
28B7 78      LD    A,B          ; A := 1
28B8 32B963      LD    (#63B9),A    ; store counter for use later
28BB 1E00      LD    E,#00        ; E := #00
28BD DD210065      LD    IX,#6500    ; load IX with spring data location
28C1 CD1329      CALL  #2913        ; check for collision with spring
28C4 C9      RET
```

Is ook fout wanneer op #28B4 een C9 gedaan wordt. Het enige verschil is dan, dat de A aangepast is.

Eventjes testen met #28AF t/m 28B3 NOP's: dan niet meer scherm afhankelijk. Dus de spring wordt altijd getest. Gaat dan ook goed.

Wat zou kunnen is een PUSH AF en POP AF toevoegen. En dan wel de screen test behouden.

Aanpassen:

```
28AF F5      PUSH  AF          ; save A for later
28B0 3A2762      LD    A,(#6227)    ; load A with screen number
28B3 FE0B      CP    #0B          ; is this screen 11 ?
28B5 2802      JR    Z,#28B9        ; yes, do check collision spring – jump to label AAAA
;
28B7 F1      POP   AF          ; restore A
```

```

28B8 C9          RET           ; return
;
28B9 0601        LD   B,#01      ; B := 1 – label AAAA
28BB 78          LD   A,B      ; A := 1
28BC 32B963      LD   (#63B9),A ; store counter for use later
28BF 1E00        LD   E,#00      ; E := #00
28C1 DD210065    LD   IX,#6500  ; load IX with spring data location
28C5 CD1329      CALL #2913    ; check for collision with spring
28C8 C9          RET           ; return

```

Dit gaat niet goed. Dan gaat jumpman niet af door de spring. Maar alle andere velden werken wel goed.

Daarom de test op de spring gewoon altijd doen. Is ook maar een kleine test. Dus #28AF t/m 28B3 NOP's maken.

Dit gaat goed.

Nu versie v0.63 opgeslagen.

Interessant voor screen 1. Bij steering coloring van de barrels doen door ophogen kleurcode. Als x aantal keer gestuurd dan bonus te verdienen. Hoe wordt blue barrel herkend (aan kleurcode? Dan kun je wellicht een normale in een blauwe veranderen?. Nee. Blijkbaar wordt IX+#15 gebruikt om aan te geven of het een normale barrel of een blue barrel is. Dus de kleur van de sprite doet er niet toe. Wel eventjes kijken of het veranderen van de kleur van de sprite voldoende is. Wellicht wordt die weer overschreven vanuit de barrel data?

Dit is de code die het sturen van barrels regelt:

```

2197 3A0362    LD   A,(#6203)  ; load A with mario's x position
219A BB          CP   E         ; compare with barrel's x position
219B CAB221     JP   Z,#21B2   ; if equal, then go down ladder

219E D2A921     JP   NC,#21A9  ; if barrel is to right of mario, then check for moving to left

21A1 CB46       BIT  0,(HL)   ; else is mario trying to move right ?
21A3 CAAE21     JP   Z,#21AE  ; no, skip ahead and return without going down ladder

21A6 C3B221     JP   #21B2   ; yes, make barrel go down ladder

21A9 CB4E       BIT  1,(HL)   ; is mario trying to move left ?
21AB C2B221     JP   NZ,#21B2 ; yes, make barrel go down ladder

21AE 79          LD   A,C      ; else load A with random timer computed above
21AF E618       AND  #18      ; mask with #18. 25% chance of being zero?
21B1 C0          RET   NZ     ; else return without going down ladder. If zero then go down the ladder anyway

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

```

#219B is sturen doordat Mario op dezelfde X-positie, #21A6 is sturen rechts en #21AB is sturen links. Daar springen naar stukje additonele code die de kleur veranderd.

IX+#08 is de barrel color in de barrel data: barrel data: #6700 in blokken van #20.

HL+#02 is de barrel color in de sprite:

Bij overgaan van girder naar trap (en omgekeerd) wordt de sprite aangepast. Daarbij wordt de sprite kleur weer ingelezen vanuit de barrel data. Dus beste is om beiden aan te passen.

Er was ook nog iets met de aanpassing van de kleurwaarde. Een hoger bit die ergens voor gebruikt wordt. Eventjes terugzoeken bij screen met slow barrels.

Ja, de code doet iets met de kleur: propt er een bit 7 in. Geen idee waarom. Daarom wellicht beter om te checken op het tweede byte. Dus eerste een AND met #0F doen voor je de kleur checkt.

```
18C7 E60F    AND   #0F          ; get low byte
18C9 FE07    CP    #07          ; is this a slow barrel?
18CB C2BC18   JP    NZ, #18BC    ; no, jump back, do not slow barrel
18CE C3B418   JP    #18B4        ; yes, jump back, do slow barrel
```

Dus bij ophogen eerst de waarde lezen. Dan AND doen met #0F om de werkelijke kleur te bepalen en AND doen met #F0 om de bitflip te bewaren. Dan de kleur van de sprite ophogen. Daarna de kleur in de data aanpassen.

Aanpassen:

```
219B CA0726  JP   Z,#2607      ; if equal, then change color – jump to label AAAA
```

en

```
21A6 C30726  JP   #2607        ; yes, make barrel change color – jump to label AAAA
```

en

```
21AB C20726  JP   NZ,#2607    ; yes, make barrel change color – jump to label AAAA
```

en

```
2607 3A2762  LD    A,(#6227)    ; load A with screen number – label AAAA
260A FE01    CP    #01          ; is this screen 1 ?
260C C2B221  JP    NZ,#21B2    ; no, jump back without changing color
;
260F DD3408  INC   (IX+#08)    ; increase color in barrel data
2612 C3B221  JP    #21B2        ; jump back
```

Dit werkt. Alleen de kleuren niet altijd even handig want gewone krijgt ook de kleur van de blue barrel. Wellicht testen met dubbele increment? Ja dat is beter. Gewone en blue barrels volgen eigen kleuren volgorde.

Nu versie v0.64 opgeslagen.

Nu nog kijken of je bonus kunt geven bij 3 of 4 keer sturen van een barrel. Dan moet je testen op . Met kleurwaardes. Normal barrel start met 0B en blue barrel met 0C met increments kom je dan al snel boven de #0F:

0B – 0D – 0F – 11 - 13
0C – 0E – 10 – 12 - 14

En dan weet ik niet of de bit flip dan wel goed blijft gaan. Eventjes testen met een decrement.

0B – 09 – 07 – 05 - 03
0C – 0A – 08 – 06 - 04

Dat is misschien beter. Eventjes testen daarmee. Kleuren iets minder interessant, maar gaat goed. Bit flippen hoeft geen probleem te zijn. Alleen de test is iets ingewikkelder. Testen met bonus na vier succesvolle steers.

Dus dan testen kleurwaarde uit (IX+#08) en dan AND met #0F. Dan CP met #01. Als NC dan groter of gelijk aan 01 en dan bonus geven. Lastig: laat je ook bij de barrel zien dat de bonus gescoord is? Dat kan de huidige code niet. Twee varianten nu: bij jumpman (springen barrel) of ipv item (hameren of item pickup).

Eerst eens alleen de test en dan een geluid geven:

```
2607 3A2762 LD A,(#6227) ; load A with screen number – label AAAA
260A FE01 CP #01 ; is this screen 1 ?
260C C2B221 JP NZ,#21B2 ; no, jump back without changing color
;
260F DD3408 INC (IX+#08) ; increase color in barrel data
2612 DD3408 INC (IX+#08) ; increase color in barrel data
;
2615 DD7E08 LD A,(IX+#08) ; load new barrel color in A
2618 E60F AND #0F ; get low byte of barrel color
261A FE01 CP #01 ; color code >= 01?
261C DAB221 JP C,#21B2 ; no, jump back without sound
261F FE05 CP #05 ; color code < 05?
2621 D2B221 JP NC,#21B2 ; no, jump back without sound
;
2624 21AA39 LD HL,#39AA ; load HL with start of table data
2627 3E03 LD A,#03 ; load sound duration of 3
2629 328360 LD (#6083),A ; play sound for bouncer
262C C3B221 JP #21B2 ; jump back
```

Dat gaat heel goed. Nu nog punten geven?

Zou eventueel zo kunnen?

```
262C 110100 LD DE,#0001 ; 100 points
262F 067B LD B,#7B ; sprite for 100
2631 CD281E CALL #1E28 ; yes, award points
2634 C3B221 JP #21B2 ; jump back
```

Dat werkt wel, maar geeft de bonus ietwat laag aan. Maar telt er wel punten voor op bij de score. Eigen stukje code maken en alleen voor 300 punten de sprite boven jumpman in een andere kleur.

```
262C 110300 LD DE,#0003 ; 300 points
262F 067B LD B,#7B ; sprite for 100
2631 CD9F30 CALL #309F ; yes, award points
;
2634 21306A LD HL,#6A30 ; load HL with scoring sprite start
2637 3A0362 LD A,(#6203) ; load A with mario's X position
263A 77 LD (HL),A ; store X position
263B 2C INC L ; next location
263C 70 LD (HL),B ; store sprite graphic
263D 2C INC L ; next
263E 3606 LD (HL),#06 ; store color code 6
2640 2C INC L ; next
2641 3A0562 LD A,(#6205) ; load A with Mario's Y position
2644 D60C SUB A,#0C ; sub #0C
2646 77 LD (HL),A ; store X position
;
2647 C3B221 JP #21B2 ; jump back
```

Color #02 is donkerblauw.

Color #0B is lichtbruin

Color #0D is donkerbruin

Gaan voor de kleur #0B. Geeft beste contrast en is wel afwijkend van de witte kleur van normale bonuspunten.

Nog wel een dingetje. De tekst wordt niet vanzelf weggehaald. Hoe wordt dit voor de bonus punten gedaan? Debuggen op write naar #6A30 (de bonus sprite).

Lijkt hier te gebeuren.

```

1E4A 214163 LD HL,#6341 ; load HL with timer
1E4D 35 DEC (HL) ; has it run out yet ?
1E4E C0 RET NZ ; no, return

1E4F AF XOR A ; else A := 0
1E50 32306A LD (#6A30),A ; clear this
1E53 324063 LD (#6340),A ; clear this
1E56 C9 RET ; return

```

Wat wordt er in de timer gezet? Dat lijkt hier te gebeuren.

```

; an item was just picked up / jumped over / hit with hammer

1DC9 3E40 LD A,#40 ; A := #40
1DCB 324163 LD (#6341),A ; store into timer
1DCE 3E02 LD A,#02 ; A := 2
1DD0 324063 LD (#6340),A ; store into #6340 -

```

Daarom bij stuurbonus deze timer ook zetten.

Aanpassen:

```

2647 3E40 LD A,#40 ; A := #40
2649 324163 LD (#6341),A ; store into timer
264C 3E02 LD A,#02 ; A := 2
264E 324063 LD (#6340),A ; store into #6340
2651 C3B221 JP #21B2 ; jump back

```

Nu versie v0.65 opgeslagen.

Screen 12 Anti zwaartekracht

Een van de girders in het veld zo maken dat de barrels omhoog moeten rollen.
Dit betekent een nieuw veld. Kopie maken van het originele veld.

```
0CC8 4B 0D ; #0D4B for screen 12 Anti gravity
```

en

```
0D4B 115426 LD DE,#2654 ; Load DE with start of table data for screen 12
0D4E C3D00C JP #0CD0 ; jump back
```

In dat veld dan de 4e girder qua Y-waarden omdraaien:

2668: 02 DF 9A 10 8E aanpassen naar: 02 DF 8E 10 9A

En de ladders ook aanpassen.

```
2686: 00 CB 8F CB B1 ; short ladder at center right
269A: 00 33 78 33 98 ; short ladder at left side under top hammer
26A9: 01 53 96 53 B8 ; second broken ladder from bottom, on 3rd girder
26AE: 00 5B 76 5B 96 ; longer ladder under the top left hammer
26B8: 00 83 93 83 B5 ; center longer ladder
26C2: 01 BB 70 BB 90 ; third broken ladder on right side near top
```

Da werkt nu goed.

Nu versie v0.66 opgeslagen.

Nu alleen de ladders nog goed maken, omdat nog de oude definitie van het standaard veld ingeladen wordt.

```
1907 217A25 LD HL,#257A ; load HL with start of table data of screen 10
190A FE0A CP #0A ; is this screen 10 ?
190C CA7124 JP Z,#2471 ; yes, define ladders
;
190F 215A2E LD HL,#2E5A ; load HL with start of table data of screen 11
1912 FE0B CP #0B ; is this screen 11 ?
1914 CA7124 JP Z,#2471 ; yes, define ladders
;
1917 21E43A LD HL,#3AE4 ; load HL with start of table data of regular screen
191A C37124 JP #2471 ; jump back and define ladders
```

Was de laatste.

Kan er nog eentje bij (past precies, 1 byte over):

```
1917 215426 LD HL,# 2654 ; load HL with start of table data of screen 12
191A FE0C CP #0C ; is this screen 12 ?
191C CA7124 JP Z,#2471 ; yes, define ladders
;
191F 21E43A LD HL,#3AE4 ; load HL with start of table data of regular screen
1922 C37124 JP #2471 ; jump back and define ladders
```

De trappen moeten nog verplaatst worden. Nu nog niet goed. De hele trappen blijven doorklimmen. Maar als je ze opschuift blijft de afstand tussen de girders gelijk. Heeft dat wel zin dan? Toch eventjes proberen met de center longer ladder.

26B8: 00 83 93 83 B5 ; center longer ladder

00 8B 93 8B B5 – werkt niet
00 93 92 93 B4 – werkt niet
00 9B 92 9B B4 – werkt niet
00 A3 91 A3 B3 – werkt niet.

Dus anders oplossen: girders verplaatsen. Eerst proberen met verplaatsen bovenste girder:

00 83 92 83 B5 – werkt niet

00 83 91 83 B5 – werkt wel.

Dus girder moet twee omhoog:

2668: 02 DF 9A 10 8E aanpassen naar: 02 DF **8C** 10 **98**

En de ladders aanpassen:

2686: 00 CB **8D** CB B1 ; short ladder at center right

269A: 00 33 78 33 **96** ; short ladder at left side under top hammer

26A9: 01 53 **94** 53 B8 ; second broken ladder from bottom, on 3rd girder

26AE: 00 5B 76 5B **94** ; longer ladder under the top left hammer

26B8: 00 83 **91** 83 B5 ; center longer ladder

26C2: 01 BB 70 BB **8E** ; third broken ladder on right side near top

Ander probleem. Nu zijn de ladders naar de girder erboven weer niet goed. Dan moet je die weer gaan aanpassen. Mmmm, eventjes goed over nadenken.

Wellicht de vorige verschuiving niet met 2 omhoog, maar met 1 omhoog en dan de girder eronder met 1 omlaag en de girder erboven met een omhoog. Wel een boel werk dan. Eerst de girders goedzetten en dan de ladders aanpassen. Heeft impact op vrijwel het hele veld.

Dus girders moeten aangepast worden:

2663: 02 EF 6D 20 79 aanpassen naar: 02 EF **6C** 20 **78**

2668: 02 DF 9A 10 8E aanpassen naar: 02 DF **8D** 10 **99**

266D: 02 EF AF 20 BB aanpassen naar: 02 EF **AC** 20 **B8**

2672: 02 DF DC 10 D0 aanpassen naar: 02 DF **D9** 10 **CD**

2677: 02 FF **F1** **90** F7 ; bottom slanted girder

267C: 02 **8F** F8 00 F8 ; bottom flat girder where mario starts

2659: 02 **AF** 54 10 54 ; girder where kong sits

265E: 02 DF **57** **B0** **54** ; 1st slanted girder at top right

En de ladders aanpassen:

2686: 00 CB **8E** CB AE ; short ladder at center right

269A: 00 33 77 33 **97** ; short ladder at left side under top hammer

26A9: 01 53 **95** 53 B5 ; second broken ladder from bottom, on 3rd girder

26AE: 00 5B 75 5B **95** ; longer ladder under the top left hammer

26B8: 00 83 **92** 83 B3 ; center longer ladder

26C2: 01 BB **6F** BB **8F** ; third broken ladder on right side near top

En nog meer ladders aanpassen:

269F: 00 33 **B7** 33 **CF** ; short ladder at left side above oil can

26B3: 00 73 B3 73 D5 ; longer ladder to left of bottom hammer

En nog meer ladders aanpassen:

268B: 00 CB D8 CB F3 ; short ladder at bottom right

2695: 01 63 D2 63 F8 ; bottom broken ladder

2681: 00 CB 56 CB 6E ; short ladder at top right

26C7: 01 6B 54 6B 74 ; fourth broken ladder near kong

Eindelijk de boel goed qua level en klimbaarheid van de ladders.

Nu versie v0.67 opgeslagen.

Nu de bijstelling voor de slanted girders gaan aanpassen. Voor de omgedraaide girder en voor de korte girders bovenin en onderin.

De omgekeerde girder is 2668: 02 DF 8D 10 99

Dus dat is een Y-waarde grofweg tussen #88 en #9F.

De code die eerder deze aanpassingen deed:

2353 C3511A JP #1A51 ; jump to additional code: jump to label AAAA

en

```
1A51 3A2762      LD      A,(#6227)                          ; load A with screen number – label AAAA
1A54 FE09      CP      #09                                          ; is this screen 9 ?
1A56 2011      JR      NZ, #1A69                                  ; no continue original code – jump to label BBBB
;
1A58 7D      LD      A,L                                                  ; load A with L = Y-position
1A59 FE7E      CP      #7E                                                  ; check if Y-position is lower than #7E
1A5B 380C      JR      C, #1A69                                          ; yes: continue original code – jump to label BBBB
;
1A5D FEBC      CP      #BC                                                  ; check if Y-position is higher than #BC
1A5F 3008      JR      NC, #1A69                                          ; yes: continue original code – jump to label BBBB
;
; Y-position in area of moved girders
;
1A61 FE9C      CP      #9C                                                  ; check if Y-position is lower than #9C
1A63 DA5C23      JP      C, #235C                                          ; yes: perform no bit 5 – jump back 1
1A66 C35923      JP      #2359                                                  ; no: perform no bit 5 – jump back 2
;
1A69 7D      LD      A,L                                                          ; original code that performs the bit 5 – label BBBB
1A6A CB6F      BIT      5,A
```

```

1A6C CA5C23  JP   Z,#235C      ; jump back 1
1A6F C35923  JP   #2359       ; jump back 2

```

Eigenlijk eenzelfde iets ertussen voegen:

```

2353 C3C528  JP   #28C5      ; jump to additional code

```

en

```

28C5 3A2762  LD   A,(#6227)    ; load A with screen number – label AAAA
28C8 FE0C    CP   #0C          ; is this screen 12 ?
28CA C2511A  JP   NZ, #1A51    ; no continue original code
;
28CD 7D      LD   A,L         ; load A with L = Y-position
28CE FE80    CP   #80          ; check if Y-position is lower than #80
28D0 DA691A  JP   C, #1A69    ; yes: continue original code – jump to label BBBB
;
28D3 FE9F    CP   #9F          ; check if Y-position is higher than #9F
28D5 D2691A  JP   NC, #1A69    ; yes: continue original code – jump to label BBBB
;
; Y-position in area of flipped girder
;
28D8 C35923  JP   #2359      ; no: perform no bit 5 – jump back 2

```

Ja dat gaat goed. Ze rollen inderdaad omhoog.

Nu versie v0.68 opgeslagen.

Nu nog de rollover aanpassen. Dat is ook al voor meerdere schermen aangepast.

Eerst de linkerkant. Doet check op screen 3 en screen 9:

```

;
; Check barrel roll-over left edge
;
04BE 3A2762  LD   A,(#6227)    ; load A with screen number – label AAAA
04C1 FE09    CP   #09          ; is this screen 9 ?
04C3 CAA11A  JP   Z, #1AA1    ; yes, jump to label HHHH
04C6 FE03    CP   #03          ; is this screen 3 ?
04C8 CAD604  JP   Z, #04D6    ; yes, jump to label BBBB
;
; Check barrel left edge most screens
;
04CB DD7E03  LD   A,(IX+#03)   ; load A with Barrel's X position
04CE FE1C    CP   #1C          ;
04D0 DA2F20  JP   C, #202F    ; yes, jump ahead to handle
04D3 C31A20  JP   #201A      ; continue check right edge, jump to label CCCC
;
```

```

; Check barrel left edge screens #2 and #3
;
04D6 DD7E03 LD A,(IX+#03) ; load A with Barrel's X position – label BBBB
04D9 FE2C CP #2C
04DB DA2F20 JP C, #202F ; yes, jump ahead to handle
04DE C31A20 JP #201A ; continue check right edge, jump to label CCCC
;

```

en

```

1AA1 DD7E03 LD A,(IX+#03) ; load A with Barrel's X position – label BBBB
1AA4 FE10 CP #10
1AA6 DA7920 JP C, #2079 ; yes, jump ahead to handle
1AA9 C3CB04 JP #04CB ; continue check left edge, jump to label CCCC

```

Nu aanpassen naar:

```

04C1 C3CD26 JP #26CD ; jump to additional code
04C4 00 00 NOP, NOP

```

en

```

26CD FE09 CP #09 ; is this screen 9 ? - label XXXX
26CF CAA11A JP Z, #1AA1 ; yes, jump to label FFFF
;
26D2 FE0C CP #0C ; is this screen 12?
26D4 CADA26 JP Z, #26DA ; yes, perform barrel rollover more left
26D7 C3C604 JP #04C6 ; no, check for screen 3
;
26DA DD7E03 LD A,(IX+#03) ; load A with Barrel's X position – label BBBB
26DD FE18 CP #18
26DF DA2F20 JP C, #202F ; yes, jump ahead to handle
26E2 C31A20 JP #201A ; continue check left edge, jump to label CCCC

```

Om te testen moet de girder wel korter. Eerst testen met een half stuk korter.

2663: 02 EF 6D 20 79 aanpassen naar: 02 EF 6C 20 78 -> 02 EF 6C 28 78
 266D: 02 EF AF 20 BB aanpassen naar: 02 EF AC 20 B8 -> 02 EF AC 28 B8

Ook eventjes testen of de left roll over in de andere screens goed gaat. Testen screen 1, 3, 9 en 7. Ja dat gaat allemaal goed.

De linkerkant kan een half korter. Maar aan de rechterkant gaat niet. Daar alleen de ene girder korter maken.

2668: 02 DF 9A 10 8E aanpassen naar: 02 DF 8D 10 99 -> 02 CF 8E 10 99

Nu versie v0.69 opgeslagen.

Kijken hoe de rollover aan de rechterkant gedaan wordt in eerdere screens.

Dat is nog ingewikkelder samengesteld.

```
04E1 3A2762 LD A,(#6227) ; load A with screen number – label DDDD
04E4 C30722 JP #2207 ; jump to additional code
04E7 00 00 NOP, NOP
04E9 FE03 CP #03 ; is this screen 3 ?
04EB CAF904 JP Z, #04F9 ; yes, jump to label EEEE
;
; Check barrel right edge most screens
;
04EE DD7E03 LD A,(IX+#03) ; load A with Barrel's X position
04F1 FEE4 CP #E4
04F3 D22420 JP NC, #2024 ; yes, jump ahead to handle
04F6 C3BA21 JP #21BA ; continue, jump way ahead
;
; Check barrel right edge screens #3 and #10
;
04F9 DD7E03 LD A,(IX+#03) ; load A with Barrel's X position – label EEEE
04FC FED4 CP #D5
04FE D22420 JP NC, #2024 ; yes, jump ahead to handle
0501 C3BA21 JP #21BA ; continue, jump way ahead
```

en

```
2207 FE09 CP #09 ; is this screen 9 ?
2209 CA851A JP Z, #1A85 ; yes, jump to label FFFF
;
220C FE0A CP #0A ; is this screen 10?
220E CAF904 JP Z, #04F9 ; yes, perform barrel rollover more left
2211 C3E904 JP #04E9 ; no, check for screen 3
```

en

```
;
; Check barrel right edge screen #9
;
1A85 DD7E05 LD A,(IX+#05) ; load A with Barrel's Y position – label FFFF
1A88 FE8D CP #8D ; Y-position is 8D ?
1A8A 200A JR NZ,#1A96 ; no, check for other roll over – jump to label GGGG

1A8C DD7E03 LD A,(IX+#03) ; load A with Barrel's X position
1A8F FED0 CP #D0 ; X-position > D0 ?
1A91 3803 JR C,#1A96 ; no, check for other roll over – jump to label GGGG
;
1A93 C32420 JP #2024 ; yes, jump ahead to handle right roll over on girder 3
```

```

;
1A96 DD7E03 LD A,(IX+#03) ; load A with Barrel's X position – label GGGG
1A99 FEE4 CP #E4
1A9B D22420 JP NC, #2024 ; yes, jump ahead to handle
1A9E C3BA21 JP #21BA ; continue, jump way ahead

```

Aanpassen:

```

04E4 C3E526 JP #26E5 ; jump to additional code – jump to label XXXX
04E7 00 00 NOP, NOP

```

en

```

26E5 FE0C CP #0C ; is this screen 12 ? - label XXXX
26E7 C20722 JP NZ,#2207 ; no, continue check screen 9
;
; Check barrel right edge screen 12
;
26EA DD7E05 LD A,(IX+#05) ; load A with Barrel's Y position
26ED FE88 CP #88 ; Y position < 88 ?
26EF DAEE04 JP C,#04EE ; yes, above the inverted girder, handle normal rollover
26F2 FE9F CP #9F ; Y position > 9F ?
26F4 D2EE04 JP NC,#04EE ; yes, below the inverted girder, handle normal rollover
26F7 C3F904 JP #04F9 ; no, inverted girder, check barrel rollover screen 3 and 10

```

Ja dat gaat goed. Barrel rollover gaat nu ook rechts goed. Alleen nog de adjustment van de bovenste en onderste stukjes schuine girder moet nog aangepast worden.

[Nu versie v0.70 opgeslagen.](#)

De slanted girders bijstelling gebeurd in de code van #2333 t/m #236D.

Schuine girder beneden begon bij jumpman X,Y: (80,F0), nu bij (90,F0).

Schuine girder boven begon bij jumpman X,Y: (A0,4C), nu bij (B0, 4C).

Zijn deze waarden terug te vinden in het stukje code?

Eventjes testen voor de bovenste girder: CP in #2367 aanpassen naar A8? Ja dat gaat goed.
En de onderste girder werkt vergelijkbaar, maar doet bit 7: X positie jumpman halverwege scherm.

Code die bovenste en onderste girders afhandelt:

```

2347 3EF0 LD A,#F0 ; A := #F0, Y-position bottom girder
2349 BD CP L ; is A == L ?
234A CA6023 JP Z,#2360 ; yes, skip ahead

234D 3E4C LD A,#4C ; A := #4C Y-position top girder
234F BD CP L ; is A == L ?

```

```
2350 CA6623 JP Z,#2366 ; yes, skip ahead
```

Aanpassen in:

```
234A CADB28 JP Z,#28DB ; yes, jump to additional code – jump to label AAAA
```

en

```
2350 CAEE28 JP Z,#28EE ; yes, jump to additional code – jump to label BBBB
```

en

```
28DB 4F LD C,A ; save A for later
28DC 3A2762 LD A,(#6227) ; load A with screen number – label AAAA
28DF FE0C CP #0C ; is this screen 12 ?
28E1 79 LD A,C ; restore A
28E2 C26023 JP NZ,#2360 ; no, return, perform original check
;
28E5 4F LD C,A ; save A for later
28E6 7C LD A,H ; A := H
28E7 FE90 CP #90 ; > #90 ?
28E9 79 LD A,C ; restore A
28EA D25923 JP NC,#2359 ; yes, adjust for slanted girder
28ED C9 RET ; return, do not adjust for slanted girder
;
28EE 3A2762 LD A,(#6227) ; load A with screen number – label BBBB
28F1 FE0C CP #0C ; is this screen 12 ?
28F3 C26623 JP NZ,#2366 ; no, return, perform original check
;
28F6 7C LD A,H ; A := H
28F7 FEA8 CP #A8 ; < #A8 ?
28F9 D8 RET C ; no, return, do not adjust for slanted girder
;
28FA 7D LD A,L ; A := L
28FB C35C23 JP #235C ; jump back, adjust for slanted girder
```

Gaat nu goed.

Nu versie v0.71 opgeslagen.

Nog eventjes testen of het bij de andere screens nog goed gaat.

Nee die gaan niet goed bovenin. Op helft van het scherm springen de barrels ineens een heel stuk naar boven. Zo hoog dat ze half boven en half onder zichtbaar zijn. Lopt was foutje in return adres in #28F3: door kopieren stond daar nog #2360. Aangepast. Nu goed.

Nu versie v0.71 opgeslagen.

Nu nog de barrels langzaam de helling op laten rollen.

Al eerder gedaan in screen 8 met de slow barrels. Alleen nu anders, namelijk voor alle barrels, maar alleen op de gekantelde girder. Dus afhankelijk van de Y-positie.

Toen helemaal de check aan het begin. Nu anders doen.

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

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

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

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

1FA9 C35320 JP #2053 ; else jump ahead
```

In principe alleen als de barrel naar rechts beweegt. Dus dan op #1FA2 naar een stukje additionele code springen die checkt op screen 12. Zo nee, dan jump naar de originele #1FE5. Zo ja, dan checken op de Y-waarde: als die tussen #88 en #9F (checken of deze kloppen), dan het stukje code dat afhankelijk van framecounter de skip doet:

```
18B4 3A1A60 LD A,(FrameCounter) ; load A with frame counter ?
18B7 E603 AND #03 ; mask bits. random number between 0 and 3
18B9 CA8A1F JP Z,#1F8A ; if 0: return without processing barrel, do next
```

Aanpassen:

```
1FA2 DAFE28 JP C,#28FE ; yes, jump to additional code – jump to label AAAA
```

en

```
28FE 3A2762 LD A,(#6227) ; load A with screen number – label AAAA
2901 FE0C CP #0C ; is this screen 12 ?
2903 C2E51F JP NZ,#1FE5 ; no, jump back to original code
;
2906 3A1A60 LD A,(FrameCounter) ; load A with frame counter ?
2909 E603 AND #03 ; mask bits. random number between 0 and 3
290B CA8A1F JP Z,#1F8A ; if 0: return without processing barrel, do next
290E C3E51F JP #1FE5 ; jump back to original code
```

Dat gaat goed, maar nu op alle girders naar rechts. Moet nog de check op de Y-positie inbouwen. Maar past dan niet meer daar. Dus verplaatsen.

Aanpassen:

1FA2 DAB93C JP C,#3CB9 ; yes, jump to additional code – jump to label **AAAA**

en

```
3CB9 3A2762 LD A,(#6227) ; load A with screen number – label AAAA
3CBC FE0C CP #0C ; is this screen 12 ?
3CBE C2E51F JP NZ,#1FE5 ; no, jump back to original code
;
3CC1 DD7E05 LD A,(IX+#05) ; load A with the barrels Y-position
3CC4 FE88 CP #88 ; is Y-position above inverted girder
3CC6 DAE51F JP C,#1FE5 ; yes, jump back to original code
3CC9 FE9F CP #9F ; is Y-position below inverted girder
3CCB D2E51F JP NC,#1FE5 ; yes, jump back to original code
;
3CCE 3A1A60 LD A,(FrameCounter) ; load A with frame counter ?
3CD1 E603 AND #01 ; mask bits. random number between 0 and 1
3CD3 CA8A1F JP Z,#1F8A ; if 0: return without processing barrel, do next
3CD6 C3E51F JP #1FE5 ; jump back to original code
```

Gaat goed. Nu testen met invincibility off. Is niet makkelijk. Barrel gaat langzaam. Je kan niet rechtop springen. Sommige groupings zijn daardoor best lastig.

Nu versie v0.72 opgeslagen.

Elevators gaan we niet gebruiken. Daarom hele stuk van #26FA t/m #2807 vrijgegeven. Nu steekproefsgewijs testen of alle boards het nog goed doen. Lijkt goed te gaan.

Kijken of de tuning van de slow en superslow barrels in screen 8 nog aangepast moet worden.

Was: kans op slow barrel 1 op 4 en daarvan kans op super slow barrel 1 op 4 (= 1 op 16).

Aangepast: kans op slow barrel 1 op 2 en daarvan kans op super slow barrel 1 op 8 (= 1 op 16).

Nu versie v0.73 opgeslagen.

Kijken of de X positie van de spring gevarieerd kan worden. Dat is screen 11.
Nee dat is best ingewikkeld. En geeft niet veel extra's.

De board indicator gaan toevoegen: B=06 enz.

L= wordt hier getekend:

```

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:

```

06FA C3FA26 JP #26FA ; jump to additional code – jump to label AAAA
06FD 00 NOP

en
;
26FA 32C374 LD (#74C3), A ; draw level to screen (high byte)
;
26FD 218374 LD HL, #7483 ; load HL with screen location for "-"
2700 3635 LD (HL), #35 ; draw "-"

2702 3A2762 LD A, (#6227) ; load A with screen #
2705 010AFF LD BC, #ff0A ; B: = #FF, C := #0A (10 decimal)

2708 04 INC B ; increment B – label BBBB
2709 91 SUB C ; subtract 10 decimal
270A D20827 JP NC, #2708 ; not carry, loop again – jump to label BBBB
;
270D 81 ADD A, C ; add 10 back to A to get a number from 0 to 9
270E 324374 LD (#7443), A ; draw board to screen (low byte)
2711 78 LD A, B ; load a with b (number of tens)
2712 326374 LD (#7463), A ; draw board to screen (high byte)
;
```

```
2715 C9      RET          ; return
```

Aanpassen: BXX-LYY bijvoorbeeld B11-L03

```
2700 361C     LD      (HL),#1C      ; draw "L"
```

En de board en level omdraaien. Goed.

Boardcijfers één naar links. Goed.

L moet B worden. Goed.

Is nu goed.

Nu versie v0.74 opgeslagen.

HIGH SCORE aanpassen naar BARRELPALOOZA:

#3653 wijst nu naar 36B2 voor tekst 4 (HIGH SCORE) maar daar past BARRELPALOOZA niet.

Dus tekst in een apart stuk opnemen:

```
#2716 C0 76 12 11 22 22 15 1C 20 11 1C 1F 1F 2A 11 3F
```

En

```
3653 1627
```

Nu versie v0.75 opgeslagen.

Het titelscherm aanpassen:

De KONG letters 1 positie naar beneden verschuiven:

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

En het TM symbool weghalen.

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

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

#3F24 t/m #3F2F.

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

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

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

In #3F00 de copyright notice maken:

© 1981,2020 NINTENDO

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

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

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

En dan in #3F18 de titel maken:

BARRELPALOOZA

EE 76 12 11 22 22 15 1C 20 11 1C 1F 1F 2A 11 3F

En de verwijzing ernaar toe aanpassen:

3689: 18 3F

Nu nog de Donkey Kong een regel naar beneden verplaatsen.

Wordt hier gedaan:

```
0820 210869 LD HL,#6908 ; load HL with start of kong sprite X pos
0823 0E44 LD C,#44 ; load C with offset to add X
0825 FF RST #38 ; draw kong in new position
0826 210B69 LD HL,#690B ; load HL with start of kong sprite Y pos
0829 0E78 LD C,#78 ; load C with offset to add Y
082B FF RST #38 ; draw kong
082C C9 RET ; return
```

Kan door aanpassen:

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

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

De 1 is nog unused:

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

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

```

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

```

Dus daartoe #373D veranderen in 3F.

En dan de versie opnemen in stuk daarna:

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

en

#364D: 3E 37

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

```

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

```

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

```

081C: C3283F JP #3F28 ; jump to additional code to display version
3F28: 110103 LD DE,#0301 ; load task data for text "V1.00"
3F2B: CD9F30 CALL #309F ; insert task to draw text
3F2E: C31F08 JP #081F ; jump back

```

Nu versie v0.76 opgeslagen.

Het titelscherm niet laten knipperen.

Gewoon één palette kiezen. Palette A = 1 en Palette B = 0. Dat is palette van Pie factory (en ook het How High scherm).

En twee barrels tekenen in het titelscherm.

Daarvoor kijken naar het tekenen van de barrels in de barrelpile.

De gebruikt spriteadres #69A8 en verder. En vult dat met X / 18 / 0B / Y.

Dus tekenen tussenvoegen. Dit kan mooi bij de taak die ook de versie tekent:

```
3F28 110103 LD DE,#0301 ; load task data for text "V1.00"
```

```

3F2B CD9F30 CALL #309F ; insert task to draw text
;
3F2E 21A869 LD HL,#69A8 ; load HL with barrel sprite
3F31 36D6 LD (HL),#D6 ; set barrel sprite x-position
3F33 23 INC HL ; next
3F34 3618 LD (HL),#18 ; set barrel sprite
3F36 23 INC HL ; next
3F37 360B LD (HL),#0B ; set barrel sprite color
3F39 23 INC HL ; next
3F3A 36C8 LD HL,#C8 ; set barrel sprite y-position
;
3F3C C31F08 JP #081F ; jump back

```

Werkt goed, maar één barrel. Beter om de code voor het tekenen van de vier barrels in de barrel pile te hergebruiken en dan de barrels over het scherm verdelen: zijn dan vier barrels.

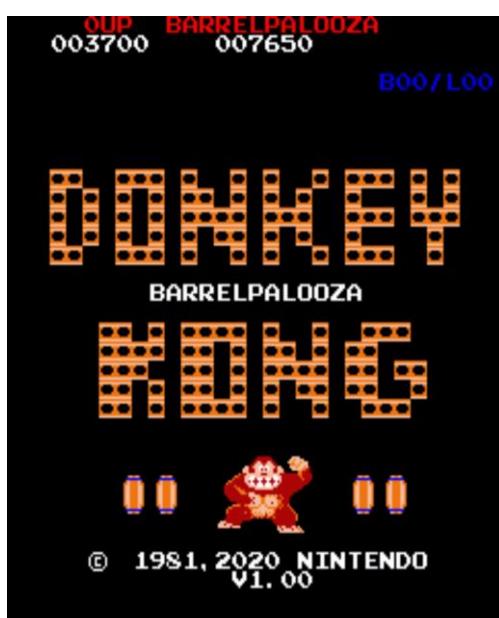
Alternatief:

```

3F28 110103 LD DE,#0301 ; load task data for text "V1.00"
3F2B CD9F30 CALL #309F ; insert task to draw text
;
3F2E 213C3F LD HL,#3F3C ; source - has the information about the barrels
3F31 11A869 LD DE,#69A8 ; destination = sprites
3F34 011000 LD BC,#0010 ; counter is #10
3F37 EDB0 LDIR ; draws the barrels pile
;
3F39 C31F08 JP #081F ; jump back
;
3F3C 40 18 0B C8 ; first barrel
3DE0 50 18 0B C8 ; second barrel
3DE4 B0 18 0B C8 ; third barrel
3DE8 C0 18 0B C8 ; fourth barrel

```

Het titelscherm ziet er nu zo uit:



Nu versie v0.77 opgeslagen.

Kijken hoe het eruit ziet als High Score screen ook met hetzelfde palette als het titelschermer weergegeven wordt. Als dat er goed uitziet hebben we een palette over dat eventueel gebruiky kan worden om een screen van een thema te voorzien.

```
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
```

Change naar 01 / 00. Nee ziet er niet mooi uit. Alle teksten in wit. Terugzetten.

Nog wel vreemd verschijnsel in het attract mode scherm. Wanneer jumpman bovenaan de trap is verspringt de 1UP naar 0UP. Hoe kan dat? Hebben de andere rom-hacks niet. Debuggen waar dat door gezet wordt. En checken of dit ook in het normale spel gebeurt? Sowieso ook nog de two-player werking testen.

Kijken of we de spring in screen 11 variabele X-positie kunnen geven. Zou kunnen door aan het begin te checken op X=0. Dan namelijk start level of barrel kapot geslagen. Als X=0 dan nieuwe X bepalen en nieuwe deployment counter. En dan aanpassen dat wanneer barrel buiten beeld, alleen de X op nul zetten.

```
2DDB DD210065 LD IX,#6500 ; load IX with start spring table
;
2DDF DD7E00 LD A,(IX+#00) ; load spring status to A
2DE2 FE00 CP #00 ; is spring active ?
2DE4 C2422E JP NZ,#2E42 ; yes, process active spring – jump to label AAAA
;
2DE7 DD7E01 LD A,(IX+#01) ; load deployment timer to A
2DEA FE00 CP #00 ; is deployment timer 0 ?
2DEC C2232E JP NZ,#2E23 ; no, do not make active – jump to label BBBB
;
2DEF DD360001 LD (IX+#00),#01 ; set spring active
2DF3 DD3603A7 LD (IX+#03),#A7 ; set spring X position to #A7
2DF7 DD360520 LD (IX+#05),#20 ; set spring Y position to #20
2DFB DD360904 LD (IX+#09),#04 ; set spring size width #04
2DFF DD360A04 LD (IX+#0A),#04 ; set spring size height #04
;
2E03 21B869 LD HL,#69B8 ; load HL with spring sprite
2E06 36A7 LD (HL),#A7 ; set spring sprite X position to #A7
2E08 23 INC HL ; next
2E09 363B LD (HL),#3B ; set spring sprite number to open spring
2E0B 23 INC HL ; next
2E0C 3600 LD (HL),#00 ; set spring sprite color
2E0E 23 INC HL ; next
2EOF 3620 LD (HL),#20 ; set spring sprite Y position to #20
```

```

;
2E11 CD5700 CALL #0057 ; load A with a random number
2E14 E63F AND #3F ; A is now number between #00 and #3F
2E16 C618 ADD A,#18 ; A is now number between #18 and #57
2E18 DD7701 LD (IX+#01),A ; store A in deployment timer
;
2E1B 3E03 LD A,#03 ; load sound duration of 3
2E1D 328460 LD (#6084),A ; play sound for falling spring
;
2E20 C3422E JP #2E42 ; process active spring – jump to label AAAA
;
2E23 FE10 CP #10 ; is deployment timer less than 10 ? – label BBBB
2E25 D2342E JP NC,#2E34 ; no, do not display the closed spring – jump to label CCCC
;
2E28 21B969 LD HL,#69B9 ; load HL with spring sprite number
2E2B 363C LD (HL),#3C ; set spring sprite number to closed spring
2E2D 23 INC HL ; next
2E2E 23 INC HL ; next
2E2F 361C LD (HL),#1C ; adjust spring sprite Y position to #1C for short spring
2E31 C33D2E JP #2E3D ; skip next steps – jump to label DDDD
;
2E34 21B969 LD HL,#69B9 ; load HL with spring sprite – label CCCC
2E37 363B LD (HL),#3B ; set spring sprite number to open spring
2E39 23 INC HL ; next
2E3A 23 INC HL ; next
2E3B 3620 LD (HL),#20 ; adjust spring sprite Y position to #20 for long spring
;
2E3D 3D DEC A ; A := A – 1 – label DDDD
2E3E DD7701 LD (IX+#01),A ; store A in deployment timer
2E41 C9 RET ; return
;
2E42 DD7E05 LD A,(IX+#05) ; load spring Y-position to A – label AAAA
2E45 3C INC A ; increment A
2E46 3C INC A ; increment A
2E47 3C INC A ; increment A
2E48 FEF8 CP #F8 ; is A greater than #F8 ?
2E4A D2552E JP NC,2E55 ; yes, make spring inactive – jump to label EEEE
;
2E4D DD7705 LD (IX+#05),A ; store A in Y-position of spring
2E50 21BB69 LD HL,#69BB ; load HL with spring sprite Y-position
2E53 77 LD (HL),A ; store A in Y-position of spring sprite
2E54 C9 RET ; return
;
2E55 DD360000 LD (IX+#00),#00 ; set spring inactive – label FFFF
2E55 DD360300 LD (IX+#03),#00 ; set spring inactive – label EEEE
;
2E59 C9 RET ; return

```

```
2DDB C32627 JP #2726 ; jump to additional code – jump to label XXXX
2DDE 00 NOP
```

en

```
2726 DD210065 LD IX,#6500 ; load IX with start spring table – label XXXX
272A 21B869 LD HL,#69B8 ; load HL with spring sprite
;
272A DD7E03 LD A,(IX+#03) ; load spring X-position to A
272D 7E LD A,(HL) ; load spring sprite X position to A
272E FE00 CP #00 ; is X-position 0 ?
2730 C2DF2D JP NZ,#2DDF ; no, jump back, continue original code
;
2733 CD5700 CALL #0057 ; load A with a random number
2736 E60F AND #0F ; A is now number between #00 and #0F
2738 C6A0 ADD A,#A0 ; A is now number between #A0 and #AF
;
273A DD360000 LD (IX+#00),#00 ; set spring in-active
273E DD7703 LD (IX+#03),A ; set spring X position to A
2741 DD360520 LD (IX+#05),#20 ; set spring Y position to #20
2745 DD360904 LD (IX+#09),#04 ; set spring size width #04
2749 DD360A04 LD (IX+#0A),#04 ; set spring size height #04
;
274D 21B869 LD HL,#69B8 ; load HL with spring sprite
2740 77 LD (HL),A ; set spring sprite X position to A
2751 23 INC HL ; next
2752 363B LD (HL),#3B ; set spring sprite number to open spring
2754 23 INC HL ; next
2755 3600 LD (HL),#00 ; set spring sprite color
2757 23 INC HL ; next
2758 3620 LD (HL),#20 ; set spring sprite Y position to #20
;
275A CD5700 CALL #0057 ; load A with a random number
275D E63F AND #3F ; A is now number between #00 and #3F
275F C618 ADD A,#18 ; A is now number between #18 and #57
2761 DD7701 LD (IX+#01),A ; store A in deployment timer
;
2764 C3DF2D JP #2DDF ; no, jump back, continue original code
```

Gaat op zich goed. Alleen bij hameren gaat het nog niet goed. Nog naar kijken.

Nu versie v0.78 opgeslagen.

Klopt, bij het hameren wordt de sprite X-positie aangepast. Dus daar op triggeren.

```
2E55 DD360300 LD (IX+#03),#00 ; set spring inactive – label EEEE
```

Aanpassen naar:

```
2E55 C3D02A JP #2AD0 ; jump to additional code  
2E58 0000 NOP NOP
```

en

```
2AD0 DD360300 LD (IX+#03),#00 ; set spring inactive – label EEEE  
2AD4 21B869 LD HL,#69B8 ; load HL with spring sprite X position  
2AD7 3600 LD (HL),#00 ; set spring sprite X position to open spring  
2AD9 C9 RET ; return
```

Ja dat werkt nu goed. Nu nog even testen met de andere velden. Maar zal wel goed gaan.

Nu versie v0.79 opgeslagen.

Het How High Can You Get scherm aanpassen qua layout en teksten en in dat scherm de randomization van het board en het level doen.

De tekst HOW HIGH CAN YOU GET ? veranderen in LET THE BARRELS ROLL !
Net zo lang: Aanpassen op #36CC:

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

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

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

Dan volgende toevoegen:

```
0BFD 21BC75 LD HL,#75BC ; load HL with screen location start for goofy kong  
;  
0C00 0E50 LD C,#50 ; C := #50 = start graphic for goofy kong  
;  
0C02 71 LD (HL),C ; draw part of goofy kong  
0C03 0C INC C ; next graphic  
0C04 2B DEC HL ; next screen location  
0C05 71 LD (HL),C ; draw part of goofy kong  
0C06 0C INC C ; next graphic  
0C07 2B DEC HL ; next screen location  
0C08 71 LD (HL),C ; draw part of goofy kong  
0C09 0C INC C ; next graphic  
0C0A 2B DEC HL ; next screen location  
0C0B 71 LD (HL),C ; draw part of goofy kong  
0C0C 79 LD A,C ; load A with graphic number  
0C0D FE67 CP #67 ; == #67 ? (are we done?)  
0C0F CA1A0C JP Z,#0C1A ; yes, skip next 4 steps  
;
```

```

0C12 0C      INC C          ; next C
0C13 112300 LD   DE,#0023    ; load DE with offset
0C16 19      ADD HL,DE      ; add to screen location
0C17 C3020C JP   #0C02      ; loop again

```

En dat is goed. Nu nog een blauwe barrel sprite in zijn handen geven.

```

0C1A 21A869 LD   HL,#69A8      ; load HL with barrel sprite
0C1D 36D6   LD   (HL),#D6      ; set barrel sprite x-position
0C1F 23     INC HL          ; next
0C20 3616   LD   (HL),#16      ; set barrel sprite
0C22 23     INC HL          ; next
0C23 360C   LD   (HL),#0C      ; set barrel sprite color
0C25 23     INC HL          ; next
0C26 36C8   LD   HL,#C8      ; set barrel sprite y-position

```

Ziet er goed uit.

Nu versie v0.80 opgeslagen.

Nu ervoor zorgen dat board en level random gekozen worden.

Variabelen die hier een rol bij spelen en die op verschillende plekken gebruikt en gezet worden zijn:

- #6227 : screen number
- #6229 : level number
- #622A : pointer in de levels/screens data
- #622E : number of Goofy Kongs to draw
- #6388 : end of level counter

Deze worden in ieder geval op twee plekken gebruikt:

Bij het finishen van girders, elevators of het conveyors:

```

; jump here from #1622 when girders or elevators is finished. step 6 of 6
; jump here from #1622 when conveyors is finished. step 5 of 5

```

```

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
17A3 110005 LD   DE,#0500    ; load task #5, parameter 0 ; adds bonus to player's score
17A6 CD9F30 CALL  #309F      ; insert task
17A9 AF     XOR  A           ; A := 0
17AA 328863 LD   (#6388),A    ; clear end of level counter
17AD 210960 LD   HL,WaitTimerMSB ; load HL with timer addr.
17B0 3630   LD   (HL),#30      ; set timer to #30
17B2 23     INC HL          ; HL := GameMode2
17B3 3608   LD   (HL),#08      ; set game mode2 to 8
17B5 C9     RET             ; return

```

17B6 00 NOP

Maar het merendeel van deze code heb je in principe niet nodig omdat elke keer bij het tonen van het level progress screen (met de goofy kongs) het level en screen toch opnieuw berekend wordt.

Dus eigenlijk allemaal eruit halen en uiteindelijk een van de variabelen gaan gebruiken als een progress indicator.

Eerst eventjes zonder progress indicator alleen het level random genereren:

0C28 CD6727 CALL #2767 ; call routine to generate random level

en

```
2767 3A1960 LD A,(RngTimer2) ; load A with random timer value
276A E603 AND #03 ; mask bits, makes into #00-#03
276C 3C INC A ; increment, makes into #01-#04
276D 3C INC A ; increment makes into #02-#05
276E FE02 CP #02 ; A = 02?
2770 2006 JR NZ,BBBB ; no, skip next steps
;
2772 3A1860 LD A,(RngTimer1) ; load A with random timer value
2775 E601 AND #01 ; mask bits, makes into #00-#01
2777 3C INC A ; increment, makes into #01-#02
;
2778 3E01 LD A,#01 ; level override: normally 3xNOP
;
277A 322962 LD (#6229),A ; store into level nr
277D C9 RET ; return
```

Dat werkt goed.

En dan ook het board random genereren:

0C2B CD7E27 CALL #277E ; call routine to generate random board

en

```
277E 3A1A60 LD A,(FrameCounter) ; load A with FrameCounter
2781 E60F AND #0F ; mask bits, makes into #00-#15
2783 3C INC A ; increment, makes into #01-#16
2784 FE0D CP #0D ; A <= 13?
2786 381E JR C,27A6 ; yes, skip next steps
;
2788 3A1A60 LD A,(FrameCounter) ; load A with FrameCounter
278B OF RRCA
278C OF RRCA
278D OF RRCA
278E OF RRCA ; swap the high and low bytes
```

```

278F E60F AND #0F ; mask bits, makes into #00-#15
2791 3C INC A ; increment, makes into #01-#16
2792 FE0D CP #0D ; A <= 13?
2794 3810 JR C,27A6 ; yes, skip next steps
;
2796 3A1A60 LD A,(RngTimer2) ; load A with random timer value
2799 0F RRCA
279A 0F RRCA
279B 0F RRCA
279C 0F RRCA ; swap the high and low bytes
279D E60F AND #0F ; mask bits, makes into #00-#15
279F 3C INC A ; increment, makes into #01-#16
27A0 FE0D CP #0D ; A <= 13?
27A2 3802 JR C,27A6 ; yes, skip next steps
;
27A4 3E01 LD A,#01 ; set default board 1 if still undecided
;
27A6 3E01 LD A,#01 ; board override: normally 3xNOP
;
27A8 322762 LD (#6227),A ; store into board nr
27AB C9 RET ; return

```

Gaat niet helemaal goed. Geeft op gegeven moment ook een board 00? Hoe kan dat? Een keer debugger laten meelopen en kijken wat er precies gebeurt.

Debuggiung geeft aan dat de CP #0C niet goed is. #0C = 12 en niet 13 ! Aangepast naar CP #0D. En de AND #0F op #278F was verkeerd: 06EF ipv E60F. Ook aangepast. Werkt nu goed. Ook op eerste gezicht goed random. Veld 1 en veld 12 zijn geweest.

[Nu versie v0.81 opgeslagen.](#)

Nu het initiële zetten van de board and screen (op basis van de array met screen volgordes) eruit halen en op die plek in ieder geval de randomization count initializeren op 0.

De #622A met de table voor de levels wordt ook hier gebruikt:

```

; arrive from #0701 when GameMode2 == 3
;
09FE 214860 LD HL,P2NumLives ; source location is ???
0A01 112862 LD DE,#6228 ; destination is player lives remaining
0A04 010800 LD BC,#0008 ; byte counter set to 8
0A07 EDB0 LDIR ; copy
0A09 2A2A62 LD HL,(#622A) ; load HL with table for screens/levels
0A0C 7E LD A,(HL) ; load A with screen number from table
0A0D 322762 LD (#6227),A ; store A into screen number
0A10 3E78 LD A,#78 ; A := #78
0A12 320960 LD (WaitTimerMSB),A ; store into timer
0A15 3E04 LD A,#04 ; A := 4

```

```

0A17 320A60 LD (GameMode2),A ; store into game mode2
0A1A C9 RET ; return

```

De #622A met de table voor de levels wordt ook hier gebruikt:

```

; jump from #0701 when GameMode2 == 1
; copy player data, set screen, set next game mode based on number of players

```

```

09AB 214060 LD HL,P1NumLives ; load HL with source data location
09AE 112862 LD DE,#6228 ; load DE with destination data location.
09B1 010800 LD BC,#0008 ; byte counter set to 8
09B4 EDB0 LDIR ; copy (HL) into (DE) from P1NumLives to P2NumLives
09B6 2A2A62 LD HL,(#622A) ; EG #3A65. start of table data for screens/levels
09B9 7E LD A,(HL) ; load screen number from table
09BA 322762 LD (#6227),A ; store screen number
09BD 3A0F60 LD A,(TwoPlayerGame) ; load A with number of players
09C0 A7 AND A ; 1 player game?
09C1 210960 LD HL,WaitTimerMSB ; load HL with timer address
09C4 110A60 LD DE,GameMode2 ; load DE with game mode2 address
09C7 CAD009 JP Z,#09D0 ; if 1 player game, skip ahead

```

Initialiseren van de progress counter:

```

3F4C: 3E00 LD A,#00 ; A := 00
3F4E: 322A62 LD (#622A),A ; load progress number hundreds with 00
3F51: 3E00 LD A,#00 ; A := 00
3F53: 322E62 LD (#622E),A ; load progress number with 00
3F56: C9 RET ; skip ahead

```

En

```

09B6: CD4C3F CALL #3F4C ; call routine to set initial progress counters
0A09: CD4C3F CALL #3F4C ; call routine to set initial progress counters

```

Stukje code voor het tekenen van progress inclusief de honderdtallen. De tientallen opslaan in #622E en de honderdtallen opslaan in #622A.

```

27AC: 3A2E62 LD A,(#622E) ; load A with variable of the progress number
27AF: FE63 CP #63 ; A = 99 ?
27B1: C2C327 JP NZ,#27C3 ; no, skip next steps: jump to label AAAA
;
27B4: 3E00 LD A,#00 ; A := 0
27B6: 322E62 LD (#622E),A ; store into variable of the progress number
27B9: 3A2A62 LD A,(#622A) ; load A with variable of the progress number hundreds
27BC: 3C INC A ; increment A
27BD: 322A62 LD (#622A),A ; store into variable of the progress number hundreds
27C0: C3C727 JP #27C7 ; skip next steps: jump to label BBBB
;
27C3: 3C INC A ; increment A – label AAAA
27C4: 322E62 LD (#622E),A ; store into variable of the progress number
;
27C7: 3A2E62 LD A,(#622E) ; load A with progress number – label BBBB
27CA: 010AFF LD BC,#FF0A ; B: = #FF, C := #0A (10 decimal)
;
27CD: 04 INC B ; increment B – label CCCC
27CE: 91 SUB C ; subtract 10 decimal

```

```

27CF: D2CD27    JP    NC,#27CD      ; not carry, loop again – jump to label CCCC
;
27D2: 81        ADD   A,C          ; add 10 back to A to get a number from 0 to 9
27D3: 322675    LD    (#7526),A    ; draw progress to screen (singles)
27D6: 78        LD    A,B          ; load A with B (number of tens)
27D7: 324675    LD    (#7546),A    ; draw level to screen (tens)
27DA: 3A2A62    LD    A,(#622A)    ; load A with variable of progress number hundreds
27DD: 326675    LD    (#7566),A    ; draw level to screen (hundreds)
;
27E0: C9        RET

```

en

```
0C2E CDAC27    CALL   #27AC      ; call routine to increase progress counter
```

Nu versie v0.82 opgeslagen.

Nu opnieuw level selectie bekijken. Vaker level 5?

```

2767 3A1960  LD   A,(RngTimer2)  ; load A with random timer value
276A E607    AND   #07          ; mask bits, makes into #00-#07
276C 3C      INC   A           ; increment, makes into #01-#08
276D 00      NOP
276E FE05    CP    #05          ; A >= 5?
2770 3806    JR    C,BBBB       ; no, skip next steps
;
2772 3E05    LD    A,#05          ; load A with 5
2774 0000    NOP, NOP          ; 2x NOP
2776 0000    NOP, NOP          ; 2x NOP
;
2778 3E01    LD    A,#01          ; level override: normally 3xNOP
;
277A 322962  LD    (#6229),A    ; store into level nr
277D C9      RET

```

Nog de tekst BOARD COUNT = voorzetten:

3F57: 08 77 12 1F 11 22 14 10 13 1F 25 1E 24 10 34 10 3F

Een vrije tekst daarvoor gebruiken:

Tekst #6 is nog vrij. Dus aanpassen:

3657: 57 3F 6 ; BOARD COUNT =

En dan de text routine aanroepen:

```

27E0: 110603  LD    DE,#0306    ; load task data for text #6 "BOARD COUNT"
27E3: CD9F30  CALL   #309F      ; insert task to draw text
27E6: C9      RET

```

En dan de positionering van de count zelf moet 2 naar beneden en 1 naar rechts:

Dan wordt de #7526: 7526+2-20 = 7508

```

27D2: 81      ADD A,C          ; add 10 back to A to get a number from 0 to 9
27D3: 320875   LD (#7508),A    ; draw progress to screen (singles)
27D6: 78      LD A,B          ; load A with B (number of tens)
27D7: 322875   LD (#7528),A    ; draw level to screen (tens)
27DA: 3A2A62   LD A,(#622A)    ; load A with variable of progress number hundreds
27DD: 324875   LD (#7548),A    ; draw level to screen (hundreds)

```

Dit gaat niet goed bij two player spel omdat de counter niet opgeslagen wordt per speler. Dus de board count niet tonen bij een two-player game.

```

0C2E 3A0F60     LD A,(TwoPlayerGame) ; load A with # of players in game
0C31 A7         AND A           ; is this a 1 player game?
0C32 2003       JR NZ,#0C36    ; no skip next step
0C34 CDAC27     CALL #27AC     ; call routine to increase progress counter
0C37 00         NOP          ;

```

Dat gaat nu goed. De BOARD COUNT = tekst wordt weergegeven in de one player game, maar niet in de two player game.

Nu versie v0.83 opgeslagen.

Nu de tekst YOU PLAY BOARD XX AT LEVEL XX toevoegen.

Dit in vier stukken: de twee teksten YOU PLAY BOARD en AT LEVEL en de twee getallen.

Teksten 8 (#365B) en B (#3661) zijn nog vrij.

YOU PLAY BOARD: #3F68: 0D 77 29 1F 25 10 20 1C 11 29 10 12 1F 11 22 14 3F

AT LEVEL: #3F79: 4E 76 11 24 10 1C 15 26 15 1C 3F

365B 68 3F
3661 79 3F

```

2ADD 110803   LD DE,#0308    ; load task data for text #8 "YOU PLAY BOARD"
2AE0 CD9F30    CALL #309F     ; insert task to draw text
2AE3 110B03   LD DE,#030B    ; load task data for text #B "AT LEVEL"
2AE6 CD9F30    CALL #309F     ; insert task to draw text
2AE9 C9        RET          ; return

```

0C37 CDDD2A CALL #2ADD ; call routine to draw text YOU PLAY BOARD XX AT LEVEL YY

Nu versie v0.84 opgeslagen.

Het Board nummer weergeven:

```

2AE9 3A2762   LD A,(#6227)    ; load A with board #
2AEC 010AFF   LD BC,#FF0A    ; B: = #FF, C := #0A (10 decimal)

```

```

2AEF 04       INC B          ; increment B
2AF0 91       SUB C          ; subtract 10 decimal
2AF1 D2EF2A   JP NC,#2AEF    ; not carry, loop again (counts tens)

```

```

2AF4 81       ADD A,C          ; add 10 back to A to get a number from 0 to 9
2AF5 32ED74   LD (#74ED),A    ; draw level to screen (low byte)
2AF8 78       LD A,B          ; load a with B (number of tens)
2AF9 320D75   LD (#750D),A    ; draw level to screen (high byte)
2AFC C9       RET          ; return

```

En het level nummer weergeven:

```
2AFC 3A2962 LD A,(#6229) ; load A with board #
2AFF 010AFF LD BC,#FF0A ; B: = #FF, C := #0A (10 decimal)

2B02 04 INC B ; increment B
2B03 91 SUB C ; subtract 10 decimal
2B04 D2022B JP NC,#2B02 ; not carry, loop again (counts tens)

2B07 81 ADD A,C ; add 10 back to A to get a number from 0 to 9
2B08 32EE74 LD (#74EE),A ; draw level to screen (low byte)
2B0B 78 LD A,B ; load a with B (number of tens)
2B0C 320E75 LD (#750E),A ; draw level to screen (high byte)
2B0F C9 RET ; return
```

Dat gaat goed.

Nu versie v0.85 opgeslagen.

De heart sprite random maken.

De originele heart sprite zit op nummers: #76 en #77. De alternatieve heart sprite zit op nummers #56 en #57.

De heart sprite wordt hier gezet:

```
170B 21206A LD HL,#6A20 ; load HL with heart sprite
170E 3680 LD (HL),#80 ; set heart sprite X position
1710 23 INC HL ; next
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
1717 3620 LD (HL),#20 ; set heart sprite Y position
```

En wordt hier in de broken heart sprite veranderd:

```
174E 21216A LD HL,#6A21 ; load HL with heart sprite
1751 34 INC (HL) ; change heart to broken
```

Dat laatste is eenvoudig de volgende sprite kiezen. Omdat ze in allebei de gevallen na elkaar zitten, hoeft dit niet aangepast te worden.

Aanpassen:

170B C3CB2F JP #2FCB ; jump to additional code – jump to label **AAAA**

en

1711 77 LD (HL),A ; set heart sprite
1712 00 NOP

en

2FCB 3A1A60 LD A,(#181A) ; load A with RngTimer1 – label **AAAA**
2FCE E601 AND #01 ; AND A with #01, A = 0 or 1.
2FD0 FE00 CP #00 ; A==0?
2FD2 2004 JR NZ,#2FD8 ; No, skip next steps
;
2FD4 3E76 LD A,#76 ; load original heart sprite
2FD6 1802 JR #2FDA ; skip next step
;
2FD8 3E56 LD A,#56 ; load alternative heart sprite
;
2FDA 21206A LD HL,#6A20 ; load HL with heart sprite
2FDD C30E17 JP #170E ; jump back

Gaat nu goed.

Nu versie v0.86 opgeslagen.

Even kijken hoe dat werkt met die tabel met teksten. Kunnen we dat ook met een eigen tabel aanroepen?

De tabel wordt hier uitgelezen.

```
; task #3
; draws text to screen
; called from #0306 with code for text to draw in A

05E9 214B36    LD    HL,#364B     ; start of table data
05EC 87        ADD   A,A          ; double the parameter
05ED F5        PUSH AF          ; save AF to stack
05EE E67F       AND   #7F          ; mask bits
05F0 5F        LD    E,A          ; copy to E
```

En de tabel zelf ziet er zo uit:

```

; #364B is used from #05E9

364B: 8B 36      0      ; #368B "GAME OVER"
364D: 01 00      1      ; unused ?
364F: 98 36      2      ; #3698 "PLAYER <I>"
3651: A5 36      3      ; #36A5 "PLAYER <II>"
3653: B2 36      4      ; #36B2 "HIGH SCORE"

```

En dan op de plek waar de tabel heenwijst start de byte sequence met eerst twee bytes voor de schermpositie, daarna de bytes met de tekst afgesloten met 3F.

In principe zou het boardnummer (screen nr) als index gebruikt worden in de tabel. Dan board number – 1 geeft getal van #00 t/m #0B (0 t/m 11).

Lastige is dat de code terug springt met een dubbele return (volgens de comments). Gaat dat dan wel goed?

Nieuwe functie maken:

```

27E7 21F027    LD     HL,#27F0    ; load HL with start of table
27EA C3EC05    JP     #05EC    ; jump to text draw routine
27ED C9        RET    ; return

27F0 F4 27      ; table index

27F4 EF 74 18 15 1C 1C 1F 3F    ; string definition

0C3A 3E00        LD     A,#00    ; load A with index 0
0C3C CDE727    CALL   27E7    ; call additional text draw routine with additional table

```

En dat gaat goed. Ook nog testen met een tweede entry:

27F2 FC 27

27FC EF 74 27 1F 22 1C 14 3F

```

0C3A 3E01        LD     A,#01    ; load A with index 1
0C3C CDE727    CALL   27E7    ; call additional text draw routine with additional table

```

Ja dat gaat ook goed.

Dan nu eerst ruimte vrijmaken hiervoor. Terug naar versie v0.86 en dan de code voor de animatie waarin Donkey Kong met Pauline naar boven klimt vrijgeven.

Dit heet in de comments het intro screen. De code is #0A76 t/m #0BD9. De aanroep daarnaartoe staat hier:

```

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

```

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

Dat gaat niet goed. Klopt: de gamemode moet wel aangepast worden. Dus de aanroep weer terugzetten en dan de code aanpassen:

```

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

```

En dan vrijgeven #0A7B t/m 0BD9.

Nieuwe tekst teken functie die werkt met een additionele tabel:

```

27E7 217B0A LD HL,#0A7B      ; load HL with start of table
27EA C3EC05 JP #05EC        ; jump to text draw routine
27ED C9      RET           ; return

```

De tabel begint op #0A7B en is 12 indexen lang: #00 t/m #0B.

De aanroep voor het tekenen:

```

0C3A 3A2762 LD A,(#6227)    ; load A with board number
0C3D 3D      DEC A
0C3E CDE727 CALL 27E7       ; call additional text draw routine with additional table

```

In eerste instantie die allemaal laten verwijzen naar dezelfde tekst die begint op positie 0A93.

```

0A7B 93 0A      ; draw tekst board 01
0A7D A6 0A      ; draw tekst board 02
0A7F B8 0A      ; draw tekst board 03
0A81 C7 0A      ; draw tekst board 04
0A83 D6 0A      ; draw tekst board 05
0A85 EA 0A      ; draw tekst board 06
0A87 FB 0A      ; draw tekst board 07
0A89 0B 0B      ; draw tekst board 08
0A8B 1C 0B      ; draw tekst board 09
0A8D 30 0B      ; draw tekst board 10
0A8F 44 0B      ; draw tekst board 11
0A91 54 0B      ; draw tekst board 12

```

0A93: D3 76 3B 23 24 15 15 22 19 1E 17 10 12 1F 1E 25 23 3B 3F

Board 01: "STEERING BONUS"

0AA6: F3 76 3B 19 1E 24 1F 10 24 18 15 10 14 11 22 1B 3B 3F
Board 02: "INTO THE DARK"

0AB8: B3 76 3B 13 22 11 2A 29 10 1B 1F 1E 17 3B 3F
Board 03: "CRAZY KONG"

0AC7: B3 76 3B 1D 1F 22 15 10 27 19 1C 14 23 3B 3F
Board 04: "MORE WILDS"

0AD6: 13 77 3B 20 11 25 1C 19 1E 15 3A 23 10 18 15 11 22 24 3B 3F
Board 05: "PAULINE'S HEART"

0AEA: D3 76 3B 12 1F 24 18 10 18 11 1D 15 22 23 3B 3F
Board 06: "BOTH HAMMERS"

Nu versie v0.87 opgeslagen.

0AFB: D3 76 3B 22 15 26 15 22 23 15 10 22 25 1E 3B 3F
Board 07: "REVERSE RUN"

0B0B: D3 76 3B 23 1C 1F 27 10 12 11 22 22 15 1C 23 3B 3F
Board 08: "SLOW BARRELS"

0B1C: 13 77 3B 23 18 19 16 24 15 14 10 17 19 22 14 15 22 23 3B 3F
Board 09: "SHIFTED GIRDERS"

0B30: 13 77 3B 20 11 25 1C 19 1E 15 3A 23 10 19 24 15 1D 23 3B 3F
Board 10: "PAULINE'S ITEMS"

0B44: D3 76 3B 23 20 22 19 1E 17 10 24 19 1D 15 3B 3F
Board 11: "SPRING TIME"

0B54: D3 76 3B 11 1E 24 19 2C 17 22 11 26 19 24 29 3B 3F
Board 12: "ANTI-GRAVITY"

Nu versie v0.88 opgeslagen.

Het jaartal aanpassen naar 2021.

In #3F00 de copyright notice maken:

© 1981,2021 NINTENDO

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

En de alternatieve broken heart sprite aangepast.

Nu versie v0.89 opgeslagen.

Nu nog eventjes een test doen met de randomization verdeling. Test gedaan met 66 games. De verdeling bij de levels is netjes zoals beoogd. De verdeling bij de boards is redelijk, met board 7 relatief veel en board 5 en 6 relatief weinig. Maar daarvoor wellicht te weinig games gespeeld.

Nog eventjes testen met two-player game en met herstarten game. Gaat goed.

Nu de invincibility uitzetten.

[Nu versie v0.90 opgeslagen.](#)

Randomization van de boards wellicht aanpassen:

```
277E 3A1A60 LD A,(FrameCounter) ; load A with FrameCounter
2781 E60F AND #0F ; mask bits, makes into #00-#15
2783 3C INC A ; increment, makes into #01-#16
2784 FE0D CP #0D ; A <= 13?
2786 381E JR C,27A6 ; yes, skip next steps
;
2788 3A1A60 LD A,(FrameCounter) ; load A with FrameCounter
278B OF RRCA
278C OF RRCA
278D OF RRCA
278E OF RRCA ; swap the high and low bytes
278F E60F AND #0F ; mask bits, makes into #00-#15
2791 3C INC A ; increment, makes into #01-#16
2792 FE0D CP #0D ; A <= 13?
2794 3810 JR C,27A6 ; yes, skip next steps
;
2796 3A1A60 LD A,(RngTimer2) ; load A with random timer value
2799 OF RRCA
279A OF RRCA
279B OF RRCA
279C OF RRCA ; swap the high and low bytes
279D E60F AND #0F ; mask bits, makes into #00-#15
279F 3C INC A ; increment, makes into #01-#16
27A0 FE0D CP #0D ; A <= 13?
27A2 3802 JR C,27A6 ; yes, skip next steps
;
27A4 3E01 LD A,#01 ; set default board 1 if still undecided
;
27A6 3E01 LD A,#01 ; board override: normally 3xNOP
;
27A8 322762 LD (#6227),A ; store into board nr
27AB C9 RET ; return
```

Doet drie keer bepalen. Maar wel foutje. De derde keer zou op basis van RngTimer2 moeten zijn, maar code geeft adres van de FrameCounter.

```
FrameCounter = #601A  
RngTimer1 = #6018  
RngTimer2 = #6019
```

Even proberen met wat de bedoeling was:

Framecounter low bits -> FrameCounter high bits -> RngTimer2 high bits

Even proberen met als eerste call naar rng generator:

Call Rng gerator -> FrameCounter high bits -> RngTimer2 high bits

```
277E CD5700 CALL #0057 ; load A with a random number
```

Ja eventjes zo houden.

Nu nog even kijken naar het vreemde gedrag tijdens de attract mode, waarin de 1UP ineens naar 0UP verandert. Dat moeten helemaal niet kunnen.

Aanpassen van de 1UP wordt in deze code gedaan:

```
0A53 3E01 LD A,#01 ; load A with "1"  
0A55 324077 LD (#7740),A ; write to screen  
0A58 3E25 LD A,#25 ; load A with "U"  
0A5A 322077 LD (#7720),A ; write to screen  
0A5D 3E20 LD A,#20 ; load A with "P"  
0A5F 320077 LD (#7700),A ; write to screen  
0A62 C9 RET ; return
```

Als er echter een watchpoint op #0A55 gezet wordt, dan schrijft hij een 0 naar #7740. Dat kan niet als het statement ervoor A vult met #01. Dus waarschijnlijk wordt ergens naar 0A55 gesprongen.

Als een watchpoint op #0A53 gezet wordt, wordt er niet gestopt. Dat betekent dus dat deze statement niet uitgevoerd wordt. Klopt met beide watchpoints aktief. Alleen het watchpoint op #0A55 wordt dan getriggerd.

Uitzoeken waar dat zit. Zit dat in de originele code, of zit er in mijn code ergens een fout?

In dit document kan ik met find niets vinden met betrekking tot #0A55.

In de originele code is er geen jump of call naar #0A55.

Ook in de code niets terug te vinden via search in HxD.

Kunnen we in de debugger een trace doen van de code, zodat we kunnen zien wat er uitgevoerd is net voor het watchpoint?

Tracelog geeft dit aan:

```

0492: nop
0493: nop
0494: nop
0495: ld a,c
0496: and a
0497: jp z,$04A1
049A: jp $22A3
22A3: cp $40
22A5: jp z,$22B0
22A8: cp $7F
22AA: jp z,$0A41
0A41: sbc a,a
0A42: jr nc,$0A55
0A55: ld ($7740),a
0A58: ld a,$25

```

En de code waar het dus blijkbaar niet goed gaat:

```

22A3 FE40      CP    #40          ; A = #40 ? - label AAAA
22A5 CAB022    JP    Z,#22B0      ; yes, display help text – jump to label BBBB
;
22A8 FE7F      CP    #7F          ; A = #7F ?
22AA CA410A    JP    Z,#0A41      ; yes, jump back and remove help text
;
22AD C3A904    JP    #04A9        ; jump back without changing help text
;
22B0 3A1A60    LD    A,(FrameCounter) ; load A with frame counter ? – label BBBB
22B3 E603      AND   #03          ; mask low 2 bits: number between 0 and 3
22B5 060A      LD    B, #0A        ; load B with 0A
22B7 80        ADD   A,B          ; add B to A: add #0A to A
22B8 0F        RRCA            ; 4 time rotate left: swap high and low bytes
22B9 0F        RRCA            ;
22BA 0F        RRCA            ;
22BB 0F        RRCA            ;
22BC F60F      OR    #0F          ; and #0F
22BE C3A304    JP    #04A3        ; jump back and display help text

```

Vanuit #049A wordt er naar de code in #22A3 gesprongen:

```

049A C3A322    JP    #22A3        ; jump to additional code – jump to label AAAA
049D 00000000  NOP, NOP, NOP, NOP

```

De gele lijn moet zijn:

```

22AA CAA104    JP    Z,#04A1      ; yes, jump back and remove help text

```

Gaat nu wel goed.

Nu versie v0.91 opgeslagen.

Terug naar een releasejaar van 2020.

In #3F00 de copyright notice maken:

© 1981,2020 NINTENDO

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

Nu als versie v1.00 opgeslagen.

Kijken of we de board randomization kunnen aanpassen naar een gelijkere verdeling. Idee is dat de boards random gekozen worden, dat vastgelegd wordt wanneer een board gekozen is en dat een board pas opnieuw gekozen kan worden als alle 12 boards een keer aan de beurt geweest zijn. Dan registratie schonen en opnieuw. Dus bij 12 boards, allemaal één keer gekozen. Bij 24 boards, allemaal twee keer gekozen., enz.

Dus twaalf geheugenplaatsen nodig. De datastructure voor de pies gebruiken? Is deze nog helemaal vrij?

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

65A0 is wel al gebruikt als blink counter voor de hamers.

Wat als we 65A1 t/m 65AC kunnen gebruiken. Eerst met debugger kijken of deze waarden niet gevuld worden nu. Nee die worden niet gebruikt. Ook niet in screen 6 met de blinking hammers.

Globale logica:

Bepaal een random board nummer

Check of deze vrij is (#65A0+boardnumber is nog 0).

Is deze vrij?

Ja:

Zet deze op gebruikt (#65A0+boardnumber op 1 zetten)

Ga naar bepalen van random level number

Nee:

Maak C gelijk aan boardnumber

Loop X

Verhoog C

Is C gelijk aan 13?

Ja:

Maak C gelijk aan 1

Is C gelijk aan boardnumber (alle twaalf gebruikt)?

Ja:

Zet #65A1 /tm #65AC op 0

Zet #65A0+boardnummer op gebruikt (op 1)

Ga naar bepalen van random level number

Check of het slot vrij is (#65A0+C is nog 0)

Is deze vrij?

Ja:

Maak boardnummer gelijk aan C

Zet deze op gebruikt (#65A0+boardnumber op 1 zetten)

Ga naar bepalen van random level number

Check volgende slot: Jump naar Loop X

Het bepalen van het boardnummer gebeurt hier:

```
.....  
27A4 3E01 LD A,#01 ; set default board 1 if still undecided  
;  
27A6 3E01 LD A,#01 ; board override: normally 3xNOP  
;  
27A8 322762 LD (#6227), A ; store into board nr  
27AB C9 RET ; return
```

Dus 'Ga naar bepalen van random level number' is een simpele RET.

Aanpassen

```
27A8 C3650B JP #0B65 ; jump to additional code, jump to label AAAA  
27AB 00 NOP
```

en

```
0B65 4F LD C,A ; save board number to C – label AAAA  
;  
0B66 21A065 LD HL,#65A0 ; load HL with start of board usage table minus 1  
0B69 5F LD E,A ; copy board number to E  
0B6A 1600 LD D,#00 ; D := 0  
0B6C 19 ADD HL,DE ; add index into board usage table  
;  
0B6D 7E LD A,(HL) ; check if board number is already used  
0B6E FE01 CP #01  
0B70 2025 JR NZ,#0B97 ; no, board number is not used, jump to label BBBB  
;  
0B72 23 INC HL ; step to next entry in board usage table – label XXXX  
0B73 1C INC E ; increment possible board number  
;  
0B74 7B LD A,E ; load possible board number into A  
0B75 FE0D CP #0D ; Is possible board number 13 (#0D)?  
0B77 2005 JR NZ,#0B7E ; no, skip next steps – jump to label CCCC  
;  
0B79 21A165 LD HL,#65A1 ; reset entry in board usage table  
0B7C 1E01 LD E,#01 ; reset possible board number to 1  
;
```

```

0B7E 79      LD   A,C          ; load A with the original board number – label CCCC
0B7F BB      CP   E           ; is it equal to the possible board number?
0B80 2807    JR   Z,#0B89    ; yes, all boards used, jump to label DDDD
;
0B82 7E      LD   A,(HL)     ; check if possible board number is already used
0B83 FE01    CP   #01        ;
0B85 28EB    JR   Z,#0B72    ; yes, already used, check next – jump to label XXXX
;
0B87 180E    JR   #0B97      ; no, board number is not used, jump to label BBBB
;
0B89 21A165  LD   HL,#65A1    ; load HL with start of board usage table – label DDDD
0B8C 060C    LD   B,#0C      ; for B = 1 to #0C (12 decimal)
0B8E 3600    LD   (HL),#00    ; reset entry in board usage table – label EEEE
0B90 23      INC  HL         ;
0B91 10FB    DJNZ #0B8E     ; next B – jump to label EEEE
;
0B93 21A065  LD   HL,#65A0    ; load HL with start of board usage table minus 1
0B96 19      ADD  HL,DE      ; add index into board usage table
;
0B97 3601    LD   (HL),#01    ; set board number to used – label BBBB
0B99 7B      LD   A,E         ;
0B9A 322762  LD   (#6227),A    ; store into board nr
0B9D C9      RET             ; return

```

Gaat niet goed. In de tabel wordt het juiste scherm op gebruikt gezet. Maar bij starten van het spel wordt de hele structuur weer overgeschreven met nullen.

Waar wordt dat gedaan? Debugging geeft aan dat dat gebeurd op #0F68.

```

0F60 0E11    LD   C,#11      ; For C = 1 to 11
0F62 1680    LD   D,#80      ; load D with 80, used to reset B in inner loop
0F64 218062  LD   HL,#6280    ; start of memory to clear
0F67 42      LD   B,D      ; For B = 1 to #80

0F68 77      LD   (HL),A    ; clear (HL)
0F69 23      INC  HL         ; next memory
0F6A 10FC    DJNZ #0F68    ; Next B

0F6C 0D      DEC  C          ; Next C
0F6D 20F8    JR   NZ,#0F67    ; loop until done

```

Checken op #65A1 t/m 65AC: die niet wissen. HL: H=65 en L>=A1 en L<=AC.

Aanpassen:

```

0F67 C3B00B  JP   #0BB0      ; jump to additional code – jump to label AAAA
0F6A 0000    NOP,NOP

```

en

```
0BB0 42      LD   B,D          ; For B = 1 to #80 – label AAAA
;
0BB1 F5      PUSH AF          ; save A for later – label XXXX
0BB2 7C      LD   A,H          ;
0BB3 FE65      CP   #65          ; H = #65?
0BB5 200C      JR   NZ,#BBBB      ; no, clear memory address – jump to label BBBB
;
0BB7 7D      LD   A,L          ;
0BB8 FEA1      CP   #A1          ; L < #A1
0BBA 3807      JR   C,#BBBB      ; yes, clear memory address – jump to label BBBB
0BBC FEAD      CP   #AD          ; L >= #AD
0BBE 3003      JR   NC,#BBBB      ; yes, clear memory address – jump to label BBBB
;
0BC0 F1      POP  AF          ; restore A
0BC1 1802      JR   #CCCC          ; do not clear memory address – jump to label CCCC
;
0BC3 F1      POP  AF          ; restore A – label BBBB
0BC4 77      LD   (HL),A          ; clear (HL)
0BC5 23      INC  HL          ; next memory – label CCCC
0BC6 10E9      DJNZ #XXXX          ; Next B – jump to label XXXX
;
0BC8 C36C0F      JP   #0F6C          ; jump back
```

Lijkt redelijk goed te gaan. Maar op gegeven moment een reset van de game.

D.m.v. debugging vier fouten eruit gehaald. Gaat nu wel goed.

Alleen nog testen verdeling random boards op basis van 48 boards en speelbaarheid boards.

Daarna v1.01 maken en invincibility uitzetten.

Versie 1.01 maken:

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

Nu versie v1.01 opgeslagen.

Kijken of we het programma zo kunnen maken dat bij starten als eerste het titelscherm getoond wordt. Niet in de volgorde aanpassen. Dat is eerder geprobeerd, maar dat ging niet goed. Maar de fase wordt aangeduid met een variabele, misschien deze bij het starten anders initialiseren.

Wordt bepaald door de GameMode2: die wordt hier gezet:

```

01E2 3E01 LD A,#01 ; A := 1
01E4 32827D LD (REG_FLIPSCREEN),A ; store into flip screen setting
01E7 320560 LD (GameMode1),A ; store into game mode 1
01EA 322762 LD (#6227),A ; initialize screen to 1 (girders)
01ED AF XOR A ; A := 0
01EE 320A60 LD (GameMode2),A ; store into game mode 2

```

En hier is te zien dat introscherm overeenkomt met GameMode2 = 06:

```

0746 7E LD A,(HL) ; else load A with game mode2
0747 EF RST #28 ; jump based on A

0748 79 07 0 ; #0779 ; clear screen, set color palettes, draw attract mode text and high score table,
0749 63 07 1 ; #0763 ;
074C 3C 12 2 ; #123C ; set initial mario sprite position and draw remaining lives and level
074E 77 19 3 ; #1977 ; set artificial input for demo play [change to #197A to enable playing in demo part 1/2]
0750 7C 12 4 ; #127C ; handle mario dying animations
0752 C3 07 5 ; #07C3 ; clears the screen and sprites and increase game mode2
0754 CB 07 6 ; #07CB ; handle intro splash screen ?
0756 48 08 7 ; #084B ; counts down a timer then resets game mode2 to 0

```

Eventueel aanpassen:

```

01ED C3813F JP #3F81 ; jump to additional code – jump to label AAAA
01F0 00 NOP

```

en

```

3F84 3E06 LD A,#06 ; A := 6 – label AAAA
3F86 320A60 LD (GameMode2),A ; store into game mode 2
3F89 C3F101 JP #01F1 ; jump back

```

Ja dat werkt goed. Leuk !

Eventjes het jaartal op 2021 zetten:

In #3F00 de copyright notice maken:

© 1981,2021 NINTENDO

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

En de versie ophogen naar v1.02:

#373E: 3D 76 26 01 2B 00 02 10 3F (V1.02)

Deze versie is de final versie v1.02.

Board #05 (Paulines Heart) iets makkelijker maken door de eerste barrel geen blauwe te laten zijn. Dan heb je iets meer spelings om bij het hartje te komen voordat er een fireball al naar boven komt naar de gilder met de onderste hamer. Anders is dat nogal een afhankelijkheid van de rng. En dan een fireball die al omhoog gekomen is, is lastig te handelen. En dan hangt het dus nogal van geluk af of de fireball beneden blijft of al omhoog gaat.

De eerste barrel blauw wordt hier bepaald:

```

2C4F 328F63 LD (#638F),A ; store A into the state of the barrel deployment between 3 and 0
2C52 3E01 LD A,#01 ; A := 1
2C54 329263 LD (#6392),A ; set barrel deployment indicator
2C57 3Ab262 LD A,(#62B2) ; load A with blue barrel counter
2C5A B9 CP C ; compare with current timer
2C5B C0 RET NZ ; return if not equal

2C5C D608 SUB #08 ; if equal then this will be a blue barrel. decrement A by 8
2C5E 32B262 LD (#62B2),A ; put back into blue barrel counter

```

Geheugenplek #62B2 is een counter op basis waarvan de blue barrel bepaald wordt. Op het moment dat het een blue barrel, is wordt er meteen #08 vanaf getrokken.

In level L=01 is #62B2 voor de eerste barrel #32 en wordt deze daarna #2A.

In level L=05 is #62B2 voor de eerste barrel #50 en wordt deze daarna #48.

Dus de startwaarde van #62B2 verschilt per level.

Via debugger is voor het bepalen van de eerste barrel de waarde meteen verminderd met #08. Dan wordt er inderdaad geen blue barrel gegooid. Dus de oplossing is om ergens voordat deze check de eerste keer gedaan wordt, de waarde van #62B2 te verminderen met #08.

Nu nog uitzoeken waar dat gedaan kan worden. De startwaarde van 62B2 wordt hier gezet:

```

; set up initial timer
; timer is either 5000, 6000, 7000 or 8000 depending on level

0F7A 3A2962 LD A,(#6229) ; load level number
0F7D 47 LD B,A ; copy to B
0F7E A7 AND a ; clear carry flag
0F7f 17 RLA ; rotate A left (double =2x)
0F80 A7 AND a ; clear carry flag
0F81 17 RLA ; rotate A left (double again =4x)
0F82 A7 AND a ; clear carry flag
0F83 17 RLA ; rotate A left (double again = 8x)
0F84 80 ADD A,b ; add B into A (add once = 9x)
0F85 80 ADD A,b ; add B into A (add again = 10x)
0F86 C628 ADD A,#28 ; add #28 (40 decimal) to A
0F88 FE51 CP #51 ; < #51 ?
0F8A 3802 JR c,#0F8E ; yes, skip next step

0F8C 3E50 LD A,#50 ; otherwise load A with #50 (80 decimal)

0F8E 21B062 LD HL,#62B0 ; load HL with start of timers
0F91 0603 LD B,#03 ; For B = 1 to 3

0F93 77 LD (HL),A ; store A into timer memory
0F94 2C INC I ; next memory
0F95 10Fc DJNZ #0F93 ; Next B

```

Dus wellicht na de loop waarin het gezet wordt #0F91 t/m #0F96 een stukje code tussenvoegen waarin alleen voor board #05 de waarde van #62B2 met #08 verlaagd wordt.

Eerst onsterfelijkheid aanzetten. Zodat het testen iets makkelijker gaat.

En ervoor zorgen dat door middel van een override altijd board 05 gekozen wordt, zodat we het goed kunnen testen.

De board override zit als het goed is hier:

```
27A4 3E01 LD A,#01 ; set default board 1 if still undecided  
;  
27A6 3E01 LD A,#01 ; board override; normally 3xNOP  
;  
27A8 322762 LD (#6227),A ; store into board nr  
27AB C9 RET ; return
```

27A6 aanpassen naar 3E05 (en later weer terugzetten naar 0000).

Aanpassing

```
0F97 CD440C CALL #0C44 ; call additional code – jump to label AAAA  
0F9A 00 NOP
```

en:

```
0C41 C3820C JP #0C82 ; skip empty space so space comes available
```

en:

```
0C44 47 LD B,A ; store A in B for later – label AAAA  
0C45 3A2762 LD A,(#6227) ; load A with screen number  
0C48 FE05 CP #05 ; this screen 05 ?  
0C4A 2008 JR NZ,#0C54 ; no skip next steps – jump to label BBBB  
;  
0C4C 3AB262 LD A,(#62B2) ; load A with blue barrel counter  
0C4F D608 SUB #08 ; subtract #08  
0C51 32B262 LD (#62B2),A ; store into blue barrel counter  
;  
0C54 78 LD A,B ; restore A from B – label BBBB  
0C55 87 ADD A,A ; add A with A (double a).  
0C56 47 LD B,A ; copy to B  
0C57 3EDC LD A,#DC ; A := #DC (220 decimal)  
0C59 C9 RET ; return
```

Waar zou het stukje code (22 bytes) kunnen komen? Vanaf #0C41 zijn er voldoende bytes vrij. Maar dat is een vrij blok dat als NOP's overgeslagen wordt in het 'How High' scherm. Dus daar kan niet zomaar iets tussengezet worden. Daarom daar eerst een jump invoegen en dan pas de extra code.

Ja dat lijkt te werken. En dan de 8e barrel is dan een blauwe. Misschien aan te passen door niet 8 eraf te trekken, maar bijvoorbeeld 4: wordt er dan na 4 barrels een blauwe gegeven? Ja dat klopt.

Eens kijken of je daarmee kunt tunen. Strategie zou kunnen zijn dat je de blauwe met de bovenste hamer kapot gaat slaan zodat je voorkomt dat er een fireball komt. Maar zelfs met een waarde #01, zodat de tweede barrel meteen een blauwe is, kun je het voor blijven en de blauwe barrel kapot slaan.

Andere oplossing zou kunnen zijn om de eerste barrel wel blauw te laten zijn, maar deze niet crazy te maken, dus gewoon rollen. Dan heb je niet de tijd om hem kapot te slaan en krijg je wel problemen met het voorblijven van de barrels. Ook eventjes proberen.

Eventjes weer terug naar final v1.02 versie en weer invincible maken en board override voor board 5.

De eerste barrel crazy maken gebeurt hier:

```
; we arrive here if timer is within first 2 clicks when deploying a barrel from #2C18

2C7B C602      ADD    A,#02          ; A := A + 2 (A had the initial clock value -2, now it has the initial clock value)
2C7D B9        CP     C              ; compare to current timer value - are we starting this round now?
2C7E CA492C    JP     Z,#2C49        ; yes, do a crazy barrel

2C81 3E02      LD     A,#02          ; else A := 2 for the second barrel; it is always normal
2C83 C34B2C    JP     #2C4B          ; jump back and continue deployment
```

Dus dan de jump in 2C7E niet doen als screen level 05 is.

Aanpassen:

```
2C7E CA9E0B  JP     Z,#0B9E        ; jump to additional code – jump to label AAAA
```

en:

```
0B9E 3A2762  LD     A,(#6227)      ; load A with screen number – label AAAA
0BA1 FE05    CP     #05            ; this screen 05 ?
0BA3 C2492C  JP     NZ,#2C49        ; no jump back, do a crazy barrel
;
0BA6 C3812C  JP     #2C81          ; yes jump back, don't do a crazy barrel
```

Ja dat werkt goed. Wat wel mooi is dat je meteen opgejaagd wordt door de blauwe barrel en dat je die niet kunt kapot slaan met de bovenste hamer. Dus die komt beneden en wordt dan een fireball. Dus je moet niet te lang treuzelen. Maar de fireball is nu niet de bepalende factor meer. En kun je je concentreren op de weg naar beneden en het ontwijken van de barrels. Het is voor gevorderde spelers wellicht wel mogelijk om met een backjump de hamer te pakken en de blauwe barrel toch te slaan, maar dat is niet eenvoudig. Dus leuke aanpassing zo.

Nogmaals kijken of er nog andere boards zijn met eventuele afhankelijkheid met de rng. Dus dat de geluksfactor een te grote rol speelt. Nee, de rest niet echt. Of een gewoon board qua spelen, dus net

zoals gewone barrels board. De echt aangepaste boards zijn voor mij te spelen, dus lijkt me geen probleem. ‘Slow barrels’ (board 08) en ‘Anti-Gravity’ (board 12) zijn wel lastig, maar niet ondoenlijk. Daar eventjes mee testen met board override.

De ‘Slow barrels’ is te doen. De ‘Anti-gravity’ is wel vrij lastig. Maar dat is gewenning. Met sturen kun je wel wat bereiken. Daarnaast goed opletten hoe je springt. Als de barrels met korte tussenruimte dan laat recht omhoog springen. Dan gaat hij net onder je door en dan kun je de volgende ook nog springen. Tenzij die te dichtbij, maar dan lange sprong opzij maken. Als barrel combinatie die niet te springen is, dan terug naar lagere girder en kijken of je kan sturen.

En de versie ophogen naar v1.03:

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

En de board-override weer weghalen (weer 2 NOP’s maken):

27A6 3E01 LD A,#01 ; **board override**: normally 2xNOP

Deze versie is de final versie v1.03.

Kijken of we de score de miljoenen kunnen laten weergeven. Ook al gedaan in DK randomized. Maar daar uiteindelijk komen te vervallen om ruimte terug te winnen. Kijken hoever we kunnen komen.

Eerst het scoren van punten aanpassen (DK RND blz 21):

3529: 00 00 00
352C: 10 00 00
352F: 20 00 00
3532: 30 00 00
3535: 40 00 00
3538: 50 00 00
353B: 60 00 00
353E: 70 00 00
3541: 80 00 00
3544: 90 00 00
3547: 00 00 00
354A: 00 01 00
354D: 00 02 00
3550: 00 03 00
3553: 00 04 00
3556: 00 05 00
3559: 00 06 00
355C: 00 07 00
355F: 00 08 00
3562: 00 09 00

De score loopt nu op met tientallen.

De verschillende startposities van de score aanpassen (DK RND blz 22):

```
056B DD218177 LD IX,#7781 ; start at the same location  
0572 DD214175 LD IX,#7541 ; start one position left  
0578 DD216176 LD IX,#7661 ; start one position left
```

De routine van #0593 t/m #059A verplaatsen naar stuk vrije ruimte (#0504):

```
0504: E60F AND #0F ; mask out left 4 bits of A  
0506: DD7700 LD (IX+#00),A ; store A on screen  
0509: DD19 ADD IX,DE ; adjust to next location  
050B: C9 RET ; return
```

En aanpassen van de twee calls:

```
0588: CD0405 CALL #0504 ; draw to screen  
058C: CD0405 CALL #0504 ; draw to screen
```

Aanpassen aanroep naar additionele code op stukje vrije ruimte (#1AAC):

```
0592: C3AC1A JP #1AAC ; jump to additional code – jump to label AAAA
```

en

```
1AAC: DD7C LD A,IXH ; load IX high byte into A – label AAAA  
1AAE: FE74 CP #74 ; check if byte contains 74  
1AB0: CAB81A JP Z,#1AB8 ; skip forward – jump to BBBB  
;  
1AB3: 3E 00 LD A,#00 ; A := 00  
1AB5: DD7700 LD (IX+#00),A ; store A on screen  
;  
1AB8: C9 RET ; return – label BBBB
```

Gaat nu goed voor player 1 score en de credits zijn niet met een extra nul. Gaan ook goed.

Aanpassen ExtraLifeThreshold: bij 7000 (afhankelijk van dip switches) krijg je een extra leven. Maar dat werkt nu niet omdat niet in honderdtallen, maar in tientallen geteld wordt. Aanpassen:

```
0355 21B360 LD HL,#60B3 ; load HL with address for player 1 score  
...  
035E 21B660 LD HL,#60B6 ; else load HL with address of player 2 score  
...  
0361 7E LD A,(HL) ; load A with a byte of the player's score  
0362 E60F AND #0F ; mask bits (was F0)  
0364 47 LD B,A ; copy to B  
0365 23 INC HL ; next score byte  
0366 7E LD A,(HL) ; load A with byte of player's score  
0367 E6F0 AND #F0 ; mask bits (was 0F)  
0369 B0 OR B ; mix together the 2 score bytes  
036A 0F RRCA  
036B 0F RRCA  
036C 0F RRCA  
036D 0F RRCA ; rotate right 4 times, this swaps the high and low bytes  
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
```

Dat werkt goed. Je krijgt een bonus leven bij 7000.

Aanpassen van de high score table met de default scores bij opstarten van game:

```
3565: 94 77 01 23 24 10 10 00 00 00 07 06 05 00 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 65 07 00 F4 76
3587: 96 77 02 1E 14 10 10 00 00 00 06 01 00 00 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 10 06 00 F6 76
35A9: 98 77 03 22 14 10 10 00 00 00 05 09 05 00 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 95 05 00 F8 76
35CB: 9A 77 04 24 18 10 10 00 00 00 05 00 05 00 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 05 05 00 FA 76
35EE: 9C 77 05 24 18 10 10 00 00 00 04 03 00 00 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 30 04 00 FC 76
```

Score werd zo vastgelegd in drie posities:

#60B2		#60B3		#60B4	
10	1	1.000	100	100.000	10.000

Nu is dit echter 'verschoven':

#60B2		#60B3		#60B4	
100	10	10.000	1.000	1.000.000	100.000

Dus de score in het eerste groene gedeelte is de score zoals deze op het scherm weergegeven wordt. De score in het tweede gedeelte voldoet aan de verschoven score in drie posities. Goed om te weten als later eventuele top 3 van het toernooi aan de default table toegevoegd gaan worden.

De koppen boven de high score table moeten ook aangepast worden:

```
#379E: 92 77 22 11 1E 1B 10 10 10 23 13 1F 22 15 10 10 1E 11 1D 15 10 10 10 3F
```

De startwaarden high score en player 1 score default kloppen nog niet. Staan op: #01BA.

Deze aanpassen naar:

```
01BA: 70 03 00 AA AA AA 65 07 00.
```

Als je hier bij beiden 12 34 56 invult. Dan wordt er op het scherm 5634120 getoond. Dus de volgende positieverdeling:

1e – hondertallen
2e – tientallen
3e – 10 duizendtallen
4e – duizendtallen
5e – miljoentallen
6e – honderdduizendtallen

Bij two-player game heeft player 2 nu slechts 6 digits score. De extra nul wordt niet getekend.

Aanpassen:

```
1AAC: DD 7D LD A,IXL ; load IX high byte into A – label AAAA
1AAE: FE 7F CP #7F ; check if byte contains 7F
1AB0: CAB81A JP Z,#1AB8 ; skip forward – jump to BBBB
;
1AB3: 3E 00 LD A,#00 ; A := 00
```

```

1AB5: DD7700 LD (IX+#00),A ; store A on screen
;
1AB8: C9 RET ; return - label BBBB

```

Dat gaat nu wel goed.

Nu versie v1.031 opgeslagen.

Het probleem is alleen dat de high score knippert, maar dat de extra nul niet mee knippert. En na het invoeren van de naam stopt het knipperen en is de extra nul veranderd in een spatie.

Stuk code aanpassen:

```

1AAC: FE0A CP #0A ; A = 0A ? - label AAAA
1AAE: 2004 JR NZ, #1AB4 ; no jump to CCCC
1AB0: 3E10 LD A,#10 ; load A with empty character
1AB2: 1808 JR #1ABC ; skip next - jump to DDDD
;
1AB4: DD7D LD A,IXL ; load IX low byte into A - label CCCC
1AB6: FE7F CP #7F ; check if byte contains 7F
1AB8: 2805 JR Z,#1ABF ; skip forward - jump to BBBB
;
1ABA: 3E 00 LD A,#00 ; A := 00
1ABC: DD7700 LD (IX+#00),A ; store A on screen - label DDDD
;
1ABF: C9 RET ; return - label BBBB

```

Hiermee knippert de extra nul wel mee.

Nu alleen nog wordt de additionele nul na invoeren van de initialen nog gewist.

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	
6100:	94	77	01	23	24	10	10	00	00	07	06	05	00	10	10	10	10	10	10	10	10	10	10	10	10	10	10	3F	00	50	76	00	F4	76
6122:	96	77	02	1E	14	10	10	00	00	06	01	00	00	10	10	10	10	10	10	10	10	10	10	10	10	10	10	3F	00	00	61	00	F6	76
6144:	98	77	03	22	14	10	10	00	00	05	09	05	00	10	10	10	10	10	10	10	10	10	10	10	10	10	10	3F	00	50	59	00	F8	76
6166:	9A	77	04	24	18	10	10	00	00	05	00	05	00	10	10	10	10	10	10	10	10	10	10	10	10	10	10	3F	00	50	50	00	FA	76
6188:	9C	77	05	24	18	10	10	00	00	04	03	00	00	10	10	10	10	10	10	10	10	10	10	10	10	10	10	3F	00	00	43	00	FC	76
	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9
	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB

Het groene stuk wordt gevuld met de score. Daarna worden er #E (=14) spaties getekend in het grijze stuk. Hier komt de spatie dus vandaan. Dit aanpassen: Eerst een nul tekenen (waarde 0) en daarna #D (=13) spaties.

```

13EB: C3601E JP #1E60 ; jump to additional code - jump to label AAAA
13EE: 00 00 00 00 4x NOP

```

en

```

1E60: 3600 LD (HL),#00 ; store #00 (character '0') into memory at (HL) - label AAAA
1E62: 23 INC HL ; next HL
;
1E63: 060D LD B,#0D ; for B = 1 to #D
;
1E65: 3610 LD (HL),#10 ; store #10 (character ' ') into memory at (HL) - label BBBB
1E67: 23 INC HL ; next HL
1E68: 10FB DJNZ #1E65 ; next B - DJNZ naar label BBBB
;
1E6A: C3F213 JP #13F2 ; jump back

```

Gaat nu allemaal goed. Nog wel een keer goed testen met waardes boven de miljoen en met ingeven van langere namen.

Invincibility aanzetten om te testen.

Nu versie v1.032 opgeslagen.

Veranderen tijd bij How High Can You Get scherm zodat je meer tijd hebt om te lezen.

De timer wordt hier gezet:

```
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
```

In #0C8B wordt de timer op #A0 gezet. Langer maken door een grotere waarde in te vullen:

```
0C8B 36D0 LD (HL),#FF ; set timer for #FF units
```

Nog niet echt lang, maar wel ietsje langer.

Daar een ander muziekje laten horen.

De playlist is:

```
; Playlist table for $57d songs
; -----
; For instance, $516 will play patterns $14, $83, $fe, etc. until it hits $00.
; FE 85 FE 85 FE 00 plays three gorilla roars
510: 10 db $10 ; How high can you get?
511: 00 db $00
512: 13 db $13 ; Rescued Pauline (even level)
513: 00 db $00
514: 11 db $11 ; Rescued Pauline (odd level)
515: 00 db $00
516: 14 db $14 ; Completed non-rivet stage
```

Lijn 510 wijst naar pattern #10 'How High Can You Get'. Dit laten verwijzen naar pattern #0C of #12.

Is een aanpassing in 's_3i_b.bin'.

#0C is net te lang. Stopt precies als jumpman getekend wordt in het nieuwe barrel board. Geen idee of je de snelheid van de pattern kunt aanpassen. Niet bekend.

Nee. Pattern #0C is ook nogal treurig en dat past niet. Daarom voor pattern #12 gaan.

Nu versie v1.033 opgeslagen.

De tekst DESIGN:PGO als naam bij de onderste default score ingeven.

14 15 23 19 17 1E 2E 20 17 1F

```
3565: 94 77 01 23 24 10 10 00 00 00 07 06 05 00 10 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 65 07 00 F4 76
3587: 96 77 02 1E 14 10 10 00 00 00 06 01 00 00 10 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 10 06 00 F6 76
35A9: 98 77 03 22 14 10 10 00 00 00 05 09 05 00 10 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 95 05 00 F8 76
35CB: 9A 77 04 24 18 10 10 00 00 00 05 00 05 00 10 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 05 05 00 FA 76
35ED: 9C 77 05 24 18 10 10 00 00 00 04 03 00 00 10 10 14 10 10 10 10 10 10 10 10 10 10 10 3F 00 30 04 00 FC 76
```

Nu versie v1.034 opgeslagen.

Nu testen met hoge begin score om de roll over naar miljoen te testen (zowel one player als two player) en de juiste invoeging in de high score tabel.

Het wissen van de score vindt hier plaats:

```
05B3 AF XOR A ; A := 0
05B4 77 LD (HL),A ; clear score
05B5 23 INC HL ; next score memory
05B6 77 LD (HL),A ; clear score
05B7 23 INC HL ; next score memory
05B8 77 LD (HL),A ; clear score
05B9 F1 POP AF ; restore AF
05BA C3C605 JP #05C6 ; jump ahead to task 2
```

Hier werken met een andere A:

```
05B3: C3C122 JP #22C1 ; jump to additional code – jump to label AAAA
05B6: 77 LD (HL),A ; clear score
```

en

```
22C1: 3E99 LD A,#09 ; A := 09 – label AAAA
22C3: 77 LD (HL),A ; clear score
22C4: 23 INC HL ; next score memory
22C5: C3B605 JP #05B6 ; jump back
```

Gaat goed er vind een overloop plaats naar de miljoen.

Dit na de test wel weer terugzetten: gedaan !

Dat gaat goed.

Herstellen door versie v1.034 terug te zetten.

De naam één positie naar rechts verschuiven en één positie minder kopiëren:

Aanpassen:

```
14CC 11F4FF LD DE,#FFF4 ; load DE with offset of -12 (decimal)
14CF 19 ADD HL,DE ; add offset
14D0 223A60 LD (#603A),HL ; store result into ???
```

en

```
15D3 060B LD B,#0B ; for B = 1 to #B (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

```

Dat is nu goed.

Invincibility weer uitgezet.

Nu versie v1.035 opgeslagen.

De miljoenen weergeven is wel leuk, maar dan moet nu de hele video van barrelpalooza opnieuw gemaakt worden. Heb daar eigenlijk niet zoveel zin in. En de miljoenen zou alleen zijn voor het geval dat in een tournooi de prijs vermelding in de default high score lijst zou zijn èn iemand een score boven de miljoen scoort. Als dat gebeurt kun je altijd nog de miljoen er weer inbouwen. Of dat niet als prijs hanteren (maar bijvoorbeeld een sticker set o.i.d.).

Dus terug naar versie 1.03, maar wel het alternatieve muziekje, de extra tijd in het HOW HIGH scherm en de design tekst toevoegen. En misschien het knipperen in het titelscherm weer terug?

Muziek HOW HIGH scherm aangepast:

Lijn 510 wijst naar pattern #10 'How High Can You Get'. Dit laten verwijzen naar pattern #0C of #12.

Is een aanpassing in 's_3i_b.bin'.

Tijd HOW HIGH scherm aangepast:

In #0C8B wordt de timer op #A0 gezet. Langer maken door een grotere waarde in te vullen:

```
0C8B 36FF LD (HL),#FF ; set timer for #FF units
```

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

DESIGN BY PAUL GOES

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

```

3565: 94 77 01 23 24 10 10 00 00 07 06 05 00 10 10 10 10 10 10 10 10 10 10 10 10 3E 00 50 76 00 F4 76
3587: 96 77 02 1E 14 10 10 00 00 06 01 00 00 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 00 61 00 F6 76
35A9: 98 77 03 22 14 10 10 00 00 05 09 05 00 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 50 59 00 F8 76
35CB: 9A 77 04 24 18 10 10 00 00 05 00 05 00 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 50 50 00 FA 76
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 v1.036 opgeslagen.

Kijken tuning van steering bonus. Wellicht al bij de tweede, maar dan maar 100 punten bonus geven?

Bonus vaststellen wordt gedaan op basis van de barrel colors: die zo oplopen bij succesvol steeren:

0B – 0D – 0F – 11 - 13

0C – 0E – 10 – 12 - 14

De code die het vaststelt is:

```
2615 DD7E08 LD A,(IX+#08) ; load new barrel color in A
2618 E60F AND #0F ; get low byte of barrel color
261A FE01 CP #01 ; color code >= 01?
261C DAB221 JP C,#21B2 ; no, jump back without sound
261F FE05 CP #05 ; color code < 05?
2621 D2B221 JP NC,#21B2 ; no, jump back without sound
;
2624 21AA39 LD HL,#39AA ; load HL with start of table data
2627 3E03 LD A,#03 ; load sound duration of 3
2629 328360 LD (#6083),A ; play sound for bouncer
262C C3B221 JP #21B2 ; jump back
```

Nu dus op basis van low byte tussen 1 en 5.

Als je bijvoorbeeld bij twee steers al wilt doen dan ook op basis van low byte. Maar dan bonus als low byte #0F of lager dan #05. Aanpassen:

```
261A FEOF CP #0F ; color code >= 0F?
261C CA2426 JP Z,#2624 ; no, jump back without sound
```

Ja dat lijkt goed te gaan. Gaat wel erg hard omhoog. Maar dan wellicht zo maken dat je 100 punten krijgt en niet de 300 punten.

Dat gebeurt hier:

```
262C 110300 LD DE,#0003 ; 300 points
262F 067D LD B,#7D ; sprite for 300
2631 CD9F30 CALL #309F ; yes, award points
```

Aanpassen naar:

```
262C 110100 LD DE,#0001 ; 100 points
262F 067B LD B,#7B ; sprite for 100
2631 CD9F30 CALL #309F ; yes, award points
```

Ja dat gaat goed. Dus veel vaker bonus, maar met minder hoog bedrag. En daardoor ook hoger in het veld nog te doen. Nog leuker is met 200 punten. Die krijg je normaal gesproken niet. Aanpassing dan:

```
262C 110200 LD DE,#0002 ; 200 points
262F 067C LD B,#7C ; sprite for 200
2631 CD9F30 CALL #309F ; yes, award points
```

Nu versie v1.037 opgeslagen.

En dan misschien ook nog de bug herstellen van de foutieve telling van de 800 punten?

Dat zit hier:

```
; award points for jumping a barrels and items
; arrive from #1DD7
; A is preloaded with 1,3, or 7
; patch ?

3E70 110100 LD DE,#0001 ; 100 points
3E73 067B LD B,#7B ; sprite for 100
3E75 1F RRA ; is the score set for 100 ?
3E76 D2281E JP NC,#1E28 ; yes, award points

3E79 1E03 LD E,#03 ; else set 300 points
3E7B 067D LD B,#7D ; sprite for 300
3E7D 1F RRA ; is the score set for 300 ?
3E7E D2281E JP NC,#1E28 ; yes, award points

3E81 1E05 LD E,#05 ; else set 500 points [bug, should be 800]
3E83 067F LD B,#7F ; sprite for 800
3E85 C3281E JP #1E28 ; award points
```

Aanpassen:

```
3E81 1E08 LD E,#08 ; else set 800 points
```

Nu versie v1.038 opgeslagen.

En de versie ophogen naar v1.04:

#373E: 3D 76 26 01 2B 00 04 10 3F (V1.03)

Deze versie is de final versie v1.04.

Eventueel kijken of we de board count in het high score overzicht kunnen stoppen.

Dit stuk code vult 14 spaties in het stuk waar de naam later ingevuld wordt.

```
; 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
```

Wat als we daar eens een ander teken invullen?

Even de invincibility aanzetten zodat we iets makkelijker kunnen testen.

Werkt niet helemaal zoals gedacht. De naam overschrijft een groot deel van de mintekens weer.

RANK	SCORE	NAME
1ST	011700--	
2ND	007650	
3RD	006100	
4TH	005950	
5TH	005050	
CREDIT 00		

Kijken waar dat overschreven wordt.

Dit stukje code:

```
15D3 060C LD B,#0C ; for B = 1 to #C (12 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
```

Hier worden twaalf waardes gekopieerd vanuit het videogeheugen (de plek waar de initialen vastgelegd zijn), naar de plek in de high score table die in #603A opgeslagen is. Alhoewel de naam maar 3 posities lang kan zijn, worden er 12 posities gekopieerd omdat de naam in de Japanse versie wel 12 posities lang kon zijn.

Als we de naam willen opschuiven, dan zou de offset van de naam verhoogd kunnen worden en zou het aantal te kopiëren posities verkleind kunnen worden.

De naam vijf posities naar rechts verschuiven en vijf posities minder kopiëren:

Aanpassen:

```
14CC 11F8FF LD DE,#FFF8 ; load DE with offset of -8 (decimal)
14CF 19 ADD HL,DE ; add offset
14D0 223A60 LD (#603A),HL ; store result into ???
```

en

```
15D3 0606 LD B,#06 ; 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
```

De tabel wordt dan:

R	A	N	K			S	C	O	R	E			N	A	M	E	
1	S	T				0	0	0	7	6	5	0			P	G	O

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.

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

#379D: 92 77 22 11 1E 1B 10 10 10 23 13 1F 22 15 10 10 12 22 14 10 10 1E 11 1D 15 3F

Ziet er goed uit. De REGI TIME gaat ook nog goed.



Nu nog ervoor zorgen dat de board count tussengevoegd wordt. Kan met een stukje code dat toegevoegd wordt aan het stuk dat ook de naam kopiëert.

Checken op two-player game. Zo ja, dan '---' invullen als board. En anders de board count kopiëren vergelijkbaar met het weergeven op het scherm.

Dit is de code om de progress counter weer te geven:

Stukje code voor het tekenen van progress inclusief de honderdtallen. De tientallen opslaan in #622E en de honderdtallen opslaan in #622A.

```

27C7: 3A2E62 LD A,(#622E) ; load A with progress number – label BBBB
27CA: 010AFF LD BC,#FF0A ; B: = #FF, C := #0A (10 decimal)
;
27CD: 04 INC B ; increment B – label CCCC
27CE: 91 SUB C ; subtract 10 decimal
27CF: D2CD27 JP NC,#27CD ; not carry, loop again – jump to label CCCC
;
27D2: 81 ADD A,C ; add 10 back to A to get a number from 0 to 9
27D3: 322675 LD (#7526),A ; draw progress to screen (singles)
27D6: 78 LD A,B ; load A with B (number of tens)
27D7: 324675 LD (#7546),A ; draw level to screen (tens)
27DA: 3A2A62 LD A,(#622A) ; load A with variable of progress number hundreds
27DD: 326675 LD (#7566),A ; draw level to screen (hundreds)

```

Dit moet dan nu opnieuw, maar dan niet naar het scherm wegschrijven, maar naar het datablok dat gebruikt wordt om de high score table aan te passen.

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 drie posities met de progress counter en dan nog negen spaties.

```

13EB 3610 LD (HL),#10 ; store #10 into memory at (HL)  

13ED 23 INC HL ; next HL  

13EE C32816 JP #1628 ; jump to additional code – jump to label AAAA  

13F1 00 NOP  

en  

1628 3610 LD (HL),#10 ; store #10 into memory at (HL) – label AAAA  

162A 23 INC HL ; next HL  

;  

162B 3A2E62 LD A,(#622E) ; load A with progress number  

162E 010AFF LD BC,#FF0A ; B: = #FF, C := #0A (10 decimal)  

;  

1631 04 INC B ; increment B – label CCCC  

1632 91 SUB C ; subtract 10 decimal  

1633 30FB JR NC,#1631 ; not carry, loop again – jump to label CCCC  

;  

1635 81 ADD A,C ; add 10 back to A to get a number from 0 to 9  

1636 77 LD (HL),A ; store progress (singles) into memory at (HL)  

1637 23 INC HL ; next HL

```

```

;
1638 78 LD A,B ; load A with B (number of tens)
1639 77 LD (HL),A ; store progress (tens) into memory at (HL)
163A 23 INC HL ; next HL
;
163B 3A2A62 LD A,(#622A) ; load A with variable of progress number hundreds
163E 77 LD (HL),A ; store progress (hundreds) into memory at (HL)
163F 23 INC HL ; next HL
;
1640 0609 LD B,#09 ; for B = 1 to #09 – label BBBB
;
1642 3610 LD (HL),#10 ; store #10 into memory at (HL)
1644 23 INC HL ; next HL
1645 10FB DJNZ #1640 ; next B – jump back to label BBBB
;
1647 C3F213 JP #13F2 ; jump back

```

Ruimte nodig van 34 bytes. Deze ruimte is er: vrije ruimte van #1628 t/m #1653. Waren oude testen Voor end of level afhandeling voor pies en rivets schermen. Die zijn verwijderd.

Gaat niet goed. Geeft bij game over de animatie van pauline met haar items. Hoe kan dat? Was toevallig dat dat veld gedaan werd op moment van game over. Geeft dus een level complete als resultaat. Dat is raar. Was een tikfout in de code.

Klopt niet helemaal: de progress counter staat verkeerd om: 500 ipv 005. Nog aanpassen !!! En dan meteen ook nog de test op 2-player game inbouwen.

```

1628 3610 LD (HL),#10 ; store #10 into memory at (HL) – label AAAA
162A 23 INC HL ; next HL
;
162B 3A0F60 LD A,(TwoPlayerGame) ; load A with # of players in game
162E A7 AND A ; is this a 1 player game?
162F C2AC1A JP NZ,#1AAC ; no skip next steps – jump to label DDDD
;
; Add the progress counter for 1 player game
;
1632 3A2A62 LD A,(#622A) ; load A with variable of progress number hundreds
1635 77 LD (HL),A ; store progress (hundreds) into memory at (HL)
1636 23 INC HL ; next HL
1637 23 INC HL ; next HL
;
1638 3A2E62 LD A,(#622E) ; load A with progress number
163B 010AFF LD BC,#FF0A ; B: = #FF, C := #0A (10 decimal)
;
163E 04 INC B ; increment B – label CCCC
163F 91 SUB C ; subtract 10 decimal
1640 30FC JR NC,#163E ; not carry, loop again – jump to label CCCC
;
1642 81 ADD A,C ; add 10 back to A to get a number from 0 to 9
1643 77 LD (HL),A ; store progress (singles) into memory at (HL)
1644 2B DEC HL ; previous HL
;
1645 78 LD A,B ; load A with B (number of tens)
1646 77 LD (HL),A ; store progress (tens) into memory at (HL)
1647 23 INC HL ; next HL
1648 23 INC HL ; next HL
1649 C3B41A JP #1AB4 ; skip next steps – jump to label BBBB

```

en

```
;  
; Add 3 dashes for 2 player game  
;  
1AAC 3E35      LD    A,#35      ; load A with dash character – label DDDD  
1AAE 77        LD    (HL),A    ; store dash into memory at (HL)  
1AAF 23        INC   HL       ; next HL  
1AB0 77        LD    (HL),A    ; store dash into memory at (HL)  
1AB1 23        INC   HL       ; next HL  
1AB2 77        LD    (HL),A    ; store dash into memory at (HL)  
1AB3 23        INC   HL       ; next HL  
;  
; Add other 9 spaces  
;  
1AB4 0609      LD    B,#09      ; for B = 1 to #09 – label BBBB  
;  
1AB6 3610      LD    (HL),#10    ; store #10 into memory at (HL)  
1AB8 23        INC   HL       ; next HL  
1AB9 10FB      DJNZ #1AB6    ; next B – jump back to label BBBB  
;  
1ABB C3F213    JP    #13F2    ; jump back
```

Leek goed te werken. Getest met 1-player game en dan brd nummer getoond. Daarna getest met 2-player game en dan worden de streepjes getoond. Daarna getest met 1-player game met 12 boards en dan geen mogelijkheid om naam in te geven en de originele tabel wordt weer getoond. Komt dat door het aantal boards of door switchen 1-player -> 2-player -> 1-player? Nog verder eventjes testen. En anders debuggen wat er gebeurd.

Ging fout doordat de JR in 1640 verkeerd was. Sprong één byte te ver terug (30FB). Is aangepast naar 30FC. Kijken of het nu dan wel goed gaat. Ja dat gaat goed.

Nu nog bij het initiele veld ook de barrel number bij de default scores invullen. Allemaal 001 (wat eigenlijk niet kan), of allemaal als mintekens? Maak er maar mintekens van. Dan nog duidelijker bij 1-player game dat dit de default scores waren.

```
3565: 94 77 01 23 24 10 10 00 00 07 06 05 00 10 10 35 35 35 10 10 10 10 10 10 10 10 3F 00 50 76 00 F4 76  
3587: 96 77 02 1E 14 10 10 00 00 06 01 00 00 10 10 35 35 35 10 10 10 10 10 10 10 10 3F 00 00 61 00 F6 76  
35A9: 98 77 03 22 14 10 10 00 00 05 09 05 00 10 10 35 35 35 10 10 10 10 10 10 10 10 3F 00 50 59 00 F8 76  
35CB: 9A 77 04 24 18 10 10 00 00 05 00 05 00 10 10 35 35 35 10 10 10 10 10 10 10 10 3F 00 50 50 00 FA 76  
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
```

Ja dat ziet er goed uit.

En de versie ophogen naar v1.05:

#373E: 3D 76 26 01 2B 00 **05** 10 3F (V1.05)

Nu nog eventjes rollover naar boardnummer boven de honderd testen en kijken of dit dan in de high score table goed weergegeven wordt.

Initialiseren van de progress counter:

```
3F4C: 3E00      LD    A,#00      ; A := 00
```

```
3F4E: 322A62 LD (#622A),A ; load progress number hundreds with 00
3F51: 3E00 LD A,#00 ; A := 00
3F53: 322E62 LD (#622E),A ; load progress number with 00
3F56: C9 RET ; skip ahead
```

Dus dan beginnen op board 98 is: #62 in de eenheden.

Ja dat gaat goed. Goede rollover en goede weergave in de high score table.

Ook nog testen met weergave van score van 523 boards. Starten met #05 in de honderdtallen en #13 in de eenheden.

De initialisatie van de progress number weer terugzetten naar #00 en #00.

Tevens de invincibility terugzetten.

Deze versie is de final versie v1.05.

Andrew Barrows gevraagd om te testen. Daar zijn drie dingen uitgekomen:

- 1) More Wilds stage: iets meer en wildere wild barrels mogelijk?
- 2) Spring Time stage: is het mogelijk om de spring te leechen?
- 3) Crazy Kong stage: is het mogelijk om de ladder cheat in te bouwen?

Eerst kijken of de wild barrels aangepast kunnen worden.

Ervoor zorgen dat altijd gestart wordt met de More Wilds stage.

De board override zit als het goed is hier:

```
27A4 3E01 LD A,#01 ; set default board 1 if still undecided
;
27A6 3E01 LD A,#01 ; board override: normally 3xNOP
;
27A8 322762 LD (#6227),A ; store into board nr
27AB C9 RET ; return
```

More Wilds board is board nr 4.

27A6 aanpassen naar 3E04 (en later weer terugzetten naar 0000).

Jumpman invincible maken.

Dat is nu goed. Het type wild barrel is afhankelijk van de difficulty setting. Dat zit hier:

```

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

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

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

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

```

Dus proberen allemaal gelijk te maken en dan kijken welke wat doet.

Allemaal naar #22F6: Of een bomb, of vrijwel recht naar beneden.

Allemaal naar #2303: Of een bomb, of grillig gericht op jumpman.

Allemaal naar #231A: Of een bomb, of met een boog gericht op jumpman.

De #2303 is de meest lastige om te ontwijken.

Is het mogelijk om dit random te maken. Dus niet afhankelijk van de invincibility, maar van de framecounter?

```

22D2 3A1A60 LD A,(Framecounter)
22D5 E603 AND #03 ; A between 0 and 3
22D7 EF RST #28 ; jump based on A
22D8 F6 22 ; #22F6 when A == 0
22DA 03 23 ; #2303 when A == 1
22DC 03 23 ; #2303 when A == 2
22DE 1A 23 ; #231A when A == 3
22EO 00 NOP

```

Dit geeft helemaal een grillig verloop: een enkele barrel gaat stukken recht, dan weer met boog en dan weer gericht op jumpman. Interessant.

Voorlopig eventjes zo houden. Later wel zo maken dat dit alleen zo werkt voor het More Wilds veld en dat de andere velden de originele wild barrels behouden.

Wat is de kans nu dat Kong een wild barrel gooit?

Dit stukje code regelt dat:

```

1A33 3A2762 LD A,(#6227) ; load A with screen number
1A36 FE04 CP #04 ; is this screen 4 ?
1A38 CA461A JP Z, #1A46 ; yes jump forward, jump to label AAAA

```

```

;
1A3B CD5700    CALL  #0057      ; load A with random number
1A3E E60F      AND   #0F       ; mask out left 4 bits to zero
1A40 C2862C    JP    NZ, #2C86  ; if result not zero, deploy normal barrel
1A43 C3492C    JP    2C49      ; else deploy crazy barrel
;
1A46 CD5700    CALL  #0057      ; load A with random number – label AAAA
1A49 E603      AND   #03       ; mask out left 6 bits to zero
1A4B C2862C    JP    NZ, #2C86  ; if result not zero, deploy normal barrel
1A4E C3492C    JP    2C49      ; else deploy crazy barrel

```

Dus nu in principe één op de vier. Voor mijn gevoel klopt dat niet helemaal. Wat als verder masken: 7 bits: dan AND #01 doen: dus 1A49 E601. Ja dan wel een boel meer.

Maar nog wel erg veel zijn bommem recht naar beneden. Waar komt dat vandaan? Kijken met de debugger. Barrel info zit op 6700 en +20, +40 enz. #6701 is crazy barrel indicator.

In de barrel release worden ook nog twee data blokken gebruikt. Waar zijn die voor?

```

; used in #2D8C for barrel release

39C3: 1E 4E BB 4C D8 4E 59 4E 7F

; table data having to do with crazy barrels.

; used in #2D83

39CC BB          ; for crazy barrels
39CD 4D          ;
39CE 7F          ; deployed when #7F

```

Dit zijn de data voor een barrel die recht naar beneden valt:

```

01 01 00 3B 00 59 08 17 0B 02 02 00 00 00 01 02
00 00 00 00 13 00 00 00 00 00 CE 00 00 00 00 00 00

01 01 00 3B 00 7A E0 16 0B 02 02 00 00 00 01 03
00 00 00 00 26 00 00 00 00 00 CE 00 00 00 00 00 00

01 01 00 3B 00 5F C0 16 0B 02 02 00 00 00 01 01
00 00 00 00 18 00 00 EE 00 CD 00 00 00 00 00 00 00

```

Dit zijn de data voor een barrel die netjes stuitert:

```

01 01 00 65 74 5E 20 16 0B 02 02 00 00 00 00 02
01 EE 00 00 16 00 00 00 00 00 4F 00 00 00 00 00 00

01 01 00 3F F5 B4 08 17 0B 02 02 00 00 00 00 00 02
FF F5 00 00 01 00 00 00 00 00 B4 00 00 00 00 00 00 00

01 01 00 6D 20 70 C8 16 0B 02 02 00 00 00 00 00 03
FF A0 00 00 05 00 00 00 00 00 70 00 00 00 00 00 00 00

```

Waar zitten de verschillen precies?

Nog niet echt duidelijk.

Misschien kijken hoe de wild barrel hack in elkaar zit en of dit daar ook optreed? Invincible maken en kijken wat daar gebeurd. Dan eventueel een diff van de hex om de aanpassingen te vinden.

Bij de wild barrel hack vallen er geen bommen rechtstreeks naar beneden. Verschillen:

Vanaf #0732 de volgende wijziging (extra code in een stuk wat normaal vrij is):

DD 7E 15 A7 CA D2 22 C3 E1 22

```
0732: DD 7E 15      LD      (IX+#15),A    ; store A in IX+#15
0735: A7            AND     A             ; A==0 ?
0736: CA D2 22      JP      Z,#22D2      ; yes, jump to 22D2
0739: C3 E1 22      JP      #22E1       ; no, jump to 22E1
```

; else this crazy barrel is no longer crazy

```
2120  DD7E07      LD      A,(IX+#07)      ; else Load A with +7 = sprite used
2123  E6FC        AND     #FC           ; clear right 2 bits
2125  F601        OR      #01           ; turn on bit 0
2127  DD7707      LD      (IX+#07),A    ; store result
212A  AF          XOR     A             ; A := 0
212B  DD7701      LD      (IX+#01),A    ; barrel is no longer crazy
212E  DD7702      LD      (IX+#02),A    ;
2131  DD3610FF    LD      (IX+#10),#FF    ; set velocity to -1 (move left)
2135  DD7711      LD      (IX+#11),A    ;
2138  DD7712      LD      (IX+#12),A    ;
213B  DD3613B0    LD      (IX+#13),#B0    ;
213F  DD360E01    LD      (IX+#0E),#01    ;
2143  C35321      JP      #2153        ; jump ahead
```

Op #2143 een aanpassing van een onconditionele jump. Springt nu naar #3FBA ipv #2153.

C3 BA 3F

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

Op #22CF een aanpassing van een conditionele jump. Springt nu naar #0732 ipv #22E1. Springt dus naar het extra stukje code.

CA 32 07

```

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

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

```

Op #22D7 een aanpassing van de difficulty afhankelijke jumps: allemaal hetzelfde gemaakt:

03 23 03 23 03 23 03 23 03 23

```

2C1B 3A8263 LD A,(#6382) ; else load A with crazy / blue barrel indicator
2C1E cb4f BIT 1,A ; test bit 1 - is this the second barrel after the first crazy ?
2C20 c2862C JP NZ,#2C86 ; if it is, then deploy normal barrel; this barrel is never crazy.

```

Op #2C20 een verwijdering van een conditionele jump. Gaat om de test van de tweede barrel die altijd een normale moet zijn (niet in de wild barrels hack)

00 00 00

```

2C41 CD5700 CALL #0057 ; else load A with a random number

;; hack to increase crazy barrels
;; 2C41 3E 00 LD A, #00
;; 2C43 00 NOP

```

Op #2C41 de aanpassing voor de increase van het aantal crazy barrels:

3E 00 00

```

; we arrive here if timer is within first 2 clicks when deploying a barrel from #2C18

2C7B C602 ADD A,#02 ; A := A + 2 (A had the initial clock value -2, now it has the initial clock value)
2C7D B9 CP C ; compare to current timer value - are we starting this round now?
2C7E CA492C JP Z,#2C49 ; yes, do a crazy barrel

2C81 3E02 LD A,#02 ; else A := 2 for the second barrel; it is always normal
2C83 C34B2C JP #2C4B ; jump back and continue deployment

```

Op #2C83 een aanpassing van een onconditionele jump. Gaat om de deployment van de tweede barrel die altijd een normale moet zijn (niet in de wild barrels hack). Springt nu naar #2C23 ipv #2C4B.

C3 23 2C

Vanaf #3FBA de volgende wijziging (extra code in een stuk wat normaal vrij is):

```
3FBA: DD 77 19 LD (IX+#19),A ; store A in IX+#19
3FBD: C3 53 21 JP #2153 ; jump back
```

Zou kunnen dat het stukje aanpassing bij de conditionele jump die checkt of de oil can al aangestoken is ervoor zorgt dat de barrels niet meer recht naar beneden vallen.

Aanpassen:

```
22CF CA3207 JP Z,#0732 ; jump to additional code, jump to label AAAA
```

en

```
0732 DD7E15 LD (IX+#15),A ; store A in IX+#15 – label AAAA
0735 A7 AND A ; A==0 ?
0736 CAD222 JP Z,#22D2 ; yes, jump to 22D2
0739 C3E122 JP #22E1 ; no, jump to 22E1
```

Nee dat is nog niet voldoende.

Dan de andere aanpassing van een jump bekijken:

Aanpassen:

```
2143 C3 BA 3F JP #3FBA ; jump to additional code, jump to label AAAA
```

en (dit blokje is in Barrelopalooza nog steeds vrij)

```
3FBA: DD 77 19 LD (IX+#19),A ; store A in IX+#19
3FBD: C3 53 21 JP #2153 ; jump back
```

Ja dat zorgt ervoor dat de bomb barrels niet meer plaatsvinden. Nog eventjes testen of dat ook niet meer gebeurd als de eerste aanpassing ongedaan gemaakt wordt.

```
22CF CAE122 JP Z,#22E1 ; no , jump ahead
```

Ja dat gaat nog steeds goed. Dus de oplossing is alleen die extra waarde die in IX+#19 gestopt wordt.

Dus nu zodanig maken:

- Grottere kans op wild barrels: aanpassing van de check waarden van #03 naar #02.
- Bij board 4 de weg van de wild barrel niet afhankelijk van difficulty maar van framecounter
- Bij board 4 de extra IX+#13 waarde setten.

Terug naar de vorige final versie.

Dan de boardoverride zodat met board 4 gestart wordt.

27A6 aanpassen naar 3E04 (en later weer terugzetten naar 0000).

En de invincibility aanzetten.

Aanpassen kans op een wild barrel:

1A49 E601 AND #01 ; mask out left 7 bits to zero

De code die het type wild barrel bepaalt op basis van de difficulty aanpassen zodat bij board 4 dit gebaseerd wordt op de framecounter.

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

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

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

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

Aanpassen:

22D2 C3EE27 JP #27EE ; jump to additional code; jump to label AAAA

en

```
27EE 3A2762 LD A,(#6227) ; load A with screen number – label AAAA
27F1 FE04 CP #04 ; is this screen 4 ?
27F3 2806 JR Z, #27FB ; yes, jump to label BBBB
;
27F5 3A8063 LD A,(#6380) ; no, load A with difficulty
27F8 C3D522 JP #22D5 ; jump back
;
27FB 3A1A60 LD A,(Framecounter) ; load A with framecounter – label BBBB
27FE E603 AND #03 ; A between 0 and 3
2800 3C INC A ; A=A+1; A between 1 and 4
2801 C3D622 JP #22D6 ; jump back
```

Gaat goed. Gooit nu de onvoorspelbaar bewegende barrels in het wild barrels screen, maar in ander scherm met level 5 (is max difficulty) wordt de barrel met een mooie boog gegooid.

Nu versie v1.051 opgeslagen.

Er voor zorgen dat bij board 4 de bomb barrels niet meer plaatsvinden:

Aanpassen:

2143 C3601E JP #1E60 ; jump to additional code, jump to label **AAAA**

en

1E60 3A2762 LD A,(#6227) ; load A with screen number – label **AAAA**

1E63 FE04 CP #04 ; is this screen 4 ?

1E65 2003 JR NZ,BBBB ; no, skip next step – jump to label **BBBB**

;

1E67: DD7719 LD (IX+#19),A ; store A in IX+#19

1E6A: C35321 JP #2153 ; jump back to 2153 – label **BBBB**

#1E60 t/m #1E6C is eerder vrijgemaakt. Dit stuk wordt overgeslagen door de JP #1E6D op #1E5D.

Ja gaat goed. In More Wilds veld geen bomb barrels meer die recht naar beneden gaan. In de andere velden nog wel.

Eventjes kijken of de grillige patronen nog grilliger worden wanneer andere random waarden gebruikt worden. Framecounter = 601A, RngTimer1 = 6018 en RngTimer2 = 6019.

27FB 3A1A60 LD A,(Framecounter) ; load A with framecounter – label **BBBB**

Geen idee of dit veel uitmaakt. Maar eventjes op RngTimer2 laten staan.

De invincibility uitzetten en kijken of dit niet te moeilijk is.

Dit is wel erg moeilijk. Misschien toch nog terugzetten naar minder vaak wilds.

Nu versie v1.052 opgeslagen.

Terugzetten naar minder vaak wilds.

1A49 E6**03** AND #03 ; mask out left **6** bits to zero

Dat zijn er wel weer erg weinig.

Maar moeilijk genoeg omdat ze zo onvoorspelbaar bewegen en dus lastig te ontwijken zijn.

Als de bovenstaande and op 00 dan alleen maar wild barrels. Dus dat klopt. Wel lastig te ontwijken omdat het er zo veel zijn. Maar wellicht minder barrels gooien mogelijk? Dus bijvoorbeeld om de barrel de deployment niet doen?

Waar zit het bepalen van de deployment van een barrel?

Misschien hier iets?

```
2D54 7E LD A,(HL) ; load A with value in HL. crazy barrel this value is #BB
2D55 DD2AAA62 LD IX,(#62AA) ; load IX with Barrel start address saved above
2D59 ED5BAC62 LD DE,(#62AC) ; load DE with sprite variable start EG #6980. set in #2CCC
2D5D FE7F CP #7F ; A == #7F ? (time to deploy out of kong's hands ?)
2D5F CA8C2D JP Z,#2D8C ; yes, jump ahead
```

Wat als die #7F veranderd wordt. Bv in FF? Nee dat gaat niet goed. Alles anders dan 7F resulteert erin dat de barrels niet goed gegooid worden.

En anders misschien hier?

```
2D15 21AF62 LD HL,#62AF ; load HL with deployment timer
2D18 35 DEC (HL) ; count it down. is the timer expired?
2D19 C0 RET NZ ; no, return

2D1A 3618 LD (HL),#18 ; else reset the counter back to #18
2D1C 3A8F63 LD A,(#638F) ; load A with the deployment indicator. 2 = kong grabbing, 1 = kong holding, 0 = deploying, 3 = kong empty
2D1F A7 AND A ; is a barrel being deployed right now?
2D20 CA512D JP Z,#2D51 ; yes, jump ahead
```

Wat als de #18 veranderd wordt. Bv in 30? Ja dan twee keer zo langzaam deploeyen van nieuwe barrels. Interessant. Daar eens mee testen.

Ja leuk. Alleen maar wild barrels en dan ontwijken. De conditie dat geen wild barrel wanneer de oil can nog niet aangestoken is, eruit halen.

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

Dus #22CF eventjes 3 NOP's maken. Nee dat is niet de goede. Nu de eerste blue barrel ook ongecontroleerd bewegen, maar de 2e en 3e barrel zijn nog gewoon.

Gaat om deze code:

```
2C1B 3A8263 LD A,(#6382) ; else load A with crazy / blue barrel indicator
2C1E cb4f BIT 1,A ; test bit 1 - is this the second barrel after the first crazy ?
2C20 c2862C JP NZ,#2C86 ; if it is, then deploy normal barrel; this barrel is never crazy.
```

Dus #2C20 eventjes 3 NOP's maken. Dat helpt iets: de 2e barrel is nog gewoon, de 3e barrel is crazy.

Dat van de tweede barrel zit hier:

```
; we arrive here if timer is within first 2 clicks when deploying a barrel from #2C18
```

```
2C7B C602 ADD A,#02 ; A := A + 2 (A had the initial clock value -2, now it has the initial clock value)
2C7D B9 CP C ; compare to current timer value - are we starting this round now?
2C7E CA492C JP Z,#2C49 ; yes, do a crazy barrel

2C81 3E02 LD A,#02 ; else A := 2 for the second barrel; it is always normal
2C83 C34B2C JP #2C4B ; jump back and continue deployment
```

Van de #2C81 eventjes een C3232C maken. Dan gaat dat goed.

Nu nog een goede snelheid tunen. Deze variant is wel erg lastig door het grillige bewegen van de barrels. Hier misschien weer terug naar de originele barrel bewegingen.

Testen door terug te gaan naar (#22D2 t/m 22E0):

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

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

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

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

Dan is de snelheid ineens wel weer erg langzaam.

Snelheid 28 is interessant. Best wel lastig nog. Maar wel steeds hetzelfde. Misschien nog iets sneller.

En dan kijken wat leuker is. De mix van normaal en wilds maar dan met de grillige banen. Of de wilds only maar dan met normale banen? Maar bijvoorbeeld level 1 geeft dan banen met weinig gevaar. Dan kun je gewoon naar boven lopen. Heel erg saai eigenlijk.

Daarom terug naar versie 1.052. En dan kijken of de kans op iets tussen 1-op-2 en 1-op-4 gezet kan worden. Bijvoorbeeld 3-op-8.

```
1A33 3A2762 LD A,(#6227) ; load A with screen number
1A36 FE04 CP #04 ; is this screen 4 ?
1A38 CA461A JP Z, #1A46 ; yes jump forward, jump to label AAAA
;
1A3B CD5700 CALL #0057 ; load A with random number
1A3E E60F AND #0F ; mask out left 4 bits to zero
1A40 C2862C JP NZ, #2C86 ; if result not zero, deploy normal barrel
1A43 C3492C JP 2C49 ; else deploy crazy barrel
```

```

;
1A46 CD5700    CALL  #0057      ; load A with random number – label AAAA
1A49 E603      AND   #03       ; mask out left 6 bits to zero
1A4B C2862C    JP    NZ, #2C86  ; if result not zero, deploy normal barrel
1A4E C3492C    JP    2C49      ; else deploy crazy barrel

```

Dat kan met een CP van een bepaalde waarde en wanneer carry dan springen voor een normale barrel. Dan kun je een verhouding kiezen x-op-256. Dus een normale barrel als kleiner dan de bepaalde waarde. Dus hoe kleiner de bepaalde waarde hoe minder normale barrels, hoe meer crazy barrels.

```

1A49 FE80      CP    #80       ; 50/50
1A4B DA862C    JP    C, #2C86  ; if result not zero, deploy normal barrel

```

#80 = 1-op-2

#C0 = 1-op-4

#AB = 1-op-3

Is het mogelijk om het type alleen aan het begin te bepalen? En dan tijdens de baan niet meer veranderen?

Mogelijk dit stuk code:

```

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

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

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

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

```

Die code is aangepast naar:

```

27EE 3A2762    LD    A,(#6227)    ; load A with screen number – label AAAA
27F1 FE04      CP    #04       ; is this screen 4 ?
27F3 2806    JR    Z, #27FB    ; yes, jump to label BBBB
;

```

```

27F5 3A8063      LD   A,(#6380)    ; no, load A with difficulty
27F8 C3D522      JP   #22D5       ; jump back
;
27FB 3A1960      LD   A,(RngTimer2) ; load A with rngtimer2 – label BBBB
27FE E603        AND  #03         ; A between 0 and 3
2800 3C          INC  A           ; A=A+1; A between 1 and 4
2801 C3D622      JP   #22D6       ; jump back

```

Dan dus bij 27FB naar stuk code springen die checkt of de Y-waarde groter is dan een bepaalde waarde en dan meteen een return doen.

```
27FB C3FE28      JP   #28FE       ; jump to additional code – jump to label AAAA
```

en

```

28FE DD7E05      LD   A,(IX+#05)  ; load A with barrel Y position – label AAAA
2901 FE60        CP   #60         ; is barrel lower than ?
2903 D0          RET  NC          ; yes, return
2904 3A1960      LD   A,(RNGTimer2) ; load A with rngtimer2
2907 C3FE27      JP   #27FE       ; jump back

```

Werkt wel. De barrels gaan nu in een rechte lijn. Maar dat is niet echt spannend.

Terug gaan naar versie 1.052.

En dan de kans aanpassing opnieuw doen.

```

1A49 FE80        CP   #80         ; 50/50
1A4B DA862C      JP   C, #2C86     ; if result not zero, deploy normal barrel

#80 = 1-op-2
#C0 = 1-op-4
#AB = 1-op-3

```

Kijken of we de baan normaal gesproken houden dat deze afhankelijk is van het difficulty level, maar zo af en toe, random er een wijziging op toepassen? Gaat om deze code:

```

27EE 3A2762      LD   A,(#6227)  ; load A with screen number – label AAAA
27F1 FE04        CP   #04         ; is this screen 4 ?
27F3 2806        JR   Z, #27FB    ; yes, jump to label BBBB
;
27F5 3A8063      LD   A,(#6380)  ; no, load A with difficulty
27F8 C3D522      JP   #22D5       ; jump back
;
27FB 3A1A60      LD   A,(Framecounter) ; load A with framecounter – label BBBB
27FE E603        AND  #03         ; A between 0 and 3
2800 3C          INC  A           ; A=A+1; A between 1 and 4

```

2801 C3D622 JP #22D6 ; jump back

Aanpassing:

27FB C3FE28 JP #28FE ; jump to additional code – jump to label AAAA

en

28FE CD5700 CALL #0057 ; load A with random number – label AAAA
2901 FE10 CP #10 ; is A < #10 ?
2903 D2F527 JP NC,27F5 ; no, jump back and work with difficulty
;
2906 3A1960 LD A,(RngTimer2) ; yes, load A with rngtimer2
2909 C3FE27 JP 27FE ; jump back and work with rngtimer

Dat werkt. Je kunt dus nu gaan tweaken met de kans op een crazy barrel (#1A49) en de kans op afwijkende baan (#2901):

1A49 FE80 CP #80 ; hoe hoger hoe kleiner de kans op crazy
2901 FE10 CP #10 ; hoe lager hoe kleiner de kans op afwijking.

Nu #AB en #80

Testen met #80 en #40: dat zijn er teveel

Testen met #AB en #40: lastig.

Testen met #C0 en #40: is te doen – nog verder mee testen.

Wellicht interessant om dit anders te maken. Barrels volgen in principe een gedefinieerde baan, maar één slot reserveren voor de grillige baan. Dan moet je dus testen op de IX.

Eerst nog even testen met 1-op-4, maar dan met andere bits in de AND.

1A46 CD5700 CALL #0057 ; load A with random number – label AAAA
1A49 E603 AND #03 ; mask out left 6 bits to zero
1A4B C2862C JP NZ, #2C86 ; if result not zero, deploy normal barrel
1A4E C3492C JP 2C49 ; else deploy crazy barrel

Misschien eens doen op de RngTimer1 of RngTimer2.

Wat goed gaat is het volgende:

1A46 3A1960 LD A,(RngTimer2) ; yes, load A with rngtimer2 – label AAAA
1A49 E611 AND #11 ; mask out left 6 bits to zero
1A4B C2862C JP NZ, #2C86 ; if result not zero, deploy normal barrel
1A4E C3492C JP 2C49 ; else deploy crazy barrel

Even testen met invincibility aan.

En dan de test wanneer grillig op basis van RngTimer aanpassen.

```
28FE CD5700      CALL   #0057      ; load A with random number – label AAAA
2901 FE10      CP     #10        ; is A < #10 ?
2903 D2F527      JP     NC,27F5    ; no, jump back and work with difficulty
;
2906 3A1960      LD     A,(RngTimer2) ; yes, load A with rngtimer2
2909 C3FE27      JP     27FE        ; jump back and work with rngtimer
```

Maar hoe test je op de IX? Wat is de waarde die nodig is? Is dat de I of de X en hoe laad je die in A. Eerst eens debuggen. Watchpoint op 28FE en dan kijken wat er in IX zit. IX is dan 67C0. Dus het interessante deel zit in X.

LD A,X is volgens de opcode reference opcode 3E.

Aanpassen naar:

```
28FE 3E          LD     A,X        ; load A with X to check barrel slot
28FF 0000        NOP,NOP
2901 FEC0        CP     #C0        ; is A = #C0 ?
2903 C2F527      JP     NZ,27F5    ; no, jump back and work with difficulty
;
2906 3A1960      LD     A,(RngTimer2) ; yes, load A with rngtimer2
2909 C3FE27      JP     27FE        ; jump back and work with rngtimer
```

```
28FE DD7D        LD     A,IXL      ; load A with X to check barrel slot
28FF 00          NOP
2901 FEC0        CP     #C0        ; is A = #C0 ?
2903 C2F527      JP     NZ,27F5    ; no, jump back and work with difficulty
```

Dit laatste werkt. Maar toch nog lastig wat te doen. Bij difficulty 2 en 3 gaat de barrel al zigzaggend.

Dus kiezen voor gewoon laten zoals het was, maar wel meer barrels. Of toch gaan voor een onvoorspelbaarder karakter: maar dan tuneable.

Terug gaan naar versie v1.052.

En dan weer de twee tunable randomizations inbouwen en gaan tunen.

De aanpassing voor het tunen van de kans op een wild barrel:

```
1A49 FE80        CP     #80        ; 50/50
1A4B DA862C      JP     C,#2C86    ; if result not zero, deploy normal barrel
```

De aanpassing voor het tunen van de kans op een afwijkende baan van de barrel:

```
27FB C3FE28      JP    #28FE      ; jump to additional code – jump to label AAAA
```

en

```
28FE CD5700      CALL   #0057      ; load A with random number – label AAAA
2901 FE10      CP    #10          ; is A < #10 ?
2903 D2F527      JP    NC,27F5      ; no, jump back and work with difficulty
;
2906 3A1960      LD    A,(RngTimer2) ; yes, load A with rngtimer2
2909 C3FE27      JP    27FE        ; jump back and work with rngtimer
```

Ja dat werkt nu beiden weer.

Tunen

#1A49: Kans barrel:

- als kleiner dan getal dan normale barrel
- hoe kleiner hoe meer crazy barrels

#2901: Kans afwijkende baan:

- als kleiner dan getal dan afwijkende barrel
- hoe groter hoe meer afwijking

Kans barrel: 1-op-3 = 85/256 dus zetten op 256-85=171 = #AB.

Kans op afwijking eventjes uitzetten = #00.

Is goed te doen. Maar blijft lastig met de barrels op difficulty 2 en 3. Die bewegen ook alle kanten op. Dat maakt ze ook zo lastig. Helemaal als er een treintje van barrels ontstaat.

Nu versie v1.053 opgeslagen.

Misschien zo te maken dat afwijkende baan gekoppeld aan een barrel slot. En dat wanneer normale baan, altijd op basis van de baan van difficulty 4 (de baan met een boog).

Dit is het stuk code voor barrel afhandeling voor screen 4 More Wilds:

```
28FE 3E          LD    A,X        ; load A with X to check barrel slot
28FF 0000        NOP,NOP
2901 FEC0        CP    #C0        ; is A = #C0 ?
2903 C2F527      JP    NZ,27F5      ; no, jump back and work with difficulty
;
2906 3A1960      LD    A,(RngTimer2) ; yes, load A with rngtimer2
2909 C3FE27      JP    27FE        ; jump back and work with rngtimer
```

Aanpassen naar:

```
28FE DD7D        LD    A,IXL      ; load A with X to check barrel slot
2900 FEC0        CP    #C0        ; is A = #C0 ?
2902 2006        JR    NZ,290A      ; no, skip next steps – jump to label AAAA
;
```

```

2904 3A1960      LD   A,(RngTimer2) ; yes, load A with rngtimer2
2907 C3FE27      JP   #27FE        ; jump back and work with rngtimer
;
290A 3E04        LD   A,#04        ; no, load A with difficulty 04 – label AAAA
290C C30128      JP   #2801        ; jump back and work with difficulty 04

```

Ja dat gaat goed. Op zich prima. Even testen met andere levels of het altijd de boogbarrels zijn.

Ja ook bij de andere levels zijn het boogballen met af en toe een wilde ertussen.

Kijken of die hele wilde barrel dan ook een andere kleur kan krijgen?

```

2904 3E01        LD   A,#01        ; load A with alternative color
2906 C3102B      JP   #2B10        ; jump to additional code – jump to label AAAA
2909 00          NOP

```

en

```

2B10 DD7708      LD   (IX+#08),A    ; store into barrel color
2B13 3A1960      LD   A,(RngTimer2) ; yes, load A with rngtimer2
2B16 C3FE27      JP   #27FE        ; jump back and work with rngtimer

```

Ja dat werkt. Alleen dan blijkt dat die speciale wel erg weinig komt. Wellicht kans groter wanneer slot aanpassen? Of meerdere slots?

Is dit stukje code:

```

28FE DD7D        LD   A,IXL       ; load A with X to check barrel slot
2900 FEC0        CP   #C0         ; is A = #C0 ?
2902 2006        JR   NZ,290A    ; no, skip next steps – jump to label AAAA

```

Ander slot bv #80 proberen. Dat helpt niet veel. Ongeveer één keer per veld.

Meer sloten testen:

```

28FE DD7D        LD   A,IXL       ; load A with X to check barrel slot
2900 FEC0        CP   #C0         ; is A >= #C0 ?
2902 3006        JR   NC,290A    ; no, skip next steps – jump to label AAAA

```

Ja, nu wel wat vaker. Misschien nog te vaak, maar eerst een goede kleur kiezen.

Kleur #02: rood met blauw.

Kleur #07: wit met bruin.

Kleur #0A: wit met roze.

Kleur #0F: zwart met rood.

Probleem is nog wel dat de eerste soms pas later zijn kleur krijgt. De kleuraanpassing op een andere manier doen?

De huidige versie behouden. De settings nu:

Een crazy barrel heeft een kans van 1-op-3.

Slots 00 en 01 geven een barrel met afwijkende baan: dus dat komt neer op 2-op-12 = 1-op-6.

En de barrel met afwijkende baan wordt rood/zwart gekleurd.

De board-override weer erafhalen: NOP maken:

```
27A6 3E01 LD A,#01 ; board override: normally 2xNOP
```

Nu versie v1.054 opgeslagen.

Nu testen of de spring geleeched kan worden.

Board override voor board 11 (#0B):

```
27A6 3E01 LD A,#01 ; board override: normally 2xNOP
```

En eventjes invincible maken zodat blijven testen.

Lijkt inderdaad niet mogelijk.

In de code gaan zoeken hoe dat kan.

De checks voor het jumpen over items (barrels en fireballs):

```
; checks for jumps over items on girders

3E99 E1 POP HL ; restore HL
3E9A AF XOR A ; A := 0
3E9B 326060 LD (NumObstaclesJumped),A ; clear counter for barrels jumped
3E9E 060A LD B,#0A ; For B = 1 to #A barrels
3EA0 112000 LD DE,#0020 ; load DE with offset
3EA3 DD210067 LD IX,#6700 ; load IX with start of barrel info table
3EA7 CDC33E CALL #3EC3 ; call sub below. check for barrels under jump

3EAA 0605 LD B,#05 ; for B = 1 to 5 fires
3EAC DD210064 LD IX,#6400 ; start of fires table
3EB0 CDC33E CALL #3EC3 ; check for fires being jumped

3EB3 3A6060 LD A,(NumObstaclesJumped) ; load A with counter for items jumped
3EB6 A7 AND A ; nothing jumped ?
3EB7 C8 RET Z ; yes, return

3EB8 FE01 CP #01 ; was 1 item jumped?
3EBA C8 RET Z ; yes, return; 1 is the code for 100 pts

3EBB FE03 CP #03 ; were less than 3 items jumped ?
3EBD 3E03 LD A,#03 ; A := 3 = code for 2 items, 300 pts score
3EBF D8 RET C ; yes, return

3EC0 3E07 LD A,#07 ; else A := 7 = code for 3+ items, awards 800 points
3EC2 C9 RET ; return
```

Er wordt inderdaad niet gecheckt op de springs. Misschien kan dat toegevoegd worden:

Aanpassen:

3EB3 C3E02F JP #2FE0 ; jump to additional code – jump to label AAAA

en

2FE0 0601 LD B,#01 ; for B = 1 to 1 springs – label AAAA
2FE2 DD210065 LD IX,#6500 ; start of springs tables
2FE6 CDC33E CALL 3EC3 ; check for springs being jumped
2FE9 3A6060 LD A,(NumObstaclesJumped) ; load A with counter
2FEC C3B63E JP #3EB6

Dit nog tussenvoegen ergens....

En dat werkt goed. De spring is nu jumpable en kan geleeched worden.

Nog wel verder testen.

Nu versie v1.055 opgeslagen.

De versie ophogen naar v1.06:

#373E: 3D 76 26 01 2B 00 06 10 3F (V1.06)

Deze versie is de final versie v1.06.

Als we in een tournament als prijs opname van de eerste drie scores in de default high score list willen bieden, dan moet eventjes getest worden hoeveel posities hiervoor beschikbaar zijn. Dan kan eventueel vrijheid genomen worden om alleen de initialen, of een langere naam te laten opnemen.

Eerder tekst toegevoegd in de default high score list:

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 01 23 24 10 10 00 00 07 06 05 00 10 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 50 76 00 F4 76
3587: 96 77 02 1E 14 10 10 00 00 06 01 00 00 10 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 00 61 00 F6 76
35A9: 98 77 03 22 14 10 10 00 00 05 09 05 00 10 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 50 59 00 F8 76
35CB: 9A 77 04 24 18 10 10 00 00 05 00 05 00 10 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 50 50 00 FA 76
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
```

Op een regel wordt na een score het volgende weergegeven:

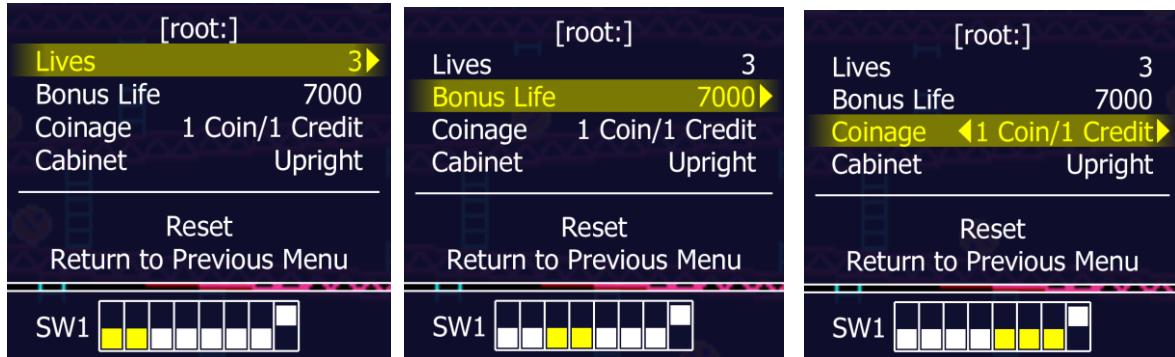
<><>BRD<><>NAME = 7 posities voordat de naam begint. Dat is bij de groene arcering bovenstaand.

Dat zijn dan maar 7 posities voor de naam. Eventjes testen door iets in te vullen. Ja dat klopt. Als je 7 posities invult, dan schuiven deze wel allemaal op als je een hogere score haalt. Dus dat gaat goed.

Dus 7 posities is de max. Dan voor de estethiek dan maar vasthouden aan de initialen die de spelers normaal hanteren. En dan op de website aangeven wie dat zijn (met eventueel link naar twitch).

Even weer teruggegaan naar final versie v1.06

Kijken of we de board override en level override kunnen koppelen aan de dipswitches. Eerst bepalen welke we daarvoor willen gebruiken:



Lives = twee bits Bonus life = twee bits Coinage = 3 bits

Board 1 t/m 12: 4 bits nodig – Combinatie van Lives en Bonus Life gebruiken

Level 1-5: 3 bits nodig – Coinage gebruiken

Het uitlezen en zetten van de dip switches gebeurd van #0207 t/m #0265. Daar wordt dan ook nog Cabinet/Upright gezet.

Deze code vervangen door statische code die de Lives (StartingLives), Bonus Life (ExtraLifeThreshold) en Coinage (CoinsPerCredit, CoinsPer2Credits, CoinsPerCredit2, CreditsPerCoin) op vaste waarden zet die overeenkomen met de defaults: allemaal switches in off positie (=0).

StartingLives = #6020

ExtraLifeThreshold = #6021

CoinPerCredit = #6022

CoinPer2Credits = #6023

CoinPerCredit2 = #6024

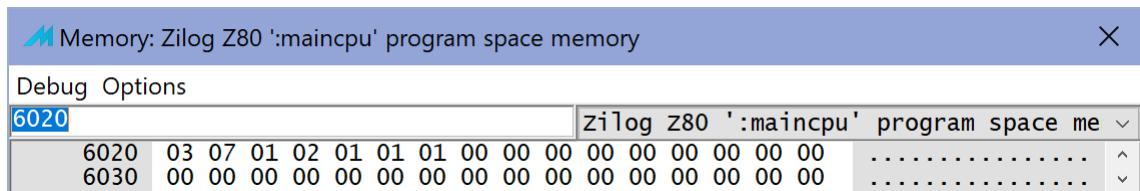
CreditsPerCoin = #6025

Dan kun je testen of dit overeenkomt met de defaults. Je krijgt daar al een stuk code vrij en dat kun je gebruiken om de BoardOverride en LevelOverride mee te configureren. Daar nog wel geheugenplaatsen voor reserveren:

```
BoardOverrideDip = #6396
LevelOverrideDip = #6397
```

En dan de code aanpassen voor de board en level override èn de code die zorgt voor het gelijkmatig verdelen van de random boards.

Met de default settings #6020 t/m 6026 gevuld met:



Aanpassen:

```
0207 212060 LD HL,StartingLives ; set HL to StartingLives
020A 3603 LD (HL),03 ; StartingLives = 03
;
020C 23 INC HL ; increment HL, set to ExtraLifeThreshold
020D 3607 LD (HL),07 ; ExtraLifeThreshold = 07
;
020F 23 INC HL ; increment HL, set to CoinsPerCredit
0210 3601 LD (HL),01 ; CoinsPerCredit = 01
;
0212 23 INC HL ; increment HL, set to CoinsPer2Credits
0213 3602 LD (HL),02 ; CoinsPer2Credits = 02
;
0215 23 INC HL ; increment HL, set to CoinsPerCredit2
0216 3601 LD (HL),01 ; CoinsPerCredit2 = 01
;
0218 23 INC HL ; increment HL, set to CreditsPerCoin
0219 3601 LD (HL),01 ; CreditsPerCoin = 01
;
021B 3A807D LD A,(DSW1) ; load A with dipswitch settings
021E 07 RLCA ; rotate left
021F 3E01 LD A,#01 ; A := 1
0221 DA5902 JP C,#0259 ; if carry, skip next step
;
0224 3D DEC A ; A := 0
;
0225 23 INC HL ; increment HL, set to UprightCab
0226 77 LD (HL),A ; store into UprightCab
```

En dan #0227 t/m #0259 allemaal NOPs.

Nu versie v1.061 opgeslagen.

Stuk code schrijven om de BoardOverrideDip te vullen. Deze wordt gevuld met de 4 bovenste bits uit de dipswitch waarde. Als we die allemaal instellen op 1 en de rest op 0 dan staat er in de dip switch waarde 0F (bekijken via debugging). Dus die waarde kan gewoon met AND gefilterd worden en dan de waarden boven twaalf terugzetten naar waarde 0.

```
0227 3A807D LD A,(DSW1) ; load A with Dip switch settings
```

```

022A E60F    AND    #0F          ; get relevant bits – is lower byte
022C FE0D    CP     #0D          ; is A less then 0D (decimal 13) ?
022E 3802    JR     C,#0232      ; yes, skip next step
0230 3E00    LD     A,#00          ; reset A to 00; non existent board nr
;
0232 329663   LD     (#6396),A      ; set BoardOverrideDip

```

Dit werkt op zich wel, maar wel twee problemen:

De volgende instelling geeft waarde 0A. Dus de bits zijn ook nog eens omgekeerd (van rechts naar links: dus zoals weergegeven: 1 – 2 – 4 – 8 : in dit geval dus $8+0+2+0=A$. Wel heel lastig configureren zo.



Daarnaast wordt de waarde van #6396 wel gezet, maar deze wordt vrij snel daarna weer overschreven met een nul. Dus we moeten zoeken naar twee geheugenlocaties die niet overschreven worden.

Eerste probleem op te lossen door F-A te doen: dat is dan weer 5. Nee, dat klopt toevallig voor deze, maar niet voor de andere.

Er is niet echt een procedure voor. Lomp zou kunnen door een bit te testen en als gezet dan het overeenkomstige bit zetten in een andere. Maar dat is nogal wat kostbare ruimte. Vooralsnog even zo laten. Komt toch een tabel met switch posities en settings.

Testen of de ongeldige boards ook goed gaan:

Board 11 = 1011 = inverse 1101 geeft BoardOverrideDip #0B – is goed.

Board 12 = 1100 = inverse 0011 geeft BoardOverrideDip #0C – is goed.

Board 13 = 1101 = inverse 1011 geeft BoardOverrideDip #00 – is goed: geen board 13 !

Stuk code schrijven om de LevelOverrideDip te vullen. Deze wordt gevuld met de 3 van de onderste bits uit de dipswitch waarde. Als we die allemaal instellen op 1 en de rest op 0 dan staat er in de dip switch waarde 70 (bekijken via debugging). Dus die waarde kan gewoon met AND gefilterd worden, swap low byte and high byte en dan de waarden boven vijf terugzetten naar waarde 0.

```

0235 3A807D LD A,(DSW1) ; load A with Dip switch settings
0238 E670 AND #70 ; get relevant bits
023A 0F RRCA ;
023B 0F RRCA ;
023C 0F RRCA ;
023D 0F RRCA ; rotate right 4 times: swaps low and high byte
;
023E FE06 CP #06 ; is A less than 06 (decimal 6) ?
0240 3802 JR C,#0246 ; yes, skip next step
0242 3E00 LD A,#00 ; reset A to 00; non existent board nr
;
0244 329763 LD (#6397),A ; set LevelOverrideDip

```

Testen of de ongeldige levels ook goed gaan:

Level 5 = 101 = inverse 101 geeft LevelOverrideDip #0B – is goed.

Level 6 = 110 = inverse 011 geeft LevelOverrideDip #00 – is goed.

Werkt ook. Alleen beide waarden worden nog overschreven. Even zoeken naar nieuwe memory locations die gebruikt kunnen worden en niet overschreven worden.

Proberen met de volgende locaties:

```

BoardOverrideDip = #6231
LevelOverrideDip = #6232

```

Aanpassen:

```
0232 323162 LD (#6231),A ; set BoardOverrideDip
```

en

```
0244 323262 LD (#6232),A ; set LevelOverrideDip
```

Getest met de debugger. Deze worden inderdaad niet overschreven. Dus hier voorlopig eventjes vanuit gaan.

Dan de code aanpassen dat deze waardes als override gebruikt worden. Eerst doen voor de level override omdat die makkelijker is (geen distributie zoals bij de boards).

De code die het level zet:

```
0C28 CD6727 CALL #2767 ; call routine to generate random level
```

en

```

2767 3A1960 LD A,(RngTimer2) ; load A with random timer value
276A E603 AND #03 ; mask bits, makes into #00-#03
276C 3C INC A ; increment, makes into #01-#04

```

```

276D 3C      INC   A          ; increment makes into #02-#05
276E FE02    CP    #02        ; A = 02?
2770 2006    JR    NZ,BBBB   ; no, skip next steps
;
2772 3A1860  LD    A,(RngTimer1) ; load A with random timer value
2775 E601     AND   #01        ; mask bits, makes into #00-#01
2777 3C      INC   A          ; increment, makes into #01-#02
;
2778 3E01     LD    A,#01      ; level override: normally 3xNOP
;
277A 322962  LD    (#6229),A   ; store into level nr
277D C9      RET

```

Aanpassen:

```
0C28 CD5302  CALL   #0253      ; call new code
```

en

```

0253 3A3262  LD    A,(#6232)   ; load A with LevelOverrideDip
0258 FE00    CP    #00        ; LevelOverrideDip = 0 ?
025A CA6727  JP    Z,#2767   ; yes, continue level is random
025D C37A27  JP    #277A     ; no, continue level is LevelOverrideDip

```

11 posities benodigd.

#025A t/m #0265 opschuiven. Dan komt er ruimte vrij van #0253 t/m #0265 (19 posities).

Dit gaat goed.

Nu versie v1.062 opgeslagen.

En dan nog de board override. De random board generatie zit op 277E.

```
0C2B CD7E27  CALL   #277E      ; call routine to generate random board
```

Aanpassen:

En dan ook het board random genereren:

```
0C2B CD440C  CALL   #0C44      ; call new code
```

en

```

0C44 3A3162  LD    A,(#6231)   ; load A with BoardOverrideDip
0C47 FE00    CP    #00        ; BoardOverrideDip = 0 ?
0C49 CA7E27  JP    Z,#277E   ; yes, continue board is random
0C4C C39A0B  JP    #0B9A     ; no, continue board is BoardOverrideDip

```

Die #0B9A is het einde van de code na gelijkmatiger verdelen van het board nr:

```
0B9A 322762 LD    (#6227), A      ; store into board nr  
0B9D C9      RET                 ; return
```

Binnen het How High Can You Get scherm is nog een heel stuk nop's: van #0C41 t/m #0C82.

Stuk vrijmaken door:

```
0C41 C3820C JP    #0C82          ; jump to remainder code How High screen
```

Daarmee komt dan #0C44 t/m #0C81 vrij.

En dat gaat goed.

Nu versie v1.063 opgeslagen.

Nog wel eventjes alle boards en levels controleren. En meteen opschrijven welke settings dat dan zijn.

Board nr	Board name	Dips	Settings
RND		0000 XXXX	Lives 3 Bonus Life 7000
01	Steering Bonus	1000 XXXX	Lives 4 Bonus Life 7000
02	Into The Dark	0100 XXXX	Lives 5 Bonus Life 7000
03	Crazy Kong	1100 XXXX	Lives 6 Bonus Life 7000
04	More Wilds	0010 XXXX	Lives 3 Bonus Life 10000
05	Pauline's Heart	1010 XXXX	Lives 4 Bonus Life 10000
06	Both Hammers	0110 XXXX	Lives 5 Bonus Life 10000
07	Reverse Run	1110 XXXX	Lives 6 Bonus Life 10000
08	Slow Barrels	0001 XXXX	Lives 3 Bonus Life 15000
09	Shifted Girders	1001 XXXX	Lives 4 Bonus Life 15000
10	Pauline's Items	0101 XXXX	Lives 5 Bonus Life 15000
11	Spring Time	1101 XXXX	Lives 6 Bonus Life 15000
12	Anti-Gravity	0011 XXXX	Lives 3 Bonus Life 20000

Level nr	Dips	Settings
RND	XXXX 000X	1 Coin/1 Credit
01	XXXX 100X	2 Coins/1 Credit
02	XXXX 010X	1 Coin/2 Credits
03	XXXX 110X	3 Coins/1 Credit
04	XXXX 001X	1 Coin/3 Credits
05	XXXX 101X	4 Coins/1 Credit

Al deze settings zijn gecheckt en kloppen.

Alleen alles getest met Cabinet Cocktail. Als daar gekozen wordt voor Cabinet Upright dan gaat het fout. Scherm wordt een rommeltje en blijft resetten. Zelfs als je niets doet gebeuren er rare dingen.

Hoe kan dat. Is het zetten van Upright/Cocktail niet goed dan?

Dit is niet het geval bij v1.061, maar wel bij v1.062 en v1.063. Rare is dat hij wel opstart, maar dat het fout gaat wanneer de eerste credit erin gestopt wordt. Eerst checken of daar dan de Startinglives, BonusLifeThreshold, CoinsPerCredit, CoinsPer2Credits, CoinsPerCredit2, CreditsPerCoin en UprightCocktail nog wel goed gezet worden. #6020 t/m #6026. Het lijkt erop dat deze wel goed gezet worden, en dat ze ook goed blijven. Dus daar ligt het niet aan.

Langzaam de code weer opbouwen vanaf v1.061 en dan steeds versie wegschrijven en testen of het nog werkt?

Terug naar versie v1.061.

Eerst de ruimte vrijmaken

#025A t/m #0265 opschuiven. Dan komt er ruimte vrij van #0253 t/m #0265 (19 posities). Dit gaat al meteen fout. Hoe kan dat? Alleen de code teruggeschoven door de NOP's ertussen te verwijderen. Maar dan gaat het fout. Dus wellicht wordt er naar deze code gesprongen? Kan in de disassembly geen code vinden die hiernaartoe springt. Als je het dubbel laat staan, dan gaat het wel goed. Geen flauw idee hoe dit kan. Maar dat betekent dat het vrijmaken van deze code op een andere manier moet. Maar wellicht wordt er gesprongen ergens tussen de NOP's. Daarom testen vrijmaken door op #0227 een JP naar #025A (C35A02) het tweede stuk waardoor de NOP's (#022A t/m #0259) ertussen vrijkomen. Lijkt goed te gaan. Eventjes testen met allemaal FF ertussen. Zelfs dat lijkt goed te gaan.

Dit herstellen in versie v1.063.

Terug naar versie v1.063.

De verplaatste code weer terug naar waar die stond:

Copy #0247 t/m #0252 naar #025A t/m #0265. En dan de ruimte daartussen (#0247 t/m #0259 komt vrij).

Dan op #0247 de jump opnemen:

0247 C35A02 JP #025A ; jump to continue setup code

Aanpassen:

0C28 CD4A02 CALL #024A ; call new code

en

024A	3A3262	LD	A,#6232)	; load A with LevelOverrideDip
024D	FE00	CP	#00	; LevelOverrideDip = 0 ?
024F	CA6727	JP	Z,#2767	; yes, continue level is random
0252	C37A27	JP	#277A	; no, continue level is LevelOverrideDip

Gaat ook niet goed. Zet nu helemaal niets.

Debugging geeft nu de werkelijke fout weer. De fout zit in het stuk dat de code voor de Cocktail/Upright bepaald. Deze is gekopieerd, maar bevat een keiharde verwijzing die niet klopt.

```
021B 3A807D LD A,(DSW1) ; load A with dipswitch settings
021E 07 RLCA ; rotate left
021F 3E01 LD A,#01 ; A := 1
0221 DA5902 JP C,#0259 ; if carry, skip next step
;
0224 3D DEC A ; A := 0
;
0225 23 INC HL ; increment HL, set to UprightCab
0226 77 LD (HL),A ; store into UprightCab
```

Moet alleen de DEC A overslaan, maar slaat zo veel meer over. Herstellen naar:

```
0221 DA2502 JP C,#0225 ; if carry, skip next step
```

Eventjes terug naar versie v1.063 en dit herstellen.

En dat gaat inderdaad nu goed. Nog wel eventjes wat settings steekproefsgewijs testen.

Nu versie v1.064 opgeslagen.

Ook nog eventjes checken of bij RND boards nog wel de gelijke verdeling binnen de 12 boards gedaan wordt.

Daartoe invincibility eventjes aanzetten.

Ja dat gaat ook allemaal goed.

Invincibility weer uitzetten.

Even alvast de versie aanpassen naar v1.07.

#373E: 3D 76 26 01 2B 00 07 10 3F (V1.07)

De default high score table aanpassen met de nummers 1 t/m 3 van het tournament. Eerder al vastgesteld dat we hiervoor de standaard 3 initialen aanhouden.

Eerder tekst toegevoegd in de default high score list:

DESIGN BY PAUL GOES

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

Ter referentie: dit staat er nu:

RANK	SCORE	BRD	NAME
1ST	007650	---	
2ND	006100	---	
3RD	005950	---	

3565: 94 77 01 23 24 10 10 00 00 07 06 05 00 10 10 35 35 35 10 10 10 10 10 3F 00 50 76 00 F4 76
3587: 96 77 02 1E 14 10 10 00 00 06 01 00 00 10 10 35 35 35 10 10 10 10 10 3F 00 00 61 00 F6 76
35A9: 98 77 03 22 14 10 10 00 00 05 09 05 00 10 10 35 35 35 10 10 10 10 10 3F 00 50 59 00 F8 76
35CB: 9A 77 04 24 18 10 10 00 00 05 00 05 00 10 10 35 35 35 10 10 10 10 10 3F 00 50 50 00 FA 76
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

Op een regel wordt het volgende weergegeven:

SCORE<><>BRD<><>INI

De uitslag van het toernooi:

- 1 - 789100 - 100 - JME - Rayfinkel – vastleggen op regel 3565 – 789100 – 100 – 1A-1D-15
- 2 - 400500 - 048 - CRT - CraigT – vastleggen op regel 3587 – 400500 – 048 – 13-22-24
- 3 - 351400 - 046 - BAR - Barra – vastleggen op regel 35A9 – 351400 – 046 – 12-11-22

3565: 94 77 01 23 24 10 10 07 08 09 01 00 00 10 10 01 00 00 10 10 1A 1D 15 10 10 10 10 3F 00 50 76 00 F4 76
3587: 96 77 02 1E 14 10 10 04 00 00 05 00 00 10 10 00 04 08 10 10 13 22 24 10 10 10 10 3F 00 00 61 00 F6 76
35A9: 98 77 03 22 14 10 10 03 05 01 04 00 00 10 10 00 04 06 10 10 12 11 22 10 10 10 10 3F 00 50 59 00 F8 76
35CB: 9A 77 04 24 18 10 10 00 00 05 00 05 00 10 10 35 35 35 10 10 10 10 10 3F 00 50 50 00 FA 76
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

Het ziet er nu zo uit:

RANK	SCORE	BRD	NAME
1ST	789100	100	JME
2ND	400500	048	CRT
3RD	351400	046	BAR

Maar dat zijn alleen nog maar de tekstuele wijzigingen. Nu de daadwerkelijke scores ook opnemen. En kijken of de high score dan ook goed gezet wordt.

Nu versie v1.065 opgeslagen.

Eerst even eventjes de high score van BAR opnemen na de 3F. Op plek 1, dan kijken of de high score aangepast wordt en wat deze dan aangeeft. 780091.

Nee dat past de high score niet aan. Eerst eens kijken hoe het zit met de scores in de high score table. Welk getal is welk deel van de score?

Aanpassen van de high score table met de default scores bij opstarten van game:

```
3565: 94 77 01 23 24 10 10 07 08 09 01 00 00 10 10 01 00 00 10 10 1A 1D 15 10 10 10 10 3F 00 50 76 00 F4 76
3587: 96 77 02 1E 14 10 10 04 00 00 05 00 00 10 10 00 04 08 10 10 13 22 24 10 10 10 10 3F 00 00 61 00 F6 76
35A9: 98 77 03 22 14 10 10 03 05 01 04 00 00 10 10 00 04 06 10 10 12 11 22 10 10 10 10 3F 00 50 59 00 F8 76
35CB: 9A 77 04 24 18 10 10 00 00 05 00 05 00 10 10 35 35 35 10 10 10 10 10 10 10 10 3F 00 50 50 00 FA 76
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
```

Score werd zo vastgelegd in drie posities:

#60B2	#60B3	#60B4
10	1	1.000

Dus:

1 - 789100 - 100 - JME - Rayfinkel – vastleggen op regel 3565 – 00 91 78 (was 50 76 00)

2 - 400500 - 048 - CRT - CraigT – vastleggen op regel 3587 – 00 05 40 (was 00 61 00)

3 - 351400 - 046 - BAR - Barra – vastleggen op regel 35A9 – 00 14 35 (was 50 59 00)

Wordt dan:

```
3565: 94 77 01 23 24 10 10 07 08 09 01 00 00 10 10 01 00 00 10 10 1A 1D 15 10 10 10 10 3F 00 00 91 78 F4 76
3587: 96 77 02 1E 14 10 10 04 00 00 05 00 00 10 10 00 04 08 10 10 13 22 24 10 10 10 10 3F 00 00 05 40 F6 76
35A9: 98 77 03 22 14 10 10 03 05 01 04 00 00 10 10 00 04 06 10 10 12 11 22 10 10 10 10 3F 00 00 14 35 F8 76
35CB: 9A 77 04 24 18 10 10 00 00 05 00 05 00 10 10 35 35 35 10 10 10 10 10 10 10 3F 00 50 50 00 FA 76
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
```

De startwaarden high score en player 1 score default kloppen nog niet. Staan op: #01BA.

01BA: 00 37 00 AA AA AA 50 76 00

Deze aanpassen naar:

01BA: 00 37 00 AA AA AA 00 91 78

Ja dat gaat goed nu. Zelfde opbouw als de scores in de high score table. Alle drie de waarden een keer geprobeerd (00 91 78, 00 05 40 en 00 14 35) en die kloppen dan allemaal in de high score table.

Nu versie v1.066 opgeslagen.

Nog wel eventjes testen met start waarde en score neerzetten net onder 3e plek, net boven 3e plek, net onder 2e plek, net boven 2e plek en net onder 1e plek en net boven 1e plek. Kijken of de high score juist gaat meelopen en of de score op de juiste plek tussen gevoegd wordt.

Nu testen met hoge begin score om de roll over naar miljoen te testen (zowel one player als two player) en de juiste invoeging in de high score tabel.

Het wissen van de score vindt hier plaats:

```
05B3 AF XOR A ; A := 0
05B4 77 LD (HL),A ; clear score
05B5 23 INC HL ; next score memory
05B6 77 LD (HL),A ; clear score
05B7 23 INC HL ; next score memory
05B8 77 LD (HL),A ; clear score
05B9 F1 POP AF ; restore AF
05BA C3C605 JP #05C6 ; jump ahead to task 2
```

Gaat om de scores

#60B2	#60B3	#60B4
10	1	1.000

De 10 en 1 (de eerste) gewoon laten wissen op basis van A en daarna de juiste waarden zetten:

Voor de derde score (351400): 00 14 35 – start score maken 340000 = 00 00 34

05B6: C34312 JP #1243 ; jump to additional code – jump to label AAAA

en

```
1243: 3600 LD (HL),#00 ; load HL address with 00 – label AAAA
1245: 23 INC HL ; next score memory
1246: 3634 LD (HL),#34 ; load HL address with 34
;
1248: C3B905 JP #05B9 ; jump back
```

Kan net op 1243 !

Getest met 350000 gespeeld: komt op plek 4!

Getest met 352000 gespeeld: komt op plek 3!

Voor de tweede score (400500): 00 05 40 – start score maken 390000 = 00 00 39

Getest met 400000 gespeeld: komt op plek 3!

Getest met 401000 gespeeld: komt op plek 2!

Voor de eerste score (789100): 00 91 78 – start score maken 780000 = 00 00 78

Getest met 789000 gespeeld: komt op plek 2!

Getest met 791000 gespeeld: komt op plek 1! En high score gaat meelopen.

[Terug naar versie v1.066.](#)

Dit is de final versie v1.07.