

Donkey Kong pac-man Mashup

Het idee is een romhack die een mashup is tussen Donkey Kong en Pac-man.

Met de volgende kenmerken:

- Op de girders liggen pellets die allemaal gegeten moeten worden
- Als alle pellets opgegeten dan veld laten knipperen zoals in Pac-man
- En dan verschijnt de ladder naar Pauline/Ms Pac-man
- De fireballs/firefoxes vervangen door ghosts met elke een eigen kleur?
- Gedrag van de ghosts aanpassen zodat ze meer op jumpman afgaan.
- Maximum aantal ghosts is vier.
- De hamers vervangen door power pellets: daardoor invincible
- Wanneer power pellet dan jumpman vervangen door Pac-man
- Wanneer power pallet dan Pauline vervangen door Ms pac-man
- Wanneer power pellet dan de ghosts blauw maken
- Wanneer een ghost 'gegeten' dan animatie met ogen voordat hij verdwijnt
- Wanneer andere enemies 'vernietigd' dan originele animatie
- Girders zijn blauwe lijnen zoals in Pac-man
- Trappen zijn groen
- De bonus items vervangen door fruit uit Pac-man. Wellicht meerdere random.
- Wellicht andere zaken uit Pac-man als versiering overnemen?
- Kill screen à la Pac-man
- Muziekjes van pac-man., geluidseffecten van pac-man?
- Bij beëindigen van board, de girders laten knipperen door palette wijzigingen?

Starten met het rivets scherm en met invincibility aan. En dan eerst eens kijken naar een aantal cosmetische zaken: de girders, ladders en kleuren aanpassen en fireballs vervangen door spookjes. Die ook nog de goede kant opkijken.

De girders aangepast naar lijnen en de fireballs naar spookjes. Mooie is dat de fireballs al van de normale sprite en de horizontaal gespiegelde sprite gebruik maken afhankelijk van welke kant ze op bewegen. Dus ze kijken altijd in de richting waarin ze bewegen. Bij het beklimmen van de trappen ook andere sprites gebruiken (omhoog en omlaag kijken). Maar dat kan later nog.

De fireballs bewegen echter ook op en neer. Dat is bij de pac-man spookjes normaal niet zo. Dat aanpassen in de code.

Dat zit mogelijk hier:

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

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

Nu versie v0.01 opgeslagen.

Kleuren aanpassen.

Eerst van de girders:

Blaauw van de randen: kleurcode pacman: 33/33/255	- C-2J/C-2K: 0F/08.
Rozegeel van de pellets: kleurcode pacman: 255/183/174	- C-2J/C-2K: 00/0A.
Groen voor de ladders erbij kiezen:	- C-2J/C-2K: 0F/03.

In de palette files zitten de rivets girder kleuren in de blokken: CC, D0, D4 en D8.

Code: C-2J: 0F 0F 00 0F

Code: C-2K: 0F 08 0A 03

Aanpassen. Dat ziet er goed uit.

De ladder aanpassen meer naar een stijl die past bij pacman. Ziet er ook goed uit.

Nu de kleurcodes van de fireballs aanpassen naar de kleuren voor het rode spookje.

Rood van het spookje: kleurcode pacman: 255/0/0	- C-2J/C-2K: 01/0B.
Wit van het oog: kleurcode pacman: 222/222/255	- C-2J/C-2K: 02/04.
Blauw van de pupil: kleurcode pacman: 33/33/255	- C-2J/C-2K: 0F/08

In de palette files zitten de fireball kleuren in de blokken: 04, 44, 84 en C4 (oog, spook, pupil)

Code: C-2J: 0F 02 01 0F

Code: C-2K: 0F 04 0B 08

Nu versie v0.01 opgeslagen.

Nu eerst de spookjes animaties aanpassen:

- [Twee versies van de spook sprite met ogen omhoog maken.](#)
- [Twee versies van de spook sprite met ogen omlaag maken.](#)
- [Als een spookje begint met trap omhoog, sprite aanpassen.](#)
- [Als een spookje begint met een trap omlaag, sprite aanpassen.](#)
- [Als een spookje in freeze stand dan ogen random laten wisselen.](#)
- [De spookjes in andere velden sprite waarden aanpassen.](#)

De sprites van de spookjes omhoog en spookjes omlaag zijn toegevoegd.

Omhoog zitten op #3D en #3E. Omlaag zitten op #5D en #5E.

De code waar bepaald wordt of een fireball een trap moet nemen of niet, zit hier:

```
336C DD360D04 LD      (IX+#0D),#04    ; Else set direction to descending ladder  
3370 C9          RET                 ; return
```

en hier:

```
339C DD360D08 LD      (IX+#0D),#08    ; Else set direction to ascending ladder  
33A0 C9          RET                 ; return
```

Op beide plekken een jump naar additionele code doen en dan daar de direction zetten en de sprite aanpassen naar de juiste sprite (omhoog dan wel naar beneden kijkend).

Even uitzoeken met de debugger in welke IX+## de spritewaarde zit. Zit in de IX+#07.

Aanpassen eerst voor trap afgaan:

```
336C C3383F        JP      #3F38       ; Jump to additional code – jump to label AAAA
```

En:

```
3F38 DD7E07        LD      A,(IX+#07)   ; load A with sprite value – label AAAA  
3F3B 0610          LD      B,#10        ; load B with #10  
3F3D 80            ADD     A,B          ; subtract #10 from value  
3F3E DD7707        LD      (IX+#07),A   ; store new sprite value  
3F41 DD360D04        LD      (IX+#0D),#04 ; set direction to descending ladder  
3F45 C9          RET
```

Dat gaat goed. Alleen nog ook aanpassen sprite value als de fireball de trap verlaat, want blijft nu naar beneden kijken.

Aanpassen voor de trap opgaan:

```
339C C3463F        JP      #3F46       ; Jump to additional code – jump to label AAAA
```

En:

```
3F46 DD7E07        LD      A,(IX+#07)   ; load A with sprite value – label AAAA  
3F49 0610          LD      B,#10        ; load B with #10  
3F4B 90            SUB     B             ; subtract #10 from value  
3F4C DD7707        LD      (IX+#07),A   ; store new sprite value  
3F4F DD360D08        LD      (IX+#0D),#08 ; set direction to ascending ladder  
3F53 C9          RET
```

Dat gaat goed. Alleen nog ook aanpassen sprite value als de fireball de trap verlaat, want blijft nu naar boven kijken.

Zoeken waar de trap verlaten wordt en dan weer terugzetten.

Lijkt hier te zitten:

```
; Fireball is moving left
33B5 DD7E07 LD A,(IX+#07) ; \ Set direction bit in fireball graphics to face left
33B8 E67F AND #7F ; |
33BA DD7707 LD (IX+#07),A ; /
33BD DD350E DEC (IX+#0E) ; Decrement X-position
```

Nee dat zet alleen de richting van de sprite (links of rechts kijken) door bit7 aan te passen.

Lijkt hier te zitten:

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

en hier:

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

Dit allebei aanpassen naar een call naar een extra routine en daar dan de fireball direction aanpassen (altijd naar links) en de sprite value naar de links kijkende.

Aanpassen:

```
337B CD543F CALL #3F54 ; call additional code – call label AAAA
3377E 00 NOP
```

en:

```
3394 CD543F CALL #3F54 ; call additional code – call label AAAA
3397 00 NOP
```

en:

```
3F54 DD360D00 LD (IX+#0D),#00 ; Set fireball direction to left – label AAAA
3F58 DD3607CD LD (IX+#07),#CD ; Set fireball sprite value facing to left
3F5C C9 RET ; return
```

Gaat allemaal goed. De spookjes kijken in de richting waarin ze bewegen.

[Nu versie v0.03 opgeslagen.](#)

Nu gaan kijken of het mogelijk is om boven de girders van dit type een pellet te tekenen.

Eerst een pellet toevoegen aan de sprites. Is toegevoegd op plek #DC.

De code die de griders tekent lijkt hier te zitten:

```
0F28 FE05 CP #05 ; first data item == 5 ?
0F2A CA510F JP Z,#0F51 ; yes, skip ahead to handle

; redraws screen when rivets has been completed

0F2D 3EFE LD A,#FE ; A := #FE

0F2F 32B563 LD (#63B5),A ; store into ???
0F32 2AAB63 LD HL,(#63AB) ; load HL with ???
|
0F35 3AB563 LD A,(#63B5) ; load A with ???
0F38 77 LD (HL),A ; store into ???
0F39 012000 LD BC,#0020 ; set offset to #20
0F3C 09 ADD HL,BC ; add offset for next
0F3D 3AB163 LD A,(#63B1) ; load A with ???
0F40 D608 SUB #08 ; subtract 8
0F42 32B163 LD (#63B1),A ; store result. done ?
0F45 D2350F JP NC,#0F35 ; no, loop again

0F48 13 INC DE ; else increase DE
0F49 C3A70D JP #0DA7 ; jump back

0F4C 3EE0 LD A,#E0 ; A := #E0
0F4E C32F0F JP #0F2F ; jump back

0F51 3EB0 LD A,#B0 ; A := #B0
0F53 C32F0F JP #0F2F ; jump back
```

De girder van rivets is de definitie met type 05, dus springt naar #0F51. Daar wordt het karakter gezet op #B0 (wat het karakter is van een rivets girder). En dan wordt gesprongen naar #0F2F. En dan lijkt het erop dat de locatie waar getekend dient te worden, geladen wordt vanuit #63AB. Je zou daar dan naar additionele code kunnen springen waarin ook het pellet karakter getekend wordt.

Aanpassen:

```
0F35 C35D3F JP #3F5D ; jump to additional code – jump to label AAAA
0F38 00 NOP
```

En:

```
3F5D 3AB563 LD A,(#63B5) ; load screen position to draw to – label AAAA
3F60 77 LD (HL),A ; draw character to screen
```

```

3F61 2B          DEC   HL      ; get screen position one higher
3F62 36DC        LD    (HL),#DC ; draw pellet
3F64 23          INC   HL      ; get original screen position
3F65 C3390F      JP    #0F39  ; jump back

```

Ja dat werkt goed. Ziet er zo uit:



Maar wel een probleem zo. Als een pellet voor een ladder zit, zie je een stuk van de ladder niet. Dan lijkt het een broken ladder. En als je zo'n pellet zou opeten, dan komt de ladder niet te voorschijn. Dus misschien handiger om de pellets apart te tekenen en een definitie aan te houden per veld. En dan dus beperkt houden waar de pellets liggen. Pakken van een pellet doen op basis van test van de screenpositie en dan of daar een pellet getekend staat. Kijken of je per pellet een score van 10 kan geven en de score daarmee uitbreiden? Maar geen voorwaarde dat alle pellets gepakt worden, maar wel een extra bonus geven en geluid wanneer wel alle pellets gepakt zijn (inclusief de power pellets).

Pellets tekenen op basis van definitietabel met groepen pellets: met per groep de startwaarde op het scherm en het aantal pellets te tekenen (naar links). Maar als we dat gaan doen, dan wel kijken in hoeverre we het rivets veld houden zoals het nu is, of er een variatie op maken. Voorkeur heeft het hanteren van variaties. Sowieso belangrijk voor het elevators veld. Dus dan bij allemaal doen. Hier gaan voor een 'zandloper' indeling. Eerst het veld aanpassen.

Eerst de bovenste girder net zo lang maken als de onderste en de grider daaronder zo lang maken als de tweede girder

```

3CD6: 05 EF 58 10 58 ; girder kong stands on
3CDB: 05 E7 80 18 80 ; level 4 girder
3CEO: 05 DF A8 20 A8 ; level 3 girder
3CE5: 05 E7 D0 18 D0 ; level 2 girder
3CEA: 05 EF F8 10 F8 ; bottom level girder
3CEF: AA              ; end code

```

En dan eventueel kijken of de level 4 girder één omhoog kan en de level 2 girder 1 omlaag kan.

```
3CD6: 05 EF 58 10 58 ; girder kong stands on  
3CDB: 05 E7 78 18 78 ; level 4 girder  
3CEO: 05 DF A8 20 A8 ; level 3 girder  
3CE5: 05 EO D8 18 D8 ; level 2 girder  
3CEA: 05 EF F8 10 F8 ; bottom level girder  
3CEF: AA ; end code
```

Ja dat kan en dat benadrukt de zandloper vorm een beetje. Maar moet wel veel aangepast worden en minder symmetrisch. Alle ladders moeten aangepast worden, de rivets moeten aangepast worden inclusief de afhandeling ervan en de startwaarden van de spookjes moeten aangepast worden.

Daarom zo laten. Alleen de twee buitenste bovenste ladders, verder naar buiten plaatsen. Dat helpt ook bij het versterken van de zandlopervorm.

Eerst terug naar versie v0.03 en de bovenste girder weer langer maken.

De buitenste ladders aanpassen:

```
3C95: 00 1B 58 1B 80 ; top left ladder  
3C9A: 00 53 58 53 80 ; top left ladder (right side)  
3C9F: 00 AB 58 AB 80 ; top right ladder (left side)  
3CA4: 00 E3 58 E3 80 ; top right ladder
```

Ja dat is nu goed.

Nu versie v0.04 opgeslagen.

Dan nu kijken voor een routine om de pellets te tekenen.

Is er een snelle manier om de screen positie te bepalen?

Het lijkt erop dat de game dit bepaalt met de code op #2FF0. Aan het einde daarvan bevat HL de screen location.

Deze code wordt op verschillende plekken aangeroepen. Maar op #2AA2 wordt dit voor jumpman gedaan om te checken wat er bij zijn voeten zit (dus onder hem?). Eerst maar eens kijken met de debugger met een watchpoint op 2AA5 en dan wat er dan teruggegeven wordt. Net links naast de rivet rechtsonder geeft #75BA. Een positie hoger is dan #75B9. Daarvandaan naar links zou je 7 pellets kunnen tekenen.

Eerst een routine maken voor het tekenen van de pellets. Je zou dat kunnen doen met een aparte definitie net als het tekenen van de girders. Zie het dan als een soort girder maar dan met pellets als karakter dat getekend wordt. Wat bijvoorbeeld zou kunnen is het ‘misbruiken’ van het tekenen van de kruisen. Maar hierdoor wordt de definitie van een veld wel langer. Daarom de rivets board definitie laten starten op de plek van het barrels board.

Rivets start op #3C8B en eindigt op #3CEF. Dit kopiëren naar #3AE4 en de rest allemaal wissen en later weer board voor board toevoegen.

En dan de verschillende verwijzingen naar #3C8B aanpassen naar #3AE4.

```
0CC3 11E43A LD DE,#3AE4 ; load DE with start of table data for rivets  
246E 21E43A LD HL,#3AE4 ; otherwise we're on rivets. load HL with table data for rivets
```

En dan de definitie gaan uitbreiden. Kruisen type code is #06. En dan het kruis aanpassen naar een pellet.

```
06 70 F0 20 F0 ; bottom girder left  
06 D8 F0 80 F0 ; bottom girder right  
06 40 C8 28 C8 ; girder 2 left  
06 D0 C8 B8 C8 ; girder 2 right
```

De twee middelste ladders op het tweede niveau één plek naar buiten schuiven.

```
3CB8: 00 5B A8 5B D0 ; level 2 ladder #2 of 4 → 00 53 A8 53 D0  
3CBD: 00 A3 A8 A3 D0 ; level 2 ladder #3 of 4 → 00 AB A8 AB D0
```

```
06 A0 C8 58 C8 ; girder 2 center  
06 70 A0 30 A0 ; girder 3 left  
06 C8 A0 80 A0 ; girder 3 right  
06 40 78 20 78 ; girder 4 left  
06 D8 78 B8 78 ; girder 4 right  
06 A0 78 58 78 ; girder 4 center  
06 40 50 10 50 ; top girder left  
06 E8 50 B8 50 ; top girder right
```

Ziet er nu zo uit:



Nu versie v0.05 opgeslagen.

Dan nu kijken afhandeling van het pakken van de pellets.

Code om de positie van jumpman om te zetten naar een screen positie begint op #2FF0. Bij de aanroep moet er in H de X-positie en in L de Y-positie van jumpman meegegeven worden. In het antwoord zit de HL die naar de corresponderende screen positie wijst.

Extra routine toevoegen aan de main waarin de X/Y positie van jumpman (lopend, niet springend of klimmend) omzet wordt naar de screen positie. Die kun je dan testen op de waarde van het pellet karakter. Als aanwezig dan verwijderen, een teller verhogen en punten toekennen. Als teller boven een bepaalde waarde dan alle pellets gepakt en dan geheugenplaats vullen. En dan twee varianten van de game: normal (veld af te ronden zonder alle pellets), hard (veld alleen af te ronden met alle pellets). Hierbij ook kijken naar de geluiden.

Toevoegen code voor oppakken pellets:

```
0A7B 3A1562      LD   A,(#6215)      ; load A with jumpman climbing status
0A7E FE00        CP   #00          ; is jumpman climbing a ladder?
0A80 C0          RET  NZ          ; yes, return
;
0A81 3A1662      LD   A,(#6216)      ; load A with jumpman jumping status
0A84 FE00        CP   #00          ; is jumpman jumping?
0A86 C0          RET  NZ          ; yes, return
;
0A87 3A0362      LD   A,(#6203)      ; load A with jumpman X-position
0A8A 67          LD   H,A          ; store in H
0A8B 3A0562      LD   A,(#6205)      ; load A with jumpman Y-position
0A8E 6F          LD   L,A          ; store in L
;
0A8F CDF02F      CALL #2FF0        ; call routine to get jumpman screen RAM
;
0A92 7E          LD   A,(HL)        ; load A with character at screen RAM
0A93 FEEF        CP   #FE          ; is it a pellet character?
0A95 C0          RET  NZ          ; no, return
;
0A96 3600        LD   (HL),#0A      ; remove pellet, overwrite with blank
0A98 C9          RET            ; return
```

En de extra call toevoegen aan de main:

```
19C2 CD7B0A      CALL #0A7B        ; call pellet pickup routine
```

Ja dat werkt helemaal goed.

Nu versie v0.06 opgeslagen.

Dan nu de score bijwerken met 10 punten voor elke score.

De player1 en player2 scores:

```
#60B2, #60B3, #60B4 - player 1 score
```

```
#60B5, #60B6, #60B7 - player 2 score
```

#60B2 en #60B5 zijn de 1 en 10.

#60B3 en #60B6 zijn de 100 en 1000.

#60B4 en #60B7 zijn de 10000 en 100000.

Met een beetje mazzel worden de 1 en 10 wel gewoon gebruikt in de score (alleen nooit gevuld). Dus $90+10=100$ (overflow), bij vaststellen high score en bij weergeven in high score list.

Maar hoe vullen. Er is een routine voor het toekennen van de 100, 300 en 500 bonussen. Hoe werkt dat?

Die zit op #1E15 en daar wordt #309F aangeroepen met waarde 01, 03 en 05 in DE. Dit betekent dat task 0 aangeroepen wordt en dat als parameter de score increment in honderdtallen meegegeven wordt. De routine werkt dus alleen met honderdtallen.

Dat betekent dat we zelf in de pellet afhandeling moeten tellen in #60B2 of 60B5. En dan wanneer de 10 bereikt wordt de #60B2 weer naar nul zetten en op dezelfde manier task 0 aanroepen met 1 als parameter.

Aanpassen:

0A98	3AB260	LD	A,(#60B2)	; load score tens
0A9B	C60A	ADD	A,#0A	; increment with 10
0A9D	27	DAA		; decimal adjust
0A9E	32B260	LD	(#60B2),A	; store score tens
;				
0AA1	C9	RET		; return

De #60B2 doorloopt wel de tientallen. Maar die worden niet weergegeven. En bij 100 nog geen overflow naar een score van 100.

0A98	3AB260	LD	A,(#60B2)	; load score tens
0A9B	C60A	ADD	A,#0A	; increment with 10
0A9D	27	DAA		; decimal adjust
0A9E	32B260	LD	(#60B2),A	; store score tens
;				
0AA1	3A0D60	LD	A,(PlayerTurnA)	; 0 for player 1, 1 for player 2

```

0AA4 11B460      LD   DE,#60B4          ; load DE with last digit score
0AA7 CD6B05      CALL #056B          ; update onscreen score
0AAA C9          RET             ; return

```

Gaat goed. Alleen score rollover moet nog ingebouwd worden.

```

0A98 3AB260      LD   A,(#60B2)        ; load score tens
0A9B C60A        ADD  A,#0A          ; increment with 10
;
0A9D 27          DAA             ; decimal adjust
0A9E 32B260      LD   (#60B2),A       ; store score tens
;
0A91 110000     LD   DE,#0000        ; do not update hundreds
0A94 FE00        CP   #00            ; A=00?
0A96 2003        JR   NZ,BBBB        ; no, no rollover, skip next step
0A99 110100     LD   DE,#0001        ; do update hundreds
;
0A9C CD9F30      CALL #309F          ; insert task to add score
;
0A9F C9          RET             ; return

```

Ja dat werkt goed. Zelfs high score gaat op goede moment meelopen en de score wordt op de juiste wijze in de high score tabel opgenomen. Alleen voor player 2 werkt het niet. Zou wel moeten kunnen. Iets aanpassen bij het laden en updaten van de tens.

Dit stuk code laad het juiste score adres voor zowel player1 als player 2:

```

055F 11B260      LD   DE,#60B2          ; load DE with player 1 score
0562 3A0D60      LD   A,(PlayerTurnA)    ; load number of players
0565 A7          AND  A               ; is this player 2 ?
0566 C8          RET  Z               ; no, return

0567 11B560      LD   DE,#60B5          ; else load DE with player 2 score
056A C9          RET             ; return

```

Hier gebruik van maken?

```

0A98 CD5F05      CALL #055F          ; load DE with start address score
0A9B 1A          LD   A,(DE)          ; load score tens
0A9C C60A        ADD  A,#0A          ; increment with 10
;
0A9E 27          DAA             ; decimal adjust
0A9F 12          LD   (DE),A         ; store score tens
;
0AA0 110000     LD   DE,#0000        ; do not update hundreds
0AA3 FE00        CP   #00            ; A=00?
0AA5 2003        JR   NZ,BBBB        ; no, no rollover, skip next step

```

```

0AA7 110100      LD   DE,#0001          ; do update hundreds
;
0AAA CD9F30      CALL  #309F          ; insert task to add score
;
0AAD C9          RET

```

Dat gaat goed. En grappig dat dit dus nog kortere code is ook !

Nu versie v0.07 opgeslagen.

Nu geluid lopen alleen als er een pellet gegeten wordt. Of in ieder geval drie NOP's voor een toekomstige call reserveren.

Nu kijken voor tellen van het aantal pellets. En als alle pellets van een board gegeten zijn een geluid geven en een bonus van 1000 toekennen. De pellet bonus sprite zit op #6D qua value.

Ongebruikte geheugenplaats zoeken die wel steeds op nul gezet wordt wanneer een board begint. Lijkt erop dat #64E0 niet gebruikt wordt. In debugger checken. Ja lijkt erop. Die nemen.

```

0AAD 3AE064      LD   A,(#64E0)      ; load pellet counter
0AB0 3C           INC  A             ; increment pellet counter
0AB1 32E064      LD   (#64E0),A    ; save pellet counter
0AB4 C9          RET

```

Wordt goed geteld en gereset bij starten van het board. Voor rivets veld zijn er #5C (92) pellets.

```

0AAD 000000      NOP,NOP,NOP      ; gereserveerd voor call naar routine voor geluid
0AB0 3AE064      LD   A,(#64E0)      ; load pellet counter
0AB3 3C           INC  A             ; increment pellet counter
0AB4 32E064      LD   (#64E0),A    ; save pellet counter
;
0AB7 47           LD   B,A          ; store A into B for comparison later
0AB8 3A2762      LD   A,(#6227)      ; load A with screen number
;
0ABB FE01         CP   #01           ; is this barrel board?
0ABD 2004         JR   NZ,#YYYY      ; no, skip next steps
0ABF 3E00         LD   A,#00           ; load A with nr of pellets on barrels
0AC1 1812         JR   AAAA          ; jump ahead – jump to label AAAA
;
0AC3 FE02         CP   #02           ; is this pies board?
0AC5 2004         JR   NZ,#YYYY      ; no, skip next steps
0AC7 3E00         LD   A,#00           ; load A with nr of pellets on pies
0AC9 180A         JR   AAAA          ; jump ahead – jump to label AAAA
;
0ACB FE03         CP   #03           ; is this elevators board?
0ACD 2004         JR   NZ,#YYYY      ; no, skip next steps
0ACF 3E00         LD   A,#00           ; load A with nr of pellets on elevators
0AD1 1802         JR   AAAA          ; jump ahead – jump to label AAAA
;

```

```

0AD3 3E5C      LD   A,#5C      ; load A with nr of pellets on rivets
;
0AD5 B8        CP   B          ; are all pellets picked up? – label AAAA
0AD6 C0        RET  NZ         ; no, return
;
0AD7 3E01      LD   A,#01      ; load a with #01
0AD9 32E164    LD   (#64E1),A ; set pellet bonus
;
0ADC 218560    LD   HL,#6085  ; else load HL with bonus sound address
0ADF 3603      LD   (HL),#03  ; play bonus sound for 3 duration
;
0AE1 110B00    LD   DE,#000B  ; 1000 points
0AE4 066B      LD   B,#6B    ; sprite for pellet bonus
0AE6 CD281E    CALL #1E28   ; yes, award points
;
0AE9 3E40      LD   A,#60    ; A := #60
0AEB 324163    LD   (#6341),A ; store into timer
0AEE 3E02      LD   A,#02    ; A := 2
0AF0 324063    LD   (#6340),A ; store into #6340
;
0AF3 C9        RET           ; return

```

Dat werkt goed

Nu versie v0.08 opgeslagen.

Maar de pellet bonus sprite wordt onder de girder weergegeven en is, wanneer de laatste pellet op de onderste girder lag, niet goed te zien.

In de code op #1E28 wordt een offset gedaan van #14 t.o.v. jumpman Y-position:

```

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

```

Daarom aanpassen en pas na de offset de routine aanroepen:

```

0AE6 CD9F30    CALL #309F      ; insert task to add score
0AE9 3A0562    LD   A,(#6205)  ; load A with Mario's Y position
;
0AEC CD301E    CALL #1E30      ; call subroutine to draw pellet bonus sprite
;
0AEF 3E60      LD   A,#60    ; A := #60
0AF1 324163    LD   (#6341),A ; store into timer
0AF4 3E02      LD   A,#02    ; A := 2

```

0AF6 324063 LD (#6340),A ; store into #6340

;

0AF9 C9 RET ; return

Kijken of coin insert sound voor elke pellet gespeeld kan worden.

0AFA 3E03 LD A,#03 ; load sound duration

0AFC 328360 LD (#6083),A ; plays the coin insert sound

0AFF C9 RET ; return

en:

0AAD CDFA0A CALL #0AFA ; call naar routine voor geluid

Gaat nu allemaal goed. Later nog het geluid meer een waka waka achtig maken als dat kan.

Nu versie v0.09 opgeslagen.

Nog doen:

- Titel weer terug naar High Score
- Titel en 1Up, 2Up en scores wit maken
- Blauw van girders iets paarser
- De girders smaller door onderste iets naar boven
- De trappen dezelfde kleur als de girders
- Rivets ook alleen in blauw
- Ghosts net zo snel naar links als naar rechts en naar boven als naar beneden
- Geen freezer ghosts meer
- Maximum van vier ghosts
- Altijd maximale difficulty en dus snelste ghosts

Eerst de girders dichter op elkaar maken door karakter aan te passen.

De tekst HIGH SCORE weer terug.

Vanaf #36B2 t/m #36BE: 80 76 18 19 17 18 10 23 13 1F 22 15 3F.

De bovenste drie regels altijd witte tekst = #07.

Pacman blauw = #0D/0C

Blauw van de randen: kleurcode pacman: 33/33/255 - C-2J/C-2K: 0F/08.

Rozegeel van de pellets: kleurcode pacman: 255/183/174 - C-2J/C-2K: 00/0A.

Groen voor de ladders erbij kiezen: - C-2J/C-2K: 0F/03.

In de palette files zitten de rivets girder kleuren in de blokken: CC, D0, D4 en D8.

Code: C-2J: 0F 0F 00 0F -> 0F 0D 00 0F

Code: C-2K: 0F 08 0A 03 -> 0F 0C 0A 03

Maximum van vier ghosts.

De trappen ook in blauw gemaakt.

De rivets alleen in blauw gemaakt. Soort van crossover tussen de twee girder lijnen. Ziet er minimalistisch wel heel goed uit.

Nu versie v0.10 opgeslagen.

Maximale aantal van vier ghosts:

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

Geen idee of dit voor alle boards werkt, maar moet in ieder geval aangepast worden. Werkt in ieder geval wel voor het rivets board. Er zijn nu maar vier ghosts. Helemaal top dus.

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

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

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

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

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

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

Ja dat klopt gaat drie keer zo snel omhoog nu. Is gewoon wennen.

De internal difficulty altijd op maximum zetten.

Dat wordt gedaan in de code van #037F t/m #03A1.

Aanpassen:

```

037F 3E05 LD A,#05 ; A := 5

0381 328063 LD (#6380),A ; store result into difficulty
0384 C9 RET ; return to #02DE

```

En de rest t/m #03A1 vrijgeven.

Ja dat gaat goed.

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

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

```

337B DD360D00 LD (IX+#0D),#00 ; Set fireball direction to left
3394 DD360D00 LD (IX+#0D),#00 ; Fireball has reached the bottom, set the direction to left

```

Deze zijn allebei al aangepast naar aanroep naar extra code om de sprite ook een andere kant op te laten kijken:

```

3F54 DD360D00 LD (IX+#0D),#00 ; Set fireball direction to left – label AAAA
3F58 DD3607CD LD (IX+#07),#CD ; Set fireball sprite value facing to left
3F5C C9 RET ; return

```

Aanpassen:

```

3F54 3A1860 LD A,(RngTimer1) ; load timer, a psuedo random number
3F57 C3000B JP #0B00 ; jump to additional code
3F5A 000000 3xNOP

```

en:

```

0B00 1F RRA ; roll right = 50% chance
0B01 3809 JR C,#AAAA ; jump to label AAAA to go right
;
0B03 DD360D00 LD (IX+#0D),#00 ; Set fireball direction to left
0B07 DD3607CD LD (IX+#07),#CD ; Set fireball sprite value facing to left
0B0B C9 RET ; return
;
0B0C DD360D01 LD (IX+#0D),#01 ; Set fireball direction to right – label AAAA
0B10 DD36074D LD (IX+#07),#4D ; Set fireball sprite value facing to right
0B14 C9 RET ; return
;

```

Vreemd. Nu soms helemaal links op de girder niet omdraaien, maar blijft naar links lopend hangen. Maar dat gebeurt niet rechts. Vreemd.

Even terug naar oorspronkelijke routine op #3F54. Toen was dat toch niet? Ja dat deed het toen ook al. Dus dat komt waarschijnlijk door iets anders (eerdere aanpassing). Hij gaat nu wel naar rechts bij verlaten van de ladder. Dus dat wel handhaven.

Soms ook bij verlaten ladder meteen weer de ladder nemen. Dat is ook wel raar. Toch nog eens goed kijken hoe of wat.

Teruggezet. En dan stapsgewijs testen.

- Maximaal vier ghosts: werkt nog.
- Difficulty standaard op 5: werkt nog.
- Sneller omhoog: werkt nog.

Links bewegen sneller maken:

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

Deze waren allemaal NOP gemaakt. Maar daarmee is de IX+#0D niet gezet en de A ook niet aangepast. Zou kunnen dat die later (na de JP Z) gebruikt worden en daardoor e.e.a. niet goed gaat. Beter is om alleen de #322D aan te passen naar NOP's. En dan eens kijken of dat helpt. Nee, gaat dan weer soms links 'hangen'.

Zou het misschien kunnen dat dat de freezers (nr 2 en 4) zijn die freezeen? Na een bepaalde tijd gebeurd het niet meer. Wellicht dan geen freezer functionaliteit meer? Even kijken welke het zijn. Wachten totdat het gebeurd, dan pauzeren en welke fireball heeft dan de laagste X-waarde (meest links)? Fireballs is #6400, 6420, 6440, 6460 met IX+#03 de X-positie. Resultaten: 4, 1, 4. En inderdaad op dat moment ook de freezer flag gezet IX+#19.

Is dit dan wellicht te voorkomen als de fireballs niet meer freezeen?

Freezer mode activeren wordt hier gedaan.

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

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

Deze routine wordt alleen vanaf #31B1 aangeroepen.

Testen geen freezers wanneer de call op #31B1 in 3 NOP's veranderd wordt. Ja dat gaat goed. Geen freezing meer en ook niet meer aan de linkerkant 'hangen'.

Ook nog eventjes weer terugbrengen dat de ghosts bij verlaten van een trap ok rechtsaf mogen gaan. Ja dat gaat ook goed.

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

Dat zit hier:

```
3360 DD701F LD    (IX+#1F),B      ; Store B into +#1F = Y-position of bottom of ladder
3363 3A0562 LD    A,(#6205)      ; \ Return without taking the ladder if Mario is at or above the Y-position of the fireball
3366 47      LD    B,A          ; |
3367 DD7E0F LD    A,(IX+#0F)      ; |
336A 90      SUB   B            ; |
336B D0      RET   NC          ; /
```

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

Het lijkt nu goed te gaan. Maar wel erg lastig door het snelle omhoog gaan van de spookjes. Maar daar later maar eens goed mee testen.

Nu versie v0.11 opgeslagen.

Nu gaan werken aan het vervangen van de hamers door power pellets.

De volgende zaken aanpassen:

- De sprites aanpassen en de plek ervan bepalen.
- Oppakken en 'hammer' mode activeren wanneer door pellet heen lopen, dus niet springen!
- Wanneer 'hammer' mode actief dan spookjes in bange blauwe spookjes veranderen
- De bange blauwe spookjes hebben geen richtingsanimatie: alleen de 'wave' aan de onderkant. Maar wel kijken hoe dat werkt met de generieke bewegingsanimatie.
- Wanneer 'hammer' mode actief dan jumpman in pacman veranderen (als laatste, lastig)
- Wanneer 'hammer' mode actief dan aan het einde kenbaar maken – hoe?
- De hammer-afhandeling verwijderen.
- Tijdens 'hammer' mode jumpman invincible maken.
- Als collision met enemy dan stopzetten spel en sequence met ogen en bonus toekennen

Sprite maken voor de powerpellet. De hamer daarvoor gebruikt. De kleur van de girders geven. Dat is bijvoorbeeld kleur #03. De kleur van de hamers zit in de definitie van de hamers.

Dat zit hier:

```

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

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

3E14: 1B 8C 7C 64 ; for rivets
3E18: 4B 0E 04 02 ; ???

```

Dus aanpassen voor de drie velden met hamers. Lijkt een enkele definitie te zijn. #3E09 aanpassen naar #03.

Dat is nu goed. Nu plaatsen op strategische plaatsen: op de twee girders met de middelste vrije delen. 1B 8C 7C 64 aanpassen naar 7C C6 7C 76. Dat is goed.

Nu versie v0.12 opgeslagen.

Het lastige is wel dat de ghosts achter de hamer sprites langs bewegen. Dat is tegennatuurlijk. Heeft te maken met de volgorde van sprite slots. Hoe kun je dat verbeteren?

Misschien makkelijker als tekenen op basis van achtergrondkarakter en het oppikken op dezelfde wijze als de gewone pellets. En dan mode zetten en op basis van de mode een aantal zaken regelen.

Misschien is dat wel gewoon het allerbeste.

Misschien terug naar versie v0.11 en dan als karakter beschouwen.

Maar eerst eventjes getest met de snelheid en beweging van de ghosts, maar dit is wel verrekte moeilijk. Voor het gevoel gaan de ghosts veel te snel op de trap.

Eerst eens de snelheid omhoog weer afremmen, maar dan met een timer vanaf 2 i.p.v. 3:

```

33F2 DD7E14 LD A,(IX+#14) ; \ Jump if it is not time to climb one pixel yet
33F5 A7 AND A ; |
33F6 C20134 JP NZ,#3401 ; /

33F9 DD361402 LD (IX+#14),#02 ; Reset ladder climb timer to 2
33FD DD350F DEC (IX+#0F) ; Decrement fireball's Y position, move up one pixel
3400 C9 RET ; return

```

Dit terugbrengen en op #33F9 een LD (IX+#14),01 doen. Ja dat gaat voor gevoel veel beter èn omlaag is snel genoeg, maar valt niet uit de toon.

Nu versie v0.13 opgeslagen.

Dus i.p.v. het tekenen van de hamers naar een stukje code die de twee power pellets tekenen. En dan dezelfde code voor het oppikken van de power pellet. En dan later de all pellets check aanpassen: aantal gewone pellets-2 en testen 2 power pellets.

Eerst alle hamerspecifieke code eruit halen en de ruimte vrijgeven?

#6217 is de geheugenplaats voor het vastleggen van de hammer active mode

Deze code is voor het plaatsen van de hamers op girders:

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

Deze code is voor het plaatsen van de hamers op conveyors:

```
1070 21103E LD HL,#3E10 ; load HL with table data for hammers on conveyors  
1073 CDA611 CALL #11A6 ; ???
```

Deze code is voor het plaatsen van de hamers op rivets:

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

Allemaal roepen deze de code op #11A6 aan. Daar worden de hamers getekend. Twee hamers van elk een X-positie en een Y-positie. Daar kun je twee screen posities plaatsen die je uitleest en op basis daarvan een karakter voor de power pellet tekent. De code zet ook twee geheugenplaatsen op #01 zodat de hamers actief zijn. Maar is dat benodigd? Maakt het uit welke hamer al gepakt is? Nee, want we wissen het power pellet karakter, dus kan niet nogmaals gepakt worden. En we zetten een counter en dan weten we of ze inderdaad allebei gepakt zijn.

Dus nu eerst de hamerdefinities voor rivets aanpassen. En dan een routine maken om de power pellets te tekenen.

De hamer definitie veranderen in de screen location voor de power pellets:

```
3E14: 1B 8C 7C 64 ; for rivets
```

```
3E14 0F 76 19 76
```

Midden in het scherm eerst uitmikken (ongeveer rond 75FF)

En dan een routine maken die dit uitleest en op die posities een karakter tekent voor de power pellet. De power pellet is karakter #DB.

```

11A6 3EDB        LD   A,#DB      ; A := #DB = character power pellet
;
11A8 5E          LD   E,(HL)    ; load E with LSB screen position
11A9 23          INC  HL       ; next entry
11AA 56          LD   D,(HL)    ; load D with MSB screen position
11AB 12          LD   (DE),A   ; draw power pellet on screen
;
11AC 23          INC  HL       ; next entry
;
11AD 5E          LD   E,(HL)    ; load E with LSB screen position
11AE 23          INC  HL       ; next entry
11AF 56          LD   D,(HL)    ; load D with MSB screen position
11B0 12          LD   (DE),A   ; draw power pellet on screen
;
11B1 C9          RET           ; return

```

De twee power pellets worden nu goed getekend.

Nu versie v0.14 opgeslagen.

Extra ruimte vrijmaken van de hamer teken routine.

Dan nu kijken voor copy van pellet eet routine en die dan omzetten naar power pellets.

```

11B2 3A1562        LD   A,(#6215)    ; load A with jumpman climbing status
11B5 FE00          CP   #00        ; is jumpman climbing a ladder?
11B7 C0            RET  NZ       ; yes, return
;
11B8 3A1662        LD   A,(#6216)    ; load A with jumpman jumping status
11BB FE00          CP   #00        ; is jumpman jumping?
11BD C0            RET  NZ       ; yes, return
;
11BE 3A0362        LD   A,(#6203)    ; load A with jumpman X-position
11C1 67            LD   H,A       ; store in H
11C2 3A0562        LD   A,(#6205)    ; load A with jumpman Y-position
11C5 6F            LD   L,A       ; store in L
;
11C6 CDF02F        CALL #2FF0     ; call routine to get jumpman screen RAM
;
11C9 7E            LD   A,(HL)    ; load A with character at screen RAM
11CA FEDB          CP   #DB       ; is it a power pellet character?
11CC C0            RET  NZ       ; no, return
;
11CD 360A          LD   (HL),#0A   ; remove pellet, overwrite with blank
;
11CF 3E00          LD   A,#00     ; load A with #00
;
```

```

11D1 32E364      LD   (#64E3),A          ; reset power pellet mode timer 1
11D4 32E464      LD   (#62E4),A          ; reset power pellet mode timer 2
;
11D7 3C          INC  A                 ; load A with #01
;
11D8 32E264      LD   (#64E2),A          ; set power pellet mode active
;
11DB C3980A      JP    #0A98            ; process score, pellet counter and bonus

```

#6394 en #6395 waren de oude hammer timers: 6394 loopt twee keer rond en dan klaar met timer.

De laatste JP springt naar de originele pellet routine waar de score bijgewerkt wordt en de pellet counter geupdate wordt en eventueel gecheckt wordt op de all pellets bonus. Dat ook doen voor de power pellets.

Dit moet aan de main toegevoegd worden. De vorige was op deze manier toegevoegd:

```
19C2 CD7B0A      CALL  #0A7B            ; call pellet pickup routine
```

Aanpassen:

```
19C2 CDE511      CALL  #11E5            ; call pellet pickup routines
```

en:

```

11E5 CD7B0A      CALL  #0A7B            ; call pellet pickup routine
11E8 CDB211      CALL  #11B2            ; call power pellet pickup routine
11EB C9          RET               ; return

```

Gaat op zich allemaal goed. Maar aan het begin wordt vanuit de #11D4 de waarde #E7 gezet voor de #64E4 (power pellet mode timer 2) en wanneer je dan doorstept wordt de waarde #E8 gezet voor de #64E2 (power pellet mode active). Vreemd want de #64E3 wordt niet gezet. En in A zit op dat moment #E7. Lijkt erop dat dus gesprongen wordt naar #11D4. Maar waar vandaan is dat dan, want het gebeurd maar één keer. En er wordt ook eenmalig een extra pellet geteld waardoor de pellet bonus eentje te vroeg weergegeven wordt.

Hier wordt nergens in de originele code naartoe gesprongen. Kijken met HXD of er in de hex iets zit met D411? Nee ook niet te vinden. Hoe kom je er dan achter waarvandaan er naar dit adres gesprongen wordt. Kan ook met een JR o.i.d. maar hoe vind je dat?

Met twee watchpoints op #11D1 en #11D4 wordt #11D4 wel getriggerd, maar #11D1 niet. Dus dan is de power pellet routine nog niet doorlopen. Want anders had de #11D1 ook moeten triggeren. Is dit ook nog voordat de gewone pellet routine loopt? #0A8F is instructie uit de gewone pellet routine. Ook daar een watchpoint op zetten. Bij de drie watchpoints actief wordt ook de gewone pellet routine niet doorlopen. Dus nog voordat de pellet routines gaan lopen. De power pellet routine zit na de code om de pellets te plaatsen. Loopt die dan door naar de power pellet routine?

Watchpoint op het tekenen van de pellets zetten. Dat begint bij #11A6.

Gevonden. Dit gaat niet goed. Het verwijderde stukje voor het tekenen van de hamers wordt namelijk ook gebruikt voor het tekenen van andere elementen zoals de onzichtbare sprites naast Kong, de bonus items, de pies en de elevators. Meerdere aanroepen naar #11D3.

Dus het stuk van #11D3 weer herstellen, zodat dat weer goed gaat. En dan het laatste stuk van de power pellet routine ergens anders plaatsen.

```
11CF 3E00 LD A,#00 ; load A with #00
;
11CF C3150B JR #0B15 ; jump to additional code – label AAAA
11D2 00 NOP
```

en:

```
0B15 3E00 LD A,#00 ; load A with #00
0B17 32E364 LD (#64E3),A ; reset power pellet mode timer 1
0B1A 32E464 LD (#62E4),A ; reset power pellet mode timer 2
;
0B1D 3C INC A ; load A with #01
;
0B1E 32E264 LD (#64E2),A ; set power pellet mode active
;
0B21 C3980A JP #0A98 ; process score, pellet counter and bonus
```

Gaat ook niet goed. Jumpman gaat dood. Lijkt nu te hangen in de routine op #11D3 en alle sprites (bonus items en jumpman) gaan naar rechts schuiven totdat jumpman dood gaat.

Opnieuw beginnen met versie v0.14 en dan opnieuw toevoegen in kleine stappen. De routine op #11D3 laten bestaan.

In originele versie v0.14 gaat het allemaal nog goed. De bonus items worden ook gewoon goed getekend.

Aanpassen:

```
19C2 CDB211 CALL #11B2 ; call pellet pickup routines
```

en:

```
11B2 CD7B0A CALL #0A7B ; call pellet pickup routine
11B5 CD150B CALL #0B15 ; call power pellet pickup routine
11B8 C9 RET ; return
```

En op #0B15 heel even een C9 RET om te testen.

Ja dat gaat goed.

```

0B15 3A1562      LD   A,(#6215)      ; load A with jumpman climbing status
0B18 FE00        CP   #00          ; is jumpman climbing a ladder?
0B1A C0          RET  NZ          ; yes, return
;
0B1B 3A1662      LD   A,(#6216)      ; load A with jumpman jumping status
0B1E FE00        CP   #00          ; is jumpman jumping?
0B20 C0          RET  NZ          ; yes, return
;
0B21 3A0362      LD   A,(#6203)      ; load A with jumpman X-position
0B24 67          LD   H,A          ; store in H
0B25 3A0562      LD   A,(#6205)      ; load A with jumpman Y-position
0B28 6F          LD   L,A          ; store in L
;
0B29 CDF02F      CALL #2FF0        ; call routine to get jumpman screen RAM
;
0B2C 7E          LD   A,(HL)        ; load A with character at screen RAM
0B2D FEDB        CP   #DB          ; is it a power pellet character?
0B2F C0          RET  NZ          ; no, return
;
0B30 360A        LD   (HL),#0A      ; remove pellet, overwrite with blank
;
0B32 3E00        LD   A,#00        ; load A with #00
;
0B34 32E364      LD   (#64E3),A    ; reset power pellet mode timer 1
0B37 32E464      LD   (#62E4),A    ; reset power pellet mode timer 2
;
0B3A 3C          INC  A           ; load A with #01
;
0B3B 32E264      LD   (#64E2),A    ; set power pellet mode active
;
0B3E C3980A      JP   #0A98        ; process score, pellet counter and bonus

```

Ja dat gaat nu wel goed.

Nu versie v0.15 opgeslagen.

Dan moet nu wanneer de power pellet mode actief is, de sprite en de kleur van het spookje aangepast worden. Er is een routine die de fireball definitie omzet naar sprites. Daar checken op mode en de sprite aanpassen?

Ja dat is de code van #34F3 t/m #351D.

Eerst eventjes de sprites van de spookjes tekenen. Daarvoor de kleuren van de background gebruiken: het blauw van de girders en het rozegeel van de pellets.

Eerst de sprite waarde eventueel aanpassen.

```

3514 7E      LD    A,(HL)      ; load A with sprite value
3515 12      LD    (DE),A      ; store sprite value to hardware
3516 13      INC   DE          ; next DE

```

Dus dit aanpassen naar een CALL naar additionele code

```
3514 CD410B      CALL #0B41      ; jump to additional code – jump to label AAAA
```

en:

```

0B41 7E      LD    A,(HL)      ; load A with sprite value – label AAAA
0B42 4F      LD    C,A        ; save A to C for later
;
0B43 3AE264    LD    A,(#64E2)    ; load A with pellet mode active indicator
0B46 FE01      CP    #01        ; is pellet mode active?
0B48 79      LD    A,C        ; restore A from C
0B49 2004    JR    NZ,#BBBB    ; no, set original sprite value – jump to label BBBB
;
0B4B E60F      AND   #0F        ; reset MSB part of sprite value
0B4D F660    OR     #60        ; set MSB part of sprite
;
0B4F 12      LD    (DE),A      ; store sprite value to hardware – label BBBB
0B50 13      INC   DE          ; next DE
0B51 C9      RET           ; return

```

Dit werkt in eerste instantie niet goed. De Y-waarde wordt aangepast. Dat is niet goed.

Het commentaar in de DKASM klopt niet. Sprite altijd Xpos -> value -> color -> Y-pos. Maar is anders gedocumenteerd in het commentaar. Dus dit moet op een andere plek tussengevoegd worden.

#3514 weer terugzetten naar de oorspronkelijke waarde.

Op de volgende plek tussenvoegen:

```

350A 7E      LD    A,(HL)      ; load A with firefox Y position
350B 12      LD    (DE),A      ; store into hardware sprite Y position
350C 2C      INC   L          ; next HL

```

Aanpassen:

```
350A CD410B      CALL #0B41      ; jump to additional code – jump to label AAAA
```

en:

```

0B41 7E      LD    A,(HL)      ; load A with sprite value – label AAAA
0B42 4F      LD    C,A        ; save A to C for later
;
0B43 3AE264    LD    A,(#64E2)    ; load A with pellet mode active indicator
0B46 FE01      CP    #01        ; is pellet mode active?

```

```

0B48 79          LD   A,C          ; restore A from C
0B49 2004        JR   NZ,#BBBB      ; no, set original sprite value – jump to label BBBB
;
0B4B E60F        AND  #0F          ; reset MSB part of sprite value
0B4D F660        OR   #60          ; set MSB part of sprite
;
0B4F 12          LD   (DE),A       ; store sprite value to hardware – label BBBB
0B50 2C          INC  L           ; next L
0B51 C9          RET

```

Ja dat gaat goed nu.

Nu versie v0.16 opgeslagen.

Eerst de hamerafhandeling verwijderen en vrijgeven.

In de main een aanroep naar #1E8C voor de hamer afhandeling (op #197D). Deze kan vrijgegeven worden en de code van #1E8C t/m #1F45 kan vrijgegeven worden. Ja spel blijft werken.

In de main een aanroep naar #2ED4 voor hamer grab afhandeling (op #1998). Deze kan vrijgegeven worden en de code van #2ED4 t/m #2FCA kan vrijgegeven worden. Ja spel blijft werken.

In de main een aanroep naar #281D voor hammer afhandeling (op #19B6). Deze kan vrijgegeven worden en de code van #281D t/m #2852 kan vrijgegeven worden. Ja spel blijft werken.

Nu versie v0.17 opgeslagen.

Dan een extra stuk routine maken in de main die de power pellet counters bijhoud en uiteindelijk ervoor zorgt dat de power pellet mode weer uitgezet wordt. Dat is nodig om de kleuren goed te krijgen. Aan het einde van de periode gaat het blauwe namelijk knipperen en dat ook namaken.

Het vrijgemaakte slot op #197D in de main gebruiken hiervoor.

Aanpassen:

```
197D CD520B      CALL  #0B52      ; call routine for power pellet mode timing
```

en:

```

0B52 21E264      LD   HL,#64E2      ; load HL with address pellet mode active indicator
0B55 7E          LD   A,(HL)       ; load A with pellet mode active indicator
0B56 FE01        CP   #01          ; is power pellet mode active?
0B58 C0          RET  NZ          ; no, return
;
0B59 23          INC  HL           ; load HL with address power pellet mode timer 1
0B5A 34          INC  (HL)        ; increase power pellet mode timer 1
;
0B5B 7E          LD   A,(HL)       ; load A with power pellet mode timer 1

```

0B5C	FEFF	CP	#FF	; is power pellet mode timer 1 at maximum?
0B5E	C0	RET	NZ	; no, return
;				
0B5F	23	INC	HL	; load HL with address power pellet mode timer 2
0B60	34	INC	(HL)	; increase power pellet mode timer 2
;				
0B61	7E	LD	A,(HL)	; load A with power pellet mode timer 2
0B62	FE02	CP	#02	; is power pellet mode timer 2 at maximum?
0B64	C0	RET	NZ	; no, return
;				
0B65	3E00	LD	A,#00	; load A with value 00
0B67	77	LD	(HL),A	; reset power pellet mode timer 2
0B68	2B	DEC	HL	
0B69	77	LD	(HL),A	; reset power pellet mode timer 1
0B6A	2B	DEC	HL	
0B6B	77	LD	(HL),A	; reset power pellet mode activity indicator
;				
0B6C	C9	RET		; return

Ja dat werkt goed. De mode wordt na verloop van tijd weer uitgezet en wordt eventueel verlengd wanneer tijdens de power pellet active mode ook de andere power pellet gepakt wordt.

Nu versie v0.18 opgeslagen.

Dan de kleurcode van de ghosts aanpassen. Afwijkende kleur toepassen wanneer power pellet mode is active AND power pellet timer 2 is 0 of power pellet timer 2 = 1 AND BIT3 van power pellet timer 1 is gezet. Effectief dan dus eerste helft power pellet mode de afwijkende kleur en inde tweede helft met een bepaalde frequentie knipperen.

Dit is het stukje code dat de sprite kleur bepaalt:

```
350D 1C      INC    E          ; next DE
350E 7E      LD     A,(HL)    ; load A with firefox sprite color value
350F 12      LD     (DE),A    ; store sprite color
```

Aanpassen:

```
350D CD6D0B    CALL   #0B6D    ; call additional code, call label AAAA
```

en:

```
0B6D 1C      INC    E          ; next DE – label AAAA
0B6E 7E      LD     A,(HL)    ; load A with original sprite color value
0B6F 4F      LD     C,A       ; store original sprite color in C
;
0B70 3AE264   LD     A,(#64E2)  ; load A with power pellet mode active indicator
0B73 FE01    CP     #01       ; is the power pellet mode active?
0B75 200E   JR     NZ,#BBBB  ; no, jump to label BBBB
```

```

;
0B77 3AE464 LD A,(#64E4) ; load A with power pellet mode timer 2
0B7A FE00 CP #00 ; is power pellet mode time 2 0?
0B7C 280A JR Z,#CCCC ; yes, first half, jump to label CCCC
;
0B7E 3AE364 LD A,(#64E3) ; load A with power pellet mode timer 1
0B81 CB6F BIT 5,A ; is bit 5 set?
0B83 2003 JR NZ,#CCCC ; yes, second half blink, jump to label CCCC
;
0B85 79 LD A,C ; restore original color into A – label BBBB
0B86 1802 JR #DDDD ; skip next step, jump to label DDDD
;
0B88 3E03 LD A,#03 ; set different color into A – label CCCC
;
0B8A 12 LD (DE),A ; store sprite color – label DDDD
0B8B C9 RET ; return

```

Ja dat werkt helemaal goed.

Nu versie v0.19 opgeslagen.

Kijken hoe jumpman tijdens de power pellet mode veranderd kan worden in pacman.

Dit zijn de jumpman sprites (waardes #00 t/m #0F):



Hoe wordt de happende pacman geanimeerd? Twee varianten: mond klein beetje open en mond wat verder open en dan in de vier richtingen. Links rechts variëren kan door bit7 te flippen.

Wellicht te doen door tijdens de power pellet mode de sprites van jumpman via een tabel te vertalen naar corresponderende pac-man sprites?

Maar kan dat net als bij de ghosts op een enkele plek? Waar wordt de sprite van jumpman gevuld? Jumpman sprite is #694C (Xpos), #694D (value), #694E (color) en #694F (Ypos). Ja dat zit hier:

```

1DA6 214C69 LD HL,#694C ; load HL with mario sprite X position
1DA9 3A0362 LD A,(#6203) ; load A with mario's X position
1DAC 77 LD (HL),A ; store into hardware sprite mario X position
1DAD 3A0762 LD A,(#6207) ; load A with movement indicator
1DB0 2C INC L ; HL := #694D = hardware mario sprite
1DB1 77 LD (HL),A ; store into hardware mario sprite value
1DB2 3A0862 LD A,(#6208) ; load A with mario color
1DB5 2C INC L ; HL := #694E = hardware mario sprite color
1DB6 77 LD (HL),A ; store into mario sprite color
1DB7 3A0562 LD A,(#6205) ; load A with mario Y position
1DBA 2C INC L ; HL := #694F = mario sprite Y position
1DBB 77 LD (HL),A ; store into mario sprite Y position
1DBC C9 RET ; return

```

Maar er zijn ook een paar additionele plekken waar de sprite value explicet aangepast wordt, dan dan wellicht ook aangepast moeten worden. Een paar daarvan zijn voor hammer afhandeling en die zijn er al niet meer, dus valt wel mee.

Eerst de pacman sprites tekenen.

Dit zijn de pacman sprites (waardes #54 t/m #59):



Misschien handiger om te bepalen wat de richting van jumpman is en dan op basis van de richting de sprite bepalen en variëren op basis van de framecounter?

Wat wordt er gebruikt voor de richting van jumpman? De sprite van jumpman wordt bijgehouden in #6207. Met bijvoorbeeld lopen naar links: #00, #01 en #02 en naar rechts #80, #81 en #83. Maar op de trap wordt de hele tijd gewisseld tussen links en rechts gespiegelde: #03, #83, #04, #84, #05, #85. En dan aan einde van ladder #06 (recht vooruit kijkend). Maar dit is hetzelfde voor ladder omhoog en ladder omlaag. Tijdens het springen naar links is het #0E en bij springen naar rechts #8E. Dus dat is voldoende. Er is niet echt een waarde waarin vastgelegd wordt of jumpman een ladder omhoog of omlaag gaat. Maar misschien af te leiden van de input en die dan registreren en wanneer geen input dan de geregistreerde waarde gebruiken.

Misschien zelfs handiger om sowieso van de InputState (#6010) gebruik te maken. Die geeft altijd de richting aan. Rechts = #01, links = #02, omhoog = #04 en omlaag = #08 en wanneer geen input dan #00. Routine maken die altijd aangeroepen wordt, de InputState uitleest en wanneer deze ongelijk aan #00 is wegschrijft. Als gelijk aan #00 dan inlezen weggeschreven waarde en dan de sprites afhankelijk van de waarde en variëren op basis van bepaalde BIT van de Framecounter (afhankelijk hoe snel knipperen).

Aanpassen:

1DAD CD8C0B CALL #0B8C ; call additional code – call label **AAAA**

en:

```
0B8C 3A1060            LD     A,(InputState) ; load A with input – label AAAA
0B8F FE00            CP     #00            ; is input 0?
0B91 2803            JR     Z,#0B96            ; yes, skip next step
;
0B93 32E564            LD     (#64E5),A            ; replicate input to input replica
;
0B96 3AE264            LD     A,(#64E2)            ; load power pellet mode active indicator
0B99 FE01            CP     #01            ; is the power pellet mode active?
0B9B 2804            JR     Z,#BBBB            ; yes, jump to label BBBB
;
0B9D 3A0762            LD     A,(#6207)            ; load A with movement indicator
```

```

0BA0 C9          RET          ; return
;
0BA1 3AE564      LD   A,(#64E5)    ; load input from input replica – label BBBB
;
0BA4 FE01        CP   #01         ; is last input #01 (moving right)
0BA6 2004        JR   NZ,#CCCC    ; no, jump ahead – jump to label CCCC
0BA8 3E54        LD   A,#54        ; load graphic facing right
0BA9 1812        JR   #YYYY        ; jump ahead – jump to label YYYY
;
0BAC FE02        CP   #02         ; is last input #02 (moving left) – label CCCC
0BAE 2004        JR   NZ,#DDDD    ; no, jump ahead – jump to label DDDD
0BB0 3ED4        LD   A,#D4        ; load graphic facing left
0BB2 180A        JR   #YYYY        ; jump ahead – jump to label YYYY
;
0BB4 FE04        CP   #04         ; is last input #02 (moving up) – label DDDD
0BB6 2004        JR   NZ,#EEEE    ; no, jump ahead – jump to label EEEE
0BB8 3E58        LD   A,#58        ; load graphic facing up
0BB9 1802        JR   #YYYY        ; jump ahead – jump to label YYYY
;
0BBC 3E56        LD   A,#56        ; load graphic facing down – label EEEE
;
0BBE 4F          LD   C,A        ; store graphic number in C – label YYYY
0BBF 3A1A60      LD   A,(FrameCounter) ; load A with the FrameCounter
0BC2 E602        AND  #02        ; reset all BIT's except BIT2
0BC4 1F          RRA           ; shift BIT2 to BIT1
0BC5 1F          RRA           ; shift BIT1 to BIT0
0BC6 81          ADD  A,C        ; add saved graphic number
;
0BC7 C9          RET          ; return – label ZZZZ

```

Gaat bijna goed. Laat tijdens de power pellet mode inderdaad een pacman sprite zien, maar de richting gaat nog niet helemaal goed. Even debuggen met de InputState en input replica, kijken of dat goed gaat. En dan de mapping controleren. Er zit ook nog een denkfout in, want de sprite verandert alleen wanneer er een nieuwe input komt. Ook als de mode start, blijft de sprite jumpman, totdat je beweegt en de sprite in pacman verandert. En andersom als de mode eindigt, blijft de pacman sprite totdat je beweegt en de sprite weer in jumpman verandert.

Mmm. Zou het kunnen dat deze jumpman routine voor het aanpassen van de sprite, niet altijd aangeroepen wordt? Maar alleen als de richting van jumpman daadwerkelijk verandert? Dus eventjes met debugger kijken wanneer de routine op #1DA6 aangeroepen wordt. Ja wordt inderdaad alleen aangeroepen bij update van de sprite na beweging. Maar dat betekent dat de sprite niet aangepast wordt als de power pellet mode actief wordt en jumpman stil staat. Kan het daarnaast zijn dat er soms ook nog keihard de sprite value beïnvloed wordt op andere plaatsen? Eventjes debuggen op #694D en kijken of deze nog ergens anders vandaan gezet wordt?

De richting gaat nu wel goed. Was een tiffout op #0BA1. Daar werd de #64E4 gelezen. Nog wel twee problemen: 1) bij overgang modes niet meteen aangepast en 2) geen animatie van happen. Dat klopt beiden. Omdat alleen bij veranderen van de richting, wordt steeds de sprite eenmalig aangepast bij

omdraaien van de richting. Maar daarna niet meer, dus de tweede sprite voor de animatie wordt niet geladen.

Misschien dit alles op te lossen door de routine voor het updaten van de jumpman sprite elke ronde vanuit de main aan te roepen?

Kan toegevoegd worden aan het stukje met additionele mains waarin ook de pellet pickup routines zitten:

```
11B2 CD7B0A    CALL #0A7B          ; call pellet pickup routine
11B5 CDB211    CALL #11B2          ; call power pellet pickup routine
11EB C9        RET               ; return
11B8 CDA61D    CALL #1DA6          ; call jumpman sprite update routine
11BB C9        RET               ; return
```

Ja dat update wel meteen. Alleen nog geen animatie. Even met de debugger kijken waarom dat niet goed gaat.

Ja gaat nu goed.

Nu versie v0.20 opgeslagen.

Maar de sprites kunnen beter en de positionering wanneer naar boven en naar beneden niet duidelijk wanneer bovenaan en onderaan de ladder.

De sprites zijn aangepast en allemaal onderin gepositioneerd. Ziet er nu wel goed uit.

Nu versie v0.21 opgeslagen.

Nu gaan kijken naar het kleurenpalet en vastleggen welke kleuren voor welke zaken. In ieder geval nog drie kleuren voor de spookjes en minimaal één kleur voor bonus items.



00: pacman	01: red ghost	02: jumpman	03: girder/ghost
04: girder/ghost	05: girder/ghost	06: girder/ghost	07: dk1, titel
08: dk2	09: pauline1	0A: pauline2	0B: normal barrel
0C: blue barrel	0D: pink ghost	0E: blue ghost	0F: orange ghost

Ook nog checken gebruik kleuren voor de bovenkant van het scherm. 3,4,5 en 6 zijn allemaal hetzelfde. Dat zou je kunnen vervangen door één. Heeft ook te maken met de andere schermen. Maar titel scherm ook te maken met de normale girders. Of terugbrengen naar twee en om en om gebruiken. Dan kun je in de andere schermen twee verschillende kleuren (afwisselend per regel) gebruiken.

Dit is de huidige regelverdeling:

```
Offset(h) 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
00000000 07 07 07 02 06 06 04 06 06 06 03 06 03 04 03 06 06 06 04 03 04 05 04 06 05 03 05 03 06 03
00000020 07 07 07 02 06 06 04 06 06 06 03 06 03 04 03 06 06 06 04 03 04 05 04 06 05 03 05 03 06 03
00000040 07 07 07 02 06 06 04 06 06 06 03 06 03 04 03 06 06 06 04 03 04 05 04 06 05 03 05 03 06 03
00000060 07 07 07 02 06 06 04 06 06 06 03 06 03 04 03 06 06 06 04 03 04 05 04 06 05 03 05 03 06 03
00000080 07 07 07 02 06 06 04 06 06 06 03 06 03 04 03 06 06 06 04 03 04 05 04 06 05 03 05 03 06 03
000000A0 07 07 07 02 06 06 04 06 06 06 03 06 03 04 03 06 06 06 04 03 04 05 04 06 05 03 05 03 06 03
000000C0 07 07 07 02 06 06 04 06 06 06 03 06 03 04 03 06 06 06 04 03 04 05 04 06 05 03 05 03 06 03
000000E0 07 07 07 02 06 06 04 06 06 06 03 06 03 04 03 06 06 06 04 03 04 05 04 06 05 03 05 03 06 03
```

Daarin regel 03 vervangen door 07 (daarmee wordt L=01 ook wit). Ook jumpman (aantal levens) worden nu de verkeerde kleur. Dat wel op 02 houden.

En dan regel 04 t/m 0F vervangen door afwisselend 03 en 04.

Dat gaat goed. Is nu zo:

```
Offset(h) 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
00000000 07 07 07 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03
00000020 07 07 07 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 04
00000040 07 07 07 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 04
00000060 07 07 07 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 04
00000080 07 07 07 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 04
000000A0 07 07 07 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 04
000000C0 07 07 07 02 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04
000000E0 07 07 07 02 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04 03 04
```

Dan de reserveringen van de verschillende blokken:



00: pacman	01: blue barrel	02: jumpman	03: girder/ghost
04: girder/ghost	05: bonusitem1	06: bonusitem2	07: dk1, titel
08: dk2	09: pauline1	0A: pauline2	0B: normal barrel
0C: red ghost	0D: pink ghost	0E: blue ghost	0F: orange ghost

De kleurcodes van de verschillende sprites:

Pacman geel: C-2J/C-2K: 00/03.

Ghost roze: C-2J/C-2K: 00/08.

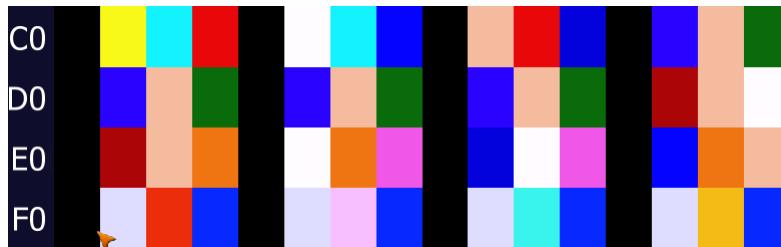
Ghost blauw: C-2J/C-2K: 0C/01.

Ghost oranje: C-2J/C-2K: 00/0B.

Dus de definitie van blok 01 kopiëren naar blok 0C, 0D, 0E en 0F. En dan aanpassen. En dan de definitie van blok 0C kopiëren naar blok 01. De blue barrel kleurcode is daarmee aangepast van #0C naar #01. Maar dan zitten de vier ghosts op een rijtje en kan je de instelling veel eenvoudiger doen.

Ja dat is aangepast en ziet er goed uit in de kleurentabel. Nu nog de kleuren aanpassen van pacman (00), en de drie ghosts (0D, 0E, 0F).

Ziet er goed uit. Ziet er nu zo uit:



Nu de kleurcodes gaan aanpassen in de code.

Bepalen van de initiële kleur van de ghosts gebeurt hier:

```

3151 3AA163 LD A,(#63A1) ; \ This fire slot is active. Increment count for # of fires onscreen
3154 3C INC A ; |
3155 32A163 LD (#63A1),A ; /
3158 3E01 LD A,#01 ; \ Set fire color to #01 (normal) if hammer is not active, and #00 (blue) if hammer is active
315A DD7708 LD (IX+#08),A ; |
315D 3A1762 LD A,(#6217) ; |
3160 FE01 CP #01 ; |
3162 C26A31 JP NZ,#316A ; |
3165 3E00 LD A,#00 ; |
3167 DD7708 LD (IX+#08),A ; /

```

Deze wordt eventueel vier keer doorlopen met B en DJNZ als loopmechanisme. Dus de waardes van B zijn 4, 3, 2, 1 en bij 0 stoppen met loop.

Dus dat betekent dat de kleurcode dan #0C+(#04-B) moet zijn = #10-B.

Aanpassen:

```

3158 CD8C11 CALL #11BC ; call additional code – call to label AAAA
315B 0000 NOP,NOP

```

en:

```

11BC 3E10 LD A,#10 ; load A with #10 – label AAAA
11BE 90 SUB B ; subtract B from A
;
11BF DD7708 LD (IX+#08),A ; set color of ghost
11C2 C9 RET ; return

```

Ja dat gaat helemaal goed. Ook wanneer de sprites in power pellet mode veranderen en daarna weer terug gaan naar oorspronkelijke sprites krijgen ze weer de juiste kleur.

Nu versie v0.22 opgeslagen.

Dan nu pacman de goede kleur geel geven. Dus bij zetten van de jumpman kleuren de kleur afhankelijk maken van de power pellet mode active.

Dat zit hier:

```
1DA6 214C69 LD HL,#694C ; load HL with mario sprite X position
1DA9 3A0362 LD A,(#6203) ; load A with mario's X position
1DAC 77 LD (HL),A ; store into hardware sprite mario X position
1DAD 3A0762 LD A,(#6207) ; load A with movement indicator
1DB0 2C INC L ; HL := #694D = hardware mario sprite
1DB1 77 LD (HL),A ; store into hardware mario sprite value
1DB2 3A0862 LD A,(#6208) ; load A with mario color
1DB5 2C INC L ; HL := #694E = hardware mario sprite color
1DB6 77 LD (HL),A ; store into mario sprite color
1DB7 3A0562 LD A,(#6205) ; load A with mario Y position
1DBA 2C INC L ; HL := #694F = mario sprite Y position
1DBB 77 LD (HL),A ; store into mario sprite Y position
1DBC C9 RET ; return
```

Waarin dit aangepast om de sprite aan te passen:

```
1DAD CD8C0B CALL #0B8C ; call additional code – call label AAAA
```

Na die aanroep wordt naar #1DB0 teruggekeerd.

Aanpassen:

```
1DB2 CDC311 CALL #11C3 - call additional code – call label AAAA
```

en:

```
11C3 3AE264 LD A,(#64E2) ; load power pellet mode active indicator
11C6 FE01 CP #01 ; is the power pellet mode active?
11C8 3A0862 LD A,(#6208) ; load A with jumpman color
11CB C0 RET NZ ; no, return
;
11CC 3E00 LD A,#00 ; load A with pacman color
11CE C9 RET ; return
```

Ja dat gaat goed.

Nu versie v0.23 opgeslagen.

Dan nu kijken naar de bonus items (het fruit). Twee sloten vrij om daar de kleuren van vast te leggen. Zijn er drie bonus items te gebruiken met twee verschillende kleurencombinaties? Ja kers, aardbei en appel waarbij kers en appel dezelfde kleuren gebruiken en ze allemaal voornamelijk rood zijn.

Kers en appel: kleurblok #05: rood, wit, bruin.

Aardbei: kleurblok #06: rood, wit, groen.

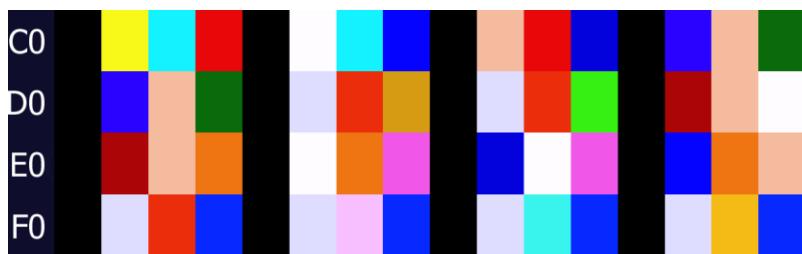
Zelfde rood en wit als de rode ghost. En dan de blauw daarvan vervangen door bruin en groen. De kleurcodes daarvan zijn:

Bruin: C-2J/C-2K: 02/0F.

Groen: C-2J/C-2K: 0C/03.

Dus kopie van rode ghost maken en dan aanpassen.

Ziet er nu zo uit:



En dan nog de bonus-item sprites aanpassen. Meteen voor bonus items op alle boards.

; bonus items on conveyors

```
3E3C 53 73 05 A0      ; position of hat on pies is 53,A0  
3E40 8B 74 06 F0      ; position of purse on pies is 8B,F0  
3E44 DB 75 05 A0      ; umbrella on the pies is at DB,A0
```

; bonus items for elevators

```
3E48 5B 73 05 C8      ; hat at 5B,C8  
3E4C E3 74 06 60       ; purse at E3,60  
3E50 1B 75 05 80       ; umbrella on elevator is 80,1B
```

; bonus items for rivets

```
3E54 DB 73 05 C8      ; hat on rivets at DB,C8  
3E58 93 74 06 F0      ; purse on rivets at 93,F0  
3E5C 33 75 05 50       ; umbrella on rivets at 33,50
```

Ziet er nu zo uit:



Nu versie v0.24 opgeslagen.

Nog wel een probleem. Wanneer jumpman (of eigenlijk pacman) afgaat terwijl de power pellet mode actief is dan wordt de sprite teruggezet naar jumpman (op rug liggend draaiend), maar maakt dan nog gebruik van de pacman kleuren.

Waar wordt dit gedaan?

```
127C CDBD1D    CALL    #1DBD      ; check for bonus items and jumping scores, rivets
127F 3A9D63    LD     A,(#639D)   ; load A with this normally 0. 1 while mario dying, 2 when dead
1282 EF        RST    #28       ; jump based on A

1283 8B 12      ; #128B      ; 0 normal
1285 AC 12      ; #12AC      ; 1 mario dying
1287 DE 12      ; #12DE      ; 2 mario dead
1289 00 00      ; unused ?
```

De eerste call is om eventueel verkregen scores nog bij te werken voor daadwerkelijk dood gaan.

Eigenlijk de code van het aanpassen van de jumpman color aanpassen dat dit alleen gedaan wordt wanneer #639D 0 is (normal en niet dying of dead).

```
11C3 3AE264    LD     A,(#64E2)   ; load power pellet mode active indicator
11C6 FE01      CP     #01       ; is the power pellet mode active?
11C8 3A0862    LD     A,(#6208)   ; load A with jumpman color
11CB C0        RET    NZ       ; no, return
;
11CC 3E00      LD     A,#00      ; load A with pacman color
11CE C9        RET    ; return
```

Aanpassen:

```
11CC C38C1E    JP     #1E8C      ; jump to additional code
```

en:

```
1E8C 3A9D63    LD     A,(#639D)   ; load jumpman alive indicator
1E8F FE00      CP     #00       ; is jumpman alive?
1E91 3A0862    LD     A,(#6208)   ; load A with jumpman color
1E94 C0        RET    NZ       ; no, return
;
1E95 3E00      LD     A,#00      ; load A with pacman color
1E97 C9        RET    ; return
```

Dit gaat niet goed. Komt daar helemaal niet meer in de routine die vanuit de main aangeroepen wordt. Dus waarschijnlijk uit de main al weggesprongen. Als dat zo is, dan de kleur hier gewoon terugzetten (ongeacht power pellet mode).

Ja, dit is op basis van GameMode2, dus buiten de normale spel main.

Dan zo aanpassen:

```
127F CD8C1E      CALL #1E8C      ; call additional code – call to label AAAA
```

en:

```
1E8C 3E02      LD A,#02      ; load A with jumpman color
1E8E 324E69    LD (#694E),A ; set jumpman sprite to original color
1E91 3A9D63    LD A,(#639D) ; load A with jumpman dying state
1E94 C9        RET          ; return
```

Dat werkt gedeeltelijk. De kleur wordt wel meteen aangepast, maar de sprite is eerst nog een aantal seconden de pacman sprite. Dus dat is waarschijnlijk door de verschillende dying states. Misschien hier ook meteen de sprite aanpassen. De sprite #78 gebruiken.

```
1E8C 3A9D63    LD A,(#639D) ; load A with jumpman dying state
1E8F FE00      CP #00        ; is the dying state 0 (start dying)
1E91 C0        RET NZ       ; no, return
;
1E92 47        LD B,A       ; save A to B for later
;
1E93 3E78      LD A,#78       ; load A with jumpman dying sprite
1E95 324D69    LD (#694D),A ; set jumpman sprite to dying sprite
1E98 3E02      LD A,#02       ; load A with jumpman color
1E9A 324E69    LD (#694E),A ; set jumpman sprite to original color
;
1E9D 78        LD A,B       ; restore A from B
1E9E C9        RET          ; return
```

Ja, dat gaat nu goed.

[Nu versie v0.25 opgeslagen.](#)

Nu kijken naar de invincibility mode en opeten ghosts tijdens de power pellet mode. En alleen ghosts eten. Je kan geen barrels eten, maar je gaat er ook niet door af. En door de invincibility kun je ook bijvoorbeeld achter DK langslopen (op rivets), maar als de power pellet mode afloopt zul je waarschijnlijk meteen afgaan.

Invincibility mode wordt gedaan vanuit #19B3. Die weer herstellen, maar er een extra stukje tussendoen die het geheel overslaat wanneer de poer pellet mode actief is.

Aanpassen:

```
19B3 CD9F1E      CALL #1E9F      ; call additional code – call to label AAAA
```

en:

```
1E9F 3AE264      LD    A,(#64E2)      ; load power pellet mode active indicator
1EA2 FE01        CP    #01          ; is the power pellet mode active?
1EA4 C8          RET   Z           ; yes, return
;
1EA5 CD0828      CALL  #2808       ; check for collisions with hostile sprites
1EA8 C9          RET   ; return
```

Ja dat werkt goed.

Nu versie v0.26 opgeslagen.

Nu gaan kijken of er in de normale uitvoering ergens een check gedaan kan worden op collisions met een ghost. Bij hamers werd de hit zo gedetecteerd:

```
2832 FD4E05      LD    C,(IY+#05)    ; C := +5 (X position??)
2835 FD6609      LD    H,(IY+#09)    ; H := +9 (size? width?)
2838 FD6E0A      LD    L,(IY+#0A)    ; L := +A (size? height?)
283B CD6F28      CALL  #286F       ; checks for collisions based on the screen. A := 1 if collision
283E A7          AND   A           ; was there a collision?
283F C8          RET   Z           ; no, return
```

Dit roept code op #286F aan en die checkt eerst het board en roept dan verschillende collision detections aan afhankelijk van het board (fireballs, barrels, pies, spring, enz). Voor rivets is dat:

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

Dit checkt dus 7 elementen (5 fireballs en 2 kong boxes), maar dat is in ons geval niet nodig. Wij hoeven er maar 4 te checken (4 ghosts). De aanroep naar #2913 bepaalt of er een collision heeft opgetreden. Zo ja, dan wordt A op 1 gezet en anders op 0. En als 1 dan resultaat opslaan in de geheugenplaats #6350 (was origineel item hit with hammer, maar nu ghost eaten by pacman).

En dan een aparte routine aan het begin van de main die de grafische afhandeling met ogen animatie en bonuspunten toekenning doet.

Leeg slot in de main voor de collision detection: #19B6.

Aanpassen:

```
19B6 CDA91E      CALL  #1EA9       ; call additional code – call to label AAAA
```

en:

```
1EA9 3AE264      LD   A,(#64E2)    ; load power pellet mode active indicator
1EAC FE01        CP   #01         ; is the power pellet mode active?
1EAE C0          RET  NZ          ; no, return
;
1EAF 0604        LD   B,#04       ; B := 4
1EB1 112000       LD   DE,#0020    ; load DE with offset
1EB4 DD210064     LD   IX,#6400    ; load IX with start of firefox arrays
1EB8 CD1329       CALL #2913     ; check for collisions with firefoxes/squares
;
1EBB FE01        CP   #01         ; is there a collision detected?
1EBD C0          RET  NZ          ; no, return
;
1EBE 325063       LD   (#6350),A  ; set ghost eaten indicator
1EC1 C9          RET           ; return
```

Dat gaat nog niet goed. Lijkt erop dat hij elke keer in de loop van de main naar #1EA9 springt dan naar de collision detection, daar dan weer uitkomt maar dan naar de call een niveau hoger terugkeert. Dus naar de call in de main na de call naar #1EA9. En dus het resultaat niet beoordeeld wordt. Lijkt te kloppen. Dus daar rekening mee houden.

```
1EA9 3AE264      LD   A,(#64E2)    ; load power pellet mode active indicator
1EAC FE01        CP   #01         ; is the power pellet mode active?
1EAE C0          RET  NZ          ; no, return
;
1EAF CD391F       CALL #1F39     ; call additional routine – call BBBB
;
1EB2 FE01        CP   #01         ; is there a collision detected?
1EB4 C0          RET  NZ          ; no, return
;
1EB5 325063       LD   (#6350),A  ; set ghost eaten indicator
1EB8 C9          RET           ; return
```

en:

```
1F39 0604        LD   B,#04       ; B := 4
1F3B 112000       LD   DE,#0020    ; load DE with offset
1F3E DD210064     LD   IX,#6400    ; load IX with start of firefox arrays
1F32 CD1329       CALL #2913     ; check for collisions with firefoxes/squares
1F35 C9          RET           ; return
```

Ja, dit lijkt goed te werken. De #6350 wordt gevuld.

Nu versie v0.27 opgeslagen.

En dan nu gaan kijken of je kan zien welke ghost dat dan is? Kan dat op basis van de IX? Eerst eens kijken wat er in de IX staat wanneer ghost eaten.

Nee toch nog niet goed. De #1EB5 wordt meteen gedaan na oppakken power pellet en dus wordt dan ook meteen de 6350 met #01 gevuld. Dat is niet goed.

En is ook erg logisch. Duh! Je moet voor het aanroepen niet alleen de gegevens van de fireballs klaarzetten, maar ook de gegevens van jumpman!

```
; checks for collisions with hostiles sprites

2808 FD210062 LD IY,#6200 ; load IY with start of mario sprite
280C 3A0562 LD A,(#6205) ; load A with mario's Y position
280F 4F LD C,A ; copy to C
2810 210704 LD HL,#0407 ; H := 4, L := 7
2813 CD6F28 CALL #286F ; checks for collisions based on the screen.
2816 A7 AND A ; was there a collision ?
2817 C8 RET Z ; no, return
```

Aanpassen:

```
1EAF FD210062 LD IY,#6200 ; load IY with start of mario sprite
1EB3 3A0562 LD A,(#6205) ; load A with mario's Y position
1EB6 4F LD C,A ; copy to C
1EB7 210704 LD HL,#0407 ; H := 4, L := 7
;
1EBA CD391F CALL #1F39 ; call additional routine – call BBBB
;
1EBD FE01 CP #01 ; is there a collision detected?
1EBF C0 RET NZ ; no, return
;
1EC0 325063 LD (#6350),A ; set ghost eaten indicator
1EC3 C9 RET ; return
```

Ja gaat nu wel goed.

Nu versie v0.28 opgeslagen.

Na collision detect is de IX weer #6400. Dus daar kun je niets aan zien. Wel aan de B.

B = 4 Rode spook = 6400 = 69D0
B = 3 Roze spook = 6420 = 69D4
B = 2 Blauwe spook = 6440 = 69D8
B = 1 Oranje spook = 6460 = 69DC

Dan slim gaan kijken en de betreffende ghost inactief maken en verwijderen. Daarna een sprite voor de ogen animatie starten en aparte routine om deze af te handelen (terwijl stilstaan).

De animatie wordt gedaan met sprite op #6A2C. En animatie is door bit flip zodat tussen de twee spites geswitched wordt. Wij nu doen door twee ogen laten bewegen van links naar rechts!

```

1EC3 21D069      LD   HL,#69D0    ; load HL with first ghost sprite address
1EC6 3E04        LD   A,#04      ; load A with #04
1EC8 90          SUB  B          ; subtract B
;
1EC9 112000      LD   DE,#0020    ; load DE with definition offset
1ECC 010400      LD   BC,#0004    ; load BC with sprite offset

;

1ECF FEO0        CP   #00        ; is A already 0? - label AAAA
1ED1 2806        JR   Z,#BBBB    ; yes, jump to label BBBB
;
1ED3 DD19        ADD  IX,DE     ; adjust to next ghost in definition
1ED5 09          ADD  HL,BC     ; adjust to next ghost sprite
1ED6 3D          DEC  A         ; decrement A
1ED7 18F6        JR   #AAAA     ; jump back to label AAAA
;
1ED9 DD360000    LD   (IX+#00),#00 ; set fire inactive – label BBBB
1EDD 3600        LD   (HL),#00    ; move sprite offscreen
;
1EDF C9          RET           ; return

```

Ja dat gaat goed.

Nu versie v0.29 opgeslagen.

Nu dan de animatie doen. De sprites van de hamer hit omzetten naar oogjes en die dan invullen in de bovenstaande routine (op de plek van de geresette ghost) en dan een extra routine de die animatie doet en aan het einde de bonus punten geeft.

De vier sprites voor de animatie zitten van 60 t/m 63. De sprite die in het originele programma gebruikt wordt voor het weergeven van de rondjes is de #6A2C. Die hergebruiken.

Aanpassen:

```

1EDD 01306A      LD   BC,#6A30    ; load BC with scoring sprite address
1EE0 7E          LD   A,(HL)     ; load A with eaten ghost X-position
1EE1 02          LD   (BC),A    ; store in scoring sprite X-position
1EE2 3600        LD   (HL),#00    ; move ghost sprite offscreen
1EE4 0C          INC  C         ; next position
1EE5 2C          INC  L         ; next position
1EE6 3E60        LD   A,#60      ; load A with value first eyes sprite
1EE8 02          LD   (BC),A    ; store in scoring sprite value
1EE9 0C          INC  C         ; next position
1EEA 2C          INC  L         ; next position
1EEB 3E0C        LD   A,#0C      ; load A with sprite color normal ghost
1EED 02          LD   (BC),A    ; store in scoring sprite color
1EEE 0C          INC  C         ; next position

```

```

1EEF  2C          INC   L           ; next position
1EF0  7E          LD    A,(HL)      ; load A with eaten ghost Y-position
1EF1  02          LD    (BC),A     ; store in scoring sprite Y-position
;
1EF2  3E00         LD    A,#00       ; load A with #00
1EF4  32E664       LD    (64E6),A   ; reset eaten animation counter
;
1EF7  C9          RET              ; return

```

Dit werkt goed. Er worden ogen getoond op de plaats waar het spookje verdwijnt. Alleen nu nog de animatie van de ogen maken.

Dat kan door in één van de eerste sloten van de main een aanroep te doen en de animatie afhandeling te regelen. En dan steeds weer terug springen naar een niveau hoger zodat de main meteen weer verlaten wordt. Zodoende staat het spel even stil. Daar testen op #6350 ghost eaten indicator. Als die 1 is dan uitvoeren. Als die 0 is dan een return doen. En dan op basis van #64E6 eaten animation counter de oog sprite value verhogen. En wanneer klaar, dan oog sprite buiten beeld plaatsen, de #6350 resetten en bonus toekennen en weergeven.

Het main slot van #197D wordt al gebruikt. Voor aanroepen van de routine die de power pellet mode timers bijwerkt. Maar dat hoeft niet per sé aan het begin van de main. En dit wel!

Dus die verplaatsen:

Aanpassen:

```
1998 CD520B      CALL  #0B52      ; call routine for power pellet mode timing
```

Aanpassen:

```
197D CDF81E      CALL  #1EF8      ; call additional routine – call to label AAAA
```

en:

```

1EF8  3A5063      LD    A,(6350)   ; load A with ghost eaten indicator -label AAAA
1EFB  FE01          CP    #01        ; is ghost eaten
1EFD  C0            RET   NZ        ; no, return
;
1EFE  CD031F      CALL  #1F03      ; call routine – call to label BBBB
1F01  E1            POP   HL        ; then return to a higher sub
1F02  C9            RET              ; returns to #00D2
;
1F03  3AE664      LD    A,(64E6)   ; load A with eaten animation counter - BBBB
1F06  3C            INC   A          ; increment eaten animation counter
1F07  32E664      LD    (64E6),A  ; store back
;
1F0A  FE20          CP    #20        ; is A 20?
1F0C  280D          JR    Z,CCCC    ; yes, jump to label CCCC

```

```

1FOE FE40      CP    #40      ; is A 40?
1F10 2809      JR    Z,CCCC   ; yes, jump to label CCCC
1F12 FE60      CP    #60      ; is A 60?
1F14 2805      JR    Z,CCCC   ; yes, jump to label CCCC
1F16 FE80      CP    #80      ; is A 80?
1F18 2809      JR    Z,DDDD   ; yes, jump to label DDDD
;
1F1A C9        RET   .       ; return
;
1F1B 3A2D6A   LD    A,(#6A31) ; load A with eyes sprite value - CCCC
1F1E 3C        INC   A       ; get next sprite value
1F1F 322D6A   LD    (#6A31),A ; set next sprite
1F22 C9        RET   .       ; return
;
1F23 3E00      LD    A,#00   ; load A with #00 - DDDD
1F25 325063   LD    (#6350),A ; reset ghost eaten indicator
;
1F28 110500   LD    DE,#0005 ; 500 points
1F2B 067E      LD    B,#7E   ; sprite for 500
1F2D CDE10A   CALL  #0AE1   ; yes, award points
1F30 C9        RET   .       ; return

```

Gaat op zich nu goed. Maar de bonus sprite blijft staan.

Nu versie v0.30 opgeslagen.

Dit wordt al gedaan bij het zetten van de all pellet bonus.

Er bestaat een bonus timer #6341. Die moet je waarschijnlijk zetten.

Die even aanpassen zodat je er hiervandaan ook heen kan springen:

0ADC

```

0ADC 218560   LD    HL,#6085   ; else load HL with bonus sound address
0ADF 3603     LD    (HL),#03   ; play bonus sound for 3 duration
;
0AE1 110B00   LD    DE,#000B   ; 1000 points
0AE4 066B     LD    B,#6B     ; sprite for pellet bonus
0ADC 110B00   LD    DE,#000B   ; 1000 points
0ADF 066B     LD    B,#6B     ; sprite for pellet bonus
;
0AE1 218560   LD    HL,#6085   ; else load HL with bonus sound address
0AE4 3603     LD    (HL),#03   ; play bonus sound for 3 duration
;
0AE6 CD9F30   CALL  #309F     ; insert task to add score
0AE9 3A0562   LD    A,(#6205)  ; load A with Mario's Y position
;
0AEC CD301E   CALL  #1E30     ; call subroutine to draw pellet bonus sprite

```

```

;
0AEF 3E60      LD   A,#60      ; A := #60
0AF1 324163    LD   (#6341),A ; store into timer
0AF4 3E02      LD   A,#02      ; A := 2
0AF6 324063    LD   (#6340),A ; store into #6340
;
0AF9 C9        RET            ; return

```

Ja, gaat nu wel goed.

Nu versie v0.31 opgeslagen.

Maar de bonus score wordt weergegeven op de positie van jumpman. Beter om dat doen op de plek van de originele ghost. En die X/Y positie staat al in de sprite. Dus je hoeft in principe alleen maar de waarde nog éénmaal te verhogen en dan de timers te zetten.

```

1F28 110500    LD   DE,#0005  ; 500 points
1F2B CD9F30    CALL #309F    ; award points
;
1F2E 3E7E      LD   A,#7E      ; sprite for 500
1F30 32316A    LD   (#6A31),A ; set eyes sprite to bonus 500 sprite
1F33 CDEFOA    CALL #0AEF    ; set bonus item display timers
1F36 C9        RET            ; return

```

Ja, geeft nu bonus op de juiste plaats, maar in het blauw en niet met een bonus sound. Zou nog kunnen door de eerdere pellet bonus routine nogmaals aan te passen.

```

0ADC 110B00    LD   DE,#000B  ; 1000 points
0ADF 066B      LD   B,#6B      ; sprite for pellet bonus
;
0AE1 218560    LD   HL,#6085  ; else load HL with bonus sound address
0AE4 3603      LD   (HL),#03  ; play bonus sound for 3 duration
;
0AE1 CD9F30    CALL #309F    ; insert task to add score
0AE4 3A0562    LD   A,(#6205) ; load A with Mario's Y position
;
0AE7 CD301E    CALL #1E30    ; call subroutine to draw pellet bonus sprite
;
0AEA 218560    LD   HL,#6085  ; else load HL with bonus sound address
0AED 3603      LD   (HL),#03  ; play bonus sound for 3 duration
;
0AEF 3E60      LD   A,#60      ; A := #60
0AF1 324163    LD   (#6341),A ; store into timer
0AF4 3E02      LD   A,#02      ; A := 2
0AF6 324063    LD   (#6340),A ; store into #6340
;
0AF9 C9        RET            ; return

```

en:

```
1F33 CDEA0A          CALL #0AEA           ; set bonus sound and bonus display timers
```

Dat gaat nu goed.

Nu versie v0.32 opgeslagen.

De bonus tekst wordt echter weergegeven in blauw. Kan dat in wit? Welke kleur blok wordt er gebruikt voor de ogen? #0C, maar wit is #07. Die zetten.

```
1F28 110500          LD   DE,#0005      ; 500 points
1F2B CD9F30          CALL #309F       ; award points
;
1F2E 3E7E             LD   A,#7E        ; sprite for 500
1F30 32316A          LD   (#6A31),A    ; set eyes sprite to bonus 500 sprite
1F33 CD8503          CALL #0385      ; call additional code
1F36 C9               RET            ; return
```

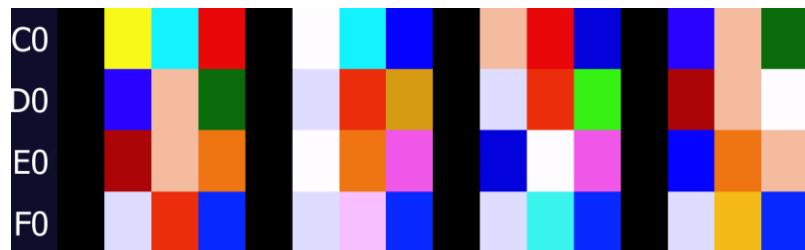
en:

```
0385 3E07             LD   A,#07        ; color white
0387 32326A          LD   (#6A32),A    ; set bonus sprite to color white
038A CDEA0A          CALL #0AEA      ; set bonus item display timers
038D C9               RET            ; return
```

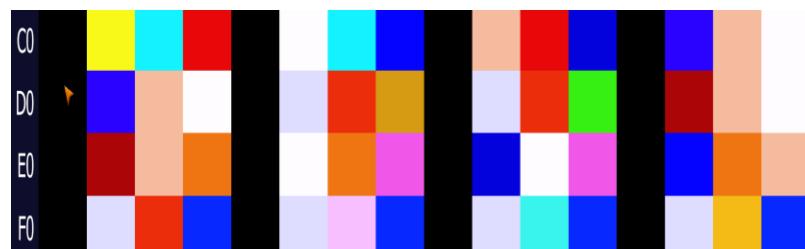
Ja dat gaat nu goed.

Nu versie v0.33 opgeslagen.

Dan nu het groen in de background wit maken: mooier boven in bij bonus en bij game over.



Dat zijn dus de derde kleuren in blok #03 en #04. Dus #CF en #D3 gelijk maken aan #DF.



En dan de all pellet bonus 800 points maken en 800 bonus sprite weergeven.

```
0ADC 110800      LD   DE,#0008      ; 800 points
0ADF 067F        LD   B,#7F        ; sprite for pellet bonus
```

Even testen met minder pellets op te eten:

```
0AD3 3E5C        LD   A,#5C        ; load A with nr of pellets on rivets
```

Even op #10 zetten: na 16 pellets voldoende.

Ja dat gaat nu goed.

En dan knipperen met ander kleurenpalette. Net als ghost eaten aan begin en stil staan.

Het palette nu is die van rivets.

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

Die van de pie factory als alternatief gebruiken. Dus paletteB flippen.

#64E1 bevat de pellet bonus indicator. Als #00 dan geen pellet bonus, dus nog niet alle pellets opgegeten. Dan als dat gezet is gaan ophogen in een routine die terug springt naar een niveau hoger en dus de main meteen weer verlaat. En dan op basis van een BIT het palette aanpassen.

```
281D 3AE164      LD   A,(#64E1)    ; load A with pellet bonus indicator
2820 FE00        CP   #00          ; is pellet bonus indicator 0?
2822 C8          RET  Z           ; yes, return and continue main
;
2823 FE80        CP   #80          ; is pellet bonus indicator 80?
2825 C8          RET  Z           ; yes, return and continue main
;
2826 CD2A28      CALL #282A      ; call additional routine, call label AAAA
2829 C9          RET           ; return
;
282A 3C          INC  A           ; increment A – label AAAA
282B 32E164      LD   (#64E1),A  ; store in pellet bonus indicator
;
282E CB67        BIT  4,A        ; is BIT 4 set?
2830 3E00        LD   A,#00      ; set A to 00 to clear reg palette B
2832 2002        JR   NZ         ; yes, jump ahead – jump to label BBBB
;
```

```

2834 3E01          LD   A,#01           ; set A to 01 to set reg palette B
2836 32877D        LD   (REG_PALETTE_B),A ; change reg palette B – label BBBB
;
2839 E1            POP  HL              ; return to a higher sub
283A C9            RET               ; returns to #00D2

```

En dan het tweede slot uit de main nemen.

```
197D CD8E03        CALL  #038E         ; call additional code
```

en:

```

038E CDF81E        CALL  #1EF8         ; call ghost eaten animation
0391 CD1D28        CALL  #281D         ; call pellet bonus board blinking
0394 C9            RET               ; return

```

Dat gaat dus niet. Want een return to higher sub gaat dan naar de tussenliggende sub op #038E en dan wordt er dus niet stilgestaan. Ook niet meer bij eten van een ghost. Dat willen we wel, dus dan wellicht herstellen naar tweede slot in de main en dan voor de blinking board animation het derde slot nemen.

Herstellen tweede slot:

```
197D CDF81E        CALL  #1EF8         ; call ghost eaten animation
```

En het stukje code op #038E weer wissen.

Dat is weer goed. De ghost eaten animatie pauzeert het spel weer.

Dan dit anders oplossen. In slot 3 wordt gesprongen naar code die iets doet met jumpen. Daar de animatie voor board blinking aan toevoegen. Als geen board blink dan geen return doen, maar jump naar code die eerst in het derde slot aangeroepen werd.

```

281D 3AE164        LD   A,(#64E1)      ; load A with pellet bonus indicator
2820 FE00          CP   #00           ; is pellet bonus indicator 0?
2822 CAC31A        JP   Z,#1AC3       ; yes, continue main – check for jumping
;
2825 FE80          CP   #80           ; is pellet bonus indicator 80?
2827 CAC31A        JP   Z,#1AC3       ; yes, continue main – check for jumping
;
282A CD2E28        CALL  #282E         ; call additional routine, call label AAAA
282D C9            RET               ; return
;
282E 3C             INC   A             ; increment A – label AAAA
282F 32E164        LD   (#64E1),A    ; store in pellet bonus indicator
;
2832 CB67          BIT   4,A          ; is BIT 4 set?
2834 3E00          LD   A,#00         ; set A to 00 to clear reg palette B

```

```

2836 2002          JR   NZ,#283A      ; yes, jump ahead – jump to label BBBB
;
2838 3E01          LD   A,#01        ; set A to 01 to set reg palette B
283A 32877D        LD   (REG_PALETTE_B),A ; change reg palette B – label BBBB
;
283D E1            POP  HL        ; return to a higher sub
283E C9            RET           ; returns to #00D2

```

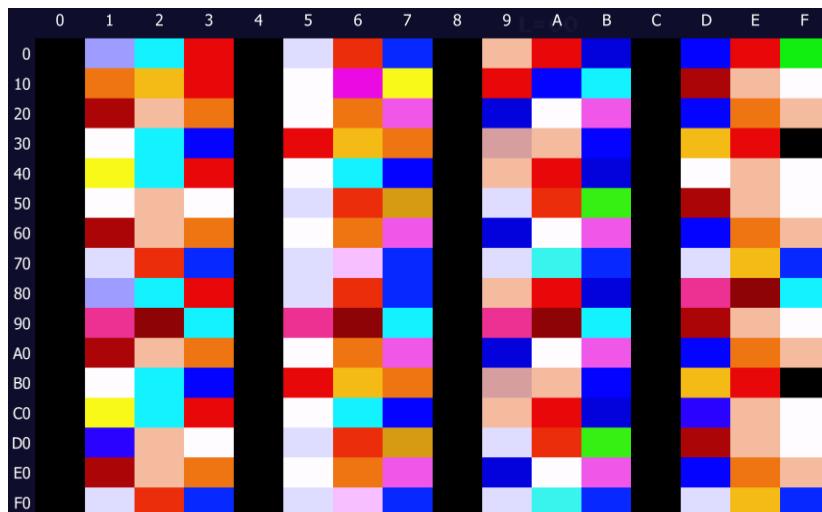
en:

```
1980 CD1D28        CALL #281D      ; do board blink animation or check for jumping
```

Ja dat gaat goed.

Nu versie v0.34 opgeslagen.

Even het tweede palette gelijk maken aan het vierde palette en dan het blauw van de girders aanpassen naar wit.



Nee dat gaat toch niet goed. Jumpman kan niet bewegen maar de rest van de main wordt wel doorlopen, dus de ghosts bewegen nog wel. Klopt de POP HL wel?

Checken met debugger. Wanneer in de routine van blink animation dan staat de SP te ver en lijkt het erop dat er twee keer een POP HL gedaan moet worden.

Eventjes testen:

```

283D E1            POP  HL        ; return to a higher sub
283E E1            POP  HL        ; return to a higher sub
283F C9            RET           ; returns to #00D2

```

Ja dat gaat wel goed.

Wel sneller laten knipperen. Dan dus BIT 3,A doen:

2832 CB5F BIT 3,A ; is BIT 3 set?

En minder lang dus laten knipperen.

Stoppen bij #40:

2825 FE40 CP #40 ; is pellet bonus indicator 40?

Ja dat is helemaal perfect.

En dan het aantal pellets weer terugzetten:

0AD3 3E5C LD A,#5C ; load A with nr of pellets on rivets

Ja dat is allemaal goed.

Nu versie v0.35 opgeslagen.

Kijken herpositioneren van de power pellets. In ieder geval niet direct boven een ladder plaatsen. Eerst maar eens een positie opschuiven t.o.v. waar ze nu staan en dan gaan testen. De bovenste één positie naar links (+#20) en de onderste één positie naar rechts (-#20).

Definitie nu:

3E14 0F 76 19 76

#760F = bovenste → #762F en #7619 = onderste → #75F9

Aanpassen:

3E14 2F 76 F9 75

Ja dat is goed zo.

De spookjes gaan in power pellet mode knipperen. Daarbij worden de oogjes de oorspronkelijke kleur van het spookje. In de originele pacman is het altijd rood. Kijken of dat aangepast kan worden.

Hij wisselt tussen de aangepaste kleur #03 en de oorspronkelijke kleur van het spookje. Misschien aanpassen naar wisselen tussen de aangepaste kleur en de oorspronkelijke kleur van het rode spookje. Alleen dan aan het einde wellicht wel de kleur weer terugzetten naar de oorspronkelijke kleur van het spookje.

De code die dat nu doet:

```
0B6D 1C      INC      E      ; next DE – label AAAA  
0B6E 7E      LD      A,(HL)      ; load A with original sprite color value  
0B6F 4F      LD      C,A      ; store original sprite color in C  
;
```

```

0B70 3AE264      LD   A,(#64E2)    ; load A with power pellet mode active indicator
0B73 FE01        CP   #01         ; is the power pellet mode active?
0B75 200E        JR   NZ,#BBBB    ; no, jump to label BBBB
;
0B77 3AE464      LD   A,(#64E4)    ; load A with power pellet mode timer 2
0B7A FE00        CP   #00         ; is power pellet mode time 2 0?
0B7C 280A        JR   Z,#CCCC    ; yes, first half, jump to label CCCC
;
0B7E 3AE364      LD   A,(#64E3)    ; load A with power pellet mode timer 1
0B81 CB6F        BIT  5,A         ; is bit 5 set?
0B83 2003        JR   NZ,#CCCC    ; yes, second half blink, jump to label CCCC
;
0B85 79          LD   A,C         ; restore original color into A – label BBBB
0B86 1802        JR   #DDDD      ; skip next step, jump to label DDDD
;
0B88 3E03        LD   A,#03      ; set different color into A – label CCCC
;
0B8A 12          LD   (DE),A     ; store sprite color – label DDDD
0B8B C9          RET           ; return

```

Dus wanneer power pellet mode niet (meer) actief, springen naar label BBBB en daar de originele kleur terugzetten.

Wanneer de power pellet mode wel actief dan twee mogelijkheden waarbij onderscheid op basis van BIT5. De helft van de tijd naar label CCCC en de aangepaste kleur gebruiken of doorlopen naar label BBBB en de oorspronkelijke kleur gebruiken. Dus het stuk van de test aanpassen naar een jump naar additionele code en daar de check doen en juist springen.

Aanpassen:

```

0B81 C3C80B      JP   #0BC8      ; jump to additional code – jump to label YYYY
0B84 00          NOP

```

en:

```

0BC8 CB6F        BIT  5,A         ; is bit 5 set? – label YYYY
0BCA C2880B      JP   NZ,#0B88    ; yes, blink other color, jump to label CCCC
;
0BCD 3E0C        LD   A,#0C      ; no, blink red ghost color
0BCF C38A0B      JP   #0B8A      ; jump to label DDDD

```

Nog testen...

Ja dat gaat goed.

Nu versie v0.36 opgeslagen.

Kijken naar een nieuwe versie van het barrel board. Kijken naar rechte horizontale girders met aan de zijkanten een klein stukje dat schuin loopt om de illusie te geven van naar beneden rollen. Eerst eens kijken naar de positionering van de huidige girders.

```

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

```

Dat zijn van boven naar beneden: 6D/79 – 9A/8E – AF/BB – DC/D0

En de middenstukken daarvan: 73 94 B5 D6

En de verschillen daartussen: 21 21 21

Dus gebruik gaan maken van een afstand van #20. En dan uitlijnen op #0 en #08.

Dus de girders van beneden naar boven op #D8, #B8, #98, #78. Dat lijkt goed, maar zit een beetje naar beneden geschoven en de schuine stukken gaan ook naar beneden. Dus wellicht juist iets hoger zetten? Nee. Zo houden. Eventueel de bottom girder verder horizontaal maken, dus pas verder naar rechts pas schuin maken. Een beetje afhankelijk van de trap afstanden/positionering.

De rechte stukken zitten nu van #40 tot #C0. Voor eenvoudige logica voor het bijstellen van de Y-positie op basis van de helling is het wellicht handig als de top en bottom schuine delen hier ook mee uitlijnen.

3AE9: 02 9F 54 10 54 ; girder where kong sits	-> 02 BF 54 10 54
3AED: 02 DF 58 A0 55 ; 1st slanted girder at top right	-> 02 DF 56 C0 55
3B07: 02 FF F0 80 F7 ; bottom slanted girder	-> 02 FF F4 C0 F5
3B0C: 02 7F F8 00 F8 ; bottom flat girder where mario starts	-> 02 BF F8 00 F8

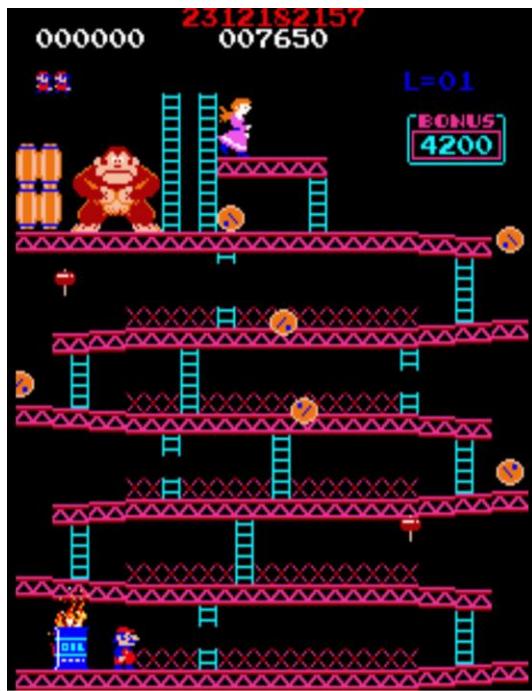
```

$girderarray = @C ((0x02,0x3F,0xD7,0x10,0xD5), (0x02,0xBF,0xD8,0x40,0xD8), (0x02,0xDF,0xDA,0xC0,0xD8), # girder 1
(0x02,0x3F,0xB9,0x20,0xBA), (0x02,0xBF,0xB8,0x40,0xB8), (0x02,0xEF,0xB5,0xCD,0xB7), # girder 2
(0x02,0x3F,0x97,0x10,0x95), (0x02,0xBF,0x98,0x40,0x98), (0x02,0xDF,0x9A,0xC0,0x98), # girder 3
(0x02,0x3F,0x79,0x20,0x7A), (0x02,0xBF,0x78,0x40,0x78), (0x02,0xEF,0x75,0xC0,0x77), # girder 4
(0x06,0xBF,0xD0,0x40,0xD0), (0x06,0xBF,0xB0,0x40,0xB0), # pellets
(0x06,0xBF,0x90,0x40,0x90), (0x06,0xBF,0x70,0x40,0x70), # pellets
(0x06,0xBF,0xF0,0x40,0xF0), # pellets
(0x00,0xD3,0xDA,0xD3,0xF6), (0x01,0x63,0xD8,0x63,0xF8), # ladders bottom - 1
(0x00,0x73,0xBB,0x73,0xD8), (0x00,0x2B,0xBA,0x2B,0xD6), # ladders 1 - 2
(0x01,0x53,0x98,0x53,0xBB), (0x00,0x83,0x98,0x83,0xBB), (0x00,0xD3,0x9A,0xD3,0xB6), # ladders 2 - 3
(0x00,0x2B,0x7A,0x2B,0x96), (0x00,0x5B,0x78,0x5B,0x98), (0x01,0xBB,0x78,0xBB,0x98), # ladders 3 - 4
(0x01,0x6B,0x54,0x6B,0x78), (0x00,0xD3,0x56,0xD3,0x76), (0,0,0,0,0) ) # ladders 4 - top

```

\$girderarrayindex = 29

Dit ziet er nu uiteindelijk zo uit:



En de uiteindelijke gegenereerde codes:

```
02 97 38 68 38
02 BF 54 10 54
02 DF 56 C0 55
02 FF F4 C0 F5
02 BF F8 00 F8
00 63 18 63 54
00 53 18 53 54
00 93 38 93 54
02 3F D7 10 D5
02 BF D8 40 D8
02 DF DA C0 D8
02 3F B9 20 BA
02 BF B8 40 B8
02 EF B5 C0 B7
02 3F 97 10 95
02 BF 98 40 98
02 DF 9A C0 98
02 3F 79 20 7A
02 BF 78 40 78
02 EF 75 C0 77
06 BF D0 40 D0
06 BF B0 40 B0
06 BF 90 40 90
06 BF 70 40 70
06 BF F0 40 F0
00 D3 DA D3 F6
01 63 D8 63 F8
```

```
00 73 B8 73 D8
00 2B BA 2B D6
01 53 98 53 B8
00 83 98 83 B8
00 D3 9A D3 B6
00 2B 7A 2B 96
00 5B 78 5B 98
01 BB 78 BB 98
01 6B 54 6B 78
00 D3 56 D3 76
AA
```

Nu versie v0.37 opgeslagen.

Dit hele blok invoegen in de Pacman versie. De definitie van het rivets board eindigt op #3B84. Dus de definitie van het barrels board begint nu op #3B85 (dat was op #3AE4).

Dan ook de verschillende verwijzingen aanpassen:

```
0CD4 11853B LD DE,#3B85 ; Load DE with start of table data for girders
245A 21853B LD HL,#3B85 ; load HL with start of table data for girders
```

Dat gaat nu goed. Alleen nog een heleboel sprites aanpassen en het kleurenpalette goedzetten voordat het er echt goed uitziet. Eerst de sprites aanpassen.

Nu versie v0.38 opgeslagen.

Dan nu het kleurenpalette aanpassen.

Dat lijkt voor het barrels board hier te gebeuren met overrides voor pies en rivets boards net er na.

```
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
```

Het palette van rivets (het vierde palette) is goed voor de achtergrond van de boards. Dus die gebruiken. Dat is een #01 voor zowel REG_PALETTE_A en REG_PALETTE_B.

Aanpassen:

```
OCA2 3601 LD (HL),#01 ; set palette bank selector
```

Ja dat is goed.

Nu ervoor zorgen dat de vlakke stukken op de girders goed verwerkt worden. Dus eigenlijk aan het begin van de routine die dat verzorgt checken op X-waarde: tussen twee waardes dan géén aanpassing doen. De twee grenzen: #3F/#40 en #BF/#C0.

De aanpassing voor de schuine girders wordt gedaan door de code van #2333 t/m 236D. Deze code wordt vanaf verschillende plaatsen aangeroepen (jumpman, fireballs, barrels) met in HL de X-positie en Y-positie van het object dat eventueel aangepast dient te worden en bij afronden bevat A de aangepaste Y-positie.

Dus zodanig aanpassen dat aan het begin de H-gelezen wordt en als deze tussen de twee waarden, dan A met L vullen en terug springen en anders de rest van de code doen.

Aanpassen:

2333 C34028 JP #2840 ; jump to additional code – jump to label **AAAA**

en:

```
2840 7C            LD     A,H        ; load A with X-position – label AAAA
2841 FE48           CP     #48        ; Is A < #48?
2843 3806           JR     C,#BBBB    ; yes, continue processing slanted girder – BBBB
2845 FEB8           CP     #B8        ; is A > #B8?
2847 3002           JR     NC,#BBBB   ; yes continue processing slanted girder - BBBB
;
2849 7D            LD     A,L        ; load A with unaltered Y-position
284A C9            RET              ; return
;
284B 3EOF           LD     A,#0F      ; load A with binary 00001111 – label BBBB
284D A4            AND    H        ; and with H. A now has between 0 and F
284E C33623        JP     #2336      ; jump back
```

Dat gaat in principe wel goed. Alleen de barrels rollen op de bovenste girder net iets te hoog. Dus de processing daar is niet goed. Komt omdat op de bovenste girder, net nadat Kong de barrel gerold heeft, de barrel 1 pixel aangepast wordt. Dat zit in de oorspronkelijke routine, maar wordt nu niet meer gedaan. Misschien op te lossen door de barrel 1 positie naar beneden al los te laten. Waar zit dat?

Er wordt #4E ingezet door code op #2DCB:

```
2DC9 13            INC    DE        ; DE := DE + 3 = DE now has kong hand sprite Y position
2DCA 1A            LD     A,(DE)    ; load A with kong hand Y position
2DCB DD7705        LD     (IX+#05),A ; store in barrel's Y position
```

Dat is dus afhankelijk van de positie van de hand van Kong. Op #2DCB even een call doen naar additionele code en daar dan de DE nog één verhogen en dan opslaan in de Y-positie van de barrel.

Aanpassen:

2DCB CDD42E CALL #2ED4 ; call additional code – call to label **AAAA**

en:

```
2ED4 3C           INC   A          ; increment A to adjust barrel deployment - AAAA  
2ED6 DD7705       LD    (#IX+05),A ; store in barrel's Y position  
2ED9 C9           RET              ; return
```

Gaat nu goed.

Nu versie v0.39 opgeslagen.

Gaat alleen niet goed bij de barrel roll-over. De barrels rollen over de rand, maar missen de onderliggende girder. De onderliggende girders verder naar buiten doortrekken.

Eerst aan de linkerkant: van X=#10 naar X=#08 gaan.

(0x02,0x3F,0xD7,0x10,0xD5) – 02 3F D7 10 D5 → 02 3F D7 08 D5
(0x02,0x3F,0x97,0x10,0x95) 02 3F 97 10 95 → 02 3F 97 08 95

Ja dat gaat goed.

Dan die aan de rechterkant: van X=

(0x02,0xEF,0xB5,0xC0,0xB7) – 02 EF B5 C0 B7 → 02 F7 B4 C0 B6 → 02 FF B4 C0 B6
(0x02,0xEF,0x75,0xC0,0x77) – 02 EF 75 C0 77 → 02 F7 74 C0 76 → 02 FF 74 C0 76

Ja dat gaat goed.

Nu versie v0.40 opgeslagen.

Er zijn met het barrel board nu nog de volgende zaken die gedaan moeten worden:

- Kong ladder wissen in verkeerde kleur
- Kleur blauwe barrels
- Ghost eerste keer omhoog met verkeerde sprite.
- Power pellets plaatsen
- Van de oil can een ghost box maken
- Geen vuur boven oil can maar ogen in de ghost box

Eerste de kluer van de blauwe barrels aanpassen. Dat komt doordat de blauwe kleur verplaatst is van slot #0C naar slot #01. Kijken waar de kleur van de blauwe barrels gezet wordt.

Dat lijkt hier te gebeuren:

```
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
```

Aanpassen:

```
2D0D DD360801 LD (IX+#08),#01 ; set sprite color to blue
```

Ja dat is nu goed.

Bovenaan de Kong ladder wordt een deel gewist. Dat is nu echter in het blauw. Dat is niet goed. Waar zit dat en hoe wordt dat gedaan?

Dat wordt hier gedaan:

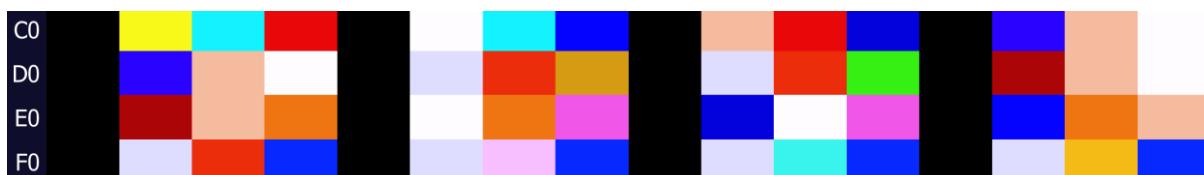
```
0FB5 21006A LD HL,#6A00 ; else load HL sprite RAM - used for blank space sprite
0FB8 3E4F LD A,#4F ; A := #4F = X position of this sprite
0FBA 0603 LD B,#03 ; For B = 1 to 3

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

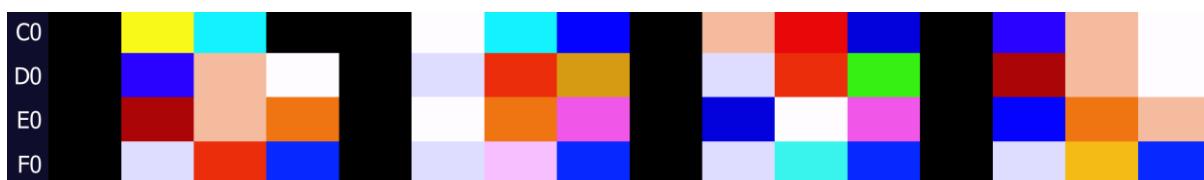
0FCB 79 LD A,C ; load A with screen number
0FCC EF RST #28 ; jump depending on the screen
```

Dit komt dus doordat de gebruikte color voor deze sprites (#0F blok) aangepast is naar de kleuren van het vierde spookje. Dus kijken of er een ander kleurblok is met zwart en eventueel de betreffende sprite aanpassen.

Die is er niet.



Maar van het eerste blok wordt alleen de gele kleur van pacman gebruikt. De derde kleur gebruiken voor het zwart.



In beide palettes aangepast (dus ook bij het knipperen bij eten van alle pellets).

Dan nu de waarde aanpassen in de code:

```
0FC1 3600 LD (HL),#00 ; set color to black
```

Ja, dat is nu goed.

Nu versie v0.41 opgeslagen.

Waarom gaat de ghost met de verkeerde sprites omhoog. Welke sprites en waar gezet? Watchpoint op change van de sprite value van de eerste ghost = #69D0+1 = #69D1.

Nee, dat is niet zo handig. Want wordt elk frame geupdate. Daarom kijken naar de waarde in de definitie = #6407.

```
Stopped at watchpoint 1 writing BD to 00006407 (PC=33DE)
Stopped at watchpoint 1 writing BD to 00006407 (PC=33DE)
Stopped at watchpoint 1 writing BE to 00006407 (PC=3414)
Stopped at watchpoint 1 writing BE to 00006407 (PC=33DE)
Stopped at watchpoint 1 writing BE to 00006407 (PC=33DE)
Stopped at watchpoint 1 writing BE to 00006407 (PC=33DE)
Stopped at watchpoint 1 writing BF to 00006407 (PC=3414)
Stopped at watchpoint 1 writing BD to 00006407 (PC=3424)
Stopped at watchpoint 1 writing BD to 00006407 (PC=33DE)
Stopped at watchpoint 1 writing BD to 00006407 (PC=33DE)
Stopped at watchpoint 1 writing BD to 00006407 (PC=33DE)
Stopped at watchpoint 1 writing BE to 00006407 (PC=3414)
Stopped at watchpoint 1 writing BE to 00006407 (PC=33DE)
Stopped at watchpoint 1 writing BE to 00006407 (PC=33DE)
Stopped at watchpoint 1 writing BE to 00006407 (PC=33DE)
Stopped at watchpoint 1 writing AE to 00006407 (PC=3F4C)
Stopped at watchpoint 1 writing AF to 00006407 (PC=3414)
Stopped at watchpoint 1 writing AD to 00006407 (PC=3424)
Stopped at watchpoint 1 writing AE to 00006407 (PC=3414)
Stopped at watchpoint 1 writing AF to 00006407 (PC=3414)
Stopped at watchpoint 1 writing AD to 00006407 (PC=3424)
Stopped at watchpoint 1 writing AE to 00006407 (PC=3414)
Stopped at watchpoint 1 writing AF to 00006407 (PC=3414)
Stopped at watchpoint 1 writing AD to 00006407 (PC=3424)
Stopped at watchpoint 1 writing AE to 00006407 (PC=3414)
Stopped at watchpoint 1 writing AF to 00006407 (PC=3414)
Stopped at watchpoint 1 writing AD to 00006407 (PC=3424)
Stopped at watchpoint 1 writing AE to 00006407 (PC=3414)
Stopped at watchpoint 1 writing AF to 00006407 (PC=3414)
Stopped at watchpoint 1 writing AD to 00006407 (PC=3424)
```

#BD en #BE zijn gespiegelde #3D en #3 E (en dat zijn spookjes sprites).

#AD, #AE en #AF zijn gespiegelde #2D, #2E en #2F (en dat zijn DK barrel roll sprites).

Wat is dat voor code op #3414 en #3424? En waarom dan ‘lopen’ door drie waardes? Dat is de fireball animatie op de trappen. Dat is op zich geen probleem. Het probleem is dat er een reset is naar de #AE. Dat wordt gedaan door #3F4C.

Dat is geen originele code, maar eigen code voor het aanpassen van de sprite value bij het opgaan van de trap.

```
339C C3463F JP #3F46 ; Jump to additional code – jump to label AAAA
```

En:

```
3F46 DD7E07 LD A,(IX+#07) ; load A with sprite value – label AAAA
3F49 0610 LD B,#10 ; load B with #10
3F4B 90 SUB B ; subtract #10 from value
3F4C DD7707 LD (IX+#07),A ; store new sprite value
3F4F DD360D08 LD (IX+#0D),#08 ; set direction to ascending ladder
3F53 C9 RET
```

Maar waarschijnlijk heeft de sprite bij het aanmaken van een nieuwe ghost niet de waarde van links/rechts kijken, maar al van het omhoog kijken. Daardoor wordt de waarde verkeerd aangepast.

Inderdaad bij het aanmaken van een ghost kijkt deze eerst omhoog.

Dus dat aanpassen. Waar wordt de ghost voor het eerst aangemaakt?

Het lijkt erop dat dit data is die initieel gebruikt wordt:

```
; the following is table data that gets copied to #6407  
; 05 is a loop varialbe  
; 1C loops value corresponds to total length of table
```

```
3DEC 3D 01 03 02
```

Aanpassen:

```
3DEC 4D 01 03 02
```

En ja hoor, dat klopt. Gaat nu goed. Ghost komt naar links kijkend uit de oil can. En gaat nu wel goed de trap op.

Nu versie v0.42 opgeslagen.

Dan de sprite van de oil can aanpassen. Maken als een soort ghost hok met daarin dan kleine versies van de ogen die door elkaar krioelen.

Sprites zijn aangepast. Nu nog naar beneden verplaatsen.

```
3DF4: 27 70 01 E0 00 00 ; initial data for fires on girders ?
```

Aanpassen naar:

```
3DF4: 27 70 01 F0 00 00 ; initial data for fires on girders ?
```

Ja dat is leuk. Dan nu de power pellets neerleggen. Eerst eventjes met de locaties van rivets.

Deze code is voor het plaatsen van de hamers op girders:

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

#3E0C daar ook naar aanpassen en dan gaan schuiven.

#762E + 5 * #20 = #76CE en #75FA – 6 * #20 = #753A → CE 76 3A 75.

Ja dat is goed zo.

Nu versie v0.43 opgeslagen.

Aantal pellets voor girders instellen. Snel geteld zijn dat er 73 (#49).

```
0ABF 3E49      LD A,#49      ; load A with nr of pellets on barrels
```

Ja dat klopt.

Het is vrij eenvoudig door de invincibility en tussenpozen tussen de barrels waardoor de spookjes niet echt een gevaar opleveren.

Kijken of de blue barrels sneller/vaker gegeven kunnen worden.

```
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
2C61 112000    LD DE,#0020   ; now check if all 5 fires are out
2C64 210064    LD HL,#6400   ; #6400 by 20's contain 1 if these fires exist
```

Niet helemaal duidelijk hoe dat werkt. Er is een blue barrel counter (#62B2) en die start op #32 en wordt steeds met 8 verlaagd: dus van #32, naar #2A en dan naar #22. Geen idee wat dat betekent.

Geprobeerd met SUB #01 en dan wordt elke barrel een blue barrel. Eventjes proberen met #03. Dat is wel erg snel druk en bijna niet te doen. Voorlopig op #04 houden.

Nu versie v0.44 opgeslagen.

Het is mogelijk om het board te beeindigen als pacman en dan wordt jumpman met de verkeerde kleuren weergegeven. Dat nog aanpassen.

Jumpman goedzetten bij finish van een board wordt hier gedaan:

```
1E72 3E80      LD A,#80      ; else load A with sprite facing right
1E74 324D69    LD (#694D),A   ; set mario sprite
1E77 C3851E    JP #1E85      ; jump ahead
```

Daar het goedzetten van de kleur aan toevoegen. De juiste kleur is #02.

Aanpassen:

```
1E77 C3D92E    JP #2ED9      ; jump to additional code – jump to label AAAA
```

en:

```
2ED9 3E02      LD A,#02      ; load A with color of jumpman - label AAAA
2EDB 324E69    LD (#694E),A   ; set jumpman color
2EDE C3851E    JP #1E85      ; jump ahead
```

Ja dat werkt goed voor barrels board. Voor rivets iets soortgelijks doen. Geen idee waar dat zit. Rivets clearen in debugger en wanneer de animatie start een watchpoint zetten op change van #694D.

Het lijkt hier te gebeuren:

```
18EE 3E50 LD A,#50 ; A := #50
18F0 324F69 LD (#694F),A ; store into mario sprite Y value
18F3 3E00 LD A,#00 ; A := 0
18F5 324D69 LD (#694D),A ; store into mario sprite value
18F8 3E9F LD A,#9F ; A := #9F
18FA 324C69 LD (#694C),A ; set mario sprite X value at #9F
```

Aanpassen:

```
18F5 C3E12E JP #2EE1 ; jump to additional code – jump to label AAAA
```

en:

```
2EE1 324D69 LD (#694D),A ; store into jumpman sprite value - label AAAA
2EE4 3E02 LD A,#02 ; load A with color of jumpman
2EE6 324E69 LD (#694E),A ; set jumpman color
2EE9 C3F818 JP #18F8 ; jump back
```

Ja dat gaat goed.

Nu versie v0.45 opgeslagen.

Het titelscherm is nu DONKEY KONG met blauwe lijnen. Kijken of daar een origineel karakter van Donkey Kong gebruikt kan worden. Maar dan wel in blauw. Dat is mooi met dan straks een pacman logo eroverheen in het geel.

De definitie van de letters begint op #3D08. En het karakter waarmee getekend wordt, wordt hier bepaald:

```
; draws DONKEY KONG logo to screen

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

07FA 3EB0 LD A,#B0 ; A := #B0 = code for girder on screen
```

Dus een nieuw karakter maken, met de oude definitie van een girder en dan dat gaan gebruiken.

Karakter op #DC gebruikt.

Ja dat is nu goed.

Ook de attract mode aanpassen zodat jumpman ook omhoog klimt, de power pellet pakt en pacman wordt.

De aangepaste attract mode:

```
80      - spring omhoog
F8 01   - loop naar rechts
10 01   - loop naar rechts
50 04   - klim omhoog
1B 02   - loop naar links
10 82   - spring naar links (over power pellet)
50 02   - loop naar links
50 01   - loop naar rechts (en pak power pellet)
C0 02   - loop naar links (pak de meest linkse pellet)
70 01   - loop naar rechts
24 02   - loop naar links
70 04   - klim omhoog en wacht bovenaan trap
30 01   - loop naar rechts
10 81   - spring naar rechts (over barrels in eerste run na startup)
FF 02   - loop naar links (en val van girder als niet eerder door barrel of spook geraakt)
```

Nu versie v0.46 opgeslagen.

Nu gaan werken aan het elevators board.

Eerst de basis van het board terugzetten en daarnaar verwijzen en starten met board 03.

Invoegen op #3C3F.

```
3BE5: 00 63 18 63 58 ; kong's ladder (right)
3BEF: 00 53 18 53 58 ; kong's ladder (left)
3C12: 00 93 38 93 58 ; ladder leading up to girl
3C21: 02 97 38 68 38 ; girder girl is on
3C26: 02 B7 58 10 58 ; kong's girder
3C80: 02 27 E8 10 E8 ; girder where mario starts
3C85: 02 EF F8 10 F8 ; long bottom girder (mario dies if he gets that low)
3C8A: AA          ; end code
```

En de verwijzingen aanpassen:

```
0CFA  113F3C      LD    DE,#3C3F      ; load DE with start of table data for the elevators
2468  213F3C      LD    HL,#3C3F      ; load HL with start of table data for elevators
```

Kijken of we de springs eerder kunnen laten vallen.

De overgang van bouncer naar falling spring wordt hier gedaan:

```

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

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

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

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

```

Op #2E54 wordt een check gedaan op de X-waarde: zoalng kleiner dan B7, dan door blijven springen, maar wanneer groter dan overgaan op vallen (IX+#0D wordt dan op 4 gezet).

Als die waarde verlaagd wordt, gaat de bouncer waarschijnlijk eerder vallen. Testen bij welke waarden.

Testen met #77 voor een bounce eerder vallen. Ja op 77 houden. Gaat goed.

De Kong girder langer maken:

3C26: 02 EF 58 10 58 ; kong's girder

De rechter ladder naar rechts verplaatsen. De definitie van de ladders zit hier:

```

3E64: 37 F4
3E66: 37 C0
3E68: 37 8C

```

```

3E6A: 77 70
3E6C: 77 A4
3E6E: 77 D8

```

Twee extra girders om symmetrie van het veld te testen:

02 EF E8 D8 E8 ; girder at jumpman start height but at right side
 02 B0 D0 48 D0 ; central girder section

De rechter elevator zit goed op positie C7.

Nu zorgen dat de rest van de rechter elevator getekend wordt en gaat werken.

De kabelpositie wordt hier bepaald:

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

Aanpassen:

```
0D2D 21CD74 LD HL,#74CD ; load HL with screen RAM location for right side cable
```

De eindpunten van de elevators worden hier gedefinieerd:

```
1121 37 45 0F 60  
1125 37 45 8F F7  
1129 77 45 0F 60  
112D 77 45 8F F7
```

Aanpassen #77 naar #C7.

En de aanpassing van de elevators zodat ze weer terugkomen op de opgeschoven rechter positie zit hier:

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

Aanpassen #77 naar #C7.

En dan nog aanpassen dat jumpman op de rechter elevator meebeveegt. Dat zit hier:

```
275C FE6C CP #6C ; else is position < left edge of second elevator?  
275E DA6627 JP C,#2766 ; yes, jump ahead  
  
2761 FE83 CP #83 ; else is position < right edge of second elevator ?  
2763 DA8727 JP C,#2787 ; yes, jump ahead for second elevator checks
```

#C7 - #77 = #50 → #6C + #50 = #BC → #83 + #50 = #D3. Aanpassen.

Ja, dat gaat ook goed.

Nu versie v0.47 opgeslagen.

En dan nu de sprites van de eindpunten en de elevators zelf aanpassen naar een goede stijl.

En de sprites van de springs aanpassen naar De boze,bange ghosts en dan de kleur aanpassen. Is als het goed is kleurcode #03.

Waar wordt de kleur gezet? Lijkt nergens te gebeuren. #IY+#02 wordt niet gebruikt. Staat op default van #00 en dat is de kleur. Dus we zouden kunnen experimenteren met het zetten van de spring color naar een willekeurige kleur. Om te testen met de verschillende kleuren.

```

; deploy new bouncer

2EA8 AF      XOR   A      ; A := 0
2EA9 329663 LD    (#6396),A ; reset bouncer release flag
2EBA DD360550 LD    (IX+#05),#50 ; set bouncer's Y position to #50
2EB6 DD360D01 LD    (IX+#0D),#01 ; set value to sprite bouncing across, not down
2EBA C05700 CALL  #0057 ; load A with random number
2EBD E60F AND   #0F ; mask bits, result is between 0 and #F
2EBF C6F8 ADD   A,#F8 ; add #F8 = result is now between #F8 and #07
2EC1 DD7703 LD    (IX+#03),A ; store A into initial X position for bouncer sprite
2EC4 DD360001 LD    (IX+#00),#01 ; set sprite as active
2EC8 21AA39 LD    HL,#39AA ; values #39 and #AA to be inserted below. #39AA is the start of table data for Y offsets to add for each movement
2ECB D0750E LD    (IX+#0E),L ;
2ECF D0740F LD    (IX+#0F),H ; store HL into +E and +F
2ED1 C3782E JP    #2E78 ; jump back

```

Dus hier ergens tussenvoegen:

```
2ED1 C3EC2E      JP    #2EEC ; jump to additional code – jump to label AAAA
```

en:

```
2EEC 3E03      LD    A,#03 ; load A with color – label AAAA
2EEE FD7702     LD    (IY+#02),A ; set sprite color
2EF1 C3782E     JP    #2E78 ; jump back
```

Nee niet de blauwe kleur (#03), maar de kleur van een spookje nemen. Bijvoorbeeld van het rode spookje is #0C.

Dat gaat nu goed. Ziet er erg leuk uit.

Nu versie v0.48 opgeslagen.

Dan nu additionele girders aan gaan maken.

De structuur links:

```
00 23 88 23 C0 ; top left ladder
00 1B C0 1B E8 ; bottom left ladder
02 27 88 10 88 ; top girder left side
02 27 C0 10 C0 ; middle girder left side
```

En dat repliceren aan de rechterkant in spiegelbeeld:

```
00 DB 88 DB C0 ; top right ladder
00 E3 C0 E3 E8 ; bottom right ladder
02 EF 88 D8 88 ; top girder right side
02 EF C0 D8 C0 ; middle girder right side
```

En dan de ladder naar de top:

02 E3 58 E3 88 ; ladder to kongs girder

Lange girders in middenstuk:

02 B0 E0 48 E0 ; center bottom girder
02 B0 C0 48 C0 ; center second girder
02 B0 A0 48 A0 ; center third girder
02 B0 80 48 80 ; center top girder

En dan de pellets daarop:

06 B0 D8 48 D8 ; center bottom pellets
06 B0 B8 48 B8 ; center second layer pellets
06 B0 98 48 98 ; center third layer pellets
06 B0 78 48 78 ; center top pellets

De onderste ladders:

00 63 C0 63 E0 ; center bottom left ladder
00 9B C0 9B E0 ; center bottom right ladder

De bovenste ladders:

00 63 80 63 A0 ; center top left ladder
00 9B 80 9B A0 ; center top right ladder

De middelste ladders:

00 73 A0 73 C0 ; center top left ladder
00 8B A0 8B C0 ; center top right ladder

Dat is goed. Gaan test spelen, maar eerst de twee spookjes goed plaatsen op het middengedeelte en eventueel de bonus items verplaatsen.

Dat zit hier:

```

; set up the 2 fireballs

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

; set up 2nd fireball

1101 DD362001 LD    (IX+#20),#01  ; set fire active
1105 DD3623EB LD    (IX+#23),#EB  ; set fire X position
1109 DD362EEB LD    (IX+#2E),#EB  ; set fire X position
110D DD362560 LD    (IX+#25),#60  ; set fire Y position
1111 DD362F60 LD    (IX+#2F),#60  ; set fire Y position

```

De eerste starten op X=#58 en Y=#78. De tweede starten op X=#A0 en Y=#D8.

Even de pellet counter max op #32 (=50) zetten.

```

0ACB FE03      CP    #03      ; is this elevators board?
0ACD 2004      JR    NZ,#YYYY  ; no, skip next steps
0ACF 3E32      LD    A,#32    ; load A with nr of pellets on elevators
0AD1 1802      JR    AAAA    ; jump ahead – jump to label AAAA

```

Nu versie v0.49 opgeslagen.

Nog aanpassen:

- Positie bonus items
- Snelheid elevators van L=05 nemen.
- Deploy frequentie springs van L=05.

Eerste de bonus items goedzetten.

De appel (was paraplu) staat al goed bovenin. De andere twee verplaatsen. De hat naar beneden verplaatsen (#7D, D8). De purse een heel stuk naar beneden en beetje naar rechts verplaatsen (#E8,B8).

```

; bonus items for elevators

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

```

Nu versie v0.50 opgeslagen.

In principe de bedoeling dat alles werkt op basis van internal difficulty 5. Dat is al aangepast. Even checken of dat inderdaad zo is.

Internal difficulty is #6380. Ja dat is zo, de internal difficulty staat de hele tijd op #05.

Maar dat betekent dat de snelheid van de elevators niet afhangt van de internal difficulty, maar wellicht van het level? Waar wordt dat bepaald?

Is een aparte setting voor alleen level L=01:

```
2705 3A2962 LD A,(#6229) ; else load A with level number
2708 3D DEC A ; decrement and check for zero
2709 3A1A60 LD A,(FrameCounter) ; load A with this clock counts down from #FF to 00 over and over...
270C C21A27 JP NZ,#271A ; if level <> 1 then jump ahead
```

Dus de trage elevators in level L=01 te voorkomen door onvoorwaardelijk te jumpen op #270C.

Ja dat gaat goed.

Nu nog kijken voor de deploy snelheid van de springs. Waar hangt dat van af? Of dat zo laten, waardoor het steeds moeilijker wordt? Ja laten we dat maar doen.

Door te spelen blijkt dat twee spookjes in het middenste gedeelte wel heel erg moeilijk is. Daarom het rode spookje verplaatsen naar de girder links.

```
; set up the 2 fireballs

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

; set up 2nd fireball

1101 DD362001 LD (IX+#20),#01 ; set fire active
1105 DD3623EB LD (IX+#23),#EB ; set fire X position
1109 DD362EEB LD (IX+#2E),#EB ; set fire X position
110D DD362560 LD (IX+#25),#60 ; set fire Y position
1111 DD362F60 LD (IX+#2F),#60 ; set fire Y position
```

De eerste starten op X=#28 en Y=#80. De tweede starten op X=#A0 en Y=#D8.

Ja dat is een stuk beter. Nog steeds lastig genoeg met de vallende springs. En wordt in latere levels nog lastiger omdat ze dan sneller achter elkaar komen. En ze vallen soms over een trap. En je kunt niet meer zondermeer terug aan de linkerkant want daar zit dan nu het rode spookje.

Nu versie v0.51 opgeslagen.

Nu nog een laatste board maken gebaseerd op het pies board. Maar dat kan niet zondermeer omdat er géén ruimte vrij is voor een complete board definitie. Er moet dus e.e.a. vrijgegeven worden.

Wellicht de pies zelf niet meer doen, maar wellicht nog wel het bewegen van de conveyor belts?
Komt er dan voldoende vrij.

Eerst stukken vrijgeven voor de afhandeling van de pies.

In de main wordt op #1992 een stuk code voor de afhandeling van pies aangeroepen. Dat is het stuk code van #24EA t/m #25F1.

De aanroep in NOP's veranderen en het hele stuk vrijgeven. Eens kijken hoeveel dat dan is en of het voldoende is voor de pies board definitie (inclusief pellets). Ja dat is voldoende ruimte voor het board. Dan kunnen de conveyors gewoon blijven draaien. Dat is wel leuk. En dan de twee kort girders aan elkaar maken tot één geheel. En dan wellicht tunnel zodat je naar de andere kant van het scherm kan. En ook zo maken dat je pas klaar bent als je één van de ladders beklimt en dus langs kong moet. Dit is eerder gedaan in DK Wizardry, je moet ook terug de ladders op. Wellicht ook pellets op de kong girder aan beide kanten, dan moet je wel een keer terug (of langs kong zien te komen).

De originele pies definitie grotendeels overnemen en een girder verlengen, de kruisen verwijderen en de pellets toevoegen).

```
06 EF F0 10 F0 ; pellets bottom girder
06 EF A0 10 A0 ; pellets second girder
06 EF C8 10 C8 ; pellets bottom conveyor
06 EF 50 10 50 ; pellets top conveyor girder
06 F7 78 88 78 ; pellets top right conveyor next to oil can
06 77 78 08 78 ; top left conveyor next to oil can
;
00 63 18 63 58 ; kong's ladder (right)
00 63 80 63 A8 ; center ladder to left of oil can fire
00 63 D0 63 F8 ; bottom level ladder #2 of 4
00 53 18 53 58 ; kong's ladder (left)
00 53 A8 53 D0 ; ladder under the hat
00 9B 80 9B A8 ; center ladder to right of oil can fire
00 9B D0 9B F8 ; bottom level ladder #3 of 4
01 23 58 23 80 ; top broken ladder left side
01 DB 58 DB 80 ; top broken ladder right side
00 2B 80 2B A8 ; ladder on left platform with hammer
00 D3 80 D3 A8 ; ladder on right platform with umbrella
00 A3 A8 A3 D0 ; ladder to right of bottom hammer
00 2B D0 2B F8 ; bottom level ladder #1 of 4
00 D3 D0 D3 F8 ; bottom level ladder #4 of 4
00 93 38 93 58 ; ladder leading to girl
02 97 38 68 38 ; girder where girl sits
03 EF 58 10 58 ; top conveyor girder
03 F7 80 88 80 ; top right conveyor next to oil can
03 77 80 08 80 ; top left conveyor next to oil can
02 A7 A8 50 A8 ; center ledge
02 E7 A8 B8 A8 ; right center ledge
02 3F A8 18 A8 ; left center ledge (has hammer)
```

```
02 EF A8 10 A8 ; second level girder  
03 EF D0 10 D0 ; main lower conveyor girder (has hammer)  
02 EF F8 10 F8 ; bottom level girder  
AA ; end code
```

De oorspronkelijke definitie zat op #3B5D t/m #3BE4.

Plaatsen in het stuk startend op #24EA. En de verwijzingen aanpassen.

```
0CDF 11EA24 LD DE,#24EA ; load DE with start of table data for conveyors  
2461 21EA24 LD HL,#24EA ; load HL with start of table data for conveyors
```

Ja gaat goed.

Nu versie v0.52 opgeslagen.

Nog aanpassen:

- Palette, zodat juiste kleurgebruik.
- Graphics van de conveyor.
- Graphics en kleuren van de draaipunten.
- Graphics van de bewegende ladders.
- Het ghosthok een girder naar beneden en de ogen daarin plaatsen.
- Die girder splitsen in twee kleinere.
- De ogen naar beneden in het ghosthok.
- Spawning van de ghosts ook naar beneden.
- Ghosts niet omhoog laten gaan naar bovenste conveyor
- Power pellets plaatsen
- Jumpman ook op de Kong conveyor laten lopen – einde bij Pauline
- Jumpman ook weer laten afdalen vanaf de Kong conveyor
- Check aantal pellets
- Eventueel tunnel functionaliteit

Het palette van het pies board wordt hier gezet:

```
0CE2 21867D LD HL,REG_PALETTE_A ; load HL with palette bank selector  
0CE5 3601 LD (HL),#01 ; set palette bank selector  
0CE7 23 INC HL ;  
0CE8 3600 LD (HL),#00 ; clear palette bank selector
```

Allebei op #01 zetten. Dat gaat goed.

Nu versie v0.53 opgeslagen.

De lange girder splitsen in twee girders.

02 EF A8 10 A8 ; second level girder

02 EF A8 88 A8

02 70 A8 10 A8

~~02 A7 A8 50 A8 ; center ledge~~

~~02 E7 A8 B8 A8 ; right center ledge~~

~~02 3F A8 18 A8 ; left center ledge (has hammer)~~

Dat is gedaan. De ghost box naar beneden verplaatsen. Dat is de oil can naar beneden verplaatsen.

Dat zit hier:

3E04 7F 49 0C 88 ; oil can for conveyors

De #88 is de Y-positie. Moet naar beneden, naar #AD.

Daarnaast de ogen ook naar beneden verplaatsen. En zelfs nog iets verder omdat het niet een fire boven de oil can is, maar ogen in de ghost box. Dat zit hier:

3DFA: 7F 40 01 78 02 00 ; initial data for conveyors to release a fire

De #78 is de Y-positie. Moet naar beneden, naar #AD.

En het lanceren van de ghosts moet ook naar beneden verplaatst worden.

Dat wordt hier geregeld:

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

Aanpassen:

80 7B 78 76 74 73 72 71 70 70 6F 6F 6F 6F 70 70 71 72 73 74 75 76 77 78 AA
A0 9B 98 96 94 93 92 91 90 90 8F 8F 8F 90 90 91 92 93 94 95 96 97 98 AA
A8 A3 A0 9E 9C 9B 9A 99 98 98 97 97 98 98 99 9A 9B 9C 9D 9E 9F A0 AA

De kleuren van de draaipunten aanpassen:

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

Kleur #00 aanpassen naar kleur #03.

Ghosts niet naar boven laten gaan op conveyors. Beetje hetzelfde als niet naar beneden laten gaan vanaf de bovenste girder.

Het stukje code dat regelt dat ze niet meer naar beneden kunnen is van #33A1 t/m #33AC. Deze code wordt maar vanaf één plek aangeroepen: vanaf #334A (bij de beslissing om een ladder af te dalen).

```
33A1 3E07 LD A,#07 ; \ Return if immediately we are on rivets, fireballs do not get stuck on the top in this case
33A3 F7 RST #30 ; /
33A4 DD7E0F LD A,(IX+#0F) ; \ Return if Y-position is >= 59 (i.e., fireball is not on the top girder)
33A7 FE59 CP #59 ; |
33A9 D0 RET NC ; /
33AA 33 INC SP ; \ Else return to higher subroutine. This prevents fireballs from coming down on conveyors & girders once
33AB 33 INC SP ; | they reach the top level.
33AC C9 RET ; /
; Else firefox is not on a ladder, but will mount one if permitted to do so
334A CDA133 CALL #33A1 ; Return without taking ladder if fireball is on the top girder and the screen is not rivets
334D DD7E0F LD A,(IX+#0F) ; \ D := Y-position of bottom of fireball
3350 C608 ADD A,#08 ; |
```

Deze code checkt of de ladder beslissing onderhevig is aan een uitzondering. Als een uitzondering dan wordt er naar een hogere routine teruggesprongen.

We kunnen in ieder geval eventueel een uitzondering maken dat ze wel weer naar beneden mogen. En dan kijken hoe ze zich verspreiden en of het board dan te spelen is.

Daartoe het nummer op 33A1 aanpassen zodat ook voor de pies meteen teruggesprongen wordt. De waarde #07 betekent dat 0111 bij rivets weggesprongen wordt. Het zou kunnen dat met 0101 er ook bij pies weggesprongen wordt. Dus dat is dan #05. Eens eventjes testen. Ja dat werkt. Maar niet helemaal goed. De ghosts gaan ook de ladder naar Pauline op en zelfs de ladders van Kong op. Dat is niet helemaal de bedoeling. Dit even terugdraaien.

Anders maken. Alleen de Kong conveyor behouden (met pulleys). De andere weer gewone girders maken. Zorgen dat de spookjes niet naar de Kong conveyor gaan. Dan alle vier in het onderste gedeelte. Dat is lastig, maar power pellets te gebruiken en tunnel om te ontsnappen.

Eerst maar de pulleys weglaten, behalve die va de Kong conveyor.

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

Dat betekent dat alleen de eerste 2 getekend hoeven te worden. Deze datababel wordt hier gebruikt:

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

Dus minder kopiëren door niet #18, maar slechts #08 te kopiëren. Ja dat gaat goed.

En de conveyors te tekenen als girders.

03 EF 58 10 58 ; top conveyor girder
03 F7 80 88 80 ; top right conveyor next to oil can
03 77 80 08 80 ; top left conveyor next to oil can
03 EF D0 10 D0 ; main lower conveyor girder (has hammer)

Allemaal #02 als type geven.

Wel de pulleys meer naar buiten plaatsen:

3E24: 14 50 00 5C
3E28: EA D0 00 5C

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

De twee korte conveyors samenvoegen tot één lange girder:

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

Aanpassen naar:

02 F7 80 08 80

En de pellets van twee korte conveyors en de lange eronder omwisselen.

06 EF F0 10 F0 ; pellets bottom girder
06 EF A0 10 A0 ; pellets second girder → 06 EF 78 10 78
06 EF C8 10 C8 ; pellets bottom conveyor
06 EF 50 10 50 ; pellets top conveyor girder
06 F7 78 88 78 ; pellets top right conveyor next to oil can → 06 F7 A0 88 A0
06 77 78 08 78 ; top left conveyor next to oil can → 06 77 A0 08 A0

Nu versie v0.54 opgeslagen.

En dan nu kijken dat alle handelingen voor de niet Kong conveyors ook verwijderd worden. En daarna kijken hoe een tunnel te maken.

De afhandeling van de conveyors zit hier:

```
25F5 CD0226 CALL #2602 ; handle top conveyor and pulleys
25F8 CD2F26 CALL #262F ; handle middle conveyor and pulleys
25FB CD7926 CALL #2679 ; handle lower conveyor and pulleys
25FE CDD32A CALL #2AD3 ; handle mario's different speeds when on a conveyor
2601 C9 RET ; return
```

Dus afhandeling overslaan door #25F8 en #25FB over te slaan. De #25F5 wel laten staan omdat dat de Kong conveyor is.

Ja dat is goed. Meteen geen conveyors meer op de onderste girders. Kong girder waarschijnlijk nog wel. Kong beweegt nog en de pulleys draaien ook nog.

De betreffende blokken vrijgeven: #262F t/m #2678 en #2679 t/m #26A5.

Laat de rest maar bestaan.

Random kant spawnen van de fireballs:

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

Ja dat gaat goed.

Nu versie v0.55 opgeslagen.

Nu kijken dat jumpman pas klaar is wanneer helemaal bij Pauline. Is al eerder gedaan in DK Wizardry. Daarvandaan kopiëren.

De CP op #1E66 aanpassen naar #31. Ja nu weer eindigen bij Pauline. Dat wel weer aanpassen zodat jumpman bij Pauline gezet wordt. De settings daarvoor eventjes bekijken met de debugger. De jumpman sprite is: #694C.

#694C (X-pos) = 93, #694D (sprite nr) = 00, #694E (color) = 02, #694F (Y pos) = 30.

De X-pos en de sprite nr moeten aangepast worden bij finish aan de linkerkant.

Dit is de code die dit afhandelt:

```

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

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

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

1E69 3A0362 LD A,(#6203) ; else load A with mario's X position
1E6C 17 RLA ; on left or right side of screen?

1E6D 3E00 LD A,#00 ; load A with #00. sprite for facing left
1E6F DA741E JP C,#1E74 ; if on left side, skip next step

1E72 3E80 LD A,#80 ; else load A with sprite facing right
1E74 324D69 LD (#694D),A ; set mario sprite
1E77 C3851E JP #1E85 ; jump ahead

```

Klopt niet. Comments suggereren dat #1E72 geskipt wordt wanneer aan de linker kant, maar volgens mij moet dat juist zijn dat dat stuk wel uitgevoerd wordt. Even testen met debugger. Ja hoor wanneer links dan wordt 1E72 wel uitgevoerd en rechts niet.

Dus #1E72 wordt alleen uitgevoerd wanneer aan de linkerkant. Dus op #1E6F een conditionele jump doen naar additionele code waar de juiste aanpassingen gedaan worden. Jump doen als links. Maar commenst kloppen niet , dus juist jumpen als NC.

~~Wellicht hier ook alvast vijf NOPs reserveren zodat daar later een waarde in een memory veld gezet kan worden zodat in rivets een alternatief muziekje gespeeld kan worden.~~

```

1E6F D22F26 JP C,#262F ; jump to additional code – jump to label AAAA
1E72 3E00 LD A,#00 ; else load A with sprite facing left

```

en:

```

262F 3E93 LD A,#93 ; load A with new X position – label AAAA
2631 324C69 LD (#694C),A ; set new X position
;
2634 0000 NOP, NOP
134B 000000 NOP, NOP, NOP
;
2634 3E00 LD A,#00 ; load A with sprite facing left
2636 C3741E JP #1E74 ; jump back

```

Ja dat gaat nu allemaal goed.

Kijken oplossen retractable ladders. Lijkt erop dat de retractable ladders adjustment gedaan wordt op basis van X-waarde kleiner dan bepaalde waarde. Maar daardoor ook doen wanneer jumpman op de conveyor van DK staat.

Dat lijkt hier getest te worden:

```

2243 3A0562 LD A,(#6205) ; load mario's Y position
2246 FE7A CP #7A ; is mario on the top pie tray level or above?
2248 D25722 JP NC,#2257 ; no, skip ahead and return to higher sub

```

Maar kijken op welke waarde de Y-positie dan staat wanneer jumpman meerijdt op de bovenste conveyor (DK's) conveyor. Dus dit alleen doen als jumpman hoger is dan #7A. De onderkant van de retractable ladder zit op de girder van #78. Dus jumpman moet eerst een stukje klimmen. Maar op de conveyor van DK zit jumpman op Y-positie #50. Dus dan niet doen.

Aanpassen:

```

2246 C33926 JP #2639 ; jump to additional code – jump to label AAAA
2249 0000 NOP. NOP

```

en:

```

2639 FE51 CP #51 ; is jumpman at #50 or lower ? – label AAAA
263B DA5722 JP C,#2257 ; yes, do not process retractable ladder
;
263E FE7A CP #7A ; is jumpman at 7A or higher?
2640 D25722 JP NC,2257 ; yes, do not process retractable ladder
;
2643 C34B22 JP #224B ; jump back

```

Dat gaat goed.

Ga je af door tegen Donkey Kong aan te lopen? Testen met invincibility uit.

Heel lastig. De twee gebroken ladders heel maken:

3BC1: 00 23 A8 23 D0 ; broken ladder left
 3BC6: 00 DB A8 DB D0 ; broken ladder right

Je gaat dus niet af door tegen Donkey Kong aan te lopen. Kijken hoe dat opgelost is in Donkey Kong Reverse.

De X-positie van Kong wordt vastgelegd in variabele op #6910.

De volgende code handelt de collision detection af:

```

; called from main routine at #19B3
; checks for collisions with hostiles sprites

2808 FD210062 LD IY,#6200 ; load IY with start of mario sprite
280C 3A0562 LD A,(#6205) ; load A with mario's Y position
280F 4F LD C,A ; copy to C
2810 210704 LD HL,#0407 ; H := 4, L := 7
2813 CD6F28 CALL #286F ; checks for collisions based on the screen. A := 1 if collision, otherwise zero
2816 A7 AND A ; was there a collision ?
2817 C8 RET Z ; no, return

; mario collided with hostile sprite

2818 3D DEC A ; else A := 0
2819 320062 LD (#6200),A ; store into mario life indicator, mario is dead
281C C9 RET ; return

```

Dit uitbreiden met een check wanneer in conveyors screen en de y-coordinaat van jumpman is tussen de conveyor en de gilder van Pauline (dat is de hoogte van Donkey Kong) en de x-coordinaat van jumpman is tussen linker en rechterkant van Donkey Kong. Dit allemaal proefondervindelijk achterhalen welke coordinaten dat zijn.

Start y-coördinaat van jumpman naast Pauline is #31 en waarschijnlijk is de y-coördinaat van de conveyor dan #51.

Aanpassen:

```

2816 A7 AND A ; was there a collision ?
2817 CA4626 JP Z,#2646 ; no, jump ahead - jump to label AAAA
281A C36526 JP #2665 ; yes, jump ahead, handle - jump to label BBBB

```

en:

; check jumpman collided with Donkey Kong in conveyors screen

```

2646 3A2762 LD A,(#6227) ; load A with screen number - label AAAA
2649 FE02 CP #02 ; are we on the conveyors?
264B C0 RET NZ ; no, return no collision

264C 3A0562 LD A,(#6205) ; load A with jumpman's y position
264F FE31 CP #31 ; jumpman higher than Donkey Kong ?
2651 D8 RET C ; yes, return no collision
2652 FE51 CP #51 ; jumpman lower than Donkey Kong ?
2654 D0 RET NC ; yes, no collision

2655 3A1069 LD A,(#6910) ; load A with Donkey Kongs x position
2658 D619 SUB #13 ; calculate left side of Donkey Kong
265A 47 LD B,A ; copy to B, left side of Donkey Kong
265B C630 ADD A,#2A ; calculate right side of Donkey Kong
265D 4F LD C,A ; copy to C, right side of Donkey Kong
265E 3A0362 LD A,(#6203) ; load A with x position of jumpman
2661 B8 CP B ; jumpman left of Donkey Kong ?
2662 D8 RET C ; yes, no collision
2663 B9 CP C ; jumpman right of Donkey Kong ?
2664 D0 RET NC ; yes, no collision

```

; jumpman collided with hostile sprite

```

2665 3E00 LD A,#00 ; A := 0 - label BBBB
2667 320062 LD (#6200),A ; store into jumpman life indicator, means dead
266A C9 RET ; return

```

Interessant als dit werkt. Dan ook nog testen of de collisions met andere enemies (barrels, fireballs, springs nog goed werken).

Gaat goed. Maar de hitbox van DK is zo wel erg groot. Aan beide kanten kleiner maken.

```
1372 D613    SUB    #13          ; calculate left side of Donkey Kong  
1375 C62A    ADD     A,#2A        ; calculate right side of Donkey Kong
```

Ja dat gaat goed.

Wel zou het logisch zijn (en soms gewenst), als je ook de retractable ladder af kan gaan. Dit is al eerder geregeld voor DK Reverse. De retractable ladder functionaliteit zorgt alleen voor verschuiving van jumpman. Maar de onderliggende ladder is een broken ladder. Daar kan jumpman niet opgaan vanaf de bovenkant. Dat betekent dat de code hiervoor aangepast moet worden.

```
; else mario at top of ladder  
  
1B31 7E      LD     A,(HL)       ; load A with broken ladder indicator  
1B32 A7      AND    A           ; is this ladder broken?  
1B33 C0      RET    NZ          ; yes, return. we can't go down broken ladders
```

Op regel 1B33 niet meteen een return doen als broken ladder, maar springen naar stuk code waar die test gedaan wordt (en daarvandaan dan eventueel returnen) als juiste ladder uitgeschoven is, niet returnen, maar jump naar #1B34.

Aanpassen:

```
1B31 C36B26    JP    #266B        ; jump to additional code – jump to label AAAA
```

en:

```
266B 3A2762    LD    A,(#6227)    ; load A with screen number – label AAAA  
266E FE02      CP    #02          ; are we on the conveyors?  
2670 2019      JR    NZ,#13C1    ; no, jump ahead – jump to label BBBB  
;  
2672 3A0562    LD    A,(#6205)    ; load A with jumpman's Y position  
2675 FE50      CP    #50          ; check jumpman at upper conveyor  
2677 2012      JR    NZ,#13C1    ; no, jump ahead – jump to label BBBB  
;  
2679 3A0362    LD    A,(#6203)    ; load A with X position of jumpman  
267C FE80      CP    #80          ; jumpman on left side of screen ?  
267E 3A8862    LD    A,(#6288)    ; load A with right retractable ladder status  
2681 3003      JR    NC,13BC    ; no, skip next step – jump to label CCCC  
;  
2683 3A8062    LD    A,(#6280)    ; load A with left retractable ladder status  
;  
; check if ladder is all the way up  
;  
2685 FE00      CP    #00          ; ladder is all the way up? – label CCCC  
2687 2804      JR    Z,#13C4    ; yes, - take ladder – jump to label DDDD  
2689 C9        RET   NZ          ; no, return, we can't go down if ladder not up  
;  
268A 7E        LD    A,(HL)      ; load A with broken ladder indicator – label BBBB  
268B A7        AND   A           ; is this ladder broken?  
268C C0        RET   NZ          ; yes, return, we can't go down broken ladders  
268D C3341B    JP    #1B34        ; jump back – label DDDD
```

Ja dat gaat allemaal goed.

Nu versie v0.56 opgeslagen.

Plaatsen power pellets:

3E10 OF 76 19 76

Ja die staan goed.

Er voor zorgen dat een ghost niet naar de kong conveyor kan gaan. Ze mogen de retractable ladders (zijn gebroken ladders) niet opgaan. Het opgaan van een ladder zit hier:

```
3359 CD6E23 CALL #236E ; Check for ladders nearby, return if none, else A := 0 if at bottom of ladder, A := 1 if at top
335C A7 AND A ; \ Jump if there is a ladder nearby to go up
335D CA9933 JP Z,#3399 ; /
```

Dus hier springen naar additionele code en daar checken of pies board èn op de bovenste girder. Zo ja dan een return doen (geen ladder beklimmen) en anders jump naar #3399 (wel ladder beklimmen).

335D CA9126 JP Z,#2691 ; jump to additional code – jump to label AAAA

en:

```
2691 3A2762 LD A,(#6227) ; load A with screen number – label AAAA
2694 FE02 CP #02 ; is this the pies board?
2696 2006 JR NZ,#BBBB ; no, skip additional check – jump to BBBB
;
2698 DD7EOF LD A,(IX+#0F) ; load A with Y-pos of fireball
269B FE78 CP #78 ; is Y-pos 78?
269D C8 RET Z ; yes, return without processing ladder
;
269E C39933 JP #3399 ; jump back, process the ladder - BBBB
```

Ja dat gaat nu goed.

Nu versie v0.57 opgeslagen.

Kijken of er tunnels gedefinieerd kunnen worden.

Waar zit de check voor de buitenkanten van de girders?

In ieder geval niet verder naar links dan #16 en niet verder naar rechts dan #E9.

Er is een routine die iets checkt met de positie van jumpman aan de grenzen:

```

241F 110001 LD DE,#0100 ; DE:= #0100
2422 3A0362 LD A,(#6203) ; load A with Mario's X position
2425 FE16 CP #16 ; is this greater than #16 ?
2427 D8 RET C ; yes, return

2428 15 DEC D ; no,
2429 1C INC E ; DE := #0001
242A FEEA CP #EA ; is Mario's position > #EA ?|
242C D0 RET NC ; yes, return

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

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

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

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

```

Dit lijkt een waarde van DE te zetten afhankelijk van deze grenzen en lijkt er ook voor te zorgen dat je in girders en elevators boards op de bovenste girder voorbij de ladder bij Kong kan komen. Geeft #0000 terug als op toegestaan gedeelte, geeft #0100 terug als te ver links en geeft #0001 terug als te ver rechts.

Eigenlijk wil je deze code dus #0000 laten teruggeven als je op het pies board bent en de Y-waarde tussen twee waardes zit.

Even testen wat er gebeurd als je altijd #0000 teruggeeft:

```

241F 110000 LD DE,#0000
2422 C9 RET

```

Je kan dan inderdaad doorlopen. Maar de girder is te kort, dus net buiten beeld valt jumpman naar beneden. Springen gaat ook goed en komt aan de andere kant uit. Dus eventjes de girder langer maken. De girders in de pies definitie vanaf #24EA uitbreiden aan de buitenkanten.

Ja dan gaat dat wel goed. Dus extra code maken die op de betreffende girders doorlopen mogelijk maakt.

Even met de debugger kijken welke Y-waarden de tunnels tussen vallen.

Onderste tunnel: #C8-#BF, bovenste tunnel: #78-#6A. Nemen #68-#78 en #B8-#C8.

Aanpassen:

241F	C3F42E	JP	#2EF4	- jump to additional code – jump to label AAAA
2422	3A0362	LD	A,(#6203)	; load A with Mario's X position

en:

2EF4	3A2762	LD	A,(#6227)	; load A with screen number – label AAAA
2EF7	FE02	CP	#02	; is this the pies board?
2EF9	2014	JR	NZ,#BBBB	; no, skip tunnel check – jump to BBBB
;				
2EFB	110000	LD	DE,#0000	; DE:= #0000
2EFE	3A0562	LD	A,(#6205)	; load A with jumpman Y-position
2F01	FE64	CP	#64	; A<64?
2F03	380A	JR	C,#BBBB	; yes, not at tunnel level – jump to BBBB
;				
2F05	FE7B	CP	#7B	; A<7B?
2F07	D8	RET	C	; yes, at tunnel level - return
;				
2F08	FEB4	CP	#B4	; A<B4?
2F0A	3803	JR	C,#BBBB	; yes, not at tunnel level – jump to label BBBB
;				
2F0C	FECB	CP	#CB	; A<CB?
2F0E	D8	RET	C	; yes, at tunnel level – return
;				
2F0F	110001	LD	DE,#0100	; DE:= #0100 – label BBBB
2F12	C32224	JP	#2422	; jump back and process edge boundaries

Werkte eerst niet goed. Deed een bounce in de tunnel. De grenzen iets ruimer gemaakt en dan werkt het wel goed.

Nu versie v0.58 opgeslagen.

Nu ervoor zorgen dat er pijltjes ter indicatie van de tunnels komen. Pijltjes tekenen.
Pijltje links is #BB en pijltje rechts is #BC.

En dan code maken om de vier pijltjes te tekenen.

2F15	21AE77	LD	HL,#77AE	; load HL with screen address
2F18	36BB	LD	(HL),#BB	; draw left pointing arrow
2F1A	21B877	LD	HL,#77B8	; load HL with screen address
2F1D	36BB	LD	(HL),#BB	; draw left pointing arrow
;				
2F1F	214E74	LD	HL,#744E	; load HL with screen address
2F22	36BC	LD	(HL),#BC	; draw right pointing arrow
2F24	215874	LD	HL,#7458	; load HL with screen address
2F27	36BC	LD	(HL),#BC	; draw right pointing arrow
;				

```
2F29 C9          RET          ; return
```

En dat moet ergens bij het tekenen van het conveyors board tussengevoegd worden.

Hier worden de boards getekend en de uitzondering voor het tekenen van de rivets gemaakt. Dit is net zoets, maar dan voor het pies board.

```
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
```

Aanpassen:

```
0CCE CD2A2F      CALL #2F2A      ; call additional code – call to label AAAA
```

en:

```
2F2A 2004        JR   NZ,#BBBB    ; screen is not rivets – jump to label BBBB  
;  
2F2C CD000D      CALL #0D00    ; yes, call sub to draw the rivets  
2F2F C9          RET          ; return  
;  
2F30 FE02        CP   #02        ; screen is pies? – label BBBB  
2F32 C0          RET          NZ          ; no, return  
;  
2F33 CD152F      CALL #2F15    ; yes, call sub to draw the arrows  
2F36 C9          RET          ; return
```

Ja dat gaat goed.

Nu versie v0.59 opgeslagen.

Het random spawnen van ghosts ongedaan maken:

Dit was aangepast, en dus weer terugzetten:

```
348A 3A1960      LD   A,(RngTimer2)  ; Determine side to spawn fireball  
348D CB47        BIT  0,A        ; If timer value is even ...  
348F CAA834      JP   Z,#34A8    ; ... jump and spawn the fireball at left side  
  
348A 3A0362      LD   A,(#6203)    ; \ Jump if Mario is on left side of the screen,  
348D CB7F        BIT  7,A        ; |  
348F CAA834      JP   Z,#34A8    ; /
```

Mmm. Dat doet juist een spawn aan dezelfde kant. Dat is niet handig. Want dan nog steeds het probleem van spawn – hap – spawn – hap – spawn – hap. Aan de andere kant spawnen? Kan dat door de JP Z in een JP NZ te veranderen?

```
348F C2A834      JP      NZ,#34A8
```

Ja dus, Dat werkt goed.

Nu het pellet aantal tellen en configureren. 122 geteld → #7A.

```
0AC3 FE02      CP      #02      ; is this pies board?  
0AC5 2004      JR      NZ,#YYYY  ; no, skip next steps  
0AC7 3E7A      LD      A,#7A    ; load A with nr of pellets on pies  
0AC9 180A      JR      AAAA    ; jump ahead – jump to label AAAA
```

Nu versie v0.60 opgeslagen.

Heel soms gaat een ghost met de verkeerde sprites de trap op. Dat kwam eerder doordat de ghosts initieel met de verkeerde sprite aangemaakt werden (omhoog kijkend). Maar dat is al aangepast. Maar is dat voor alle boards hetzelfde?

Zo gedaan:

Het lijkt erop dat dit data is die initieel gebruikt wordt:

```
; the following is table data that gets copied to #6407  
; 05 is a loop varialbe  
; 1C loops value corresponds to total length of table
```

```
3DEC 3D 01 03 02
```

Aanpassen:

```
3DEC 4D 01 03 02
```

Omhoog kijkend = #3D en #3E.

Links/rechts kijkend = #4D en #4E.

Omlaag kijkend = #5D en #5E.

Fireball data table start op #6400 met blokken van #20 met sprite value op IX+#07.

Wat wel vreemd is in de code voor het laten kijken van de ghosts is dat ladder op en ladder af gedaan worden door de oude waarde te pakken en dan #10 eraf te trekken of erbij op te tellen. Waarom niet gewoon een keiharde waarde zetten? Wellicht omdat er gewisseld wordt tussen de twee versies (onderkant sprite anders waardoor ‘wave’ effect). En het dan verschuift? Maar wordt dat niet gedaan door een bit flip?

Wat ook vreemd is, is dat bij het verlaten van een ladder altijd naar links kijken geconfigureerd wordt. Dat kon eerst omdat bij verlaten van ladder de ghost in de oorspronkelijke code altijd naar links ging. Maar dat is nu niet meer zo dacht ik. Dan klopt dat niet: verkeerde sprite als ghost de ladder naar rechts verlaat?

Gewoon eens kijken wat er gebeurd als we de sprite waarden op de trap keihard zetten i.p.v. een relatieve waarde.

Dat is deze code:

3F38	DD7E07	LD	A,(IX+#07)	; load A with sprite value – label AAAA
3F3B	0610	LD	B,#10	; load B with #10
3F3D	80	ADD	A,B	; subtract #10 from value
3F3E	DD7707	LD	(IX+#07),A	; store new sprite value
3F41	DD360D04	LD	(IX+#0D),#04	; set direction to descending ladder
3F45	C9	RET		

En:

3F46	DD7E07	LD	A,(IX+#07)	; load A with sprite value – label AAAA
3F49	0610	LD	B,#10	; load B with #10
3F4B	90	SUB	B	; subtract #10 from value
3F4C	DD7707	LD	(IX+#07),A	; store new sprite value
3F4F	DD360D08	LD	(IX+#0D),#08	; set direction to ascending ladder
3F53	C9	RET		

Aanpassingen:

3F3B	3E5D	LD	A,#5D	; load A with value of down facing ghost
3F3D	00	NOP		
3F49	3E3D	LD	A,#3D	; load A with value of down facing ghost
3F4B	00	NOP		

Ja dat lijkt goed te gaan. Zo houden en kijken of het verschijnsel wegblift.

Bij game over blijft het ghost hok staan en staat de game over er eigenlijk achter. Dat is niet mooi. Ergens worden alle sprites gewist, maar daarbij worden het hok en de ogen niet gewist.

De game over tekst wordt hier weergegeven:

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

Maar dan zijn de sprites al gewist.

De sprites worden al eerder gewist, bij het doodgaan van jumpman. Maar dat is niet altijd game over. En dan is het wel logisch dat het hok blijft staan.

Dus gewoon hier aan toevoegen.

Het ghosthok is de oude oil can en de ogen is de fire. Dat zijn de sprites: #69FC voor de oil can en #6A28 voor de fire. Dus die beiden op #00 zetten betekent een X-positie van 0 geven en dus buiten beeld plaatsen.

Aanpassen:

```
1322 CD372F      CALL  #2F37      ; call additional code – call to label AAAA
```

En:

```
2F37 CD2618      CALL  #1826      ; clear an area of the screen – label AAAA  
;  
XXXX 3E00          LD    A,#00      ; load A with #00  
XXXX 32FC69        LD    (#69FC),A   ; move ghost home off screen  
XXXX 32286A        LD    (#6A28),A   ; move ghost eyes off screen  
XXXX  C9           RET             ; return
```

Ja dat gaat goed.

Nu versie v0.61 opgeslagen.

Nu kijken dat board pas gefinished kan worden als alle pellets gepakt zijn. Als alle pellets gepakt zijn, dan wordt #64E1 (pellet bonus indicator) op #01 gezet. Dus daar moet op gecheckt worden.

De end of board routine zit hier:

```

; called from main routine at #19B9
; checks for end of level ?

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

1E5f 1F rra ; else rotate right with carry
1E60 3A0562 LD A,(#6205) ; load A with y position of mario
1E63 dA7A1E JP c,#1E7A ; skip ahead on girders and elevators

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

1E69 3A0362 LD A,(#6203) ; else load A with mario's X position
1E6C 17 RLA ; on left or right side of screen?

1E6D 3E00 LD A,#00 ; load A with #00. sprite for facing left
1E6F DA741E JP C,#1E74 ; if on left side, skip next step

1E72 3E80 LD A,#80 ; else load A with sprite facing right
1E74 324D69 LD (#694D),A ; set mario sprite
1E77 C3851E JP #1E85 ; jump ahead

; check for end of level on girders and elevators

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

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

```

Dus op 1E66 naar additionele routine springen en checken op #51 en RET NC, anders checken op gezet zijn van pellet bonus indicator: niet gezet dan return, wel gezet dan vervolgen end of level (wellicht aangepast i.v.m. andere manier van eindigen).

Maar is al aangepast i.v.m. eindigen bij Pauline:

De CP op #1E66 aanpassen naar #31. Ja nu weer eindigen bij Pauline.

En extra code aanroepen vanaf #1E6F zodat jumpman bij Pauline gezet wordt.

```

1E6F D22F26 JP C,#262F ; jump to additional code – jump to label AAAA
1E72 3E00 LD A,#00 ; else load A with sprite facing left

```

en:

```

262F 3E93 LD A,#93 ; load A with new X position – label AAAA
2631 324C69 LD (#694C),A ; set new X position
;
2634 0000 NOP, NOP
134B 000000 NOP, NOP, NOP
;
2634 3E00 LD A,#00 ; load A with sprite facing left
2636 C3741E JP #1E74 ; jump back

```

Dus nu aanpassen:

```
1E66 C3432F      JP    #2F43      ; jump to additional code – jump to label AAAA
```

en:

```
2F43 FE31      CP    #31      ; check high enough to end level – label AAAA
2F45 D0        RET   NC       ; no, return
;
2F46 3AE164    LD     A,(#64E1) ; load A with pellet bonus indicator
2F49 FE40      CP     #40      ; are all pellets eaten? Bonus indicator #40?
2F4B C0        RET   NZ       ; no, return
;
2F4C C3691E    JP     #1E69      ; jump back
```

Ja dat gaat nu goed.

Jumpman valt wel bij Pauline van de girder af. Maar dat is op zich logisch.

Wanneer hij omhoog klimt op de Kong ladders dan kan hij niet helemaal doorklimmen naar de onderkant. Het stopt ergens achter de score. Dus daar lijkt al een begrenzing op te zitten. Even kijken in de code waar dat zit.

Met de debugger blijkt dat hij stopt bij #10. Is dat ergens begrensd? Of is dat de top van de betreffende Kong ladder?

Dit zijn de Kong ladders op conveyors board:

```
00 63 18 63 58 ; kong's ladder (right)
```

```
00 53 18 53 58 ; kong's ladder (left)
```

Wat als je deze korter maakt? Dus top naar beneden: #28 i.p.v. #18. Nee dat is niet mooi. Je ziet dan de girder aan de bovenkant van de ladder. Vandaar ook de lege sprites die ervoor getekend worden.

Dan kijken of je bij het beklimmen van een ladder sowieso altijd een check kan doen wanneer kleiner dan een bepaalde waarde dan niet verder klimmen.

De joystick pushed up wordt hier afgehandeld. Niet meer doen als jumpman Y-pos < 28.

```
1B45 3A1060    LD     A,(InputState)      ; load A with input
1B48 CB57      BIT    2,A      ; is joystick pushed up ?
1B4A C2031D    JP     NZ,#1D03      ; yes, skip ahead to handle
```

Aanpassen:

```
1B4A C24F2F      JP     NZ,#2F4F      ; yes, jump to additional code, jump to label AAAA
```

en:

```
2F4F 3A0562      LD   A,(#6205)    ; load A with jumpman Y-position – label AAAA
2F52 FE2A        CP   #2A         ; jumpman Y-pos lower than #2A?
2F54 D8          RET  C           ; yes, return
;
2F55 C3031D      JP   #1D03       ; yes, skip ahead to handle
```

Ja dat gaat nu goed.

Nu versie v0.62 opgeslagen.

Nu ook doen voor barrels en elevators board.

```
; called from main routine at #19B9
; checks for end of level ?

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

1E5f  1F          RRA             ; else rotate right with carry
1E60  3A0562      LD   A,(#6205)    ; load A with y position of mario
1E63  dA7A1E      JP   c,#1E7A     ; skip ahead on girders and elevators

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

1E69  3A0362      LD   A,(#6203)    ; else load A with mario's X position
1E6C  17          RLA             ; on left or right side of screen?

1E6D  3E00        LD   A,#00         ; load A with #00. sprite for facing left
1E6F  DA741E      JP   C,#1E74     ; if on left side, skip next step

1E72  3E80        LD   A,#80         ; else load A with sprite facing right
1E74  324D69      LD   (#694D),A   ; set mario sprite
1E77  C3851E      JP   #1E85       ; jump ahead

; check for end of level on girders and elevators

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

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

He dat is grappig. Die test ook op de hoogte #31. Dat is nu hetzelfde als voor de conveyors board. Dus de code van conveyors is hiervoor te gebruiken.

Aanpassen:

```
1E7A  C3432F      JP   #2F43       ; jump to additional code – jump to label AAAA
```

Ja dat gaat in beide velden helemaal goed.

Nu kijken hoe dit in te bouwen bij de rivets. Dan de laatste rivet niet pakken wanneer nog niet alle pellets gegeten zijn.

Dat lijkt hier te zitten:

```
1A81 7E      LD      A,(HL)      ; get the status of the rivet mario is crossing
1A82 A7      AND     A          ; has this rivet already been traversed?
1A83 C8      RET      Z          ; yes, return

; a rivet has been traversed

1A84 3600      LD      (HL),#00      ; set this rivet as cleared
1A86 219062    LD      HL,#6290      ; load HL with address of number of rivets remaining
```

Een extra check toevoegen of dit de laatste is èn nog niet alle pellets opgegeten zijn.

Aanpassen:

```
1A84 C3582F      JP      #2F58      ; jump to additional code – jump to label AAAA
1A87 0000          NOP,NOP
en:

2F58 3A9062      LD      A,(#6290)    ; load A with number of rivets remaining – AAAA
2F5B FE01          CP      #01          ; is this the last rivet?
2F5D 2006          JR      NZ          ; no, it is not, jump ahead – jump to label BBBB
;
2F5F 3AE164      LD      A,(#64E1)    ; load A with pellet bonus indicator
2F62 FE40          CP      #40          ; are all pellets eaten? Bonus indicator #40?
2F64 C0            RET     NZ          ; no, do not remove last pellet
;
2F65 3600          LD      (HL),#00      ; set this rivet as cleared – label BBBB
2F67 219062        LD      HL,#6290      ; load HL with address of number of rivets remaining
2F6A C3891A        JP      #1A89
```

Ja dat gaat goed zo.

Nu versie v0.63 opgeslagen.

Nu nog kijken dat jumpman bij Pauline op conveyors board niet er vanaf kan vallen aan de linkerkant. Daarmee hetzelfde als barrels en elevators board.

Dat lijkt de volgende code te zijn:

```

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

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

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

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

```

Dit stuk wordt alleen gedaan voor girders or elevators. Hier wordt wanneer de X-waarde van jumpman links begrenst (#6C) wanneer Y-waarde jumpman lager dan een bepaalde waarde. Maar dat is niet alleen op het niveau van Pauline, maar ook op de girder van Kong. Dus dat werkt niet echt lekker voor het conveyors board. Dat klopt: op Kongs girder jumpman's Y-position is #50 en bij Pauline is jumpman's Y-position #30.

Aanpassen:

```

2431 FE04 CP #04 ; is this the rivets?
2433 C8 RET Z ; yes, return
;
2434 C36D2F JP #2F6D ; jump to additional code – jump to label AAAA
2437 0000 NOP,NOP

```

en:

```

2F6D FE02 CP #02 ; is this conveyors? - label AAAA
2F6F 2004 JR NZ,#BBBB ; no, skip next steps – jump to label BBBB
;
2F71 0638 LD B,#38 ; load B with Y check for conveyors
2F73 1802 JR #CCCC ; jump ahead – jump to label CCCC
;
2F75 0658 LD B,#58 ; load B with Y check for barrels and elevators - BBBB
;
2F77 3A0562 LD A,(#6205) ; load A with jumpman's Y-position - CCCC
2F7A B8 CP B ; jumpman higher than Y check?
2F7B D0 RET NC ; no, return
;
2F7C C33924 JP #2439 ; jump back

```

Ja dat lijkt goed te gaan.

Ook nog even checken met barrels board en elevators board. Ja die gaan allebei ook goed.

Nu versie v0.64 opgeslagen.

Nog doen:

- Aanpassen sprite van de appel (minder langwerpig maken).
- Boardvolgorde 1-2-3-4 voor elk level.
- Weergeven sub-levels.
- Opnemen level en sublevel in High Score screen.
- Aanpassen design High Score screen.
- Bonus items op barrels board.
- Blauwe barrels met ghost ipv doodshoofd.
- Kill screen a la pacman met deel scherm random karakters.
- Aanpassen design How High screen.
- Aanpassen design Title screen: Donkey Kong & Pac-man UNITED.
- Kijken eventueel geen stop wanneer alle pellets gegeten.
- Muziek Pac-man invoegen (how high of end of level)?
- Wellicht pellet eten geluid (is geluid van credits inserted) hoger pitchen?

Aanpassen van de sprite van de appel. Nee niet doen. Is juist de goede sprite kopie van het originele pac-man spel.

De boardvolgorde aanpassen:

```
178E DF      RST    #18          ; count down timer and only continue here if zero, else RET
178F 2A2A62  LD     HL,(#622A)   ; load HL with address for this screen/level
1792 23      INC    HL           ; next screen
1793 7E      LD     A,(HL)      ; load A with the screen for next
1794 FE7F    CP     #7F          ; at end ?
1796 C29D17  JP     NZ,#179D    ; no, skip next 2 steps

1799 21733A  LD     HL,#3A73    ; load HL with table for screens/levels for level 5+
179C 7E      LD     A,(HL)      ; load A with the screen

179D 222A62  LD     (#622A),HL  ; store screen address lookup for next time
17A0 322762  LD     (#6227),A   ; store A into screen number
```

Met als datababel:

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

Eigenlijk heel simpel. Loopt door de hele tabel heen en springt dan bij 7F terug naar 3A73: de beginwaarde van eerste screen bij level 5+

Dus nu aanpassen:

3A65: 01 02 03 04 ; level 1+

3A69: 7F

en:

1799 21653A LD HL,#3A65 ; load HL with table for screens/levels for level 1+

Voor rivets beëindigen is er soortgelijke code:

Ja dus:

```
193D 2A2A62 LD HL,(#622A) ; load HL with contents of #622A. this is a pointer to the levels/screens data
1940 23 INC HL ; increase HL. = next level
1941 7E LD A,(HL) ; load A with contents of HL = the screen we are going to play next
1942 fe7f CP #7F ; is this the end code ?
1944 c24B19 JP NZ,#194B ; no, skip next 2 steps

1947 21733A LD HL,#3A73 ; yes, load HL with #3A73 = start of table data for screens/levels for level 5+
194A 7E LD A,(HL) ; load A with screen number from table
```

Aanpassen:

1947 21653A LD HL,#3A65 ; yes, load HL table for screens/levels for level 1+

Testen is wel lastig. Kijken hoe nu invincibility aan te zetten.

Eerdere aanpassing:

```
1E9F 3AE264 LD A,(#64E2) ; load power pellet mode active indicator
1EA2 FE01 CP #01 ; is the power pellet mode active?
1EA4 C8 RET Z ; yes, return
;
1EA5 CD0828 CALL #2808 ; check for collisions with hostile sprites
1EA8 C9 RET ; return
```

Hier op 1EA5 3xNOP doen zodat er geen check meer op collisions gedaan wordt.

Ja dat gaat goed. De velden worden goed doorlopen.

Nu versie v0.65 opgeslagen.

Dan nu in het level ook een sublevel angeven (het board dat gespeeld wordt). Dit wordt in DK RNDMZR ook gedaan.

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

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

```

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

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

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

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

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

```

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

Aanpassing:

```

06FA C37F2F JP #2F7F ; jump to additional code – jump to label AAAA
en

2F7F 32C374 LD (#74C3),A ; draw level to screen (high byte) – label AAAA
;
2F82 3E2C LD A,#2C ; load A with "-"
2F84 328374 LD (#7483),A ; draw hyphen
;
2F87 3A2962 LD A,(#6229) ; load A with level number
2F8A FE00 CP #00 ; is the level number 0?
;
2F8C 2804 JR Z, #3820 ; yes, skip next lines
2F8E 3A2E62 LD A,(#622E) ; load A with number of Goofy Kongs to draw
2F91 3C INC A ; increment A
2F92 326374 LD (#7463),A ; draw screen in level number
;
2F95 C9 RET ; return

```

Dit werkt nog niet omdat de variabele #622E steeds 1 te hoog is. De update daarvan wordt gedaan bij het tekenen van de Goofy Kongs. Daar eruit halen en eigen routines voor maken.

How High scherm toch nog aanpassen. Nu alvast een heel deel eruit halen: wissen van #0BFA t/m #0C81.

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

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

```

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

```

Aanpassen:

```

17B3 C3962F JP #2F96 ; jump to additional code, jump to label AAAA
en
2F96 3608 LD (HL),#08 ; set game mode2 to 8 – label AAAA
;
2F98 212E62 LD HL, #622E ; load HL with variable
2F9B 34 INC (HL) ; increase variable
;
2F9C C9 RET ; return

```

Finish level rivets is code vanaf #18C6.

```

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

```

Aanpassen:

```

1939 C39D2F JP #2F9D ; jump to additional code, jump to label AAAA
en
2F9D 32206A LD (#6A20),A ; store A into heart sprite X position – label AAAA
;
2FA0 3E00 LD A,#00 ; A := 00
2FA2 322E62 LD (#622E),A ; store A in variable
;
2FA5 C9 RET ; return

```

Ja, dat gaat goed zo.

Nu versie v0.66 opgeslagen.

Nu kijken of het level en sublevel ook in de high score list opgenomen kunnen worden.

De tabelregel wordt initieel gevuld met veertien spaties:

```

; sets #61B7 through #61C4 to #10 (???)  

13EB 060E LD B,#0E ; for B = 1 to #E  

13ED 3610 LD (HL),#10 ; store #10 into memory at (HL)  

13EF 23 INC HL ; next HL  

13F0 10FB DJNZ #13ED ; next B

```

Nu dat niet doen. Eerst twee spaties, dan vier posities met het level en sub-level en dan nog acht spaties.

13EB 3610	LD	(HL),#10	; store #10 into memory at (HL)
13ED 23	INC	HL	; next HL
13EE C37C25	JP	#257C	; jump to additional code – jump to label AAAA
13F1 00	NOP		

en:

257C 3610	LD	(HL),#10	; store #10 into memory at (HL) – label AAAA
257E 23	INC	HL	; next HL
257F 23	INC	HL	
;			
2580 3A2962	LD	A,(#6229)	; load A with level nr
2583 010AFF	LD	BC,#FF0A	; B: = #FF, C := #0A (10 decimal)
;			
2586 04	INC	B	; increment B – label CCCC
2587 91	SUB	C	; subtract 10 decimal
2588 30FB	JR	NC,#1631	; not carry, loop again – jump to label CCCC
;			
258A 81	ADD	A,C	; add 10 back to A to get a number from 0 to 9
258B 77	LD	(HL),A	; store level nr (singles) into memory at (HL)
258C 2B	DEC	HL	; next HL
;			
258D 78	LD	A,B	; load A with B (number of tens)
258E 77	LD	(HL),A	; store levelnr (tens) into memory at (HL)
258F 23	INC	HL	; next HL
2590 23	INC	HL	; next HL
;			
2591 362C	LD	(HL),#2C	; store hyphen into memory at (HL)
2593 23	INC	HL	; next HL
;			
2594 3A2E62	LD	A,(#622E)	; load A with board nr
2597 3C	INC	A	; increment A
2598 77	LD	(HL),A	; store board nr into memory at (HL)
2599 23	INC	HL	; next HL
;			
259A 0608	LD	B,#08	; for B = 1 to #08 – label BBBB
;			
259C 3610	LD	(HL),#10	; store #10 into memory at (HL)
259E 23	INC	HL	; next HL
259F 10FB	DJNZ	#1640	; next B – jump back to label BBBB
;			
25A1 C3F213	JP	#13F2	; jump back

Ja dat gaat goed. Alleen wordt de naam er overheen geschreven. Dat nog aanpassen:

14CC 11F9FF	LD	DE,#FFF9	; load DE with offset of -8 (decimal)
14CF 19	ADD	HL,DE	; add offset
14D0 223A60	LD	(#603A),HL	; store result into ???

en

15D3 0605	LD	B,#05	; for B = 1 to #6 (11 decimal)
15D5 21E875	LD	HL,#75E8	; load HL with screen vram address
15D8 FD2A3A60	LD	IY,(#603A)	; load IY with ???
15DC 11E0FF	LD	DE,#FFE0	; load DE with offset of -#20
15DF 7E	LD	A,(HL)	; load A with

```

15E0 FD7700 LD (IY+#00),A ; store
15E3 FD23 INC IY ; next
15E5 19 ADD HL,DE ; add offset
15E6 10F7 DJNZ #15DF ; next B

```

Gaat nu allemaal goed. Alleen nog de header erboven aanpassen.

De kop moet dan ook waarschijnlijk aangepast worden:

```

367D: 9E 37 19 ; #379E "RANK SCORE NAME"
3790: 19 10 24 19 1D 15 10 10 30 03 00 31 10 3F 92 77 I.TIME.....
37A0: 22 11 1E 1B 10 10 23 13 1F 22 15 10 10 1E 11 1D RANK..SCORE..NAM
37B0: 15 10 10 10 10 3F 72 77 29 1F 25 22 10 1E 11 1D E.....YOUR.NAM

```

Dat past niet. Maar in de 'REGI TIME' ervoor zit extra ruimte. Maar dan past nog niet dus ook andere aanpassing: RANK aanpassen naar een #.

Dan dus de 3F op #379D één positie naar voren en de 9277 op #379E ook.

Dan begint RANK SCORE NAME op #379D, dus dan aanpassen:

367D 9D 37

En dan op de plaats van de N moet een hekje komen. Dus twee opschuiven in beeld: #7792 wordt dan #7752.

```

#      S C O R E      L E V E L      N A M E
#379D: 52 77 FA 10 10 23 13 1F 22 15 10 10 1C 15 26 15 1C 10 10 1E 11 1D 15 3F

```

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

Ja dat gaat allemaal goed.

Nu versie v0.67 opgeslagen.

Kijken voor de kleuren van het high score screen. Dat is het bovenste palette. Eerst even de kleuren van het hoofdscherm daarnaartoe kopiëren.

Maar dan zijn alle teksten in het wit.



Twee kleuren gebruiken: wit en blauw: blok #03 allemaal blauw maken en blok #04 allemaal wit maken.

En dan de regels aanpassen voor de kleuren.

```

Offset(h) 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
00000000 07 07 07 07 03 04 03 03 03 04 04 03 04 03 04 03 04 03 04 03 04 04 04 04 04 04 04 04 04 04 04 03 04
00000020 07 07 07 07 03 04 03 03 03 04 04 03 04 03 04 03 04 03 04 03 04 04 04 04 04 04 04 04 04 04 04 03 04
00000040 07 07 07 07 03 04 03 03 03 04 04 03 04 03 04 03 04 03 04 03 04 04 04 04 04 04 04 04 04 04 04 03 04
00000060 07 07 07 07 03 04 03 03 03 04 04 03 04 03 04 03 04 03 04 03 04 04 04 04 04 04 04 04 04 04 04 03 04
00000080 07 07 07 07 03 04 03 03 03 04 04 03 04 03 04 03 04 03 04 03 04 04 04 04 04 04 04 04 04 04 04 03 04
000000A0 07 07 07 07 03 04 03 03 03 04 04 03 04 03 04 03 04 03 04 03 04 04 04 04 04 04 04 04 04 04 04 03 04
000000C0 07 07 07 02 03 04 03 03 03 04 04 03 04 03 04 03 04 03 04 04 04 04 04 04 04 04 04 04 04 04 04 03 04
000000E0 07 07 07 02 03 04 03 03 03 04 04 03 04 03 04 03 04 03 04 04 04 04 04 04 04 04 04 04 04 04 04 03 04

```

Ziet er nu zo uit:



Nu versie v0.68 opgeslagen.

Nog wel de cursor bij het selecteren van de initialen blauw maken. Wordt de kleur ervan hier gezet?

```

160C 23      INC     HL
160D 360C    LD      (HL),#0C

```

Zou kunnen. Dat betekent dat kleurslot #0C aangepast moet worden.

0 1 2 3 in het blok moet worden: 0 3 1 2 Dus 3 naar 1, 1 naar 2 en 2 naar 3.

C-2j: OF 00 0E OF → OF OF 00 0E.

C-2k: OF 00 00 0C → OF OC 00 00.

Ja dat is goed nu.

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

Onderste regel: DESIGN BY PAUL GOES:

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

```

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

```

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

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

Ja dat is nu goed.

Nu versie v0.69 opgeslagen.

Bonus items op het barrels board plaatsen.

Hebben we al eens gedaan bij Anniversary Edition en bij Hearthunt. Maar in beide gevallen dubbele definities en daartussen wisselen. Nu maar één definitie gebruiken. Dus wellicht eenvoudiger te maken.

```
1000 CDB025      CALL    #25B0      ; call additional code – call label AAAA
```

en

```

25A4 18 73 05 CD      ; position of hat on barrels is 18,CD
25A8 17 74 06 8D      ; position of purse on barrels is 17,8D
25AC E8 75 05 AD      ; position of umbrella on the barrels is at E8,AD

```

```

;
25B0 21A425      LD      HL,#25A4      ; start of bonus items table – label AAAA
25B3 110C6A      LD      DE,#6A0C      ; load DE with sprite destination
25B6 010C00      LD      BC,#000C      ; 3 items x 4 bytes = 12 bytes (#0C)
25B9 EDB0      LDIR      ; draw bonus item sprites
;
25BB 210C3E      LD      HL, #3EOC      ; start of table data for hammers
25BE C9          RET      ; jump back

```

Dat gaat niet goed. Die op 25A8 kan niet gepakt worden (aardbei). Ze mogen niet dezelfde X-positie hebben. Dan kan er maar één gepakt worden. Dus 1 positie naar links verschuiven. Ja dat gaat goed.

Nu versie v0.70 opgeslagen.

Het How High scherm aanpassen.

De vier spookjes op een rij met hun namen eronder.

Dan verder naar beneden jumpman → O → pac-man.

En dan de teksten HOW HIGH CAN YOU GET ? en DO EAT ALL PELLETS !

De namen zijn: BLINKY PINKY INKY CLYDE (rood / roze / blauw / geel)

24 posities: 6 1 5 2 4 2 5

Eerst de teksten maar eens maken en positioneren.

Er zijn nog wat textblokken vrij:

```
3657: 06 00      6    ; unused ?
365B: 08 00      8    ; unused ?
3685: 1D 00      1D   ; unused ?
```

3657 gebruiken voor de tekst DO EAT ALL PELLETS !:

```
3657 5A 3F
3F5A 3E 77 14 1F 10 15 11 24 10 11 1C 1C 10 20 15 1C 1C 15 24 23 10 39 3F
```

365B gebruiken voor de tekst BLINKY PINKY INKY CLYDE

```
365B 71 3F
3F71 8E 77 12 1C 19 1E 1B 29 10 20 19 1E 1B 29 10 10 19 1E 1B 29 10 10 13 1C 29 14 15 3F
```

3685 gebruiken voor de tekst → O → (pijltje = #BC en pellet = #DB).

```
3685 8D 3F
3F8D 75 76 BC 10 10 DB 10 10 BC 3F
```

En deze teksten dan tekenen bij het How High scherm.

```
0C7C 110603 LD DE,#0306    ; load task data for text #6 "DO EAT THE PELLETS !"
0C7F CD9F30 CALL #309F     ; insert task to draw text

0C76 110803 LD DE,#0308    ; load task data for text #8 "BLINKY PINKY INKY CLYDE"
0C79 CD9F30 CALL #309F     ; insert task to draw text

0C70 111D03 LD DE,#031D    ; load task data for text #8 "→ O →"
0C73 CD9F30 CALL #309F     ; insert task to draw text
```

Nu versie v0.71 opgeslagen.

En dan nog een stuk code toevoegen voor het plaatsen van de ghosts, jumpman en pac-man sprites. Dat zijn zes sprites in totaal. Er is al een algemene routine voor een dergelijke kopiëer actie. Die hergebruiken met een definitie voor de zes sprites.

De volgende code kan de sprites plaatsen:

```
0BFA 21080C LD HL,#0C08    ; start of sprites table
0BFD 118069 LD DE,#6980    ; load DE with sprite destination
0C00 011800 LD BC,#0018    ; 6 items x 4 bytes = 24 bytes (#18)
```

```

0C03 EDB0      LDIR          ; draw sprites
;
0C05 C3700C    JP      #0C70
;
0C08 30 CD 0C 64 ; red ghost – right facing
0C0C 64 CD 0D 64 ; pink ghost – right facing
0C10 98 4D 0E 64 ; blue ghost – left facing
0C14 CC 4D 0F 64 ; yellow ghost – left facing
0C18 4A 81 02 AA ; jumpman – running right
0C1C AE 55 00 AA ; pacman – right facing

```

Ja dat gaat goed.

Nu versie v0.72 opgeslagen.

Nu het titelscherm aanpassen.

Eerst een aantal sprites toevoegen en dan gaan schuiven en teksten aanpassen.

De letters PAC-MAN zijn P/64 A/65 C/66 -/67 M/68 A/69 N/6A. Als sprites.

Dus code maken die sprites tekent en dan de code die DK tekent weglaten.

Het titel scherm wordt getekend vanaf #07CB. Het eerste stuk code is alleen maar om te knipperen. Maar dat willen we helemaal niet. Dus verwijderen. Ook het tekenen van Kong verwijderen en dan code gaan toevoegen voor de sprites en de tekst.

#07CB t/m 07F6 leegmaken en daarin dan het palette goed zetten.

REG_PALETTE_A = #7D86 moet #01 zijn.

REG_PALETTE_B = #7D87 moet #00 zijn.

```

07CB 3E01      LD   A,#00      ; load A with #01
07CD 32867D    LD   (REG_PALETTE_A),A ; set REG_PALETTE_A
07D0 3E00      LD   A,#01      ; load A with #01
07D2 32877D    LD   (REG_PALETTE_B),A ; set REG_PALETTE_B
;
07D5 21200C    LD   HL,#0C20    ; start of sprites table
07D8 118069    LD   DE,#6980    ; load DE with sprite destination
07DB 011C00    LD   BC,#001C    ; 7 items x 4 bytes = 28 bytes (#1C)
07DE EDB0      LDIR          ; draw sprites

```

En dan de sprites definiëren:

```

0C20 20 64 02 80 ; P
0C24 30 65 02 80 ; A
0C28 40 66 02 80 ; C
0C2C 50 67 02 80 ; -
0C30 60 68 02 80 ; M

```

```
0C34 70 69 02 80 ; A  
0C38 80 6A 02 80 ; N
```

Kleuren van PACMAN aanpassen. #00 #OB.

PAC-MAN staat nu goed tussen de DONKEY KONG.

Nu versie v0.73 opgeslagen.

CROSSOVER als sprites opnemen.

```
07D5 21200C LD HL,#0C20 ; start of sprites table  
07D8 118069 LD DE,#6980 ; load DE with sprite destination  
07DB 014000 LD BC,#0040 ; 16 items x 4 bytes = 64 bytes (#40)  
07DE EDB0 LDIR ; draw sprites
```

En dan de sprites definiëren:

```
0C20 20 64 02 80 ; P  
0C24 30 65 02 80 ; A  
0C28 40 66 02 80 ; C  
0C2C 50 67 02 80 ; -  
0C30 60 68 02 80 ; M  
0C34 70 69 02 80 ; A  
0C38 80 6A 02 80 ; N  
;  
0C3C 20 3F 02 B0 ; C 2C  
0C40 34 4F 02 B0 ; R 40  
0C44 48 5F 02 B0 ; O 54  
0C48 5C 6F 02 B0 ; S 68  
0C4C 70 6F 02 B0 ; S 7C  
0C50 84 5F 02 B0 ; O 90  
0C54 98 4A 02 B0 ; V A4  
0C58 AC 5A 02 B0 ; E B8  
0C5C C0 4F 02 B0 ; R CC
```

Ja dat is leuk. Ziet er goed uit.

Nu versie v0.74 opgeslagen.

En teksten:

© 1981-2024 – in de plaats van de DUEL tekst.

NINTENDO / NAMCO ipv copyright 1981-2023 NINTENDO tekst.

Dat is allemaal vanuit de basis DK Wizardry. Kijken hoe dat daar gedaan is.

In #3F00 de copyright notice maken:

© 1981-2021 NINTENDO

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

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

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

En dan in #3F18 de titel maken:

WIZARDRY

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

En de verwijzing ernaar toe aanpassen:

3689: 18 3F

Nu dan in 3F00 de tekst NINTENDO & NAMCO maken (ampersand nieuw karakter op #9D).

FC 76 1E 19 1E 24 15 1E 14 1F 10 9D 10 1E 11 1D 13 1F 3F

En de plek van de checksum ook aanpassen: INTEND: start op 3F03:

2441 21033F LD HL,#3F03 ; load HL with ROM area that has NINTENDO written

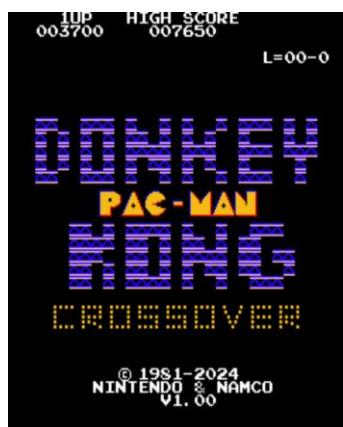
En dan de copyright maken op een andere plek: op #3FC8

BB 76 49 4A 01 09 08 01 2C 02 00 02 04 3F

En de verwijzing ook aanpassen:

3689: C8 3F

Ja dat ziet er nu goed uit.



Nu versie v0.75 opgeslagen.

Even testen of er inderdaad een kill screen komt op level L=22.

Aanpassen startlevel:

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

```
095E 16 65 3A 01 00 00 00
```

Ja dat klopt. Gewoon dan een kill screen. Kijken of dat een pac-man like tintje gegeven kan worden. Door er een deel van het scherm te vullen met rubbish karakters.

Het tekenen van de girders voor het barrels board wordt hier gedaan:

```
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  
  
0CD1 C3A03F JP #3FA0 ; fix retractable ladders for pie factory and returns to #0D5F  
  
; girders from #0CAB  
  
0CD4 11E43A LD DE,#3AE4 ; Load DE with start of table data for girders  
0CD7 3E08 LD A,#08 ; A := 8 = music code for girders  
0CD9 328960 LD (#6089),A ; set music for girders  
0CDC C3C60C JP #0CC6 ; jump back
```

Maar is aangepast voor de nieuwe plek voor de board definitie:

```
0CD4 11853B LD DE,#3B85 ; Load DE with start of table data for girders
```

Op #0CC6 wordt het scherm getekend. Dit anders maken:

```
0CC6 CDBF25 CALL #25BF ; call additional code – call to label AAAA
```

en:

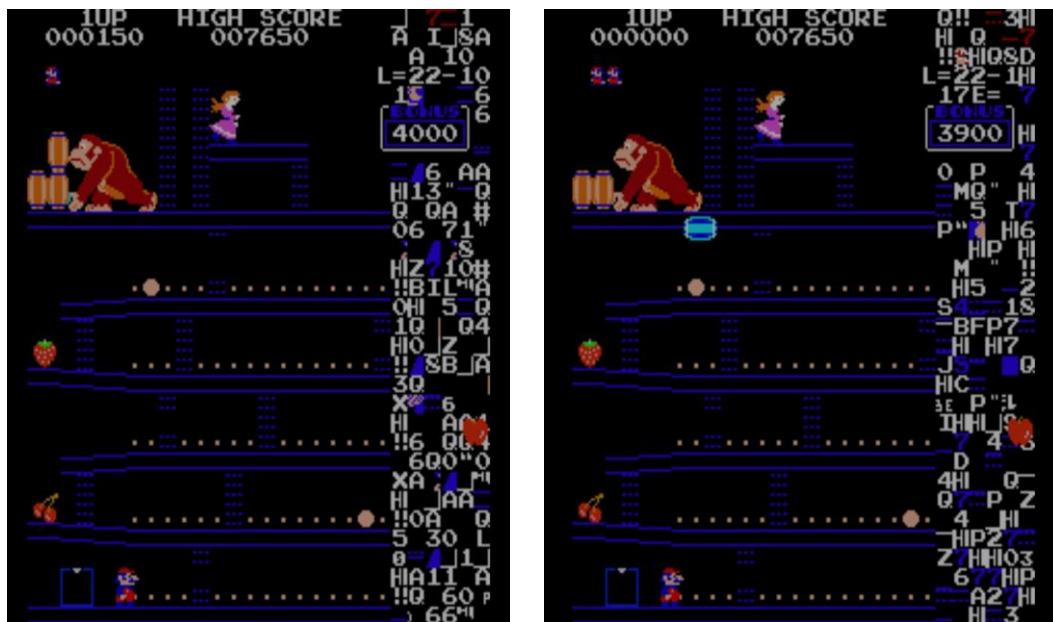
```
25BF CDA70D CALL #0DA7 ; draw the screen – label AAAA  
;  
25C2 3A2762 LD A,(#6227) ; load A with screen number  
25C5 FE01 CP #01 ; is this the barrels board?  
25C7 C0 RET NZ ; no, return  
;  
25C8 3A2962 LD A,(#6229) ; load A with level
```

```

25CB FE16      CP    #16      ; is this L=22?
25CD C0        RET   NZ       ; no, return
;
25CE 06FF      LD     B,#FF    ; load B with #FF
25D0 210074    LD     HL,#7400  ; load HL with start of screen RAM
25D3 110010    LD     DE,#1000  ; load DE with some point in the code ROM
;
25D6 1A        LD     A,(DE)   ; load A with content of DE – label BBBB
25D7 77        LD     (HL),A  ; copy into HL
25D8 13        INC   DE      ; next DE
25D9 23        INC   HL      ; next HL
25DA 10FA      DJNZ  #BBBB   ; repeat until B=0
;
25DC C9        RET   NZ       ; return

```

Ziet er zo uit:



En jumpman valt dan naar beneden, naar de bovenkant en dan weer verder totdat hij afgaat.

Maar het verziekt ook de bovenste regels en dat komt pas weer goed wanneer er een nieuwe game gestart wordt. Even bedenken of we dit wel zo willen.

Kijken om het zodanig aan te passen dat er per regel getekend wordt en dan steeds de bovenste regels overslaan.

```

25CE 0E10      LD     C,#10    ; load C with #10
25D0 210874    LD     HL,#7407  ; load HL with start of screen RAM + #08
25D3 114810    LD     DE,#1048  ; load DE with some point in the code ROM
;
25D6 0618      LD     B,#18    ; load B with #18 – label AAAA
;

```

```

25D8 1A          LD    A,(DE)      ; load A with content of DE – label BBBB
25D9 77          LD    (HL),A     ; copy into HL
25DA 13          INC   DE         ; next DE
25DB 23          INC   HL         ; next HL
25DC 10FA        DJNZ #BBBB      ; repeat until B=0, repeat at label BBBB
;
25DE 0607        LD    B,#07      ; load B with #07
25E0 23          INC   HL         ; skip video location – label CCCC
25E1 10FD        DJNZ #CCCC      ; repeat until B=7, repeat at label CCCC
;
25E3 0D          DEC   C          ; decrement C
25E4 20FO        JR    NZ,#AAAA    ; repeat until C=0
;
25E6 C9          RET

```



Zodanig positioneren dat de blauwe barrel bij de oil can kan komen. Ja dat is nu goed.

En dan dit killscreen doen bij L=21. Je hebt dan net $20 * 4 = 80$ boards weggespeeld. Je kunt dan wellicht een tijdje uithouden op dit kill screen.

```
25CB FE16          CP    #15      ; is this L=21?
```

Aanpassen start level om te testen:

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

```
095E 14 65 3A 01 00 00 00
```

Zie nu ook dat de C van CROSSOVER de blanco sprites waren die bij de rivets stage ervoor zorgen dat je afgaat bij DK. Dat moet dan een andere lege sprite worden. Het was sprite #3F.

Dat zit hier:

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

#48 is een lege sprite.

Aanpassen:

```
117E 48 0C 08 08
```

Ja dat gaat nu goed.

Nu versie v0.76 opgeslagen.

Nu werken aan de geluiden.

Stuiter geluid springs is hetzelfde als pellet eten. Stuitergeluid uitzetten.

Dat lijkt hier te zitten:

```
2E9C 21AA39 LD HL,#39AA      ; load HL with start of table data  
2E9F 3E03 LD A,#03          ; load sound duration of 3  
2EA1 328360 LD (#6083),A    ; play sound for bouncer  
2EA4 C34B2E JP #2E4B        ; jump back
```

Aanpassen:

```
2E9F 0000000000 5xNOP      ; do not play bouncer sound
```

Ja, dat gaat goed nu.

Wellicht een geluid spelen bij de power pellet tweede helft om duur beter duidelijk te maken?

Wordt hier aangezet:

```
2F00 218960 LD HL,#6089      ; load HL with music address  
2F03 3604 LD (HL),#04        ; set music for hammer
```

en hier weer uitgezet (en voorgaande muziek weer aangezet):

```
2F76 3A8963 LD A,(#6389)      ; load A with previous background music  
2F79 328960 LD (#6089),A      ; set music with what it was before the hammer was grabbed
```

Nee dat gaat hele rare effecten hebben als je daarmee gaat rommelen.

Dan het pacman muziekje maken. Dat is eerder gedaan in Anniversary Edition:

Uitleg van de patternopbouw:

```
; Format:  
; * If the value is $00, end the pattern.  
; * Else, if bit 7 is clear, the byte is the duration for all following notes until another duration code is read.  
; * Else, for channel A:  
;   * Bits 0-3 set the note (A, A#, B, C ...)  
;   * Bits 4-5 set the octave (higher values choose lower octaves)  
;   * Bit 6 indicates the next byte is a note to play on channel B.  
;  
; Channel B notes follow the same format, except bits 6-7 are ignored.  
;  
; This format means it is impossible to play a note on channel B without also playing one on channel A,  
; and the channels cannot have separate durations.
```

Notentabel:

0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011
A	A#	B	C	C#	D	D#	E	F	F#	G	G#
0	1	2	3	4	5	6	7	8	9	A	B

Voorbeeld pattern:

Eerste deel van de pattern:

10 E8 A4 E8 A4 E6 A2 E6 A2 E4

Uitschrijven:

10 0001 0000 – duration voor de rest van de notes
E8 1110 1000 – Channel A – Octave 10 – Note 1000 - F
A4 1010 0100 – Channel B – Octave 10 – Note 0100 – C#
E8 1110 1000 – Channel A – Octave 10 – Note 1000 - F
A4 1010 0100 – Channel B – Octave 10 – Note 0100 – C#
E6 1110 0110 – Channel A – Octave 10 – Note 0110 – D#
A2 1010 0010 – Channel B – Octave 10 – Note 0010 - B
E6 1110 0110 – Channel A – Octave 10 – Note 0110 – D#
A2 1010 0010 – Channel B – Octave 10 – Note 0010 - B

Pac-Man

Maxime Beauchamp

Intro Theme

Toshio Kay

Er is dus een beperking dat de duration van de A en B channel hetzelfde moeten zijn. Dat maakt het wel ingewikkeld. Herhalen van de bass note zorgt er waarschijnlijk voor dat deze meerdere keren gespeeld wordt en niet langer klinkt.

Alleen de eerste vier noten bovenin:

40 - 0100 0000 – Duration voor de rest van de notes
A2 - 1010 0010 - Octave 10 – B
B2 - 1011 0010 - Octave 11 - B
A9 - 1010 1001 - Octave 10 – F#
A4 - 1010 0110 - Octave 10 – D#

Even het How High pattern aanpassen om te testen. Dat is deze:

```
; Pattern $10, How High Can You Get?  
3EB: 1B E2 88 09 A4 12 82 A8 12 88 A4 E2 82 A4 12 E0  
3FB: 80 09 82 83 86 88 8A 8C 24 D0 88 00
```

Deze aanpassing moet doorgevoerd worden in de s_3i_b.bin.

En dan testen met de herhaalde bas noot op het B channel:

40 - 0100 0000 – Duration (1/16 note) – start op #3EB
E2 - 1110 0010 - Octave 10 – B
82 - 1000 0010 - Octave 00 - B
F2 - 1111 0010 - Octave 11 - B
82 - 1000 0010 - Octave 00 - B
E9 - 1110 1001 - Octave 10 – F#
82 - 1000 0010 - Octave 00 - B
E6 - 1110 0110 - Octave 10 – D#
92 - 1001 0010 - Octave 01 – B

20 - 0010 0000 – Duration (1/32 note) – start op #3F4

F2 - 1111 0010 - Octave 11 - B

82 - 1000 0010 - Octave 00 - B

40 - 0110 0000 – Duration (3/32 note)

E9 - 1110 1001 - Octave 10 – F#

82 - 1000 0010 - Octave 00 - B

60 - 0100 0000 – Duration (1/16 note)

E6 - 1110 0110 - Octave 10 – D#

82 - 1000 0010 - Octave 00 – B

40 - 0100 0000 – Duration (1/16 note)

E6 - 1110 0110 - Octave 10 – D#

92 - 1001 0010 - Octave 01 – B

40 - 0100 0000 – Duration (1/16 note) – start op #3FF

E3 - 1110 0011 - Octave 10 – C

83 - 1000 0011 - Octave 00 - C

F3 - 1111 0011 - Octave 11 - C

83 - 1000 0011 - Octave 00 - C

EA - 1110 1010 - Octave 10 – G

83 - 1000 0011 - Octave 00 - C

E7 - 1110 0111 - Octave 10 – E

93 - 1001 0011 - Octave 01 – C

20 - 0010 0000 – Duration (1/32 note) – start op #408

F3 - 1111 0011 - Octave 11 - C

83 - 1000 0011 - Octave 00 - C

60 - 0110 0000 – Duration (3/32 note)

EA - 1110 1010 - Octave 10 – G

83 - 1000 0011 - Octave 00 - C

40 - 0100 0000 – Duration (1/16 note)

E7 - 1110 0111 - Octave 10 – E

83 - 1000 0011 - Octave 00 - C

E7 - 1110 0111 - Octave 10 – E

93 - 1001 0011 - Octave 01 – C

40 - 0100 0000 – Duration (1/16 note) – start op #413

E2 - 1110 0010 - Octave 10 – B

82 - 1000 0010 - Octave 00 - B

F2 - 1111 0010 - Octave 11 - B

82 - 1000 0010 - Octave 00 - B

E9 - 1110 1001 - Octave 10 – F#

82 - 1000 0010 - Octave 00 - B

E6 - 1110 0110 - Octave 10 – D#

92 - 1001 0010 - Octave 01 – B

20 - 0010 0000 – Duration (1/32 note) – start op #41C

F2 - 1111 0010 - Octave 11 - B
82 - 1000 0010 - Octave 00 - B
60 - 0110 0000 – Duration (3/32 note)
E9 - 1110 1001 - Octave 10 – F#
82 - 1000 0010 - Octave 00 - B
40 - 0100 0000 – Duration (1/16 note)
E6 - 1110 0110 - Octave 10 – D#
82 - 1000 0010 - Octave 00 - B
E6 - 1110 0110 - Octave 10 – D#
92 - 1001 0010 - Octave 01 – B

08 - Duration (1/32 note) – start op #427
E6 - 1110 0110 - Octave 10 – D#
89 - 1000 1001 - Octave 00 – F#
E7 - 1110 0111 - Octave 10 – E
89 - 1000 1001 - Octave 00 – F#
10 – Duration (1/16 note)
E8 - 1110 1000 - Octave 10 – F
89 - 1000 1001 - Octave 00 – F#

08 - Duration (1/32 note) – start op #42F
E8 - 1110 1000 - Octave 10 – F
8B - 1000 1011 - Octave 00 – G#
E9 - 1110 1001 - Octave 10 – F#
8B - 1000 1011 - Octave 00 – G#
10 – Duration (1/16 note)
EA - 1110 1010 - Octave 10 – G
8B - 1000 1011 - Octave 00 – G#

08 - Duration (1/32 note) – start op #437
EA - 1110 – 1010 Octave 10 – G
81 - 1001 – 0001 Octave 01 – A#
EB - 1110 1011 - Octave 10 – G#
81 - 1001 0001 - Octave 01 – A#
10 – Duration (1/16 note)
F0 - 1111 0000 - Octave 11 – A
91 - 1001 0001 - Octave 01 – A#
20 – Duration (1/8 note)
F2 - 1111 0010 - Octave 11 – B
92 - 1001 0010 - Octave 01 – B

Even het hele blok (#3EB t/m #441) opslaan:

10 E2 82 F2 82 E9 82 E6 92 08 F2 82 18 E9 82 10 E6 82 E6 92 10 E3 83 F3 83 EA 83 E7 93 08 F3 83 18
EA 83 10 E7 83 E7 93 10 E2 82 F2 82 E9 82 E6 92 08 F2 82 18 E9 82 10 E6 82 E6 92 08 E6 89 E7 89 10
E8 89 08 E8 8B E9 8B 10 EA 8B 08 EA 81 EB 81 10 F0 91 20 F2 92

De timing kan iets beter. Ook even kijken of het sneller of langzamer moet.

10 E2 82 F2 82 E9 82 E6 92 08 F2 82 10 E9 82 18 E6 82 10 E6 92
10 E3 83 F3 83 EA 83 E7 93 08 F3 83 10 EA 83 18 E7 83 10 E7 93
10 E2 82 F2 82 E9 82 E6 92 08 F2 82 10 E9 82 18 E6 82 10 E6 92
08 E6 89 E7 89 10 E8 89 08 E8 8B E9 8B 10 EA 8B 08 EA 81 EB 81 10 F0 91 20 F2 92

Dat wordt dan (blok één langer van #3EB t/m #442):

10 E2 82 F2 82 E9 82 E6 92 08 F2 82 10 E9 82 18 E6 82 10 E6 92 E3 83 F3 83 EA 83 E7 93 08 F3 83 10
EA 83 18 E7 83 10 E7 93 E2 82 F2 82 E9 82 E6 92 08 F2 82 10 E9 82 18 E6 82 10 E6 92 08 E6 89 E7 89
10 E8 89 08 E8 8B E9 8B 10 EA 8B 08 EA 81 EB 81 10 F0 91 20 F2 92 00

Ja dat klinkt nu goed.

Maar dan zodanig gaan inpassen dat de drie rivets tunes blokken samen één blok gaan vormen (lang genoeg?) en dan bij beëindigen rivets altijd hetzelfde muziekje spelen. En dan ook pcaman sprite tonen wanneer DK naar beneden gevallen is.

Even kijken of het nog beter klinkt als de bovenste stem een octaaf naar beneden wordt getransponeerd.

10 D2 82 E2 82 D9 82 D6 92 08 E2 82 10 D9 82 18 D6 82 10 D6 92 D3 83 E3 83 DA 83 D7 93 08 E3 83
10 DA 83 18 D7 83 10 D7 93 D2 82 E2 82 D9 82 D6 92 08 E2 82 10 D9 82 18 D6 82 10 D6 92 08 D6 89
D7 89 10 D8 89 08 D8 8B D9 8B 10 DA 8B 08 DA 81 10 DB 81 10 E0 91 20 E2 92

Nee dat klinkt niet goed.

Nu op een juiste plek inbouwen.

```
; Pattern $11, Rescued Pauline (odd level)
407: 20 80 DC 98 E0 9A E2 9C 20 90 E4 88 10 E3 90 A4
417: 20 88 DA 8A EA 84 10 E8 8A A4 20 84 E2 82 E2 87
427: 16 E8 88 0A A3 10 E4 82 A0 15 C8 80 2B CA 83 40
437: CB 80 00
```

```
; Pattern $12, Rescued Pauline (unused variant)
43A: 10 E8 A4 E8 A4 E6 A2 E6 A2 E4 A0 E4 A0 E2 9C E2
44A: 9C 10 E0 90 A0 E4 88 A0 E4 80 A0 E4 88 A8 20 EA
45A: 96 E8 90 40 F0 80 00
```

```
; Pattern $13, Rescued Pauline (even level)
461: 10 E8 A4 E8 A4 E6 A2 E6 A2 E4 A0 E4 A0 E2 9C E2
471: 9C 20 E0 90 10 88 08 A0 A2 20 E4 90 E0 88 20 E2
481: 92 10 8A 08 A2 A4 20 E6 92 E2 8A 20 E8 88 10 82
491: A8 EA 88 A8 E6 82 A4 E8 80 2B 83 40 80 00
```

Al deze patterns vervangen door het nieuwe pattern. Dus laten starten op 407. En dan pattern 11 wat gebruikt wordt voor einde level (rivets) voor oneven levels voor zowel even als oneven levels gebruiken.

Hij zet bij L=01 nu #0C in het sound address (#608A). Dit ook voor de andere levels doen.

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

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

Dus aanpassen:

```
191E 360C LD (HL),#0C ; else play also sound for odd numbered rivets
```

Ja dat gaat goed.

Nu versie v0.77 opgeslagen.

Nu nog pacman weergeven bij einde level bij jumpman en Pauline.

Jumpmans positie en kijkrichting hangen af van de kant waarop het level beëindigd wordt. Ook Pauline kijkt de kant van jumpman op. Maar wanneer eindigen vanaf links dan staat het hartje niet tussen jumpman en Pauline. Maar dat is vreemd want dat is in het originele spel wel zo. Waarom is dat anders? Dat komt door extra code die we toegevoegd hebben om bij beëindigen van een board met power pellet mode nog aan, pacman weer vervangen wordt door jumpman. We zouden ook gewoon de heart sprite kunnen vervangen door de pacman sprite?

Ja dat zou kunnen. En dan de sprite waarbij pacman naar links kijkt (#D5) en dan ipv gebroken hartje de sprite dat hij naar rechts kijkt (#55). Met kleur van blok (#00).

```
1920 23 INC HL ; HL := #608B = sound duration
1921 3603 LD (HL),#03 ; set duration to 3
1923 21236A LD HL,#6A23 ; load HL with heart sprite
1926 3640 LD (HL),#40 ; set heart sprite Y position
1928 2B DEC HL ; decrement HL
1929 3609 LD (HL),#09 ; set heart sprite color
192B 2B DEC HL ; decrement HL
192C 3676 LD (HL),#76 ; set heart sprite
192E 2B DEC HL ; decrement HL
192f 368f LD (HL),#8f ; set heart sprite X position
1931 3A0362 LD A,(#6203) ; load A with mario X position
1934 fe80 CP #80 ; is mario on the left side of the screen?
1936 d0 RET nc ; yes, return
```

Voor rivets aanpassen:

```
1926 3648 LD (HL),#48 ; set pacman Y-position
1929 3600 LD (HL),#00 ; set pacman color
192C 36D5 LD (HL),#D5 ; set pacman sprite left facing
```

```
192F 3693 LD (HL),#93 ; set pacman X-position
```

Ja dat is zo goed.

Nu ook aanpassen voor de andere boards.

```
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
```

Aanpassen:

```
170E 3682 LD (HL),#82 ; set pacman X-position
1711 36D5 LD (HL),#D5 ; set pacman sprite facing left
1714 3600 LD (HL),#00 ; set pacman color
1717 3630 LD (HL),#30 ; set pacman Y-position

174E 21216A LD HL,#6A21 ; load HL with heart sprite
1751 34 INC (HL) ; change heart to broken
```

Aanpassen:

```
174E C3A62F JP,#2FA6 ; jump to additional code – jump to label AAAA
1751 00 NOP
```

en:

```
2FA6 21216A LD HL,#6A21 ; load HL with pacman sprite
2FA9 3655 LD (HL),#55 ; set pacman sprite facing right
2FAB C35217 JP #1752 ; jump back
```

Ja, dat is allemaal goed.

Maar het geluid van de springs is weer terug. Nogmaals uitzetten.

Stuiter geluid springs is hetzelfde als pellet eten. Stuitergeluid uitzetten.

Dat lijkt hier te zitten:

```
2E9C 21AA39 LD HL,#39AA ; load HL with start of table data
2E9F 3E03 LD A,#03 ; load sound duration of 3
2EA1 328360 LD (#6083),A ; play sound for bouncer
2EA4 C34B2E JP #2E4B ; jump back
```

Aanpassen:

```
2E9F 0000000000 5xNOP ; do not play bouncer sound
```

Ja, dat is nu weer uit.

Nu versie v0.78 opgeslagen.

Nog één keer een aantal levels testen met barrels: géén kill screen en wel kill screen op level L=21.

Testen met L=07, L=12, L=17, L=20 en L=21.

Aanpassen start level om te testen:

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

```
095E 14 65 3A 01 00 00 00
```

Ja dat gaat goed.

Nu de invincibility uitzetten.

Herstellen:

```
1E9F 3AE264 LD A,(#64E2) ; load power pellet mode active indicator  
1EA2 FE01 CP #01 ; is the power pellet mode active?  
1EA4 C8 RET Z ; yes, return  
;  
1EA5 CD0828 CALL #2808 ; check for collisions with hostile sprites  
1EA8 C9 RET ; return
```

Op #1EA5 waren drie NOP's gezet om invincibility aan te zetten.

Is nu helemaal goed.

Nu versie v0.79 opgeslagen.

Gaat nog niet goed bij naam ingeven bij een hoger level (getest bij level 18). Laat een score zien met allemaal nullen en geeft een jumpman in verkeerde kleuren linksboven in scherm. Je kunt geen initialen opgeven.

De score wordt goed opgegeven, maar de rest niet:



Even kijken of dat komt door level boven de tien. Testen met level L=06, L=09, L=10 en L=15.

Ja het lijkt er wel op. Levels L=06 en L=09 gaan goed en level L=10 en L=15 gaan fout.

Dit is de code die het level in de score opneemt:

```

257C 3610      LD   (HL),#10      ; store #10 into memory at (HL) – label AAAA
257E 23        INC  HL          ; next HL
257F 23        INC  HL
;
2580 3A2962      LD   A,(#6229)    ; load A with level nr
2583 010AFF      LD   BC,#FF0A    ; B: = #FF, C := #0A (10 decimal)
;
2586 04        INC  B          ; increment B – label CCCC
2587 91        SUB  C          ; subtract 10 decimal
2588 30FB      JR   NC,#2586    ; not carry, loop again – jump to label CCCC
;
258A 81        ADD  A,C        ; add 10 back to A to get a number from 0 to 9
258B 77        LD   (HL),A      ; store level nr (singles) into memory at (HL)
258C 2B        DEC  HL          ; next HL
;
258D 78        LD   A,B        ; load A with B (number of tens)
258E 77        LD   (HL),A      ; store levelnr (tens) into memory at (HL)
258F 23        INC  HL          ; next HL
2590 23        INC  HL          ; next HL
;
2591 362C      LD   (HL),#2C      ; store hyphen into memory at (HL)
2593 23        INC  HL          ; next HL
;
2594 3A2E62      LD   A,(#622E)    ; load A with board nr
2597 3C        INC  A          ; increment A
2598 77        LD   (HL),A      ; store board nr into memory at (HL)
2599 23        INC  HL          ; next HL
;
259A 0608      LD   B,#08        ; for B = 1 to #08 – label BBBB
;
```

```

259C 3610      LD    (HL),#10      ; store #10 into memory at (HL)
259E 23        INC   HL          ; next HL
259F 10FB      DJNZ  #259C      ; next B – jump back to label BBBB
;
25A1 C3F213    JP    #13F2      ; jump back

```

Maar dat is overgenomen van DK. Kijken of we het probleem daar ook hadden. Heel vaag komt me dit wel voor (met dat patroon van nullen).

Dat komt door de JR op #2588. XDie spring te ver terug. Hij moet #04 terugspringen en dat is dan een sprong van #FC en niet van #FB.

Aanpassen:

```
2588 30FC      JR    NC,#2586      ; not carry, loop again – jump to label CCCC
```

Ja, dat gaat nu wel goed.

Terugzetten van het startlevel:

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

Nu versie v0.80 opgeslagen.

En dat is versie finale versie v1.00.

Eventjes speciale aanpassing voor Pieter die extra pellets ziet in het tunnels board.

```

06 EF F0 10 F0      ; pellets bottom girder
06 EF A0 10 A0      ; pellets second girder → 06 EF 78 10 78
06 EF C8 10 C8      ; pellets bottom conveyor
06 EF 50 10 50      ; pellets top conveyor girder
06 F7 78 88 78      ; pellets top right conveyor next to oil can → 06 F7 A0 88 A0 → 06 EF A0 88 A0
06 77 78 08 78      ; top left conveyor next to oil can → 06 77 A0 08 A0 → 06 77 A0 10 A0

```