

## Make jumpman invincible

```
19B3 CD0828 CALL #2808 ; check for collisions with hostile sprites [set to NOPS to make mario invincible to enemy sprites]
```

Dus #19B3 3xNOP maken.

De trappen weggehaald door in de graphic files de trappen weg te poetsen.

Nu alle screens hetzelfde palette geven. Eerst een standaard palette maken door het palette van het barrels screen aan te passen. Naar een variant met een donker blauwe look (al een beetje donker maken, terwijl de girders nog wel zichtbaar zijn).

Eerst uitzoeken hoe het palette werkt. Is mij nog onduidelijk.

In ieder geval:

```
1922 ROM_LOAD( "c-2k.bpr", /* palette low 4 bits (inverted) */
1923 ROM_LOAD( "c-2j.bpr", /* palette high 4 bits (inverted) */
1924 ROM_LOAD( "v-5e.bpr", /* character color codes on a per-column basis */
```

En in de d\_dkong.cpp een routine die het palette samenstelt (met rgb definities):

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

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

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

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

Als we kijken in de file c-2j.br (high 4 bits inverted) en c-2k.bpr (low 4 bits inverted) dan zien we:

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

Als het 64 paletten zijn, dan staat op elke regel 4 paletten: steeds blokken van 4 beginnend met 0F.

Proefondervindelijk blijkt dat de volgende blokken (in c-2j.br en c-2k.bpr) gerelateerd zijn aan de kleuren op de barrels. En dat voor vier verschillende rijen op het scherm:

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

Dit gaan aanpassen en kijken wat er dan gebeurt. Palette heeft vier kleuren. Zijn dat de vier waarden binnen een groep? En hoe dan low en high bits en omzetting naar rgb. Daarna kijken of we er blauwtinten mee kunnen maken. Even testen door een ladder terug te zetten als een soort vlag met drie kleuren.

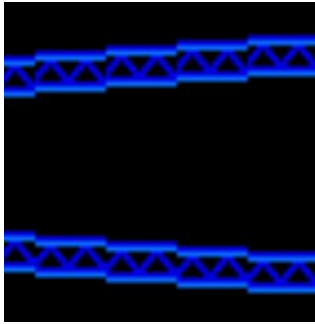
0F 01 07 0E : de 01 is de lichtrood, de 07 is de donkerrood, de 0E is de turquoise

Donkerblauw krijg je door: 2j=0F/2k=0D.

Blauw krijg je door: 2j=0F/2k=00.

Met deze kleuren de girders tekenen: lichtrood vervangen door de donkerblauw en de donkerrood vervangen door de blauw.

De combinatie: 2j: 0F 0F 0F 0E en 2k: 0F 0D 00 00 geeft leuke kleur girders:



Deze kleurcombinatie heb je hiermee op barrels en op springs.

Nu vaststellen welke velden gebruikt worden voor pies en rivets.

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

Groen: Barrels, Springs en Intro

Rood: Pies en Progression

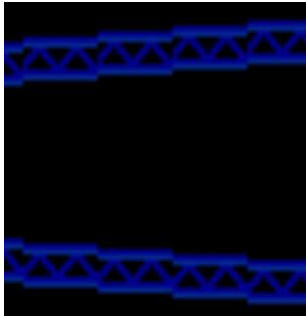
Blauw: Rivets en Titel

Dus nu nog de volgende uitdagingen:

- Als speelscherm aanpassen dan pas je ook een intro, progression of titel scherm aan. Dus hiertoe gebruik maken van twee schermen en eventuele bijeffecten herstellen.
- De bonus timer wordt ook beïnvloed door het palette, dus donker = geen timer meer. Dus in een palette een kleur behouden en deze gebruiken voor de bonus timer weergave. Dit dan wel de kleur houden van de getallen. Anders moet je die ook aanpassen, maar dat heeft ook effect op de getallen van de score e.d.

Nog donkerder blauw krijg je door: 2j=0F/2k=0A.

De combinatie: 2j: 0F 0F 0F 0E en 2k: 0F 0E 0A 00 geeft nog donkerder blauwe girders:



Hoe gaan we het spel vormgeven?

- Smalle girders?
- Wel/geen ladders zichtbaar?
- Wat zichtbaar in een flits (alleen girders of ook ladders)?

Voorstel:

Dus met girders en trappen zichtbaar beginnen. Alles in een donkerblauwe kleur: palette van girders.

Afhankelijk van het level op een bepaald moment donker maken: palette van rivets.

Afhankelijk van level met interval flitsen waarin girders en trappen even zichtbaar zijn.

Terug naar de oorspronkelijke graphics (met de trappen).

De trap buitenkant gelijk maken aan de donkere blauw en de treden aan de iets lichtere blauw.

De kleurcombinatie voor de barrels is de volgende geworden:

2j: 0F 0F 0F 0E en 2k: 0F 0D 0A 00.

De girders en trappen twee tinten donker blauw en de bonus timer de turquoise kleur.

Interessant: de bonus timer verandert op gegeven moment van kleur. Hoe werkt dat?

Code om het palette aan te passen:

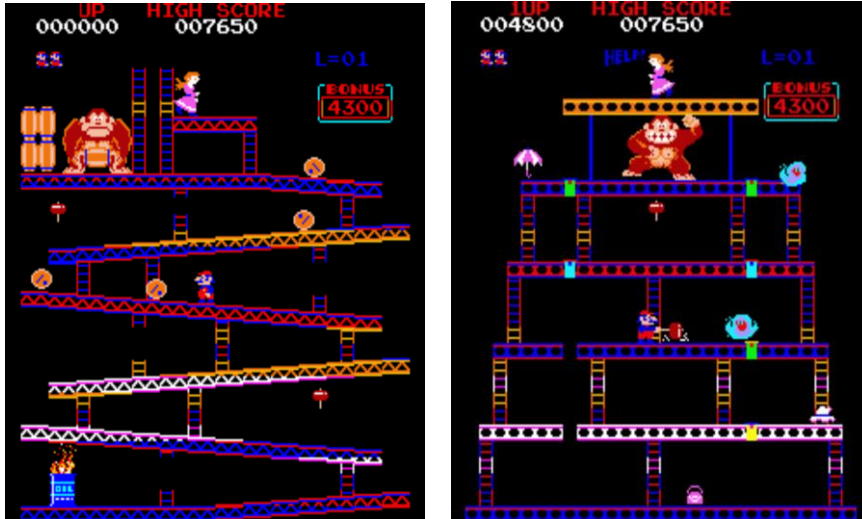
```
3F30: 21867D LD HL,REG_PALETTE_A
3F33: 3600 LD (HL),#00 ; clear palette bank selector
3F35: 23 INC HL
3F36: 3600 LD (HL),#00 ; clear palette bank selector
3F38: C9 RET
```

En

```
19C2: CD303F CALL #3F30 ; call additional code to switch palette
```

Dit eventjes altijd laten uitvoeren en testen of het switchen van palettes werkt:

Dit gaat goed: met #00 en #00 wordt het palette het hi score screen geladen. Dat geeft het volgende resultaat:



Veranderen naar het palette van rivets. Deze daarna aanpassen naar de dark variant. Palette van rivets is: Palette bank A #01 en Palette bank B #01.

```
3F30: 21867D LD HL,REG_PALETTE_A
3F33: 3600 LD (HL),#01 ; clear palette bank selector
3F35: 23 INC HL
3F36: 3600 LD (HL),#01 ; clear palette bank selector
3F38: C9 RET
```

Palette van rivets aanpassen naar dark: 2j: 0F 0F 0F 0E en 2k: 0F 0F 0F 00

Dit werkt goed. Het scherm gaat over op zwart. En de bonus timer blijft doortikken. Nog wel eventjes de bonus graphics op de lichte blauw zetten.

Gedaan. Ziet er goed uit. Alleen wanneer de bonus timer onder de 1000 komt dan verdwijnt de timer. Kleurcode klopt dan dus niet. Dat eruit halen. Dus wel het geluid, maar niet meer het veranderen van kleur. Wellicht laten knipperen een optie?

Nu alle screens laten beginnen met het palette van barrels: Palette bank A #00 en Palette bank B #01.

Voor pies:

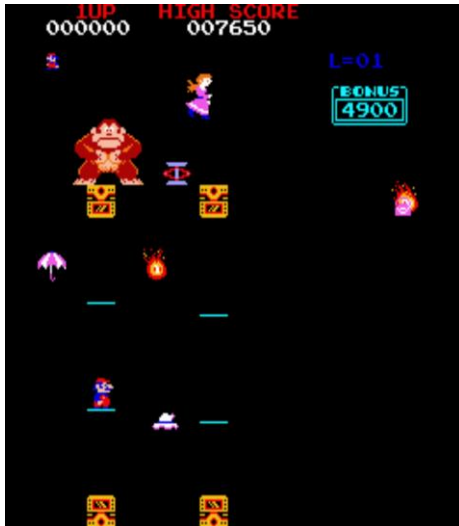
```
0CE2 21867D LD HL,REG_PALETTE_A ; load HL with palette bank selector
0CE5 3600 LD (HL),#00 ; set palette bank selector
0CE7 23 INC HL ;
0CE8 3601 LD (HL),#01 ; clear palette bank selector
```

Voor rivets:

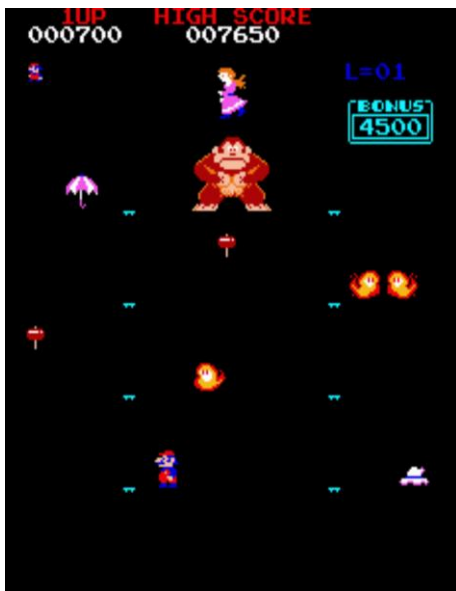
```
0CB9 21867D LD HL,REG_PALETTE_A ; load HL with palette bank selector
0CBC 3600 LD (HL),#00 ; set palette bank selector
```

Rivets zet alleen de Palette bank A goed, de Palette Bank B staat al goed.

Gaat allemaal goed. Maar de elevator plateaus zijn nu ook donker geworden. Dat is niet handig. Deze aanpassen naar de turquoise kleur. Opgelost: er blijft alleen een streep over als elevator.



De rivets ook in zo'n streep veranderen. Gedaan:



Eerst eventjes het veld laten knipperen.

```
3F30: 3A8563 LD A,(#6385) ; load A with blink counter
3F33: 3C INC A ; increment blink counter
3F34: 328563 LD (#6385), A ; store blink counter
```

```

;
3F37: FE10      CP #10          ; is blink counter #10?
3F39: CA943F    JP Z, #3F94      ; yes: make screen dark – jump to label AAAA
;
3F3C: FEC0      CP #C0          ; is blink counter #C0?
3F3E: CA9A3F    JP Z, #3F9A      ; yes: make screen visible – jump to label BBBB
;
3F41: FED0      CP #D0          ; is blink counter greater than #D0?
3F43: D8        RET C           ; no return
3F44: 3E00      LD A, #00        ; load A with #00
3F46: 328563    LD (#6385), A    ; store blink counter
3F49: C9        RET             ; return
;
...
;
3F94: 21867D    LD HL,REG_PALETTE_A ; – label AAAA
3F97: 3601      LD (HL),#01      ; set palette to dark
3F99: C9        RET             ; return
;
3F9A: 21867D    LD HL,REG_PALETTE_A ; – label BBBB
3F9D: 3600      LD (HL),#00      ; set palette to visible
3F9F: C9        RET             ; return

```

Veld dubbel laten knippen:

```

3F30: 3A8563    LD A,(#6385)      ; load A with blink counter
3F33: 3C        INC A             ; increment blink counter
3F34: 328563    LD (#6385), A          ; store blink counter
;
3F37: FED8      CP #D8          ; is blink counter #D8?
3F39: CA943F    JP Z, #3F94      ; yes: make screen dark – jump to label AAAA
;
3F3C: FEC0      CP #C0          ; is blink counter #C0?
3F3E: CA9A3F    JP Z, #3F9A      ; yes: make screen visible – jump to label BBBB
;
3F41: FEC8      CP #C8          ; is blink counter #C8?
3F43: CA943F    JP Z, #3F9A      ; yes: make screen visible – jump to label AAAA
;
3F46: FED0      CP #D0          ; is blink counter #D0?
3F48: CA9A3F    JP Z, #3F9A      ; yes: make screen visible – jump to label BBBB
;
3F4B: FEF0      CP #F0          ; is blink counter greater than #F0?
3F4D: D8        RET C           ; no return
3F4E: 3E00      LD A, #00        ; load A with #00
3F50: 328563    LD (#6385), A    ; store blink counter
3F53: C9        RET             ; return

```

Aan het einde van het level ook het palette weer eventjes goedzetten.

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

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

Dit vervangen door;

```
1E85 C38A3F JP #3F8A ; jump to additional code
1E88-1E8B NOP's
```

```
3F8A: CD9A3F CALL #3F9A ; make screen visible
3F8D: 3E16 LD A,#16 ; else A := #16
3F8F: 320A60 LD (GameMode2),A ; store into game mode2
3F92: E1 POP HL ; pop stack to get higher address
3F93: C9 RET ; return to a higher level [returns to #00D2]
```

Dit gaat helemaal goed.

Als jumpman dood gaat ook het palette weer eventjes goedzetten:

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

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

Dit vervangen door:

```
12A3 CD833F CALL #3F83 ; jump to additional code
12A6 3E03 LD A,#03 ; load A with duration of sound
12A8 328860 LD (#6088),A ; play death sound
12AB C9 RET ; return
```

```
3F83: CDBD30 CALL #30BD ; clear sprites ?
3F86: CD9A3F CALL #3F9A ; make screen visible
3F89: C9
```

Dit gaat helemaal goed.

Als eenheid van pauze de helft van de huidige tijd aanhouden. Is nu #F0, dus dat zou bijvoorbeeld #80 kunnen zijn. Dan in level L=01 t/m L=03 een random keuze van 1x of 2x deze pauze. Dan in level L=04 t/m L=11 een random keuze van 1x, 2x 3x of 4x deze pauze. Dan in level L=12 t/m L=22 een random keuze van 3x, 4x, 5x of 6x deze pauze.



Nee, anders doen. Drie tellers gebruiken. Variabele #6385 gebruiken als de flash timer 1. Deze is altijd even lang. Variabele #638D gebruiken als flash timer 2. Deze is afhankelijk van het level # maal 8. Variabele #638E gebruiken als item timer die het licht een tijdje aan houdt wanneer een item gepakt is.

Deze drie tellers aan het begin van een screen met de juiste waarden initialiseren.

```

3F30: 3A8E63    LD A,(#638E)      ; load A with item timer
3F33: FE00      CP #00          ; is item timer #00?
3F35: CA473F    JP Z, #3F47      ; yes, no item grabbed lately – jump to label FFFF
;
3F38: 3D        DEC A          ; decrement item timer
3F39: 328E63    (#638E),A      ; store item timer
;
3F3C: FE8F      CP #8F        ; is item timer #8F?
3F3E: CA9A3F    JP Z, #3F9A      ; yes: make screen visible – jump to routine
;
3F41: FE00      CP #00        ; is item timer #00?
3F43: CA943F    JP Z, #3F94      ; yes: make screen dark – jump to routine
;
3F46: C9        RET          ; return
;
3F47: 3A8D63    LD A,(#638D)      ; load A with flash timer 2 – label FFFF
3F4A: FE00      CP #00        ; is flash time 2 timer #00?
3F4C: CA543F    JP Z, #3F54      ; yes, flash timer 2 ran out – jump to label GGGG
;
3F4F: 3D        DEC A          ; decrement flash timer 2
3F50: 328D63    (#638D),A      ; store flash timer 2
3F53: C9        RET          ; return
;
3F54: 3A8563    LD A,(#6385)      ; load A with flash timer 1 - Label GGGG
3F57: 3C        INC A          ; increment flash timer 1
3F58: 328563    LD (#6385), A      ; store flash timer 1
;
3F5B: FED8      CP #D8        ; is flash timer 1 #D8?
3F5D: CA943F    JP Z, #3F94      ; yes: make screen dark – jump to routine and return
;
3F60: FEC8      CP #C8        ; is flash timer 1 #C8?
3F62: CA943F    JP Z, #3F94      ; yes: make screen dark – jump to routine and return
;
3F65: FEC0      CP #C0        ; is flash timer 1 #C0?
3F67: CAC83F    JP Z, #3FC8      ; yes: make screen visible - jump to label HHHH
;
3F6A: FED0      CP #D0        ; is flash timer #D0?
3F6C: CAC83F    JP Z, #3FC8      ; yes: make screen visible – jump to label HHHH
;
3F6F: FEF0      CP #F0        ; is flash timer greater than #F0?
3F71: D8        RET C          ; no return

```

```

;
3F72: 3E00    LD  A, #00          ; load A with #00
3F74: 328563    LD  (#6385), A      ; store flash timer 1
;
3F77: 3A2962    LD  A, (#6229)      ; load A with level number
3F7A: 17        RLA          ; rotate left: *2
3F7B: 17        RLA          ; rotate left: *2 (*4 combined)
3F7C: 17        RLA          ; rotate left: *2 (*8 combined)
3F7D: 328D63    LD  (#638D), A      ; store flash timer 2
;
3F80: C9        RET          ; return

```

en

```

3FC8: 3E03    LD  A, #03          ; load sound duration of 3 – label HHHH
3FCA: 328260    LD  (#6082), A      ; play boom sound using sound buffer
3FCD: CA9A3F    JP  Z, #3F9A        ; yes: make screen visible

```

De flash timer 1 (#6385), flash timer 2 (#638D) en item timer (#638E) aan begin van screen initialiseren:

```

; other screens return here
0CC6 CDA70D    CALL  #0DA7          ; draw the screen

```

Vervangen door:

```

0CC6: CDD03F    CALL  #3FD0          ; call additional code – jump to label EEEE

```

En

```

3FD0: CDA70D    CALL  #0DA7          ; draw the screen – label EEEE
3FD3: 3E00    LD  A  #00          ; A := 0
3FD5: 328563    LD  (#6385), A      ; store flash timer 1
3FD8: 328D63    LD  (#638D), A      ; store flash timer 2
3FDB: 328E63    LD  (#638E), A      ; store item timer
3FDE: C9        RET          ; return

```

Dit gaat goed. Geeft geluid bij flitsen.

Nog wel testen met een veel hoger level: dan ook langere pauze? En hoe lang dan? Eventueel fine tunen?

Aanpassen start level:

```

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

```

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

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

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

L=01 ongeveer elke 4 seconden knippen.

L=21 ongeveer elke 7 seconden knippen.

Dus dat gaat goed. Langer maken kan niet echt. \*16 gaat niet goed omdat dan bits er door de rotate left er uit schuiven. En dan wordt de tijd wellicht weer korter.

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

; kong is beating his chest

```
0445 21F739 LD HL,#39F7 ; start of table data
0448 CD4E00 CALL #004E ; update kong's sprites
044B 3E03 LD A,#03 ; load sound duration of 3
044D 328260 LD (#6082),A ; play boom sound using sound buffer
```

Dus waarschijnlijk uitzetten door 044B t/m 044F NOP's te maken. Ja, dat klopt. Uitgezet.

Item timer zetten wanneer een item gepakt wordt. Check of item gepakt wordt, gebeurd hier:

; bonus item has been picked up

```
19FA 2D DEC L ; decrease L. HL now has the starting address of the sprite
19FB 224363 LD (#6343),HL ; store into this temp memory. read from at #1E18
19FE AF XOR A ; A := 0
19FF 324263 LD (#6342),A ; store into ????. read from at #1DD6
1A02 3C INC A ; A := 1
1A03 324063 LD (#6340),A ; store into #6340
1A06 C9 RET ; return
```

Dit dan aanpassen in:

```
1A03 C3DF3F JP #3FDF ; jump to additional code
```

en

```
3FDF: 324063 LD (#6340),A ; store into #6340
3FE2: 3E90 LD A #90 ; A := 90
3FE4: 328E63 LD (#638E), A ; store item timer
3FE7: C9 RET ; return
```

Dat werkt ook goed. Erg leuk aspect van deze hack. Zeker als de items ook nog veranderd worden in lichtgerelateerd items zoals zaklantaarn, lantaarn en kaars.

## De HIGH SCORE aanpassen naar INTO THE DARK

```
368A:                                96 76 17 11 1D                ..GAM
3690:  15 10 10 1F 26 15 22 3F 94 76 20 1C 11 29 15 22  E..OVER...PLAYER
36A0:  10 30 32 31 3F 94 76 20 1C 11 29 15 22 10 30 33  .<I>...PLAYER.<2
36B0:  31 3F 80 76 18 19 17 18 10 23 13 1F 22 15 3F 9F  >...HIGH.SCORE..
36C0:  75 13 22 15 14 19 24 10 10 10 10 3F 5E 77 18 1F  .CREDIT.....HO
36D0:  27 10 18 19 17 18 10 13 11 1E 10 29 1F 25 10 17  W.HIGH.CAN.YOU.G
36E0:  15 24 10 FB 10 3F 29 77 1F 1E 1C 29 10 01 10 20  ET.?....ONLY.1.P
```

#36B2: C0 76 19 1E 24 1F 10 24 18 15 10 14 11 22 1B 3F

Daartoe wel de CREDIT verplaatsen:

#36C3: adres van de verplaatste CREDIT: aanpassen op: #3655 wordt dan C336

## Het titelscherm aanpassen

De KONG letters 1 positie naar beneden verschuiven:

```
3D59: 05 30 77 05 10 77 02 F1 76 02 D0 76 02 D3 76    ; K
3D68: 05 90 76 05 70 76 01 50 76 01 54 76 05 30 76    ; O
3D77: 05 F0 75 02 D1 75 02 B2 75 05 90 75              ; N
3D83: 03 51 75 05 30 75 01 10 75 01 14 75              ; G (part 1)
3D8F: 01 F0 74 01 F2 74 01 F4 74 02 D2 74              ; G (part 2)
3D9B: 00                                                  ; end code
```

En het TM symbool weghalen.

```
081C CD243F CALL #3F24 ; draw TM logo onscreen [patch? orig japanese had 3 NOPs here]
```

Hiermee wordt het stukje extra code dat het TM-symbool tekent ook vrijgemaakt:

#3F24 t/m #3F2F.

Dus op #081C weer 3 NOP's maken.

Dan kan het stuk van #3F00 t/m #3F23 ook vrijgemaakt worden en gebruikt worden voor de teksten in het titelscherm.

```
3687: 00 3F          1E      ; #3F00 "(C) 1981"
3689: 09 3F          1F      ; #3F09 "NINTENDO OF AMERICA"
```

In #3F00 de copyright notice maken:

© 1981,2020 NINTENDO

5C 77 49 4A 10 01 09 08 01 43 02 00 02 00 10 1E 19 1E 24 15 1E 14 1F 3F

En de plek van de checksum ook aanpassen: INTEND: start op 3F10:

```
2441 21103F LD HL,#3F10 ; load HL with ROM area that has NINTENDO written
```

En dan in #3FE8 de titel maken:  
INTO THE DARK

```
CE 76 19 1E 24 1F 10 24 18 15 10 14 11 22 1B 3F
```

En de verwijzing ernaar toe aanpassen:

```
3689: E8 3F
```

Nu nog de Donkey Kong een regel naar beneden verplaatsen.

Wordt hier gedaan:

```
0820 210869 LD HL,#6908 ; load HL with start of kong sprite X pos
0823 0E44 LD C,#44 ; load C with offset to add X
0825 FF RST #38 ; draw kong in new position
0826 210869 LD HL,#690B ; load HL with start of kong sprite Y pos
0829 0E78 LD C,#78 ; load C with offset to add Y
082B FF RST #38 ; draw kong
082C C9 RET ; return
```

Kan door aanpassen:

```
0829: 0E80 LD C,#80 ; load C with offset to add Y
```

Liefste een versie toevoegen aan het titel scherm. Eerste versie opnemen is v1.00.

De 1 is nog unused:

```
364B: 8B 36 0 ; #368B "GAME OVER"
364D: 01 00 1 ; unused ?
364F: 98 36 2 ; #3698 "PLAYER <I>"
3651: A5 36 3 ; #36A5 "PLAYER <II>"
3653: B2 36 4 ; #36B2 "HIGH SCORE"
3655: BF 36 5 ; #36BF "CREDIT"
3657: 06 00 6 ; unused ?
3659: CC 36 7 ; #36CC "HOW HIGH CAN YOU GET?"
```

Deze laten verwijzen naar het stukje met spaties in de Name:

```

3710:  24 24 1F 1E 3F 27 76 20 25 23 18 3F 06 77 1E 11  TTON...PUSH...NA
3720:  1D 15 10 22 15 17 19 23 24 22 11 24 19 1F 1E 3F  ME.REGISTRATION.
3730:  88 76 1E 11 1D 15 2E 3F E9 75 2D 2D 2D 10 10 10  ..NAME:----...
3740:  10 10 10 10 10 10 3F 0B 77 11 10 12 10 13 10 14  ....A.B.C.D
3750:  10 15 10 16 10 17 10 18 10 19 10 1A 3F 0D 77 1B  .E.F.G.H.I.J...K
3760:  10 1C 10 1D 10 1E 10 1F 10 20 10 21 10 22 10 23  .L.M.N.O.P.Q.R.S
3770:  10 24 3F 0F 77 25 10 26 10 27 10 28 10 29 10 2A  .T...U.V.W.X.Y.Z

```

Dus daartoe #373D veranderen in 3F.

En dan de versie opnemen in stuk daarna:

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

en

#364D: 3E 37

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

```

080C 111E03 LD DE,#031E ; load task data for text "(C) 1981"
080F CD9F30 CALL #309F ; insert task to draw text
0812 13 INC DE ; load task data for text "NINTENDO OF AMERICA"
0813 CD9F30 CALL #309F ; insert task to draw text
0816 21CF39 LD HL,#39CF ; load HL with table data for kong beating chest
0819 CD4E00 CALL #004E ; update kong's sprites
081C CD243F CALL #3F24 ; draw TM logo onscreen [patch? orig japanese had 3 NOPs here]
081F 00 NOP ; no operation
0820 210869 LD HL,#6908 ; load HL with start of kong sprite X pos
0823 0E44 LD C,#44 ; load C with offset to add X
0825 FF RST #38 ; draw kong in new position
0826 210869 LD HL,#6908 ; load HL with start of kong sprite Y pos
0829 0E78 LD C,#78 ; load C with offset to add Y
082B FF RST #38 ; draw kong
082C C9 RET ; return

```

We hadden de #081C drie NOP's gemaakt (niet meer TM logo tekenen). Daar kan dan mooi een aanroep naar een stuk additionele code (gesplitst ivm ruimtegebrek):

081C: C3BA3F JP #3FBA ; jump to additional code to display version

3FBA: 110103 LD DE,#0301 ; load task data for text "V1.00"

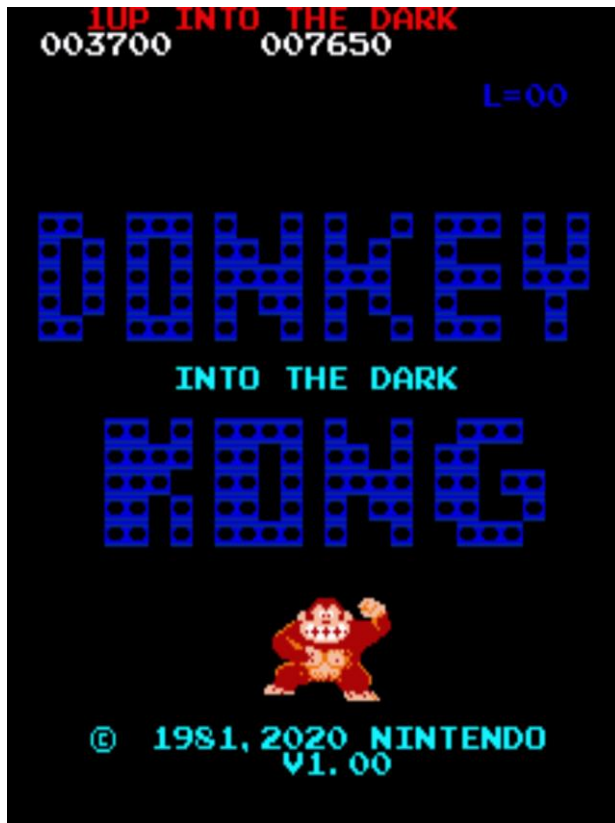
3FBD: C3F83F JP #3FF8

en

3FF8: CD9F30 CALL #309F ; insert task to draw text

3FFA: C31F08 JP #081F ; jump back

Het titelscherm ziet er nu zo uit:



Als het titel scherm geweest is, kan het spel niet meer heel goed opgestart worden. Soms donkere Kong climbing, soms klapt het er meteen uit, soms rare flitsende kleuren.

Dit zat er al in voor het aanpassen van het titelscherm.

Als het introscherm overgeslagen wordt, werkt het allemaal wel goed. Zo te zien wordt het palette bij het Kong climbing scherm wel goed gezet. Ook nog op een andere manier getracht te zetten, maar hetzelfde probleem blijft.

Introscherm overslaan:

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

Dan blijft het probleem wel weg. Althans bij drie pogingen, geen problemen.

Bonus timer verandert onder de 1000 van kleur. Maar nu door aanpassen van het palette, is die kleur zwart/transparant en zie je het dus niet meer.

Dit gebeurt hier:

```
0678 3E03 LD A,#03 ; else load A with warning sound
067A 328960 LD (#6089),A ; set warning sound
```

```

067D 3E70    LD    A,#70        ; A := #70 = color code for red?
067f 328674  LD    (#7486),A      ; store A into #7486 = paint score red (MSB) ?
0682 32A674  LD    (#74A6),A      ; store A into #74A6 = paint score red (LSB) ?

```

De #70 is geen kleurcode, maar is een offset in de graphic code. De nummers 0 t/m 9 staan er twee keer in, in verschillende kleuren. Alleen door palette wijzigingen is er slechts één zichtbaar. Dit oplossen door geen index toe te passen. Dus door de #70 te vervangen door #00.

De bonus timer blijft hierdoor wel zichtbaar, maar verandert niet van kleur. Dat de bonus timer onder de 1000 komt wordt nu alleen duidelijk gemaakt door het geluidje.

Nu de items vervangen door andere items. Is gedaan. Ook de conveyor pulleys zijn van kleur verandert: zelfde kleur als de elevator mounts gemaakt.

Nu de tekst How High Can You Get ? aanpassen.

LET'S GO INTO THE DARK!

```

1C 15 24 3A 23 10 17 1F 10 19 1E 24 1F 10 24 18 15 10 14 11 22 1B 38

```

SWITCH OFF THE LIGHTS !

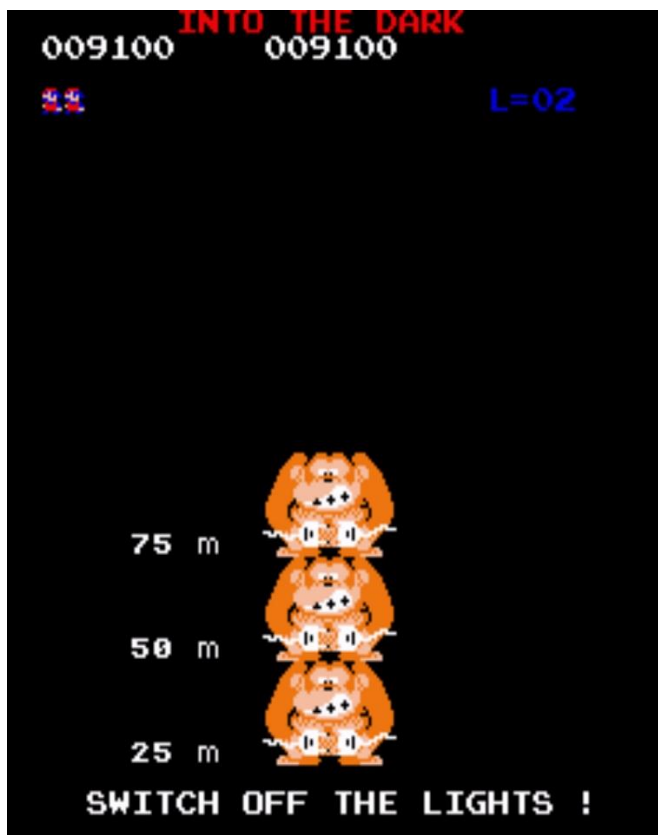
```

23 27 19 24 13 18 10 1F 16 16 10 24 18 15 10 1C 19 17 18 24 23 10 38

```

Gekozen voor de laatste.

Goofy Kong aangepast naar een versie die een stekker lostrekt:





Nogmaals testen met level 17, kijken hoe dat speelt:

Aanpassen start level:

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

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

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

Als starten L=017 dan dit aanpassen naar 11 en 73 3A.

Nu nog de invincibility opheffen.

19B3 CD0828 CALL #2808 ; check for collisions with hostile sprites [set to NOPS to make mario invincible to enemy sprites]