

## Donkey Kong Speedup

Eerst reverse engineering hoe in de Trainer de speed geregeld wordt.

Kijken waar de waarde P1 gelezen wordt en wat er dan mee gedaan wordt.

Wordt gelezen in de IN2 (#7D00):

```
3A007D LD A,(IN2)
```

In DK Barrelboss wordt dat als volgt gelezen:

```
24F9 3A007D LD A,(IN2) ; load A with IN2 – label AAAA
24FC FE08 CP #04 ; is P2 button pressed?
```

Dus debuggen totdat #7D00 de waarde #04 heeft. Of de code doorzoeken op 3A007D en dan kijken of er vlak daarna een CP #04 staat. En dan kijken wat ermee gedaan wordt.

Niet in debugger te zien. Dan kijken in de code of je ergens de 3a007d kunt vinden.

Dat zit op #0072 en op #0101 en op #02A7 en op #02C2.

Kijken met watchpoints op de verschillende stukken en kijken waar de waarde gelezen wordt.

Die op #0072 zit in het standaard spel, maar wel een heel andere afhandeling. Deze wordt ook uitgevoerd tijdens het spelen.

Die op #0101, #02A7 en #02C2 worden niet uitgevoerd tijdens het spelen.

Dus focussen op de #0072. Dat is de volgende code:

0072	ld	a, (\$7D00)	3A 00 7D
0075	and	\$04	E6 04
0077	jr	nz, \$00F0	20 77
0079	ld	a, (\$6005)	3A 05 60

Dus waarde AND'en met #04. Is niet gelijk aan 00 als het #04 is (dus P1 ingedrukt). Dan wordt er gesprongen naar #00F0.

Bij de #00F0 wordt iets gedaan met #6052 en #6053.

De #6052 begint op te lopen wanneer de P1 ingedrukt wordt en wordt weer nu wanneer de P1 losgelaten wordt. De #6053 gaat naar nul als P1 ingedrukt en krijgt de waarde van #6052 wanneer P1 weer losgelaten wordt. Daarmee wordt dus bepaald hoe lang de knop ingedrukt is. Dus ergens wordt deze ook weer uitgelezen en wordt bepaald wat er gedaan moet worden (langzamer, sneller, reset). Zo'n zelfde setje is er ook voor het indrukken van de P2 knop.

Dat is dus al handig om te weten. Dus nu zoeken in de code waar de #6053 weer ingelezen wordt om de handeling uit te voeren.

Dus zoeken naar een LD A,(#6053) dat is in code: 3A 53 60. Kan ik niet vinden. Wellicht een watchpoint op dat adres voor het lezen van de waarde. Nee ook niet. Dat is wel heel vreemd. Waarom wordt die waarde dan gezet. Wellicht via een HL uitlezen. Zoeken naar vullen van HL met #6053. Dat is dan LD HL,#6053 dat is in code 21 53 60. Nou dat is ook niet te vinden.

Enige wat ik dan kan bedenken is dat de daadwerkelijke teller de #6052 is en die ook bepaald wordt om de actie te bepalen (vertragen, versnellen, reset) en dat de #6053 alleen maar gezet wordt zodat in de debugger je kan zien welke waarde het indrukken van P1 tot gevolg had. Soort van monitordoel.

Dan zoeken waar de #6052 uitgelezen wordt. In de code dan 3A 52 60. Dat is op twee plekken. De eerste op #0084 lijkt de afhandeling te doen. In ieder geval wordt de waarde vergeleken met verschillende waarden en dan gesprongen. De tweede op #00F4 is degene die gebruikt wordt tijdens het indrukken van de P1 knop.

Per ongeluk een treffer omdat in de code een CALL gedaan wordt naar #0247. Dat is de volgende code:

0247	ld	(\$6050),a	32	50	60
024A	ld	a,(\$6050)	3A	50	60
024D	ld	b,\$10	06	10	
024F	cp	\$0A	FE	0A	
0251	jr	c,\$0257	38	04	
0253	sub	\$0A	D6	0A	
0255	ld	b,\$01	06	01	
0257	ld	de,\$FFE0	11	E0	FF
025A	ld	hl,\$7500	21	00	75
025D	ld	(hl),\$23	36	23	
025F	add	hl,de	19		
0260	ld	(hl),\$20	36	20	
0262	add	hl,de	19		
0263	ld	(hl),\$14	36	14	
0265	ld	l,\$80	2E	80	
0267	ld	(hl),b	70		
0268	add	hl,de	19		
0269	ld	(hl),a	77		
026A	add	hl,de	19		
026B	ld	(hl),\$00	36	00	
026D	ret		C9		
026F	ld		55		

En daar wordt de SPD naar het scherm geschreven: karakters #23/#20/#14 = SPD. De waarde die er achter gezet wordt, lijkt bepaald te worden door de waarde in #6050. Eens debuggen wat er met die waarde gebeurt tijdens het aanpassen van de snelheid. Ja hoor, bingo. De waarde van #6050 is gelijk aan de tientallen van de speed: dus b.v. #09 = 90% en #0B = 110%.

Dan nu kijken wat er met deze waarde gedaan wordt. Watchpoint in debugger geeft #0193.

Het hele stuk code van #0193 t/m 01F2 lijkt aangepaste code voor het inlezen van de inputs en die lijkt vertraagd/versneld te worden afhankelijk van de waarden in #6050 en #6051.

0193	ld	a,(\$6050)	3A	50	60
0196	ld	l,a	6F		
0197	ld	a,(\$6051)	3A	51	60
019A	add	a,l	85		
019B	ld	(\$6051),a	32	51	60
019E	sub	\$0A	D6	0A	
01A0	jr	c,\$01E5	38	43	
01A2	ld	(\$6051),a	32	51	60
01A5	ld	a,(\$6007)	3A	07	60
01A8	and	a	A7		
01A9	jr	nz,\$01C3	20	18	
01AB	ld	a,(\$7C00)	3A	00	7C
01AE	ld	b,a	47		
01AF	and	\$0F	E6	0F	
01B1	ld	c,a	4F		
01B2	ld	a,(\$6011)	3A	11	60
01B5	cpl		2F		
01B6	and	b	A0		
01B7	and	\$10	E6	10	
01B9	r!a		17		
01BA	r!a		17		
01BB	r!a		17		
01BC	or	c	B1		
01BD	ld	h,b	60		
01BE	ld	l,a	6F		
01BF	ld	(\$6010),hl	22	10	60
01C2	ld	a,b	78		
01C3	ld	hl,\$601A	21	1A	60
01C6	dec	(hl)	35		
01C7	call	\$0057	CD	57	00
01CA	call	\$02A7	CD	A7	02
01CD	call	\$01F3	CD	F3	01
01D0	ld	hl,\$01E0	21	E0	01
01D3	push	hl	E5		
01D4	ld	a,(\$6005)	3A	05	60
01D7	rst	\$28	EF		
01D8	rst	\$18	DF		
01D9	ld	(bc),a	02		
01DA	ld	e,l	5D		
01DB	add	hl,bc	09		
01DC	sbc	a,c	99		
01DD	ld	a,(bc)	0A		
01DE	add	hl,hl	29		
01DF	add	hl,bc	09		
01E0	ld	a,(\$6051)	3A	51	60
01E3	jr	\$019E	18	B9	
01E5	pop	iy	FD	E1	
01E7	pop	ix	DD	E1	
01E9	pop	hl	E1		
01EA	pop	de	D1		
01EB	pop	bc	C1		
01EC	ld	a,\$01	3E	01	
01EE	ld	(\$7D84),a	32	84	7D
01F1	pop	af	F1		
01F2	ret		C9		

In regel #019E wordt een SUB #0A gedaan. Dat lijkt het verschil te maken tussen versnellen en vertragen. Als carry (dus getal is kleiner dan #0A) dan wordt naar het einde gesprongen (zonder input te lezen) en wordt er de VBLANK gezet (#7D84). Maar hoe werkt dat dan met versnellen?

Een keer de snelheid op 120 % zetten en dan watchpoint op #0193 en kijken wat er precies gebeurt met de #6051 en #6052 en het doorlopen van de routine.

Met speed op 100% dan komt het in een loop via 01D7 RST #28 en die springt dan weer naar #0193. Of in ieder geval komt weer bij watchpoint en de executie stopt. Maar wat gebeurt er in het eerste stukje als de speed 110%, 120% en 130% is?

110%: Dan staat de #6050 op #0B en loopt de #6051 steeds van #00 naar #09.

120%: Dan staat de #6050 op #0C en loopt de #6051 steeds langs waarden #01, #03, #05, #07, #09.

130%: Dan staat de #6050 op #0D en loopt de #6051 steeds langs waarden #01, #04, #07, #00, #03, #06, #09, #02, #05, #08.

140%: Dan staat de #6050 op #0E en loopt de #6051 steeds langs de waarden #02, #06, #00, #04, #08.

150%: Dan staat de 6050 op #0F en loopt de #6051 steeds langs de waarden #04 en #09.

Geen idee hoe dat werkt. Wil ik daar nog meer mee doen? Op een gegeven moment ook zeggen dat je iets niet wil doen. Sockmaster heeft dat heel mooi gedaan. Maar gaat boven mijn pet. Daar genoeg mee nemen.

Eventjes anders aanvliegen.

Er is standaard code voor de interrupt routine. Die wordt denk ik 60 keer per seconde doorlopen. Dat resulteert in de frames en elk frame wordt de game uitgevoerd door een soort van switch case:

```
00C6 3A0560 LD A,(GameMode1) ; load A with game mode1

; GameMode1 is 0 when game is turned on, 1 when in attract mode.

RST #28 ; jump based on above:

00CA C3 01 ; #01C3 = startup
00CC 3C 07 ; #073C = attract mode
00CE B2 08 ; #08B2 = credits, waiting
00D0 FE 06 ; #06FE = playing game
```

Maar wat als je de playing game optie twee keer aanroept. Dan gaat het spel twee stappen verder in een frame. Dus dat betekent dat de game twee keer zo snel gaat. Dat zouden we kunnen proberen. Als dat zo is, dan moet je het zo maken dat er een teller is waarmee alleen met een bepaalde frequentie een extra aanroep uitgevoerd wordt. Bijvoorbeeld 1x van de 10 keer. Dan heb je in principe met 10% versneld. En dan het zo maken dat dat de snelheid afhankelijk gemaakt wordt van het level. En dan misschien verschillende varianten met verschillende stappen.

Eerst testen met stukje code waarbij het altijd een tweede keer uitgevoerd wordt.

```
00C6 C3303F JP #3F30 ; jump to additional code
```

en:

```
3F30 3A0560 LD A,(GameMode1) ; load A with game mode1
;
3F33 EF RST #28 ; jump based on above:
;
3F34 C3 01 ; #01C3 = startup
3F36 3C 07 ; #073C = attract mode
3F38 B2 08 ; #08B2 = credits, waiting
3F3A FE 06 ; #06FE = playing game
```

```

;
3F3C  3A0560      LD      A,(GameMode1) ; load A with game mode1
3F3F  FE03        CP      #03           ; Gamemode1 == 3 (playing a game)?
3F41  2003        JR      NZ,BBBB      ; no, skip next step
3F43  CDFE06      CALL    #06FE        ; do main playing game again
;
3F36  C3D200      JP      #00D2        ; continue interrupt routine

```

Dit werkt niet. Hij komt wel steeds op de #3F30, maar komt nooit terug naar de #3F3C. Dus dat werkt  
blijkbaar op een andere manier dan ik dacht.

Ik neem aan dat je de main routine wel extra kan aanroepen want die doet een RET aan het einde.

Ja dat zou moeten kunnen. Maar gaat nu niet goed omdat de RST 28 in feite een JP is. Maar in de  
code wordt hiervoor al een return adress gezet door HL te vullen en deze naar de stack te PUSH'en.  
Dus komt altijd terug naar dat adres (#00D2) en daardoor niet in de rest van de additionele code  
hierboven.

Dus dat moet anders. Je kunt de additionele aanroep wellicht ook iets eerder in de code stoppen?

Even terug naar de oorspronkelijke versie en dan aanpassen:

Aanpassen:

```

00C2  C3303F      JP      #3F30        ; jump to additional code

```

en:

```

3F30  3A0560      LD      A,(GameMode1) ; load A with game mode1
3F33  FE03        CP      #03           ; Gamemode1 == 3 (playing a game)?
3F35  2003        JR      NZ,BBBB      ; no, skip next step
3F37  CDFE06      CALL    #06FE        ; do main playing game again
;
3F3A  21D200      LD      HL,#00D2      ; load HL with return address
3F3D  C3C500      JP      #00C5        ; jump back

```

Ja dit werkt. Gaat twee keer zo snel. Best nog een beetje speelbaar. Wellicht dat sommigen dit wel  
aankunnen, maar dat moeten we gaan testen.

Eerst nu de code aanpassen dat op basis van een teller dit uitgevoerd wordt. Kijken of er ook  
waarden tussen 100% en 200% gezet kunnen worden en hoe goed dat dan te spelen is. Daarna kijken  
of het mogelijk is om de snelheid alleen op te schroeven als je echt aan het spelen bent: dus ook  
afhankelijk maken van gamemode2: dus in how high scherm, in einde kong climbing sequence of  
kong fallen sequence het tempo niet opschroeven. Dat zou bijvoorbeeld kunnen door te testen of  
Gamemode2 gelijk is aan #0C.

En dan later de stijl aanpassen en de snelheid aangeven en wellicht een aantal andere aspecten toevoegen. Zoals bijvoorbeeld een finish flag bij DK in het How High scherm, snelheid in how high scherm, bonus items race gebaseerd maken.

Eerst kijken dat het alleen gedaan wordt bij Gamemode2 = #0C.

Aanpassen:

```

3F30  3A0560      LD    A,(GameMode1)      ; load A with game mode1
3F33  FE03        CP    #03          ; Gamemode1 == 3 (playing a game)?
3F35  200A        JR    NZ,BBBB      ; no, skip next steps – jump to label BBBB
;
3F37  3A0A60      LD    A,(GameMode2)    ; load A with game mode2
3F3A  FE0C        CP    #0C          ; Gamemode2 == C (alive playing a game)?
3F3C  2003        JR    NZ,BBBB      ; no, skip next steps – jump to label BBBB
;
3F3E  CDFE06      CALL #06FE          ; do main playing game again
;
3F41  21D200      LD    HL,#00D2       ; load HL with return address – label BBBB
3F44  C3C500      JP    #00C5         ; jump back

```

Ja dat gaat goed.

Nu kijken hoe je de percentages voor elkaar kunt krijgen.

Met tellertjes kun je het volgende maken:

Tot 10 tellen = 1/10 = 10% versnelling  
 Tot 5 tellen = 1/5 = 20% versnelling  
 Tot 2 tellen = 1/2 = 50% versnelling

En daar dan combinaties van maken?

110% = 10%	=	10 tellen	+	0 tellen
120% = 20%	=	5 tellen	+	0 tellen
130% = 30%	=	10 tellen	+	5 tellen
140% = 40%	=	5 tellen	+	5 tellen
150% = 50%	=	2 tellen	+	0 tellen
160% = 60%	=	2 tellen	+	10 tellen
170% = 70%	=	2 tellen	+	5 tellen

En verder niet gaan. Anders wel heel erg lastig.

Dus eigenlijk combinatie van twee tellers en dan op basis van beide tellers een extra aanroep invoegen.

Even kijken welke twee tellers daarvoor gebruiken:

#64B0 om de twee counterwaardes te definiëren (gezet per level)

#64B1 voor de eerste counter

#64B2 voor de tweede counter

Bijvoorbeeld #64B0 = #A5 = counter 1 tot 10 tellen en counter 2 tot 5 tellen (130%).

Aanpassen:

3F3E CD473F CALL #3F47 ; call to additional code – call label AAAA

en:

```
3F47 3AB064 LD A,(#64B0) ; load A with counter setup – label AAAA
3F4A E60F AND #0F ; mask to get first counter reset value
;
3F4C 47 LD B,A ; load B with first counter reset value
3F4D 21B164 LD HL,#64B1 ; load HL with first counter memory address
3F50 CD673F CALL #3F67 ; call routine – call label CCCC
;
3F53 3AB064 LD A,(#64B0) ; load A with counter setup
3F56 E6F0 AND #F0 ; mask to get second counter reset value
3F58 1F RRA
3F59 1F RRA
3F5A 1F RRA
3F5B 1F RRA
3F5C FE00 CP #00 ; A == 0 ?
3F5E C8 RET Z ; yes, skip this counter – return
;
3F5F 47 LD B,A ; load B with second counter reset value
3F60 21B264 LD HL,#64B2 ; load HL with second counter memory address
3F63 CD673F CALL #3F67 ; call routine – call label CCCC
;
3F66 C9 RET ; return
;
3F67 34 INC (HL) ; increment counter – label CCCC
3F68 7E LD A,(HL) ; load A with counter
3F69 B8 CP B ; counter reached maximum?
3F6A C0 RET NZ ; no, return without extra play step
;
3F6B 3600 LD (HL),#00 ; reset counter
3F6D CDFE06 CALL #06FE ; do main playing game again
3F70 C9 RET ; return
```

Even testen door een waarde te zetten

```
3F47  3AB064      LD      A,(#64B0)      ; load A with counter setup
3F53  3AB064      LD      A,(#64B0)      ; load A with counter setup
```

Beiden tijdelijk vervangen door 3E XX 00 met XX waarde van counter setup.

Dus bijvoorbeeld 130% = #A5, 150% = #20 en 170% = #25.

Lijkt goed te gaan. Verschillende snelheden lijken inderdaad te kloppen met de juiste percentages.

Twee dingen nog uit te zoeken:

- Bij de elevators wordt jumpman te snel verplaatst: gaat sneller dan de lift.
- Het lijkt erop alsof de fireballs bij rivets te langzaam spawnen.

Kan het zijn dat dit zaken zijn die aan de framecounter gebonden zijn?

Eventjes getest, maar bij de pies gaat jumpman wel net zo snel als de pies.

#### Nu versie v0.01 opgeslagen.

Kijken in de code hoe de positie van jumpman aangepast wordt aan de elevator. Als de speed op 2x staat, dan heeft jumpman wel dezelfde snelheid als de elevator. Lijkt er ook op dat jumpman soms ook juist langzamer beweegt dan de elevator.

Ja daar lijkt wel te komen door de FrameCounter. Het verplaatsen van de lift is afhankelijk van de framecounter en wordt in 50% van de gevallen gedaan (even) en jumpman wordt verplaatst in de andere 50% van de gevallen (oneven). Als de FrameCounter oneven is bij een ingevoegde call, dan wordt de lift niet verplaatst, maar wordt wel jumpman verplaatst (2 keer). Als de FrameCounter even is bij een ingevoegde call, dan wordt de lift wel verplaatst (2 keer), maar wordt jumpman niet verplaatst. Dus die lopen door deze twee afhankelijkheden niet gelijk en dus afhankelijk van interval waarop de calls ingevoegd worden.

Even gaan bedenken wat je daar aan kunt doen. En eventjes verder kijken waar de FrameCounter nog meer gebruikt wordt. Stel bijvoorbeeld iets spawnen op basis van FrameCounter, dan klopt dat ook niet wellicht een keer niet spawnen of juist twee keer spawnen?

Wellicht een eigen FrameCounter maken, die bij elke call naar de play step opgehoogd wordt. Eigenlijk doen in het vrije slot in de main. Elke keer dat de main aangeroepen wordt, wordt deze counter verhoogd en die counter dan gebruiken i.p.v. de FrameCounter bij de elevators. Als dat het probleem oplost dan op meer plaatsen de FrameCounter aanpassen.

Even in de debugger gekeken en inderdaad wordt de FrameCounter (#601A) niet opgehoogd in opeenvolgende keren doorlopen van de main. Dus dat heeft zeker impact.

De #601B gebruiken als aangepaste FrameCounter en deze eventjes ophogen vanuit extra code die vanuit het extra slot van de main aangeroepen wordt.



Aanpassen:

19C2 CD713F CALL #3F71 ; call additional code

en:

3F71 211B60 LD HL,#601B ; load HL with FCAdjusted  
3F74 35 DEC HL ; decrement FCAdjusted  
3F75 C9 RET ; return

En dan de FrameCounter bij de elevator aanpassen:

2709 3A1B60 LD A,(FCAdjusted)

Ja dat lost het probleem bij de elevators op. Nog eventjes testen met andere snelheden.  
Als dan goed, dan overal tijdens het spelen de FrameCounter verwijzingen aanpassen naar de aangepaste FrameCounter. En dan uitgebreid testen.

Gaat ook goed met andere snelheden.

Aanpassen:

005A	211A60	LD	HL,FrameCounter	; Random number generator
041A	3A1A60	LD	A,(FrameCounter)	; Kong animation
220A	3A1A60	LD	A,(FrameCounter)	; Retractable ladders
2602	3A1A60	LD	A,(FrameCounter)	; Top conveyor reverse
261F	3A1A60	LD	A,(FrameCounter)	; Top conveyor reverse
263A	3A1A60	LD	A,(FrameCounter)	; Middle conveyor reverse
265A	3A1A60	LD	A,(FrameCounter)	; Middle conveyor reverse
2679	3A1A60	LD	A,(FrameCounter)	; Lower conveyor reverse
2696	3A1A60	LD	A,(FrameCounter)	; Lower conveyor reverse
26E9	3A1A60	LD	A,(FrameCounter)	; Pie tray speed
2709	3A1A60	LD	A,(FrameCounter)	; Elevator speed
2C27	3A1A60	LD	A,(FrameCounter)	; Crazy barrel deployment rate
2DF6	3A1A60	LD	A,(FrameCounter)	; River Firefox deployment rate

2E19	3A1A60	LD	A,(FrameCounter)	; Spring open/close rate
2FBE	3A1A60	LD	A,(FrameCounter)	; Hammer active blink rate
3110	3A 1A 60	LD	A,(FrameCounter)	; Fireball speed at difficulty 0 and 1
311B	3A1A60	LD	A,(FrameCounter)	; Fireball speed at difficulty 2
3126	3A1A60	LD	A,(FrameCounter)	; Fireball speed at difficulty 3 and 4
3131	3A1A60	LD	A,(FrameCounter)	; Fireball speed at difficulty 5
31FE	3A1A60	LD	A,(FrameCounter)	; Fireball freeze rate

Gaat op zich allemaal goed.

Alleen een keer bij Pies board dat Kong aan het einde blijft hangen. Dat komt waarschijnlijk omdat dat niet meer als spelen gezien wordt en de animatie daardoor niet afhankelijk gemaakt moet worden van de aangepaste Framecounter.

Terugzetten:

041A	3A1A60	LD	A,(FrameCounter)	; Kong animation
------	--------	----	------------------	------------------

Dat heeft nog niet geholpen.

Eerst deze aanpassingen even opslaan.

**Nu versie v0.02 opgeslagen.**

Waar wordt Kong teruggeschoven na het beëindigen van Pies? Dat gebeurt vanaf #16A3. Eventjes kijken met de debugger wat er precies gebeurt. Waarom blijft hij stil staan? Maar het vuur boven de oil can flakkerd nog wel en de 1UP knippert ook nog. Dus er loopt nog wel wat, maar blijft in een loop o.i.d. zonder verder te komen.

Hij komt dan in de top conveyor code, maar de FCAdjust staat dan stil. Dat gaat dus niet goed. De conveyor settings daarom terugzetten. Dat betekent dat de richting wellicht iets te snel of langzaam omdraait, maar wel gelijk houden.

Terugzetten:

2602	3A1A60	LD	A,(FrameCounter)	; Top conveyor reverse
261F	3A1A60	LD	A,(FrameCounter)	; Top conveyor reverse
263A	3A1A60	LD	A,(FrameCounter)	; Middle conveyor reverse
265A	3A1A60	LD	A,(FrameCounter)	; Middle conveyor reverse

2679 3A1A60 LD A,(FrameCounter) ; Lower conveyor reverse

2696 3A1A60 LD A,(FrameCounter) ; Lower conveyor reverse

Dat lost het probleem inderdaad op. Donkey Kong end board animatie op pies board gaat nu wel goed.

Alleen is er een ander probleem. En dat is dat Kong op de bovenste conveyor soms veel te vroeg van rechts naar links omdraait en soms veel te laat (deels al buiten het scherm).

Eerst eventjes opslaan.

Nu versie v0.03 opgeslagen.

Misschien is het een oplossing wanneer voor de alle conveyors gewoon de aangepaste FrameCounter gebruikt wordt: versneld zoals de bedoeling is. Maar dat er een conditie ingebouwd wordt dat wanneer de animatie loopt dan de FrameCounter gebruikt wordt. Op basis van welk geheugenadres wordt de animatie bepaalt? Lijkt gebaseerd te worden op #6388. Even met de debugger kijken wat er met deze waarde tijdens spelen pies gebeurt. Als deze #00 is tijdens spelen en oploopt tijdens de animatie, dan moet als deze #00 is de FCAdjusted gebruikt worden en anders de FrameCounter.

Debugging bevestigt het vermoeden dat dit wel eens een juiste oplossing zou kunnen zijn.

Dus deze vier op FCAdjust:

263A 3A1A60 LD A,(FrameCounter) ; Middle conveyor reverse

265A 3A1A60 LD A,(FrameCounter) ; Middle conveyor reverse

2679 3A1A60 LD A,(FrameCounter) ; Lower conveyor reverse

2696 3A1A60 LD A,(FrameCounter) ; Lower conveyor reverse

En deze twee aanpassen naar conditionele:

2602 CDF13F CALL #3FF1 ; call to additional code – call to label AAAA

en:

261F CDF13F CALL #3FF1 ; call to additional code – call to label AAAA

en:

3FF1 3A8863 LD A,#6388 ; load A with animation counter – label AAAA

3FF4 FE00 CP #00 ; A == 00 ?

3FF6 2804 JR Z,#BBBB ; yes, no animation, skip next steps – label BBBB

;

```

3FF8  3A1A60      LD      A,#601A      ; load A with FrameCounter
3FFB  C9          RET                      ; return
;
3FFC  3A1B60      LD      A,#601B      ; load A with FCAdjusted – label BBBB
3FFF  C9          RET                      ; return

```

Het leek geholpen te hebben. Bij het eerste pie board ging het goed. Maar bij het tweede pie board gaat het niet goed. Met de debugger onderzoeken. Kijken of weer gaat hangen en dan kijken welke animatie waarde en of de FC Adjusted nog loopt. Gaat hangen. De animatietimer staat op 1 en de FCAdjusted loopt niet: dat komt omdat de play main niet uitgevoerd wordt. Maar de code voor de extra frames die ingevoegd worden, die loopt nog wel.

Twee aanvliegroutes:

- Gebruik maken van de FCAdjusted en dan kijken of de FCAdjusted ook nog bijgehouden kan worden wanneer de aanroep naar de conveyor code gedaan wordt.
- Gebruik maken van de FrameCounter en dan kijken of de draaimomenten aangepast kunnen worden.

Eerst nog even testen. Gaat het met het gebruik van de FrameCounter wel goed qua animaties? Dat moeten we wel zeker weten. Daarom terug naar versie v0.03. Nee hoor, dan hangt hij ook. Dus dat komt niet daardoor. Dus dat maakt eigenlijk niets uit.

Maar toch komt hij in een soort van loop. Waarschijnlijk die steeds step 1 van de animatie doet. Ja dat klopt de 6388 staat op 1. Dus loopt steeds door code vanaf #16BB. Kijken met debugger of dat klopt. En dan stappen door de code en kijken of hij bij #16DF komt waar blijkbaar Kong removed wordt. En als hij daar niet komt: waarom wordt Kong niet removed.

Ja hij komt wel steeds in #16DF en hij doet ook de RST die de move van Kong doet. Maar Kong moved niet. RST is wel een aparte instructie. Rommelt met de stack en daardoor wellicht probleem met aanroepen van de extra main. Wat als we deze RST 38 vervangen door een CALL?

```

16D5  CD0226      CALL     #2602          ; ???
16D8  3AA363      LD       A, (#63A3)        ; load A with direction vector for top conveyor
16DB  4F           LD       C,A          ; C := 1
16DC  210869      LD       HL, #6908          ; load HL with start of Kong sprite
16DF  FF          RST       #38          ; move kong
16E0  C9          RET                      ; return

```

Aanpassen:

```

16DC  C3F93F      JP       #3FF9          ; jump to additional code
16DF  0000      NOP, NOP

```

en:

```

3FF9  210869      LD       HL, #6908          ; load HL with start of Kong sprite
3FFC  CD3800      CALL     0038          ; move Kong

```

3FF C9 RET ; return

Dit helpt niet. Blijft af en toe nog steeds hangen. Hoe kan dat?

Kijken met debugger. Positie van Kong zou uitgelezen moeten worden en daarna aangepast moeten worden, waarna de sprite geupdate wordt. Hoe werkt dat? En wat gebeurt er met de bijbehorende geheugenplaatsen tijdens de loop?

#6910 = Kongs X-position.

Bij de aanroep van de RST #38 wanneer het hangt: HL = #6908 (begin van de Kong sprites) en de C = #00.

De RST#38 doet voor een heel blok steeds C opstellen bij de HL, HL+4, HL+8 ... enz. Dus voor alle sprites de X-waarde verschuiven zodat de hele samengestelde sprite verplaatst. Maar als C #00 is, dan gebeurt er dus effectief niets. Dus verplaatst Kong wel, alleen de offset wordt niet goed berekend.

Eens kijken wat er in C staat als het wel goed gaat. Waarschijnlijk #01 of #FF afhankelijk van de richting waarop Kong moet bewegen. Ja dat klopt: bij bewegen naar links staat er #FF.

Dus nu gericht zoeken naar de reden waarop de C niet goed berekend wordt.

C wordt gebaseerd op de #63A3. Die in de routine op #2602 bepaald wordt. Daar wordt een call gedaan naar de code op #26E9 en daar staat nog een stukje code dat afhankelijk is van de FrameCounter en nu dus van de FCAdjusted. Daar kon het wel eens fout gaan.

Eerst eens proberen of de eerdere truuk met de afhankelijke keuze voor de FrameCounter of de FCAdjusted ook hier werkt en dat wellicht het probleem oplost.

Dus alles weer op FCAdjusted door terug te gaan naar versie v0.02. En de volgende drie calls aanpassen naar call naar extra stukje code:

En deze twee aanpassen naar conditionele:

2602	CDF13F	CALL	#3FF1	; call to additional code – call to label	AAAA
261F	CDF13F	CALL	#3FF1	; call to additional code – call to label	AAAA
26E9	CDF13F	CALL	#3FF1	; call to additional code – call to label	AAAA

en:

3FF1	3A8863	LD	A,#6388	; load A with animation counter – label	AAAA
3FF4	FE00	CP	#00	; A == 00 ?	
3FF6	2804	JR	Z,#BBBB	; yes, no animation, skip next steps – label	BBBB
;					
3FF8	3A1A60	LD	A,#601A	; load A with FrameCounter	
3FFB	C9	RET		; return	
;					

3FFC	3A1B60	LD	A,#601B	; load A with FCAdjusted – label <b>BBBB</b>
3FFF	C9	RET		; return

Ja dit gaat goed zo.

**Nu versie v0.04 opgeslagen.**

Dan nu kijken voor code om de instellingen afhankelijk te maken van het level:

L=01	110% = 10%	=	10 tellen	+	0 tellen
L=02	120% = 20%	=	5 tellen	+	0 tellen
L=03	130% = 30%	=	10 tellen	+	5 tellen
L=04	140% = 40%	=	5 tellen	+	5 tellen
L=05	150% = 50%	=	2 tellen	+	0 tellen
L=06	160% = 60%	=	2 tellen	+	10 tellen
L=07+	170% = 70%	=	2 tellen	+	5 tellen

Tabelletje maken in de code:

3FEA	A0 50 A5 55 20 2A 25		; table with speeds per level
------	----------------------	--	-------------------------------

Je moet dan het level lezen en dan door tabel heen stappen, de waarde lezen en deze zetten in de #6B40 counter setup. Nog wel eventjes ook de vaste waarden op #3F47 en #3F53 herstellen:

3F47	3AB064	LD	A,(#64B0)	; load A with counter setup
3F53	3AB064	LD	A,(#64B0)	; load A with counter setup

Het vaststellen van de counter setup gaan toevoegen aan het bestaande stukje code dat aangeroepen wordt vanuit het vrije slot van de main waarin nu nog alleen de FCAdjusted geconfigureerd wordt.

3F71	211B60	LD	HL,#601B	; load HL with FCAdjusted
3F74	35	DEC	HL	; decrement FCAdjusted
3F75	C9	RET		; return

Aanpassen:

3F75	21EA3F	LD	HL,#3FEA	; load HL with start of table
				;
3F78	3A2962	LD	A,(#6229)	; load A with level number
3F7B	FE08	CP	#08	; A >= 8 ?
3F7D	3802	JR	C,BB	; no, skip next step
3F7F	3E07	LD	A,#07	; force maximum of #07
				;
3F81	47	LD	B,A	; for B=1 to level number
				;
3F82	7E	LD	A,(HL)	; load value from table – label <b>AAAA</b>
3F83	23	INC	HL	; next table element

```

3F84  10FC      DJNZ  #BBBB      ; next B – jump to label AAAA
;
3F86  32B064    LD    (#64B0),A    ; store read value in counter setup
3F89  C9        RET                ; return

```

Ja dat werkt goed. Boven level 7 krijg je ook steeds de waarde van level L=07 (getest met zowel L=08 als L=09).

Nu versie v0.05 opgeslagen.

HIGH SCORE aanpassen naar ACCELERATE

#36B2: 60 76 11 13 13 15 1C 15 22 11 24 15 3F

### 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:

© 1981-2023 NINTENDO

5C 77 49 4A 10 01 09 08 01 2C 02 00 02 03 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:

ACCELERATE

8E 76 11 13 13 15 1C 15 22 11 24 15 3F

En de verwijzing ernaar toe aanpassen:

3689: 18 3F

Nu nog de Donkey Kong een regel naar beneden verplaatsen.

Wordt hier gedaan:

```
0820 210869 LD HL,#6908 ; load HL with start of kong sprite X pos
0823 0E44 LD C,#44 ; load C with offset to add X
0825 FF RST #38 ; draw kong in new position
0826 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

Ja dat is goed.

## Meteen beginnen met het titelscherm

Aanpassen:

```
01ED C3983F JP #3F98 ; jump to additional code – jump to label AAAA
01F0 00 NOP
```

en

```
3F98 3E06 LD A,#06 ; A := 6 – label AAAA
3F9A 320A60 LD (GameMode2),A ; store into game mode 2
3F9D C3F101 JP #01F1 ; jump back
```

## Versienummer toevoegen

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

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

081C: C3C83F JP #3FC8 ; jump to additional code to display version

3FC8: 110103 LD DE,#0301 ; load task data for text "V1.00"

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

3FCE: C31F08 JP #081F ; jump back

Ja dat is goed.

### De animatie waar Kong de trappen met Pauline beklimt overslaan

Dit stuk begint bij 0A76. Hele stuk niet doen, alleen de gamemode2 verhogen:

0A76 210A60 LD HL,GameMode2

0A79 34 INC (HL) ; increase game mode2 (to 8?)

0A7A C9 RET

En dan het hele stuk van 0A7B t/m 0BD9 vrijgeven: is allemaal code die de kong climbing regelt.

Ja dat gaat ook goed.

## Aanpassen tekst How High Can You Get?

De tekst HOW HIGH CAN YOU GET ? veranderen in HOW FAST CAN YOU GET ?  
Zelfde lengte dus aanpassen op originele plek #36CC:

5E 77 18 1F 27 10 **16 11 23 24** 10 13 11 1E 10 29 1F 25 10 17 15 24 10 FB 10 3F

Ja dat is ook goed.

De meter aanduidingen ook weghalen.

```
0C5C 7E LD A,(HL) ; get table data
0C5D DD7760 LD (IX+#60),A ; write to screen
0C60 23 INC HL ; next
0C61 7E LD A,(HL) ; get data
0C62 DD7740 LD (IX+#40),A ; write to screen
0C65 23 INC HL ; next
0C66 7E LD A,(HL) ; get table data
0C67 DD7720 LD (IX+#20),A ; write to screen
0C6A DD36E08B LD (IX-#20),#8B ; write "m" to screen
```

#0C5D, #0C62, #0C67 en #0C6A allemaal NOP's maken.

Is goed.

**Nu versie v0.06 opgeslagen.**

## Player 1 only maken

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

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

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

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

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

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

Ja dat werkt.

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

Dat wordt hier afgehandeld:

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

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

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

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

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

Dat klopt. Nu geen two-player mode meer.

Nu versie v0.07 opgeslagen.

Volgens mij nu alles aangepast. Dan nu het kleurenpalette aanpassen en alle schermen hetzelfde palette geven. Daarna Goofy Kong aanpassen en de bonus items aanpassen. Dan ook kijken naar de barrels en de elevators en andere zaken die aangepast dienen te worden om een grafisch aantrekkelijke romhack te krijgen.

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 10 00 00 10 10 2C 10 10 00 00 10 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 00 10 10 10 00 00 00 00 00 10 10 3F 00 00 61 00 F6 76
35A9: 98 77 20 19 15 23 10 10 10 10 10 10 00 00 00 00 00 10 10 10 00 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 00 10 10 10 00 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
```

Ja is zo goed.

Nu opnemen van de kleurendefinitie (c-2j.bpr en c-2k.bpr) en kleurenmapping (v-5e.bpr) van de andere romhack.

Mmm. De girders zijn nu wel goed. Maar de kleuren van jumpman, Pauline en de items zijn ook aangepast. Herstellen is wel weer veel werk. Wellicht handiger om de kleuren van de girders en ladders alleen aan te passen.

Dus de originele versies van deze drie bestanden terugzetten.

Alleen de kleuren van het giders veld aanpassen:

	c-2j.bpr								c-2k.bpr							
Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	0F	06	0E	01	0F	00	01	00	0F	00	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	06	0D	0D
00000090	0F	06	0D	0D	0F	06	0D	0D	0F	06	0D	0D	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

	c-2j.bpr								c-2k.bpr							
Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	0F	0C	00	0F	0F	0B	0F	00	0F	0A	0F	0D	0F	0C	0F	03
00000010	0F	03	0B	0F	0F	00	0D	03	0F	0F	0C	00	0F	0F	0A	00
00000020	0F	0F	0A	03	0F	00	03	05	0F	0D	00	05	0F	0C	03	0A
00000030	0F	00	00	0C	0F	0F	0B	03	0F	0E	0A	0C	0F	0B	0F	0F
00000040	0F	0C	00	0F	0F	0B	0F	00	0F	0A	0F	0D	0F	03	0A	00
00000050	0F	03	0A	00	0F	03	0A	00	0F	03	0A	00	0F	0F	0A	00
00000060	0F	0F	0A	03	0F	00	03	05	0F	0D	00	05	0F	0C	03	0A
00000070	0F	00	00	0C	0F	0F	0B	03	0F	0E	0A	0C	0F	0B	0F	0F
00000080	0F	0C	00	0F	0F	0B	0F	00	0F	0A	0F	0D	0F	0A	0E	02
00000090	0F	0A	0E	02	0F	0A	0E	02	0F	0A	0E	02	0F	0F	0A	00
000000A0	0F	0F	0A	03	0F	00	03	05	0F	0D	00	05	0F	0C	03	0A
000000B0	0F	00	00	0C	0F	0F	0B	03	0F	0E	0A	0C	0F	0B	0F	0F
000000C0	0F	0C	00	0F	0F	0B	0F	00	0F	0A	0F	0D	0F	0C	00	0B
000000D0	0F	0C	00	0B	0F	0C	00	0B	0F	0C	00	0B	0F	0F	0A	00
000000E0	0F	0F	0A	03	0F	00	03	05	0F	0D	00	05	0F	0C	03	0A
000000F0	0F	00	00	0C	0F	0F	0B	03	0F	0E	0A	0C	0F	0B	0F	0F

Ja dat gaat goed. Om niet het High score screen te beïnvloeden de andere boards ook van dit palette gebruik laten maken.

Palette van elevators is al gelijk aan die van barrels dus niet aanpassen.

Palette van rivets wordt hier gezet:

```

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

```

Aanpassen:

```
0CBC 0000 NOP, NOP
```

Palette van pies wordt hier gezet:

```

0CDF 115D3B LD DE,#3B5D ; load DE with start of table data for conveyors
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
0CEA 3E09 LD A,#09 ; load A with conveyor music
0CEC 328960 LD (#6089),A ; set music for conveyors
0CEF C3C60C JP #0CC6 ; jump back

```

Aanpassen:

```

0CE5 0000 NOP, NOP
0CE8 0000 NOP, NOP

```

Controleren of dit werk en of er sprites zijn, die hierdoor de verkeerde kleur krijgen.

**Nu versie v0.08 opgeslagen.**

Dat gaat allemaal goed. Wel nog het title screen en het high score screen aanpassen.

Ziet er nu goed uit.

Nu nog de volgende zaken doen: **kleuren van het high score screen**, structuur van de lijnen op rivets, kleur van de lijnen op rivets en elevators, de elevator platforms, kleur van de barrels en de pies, **hartje rood maken, kleur van het selectie blokje name registration.**

### Kleur van het hartje

Het hartje rood maken. Is nu kleur #09. Moet worden kleur #00.

Zit op twee plaatsen:

```

1708 CD1C01 CALL #011C ; clear all sounds
170B 21206A LD HL,#6A20 ; load HL with heart sprite
170E 3680 LD (HL),#80 ; set heart sprite X position
1710 23 INC HL ; next
1711 3676 LD (HL),#76 ; set heart sprite
1713 23 INC HL ; next
1714 3609 LD (HL),#09 ; set heart sprite color
1716 23 INC HL ; next
1717 3620 LD (HL),#20 ; set heart sprite Y position

```

```

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

```

### Selectieblokje name registration

High Score selectie blokje is niet een mooie kleur. Liefste geel met rood maken. Wordt de kleur ervan hier gezet?

```

160C 23 INC HL
160D 360C LD (HL),#0C

```

Zou kunnen. Eventjes testen met #0F i.p.v. #0C. Ja dat gaat goed.

### Kleuren van de scorelijnen in High Score list

Aangepast door de kleuren van de High Score blokjes aan te passen.

### Kleuren van de barrels aanpassen

De barrels aanpassen. De kleuren van de barrels aanpassen naar #0F.

```

; data used for the barrel pile next to kong
; called from #0FD7

```

```

3DDC 1E 18 0B 4B ; first barrel
3DE0 14 18 0B 4B ; second barrel
3DE4 1E 18 0B 3B ; third barrel
3DE8 14 18 0B 3B ; fourth barrel

```

In deze datastructuur de kleuren aanpassen naar #0F.

Nu de rollende barrels ook aanpassen.

Maar dan moet nu ook voor nieuw gedeployede barrels de kleur op #0F gezet worden. Dat is hier:

```

2CF6 DD360715 LD      (IX+#07),#15    ; set barrel sprite value to #15
2CFA DD36080B LD      (IX+#08),#0B    ; set barrel color to #0B
2CFE DD361500 LD      (IX+#15),#00    ; set +15 indicator to 0 = normal barrel, [1 = blue barrel]
2D02 3A8263 LD      A, (#6382)        ; load A with Crazy/Blue barrel indicator
2D05 07          RLCA                  ; is this a blue barrel ?
2D06 D2152D JP      NC, #2D15          ; No blue barrel, then skip next 3 steps

; blue barrel

2D09 DD360719 LD      (IX+#07),#19    ; set sprite for blue barrel
2D0D DD36080C LD      (IX+#08),#0C    ; set sprite color to blue
2D11 DD361501 LD      (IX+#15),#01    ; set blue barrel indicator

```

Aanpassen kleur op regel #2CFA naar #0F. Ja dat gaat goed.

Het palette is nu wel goed, maar de kleuren moeten nog omgedraaid worden door de barrels sprites aan te passen: kopie maken van de barrels uit de andere romhack.

### Kleuren van de pies aanpassen

Kleur van de pies naar groen aanpassen.

Welke kleur wordt daarvoor gebruikt nu? Dat is de #0E. En binnen het palette is dat de kleur #7A. Die aanpassen naar een groene kleur (c-2j.bpr/c-2k.bpr = #0E / #0F).

c-2j.bpr:

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

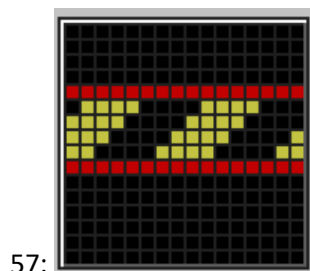
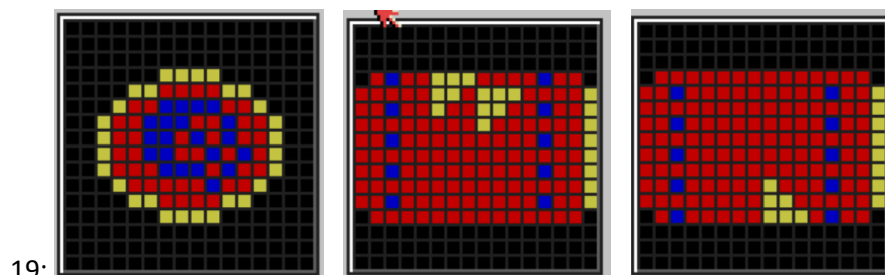
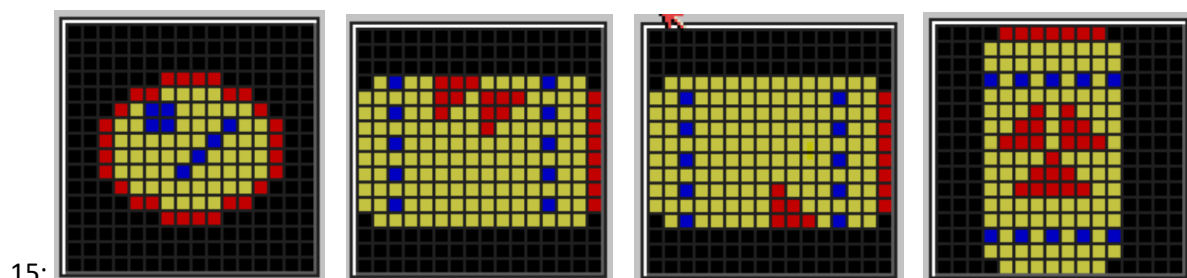
c-2k.bpr:

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

Nu versie v.009 opgeslagen.

De sprites van de barrels aanpassen. Vormgeving en de kleuren omdraaien.

Voorbeelden:





Nu kijken naar de blue barrels. Eigenlijk dezelfde sprites nemen, maar de kleur omdraaien. Rode barrel, met gele rand. Sprites aangepast. Aanpassen kleur op regel #2D0D naar #0F. Ja dat gaat goed en is ook erg mooi.

Nu versie v0.10 opgeslagen.

De sprite van de elevators is ook aangepast en moet de spritekleur krijgen van de barrels (#0F).

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

```
3E60: 44 03 08 04
```

De #03 is de kleur van de elevators.

Ja dat gaat nu ook goed.

De spites naast Kong op rivets aangepast. Maar die worden ook gebruikt voor de kabels van de elevators. Die moeten origineel blijven. Hiervoor nieuwe gemaakt i.p.v. #FC / #FD is dat nu #DB / #DC geworden.

Dat zit hier in de code:

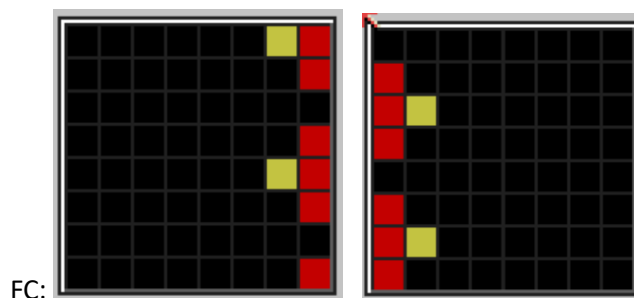
```
0D30 0611    LD      B,#11          ; for B = 1 to #11

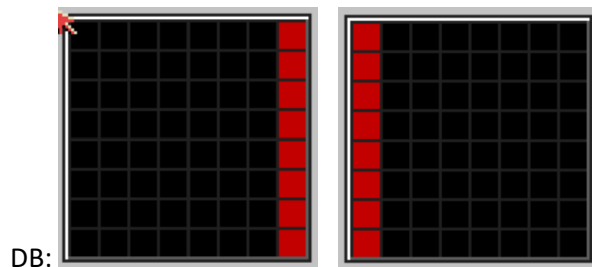
0D32 36FD    LD      (HL),#FD        ; draw the cable to screen
0D34 23      INC     HL             ; next location
0D35 10FB    DJNZ    #0D32         ; Next B

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

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

Sprites lijnen:





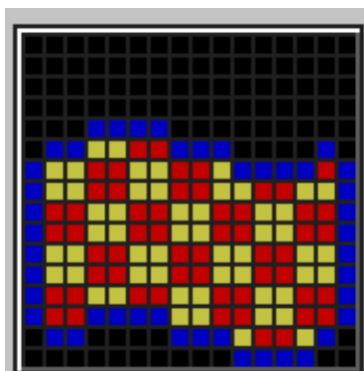
De kleuren zijn allemaal aangepast.

Nu versie v0.11 opgeslagen.

Nu bonus item sprites aanpassen. Wellicht mooi als deze bijvoorbeeld net als de ballonnen in anniversary edition gaan bewegen. Denk aan een zwart witte finish vlag die wappert via drie verschillende versies. En dan alle bonus items hetzelfde maken en in de routine in het extra slot de sprites van waarde laten veranderen op basis van een randomizer. Dus wapperen is randomized en zal ook sneller gaan bij hogere snelheden.

Leuk idee. Eerst een enkele sprite maken.

De bonus items sprites vervangen door drie finishvlaggen. #73, #74 en #75.



Blauw = roze, Rood = wit en Geel = Blauw.



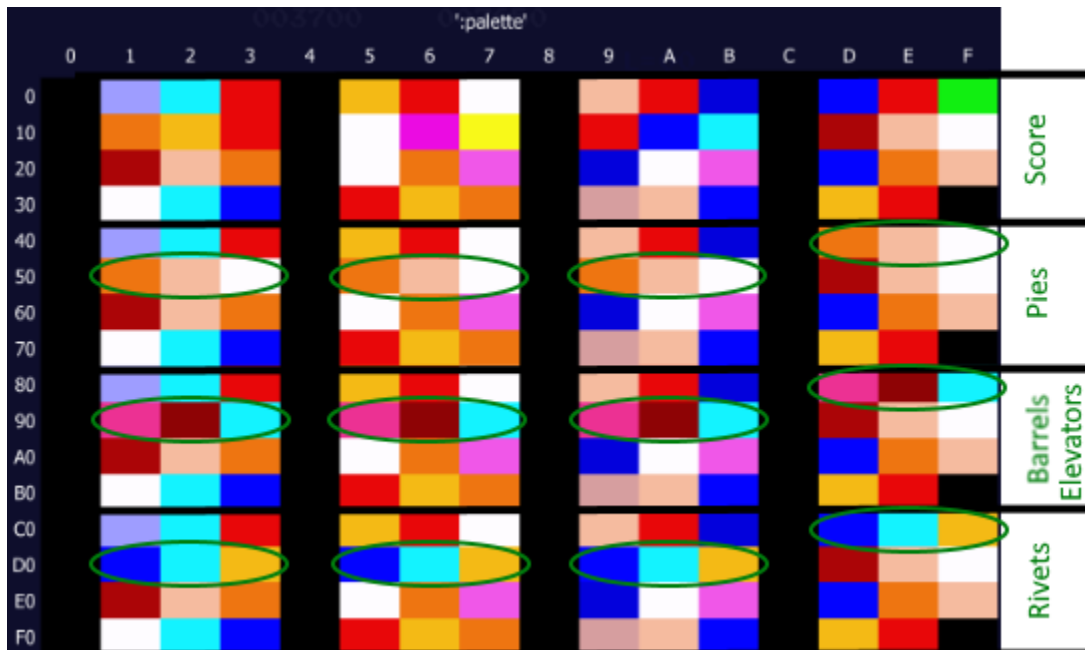
De eerste waarde = geel = moet zwart worden = #0F/#0F

De tweede waarde = rood = moet wit worden = #00/#00.

De derde waarde = blauw = moet blauw worden = #0F/#09.

Dus 0F 00 0F in c-2j.bpr en 0F 00 09 in c-2k.bpr.

De kleuren van Pauline echter niet aanpassen. Dus welke zijn er nog vrij om te gebruiken?



De kleuren 00 wordt voor de bovenste regel van het scherm gebruikt. Als we die aanpassen dan wordt de kleur van de titel blauw. Nee is niet handig. Dan wordt namelijk het hartje ook blauw.

Even kijken waar kleuren voor gebruikt worden:

00 = Heart, Springs	01 = Fireballs	02 = Jumpman,	03 = Background
04 = Background,	05 = Back	06 = Background,	07 = DK, Hammer
08 = DK,	09 = Pauline,	0A = Pauline,	0B = ?,
0C = Oil Can,	0D = ?,	0E = Pies,	0F = barrels, elevator

Lijkt erop dat 0B en 0D vrij zijn.

De 0B gebruiken.

En dan de bonus items naar de juiste kleur laten verwijzen.

```

; bonus items on conveyors

3E3C  53 73 0A A0      ; position of hat on pies is 53,A0
3E40  8B 74 0A F0      ; position of purse on pies is 8B,F0
3E44  DB 75 0A A0      ; umbrella on the pies is at DB,A0

; bonus items for elevators

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

; bonus items for rivets

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

```

De 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 v0.12 opgeslagen.**

Nu de animatie van de vlaggen maken. Eventjes kijken bij anniversary edition hoe dat gedaan is.  
De vlaggen verplaatsen naar:

```

#54:  0101 0100
#55:  0101 0101
#56:  0101 0110
#57:  0101 0111

```

Initialiseren met verschillende waarden.

Dan een stuk code tussenvoegen met een teller: optellen tot waarde (snelheid wapperen) en dan spritewaardes aanpassen. Lees de waarde van de sprite, increment, als 58 dan reset naar 54 en terugzetten in de spritewaarde. En dat voor de drie sprites.

De drie sprites beginnen op: #6A0C. Dus #6A0D, #6A11, #6A15.

Toevoegen aan de code voor die in het extra slot uitgevoerd wordt:

```

3F75  21EA3F      LD      HL,#3FEA      ; load HL with start of table
;
3F78  3A2962      LD      A,(#6229)    ; load A with level number
3F7B  FE08        CP      #08          ; A >= 8 ?

```

3F7D	3802	JR	C,BB	; no, skip next step
3F7F	3E07	LD	A,#07	; force maximum of #07
;				
3F81	47	LD	B,A	; for B=1 to level number
;				
3F82	7E	LD	A,(HL)	; load value from table – label AAAA
3F83	23	INC	HL	; next table element
3F84	10FC	DJNZ	#BBBB	; next B – jump to label AAAA
;				
3F86	32B064	LD	(#64B0),A	; store read value in counter setup
<del>3F89</del>	<del>C9</del>	<del>RET</del>		<del>; return</del>
3F89	CD800A	CALL	#0A80	; call to animate the flags – call to label AAAA
3F8C	CDBBBB	CALL	#BBBB	; call to display speed meter – call to label BBBB
;				
3F8F	C9	RET		; return

Counter gebruiken voor animatie: #64B3.

0A80	3AB364	LD	A,(#64B3)	; load A with flag counter – label AAAA
0A83	3C	INC	A	; increment flag counter
0A84	32B364	LD	(#64B3),A	; store flag counter
;				
0A87	FE40	CP	#40	; A == #40?
0A89	C0	RET	NZ	; no, return
;				
0A8A	3E00	LD	A,#00	; load A with #00
0A8C	32B364	LD	(#64B3),A	; reset flag counter
;				
0A8F	210D6A	LD	HL,#6A0D	; load first flag sprite number
0A92	CDA20A	CALL	#0AA2	; adjust flag sprite number first flag – call CCCC
0A95	21116A	LD	HL,#6A11	; load second flag sprite number
0A98	CDA20A	CALL	#0AA2	; adjust flag sprite number second flag – call CCCC
0A9B	21156A	LD	HL,#6A15	; load third flag sprite number
0A9E	CDA20A	CALL	#0AA2	; adjust flag sprite number third flag – call CCCC
;				
0AA1	C9	RET		; return
;				
0AA2	7E	LD	A,(HL)	; load A with sprite number – label CCCC
0AA3	3C	INC	A	; increment sprite number
;				
0AA4	FE58	CP	#58	; A == 58?
0AA6	2002	JR	NZ,#DDDD	; no skip next step
;				
0AA8	3E54	LD	A,#54	; reset sprite number
;				
0AAA	77	LD	(HL),A	; store sprite number
0AAB	C9	RET		; return

Ja dat gaat goed.

#### Versie v0.13 opgeslagen.

Aangepaste vlag sprites waardoor het beter lijkt.

#### Versie v0.14 opgeslagen.

Nu nog de tekst op how fast scherm grijs maken en de goofy kong van een finish vlag voorzien.  
En de progress bar met de snelheidsindicatie

De drie verschillende progress bar karakters #BA (geel), #BB (rood) en #BC (blauw).

Meest rechtse karakter = positie  $\#7440 + \#02 + \#20 = \#7462$ .  
Karakters dan dus #7522, #7502, #74E2, #74C2, #74A2, #7482, #7462.

Stukje code maken voor simpel tekenen van de karakters:

```
3F8C  CDB00A      CALL  #0AB0      ; call to display speed meter – call to label BBBB
```

en:

```
0AB0  11E0FF      LD      DE,#FFE0      ; load DE with character offset
0AB3  212275      LD      HL,#7522      ; load HL with first position
0AB6  0607        LD      B,#07        ; load B with number of characters
0AB8  3EBA        LD      A,#BA        ; load A with character number
;
0ABA  77          LD      (HL),A      ; draw character
0ABB  19          ADD     HL,DE        ; add offset
0ABC  10FC        DJNZ   XXXX        ; next character
;
0ABE  C9          RET                ; return
```

Gaat op zich goed. Moet wel een regel naar boven. En wordt hier nog deels gewist. In ieder geval de randen tekenen bij de beginopbouw van het veld. Zodat het al meteen zichtbaar is. Zonder gevulde snelheid.

Toch nog een regel omhoog en een positie naar rechts:

```
0AB3  210175      LD      HL,#7501      ; load HL with first position
```

En dan nu zo maken dat het aantal balkjes afhankelijk is van het level en dat de kleur van de balkjes ook afhankelijk is van het level.

Aanpassen:

0AB0	112000	LD	DE,#0020	; load DE with character offset
0AB3	216174	LD	HL,#7461	; load HL with first position
0AB6	0607	LD	B,#07	; load B with number of characters
;				
0AB8	CDBE0A	CALL	0ABE	; call routine to process character – label AAAA
0ABB	10FB	DJNZ	#0AB8	; next character
;				
0ABD	C9	RET		; return
;				
0ABE	3A2962	LD	A,(#6229)	; load A with level number – label AAAA
0AC1	B8	CP	B	; level nr <= B?
0AC2	3814	JR	C,#EEEE	; yes, return – don't draw character – label EEEE
;				
0AC4	78	LD	A,B	; load A with step number
0AC5	FE03	CP	03	; A >=3 ?
0AC7	3004	JR	NC, #BBBB	; yes, skip next steps – jump to label BBBB
;				
0AC9	3EBA	LD	A,#BA	; A = 1 or 2 – load green character
0ACB	180A	JR	#DDDD	; skip ahead – jump to label DDDD
;				
0ACD	FE06	CP	06	; A >=6 ? – label BBBB
0ACF	3004	JR	NC, #CCCC	; yes, skip next steps – jump to label CCCC
;				
0AD1	3EBB	LD	A,#BB	; A = 3, 4 or 5 – load orange character
0AD3	1802	JR	#DDDD	; skip ahead – jump to label DDDD
;				
0AD5	3EBC	LD	A,#BC	; A = 6 or 7 – load red character – label CCCC
;				
0AD7	77	LD	(HL),A	; draw character – label DDDD
0AD8	19	ADD	HL,DE	; add offset – label EEEE
0AD9	C9	RET		; return

Het tekenen van de speed bar gaat nu goed.

Nu versie v0.15 opgeslagen.

Nu testen met #0D kleurslot (#B5, #B6, #B7) . Wordt dat inderdaad nergens anders voor gebruikt?

Kleuren aanpassen naar groen – oranje – rood. In die volgorde in de definitie.

Groen: #0E / #0B      Oranje: #00 / #0F      Rood: #01 / #0F.

Aanpassen in kleurtabel: c-2j.bpr : 0E 00 01 en c-2k.bpr : 0B 0F 0F.

Goede kleuren. En worden inderdaad nergens gebruikt.

Dus nu in de kleurvelddefinitie het stuk van de progress bar aanpassen naar #0D in de v-5e.bpr.

Gaat goed. Alleen bij How Fast scherm wordt nog de oude kleur gebruikt deze ook aanpassen.

### Nu versie v0.16 opgeslagen.

Nu nog SPEED boven de bar en een ruler eronder. En dan nog mogelijk om de attractmode te versnellen? En dan nog het How Fast scherm aanpassen.

SPEED neerzetten ipv 2UP? Eigenlijk op zelfde moment weergeven als HIGH SCORE.

```
01F4 110403 LD DE,#0304 ; load task data to draw "HIGH SCORE"
01F7 CD9F30 CALL #309F ; insert task to draw text
01FA 110202 LD DE,#0202 ; load task #2, parameter 2 to display the high score
01FD CD9F30 CALL #309F ; insert task|
```

Hier tussenvoegen.

Task 6 is nog vrij.

Aanpassen:

```
01F4 CD400B CALL #0B40 ; call additional code
01F7 000000 NOP, NOP, NOP
```

en:

```
0B40 110403 LD DE,#0304 ; load task data to draw "HIGH SCORE"
0B43 CD9F30 CALL #309F ; insert task to draw text
;
0B46 110603 LD DE,#0306 ; load task data to draw "SPEED"
0B49 CD9F30 CALL #309F ; insert task to draw text
;
0B4C C9 RET ; return
```

en:

```
3657 D0 0B 6 ; task to draw SPEED
```

en:

```
0BD0 20 75 23 20 15 15 14 3F
```



Wordt wel neergezet, maar ook weer weggehaald.

Met debugger kijken waar dat gedaan wordt. Blijkbaar op #085B. Daar wordt het hele scherm gewist. Inclusief de HIGH SCORE (is nu ACCELERATE).

Op #0A37 zit nog een call naar task 3 om de HIGH SCORE te plaatsen. Die ook vervangen:

```
0A37 110403 LD DE,#0304 ; load task data for text #4 "HIGH SCORE"
0A3A CD9F30 CALL #309F ; insert task to draw text
0A3D 110202 LD DE,#0202 ; load task #2, parameter 2 to display high score
0A40 CD9F30 CALL #309F ; insert task
```

Aanpassen:

```
0A37 CD400B CALL #0B40 ; call additional code
0A3A 000000 NOP, NOP, NOP
```

Ziet er goed uit.

Starten met 6 jumpmans ongeacht de dip switch instelling.

Wordt hier gezet:

```
0207 3A807D LD A,(DSW1) ; load A with Dip switch settings
020A 4F LD C,A ; copy to C
020B 212060 LD HL,StartingLives ; set destination address to initial number of lives
020E E603 AND #03 ; mask bits, now between 0 and 3 inclusive
0210 C603 ADD A,#03 ; Add 3, now between 3 and 6 inclusive
0212 77 LD (HL),A ; store in initial number of lives
0213 23 INC HL ; next HL, now at ExtraLifeThreshold = score needed for extra life
0214 79 LD A,C ; load A with original value of dip switches
```

Aanpassen:

```
020E 3E06 LD A,#06 ; set A to number of initial lives of 6
0210 0000 NOP, NOP
```

Ja dat gaat goed.

Nu versie v0.17 opgeslagen.

Kijken of de attract mode sneller kan. Kun je de level instellen voor de attract mode? Dan L=05 instellen.

Dat wordt hier gedaan:

```

0764 AF      XOR      A          ; A := 0
0765 329263 LD      (#6392),A      ; clear barrel deployment indicator
0768 32A063 LD      (#63A0),A      ; clear fireball release indicator
076B 3E01 LD      A,#01          ; A := 1
076D 322762 LD      (#6227),A      ; load screen number with 1
0770 322962 LD      (#6229),A      ; load level # with 1
0773 322862 LD      (#6228),A      ; load number of lives with 1
0776 C3920C JP       #0C92        ; skip ahead

```

Aanpassen:

```

0770 322862 LD      (#6228),A      ; load number of lives with 1
0773 CD500B CALL     #0B50          ; call additional code

```

en:

```

0B50 3E07 LD      A,#07          ; load level # with 7
0B52 322962 LD      (#6229),A      ; load level # with 7
0B55 C9 RET                      ; return

```

Nee dat gaat niet sneller. Dat komt waarschijnlijk omdat de gamemode nu anders is dan in een normale game. Ja tijdens de attract mode is de Gamemode1 1 en is de Gamemode2 3.

Dat wordt hier gecheckt:

```

3F30 3A0560 LD      A,(GameMode1) ; load A with game mode1
3F33 FE03 CP       #03            ; Gamemode1 == 3 (playing a game)?
3F35 200A JR       NZ,BBBB        ; no, skip next steps – jump to label BBBB
;
3F37 3A0A60 LD      A,(GameMode2) ; load A with game mode2
3F3A FE0C CP       #0C            ; Gamemode2 == C (alive playing a game)?
3F3C 2003 JR       NZ,BBBB        ; no, skip next steps – jump to label BBBB
;
3F3E CDFE06 CALL     #06FE          ; do main playing game again
;
3F41 21D200 LD      HL,#00D2       ; load HL with return address – label BBBB
3F44 C3C500 JP       #00C5          ; jump back

```

Dit is dubbelop er zou volstaan kunnen worden met alleen de check op GameMode2 = #0C. Dus dan zou je de check op GameMode1 kunnen vervangen door een check op GameMode2 = #03.

Aanpassen:

```

3F30 3A0A60 LD      A,(GameMode2) ; load A with game mode1
3F33 FE03 CP       #03            ; Gamemode2 == 3 (playing a game)?
3F35 2807 JR       Z,CCCC         ; yes, skip next steps – jump to label CCCC
;
3F37 3A0A60 LD      A,(GameMode2) ; load A with game mode2

```

3F3A	FE0C	CP	#0C	; Gamemode2 == C (alive playing a game)?
3F3C	2003	JR	NZ,BBBB	; no, skip next steps – jump to label BBBB
;				
3F3E	CDFE06	CALL	#06FE	; do main playing game again – label CCCC
;				
3F41	21D200	LD	HL,#00D2	; load HL with return address – label BBBB
3F44	C3C500	JP	#00C5	; jump back

Nee. Dit gaat niet goed.

Terug naar versie v0.17.

Nog kijken: in het titelscherm knippert de Donkey Kong tekst niet meer. Als er al gespeeld is en er staat een speed op de meter dan knippert die wel. Dus kijken met welke het dan knippert en die aanpassen qua kleuren voor de meter en eventueel omgekeerde kleuren voor de Donkey Kong tekst?

Eerst kijken om de kleurcode voor de speed meter in alle palettes op te nemen.

Dat is kleurslot #0D met waarden:

Aanpassen in kleurtabel: c-2j.bpr : 0E 00 01 en c-2k.bpr : 0B 0F 0F.

Het titel scherm standaard is de kleuren van rivets en dan flitsend naar de kleuren van barrels. Dus 4<sup>de</sup> palette, flitsend naar 3<sup>e</sup> palette.

Ja dat gaat goed.

Nu versie v0.18 opgeslagen.

Dan nu kijken voor het How Fast scherm. De tekst is erg wit. Maar de vraag is of dat aangepast kan worden. Dan wordt waarschijnlijk ook de Goofy Kong aangepast. Dat klopt dat kan niet anders.

Dus maar zo laten dan.

Wel de Goofy Kong een finish vlag geven. Design maken met Paint.Net.

Ziet er helemaal goed uit.

Eventjes de invincibility uitzetten.

Nu versie v0.19 opgeslagen.

Is finale versie v1.00

Na testen door Justin is het idee nu om te gaan werken met vier verschillende difficulty keuzes: normal, hard, expert en insane. Met de volgende topsnelheden: normal 150%, hard 160% en expert 170%. En dan voor insane kijken.

Eerst op basis van een memorywaarde de mode bepalen en dan voor elke mode een andere datatabel met snelheden en de bijbehorende HL pointer vaststellen.

De tabel die de snelheid configureert is nu:

3FEA A0 50 A5 55 20 2A 25 ; table with speeds per level

Met de volgende code:

```

3F75 21EA3F LD HL,#3FEA ; load HL with start of table
;
3F78 3A2962 LD A,(#6229) ; load A with level number
3F7B FE08 CP #08 ; A >= 8 ?
3F7D 3802 JR C,BB ; no, skip next step
3F7F 3E07 LD A,#07 ; force maximum of #07
;
3F81 47 LD B,A ; for B=1 to level number
;
3F82 7E LD A,(HL) ; load value from table – label AAAA
3F83 23 INC HL ; next table element
3F84 10FC DJNZ #BBBB ; next B – jump to label AAAA
;
3F86 32B064 LD (#64B0),A ; store read value in counter setup
3F89 C9 RET ; return

```

Eerst eventjes de waarden bepalen voor de vier verschillende profielen:

Normal: 150% max speed bereiken in 7 stappen: 7,14% per stap.

Ideaal:	107	114	121	129	136	143	150
3F90:	E0	EE	B8	B5	94	A3	20

Hard: 160% max speed bereiken in 7 stappen: 8,57% per stap.

Ideaal:	109	117	126	134	143	151	160
3FD1:	B0	EA	B6	B4	A3	20	A2

Expert: 170% max speed bereiken in 7 stappen: 10% per stap.

Ideaal:	110	120	130	140	150	160	170
3FEA:	A0	50	A5	55	20	2A	25

Insane: 200% max speed bereiken in 7 stappen: 14,28 per stapper stap.

Ideaal:	114	129	143	157	171	186	200
3FD8:	70	B5	A3	E2	52	32	10

F: Tot 15 tellen = 6% versnelling  
 E: Tot 14 tellen = 7% versnelling  
 D: Tot 13 tellen = 8% versnelling  
 C: Tot 12 tellen = 8% versnelling  
 B: Tot 11 tellen = 9% versnelling  
 A: Tot 10 tellen = 10% versnelling  
 9: Tot 9 tellen = 11% versnelling  
 8: Tot 8 tellen = 12% versnelling  
 7: Tot 7 tellen = 14% versnelling  
 6: Tot 6 tellen = 17% versnelling  
 5: Tot 5 tellen = 20% versnelling  
 4: Tot 4 tellen = 25% versnelling  
 3: Tot 3 tellen = 33% versnelling  
 2: Tot 2 tellen = 50% versnelling  
 1: Tot 1 tellen = 100% versnelling

De #64B4 gebruiken voor vastleggen van de difficulty mode:

00 = Normal    01 = Hard    02 = Expert    03 = Insane

En dan aanpassen:

3F75    CDD80A    JP    #0ADB    ; call to additional code – call to label **AAAA**

en:

```

0ADB  3AB464      LD      A,(#64B4)    ; load difficulty mode – label AAAA
;
0ADE  FE00        CP      #00          ; difficulty mode = normal?
0AE0  2004        JR      NZ, #BBBB    ; no, skip next steps
0AE2  21903F      LD      HL,#3F90     ; load with start data able normal mode
0AE5  C9          RET                  ; return
;
0AE6  FE01        CP      #01          ; difficulty mode = hard?
0AE8  2004        JR      NZ, #BBBB    ; no, skip next steps
0AEA  21D13F      LD      HL,#3FD1     ; load with start data table hard mode
0AED  C9          RET                  ; return
;
0AEE  FE02        CP      #02          ; difficulty mode = expert?
0AF0  2004        JR      NZ, #BBBB    ; no, skip next steps
0AF2  21EA3F      LD      HL,#3FEA     ; load with start data table expert mode
0AF5  C9          RET                  ; return
;
0AF6  21D83F      LD      HL,#3FD8     ; load with start data table insane mode
0AF9  C9          RET                  ; return

```

Te testen door in 0ADB een vaste waarde te configureren: 3E01 LD A,#01. En dan met debugger kijken of de waarden in de juiste volgorde gekozen worden. Ja dat gaat goed.

## Nu versie v1.00-01 opgeslagen.

Nu bij opstarten van het spel een scherm tussenvoegen waarop de difficulty mode gekozen kan worden. We hebben er juist een scherm tussenuit gehaald: het scherm met de tekst voor de twee spelers.

Maar wat ook zou kunnen is het scherm van Kong climbing te gebruiken. Of in ieder geval die GameMode. Die wordt nu overgeslagen. Ook eventjes kijken hoe we dat gedaan hebben bij andere romhacks met een keuzescherf (DK Duel).

Gamemode voor de Kong climbing sequence is als volgt overgeslagen:

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

Nu eigenlijk zo maken dat hier naar additionele code gesprongen wordt (zolang de Gamemode op 7 staat) waarin het scherm getoond wordt, met de vier keuzeopties en een pijltje om door de opties heen te lopen. Je kan naar boven en naar beneden bewegen en wanneer de jumpbutton gedrukt wordt, wordt de laatst gekozen optie in de difficulty mode weggeschreven en wordt de Gamemode2 verhoogt.

Teksten op het scherm:

SELECT DIFFICULTY

NORMAL

HARD

EXPERT

INSANE

PRESS JUMP TO START GAME

Hebben we daar ruimte voor? Welke teksten zijn nog vrij? De #08 (op #365B), de #0B (op #3661), de #1D (op #3685). Dan nog 3 anderen. In ieder geval worden de volgende niet meer gebruikt en kunnen hergebruikt worden: #02 (op #364F was PLAYER <I>), #03 (op #3651 was PLAYER <II>) en #0A (op #365F was 1 OR 2 PLAYERS BUTTON).

Dus configureren:

0A – #365F -> #36FD ->

SELECT DIFFICULTY = 0C 77 23 15 1C 15 13 24 10 14 19 16 16 19 13 25 1C 24 29 3F

02 - #364F -> #3698 ->

NORMAL = 4F 76 1E 1F 22 1D 11 1C 3F

03 - #3651 -> #36A5 ->

HARD = 51 76 18 11 22 14 3F

08 - #365B -> #3FDF ->

EXPERT = 53 76 15 28 20 15 22 24 3F

0B - #3661 -> #3F25 ->

INSANE = 55 76 19 1E 23 11 1E 15 3F

1D - #3685 -> #0AFA ->

PRESS JUMP TO START = 37 77 20 22 15 23 23 10 1A 25 1D 20 10 24 1F 10 23 24 11 22 24 10  
3F

Volgens mij is het scherm dan al gewist en kan volstaan worden met het plaatsen van de teksten. Dat eerst eens proberen:

```
0A76  CD170B      CALL  #0B17      ; call to additional code – call to label AAAA
0A79  00          NOP
0A7A  C9          RET      ; return
```

en:

```
0B11  0A 02 03 08 0B 1D      ; table with text entries
;
0B17  21110B      LD      HL,#0B11  ; load HL with start address of table – label AAAA
0B1A  0606      LD      B,#06      ; load B with number of text entries
;
0B1C  1603      LD      D,#03      ; load D with task number – label BBBB
0B1E  5E          LD      E,(HL)    ; load E with text number
0B1F  CD9F30     CALL  #309F      ; insert task to draw text
0B22  23          INC      HL
0B23  10F7      DJNZ   #AAAA      ; next B – jump to label BBBB
;
0B25  C9          RET      ; return
```

Bovenste pijltje op #768F en dan steeds 2 verder.

Nu versie v1.00-02 opgeslagen.

Nu nog doen:

Aanroep om pijltje te tekenen

Kijken input timer = 0

Als niet dan input timer decrement en return

Als wel dan:

Kijken jump button press:

Ja, dan gamemode aanpassen

Nee dan:

Kijken joystick naar beneden:

Ja, dan aanroep om pijltje te wissen en difficulty increment, als 4 dan reset naar 0

Kijken joystick naar boven:

Ja, dan aanroep om pijltje te wissen en difficulty decrement, als FF dan reset naar 3

Difficulty mode = #64B4. Input timer = #64B5.

Routine om pijltje te tekenen (A bevat karakter om te tekenen)

Aanpassen:

```

0B4D  4F          LD      C,A          ; store A in C for later use
;
0B4E  3AB464      LD      A,(#64B4)     ; load difficulty mode
0B52  47          LD      B,A          ; save to B as loop counter
0B53  04          INC      B           ; increment B
;
0B54  218D76      LD      HL,#768D     ; load HL with position2 above first position
;
0B57  23          INC      HL          ; increment HL – one line down
0B58  23          INC      HL          ; increment HL – one line down
0B59  10FC        DJNZ     ; again until B=0
;
0B5B  71          LD      (HL),C
0B5C  C9          RET                  ; return

```

en:

```

0B25  3E35      LD      A,#35          ; load a with character used as cursor
0B27  CD4D0B     CALL    #0B4D         ; call routine to draw cursor
0B2A  C9          RET                  ; return

```

Ja dat werkt. Tekent een minteken op de eerste keuze: klopt want bij start is de difficulty mode 0!

```

0B2A  3AB564      LD      A,(#64B5)     ; load A with input timer
0B2D  FE00      CP      #00            ; is input timer 0?
0B2F  CA600B     JP      Z,#0B60        ; yes, skip next steps – jump to label AAAA
;
0B32  3D          DEC      A            ; decrement A
0B33  32B564      LD      (#64B5),A     ; store decremented input timer
0B36  C9          RET                  ; return

```

en

```

0B60  3A007C      LD      A,IN0         ; read player 1 input – label AAAA
0B63  FE10      CP      #10            ; is input #10 (jump button pressed)?
0B65  2005      JR      NZ, #BBBB      ; no, skip next steps – jump to label BBBB
;
0B67  210A60      LD      HL,GameMode2
0B6A  34          INC      (HL)         ; increase game mode2 (to 8?)
0B6B  C9          RET                  ; return
;
0B6C  FE04      CP      #04            ; is input >= #04 (joystick down or up)? – label BBBB
0B6E  D8          RET      C            ; no, return
;
0B6F  F5          PUSH    AF            ; store A for later
0B70  3E10      LD      A,#10          ; load A with space character
0B72  CD4D0B     CALL    #0B4D         ; call routine to remove cursor
;

```



0B75	3E0F	LD	A,#0F	; load A with #0F
0B77	32B564	LD	(#64B5),A	; set input timer
;				
0B7A	F1	POP	AF	; restore A
;				
0B7B	FE08	CP	#08	; is input 08? (joystick down)?
0B7D	200E	JR	NZ, #CCCC	; no, skip next steps – jump to label CCCC
;				
0B7F	3AB464	LD	A,(#64B4)	; load A with difficulty mode
0B82	3C	INC	A	; increment A
0B83	FE04	CP	#04	; A == 04?
0B85	2002	JR	NZ,#DDDD	; no, skip next step
0B87	3E00	LD	A,#00	; reset A
0B89	32B464	LD	(#64B4),A	; store A into difficulty mode
0B8C	C9	RET		; return
;				
0B8D	3AB464	LD	A,(#64B4)	; load A with difficulty mode – label CCCC
0B90	3D	DEC	A	; decrement A
0B91	FEFF	CP	#FF	; A == FF?
0B93	2002	JR	NZ,#EEEE	; no, skip next step
0B95	3E03	LD	A,#03	; reset A
0B97	32B464	LD	(#64B4),A	; store A into difficulty mode
0B9A	C9	RET		; return

Ja dat gaat nu goed.

Nee gaat nog niet goed. Wordt altijd op normal mode gespeeld. Dat komt doordat bij start van een spel een heel gebied op 0 wordt gezet. Daarbij wordt ook de #64B4 (difficlyty mode) op nul gezet. Dat aanpassen naar een waarde die niet gewist wordt. Eventjes kijken bij DK Duel welke waarde eventueel gebruikt kan worden. Daar moet de score bijvoorbeeld ook over spellen heen bijgehouden worden. Kan bijvoorbeeld #6235 voor gebruikt worden.

Dus difficulty mode aanpassen van #64B4 naar #6235:

0ADB	3A3562	LD	A,(#6235)	; load difficulty mode – label AAAA
0B4E	3A3562	LD	A,(#6235)	; load difficulty mode
0B7F	3A3562	LD	A,(#6235)	; load A with difficulty mode
0B89	323562	LD	(#6235),A	; store A into difficulty mode
0B8D	3A3562	LD	A,(#6235)	; load A with difficulty mode – label CCCC
0B97	323562	LD	(#6235),A	; store A into difficulty mode

Ja, gaat nu wel goed. Leuke bijkomstigheid is dat de ingestelde difficulty mode onthouden wordt, dus bij de volgende keer staat hij als eerste op de laatst gespeelde difficulty mode.

Nu versie v1.00-03 opgeslagen.

Cursor aanpassen naar een pijltje. Karakter #FA daarvoor gebruikt.

Aanpassen:

0B25 3EFA LD A,#FA ; load a with character used as cursor

De kleuren van het pijltje ook aangepast.

Nu versie v1.00-04 opgeslagen.

Nog aanpassen:

- Difficulty mode d.m.v. letter weergeven achter speed.
- Versienummer aanpassen naar v1.01.
- Vlag in handen Goofy Kong aanpassen.

Versienummer aanpassen:

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

Ja dat is nu goed.

Achter de SPEED ook een letter voor de difficulty mode: N H E I.

Toevoegen achter het stuk code dat vanuit het vrije slot uitgevoerd wordt:

3F89 CD800A CALL #0A80 ; call to animate the flags – call to label AAAA  
3F8C CDB00A CALL #0AB0 ; call to display speed meter – call to label BBBB  
;  
3F8F C9 RET ; return

Aanpassen:

3F8C CD9B0B CALL #0B9B ; call additional code – call to label AAAA

en:

0B5C 1E 18 15 19

en:

0B9B CDB00A CALL #0AB0 ; call to display speed meter  
;  
0B9E 3A3562 LD A,(#6235) ; load difficulty mode  
0BA1 47 LD B,A ; save to B as loop counter  
0BA2 04 INC B ; increment B  
;  
0BA3 215B0B LD HL,#0B5B ; load HL with start table mode characters-1  
;  
0BA6 23 INC HL ; increment HL – next character  
0BA7 10FD DJNZ ; again until B=0

```
;
OBA9  7E          LD    A,(HL)      ; load difficulty mode character
OBAA  328074      LD    (#7480),A    ; display difficulty mode character
;
OBAD  C9          RET                ; return
```

Ja dat gaat goed. Toch wel mooi om dat weer te geven. Nogal teleurstellend als je denkt op een bepaalde difficulty level een mooie score neer te zetten en dan blijkt je op de verkeerde gespeeld te hebben.

**Nu versie v1.00-05 opgeslagen.**

De Goofy Kong aangepast met een meer wapperende finish vlag.

De invincibility weer uitgezet.

**Nu versie v1.00-06 opgeslagen.**

**Dit is de final version v1.01.**