

Donkey Kong Duel

Het idee is een game voor twee personen waarin er onderling strijd gestreden wordt. Dus P1 en P2 spelen om de beurt dezelfde boards en dan kijken wie er beter is.

Elementen:

- ~~Bij starten alleen een 2P mode: dus alleen met P2 knop en voldoende credits 2P mode?~~
- ~~Instelbaar ronde type: JP: 1-2-3-4 of US: 1-2-1-3-1-4~~
- ~~Instelbaar aantal rondes~~
- ~~Instelbaar scoring: punten per 'gewonnen board' of 'totaal score'~~
- ~~Altijd spelen op L=05.~~
- ~~Speler P1 en P2 spelen om de beurt: per ronde of per board? Eerlijk maken.~~
- ~~Speler die moet spelen start het spel met zijn/haar startknop: P1 of P2.~~
- ~~Wanneer een speler een board finished, is de andere speler aan de beurt.~~
- ~~Wanneer een speler dood gaat, is de andere speler aan de beurt.~~
- ~~I.p.v. L=XX het volgende weergeven: R=US/01-1~~
- ~~Aan het einde aangeven wie de winnaar is (meeste rondes, hoogste totaal score).~~
- ~~Tijdens spelen 'gewonnen board': de stand in punten weergeven bij P1 en P2~~
- ~~Tijdens spelen 'gewonnen board': de weergegeven score per board resetten: vergelijken~~
- ~~Tijdens spelen 'totaal score': aangeven wie op dat moment de hoogste score heeft.~~
- ~~Bijhouden statistics per speler en weergeven i.p.v. high score table:~~
 - ~~Hoogste score per board (barrels, pies, springs, rivets)~~
 - ~~Hoogste score voor een complete ronde~~

De game afhandeling met wisselen van spelers vindt hier plaats:

```
; start of main routine when playing a game
; arrive here from #00C9

06FE 3A0A60 LD      A,(GameMode2)      ; load A with game mode2
0701 EF      RST      #28              ; jump based on what the game state is

0702 86 09      ; (0) #0986      ; game start = clears screen, clears sounds, sets screen flip if needed
0704 AB 09      ; (1) #09AB      ; copy player data, set screen, set next game mode based on number of players
0706 D6 09      ; (2) #09D6      ; clears palettes, draws "PLAYER <I>", draws player2 score, draws "2UP" (2 player game only)
0708 FE 09      ; (3) #09FE      ; copy player data into correct area (2 player game only)
070A 1B 0A      ; (4) #0A1B      ; clears palletes, draws "PLAYER <II>", update player2 score, draw "2UP" to screen (2 player game only)
070C 37 0A      ; (5) #0A37      ; updates high score, player score, remaining lives, level, 1UP
070E 63 0A      ; (6) #0A63      ; clears screen and sprites, check for intro screen to run
0710 76 0A      ; (7) #0A76      ; kong climbs ladders and scary music played
0712 DA 0B      ; (8) #0BDA      ; draw goofy kongs, how high can you get, play music
0714 00 00      ; (9)              ; unused
0716 91 0C      ; (A) #0C91      ; clears screen, update timers, draws current screen, sets background music
0718 3C 12      ; (B) #123C      ; set initial mario sprite position and draw remaining lives and level
071A 7A 19      ; (C) #197A      ; for when playing a game. this is the main routine
071C 7C 12      ; (D) #127C      ; mario died. handle mario dying animations
071E F2 12      ; (E) #12F2      ; clear sounds, decrease life, check for and handle game over
0720 44 13      ; (F) #1344      ; clear sounds, clear game start flag, draw game over if needed PL2, set game mode2 accordingly
0722 8F 13      ; (10) #138F      ; check for game over status on a 2 player game
0724 A1 13      ; (11) #13A1      ; check for game over status on a 2 player game
0726 AA 13      ; (12) #13AA      ; flip screen if needed, reset game mode2 to zero, set player 2
0728 BB 13      ; (13) #13BB      ; set player 1, reset game mode2 to zero, set screen flip to not flipped
072A 1E 14      ; (14) #141E      ; draw credits on screen, clears screen and sprites, checks for high score, flips screen if necessary
072C 86 14      ; (15) #1486      ; player enters initials in high score table
072E 15 16      ; (16) #1615      ; handle end of level animations
0730 6B 19      ; (17) #196B      ; clear screen and all sprites, set game mode2 to #12 for player1 or #13 for player2
0732 00 00 00 00 00 00 00 00 00 00 ; unused
```

En is dus afhankelijk van de GameMode2: Door die verschillende waardes mee te geven, kan de volgende stap in de afhandeling bepaald worden. Hier heel goed over nadenken en dan de nieuwe stappen uitschrijven en daar zelf GameMode2 waardes voor definiëren.

De High Score table blijven gebruiken, maar heel anders initieel vullen en daarna vullen met de statistics van het laatst gespeelde duel. Maar geen mogelijkheid om de naam in te geven wanneer er een high score behaald is. Dus die code kan komen te vervallen.

Voor dat je deze main loop ingaat, moet er een scherm zijn waarin de keuze gemaakt wordt voor de board order (US,JP), number of rounds en scoring system. Je zou eerst kunnen focussen op het tonen van dat scherm en het kiezen van de drie settings en het vastleggen in vrije geheugenplaatsen. Dat kun je onafhankelijk testen met de debugger. Als die settings vastgelegd zijn, dan kan je met de main aan de slag om een duel daadwerkelijk te kunnen uitvoeren. Daar heb je namelijk deze settings voor nodig.

Normaal gesproken de loop van Title Screen, High Score Screen en Attract Mode. Wanneer credits ingegooid dan nu alleen 2P te starten. En wanneer je dat doet, dan het extra scherm voor het invoeren van de settings tussenvoegen. De loop daarvoor is de volgende (gestuurd met waarden voor GameModel1):

```
; GameModel1 is 0 when game is turned on, 1 when in attract mode. 2 when credits in waiting for start, 3 when playing game

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
```

Eerst eventjes starten met template voor game en daar de titel aanpassen naar 'DUEL'. En de titel bovenin het scherm aanpassen !

Eerst een basisversie maken. Basisversie van DK Wizardry pakken (versie v0.02) en dan alleen nog de titel en de HIGH SCORE tekst aanpassen.

Op #36B2 staat de HIGH SCORE tekst:

#36B2: 80 76 18 19 17 18 10 23 13 1F 22 15 3F

Veranderen naar DUEL:

#36B2: 80 76 10 10 10 14 25 15 1C 10 10 10 3F

En dan in #3F18 de titel maken:

D U E L

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

Nu versie v0.01 opgeslagen.

Als de game nu opgestart wordt, wordt in het high score scherm de optie getoond met keuze 1 player of 2 players. De teksten daarvan aanpassen.



In het eerste scherm aangeven INSERT COINS en 2 PLAYERS ONLY en de informatie over de coins weghalen.

Eerste aanpassing INSERT COINS:

#37D2: A7 76 19 1E 23 15 22 24 10 13 1F 19 1E 23 3F

Tweede aanpassing 2 PLAYERS ONLY:

#37E1: 0A 77 10 10 20 1C 11 29 15 22 10 10 10 10 13 1F 19 1E 3F naar

#37E1: 0A 77 10 10 02 10 20 1C 11 29 15 22 23 10 1F 1E 1C 29 3F

Derde aanpassing weglaten credits/coins:

```

07A3 ED5B2260 LD DE,(CoinsPerCredit) ; D := CoinsPer2Credits; E := CoinsPerCredit
07A7 216C75 LD HL,#756C ; load HL with screen RAM location
07AA CDAD07 CALL #07AD ; run this sub below twice

07AD 73 LD (HL),E ; draw to screen number of coins needed for 1 player game
07AE 23 INC HL ;
07AF 23 INC HL ; next screen location 2 rows down
07B0 72 LD (HL),D ; draw to screen number of coins needed for 2 player game
07B1 7A LD A,D ; A := D
07B2 D60A SUB #0A ; subtract #A (10 decimal). result == 0 ?
07B4 C2BC07 JP NZ,#07BC ; no, skip next 3 steps

07B7 77 LD (HL),A ; else draw this zero to screen
07B8 3C INC A ; increase A, A := 1 now
07B9 328E75 LD (#758E),A ; draw 1 to screen in front of the zero, so it draws "10" credits needed for 2 players

07BC 110102 LD DE,#0201 ; D := 2, E := 1, used for next loop for 1 player and 2 players
07BF 218C76 LD HL,#768C ; set screen location to draw for next loop if needed
07C2 C9 RET ; return

```

Dit hele stuk niet uitvoeren. Kortsluiten door 07A3 te vervangen door een return:

```
07A3  C9          RET          ; return
07A4  t/m 07C2  allemaal NOP's
```

Dat gaat goed zo.

Nu versie v0.02 opgeslagen.

De code voor het tweede en derde scherm zitten verweven. Per stuk tekst daarom een stukje additionele code die op basis van het aantal credits (als 1: dan teksten voor scherm 2, als meer dan 1 dan teksten voor scherm 3).

De eerste tekst wordt hier neergezet:

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

Dit aanpassen naar:

```
08C1  C3A407    JP      #07A4        ; jump to additional code – jump to label AAAA
```

en

```
07A4  110C03    LD      DE,#030C    ; task code to display "PUSH" onscreen – label AAAA
;
07A7  3A0160    LD      A,(NumCredits) ; load A with number of credits
07AA  FE01      CP      #01          ; == 1 ?
07AC  2003      JR      NZ, #07B1    ; no, skip next step
07AE  111B03    LD      DE,#031B    ; task code to display "INSERT COINS" onscreen
07B1  C3C408    JP      #08C4
```

Dat gaat niet goed. Deze code wordt maar één keer doorlopen: wanneer de credits naar 1 verhoogd wordt. Maar daarna niet meer. Dus het volstaat om dan de PUSH text te vervangen door de INSERT COINS, maar wanneer de tweede coin ingegooid wordt, dan moet deze tekst vervangen worden door PUSH.

Eerste stap oorspronkelijke PUSH melding vervangen:

```
08C1  111B03    LD      DE,#021B    ; task code to display "INSERT COINS" onscreen
```

Dat gaat goed.

Tweede stap teksten daaronder aanpassen:

```

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"

```

De teksten 09 en 0A aanpassen naar:

09: 2 PLAYERS ONLY:

36E6: 0A 77 10 10 02 10 20 1C 11 29 15 22 23 10 1F 1E 1C 29 3F

0A: PUSH 2 PLAYERS BUTTON:

36FD: 2A 77 20 25 23 18 10 02 10 20 1C 11 29 15 22 23 10 12 25 24 24 1F 1E 3F

Derde stap: ENOUGH COINS weergeven

Voor deze derde stap moet de code 08E1 t/m 08E3 aangepast worden naar een jump naar additionele code. Daar kun je dan de juiste tekst direct selecteren zonder de inc. Dus de teksten hoeven niet achter elkaar meer: dus dan op 08D7 niet naar 09 verwijzen, maar naar dezelfde tekst met nummer 1C (die op #37E1 staat).

Aanpassen:

```

08D7  1E1C      LD      E,#1C          ; E := 1C, code for 'TWO PLAYERS ONLY'

```

en

```

08E1  C3A407     JP      #07A4          ; jump to additional code

```

en

```

07A4  110903     LD      DE,#0309        ; load task data for text "ENOUGH COINS"
07A7  CD9F30     CALL    #309F          ; insert task to draw text
07AA  060C      LD      B,#0C          ; B := 0C = 1100 binary
07AC  1E0A      LD      E,#09          ; E := 0A, code for 'ENOUGH COINS'
07AE  C3E408     JP      #08E4          ; jump back

```

en

36E6: A7 76 15 1E 1F 25 17 18 10 13 1F 19 1E 23 3F

Dit gaat nu goed.

De schermen zien er nu zo uit:



Nu versie v0.03 opgeslagen.

Nog wel aanpassingen doen. L=00 alvast op L=05 zetten. De weghalen. Op die plek komt de stand in gewonnen rondes te staan. Misschien ook alvast de tekst met de ronde informatie aan de linkerkant weergeven. En de High score tabel alvast aanpassen naar de informatie die er later komt te staan (en aanpassen van de kleuren van de regels). En daarnaast onmogelijk maken dat P1 ingedrukt wordt. Want dat kan nu nog steeds.

Eerst kijken om de P1 button te disablen:

```
; 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
```

#08FB t/m 08FF NOP's maken.

En dan kan het stuk van P1 afhandeling vrijgegeven worden: #0906 t/m #0918.

Ja dat gaat goed.

Nu versie v0.04 opgeslagen.

Kijken waar de bovenste 1UP en High Score gezet worden. Dat lijkt hier te gebeuren.

```

01EE 320A60 LD (GameMode2),A ; store into game mode 2
01F1 CD530A CALL #0A53 ; draw "1UP" on screen
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
0200 110002 LD DE,#0200 ; load task #2, parameter 0 to display player 1 score
0203 CD9F30 CALL #309F ; insert task
0206 C9 RET ; return

```

Dan de score voor 1UP en de High score niet tonen. En de 2UP wel tekenen.

Routine om 2UP te tekenen:

```

09EE 3E02 LD A,#02 ; load A with "2"
09F0 32E074 LD (#74E0),A ; write to screen
09F3 3E25 LD A,#25 ; load A with "U"
09F5 32C074 LD (#74C0),A ; write to screen
09F8 3E20 LD A,#20 ; load A with "P"
09FA 32A074 LD (#74A0),A ; write to screen
09FD C9 RET ; return

```

Aanpassen:

```

01FA CDEE09 CALL #09EE ; draw "2UP" on screen
01FD t/m 0205 NOP's

```

Dat werkt. Alleen misschien wel bij beiden een score van 000000 laten zien. Bij P1 staat standaard een score van 003700.

Task #2, parameter 1 zou de player 2 score moeten laten zien. Als dit aangepast wordt, wordt er helemaal geen score getoond. Misschien omdat de player 2 score nog niet gevuld is? Even debuggen wat er in de waarden zit.

```

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

```

```

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

```

Ja in de player 2 score staat AA AA AA. Even kijken waardoor dat gezet wordt. Blijkbaar op #026D. Nee dat is wissen naar 0. Maar wel op #01CF.

Gevonden. Dit is de data table waarmee de P1 score, P2 score en High score geïnitieerd worden.

```

; table data used below in 01C6

```

```

01BA 00 37 00 AA AA AA 50 76 00

```

Daar dus aanpassen. Alle drie op 00 00 00 zetten. Ook de high score. Handig als deze later nog ergens voor hergebruikt wordt. Kan bijvoorbeeld voor de settings voor het spel.

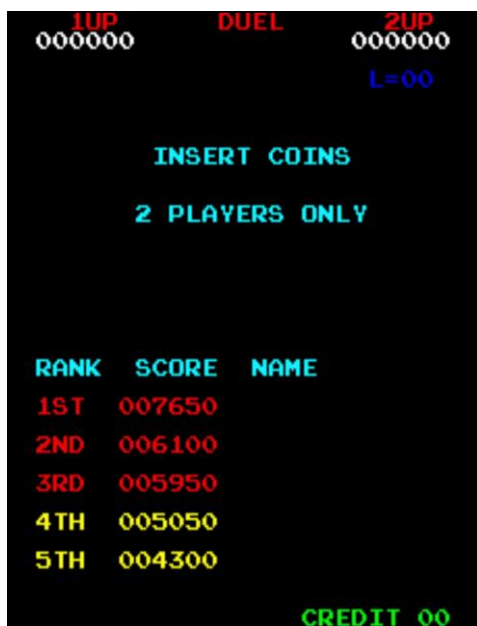
Ja werkt nu wel. Wel alles nog eventjes samenbrengen.

```
01FA  CDB107      CALL  #07B1      ; call additional code – call label AAAA
01FD  CD9F30      CALL  #309F      ; insert task
0200  110002      LD     DE,#0200   ; load task #2, parameter 0 to display player 1 score
0203  CD9F30      CALL  #309F      ; insert task
0206  C9          RET              ; return
```

en

```
07B1  CDEE09      CALL  #09EE      ; draw “2UP” on screen – label AAAA
07B4  110102      LD     DE,#0201   ; load task #2, parameter 1 to display player 2 score
07B7  C9          RET              ; return
```

Ja dat gaat goed. Ziet er nu aan het begin zo uit:



Nu versie v0.05 opgeslagen.

Nu nog de default high score tabel aanpassen. Ander formaat aan het begin laden. Doet verder niets. Updaten later met nieuwe informatie over de hoogste scores per board.

De default high score table staat hier:

```
3565:  94 77 01 23 24 10 10 00 00 07 06 05 00 10 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 50 76 00 F4 76
3587:  96 77 02 1E 14 10 10 00 00 06 01 00 00 10 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 00 61 00 F6 76
35A9:  98 77 03 22 14 10 10 00 00 05 09 05 00 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 50 59 00 F8 76
35CB:  9A 77 04 24 18 10 10 00 00 05 00 05 00 10 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 50 50 00 FA 76
35EE:  9C 77 05 24 18 10 10 00 00 04 03 00 00 10 10 10 10 10 10 10 10 10 10 10 10 3F 00 00 43 00 FC 76
```

lets maken als:

		PLAYER1	PLAYER2	
RESULT	00	-	00	
BARRELS	00000		00000	
PIES	00000		00000	
SPRINGS	00000		00000	
RIVETS	00000		00000	


```

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 22 19 26 15 24 23 10 10 10 10 00 00 00 00 00 10 10 10 00 00 00 00 00 10 10 3F 00 00 43 00 FC 76

```

Dat is goed zo.

Nu nog de twee titels PLAYER1 en PLAYER2

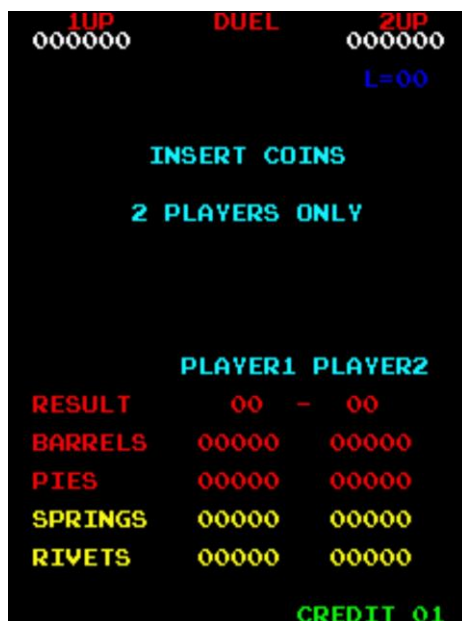
De oude titels staan op #379E:

```
92 77 22 11 1E 1B 10 10 23 13 1F 22 15 10 10 1E 11 1D 15 10 10 10 10 3F
```

Nu veranderen in:

```
32 77 10 10 10 10 10 10 20 1C 11 29 15 22 01 10 20 1C 11 29 15 22 02 3F
```

Is nu goed. Ziet er nu zo uit:



Nu versie v0.06 opgeslagen.

De kleuren in het palette veranderen. Uitzoeken hoe dat werkt. Het kleurenpalette bestaat uit 32 lijnen met elk een kleur. Die zijn te zien in het opstartpatroon met alle gekleurde nullen. Geen idee hoe de twee palette files (c-2j.bpr en c-2k.bpr) mappen op deze lijnen. Proefondervindelijk uitdokteren.

Dit zit in het bestand v-5e.bpr: De onderstaande waardes aanpassen zorgt ervoor dat de twee rode lijnen geel worden.

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

Dat ziet er nu zo uit.

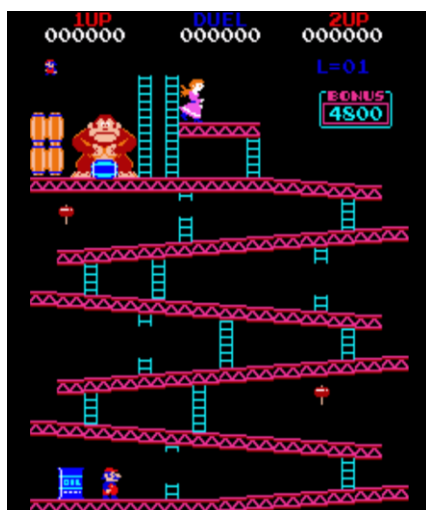


Nog wel eventjes testen of deze aanpassing geen invloed heeft op de andere palettes van de verschillende velden.

Nu versie v0.07 opgeslagen.

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

Door het aanpassen van deze twee waarden wordt de titel bovenin donkerblauw. Dat ziet er best gaaf uit:



Getest en dat gaat in alle boards goed.

Nu versie v0.08 opgeslagen.

Kijken of in plaats van het aantal jumpmans (heeft in deze variant geen betekenis) daar de tekst R=US/01-1 geplaatst kan worden (of wellicht met nullen).

De L=01 wordt daar neergezet vanuit het stukje code van #06D7 t/m #06FD als onderdeel van task #6. In deze task worden ook het aantal jumpmans getekend vanuit het stukje code van #06C7 t/m #06D5.

Eerst vaststellen wat het maximale aantal ronden is dat ingesteld kan worden. Als groter dan 9 dan moet ook de tientallen berekend en getekend worden. Als we tot max 9 gaan dan heb je het over 9x6=54 boarden. Dat is al héél lang voor een spel. Misschien zelfs lager maximeren.. Maar dan geen

decimaal toepassen. Dus dan wordt de tekst: R=US/1-1. Oh schuine streep bestaat niet. Wil eigenlijk niet de sprites aanpassen. Dan wordt het R=US-1-1

Nu nog bepalen welke geheugenplaatsen gebruiken om bij te houden welke boardvolgorde, welke ronde en welk board.

#6232-#6279 niet gebruikt? Dit testen door te spelen met een wpset in de debugger. Eerst getest met de eerste 16 geheugenplaatsen vanaf #6232. En die worden alleen helemaal aan het begin op 0 gezet, maar daarna verder niet gebruikt.

Dus:

- Boardvolgorde: #6232 0 = US, 1 = JP.
- Rondenummer: #6233 0 t/m 9.
- Boardnummer: #6234 0 t/m 6 (US) en 0 t/m 4 (JP).

Aanpassen:

06C7	218377	LD	HL,#7783	; load HL with screen location for "R="
06CA	3622	LD	(HL),#22	; draw "R"
06CC	216377	LD	HL,#7763	; next location
06CF	3634	LD	(HL),#34	; draw "="
06D1	11E0FF	LD	DE,#FFE0	; load DE with offset for each column
06D4	CD7B0A	JP	#0A7B	; call to additional code – jump to label AAAA

en:

0A7B	19	ADD	HL,DE	; add offset for next column – label AAAA
0A7C	3A3262	LD	A,(#6232)	; load A with board order
0A7F	FE00	CP	#00	; board order == 00?
0A81	2007	JR	NZ,BBBB	; no, skip next steps. – jump to label BBBB
;				
0A83	3625	LD	(HL),#25	; draw "U"
0A85	19	ADD	HL,DE	; add offset for next column
0A86	3623	LD	(HL),#23	; draw "S"
0A88	1805	JR	#CCCC	; skip next steps – jump to label CCCC
;				
0A8A	361A	LD	(HL),#1A	; draw "J" – label BBBB
0A8C	19	ADD	HL,DE	; add offset for next column
0A8D	3620	LD	(HL),#20	; draw "P"
;				
0A8F	19	ADD	HL,DE	; add offset for next column – label CCCC
0A90	362C	LD	(HL),#2C	; draw "-"
0A92	C9	RET		; return

Dat gaat goed. Nu nog de rest weergeven:

0A92	19	ADD	HL,DE	; add offset for next column
0A93	3A3362	LD	A,#6233	; load A with round number
0A96	77	LD	(HL),A	; draw round number
;				
0A97	19	ADD	HL,DE	; add offset for next column – label CCCC
0A98	362C	LD	(HL),#2C	; draw “-“
;				
0A9A	19	ADD	HL,DE	; add offset for next column
0A9B	3A3462	LD	A,#6234	; load A with board number
0A9E	77	LD	(HL),A	; draw board number
;				
0AAF	C9	RET		; return

Dit gaat nog niet goed. Geeft alleen de eerste 0, maar niet daarna nog de -0. Hoe kan dat? Wordt dat nog weer ergens door overschreven? Ja, wordt ergens door overschreven. Want in het How High scherm is het wel te zien. Ook de trappetjes waar DK naar boven gaat, worden daar normaal gesproken getekend en weer verwijderd. Dus misschien doet dat dat wel. Maar dan kijken of de tekst aan de andere kant getoond kan worden. Onder of boven de L=00? En dan misschien daarvan dan ook nog de kleur veranderen?

Misschien 2 plekken naar links:

06C7	21C377	LD	HL,#77C3	; load HL with screen location for “R=”
06CA	3622	LD	(HL),#22	; draw "R"
06CC	21A377	LD	HL,#77A3	; next location

Nee dat past niet. Dan wordt de R niet getekend. Dus dan maar naar rechts verplaatsen. Kijken of dat dan wel goed gaat. Ook niet echt mooi en de tekst is ook wel erg lang. Daarom de R= ervoor weglaten en de tekst als volgt: US:0-0.

Dan de code aanpassen:

06C7	218377	LD	HL,#7783	; load HL with screen location for “R=”
06CA	3622	LD	(HL),#22	; draw "R"
06CC	216377	LD	HL,#7763	; next location
06CF	3634	LD	(HL),#34	; draw “=”

06C7	11E0FF	LD	DE,#FFE0	; load DE with offset for each column
06CA	218377	LD	HL,#7783	; load HL with screen location for “US:”
06CE	3A3262	LD	A,(#6232)	; load A with board order
06D1	CD7B0A	CALL	#0A7B	; call to additional code – jump to label AAAA

06D4 t/m 06D6 NOP's

en:

0A7B	FE00	CP	#00	; board order == 00?
0A7D	2007	JR	NZ,BBBB	; no, skip next steps. – jump to label BBBB
;				
0A7F	3625	LD	(HL),#25	; draw “U”
0A81	19	ADD	HL,DE	; add offset for next column
0A82	3623	LD	(HL),#23	; draw “S”
0A84	1805	JR	#CCCC	; skip next steps – jump to label CCCC
;				
0A86	361A	LD	(HL),#1A	; draw “J” – label BBBB
0A88	19	ADD	HL,DE	; add offset for next column
0A89	3620	LD	(HL),#20	; draw “P”
;				
0A8B	19	ADD	HL,DE	; add offset for next column – label CCCC
0A8C	36BF	LD	(HL),#BF	; draw “/”
0A8E	19	ADD	HL,DE	; add offset for next column
0A8F	3A3362	LD	A,#6233	; load A with round number
0A92	77	LD	(HL),A	; draw round number
;				
0A93	19	ADD	HL,DE	; add offset for next column – label CCCC
0A94	362C	LD	(HL),#2C	; draw “-”
;				
0A96	19	ADD	HL,DE	; add offset for next column
0A97	3A3462	LD	A,#6234	; load A with board number
0A9A	77	LD	(HL),A	; draw board number
;				
0A9B	C9	RET		; return

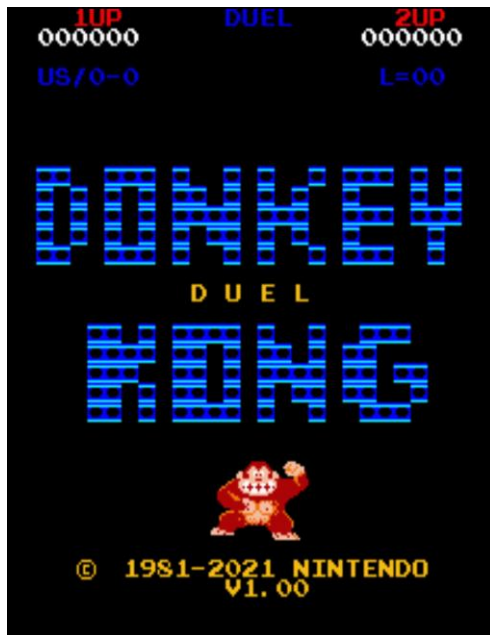
Ja dat ziet er nu wel goed uit. Alleen misschien de : nog aanpassen naar een schuine streep.

Nu versie v0.09 opgeslagen.

Een graphic voor de schuine streep toegevoegd. Dat ziet er wel beter uit. Die staat op #BF. Hierboven aangepast in de code.

Nu versie v0.10 opgeslagen.

Ziet er nu zo uit:



Nu de instellingen gaan doen van de board volgorde, het aantal ronden en de wijze van scoren. Dit op een leeg scherm. Kop weergeven met instructie. Door de drie settings lopen met UP en DOWN. De geselecteerde optie gaat knipperen. Dan met behulp van LINKS en RECHTS door de opties van het betreffende element lopen. En JUMP drukken als alles juist ingesteld.

Iets als dit:

SELECT OPTIONS

BOARD ORDER	US	(of JP)
ROUNDS #	3	(of 2 t/m 9)
SCORING	BOARD	(of TOTAL)

PRESS JUMP TO START

En dan de gemaakte keuze opslaan (steeds meteen doen als er iets nieuws geselecteerd wordt) in de daarvoor gereserveerde geheugenplaatsen.

Eerst nadenken hoe je dit voor het spelen selecteert. Is dat in de main loop die ook de credits afhandelt? Die werkt met GameMode. Daar dan een nieuwe voor bedenken en als die actief dan naar nieuwe code springen die dit scherm implementeert?

Buitenste loop is gebaseerd op GameMode1:

; GameMode1 is 0 when game is turned on, 1 when in attract mode. 2 when credits in waiting for start, 3 when playing game

```

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

```

En dan is de main routine voor het overall spel:

```
; start of main routine when playing a game
; arrive here from #00C9

06FE 3A0A60 LD A,(GameMode2) ; load A with game mode2
0701 EF RST #28 ; jump based on what the game state is

0702 86 09 ; (0) #0986 ; game start = clears screen, clears sounds, sets screen flip if needed
0704 AB 09 ; (1) #09AB ; copy player data, set screen, set next game mode based on number of players
0706 D6 09 ; (2) #09D6 ; clears palettes, draws "PLAYER <I>", draws player2 score, draws "2UP" (2 player game only)
0708 FE 09 ; (3) #09FE ; copy player data into correct area (2 player game only)
070A 1B 0A ; (4) #0A1B ; clears palletes, draws "PLAYER <II>", update player2 score, draw "2UP" to screen (2 player game only)
070C 37 0A ; (5) #0A37 ; updates high score, player score, remaining lives, level, 1UP
070E 63 0A ; (6) #0A63 ; clears screen and sprites, check for intro screen to run
0710 76 0A ; (7) #0A76 ; kong climbs ladders and scary music played
0712 DA 0B ; (8) #0BDA ; draw goofy kongs, how high can you get, play music
0714 00 00 ; (9) ; unused
0716 91 0C ; (A) #0C91 ; clears screen, update timers, draws current screen, sets background music
0718 3C 12 ; (B) #123C ; set initial mario sprite position and draw remaining lives and level
071A 7A 19 ; (C) #197A ; for when playing a game. this is the main routine
071C 7C 12 ; (D) #127C ; mario died. handle mario dying animations
071E F2 12 ; (E) #12F2 ; clear sounds, decrease life, check for and handle game over
0720 44 13 ; (F) #1344 ; clear sounds, clear game start flag, draw game over if needed PL2, set game mode2 accordingly
0722 8F 13 ; (10) #138F ; check for game over status on a 2 player game
0724 A1 13 ; (11) #13A1 ; check for game over status on a 2 player game
0726 AA 13 ; (12) #13AA ; flip screen if needed, reset game mode2 to zero, set player 2
0728 BB 13 ; (13) #13BB ; set player 1, reset game mode2 to zero, set screen flip to not flipped
072A 1E 14 ; (14) #141E ; draw credits on screen, clears screen and sprites, checks for high score, flips screen if necessary
072C 86 14 ; (15) #1486 ; player enters initials in high score table
072E 15 16 ; (16) #1615 ; handle end of level animations
0730 6B 19 ; (17) #196B ; clear screen and all sprites, set game mode2 to #12 for player1 or #13 for player2
0732 00 00 00 00 00 00 00 00 ; unused
```

Dus dat is o.a. wisselen P1 en P2 in 2 player mode, eventueel de intro scene (ladder climb) afspelen, how high can you get scherm, echte spel spelen, afhandelen van dying, enz, enz. Deze main routine zal aangepast moeten worden. Hier moet de afhandeling van het spel gemaakt worden, inclusief bijhouden van de stand, switchen van de beurten, eventueel draaien van het scherm in cocktail mode voor P2, enz, enz en aan het einde het bijwerken van de high score table met het overzicht van de scores. En dat allemaal door GameMode2 steeds bij te werken.

Klein beginnen met GameMode2 = 0: game start. En dan eventueel Kong die de ladder beklimt (nee dat kan niet want die code is al verwijderd en deels hergebruikt). En dan het selectiescherm en dan de play logica. En dan steeds de GameMode2 ophogen naar een nieuwe waarde waardoor een nieuwe fase in de game bereikt wordt. Zal wel deels met vallen en opstaan gaan.

#600A = geheugenplaats voor GameMode2.

Eerst met debugger kijken met welke GameMode2 voor het eerst na het starten van het spel (na drukken op P2) gestart wordt.

Volgorde van GameMode2 is nu:

- 0 - Game Start
- 1 - Copy player data, set screen, set GameMode depending on # players
- 2 - Clear palettes, display Player 1 and P2 score (only 2P game)
- 5 - Updates high score, player score, remaining lives, level, 1UP
- 6 (heleboel keer) - Clear screen and sprites and check for Kong climbing (is al verwijderd)

GameMode2 = 6 wacht op WaitTimerMSB (dus de tekst Player 1 staat tijdje op scherm).

De Game Start hoeft in principe nog niet op 0. We kunnen bij GameMode 0 ook beginnen met de clear screen en een eenvoudig scherm dat oneindig lang blijft open staan totdat we op de JUMP button drukken. Als dat werkt, dan kunnen we daarna daar het selectiescherm gaan opbouwen. Daarnaast komen er een aantal grote blokken vrij omdat we die niet meer gebruiken. O.a. voor het invoeren van de credentials aan het einde. Plus de blokken die origineel deze verschillende stappen implementeren. Allemaal vervangen door nieuwe stukken code.

Dus de call #0 aanpassen.

Laat die verwijzen naar #0AA0 en daar dan dit selectiescherm opbouwen. Dan de GameMode2 aanpassen naar 1 en dan zou het spel gewoon moeten gaan spelen.

0702 A00A ; (0) 0AA0 – game start, show options screen

en:

```
0AA0    CD5208            CALL    #0852            ; clear screen and all sprites
0AA3    CD1C01            CALL    #011C            ; clear all sounds
;
0AA6    3A1060            LD        A,(InputState) ; load A with input
0AA9    CB7F              BIT        7,A                ; is jump button pressed?
0AAB    2805              JR        Z,AAAA            ; no, skip next steps – jump to label AAAA
;
0AAD    210A60            LD        HL,600A           ; load HL with GameMode2 address
0AB0    3601              LD        (HL),#01          ; set game mode 2 to 1
;
0AB2    C9                RET                        ; return – label AAAA
```

Werkt wel, maar doet een complete wis van het scherm. Dat is niet wat we willen. De bovenkant mag blijven staan. Is dat wellicht de andere clear screen and all sprites: op #0874?

Dan aanpassen:

0AA0 CD7408 CALL #0874 ; clear screen and all sprites

Ja dat gaat nu wel goed.

Dan nu kijken of het mogelijk is om het keuze scherm te implementeren. Eerst steeds het scherm tekenen en dan de input bekijken. Als input dan verwerken en steeds weer opnieuw erdoor heen totdat de jump button ingedrukt is.

Maar dan moet het wissen van het scherm GameMode2 0 zijn en die moet dan de GameMode2 ophogen naar 1 en daar moet dan de code komen om de teksten van het keuzescherm te plaatsen.

Dus GameMode2 = 0:

0702 A00A ; (0) 0AA0 – game start, clear screen

en:

```
0AA0  CD7408      CALL  #0874      ; clear screen and all sprites
0AA3  CD1C01      CALL  #011C      ; clear all sounds
;
0AA6  210A60      LD     HL,600A      ; load HL with GameMode2 address
0AA9  3601         LD     (HL),#01    ; set game mode 2 to 1
;
0AAB  C9          RET              ; return
```

Eerst de vaste teksten op het scherm afdrucken. Dat kan door deze te definiëren in een datablok en dan een eigen tekst routine aan te roepen:

```
3F38  21403F      LD     HL,#3F40      ; load HL with start of table
3F3B  C3EC05      JP     #05EC      ; jump to text draw routine
3F3E  C9          RET              ; return
```

```
3F40  52 3F              ; table index "SELECT OPTIONS"
3F42  63 3F              ; table index "BOARD ORDER"
3F44  71 3F              ; table index "ROUNDS"
3F46  7A 3F              ; table index "SCORING"
3F48  84 3F              ; table index "PRESS JUMP TO DUEL"
3F4A  99 3F              ; table index "US"
3F4C  C8 3F              ; table index "JP"
3F4E  CD 3F              ; table index "BOARD"
3F50  D5 3F              ; table index "TOTAL"
;
3F52  C7 76 23 15 1C 15 13 24 10 1F 20 24 19 1F 1E 23 3F      ; SELECT OPTIONS
3F63  0B 77 12 1F 11 22 14 10 1F 22 14 15 22 3F              ; BOARD ORDER
3F71  0D 77 22 1F 25 1E 14 23 3F              ; ROUNDS
3F7A  0F 77 23 13 1F 22 19 1E 17 3F              ; SCORING
3F84  13 77 20 22 15 23 23 10 1A 25 1D 20 10 24 1F 10 14 25 15 1C 3F      ; PRESS JUMP TO DUEL
3F99  0B 75 25 23 3F              ; US
3FC8  0B 75 1A 20 3F              ; JP
3FCD  6F 75 12 1F 11 22 14 3F              ; BOARD
3FD5  6F 75 24 1F 24 11 1C 3F              ; TOTAL
```

Testen met GameMode2 = 1:

```
0704  AC0A          ; (0) 0AAC – game start, clear screen
```

en:

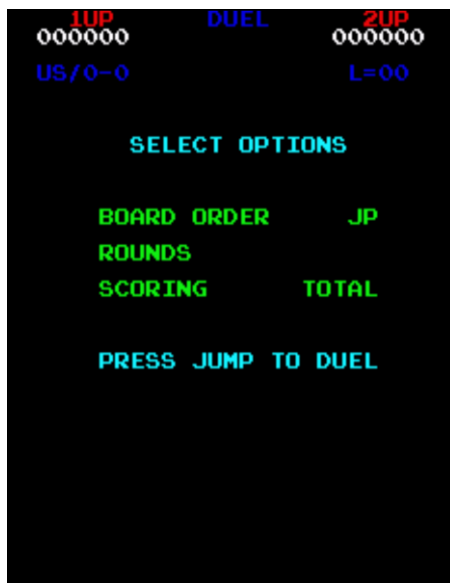
```
0AAC  3E00          LD     A,#00      ; load A with index 0
0AAE  0602          LD     B,#02      ; For B = 1 to 5
0AB0  F5           PUSH  AF        ; save AF
```

0AB1	C5	PUSH	BC	; save BC
0AB2	CD383F	CALL	3F38	; call additional text draw routine
0AB5	C1	POP	BC	; restore BC
0AB6	F1	POP	AF	; restore AF
0AB7	3C	INC	A	; load A with next index
0AB8	10F6	DJNZ	#0AB0	; next B
;				
0ABA	3A1060	LD	A,(InputState)	; load A with input
0ABD	CB7F	BIT	7,A	; is jump button pressed?
0ABF	2805	JR	Z,AAAA	; no, skip next steps – jump to label AAAA
;				
0AC1	210A60	LD	HL,600A	; load HL with GameMode2 address
0AC4	3602	LD	(HL),#02	; set game mode 2 to 2
;				
0AC6	C9	RET		; return – label AAAA

Dit tekent de 5 vaste teksten.

Ook eventjes gedaan met de 9 teksten om een beeld te krijgen hoe het er dan uit ziet.

Dat ziet er nu zo uit:



Nu versie v0.11 opgeslagen.

Nu nog zo maken dat na het tekenen van de 5 vaste teksten, de drie variabele teksten getekend worden op basis van de ingestelde variabelen. En dan daarna de regelselectie doen op basis van een variabele en dan op de juiste hoogte een pijltje tekenen. En daarna dan aanpassen van de geselecteerde regel doen.

De variabele onderdelen variabel weergeven op basis van de variabelen die daarvoor gereserveerd zijn:

- Boardvolgorde: #6232 0 = US, 1 = JP.
- Rondenummer Actueel: #6233 0 t/m 9.
- Boardnummer Actueel: #6234 0 t/m 6 (US) en 0 t/m 4 (JP).
- Rondenummer Max: #6235 0 t/m 9.
- Scoringmethode: #6236 0 = BOARDS, 1 = TOTAL

Aanpassen:

```

OABA 3A3262 LD A,(#6232) ; load A with board order
OABD FE00 CP #00 ; A == 00?
OABF 2003 JR NZ,#AAAA ; no, skip next steps
;
OAC1 3E05 LD A,#05 ; load A with index 05 = "US"
OAC3 1802 JR #BBBB ; skip next step
;
OAC5 3E06 LD A,#06 ; load A with index 06 = "JP"
OAC7 CD383F CALL 3F38 ; call additional text draw routine
;
OACA 3A3562 LD A,(#6235) ; load A with board max
OACD 21ED74 LD HL,#74ED ; load HL with screen location for max rounds
OAD0 77 LD (HL),A
;
OAD1 3A3662 LD A,(#6236) ; load A with scoring method
OAD4 FE00 CP #00 ; A == 00?
OAD6 2003 JR NZ,#AAAA ; no, skip next steps
;
OAD8 3E07 LD A,#07 ; load A with index 07 = "BOARDS"
OADA 1802 JR #BBBB ; skip next step
;
OADC 3E08 LD A,#08 ; load A with index 06 = "TOTAL"
OADE CD383F CALL 3F38 ; call additional text draw routine
;
OAE1 3A1060 LD A,(InputState) ; load A with input
OAE4 CB7F BIT 7,A ; is jump button pressed?
OAE6 2805 JR Z,AAAA ; no, skip next steps – jump to label AAAA
;
OAE8 210A60 LD HL,600A ; load HL with GameMode2 address
OAEB 3602 LD (HL),#02 ; set game mode 2 to 2
;
OAE D C9 RET ; return – label AAAA

```

Dit gaat goed.

Een pijltje (naar rechts) maken in de graphics files. Gedaan op positie #BE.

Nu versie v0.12 opgeslagen.

Even als test een pijltje tekenen.

Aanpassen:

```
0AED 214B77    LD    HL,#774B    ; load HL with screen location for selection arrow
0AF0 36BE      LD    (HL),#BE    ; display selection arrow
;
0AF2 C9        RET                ; return
```

Dat ziet er goed uit:



Dus dan de pijltjes afhankelijk maken van een variabele.

- Boardvolgorde: #6232 0 = US, 1 = JP.
- Rondenummer Actueel: #6233 0 t/m 9.
- Boardnummer Actueel: #6234 0 t/m 6 (US) en 0 t/m 4 (JP).
- Rondenummer Max: #6235 0 t/m 9.
- Scoringmethode: #6236 0 = BOARDS, 1 = TOTAL
- Selected Option: #6237 0 = volgorde, 1 = ronden, 2= scoring

En dan pijltje tekenen op #774B + (#6237)x2

```
0AE1 214977    LD    HL,#7749    ; load HL with screen location for selection arrow
0AE4 3A3762    LD    A,(#6237)    ; load A with selected option
0AE7 3C        INC    A            ; increment A
0AE8 47        LD    B,A          ; For B = 1 to A
0AE9 23        INC    HL          ; select display position 1 lower
0AEA 23        INC    HL          ; select display position 1 lower
0AEB 10FC      DJNZ   #0AF5        ; next B
0AED 36BE      LD    (HL),#BE    ; display selection arrow
;
0AEF 3A1060    LD    A,(InputState) ; load A with input
```

0AF2	CB7F	BIT	7,A	; is jump button pressed?
0AF4	2805	JR	Z,AAAA	; no, skip next steps – jump to label AAAA
;				
0AF6	210A60	LD	HL,600A	; load HL with GameMode2 address
0AF9	3602	LD	(HL),#02	; set game mode 2 to 2
;				
0AFB	C9	RET		; return – label AAAA

Dat gaat nu goed. Nu nog maken dat de pijltjes omhoog en omlaag gecheckt worden en wanneer deze gedrukt worden, dan de 6239 verhogen/verlagen en eventueel cyclen.

Aanpassen:

0AEF	3A1060	LD	A,(InputState)	; load A with input
0AF2	CB7F	BIT	7,A	; is jump button pressed?
0AF4	C20C0B	JP	NZ,0B0C	; yes, jump to label AAAA
;				
0AF7	CB5F	BIT	3,A	; is joystick pushed down ?
0AF9	C2120B	JP	NZ,0B12	; yes, jump to label BBBB
;				
0AFC	CB57	BIT	2,A	; is joystick pushed up ?
0AFE	C2CCCC	JP	NZ,CCCC	; yes, jump to label CCCC
;				
0B01	CB4F	BIT	1,A	; is joystick pushed left ?
0B03	C2DDDD	JP	NZ,DDDD	; yes, jump to label DDDD
;				
0B06	CB47	BIT	0,A	; is joystick pushed right?
0B08	C2EEEE	JP	NZ,EEEE	; yes, jump to label EEEE
;				
0B0B	C9	RET		; return
;				
0B0C	210A60	LD	HL,600A	; load HL with GameMode2 address – label AAAA
0B0F	3602	LD	(HL),#02	; set game mode 2 to 2
0B11	C9	RET		; return
;				
0B12	3A3762	LD	A,(#6237)	; load A with selected option – label BBBB
0B15	3C	INC	A	; increment A
0B16	FE03	CP	#03	; A == 03?
0B18	2002	JR	NZ,#0B1C	; no, skip next step
0B1A	3E00	LD	A,#00	; load A with 00
0B1C	323762	LD	(#6237),A	; store in selected option
0B1F	C9	RET		; return

Dit gaat in principe goed, maar wel een probleem: de oude pijl wordt niet gewist. En daarnaast lijkt het ook zo te zijn dat het scrollen wel heel snel gaat. Maar dat kan nu niet zo goed gezien worden.

Aanpassen:

0B12	CDCB0B	CALL	#0BCB	; call routine to clear selection arrow – label BBBB
0B15	3A3762	LD	A,(#6237)	; load A with selected option
0B18	3C	INC	A	; increment A
0B19	FE03	CP	#03	; A == 03?
0B1B	2002	JR	NZ,#0B1C	; no, skip next step
0B1D	3E00	LD	A,#00	; load A with 00
0B1F	323762	LD	(#6237),A	; store in selected option
0B22	C9	RET		; return

en:

0BCB	214977	LD	HL,#7749	; load HL with screen location for selection arrow
0BCE	3A3762	LD	A,(#6237)	; load A with selected option
0BD1	3C	INC	A	; increment A
0BD2	47	LD	B,A	; For B = 1 to A
0BD3	23	INC	HL	; select display position 1 lower
0BD4	23	INC	HL	; select display position 1 lower
0BD5	10FC	DJNZ	#0AF5	; next B
0BD7	3610	LD	(HL),#10	; clear selection arrow
0BD9	C9	RET		; return

Het Wissen gaat nu goed, maar het selecteren gaat wel heel erg snel: je ziet gewoon een heleboel tussenliggende geselecteerd worden. Dus eigenlijk bij indrukken van een knop een timer zetten en dan de joystick niet uitlezen zolang de timer nog niet op nul staat: dan alleen de timer verlagen en een return doen. De jump check wel altijd doen.

Bij het ingeven van de initialen gebruiken ze daarvoor de HSCursorDelay: #6030. Lijkt erop dat ze die op #A zetten. Kijken of dat een goede waarde is. Maar nu eerst nog dit ertussen krijgen.

0AEF	3A1060	LD	A,(InputState)	; load A with input
0AF2	CB7F	BIT	7,A	; is jump button pressed?
0AF4	C21A0B	JP	NZ,0B1A	; yes, jump to label AAAA
;				
0AF7	47	LD	B,A	; save A to B
0AF8	3A3060	LD	A,(#6030)	; load A with HSCursorDelay
0AFB	FE00	CP	#00	; A == 00?
0AFD	2805	JR	Z,#YYYY	; yes, no delay, check joystick – jump to label YYYY
;				
0AFF	3D	DEC	A	; decrement A
0B00	323060	LD	(#6030),A	; store into HSCursorDelay
0B03	C9	RET		; return, do not check joystick
;				
0B04	78	LD	A,B	; restore A from B – label YYYY
0B05	CB5F	BIT	3,A	; is joystick pushed down ?
0B07	C2200B	JP	NZ,0B20	; yes, jump to label BBBB
;				
0B0A	CB57	BIT	2,A	; is joystick pushed up ?
0B0C	C2CCCC	JP	NZ,CCCC	; yes, jump to label CCCC

```

;
0B0F  CB4F      BIT    1,A      ; is joystick pushed left ?
0B11  C2DDDD    JP     NZ,DDDD  ; yes, jump to label DDDD
;
0B14  CB47      BIT    0,A      ; is joystick pushed right?
0B16  C2EEEE    JP     NZ,EEEE  ; yes, jump to label EEEE
;
0B19  C9        RET             ; return
;
0B1A  210A60    LD     HL,600A   ; load HL with GameMode2 address – label AAAA
0B1D  3602      LD     (HL),#02  ; set game mode 2 to 2
0B1F  C9        RET             ; return
;
0B20  CDCB0B    CALL   #0BCB     ; call routine to clear selection arrow – label BBBB
0B23  3A3762    LD     A,(#6237) ; load A with selected option
0B26  3C        INC     A        ; increment A
0B27  FE03      CP     #03       ; A == 03?
0B29  2002      JR     NZ,#0B2D  ; no, skip next step
0B2B  3E00      LD     A,#00     ; load A with 00
0B2D  323762    LD     (#6237),A ; store in selected option
0B30  3E0A      LD     A,#0A     ; load A with #0A
0B32  323060    LD     (#6030),A ; set HSCursorDelay
0B35  C9        RET             ; return

```

Ja, dat werkt.

Nu versie v0.13 opgeslagen.

Nu ook de up implementeren.

```

0B0C  C2360B    JP     NZ,0B36   ; yes, jump to label CCCC

```

en

```

0B36  CDCB0B    CALL   #0BCB     ; call routine to clear selection arrow – label CCCC
0B39  3A3762    LD     A,(#6237) ; load A with selected option
0B3C  FE00      CP     #00       ; A == 0?
0B3E  2002      JR     NZ,#0B42  ; no, skip next step
0B40  3E03      LD     A,#03     ; load A with 03
0B42  3D        DEC     A        ; decrement A
0B43  323762    LD     (#6237),A ; store in selected option
0B46  3E0A      LD     A,#0A     ; load A with #0A
0B48  323060    LD     (#6030),A ; set HSCursorDelay
0B4B  C9        RET             ; return

```

Ja, dat werkt.

Nu versie v0.14 opgeslagen.

Nu dan het aanpassen van de settings met left en right.

0B0F	CB4F	BIT	1,A	; is joystick pushed left ?
0B11	C24C0B	JP	NZ,0B4C	; yes, jump to label DDDD
;				
0B14	CB47	BIT	0,A	; is joystick pushed right?
0B16	C25C0B	JP	NZ,0B5C	; yes, jump to label EEEE
en				
0B4C	3A3762	LD	A,(#6237)	; load A with selected option
0B4F	FE00	CP	00	; A == 00 ?
0B51	CA6C0B	JP	Z,0B6C	; yes, setting board order – jump to label AAAA
0B54	FE02	CP	02	; A == 02 ?
0B56	CA770B	JP	Z,0B77	; yes, setting scoring– jump to label BBBB
0B59	C3820B	JP	#0B82	; no, setting max rounds – jump to label CCCC
;				
0B5C	3A3762	LD	A,(#6237)	; load A with selected option
0B5F	FE00	CP	00	; A == 00 ?
0B61	CA6C0B	JP	Z,0B6C	; yes, setting board order – jump to label AAAA
0B64	FE02	CP	02	; A == 02 ?
0B66	CA770B	JP	Z,0B77	; yes, setting scoring– jump to label BBBB
0B69	C3920B	JP	#0B92	; no, setting max rounds – jump to label DDDD
;				
0B6C	3A3262	LD	A,(#6232)	; load A with board order – label AAAA
0B6F	EE01	XOR	#01	; flip bit 0: select other board order
0B71	323262	LD	(#6232),A	; store board order
0B74	C39F0B	JP	#0B9F	; jump forward – jump to label EEEE
;				
0B77	3A3662	LD	A,(#6236)	; load A with scoring method – label BBBB
0B7A	EE01	XOR	#01	; flip bit 0: select other scoring method
0B7C	323662	LD	(#6232),A	; store scoring method
0B7F	C39F0B	JP	#0B9F	; jump forward – jump to label EEEE
;				
0B82	3A3562	LD	A,(#6235)	; load A with rounds max – label CCCC
0B85	FE01	CP	#01	; A == 1?
0B87	2002	JR	NZ,#0B42	; no, skip next step
0B89	3E02	LD	A,#02	; load A with 02
0B8B	3D	DEC	A	; decrement A
0B8C	323562	LD	(#6235),A	; store in rounds max
0B8F	C39F0B	JP	#0B9F	; jump forward – jump to label EEEE
;				
0B92	3A3562	LD	A,(#6235)	; load A with rounds max – label DDDD
0B95	3C	INC	A	; increment A
0B96	FE0A	CP	#0A	; A == 0A?
0B98	2002	JR	NZ,#0B2D	; no, skip next step
0B9A	3E09	LD	A,#09	; load A with 09

```

0B9C  323562      LD      (#6235),A      ; store in rounds max
;
0B9F  3E0A        LD      A,#0A          ; load A with #0A – label EEEE
0BA1  323060      LD      (#6030),A      ; set HSCursorDelay
0BA4  C9          RET                      ; return

```

De selectie gaat nu helemaal goed.

Nu versie v0.15 opgeslagen.

Wel start de selectie nog met 0 rondes. De default 0 voor board order en scoring method (US en BOARD) is prima. Maar liefste van te voren al de rounds max op 3 zetten als default. Mooie is wel dat deze variabelen niet aangepast worden en bij een volgend spel, staan ze nog hetzelfde ingesteld en kun je met jump meteen dezelfde settings opnieuw gebruiken. Daarnaast staat het level nu ook standaard op 0. Dat moet ook niet. Het level daarom meteen al op 05 zetten.

Waar deze initiële instellingen doen?

Vlak voordat het select screen gekozen wordt. Dus in principe op de plek waar de GameMode2 nu op 1 gezet wordt. Dat is nu hier:

```

0AA0  CD7408      CALL    #0874          ; clear screen and all sprites
0AA3  CD1C01      CALL    #011C          ; clear all sounds
;
0AA6  210A60      LD      HL,600A        ; load HL with GameMode2 address
0AA9  3601        LD      (HL),#01      ; set game mode 2 to 1
;
0AAB  C9          RET                      ; return

```

Dus daar iets in aanpassen en dan het level en de default round max initiëren:

Aanpassen:

```

0AA3  CDA50B      CALL    #0BA5          ; clear sounds and initialize variables

```

en

```

0BA5  CD1C01      CALL    #011C          ; clear all sounds
0BA8  3E05        LD      A,#05          ; load A with 05
0BAA  322962      LD      (#6229), A      ; set level to 05
0BAD  3E03        LD      A,#03          ; load A with 03
0BAF  323562      LD      (#6235),A      ; set rounds max to 03
0BB2  C9          RET                      ; return

```

Gaat goed. Level wordt niet meteen getoond. Pas bij de eerste refresh zie je dat. Misschien al eerder in het spel goed zetten? Maar waarom niet bij het tekenen van de L=00 niet de waarde van de #6299 pakken, maar gewoon altijd 05 als vaste waarde nemen?

Dat zit hier:

```

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

```

Aanpassen:

```

06E1 3E05 LD A,#05 ; load A with 05
06E3 00 NOP

```

Ja dat gaat goed.

Nu versie v0.16 opgeslagen.

Doordat de GameMode2 verder niet goed gezet wordt, gaat er het een en ander fout wanneer je nu de game start. Dat moet opnieuw uitgedacht worden. Eerst wellicht blokken die niet meer nodig zijn, vrijgeven, zodat er ruimte is om de nieuwe aansturing te bouwen.

Nadenken over de volgorde en bijbehorende GameMode2 settings. Na het selectiescreen staat de GameMode2 op 2.

Welke onderdelen hebben we niet meer nodig? Die eerst vrijgeven.

How High Can you Get: is niet meer nodig: vrijgeven #0BDA t/m #0C90. – Gedaan.

Player enters initials in High Score Table: niet meer nodig: vrijgeven #1486 t/m #1614. – Gedaan.

Eerst maar hiermee doen. Behoorlijk grote blokken vrij zo. De andere kleine blokken wellicht herbruikbaar (zoals bijvoorbeeld het switchen van player1 en player 2 en goedzetten scores, enz).

Zit nu een gaatje van #0BB3 t/m #0BCA. Op #0BCB staat een subroutine. Die verplaatsen naar #0BB3. De aanroep ernaar toe ook aangepast 2x. Gaat goed nog.

Spel bestaat uit X rondes, van X stappen, waarin P1 en P2 elk een beurt.

GameMode2 = 2: Initialiseren variabelen.

- Ronde en stap in ronde allebei op 0.
- Boardscores voor beide spelers op 0.
- Verder -> GameMode2 = 3.

GameMode2 = 3: Afhandelen ronde.

- Bepalen ronde, bepalen stap in ronde,
- Bepalen board (afhankelijk van stap in ronde en board order)
- Bepalen start speler (afhankelijk van ronde)
- Stap spelen: GameMode2 = 4
- Na laatste stap in laatste ronde -> GameMode2 = 10

GameMode2 = 4: Afhandelen stap in ronde.

- Resetten van de board score van beide spelers wanneer 1e beurt.
- Bepalen wie er als eerste aan de beurt is:
 - o Als player 1 aan de beurt -> GameMode2 = 5
 - o Als player 2 aan de beurt -> GameMode2 = 6
- Bepalen wie er als tweede aan de beurt is:
 - o Als player 1 aan de beurt -> GameMode2 = 5
 - o Als player 2 aan de beurt -> GameMode2 = 6
- Als beiden geweest zijn: scherm weergeven met resultaat -> GameMode 8

GameMode2 = 5: Afhandelen beurt van player 1.

- Goedzetten van het scherm voor P1 (flip back eventueel) en de controls.
- Weergeven informatie over stap (routine aanroepen vanuit 4 en 5)
 - o Ronde en stap informatie staat links bovenin (niet weer te geven)
 - o Weergeven 1 DK beneden in (van How High scherm)
 - o Weergeven interessante tekst (zoals How High)
 - o Muziek spelen van How High scherm
 - o Weergeven wie speelt. Rode Player I en Player II teksten gebruiken.
 - o Aangeven indrukken P1 button of P2 button (afhankelijk van wie er speelt).
- Wachten totdat de juiste knop ingedrukt wordt.
- Start individueel spel van een board -> GameMode2 = 7.

GameMode2 = 6: Afhandelen beurt van player 2.

- Goedzetten van het scherm voor P2 (flip eventueel) en de controls.
- Weergeven informatie over stap (routine aanroepen vanuit 4 en 5)
 - o Ronde en stap informatie staat links bovenin (niet weer te geven)
 - o Weergeven 1 DK beneden in (van How High scherm)
 - o Weergeven interessante tekst (zoals How High)
 - o Muziek spelen van How High scherm
 - o Weergeven wie speelt. Rode Player I en Player II teksten gebruiken.
 - o Aangeven indrukken P1 button of P2 button (afhankelijk van wie er speelt).
- Wachten totdat de juiste knop ingedrukt wordt.
- Start individueel spel van een board -> GameMode2 = 7.

GameMode2 = 7: Spelen van het daadwerkelijke board door een speler.

- Dit start de andere main.
- Hierbij zowel de total score als de board score van een speler bijhouden.
- Als daarin het level gefinished wordt of jumpman doodgaat -> GameMode2 = 9.

GameMode2 = 8: Weergeven resultaat van deze ronde.

- Goedzetten scherm voor P1 (flip back eventueel) en de controls.
- Weergeven wie er deze ronde gewonnen heeft op basis van board score van spelers.
- Wanneer scoring method Board de stand bijwerken.
- Dit scherm een aantal seconden laten staan
- Terug naar afhandeling ronde -> GameMode2 = 3.

GameMode2 = 9: Bijwerken hoogste board scores per speler.

- Kijken of de behaalde score voor deze speler/dit board hoger is dan de board HS.
- Zo ja dan de board HS bijwerken
- Terug naar de beurt afhandeling -> GameMode2 = 4.

GameMode2 = 10: Bepalen wie er gewonnen heeft:

- Wanneer scoring method Board: stand bekijken
- Wanneer scoring method Total: totaalstand bekijken
- Wanneer gelijk spel dan bepalen wie de hoogste barrel board score heeft.
- Updaten scores in high score table
- Weergeven winnaar
- Wachten totdat de jump button ingedrukt wordt
- En dan terug naar de buitenste loop waarin het high score scherm, de attract mode en het titel scherm getoond worden.

Volgorde starten: Als om de beurt dan bij US 1 speler alle barrel boards starten.

Daarom: in de oneven rondes start P1 en in de even rondes start P2.

Allemaal geheugenplaatsen reserveren om e.e.a. aan boekhouding te doen. Steeds als er iets bijkomt dan deze in het overzicht erbij opnemen.

Eerst de ronde en beurt logica maken, zonder dat het spel echt gespeeld wordt. Maar dan wel kijken of alle zaken op de juiste manier en volgorde gezet worden.

• Boardvolgorde:	#6232	0 = US, 1 = JP.
• Rondenummer Actueel:	#6233	1 t/m 9.
• Boardnummer Actueel:	#6234	1 t/m 6 (US) en 0 t/m 4 (JP).
• Rondenummer Max:	#6235	1 t/m 9.
• Scoringmethode:	#6236	0 = BOARDS, 1 = TOTAL
• Turnnummer Actueel:	#6237	1 = 1e game, 2 = 2e game
• BoardsWonP1	#6238	
• BoardsWonP2	#6239	
• ScreenTimer	#623A	
• PlayerTurnA	#600D	0 = P1, 1 = P2
• PlayerTurnB	#600E	0 = P1, 1 = P2
• TwoPlayerGame	#600F	Geen idee of nog benodigd is
• P1 HS Barrels:	#6240, #6241, #6242	
• P1 HS Pies:	#6243, #6244, #6245	
• P1 HS Springs:	#6246, #6247, #6248	
• P1 HS Rivets:	#6249, #624A, #624B	
• P1 HS Barrels:	#6250, #6251, #6252	
• P1 HS Pies:	#6253, #6254, #6255	
• P1 HS Springs:	#6256, #6257, #6258	
• P1 HS Rivets:	#6259, #625A, #625B	

GameMode2 = 2

en

```
OBC2  3E01      LD      A,#01      ; load A with 01
OBC4  323362    LD      (#6233),A    ; Set actual round to 01
OBC7  320F60    LD      (#600F),A    ; Set two player game
;
OBCE  3E00      LD      A,#00      ; load A with 00
OBCC  323462    LD      (#6234),A    ; Set actual board number to 00
;
OBCF  214062    LD      HL,#6240    ; load HL with start location for board HS
OBD2  061F      LD      B,#1F      ; For B = 1 to 1F
OBD4  77        LD      (HL),A      ; reset memory location
OBD5  23        INC      HL        ; next memory location
OBD6  10FC      DJNZ     #AAAA      ; next B
;
OBD8  210A60    LD      HL,600A     ; load HL with GameMode2 address
OBD8  3603      LD      (HL),#03    ; set game mode 2 to 3
;
OBDD  C9        RET                ; return
```

Gaat goed in de debugger.

GameMode2 = 3

```
0708  DE 0B      ; (3) Processing round
```

en:

```
OBDE  3A3462    LD      A,(#6234)    ; load A with actual board number
OBE1  3C        INC      A          ; find next board number
OBE2  47        LD      B,A          ; store A to B
;
OBE3  3A3262    LD      A,(#6232)    ; load A with board order
OBE6  FE00      CP      #00          ; A == 00 ?
OBE8  3E07      LD      A,#07        ; load A with #07: max boards US +1
OBEA  2802      JR      Z,OBE6       ; no, skip next step – jump to label AAAA
OBEC  3E05      LD      A,#05        ; load A with #05: max boards JP +1
OBE6  B8        CP      B           ; A == B ? – label AAAA
OBEF  2017      JR      NZ,OC08      ; no, valid board number – jump to label BBBB
;
OBF1  3A3362    LD      A,(#6233)    ; load A with actual round number
OBF4  47        LD      B,A          ; store A to B
OBF5  3A3562    LD      A,(#6235)    ; load A with round max
OBF8  B8        CP      B           ; A == B?
OBF9  2006      JR      NZ,OC01      ; no, not end of the game – jump to label CCCC
;
```

```

0BFB 210A60    LD    HL,600A    ; load HL with GameMode2 address
0BFE 360A      LD    (HL),#0A    ; set game mode 2 to 0A
0C00 C9        RET                    ; return
;

0C01 78        LD    A,B        ; restore A from B – label CCCC
0C02 3C        INC    A          ; find next round number
0C03 323362    LD    (#6233),A    ; store next round number
0C06 0601      LD    B,#01       ; reset board number
;
0C08 78        LD    A,B        ; restore A from B – label BBBB
0C09 323462    LD    (#6234),A    ; store next board number
;
0C0C C9        RET                    ; return

```

Nu eventjes met dit testen: steeds in dezelfde routine en kijken wat er dan gebeurt qua round en board afhandeling.

Dit gaat helemaal goed. De ronden en boarden worden goed doorlopen en aan het einde wordt de GameMode2 op 0A gezet. Helemaal top dus.

Nu versie v0.17 opgeslagen.

Hier nog code om het screen nr te zetten, de player scores te resetten (wanneer scoring methode is boards) en om de gamemode te verhogen aan toevoegen.

Het stukje met de screen mapping hergebruiken met eigen mapping:

```

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

```

```

3A65: 01 02 01 03 01 04 ; mapping US
3A6B: 01 02 03 04       ; mapping JP
3A6F t/m 3A79 00

```

en

```

0C0C 3A3262    LD    A,(#6232)    ; load A with board order
0C0F FE00      CP    #00          ; A == 00?
0C11 21653A    LD    HL,#3A65    ; load HL with table mapping US
0C14 2803      JR    Z,#AAAA      ; yes, skip next step
0C16 216B3A    LD    HL,#3A6B    ; load HL with table mapping JP
;
0C19 23        INC    HL          ; next position in table

```

```

0C1A  10FD      DJNZ  #0C19      ; next B
0C1C  2B        DEC   HL        ; correct index
;
0C1D  7E        LD    A,(HL)    ; load screen number from table
0C1E  322762    LD    (#6227),A ; set screen number
;
0C21  C9        RET                ; return

```

Ja dat gaat goed. Met debugger er doorheen en dan zie de de 6227 netjes het aantal ronden en de juiste volgorde de boards doorlopen.

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

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

```

0C21  3A3662    LD    A,#6236)  ; load A with scoring method
0C24  FE00      CP    #00        ; A == 00?
0C26  2009      JR    NZ,#0C31   ; no, skip next steps – jump to label AAAA
;
0C28  21B260    LD    HL,#60B2   ; load HL with start P1 en P2 scores
0C2B  0606      LD    B,#06      ; for B = 1 to 6
0C2D  3600      LD    (HL),#00   ; reset score byte
0C2F  23        INC   HL        ; next HL
0C30  10FB      DJNZ  #0C2D      ; next B
;
0C32  210A60    LD    HL,600A    ; load HL with GameMode2 address – label AAAA
0C35  3604      LD    (HL),#04   ; set game mode 2 to 04
0C37  C9        RET                ; return

```

Dit getest met debugger en gaat goed.

Nu versie v0.18 opgeslagen.

Nu GameMode2 = 4 even instellen als spelen spel.

```

0716  91 0C          ; (A) #0C91      ; clears screen, update timers, draws current screen, sets background music
0718  3C 12          ; (B) #123C      ; set initial mario sprite position and draw remaining lives and level
071A  7A 19          ; (C) #197A      ; for when playing a game. this is the main routine
071C  7C 12          ; (D) #127C      ; mario died. handle mario dying animations

```

Dat zijn meerdere GameMode2 settings. Misschien de eigen GameModes daar ook op aanpassen.

```

070A  91 0C          ; (4) clear screen and setup
070C  3C 12          ; (5) set sprite, lives and level
070E  7A 19          ; (6) play game

```

Als je afgaat, dan wordt de GameMode2 met één verhoogd. Dus dan GameMode2 = 7. Daar een regel maken dat de GameMode2 teruggezet wordt naar 3 (ronde afhandelen)


```
0710 38 0C                ; set GameMode2 back to 3
```

```
0C38 210A60      LD      HL,600A      ; load HL with GameMode2 address – label AAAA
0C35 3603        LD      (HL),#03      ; set game mode 2 to 03
0C37 C9          RET                  ; return
```

Gaat allemaal heel erg snel achter elkaar omdat er geen How High scherm tussen zit. Maar de rondes en de boardvolgorde klopt helemaal zoals het bedoeld is. Alleen de score wordt niet gereset als scoring methode board gekozen wordt. Dat nog eventjes checken.

De score wordt wel gewist. Alleen op het scherm blijft de oude score nog wel staan, totdat in de volgende game de eerste punten weer worden gescoord. Dus na het resetten van de score deze ook weergegeven op het scherm.

Updaten van de score kan zo:

```
0538 3A0D60      LD      A,(PlayerTurnA)      ; 0 for player 1, 1 for player 2
053B CD6B05      CALL    #056B                ; update onscreen score
```

Aanpassen:

```
0C32 3E00        LD      A,#00            ; select player 1
0C34 CD6B05      CALL    #056B            ; update onscreen score
0C37 3E01        LD      A,#01            ; select player 2
0C39 CD6B05      CALL    #056B            ; update onscreen score
;
0C3C 210A60      LD      HL,600A          ; load HL with GameMode2 address – label AAAA
0C3F 3604        LD      (HL),#04          ; set game mode 2 to 04
0C41 C9          RET                      ; return
```

En aanpassen extra GameMode2 = 7 die GameMode2 terugzet naar 3

```
0710 42 0C                ; set GameMode2 back to 3

0C42 210A60      LD      HL,600A          ; load HL with GameMode2 address – label AAAA
0C45 3603        LD      (HL),#03          ; set game mode 2 to 03
0C47 C9          RET                      ; return
```

Dit gaat niet goed. Op een ander punt aanroepen. Dus dat nog eventjes ook doen. En anders simpel een routine maken die 6 keer een 0 karakter op het scherm plaatst en deze met de score positie van P1 en daarna van P2 aanroepen.

Aanpassen:

```
0C32 3E00        LD      A,#00            ; select player 1
0C34 CDC605      CALL    #05C6            ; update onscreen score
0C37 3E01        LD      A,#01            ; select player 2
```

```

0C39  CDC605      CALL  #05C6      ; update onscreen score
;
0C3C  210A60      LD      HL,600A      ; load HL with GameMode2 address – label AAAA
0C3F  3604        LD      (HL),#04      ; set game mode 2 to 04
0C41  C9          RET              ; return

```

Ja gaat nu wel goed. Wat nog wel opvalt is dat de high score ook gaat meelopen. Dat moet straks niet meer. Dan daar de stand in punten weergeven in geval van scoring method board en anders niets. De high score heeft in dit spel geen functie meer.

Nu versie v0.19 opgeslagen.

Ook het beurtnummer hier op 0 zetten. Dan kan in de volgende GameMode2 deze verhoogt worden en dan kan de beurt bepaald worden.

Aanpassen:

```

0C3C  3E00        LD      A,#00        ; load A with 00
0C3E  323762      LD      (#6237),A    ; reset actual turn
;
0C41  210A60      LD      HL,600A      ; load HL with GameMode2 address – label AAAA
0C44  3604        LD      (HL),#04      ; set game mode 2 to 04
0C46  C9          RET              ; return

```

GameMode2 = 4

```

070A  47 0C          ; (4) Processing turns

```

en:

```

0C47  3A3362      LD      A,(#6233)    ; load A with actual round
0C4A  47          LD      B,A          ; save A to B
;
0C4B  3A3762      LD      A,(#6237)    ; load A with actual turn
0C4E  3C          INC      A          ; increment A
0C4F  323762      LD      (#6237),A    ; store updated actual turn
;
0C52  FE03        CP      #03        ; A == 03?
0C54  2006        JR      NZ,#0C58    ; no, determine turn player – jump to label AAAA
;
0C56  210A60      LD      HL,600A      ; load HL with GameMode2 address
0C59  3608        LD      (HL),#08      ; set game mode 2 to 08
0C5B  C9          RET              ; return after both players played this turn
;
0C5C  FE01        CP      #01        ; A == 01? – label AAAA
0C5E  2006        JR      NZ,#0C66    ; no, second turn – jump to label BBBB
;
0C60  CB40        BIT      0,B        ; is bit 0 of actual round set?

```

0C62	2008	JR	NZ,#0C6C	; yes, odd round, player 1 starts - CCCC
0C64	180C	JR	#0C72	; no, even round, player 2 starts - DDDD
;				
0C66	CB40	BIT	0,B	; is bit 0 of actual round set? – label BBBB
0C68	2802	JR	Z,#0C6C	; no, even round, player 1 plays second - CCCC
0C6A	1806	JR	#0C72	; yes, odd round, player 2 plays second - DDDD
;				
0C6C	210A60	LD	HL,600A	; load HL with GameMode2 address – label CCCC
0C6F	3605	LD	(HL),#05	; set game mode 2 to 05 (P1 plays)
0C71	C9	RET		; return,
;				
0C72	210A60	LD	HL,600A	; load HL with GameMode2 address – label DDDD
0C75	3606	LD	(HL),#06	; set game mode 2 to 06 (P2 plays)
0C77	C9	RET		; return after both players played this turn

En om te testen:

070C	78 0C			; (5) Player1
070E	7E 0C			; (6) Player2
...				
0712	84 0C			; (8) Processing turns
0C78	210A60	LD	HL,600A	; load HL with GameMode2 address
0C7B	3604	LD	(HL),#04	; set game mode 2 to 04
0C7D	C9	RET		; return after both players played this turn
0C7E	210A60	LD	HL,600A	; load HL with GameMode2 address
0C81	3604	LD	(HL),#04	; set game mode 2 to 04
0C83	C9	RET		; return after both players played this turn
0C84	210A60	LD	HL,600A	; load HL with GameMode2 address
0C87	3603	LD	(HL),#03	; set game mode 2 to 03
0C89	C9	RET		; return after both players played this turn

En dan debuggen met watchpoints op 0C7D en 0C89. Dan om en om gaan en bij rondewissel andersom om en om gaan.

Ja dat gaat nu goed. Netjes de beurten wisselen.

Nu versie v0.20 opgeslagen.

Nu zo ver dat de speler ingesteld wordt en eventueel het scherm geflipt moet worden. Dat gebeurt hier:

```

; arrive from #0701 when GameMode2 == 12
; flip screen if needed, reset game mode2 to zero, set player 2

13AA 3A2660 LD A,(UprightCab) ; load A with upright/cocktail
13AD 32827D LD (REG_FLIPSCREEN),A ; store into hardware screen flip
13B0 AF XOR A ; A := 0
13B1 320A60 LD (GameMode2),A ; set game mode2 to 0
13B4 210101 LD HL,#0101 ; HL := #0101
13B7 220D60 LD (PlayerTurnA),HL ; store 1 into PlayerTurnA (set player2) and PlayerTurnB (set player2)
13BA C9 RET ; return

; arrive from #0701 when GameMode2 == 13
; set player 1, reset game mode2 to zero, set screen flip to not flipped

13BB AF XOR A ; A := 0
13BC 320D60 LD (PlayerTurnA),A ; set for player 1
13BF 320E60 LD (PlayerTurnB),A ; store into current player number 1
13C2 320A60 LD (GameMode2),A ; set game mode2 to 0
13C5 3C INC A ; A := 1
13C6 32827D LD (REG_FLIPSCREEN),A ; store into screen flip for no flipping
13C9 C9 RET ; return

```

Voor player 2 wordt de REG_FLIPSCREEN ingesteld op basis van UprightCab en voor player 1 wordt de FLIP_SCREEN ingesteld op 1.

Voor player1 wordt de PlayerTurnA en de PlayerTurnB op 0 gezet.

Voor player2 wordt de PlayerTurnA en de PlayerTurnB op 1 gezet.

Waar worden deze PlayerTurnA en PlayerTurnB voor gebruikt?

O.a. voor player input in de interrupt routine, voor het knipperen van 1UP en 2UP, het updaten van de onscreen score, bij bepalen high score, bij invoeren credentials.

Maar dus voor een aantal zaken erg relevant. Dus die moeten we zetten.

Hoe wordt de score voor de juiste player vastgesteld? Score wordt gedaan in code vanaf #051C en daar wordt een routine aangeroepen die het score adres van de juiste speler teruggeeft in DE. En die routine werkt weer met PlayerTurnA om de juiste speler te bepalen.

GameMode2 = 5

070C 86 14 ; (5) Set Player1

en:

```

1486 3E00 LD A,#00 ; load A with 00
1488 320D60 LD (PlayerTurnA),A ; set for player 1
148B 320E60 LD (PlayerTurnB),A ; set for player 1
;
148E 3E01 LD A,#01 ; load A with 01
1490 32827D LD (REG_FLIPSCREEN),A ; set to no flipping
;
1493 CD7408 CALL #0874 ; clear screen and all sprites
1496 210A60 LD HL,600A ; load HL with GameMode2 address

```

1499	3607	LD	(HL),#07	; set game mode 2 to 07
149B	C9	RET		; return

GameMode2 = 6

070E 9C 14 ; (6) Set Player2

en:

149C	3E01	LD	A,#01	; load A with 01
149E	320D60	LD	(PlayerTurnA),A	; set for player 2
14A1	320E60	LD	(PlayerTurnB),A	; set for player 2
;				
14A4	3A2660	LD	A,(UpRightCab)	; load A with UprightCab
14A7	32827D	LD	(REG_FLIPSCREEN),A	; flipping depends on dipswitches
;				
14AA	CD7408	CALL	#0874	; clear screen and all sprites
14AD	210A60	LD	HL,600A	; load HL with GameMode2 address
14B0	3607	LD	(HL),#07	; set game mode 2 to 07
14B2	C9	RET		; return

Om te testen Gamemode2 naar Gamemode2 11 (=B) springen en dan 11,12 en 13 laten verwijzen naar de originele spel Gamemode2 code. Dan zou je moeten kunnen spelen, in de juiste volgorde en wisselen van speler. Ook testen of in cabaret mode het scherm gedraaid wordt.

0718	91 0C	; (B) clear screen and setup
071A	3C 12	; (C) set sprite, lives and level
071C	7A 19	; (D) play game

Als je afgaat, dan wordt de GameMode2 met één verhoogd. Dus dan GameMode2 = 14 (E). Daar een regel maken dat de GameMode2 teruggezet wordt naar 4 (beurt afhandelen)

071E B3 14 ; set GameMode2 back to 4

14B3	210A60	LD	HL,600A	; load HL with GameMode2 address – label AAAA
14B6	3604	LD	(HL),#04	; set game mode 2 to 04
14B8	C9	RET		; return

Gaat nog niet helemaal goed. De borden worden wel twee keer gespeeld. Maar 1-1 wordt gespeeld door P1, 1-2 wordt gespeeld door P2 en de rest ook: gaat niet terug naar P1. Debuggen of hij wel afwisselend door de code voor GameMode5 en GameMode6 gaat: watchpoints op #1498 en #14AC zetten. Blijkt een tikfout te zijn in het zetten van de PlayerTurnA.

Gaat helemaal goed. Goede volgorde schermen en speler die moet starten. En ook in de cocktail mode wordt het scherm van P2 omgedraaid. Helemaal top.

Nu versie v0.21 opgeslagen.

GameMode2 = 7

Nu het scherm gaan maken met informatie over wie er moet spelen en wachten totdat de correcte 'start' knop ingedrukt is.

De 'start' knoppen bij Upright: 'Player1' en 'Player2'.

De 'start' knoppen bij Cocktail: Jump P1 en Jump P2.

0710 B9 14 ; (7) Display player that must play and wait for start button

en:

```
14B9 3E00 LD A,#00 ; load A with 00
14BB 32867D LD (REG_PALETTE_A),A ; clear palette bank selector
14BE 32877D LD (REG_PALETTE_B),A ; clear palette bank selector
;
14C1 3A0D60 LD A,(PlayerTurnA) ; load A with player that plays
14B4 FE00 CP #00 ; Is player 1 playing?
14B6 110203 LD DE,#0302 ; load task data for text #2 "PLAYER <I>"
14C9 2803 JR Z,AAAA ; yes, skip next step
14CB 110303 LD DE,#0303 ; load task data for text #3 "PLAYER <II>"
;
14CE CD9F30 CALL #309F ; insert task to draw text
;
14D1 3A1060 LD A,(InputState) ; load A with input
14D4 CB7F BIT 7,A ; is jump button pressed?
14D6 2805 JR Z,AAA ; no, skip next steps – jump to label AAAA
;
14D8 210A60 LD HL,600A ; load HL with GameMode2 address
14DB 360B LD (HL),#0B ; set game mode 2 to 0B
;
14DD C9 RET ; return – label AAAA
```

En de teksten aanpassen:

```
3698: 94 76 20 1C 11 29 15 22 10 1F 1E 15 3F ; PLAYER ONE
36A5: 94 76 20 1C 11 29 15 22 10 24 27 1F 3F ; PLAYER TWO
```

De teksten staan wat laag. Daarom naar boven schuiven.

```
3698: 8E 76 20 1C 11 29 15 22 10 1F 1E 15 3F ; PLAYER ONE
36A5: 8E 76 20 1C 11 29 15 22 10 24 27 1F 3F ; PLAYER TWO
```

En dan een extra tekst toevoegen afhankelijk van eerste die speelt of tweede die speelt:

YOU PLAY FIRST of YOU PLAY NEXT

Tekst 10 hergebruiken voor de eerste en tekst 11 hergebruiken voor de tweede.

Aanpassen:

```
14D1  3A3762      LD      A,(#6237)    ; load A with actual turn number
14D4  FE01        CP      #01        ; Is this the first turn?
14D6  111003      LD      DE,#0310   ; load task data for text #10 "YOU PLAY ... FIRST"
14D9  2803        JR      Z,#AAAA    ; yes, skip next step
14DB  111103      LD      DE,#0311   ; load task data for text #11 "YOU PLAY ... NEXT"
;
14DE  CD9F30      CALL    #309F      ; insert task to draw text
;
14E1  3A1060      LD      A,(InputState) ; load A with input
14E4  CB7F        BIT      7,A        ; is jump button pressed?
14E6  2805        JR      Z,#AAAA    ; no, skip next steps – jump to label AAAA
;
14E8  210A60      LD      HL,600A     ; load HL with GameMode2 address
14EB  360B        LD      (HL),#0B    ; set game mode 2 to 0B
;
14ED  C9          RET                ; return – label AAAA
```

En de teksten aanpassen:

```
3747:  D1 76 29 1F 25 10 20 1C 11 29 10 16 19 22 23 24 3F    ; YOU PLAY FIRST
375D:  B1 76 29 1F 25 10 20 1C 11 29 10 1E 15 28 24 3F    ; YOU PLAY NEXT
```

En dan nu nog de tekst van de juiste knop indrukken en dan ook de juiste knop afvangen in de verschillende varianten.

Teksten:

- Als Upright dan als P1: PRESS 1-PLAYER – hergebruiken #0D (Name Registration)
- Als Upright dan als P2: PRESS 2-PLAYER – hergebruiken #12 ("U" through "Z")
- Als Cocktail dan: PRESS JUMP – hergebruiken #13 (Regi Time).

Aanpassen:

```
14E1  3A2660      LD      A,(UprightCab) ; load A with Upright or Cocktail
14E4  FE00        CP      #00        ; Playing on CockTail?
14E6  110D03      LD      DE,#030D   ; load task data for text #0D "PRESS JUMP"
14E9  280D        JR      Z,#14F8    ; yes, skip next steps – jump to label BBBB
;
14EB  3A0D60      LD      A,(PlayerTurnA) ; load A with player that plays
14EE  FE00        CP      #00        ; Is player 1 playing?
14F0  111203      LD      DE,#0312   ; load task data for text #12 "PRESS P-1"
14F3  2803        JR      Z,#14F8    ; yes, skip next step
14F5  111303      LD      DE,#0313   ; load task data for text #13 "PRESS P-2"
;
14F8  CD9F30      CALL    #309F      ; insert task to draw text – label BBBB
```

```

;
14FB 3A1060 LD A,(InputState) ; load A with input
14FE CB7F BIT 7,A ; is jump button pressed?
1500 2805 JR Z,#AAAA ; no, skip next steps – jump to label AAAA
;
1502 210A60 LD HL,600A ; load HL with GameMode2 address
1505 360B LD (HL),#0B ; set game mode 2 to 0B
;
1507 C9 RET ; return – label AAAA

```

Uprightcab 1 = upright / 0 = cocktail

En dan de teksten aanpassen:

```

371C: F2 76 22 20 22 15 23 23 10 1A 25 1D 20 3F ; PRESS JUMP
3773: F2 76 20 22 15 23 23 10 01 2C 20 1C 11 29 15 22 3F ; PRESS 1-PLAYER
378B: F2 76 20 22 15 23 23 10 02 2C 20 1C 11 29 15 22 3F ; PRESS 2-PLAYER

```

Dat gaat nu goed.

Ook de tekst eerder aanpassen naar 2-PLAYER ipv 2 PLAYERS:

0A: PUSH 2 PLAYERS BUTTON wordt PUSH 2-PLAYER BUTTON

```

36FD: 2A 77 20 25 23 18 10 02 2C 20 1C 11 29 15 22 23 10 12 25 24 24 1F 1E 10 3F

```

Dit is nu allemaal goed.

Nu versie v0.22 opgeslagen.

Nu de knoppen op de juiste wijze afvangen. Als Cocktail dan jump button zoals het nu al is. Als Upright dan checken 1-Player of de 2-Player button.

Aanpassen:

```

14FB 3A2660 LD A,(UprightCab) ; load A with Upright or Cocktail
14FE FE00 CP #00 ; Playing on CockTail?
1500 2817 JR Z,#1519 ; yes, skip next steps – jump to label AAAA
;
1502 3A0D60 LD A,(PlayerTurnA); load A with player that plays
1505 FE00 CP #00 ; Is player 1 playing?
1507 2008 JR NZ,#1511 ; no, skip next steps – jump to label BBBB
;
1509 3A007D LD A,(IN2) ; load A with IN2
150C FE04 CP #04 ; is P1 button pressed?
150E C0 RET NZ ; no, return – wait until pressed
150F 180F JR #1520 ; skip next steps
;

```



```

1511 3A007D LD A,(IN2) ; load A with IN2 – label BBBB
1514 FE08 CP #08 ; is P2 button pressed?
1516 C0 RET NZ ; no, return – wait until pressed
1517 1807 JR #1520 ; skip next steps
;
1519 3A1060 LD A,(InputState) ; load A with input – label AAAA
151C CB7F BIT 7,A ; is jump button pressed?
151E 2805 RET Z, ; no, return - wait until pressed
;
1520 210A60 LD HL,600A ; load HL with GameMode2 address - CCCC
1523 3608 LD (HL),#08 ; set game mode 2 to 08
;
1525 C9 RET ; return

```

Ja dat gaat goed.

Nu versie v0.23 opgeslagen.

GameMode2 = 8

How High Can You Get scherm maken. Hadden we al weggehaald, dus deels weer herstellen. Stukje eenvoudiger door de m aanduiding weg te laten en het aantal Goofy Kongs te laten afhangen van het board dat gespeeld gaat worden.

GameMode2 = 8 wordt al gebruikt voor het processen van de turns:

```

0C56 210A60 LD HL,600A ; load HL with GameMode2 address
0C59 3608 LD (HL),#08 ; set game mode 2 to 08
0C5B C9 RET ; return after both players played this turn

```

Dit dan aanpassen naar GameMode2=9.

```

0C59 3609 LD (HL),#09 ; set game mode 2 to 09

```

en

```

0714 84 0C ; (9) Processing turns

```

How High scherm in eenvoudige vorm maken, zonder de Goofy Kongs te tekenen.

```

0712 26 15 ; (8) Display How High screen

```

en:

```

1526 3A3A62 LD A,(#623A) ; load A with screen timer
1529 3C INC A ; increment screen timer
152A 323A62 LD (#623A),A ; store new screen timer
;

```

```

152D FE01      CP      #01      ; is screen timer 01?
152F CA3815    JP      Z,#1538  ; yes, jump to label XXXX
;
1532 FEA0      CP      #A0      ; is screen timer A0?
1534 CA5515    JP      Z,1555    ; yes, jump to label YYYY
;
1537 C9        RET              ; return
;
1538 CD7408     CALL     #0874    ; clear the screen and sprites – label XXXX
;
153B 21867D     LD      HL,REG_PALETTE_A ; load HL with palette bank
153E 3601      LD      (HL),#01   ; set palette bank selector
1540 23        INC      HL        ; next palette bank
1541 3600      LD      (HL),#00   ; clear palette bank selector
;
1543 218A60     LD      HL,#608A  ; load HL with tune address
1546 3602      LD      (HL),#02   ; play how high can you get sound?
1548 23        INC      HL        ; HL := #608B . load HL with music timer ?
1549 3603      LD      (HL),#03   ; set to 3 units
;
154B 110703     LD      DE,#0307  ; load task data for text #7 "HOW HIGH"
154E CD9F30     CALL     #309F    ; insert task to draw text
;
1551 000000     3 NOP's for call to draw goofy kongs
;
1554 C9        RET              ; return
;
1555 210A60     LD      HL,600A    ; load HL with GameMode2 address – label YYYY
1558 360B      LD      (HL),#0B   ; set game mode 2 to 0B
155A 3E00      LD      A,#00      ; load A with 00
155C 323A62     LD      (#623A),A ; reset screen timer
;
155F C9        RET              ; return

```

Dat gaat goed. Nu nog maken dat er een aantal Goofy Kongs getekend worden.

Aanpassen:

```

1551 CD6015     CALL     #1560    ; call routine that draws goofy kongs
;
en:
1560 3A3462     LD      A,(#6234) ; load A with number of goofys to draw = board #
1563 47        LD      B,A        ; copy to B for use as loop counter
1564 21BC75     LD      HL,#75BC  ; load HL with screen location start for goofy kong
;
1567 0E50      LD      C,#50      ; C := #50 = start graphic for goofy kong – label BBBB
;

```

1569	71	LD	(HL),C	; draw part of goofy kong – label CCCC
156A	0C	INC	C	; next graphic
156B	2B	DEC	HL	; next screen location
156C	71	LD	(HL),C	; draw part of goofy kong
156D	0C	INC	C	; next graphic
156E	2B	DEC	HL	; next screen location
156F	71	LD	(HL),C	; draw part of goofy kong
1570	0C	INC	C	; next graphic
1571	2B	DEC	HL	; next screen location
1572	71	LD	(HL),C	; draw part of goofy kong
1573	79	LD	A,C	; load A with graphic number
1574	FE67	CP	#67	; == #67 ? (are we done?)
1576	CA8115	JP	Z,#1581	; yes, skip next 4 steps – jump to label AAAA
;				
1579	0C	INC	C	; next C
157A	112300	LD	DE,#0023	; load DE with offset
157D	19	ADD	HL,DE	; add to screen location
157E	C36915	JP	#1569	; loop again – jump to label CCCC
;				
1581	115FFF	LD	DE,#FF5F	; load DE with offset for goofy
1584	19	ADD	HL,DE	; add offset to draw next goofy
;				
1585	05	DEC	B	; decrease B. done drawing all? – label AAAA
1586	C26715	JP	NZ,#1567	; no, loop and do another – jump to label BBBB
;				
1589	C9	RET		; return

Ja dat gaat goed nu.

Nu versie v0.24 opgeslagen.

Eerst een aantal zaken nu goed maken.

De weergave van de actual rounds wordt niet snel genoeg geupdate in de tussenschermen.

De High score wordt nog getoond – aanpassen naar tonen stand in scoring methode board.

Er zit een timer tussen de schermen soms. Waar komt dat vandaan? En dan verhelpen.

Zorgen voor juiste afhandeling death: animation.

Zorgen voor juiste afhandeling board finish: animation.

Lijkt erop dat de mario dead de volgende GameMode2 is na de drie die wel al toegevoegd hebben:

0718	91 0C	; (B) clear screen and setup
071A	3C 12	; (C) set sprite, lives and level
071C	7A 19	; (D) play game
071E	7C 12	; (E) mario died. handle mario dying animations
0720	B3 14	; (F) set GameMode2 back to 4

Ja dat werkt. Nu wordt de jumpman died animatie uitgevoerd.

Nu kijken hoe de end-of-level afgehandeld wordt. Lijkt oorspronkelijk GameMode2 = 16 te zijn.

Even kijken wat er gebeurt als we die terugzetten:

```
072E  15 16          ; (16) handle end of level animations
```

Ja dat spelt de en of level animaties (in ieder geval voor barrels). Daarna gaat het niet goed. Dit zet ook weer een GameMode2. Achterhalen welke dat is.

Op #17B3 en #1968 wordt de GameMode2 op 8 gezet. Dat is niet goed. Moet (net als bij af gaan) teruggezet worden naar GameMode2 = 4.

```
17B2  23          INC    HL          ; HL := GameMode2
17B3  3604        LD     (HL),#04    ; set game mode2 to 4
17B5  C9          RET                ; return
```

en

```
1967  23          INC    HL          ; increase HL to GameMode2
1968  3604        LD     (HL),#04    ; set game mode2 to 4
196A  C9          RET                ; return
```

Gaat niet goed. Speelt end of level animaties van P1 barrels en schakelt dan naar P2. Maar als je die game dan start dan verschijnt heel snel de rivets en dan is het spel voorbij. Zou kunnen zijn dat de logica om het volgende veld te bepalen in de weg zit.

Wij bepalen het veld in de afhandeling van GameMode2 = 3. Maar daar komen we niet terug. Maar de code van de game bepaalt wel het volgende veld. Maar dat gaat niet goed, omdat de pointer niet goed staat en de tabel met boards/levels is aangepast. Dus dat moet uit de bestaande code gehaald worden.

Dat gebeurt hier:

; kong has climbed off the screen at end of level

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

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

194B 222A62 LD (#622A), HL ; store
194E 322762 LD (#6227), A ; store A into screen number
1951 212962 LD HL, #6229 ; load HL with level number address
1954 34 INC (HL) ; increase #6229 by one
1955 110005 LD DE, #0500 ; load task #5, parameter 0 ; adds bonus to player's score
1958 CD9F30 CALL #309F ; insert task
195B AF XOR A ; A := 0
195C 322E62 LD (#622E), A ; store into number of goofys to draw
195F 328863 LD (#6388), A ; store into end of level counter
1962 210960 LD HL, WaitTimerMSB ; load HL with timer
1965 36E0 LD (HL), #e0 ; set timer to #E0
1967 23 INC HL ; increase HL to GameMode2
1968 3608 LD (HL), #08 ; set game mode2 to 8
196A C9 RET ; return
```

Hierin aanpassen:

- De code van #193D t/m #1954: wissen: NOP's maken.
- De code van #1955 t/m #195A: behouden: voegt de bonus tijd aan de score toe.
- De code van #195B t/m #1961: wissen: NOP's maken.
- De code van #1962 t/m #196A: behouden: set timer en set de GameMode2.

Nog zo'n blok:

; jump here from #1622 when girders or elevators is finished. step 6 of 6
; jump here from #1622 when conveyors is finished. step 5 of 5

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

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

179D 222A62 LD (#622A), HL ; store screen address lookup for next time
17A0 322762 LD (#6227), A ; store A into screen number
17A3 110005 LD DE, #0500 ; load task #5, parameter 0 ; adds bonus to player's score
17A6 CD9F30 CALL #309F ; insert task
17A9 AF XOR A ; A := 0
17AA 328863 LD (#6388), A ; clear end of level counter
17AD 210960 LD HL, WaitTimerMSB ; load HL with timer addr.
17B0 3630 LD (HL), #30 ; set timer to #30
17B2 23 INC HL ; HL := GameMode2
17B3 3608 LD (HL), #08 ; set game mode2 to 8
17B5 C9 RET ; return
```

Hierin aanpassen:

- De code van #178E: behouden: afhandeling timer.
- De code van #178F t/m #17A2: wissen: NOP's maken.
- De code van #17A3 t/m #17A8: behouden: voegt de bonus tijd aan de score toe.
- De code van #17A9 t/m #17AC: wissen: NOP's maken.
- De code van #17AD t/m #17B5: behouden: set timer en set de GameMode2.

Ja dat gaat nu allemaal goed. Ook de bonus timer wordt goed aan de score toegevoegd.

Alleen de timing bij rivets is niet goed. Scherm verdwijnt al terwijl het muziekje nog klinkt. Dus dat meenemen bij het onderzoeken van de andere ongewenste timers.

Nu versie v0.25 opgeslagen.

Lijkt erop dat het nog niet helemaal goed gaat wanneer jumpman dood gaat. In ieder geval als Player 2 doodgaat gaat het niet goed. Dan na de animatie meteen naar het titelscherm.

Eventjes met debugger kijken waar dat gebeurt. Watchpoint op GameMode2 (#600A) schrijven zetten. En dan kijken wat er gezet wordt wanneer Player 2 doodgaat (en waar in de code).

Dat gebeurt dus hier:

```
; mario is completely dead
```

```
12DE DF      RST      #18          ; count down WaitTimerMSB and only continue when 0
12DF CDB30   CALL     #30DB        ; clear mario and elevator sprites from screen
12E2 210A60  LD       HL,GameMode2 ; set HL to game mode2
12E5 3A0E60  LD       A,(PlayerTurnB) ; load A with current player
12E8 A7      AND      A           ; is this player 1 ?
12E9 CAED12  JP       Z,#12ED      ; yes, skip next step

12EC 34      INC      (HL)        ; increase game mode

12ED 34      INC      (HL)        ; increase game mode
12EE 2B      DEC      HL          ; load HL with WaitTimerMSB
12EF 3601    LD       (HL),#01    ; store 1 into timer
12F1 C9      RET
```

Voor Player 1 wordt de GameMode2 met 1 verhoogd en voor Player 2 wordt de GameMode2 met 2 verhoogd. Dat hoeft nu niet.

Aanpassen:

Stuk code van #12E5 t/m #12EC NOP's maken.

Gaat nu goed.

Nu versie v0.26 opgeslagen.

Naast een probleem met de timing is er ook een probleem met geluiden die blijven spelen. Bijvoorbeeld wanneer Bonus timer afloopt, blijft dat geluid doorspelen in het scherm waarin gewacht wordt op starten van de volgende beurt.

Kijken of de high score verwijderd kan worden. En dan later vervangen door stukje code waarin bij scoring methode boards de stand weergegeven wordt.

Wordt in ieder geval hier gedaan:

```
055C C3DA05 JP #05DA ; skip ahead to update high score onscreen
```

Aanpassen door op #055C 3x NOP te maken.

De high score wordt nu niet meer getoond. Wel nog bijgewerkt, maar dat weghalen beïnvloed wellicht de timing van de game te veel. Dus laat maar staan voorlopig.

Nu versie v0.27 opgeslagen.

Stukje code op de stand weer te geven op de plek van de high score (screen position #7641):

#6236 = scoring methode, #6238 = boards won P1 en #6239 = boards won P2

De scores in #6238 en #6239 worden hexadecimaal opgeslagen. Maar decimaal weergeven. Hoe omzetten? In ieder geval te doen door een aantal keren #0A ervan af te trekken: elke keer dat dat lukt is tientallen met 1 verhogen. Als het niet meer gaat, is wat je overhad de eenheden. Daar een routine voor maken en dan twee keer aanroepen.

Dit is het stukje code uit BarrelPalooza om het progress number op het scherm te tekenen:

```
27C7: 3A2E62 LD A,(#622E) ; load A with progress number – label BBBB
27CA: 010AFF LD BC,#FF0A ; B: = #FF, C := #0A (10 decimal)
;
27CD: 04 INC B ; increment B – label CCCC
27CE: 91 SUB C ; subtract 10 decimal
27CF: D2CD27 JP NC,#27CD ; not carry, loop again – jump to label CCCC
;
27D2: 81 ADD A,C ; add 10 back to A to get a number from 0 to 9
```

Hierna heeft B de tientallen en heeft A de eenheden.

Aanpassen:

```
158A 214176 LD HL,#7641 ; load HL with screen position for score
158D 11E0FF LD DE,#FFE0 ; load DE with offset for column
;
```

1590	3A3662	LD	A,(#6236)	; load A with scoring method
1593	FE00	CP	00	; is scoring method BOARDS
1595	2808	JR	Z,#159F	; yes, skip next steps – jump to label AAAA
;				
1597	0606	LD	B,#06	; for B = 1 to 6
1599	3610	LD	(HL),#10	; clear screen position
159B	19	ADD	HL,DE	; add offset for next column
159C	10FB	DJNZ	#1599	
159E	C9	RET		; return
;				
159F	3A3862	LD	A,(#6238)	; load A with boards won P1 – label AAAA
15A2	CDB915	CALL	15B9	; call routine to convert to decimal
15A5	70	LD	(HL),B	; display tens of P1
15A6	19	ADD	HL,DE	; add offset for next column
15A7	77	LD	(HL),A	; display units of P1
;				
15A8	19	ADD	HL,DE	; add offset for next column
15A9	362C	LD	(HL),#2C	; display “-“
15AB	19	ADD	HL,DE	; add offset for next column
15AC	362C	LD	(HL),#2C	; display “-“
15AE	19	ADD	HL,DE	; add offset for next column
;				
15AF	3A3962	LD	A,(#6239)	; load A with boards won P2
15B2	CDB915	CALL	15B9	; call routine to convert to decimal
15B5	70	LD	(HL),B	; display tens of P2
15B6	19	ADD	HL,DE	; add offset for next column
15B7	77	LD	(HL),A	; display units of P2
;				
15B8	C9	RET		; return

en:

15B9	010AFF	LD	BC,#FF0A	; B: = #FF, C := #0A (10 decimal)
;				
15BC	04	INC	B	; increment B – label BBBB
15BD	91	SUB	C	; subtract 10 decimal
15BE	30FC	JR	NC,#15BC	; not carry, loop again – jump to label BBBB
;				
15C0	81	ADD	A,C	; add 10 back to A to get a number from 0 to 9
;				
15C1	C9	RET		; return

En het hele gebeuren ook nog ergens aanroepen: eerst maar eens op de plek waar de high score eerst bijgewerkt werd.

055C	CD8A15	CALL	#158A	; update board score
------	--------	------	-------	----------------------

Dit gaat goed. De aanroep daar is niet echt nodig. Maar laat voorlopig maar eventjes staan.

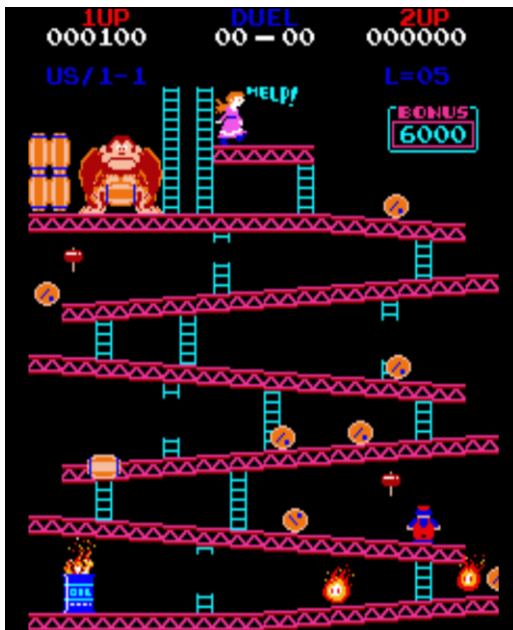
Nu versie v0.28 opgeslagen.

De board score is nu nog met twee mintekens. Nieuwe graphics maken met een linker en rechter deel van een langere “-“. Linkerdeel op #BC en rechterdeel op “BD”.

Aanpassen:

15A8	19	ADD	HL,DE	; add offset for next column
15A9	36BC	LD	(HL),#BC	; display left part of “-“
15AB	19	ADD	HL,DE	; add offset for next column
15AC	36BD	LD	(HL),#BD	; display right part of “-“
15AE	19	ADD	HL,DE	; add offset for next column

Ja dat ziet er goed uit.



Nu versie v0.29 opgeslagen.

Eerst eens kijken of we wat extra ruimte kunnen vrijmaken.

Ja dat kan door allerlei game over afhandelingen te verwijderen. Daardoor #12F2 t/m 1485 ook vrijgekomen.

Nu nadenken over het bijhouden van de score wanneer scoring methode board.

0C56	210A60	LD	HL,600A	; load HL with GameMode2 address
0C59	3609	LD	(HL),#09	; set game mode 2 to 09
0C5B	C9	RET		; return after both players played this turn

De GameMode2 = 9 wordt dus uitgevoerd wanneer beide players aan de beurt zijn geweest.

GameMode2=9 wijst naar code op #0C84. Daar wordt de GameMode2 weer teruggezet naar 03: het processen van de ronde.

Uiteindelijk zo maken:

- GameMode2 = 12 (0720) zorgt voor het afhandelen van de board statistics -> GameMode2 = 10.
- GameMode2 = 10 (0722) checkt scoring methode BOARDS
 - zo ja, dan -> GameMode2 = 11.
 - Zo nee, dan -> GameMode2 = 4.
- GameMode2 = 11 (0724) bepaald board winnaar en past score aan -> GameMode2 = 4.

Op #0C59 de GameMode2 op 12 zetten. Dat start de board statistics.

En vooralsnog voor GameMode2 = F een stukje tijdelijke code maken die de GameMode2 op 10 zet. Waardoor de board statistics daar later aan toegevoegd kunnen worden.

GameMode2 = 12

0726 E7 15 ; (F) reserved for board statistics, now advance to GameMode2=10

en

15E7	210A60	LD	HL,600A	; load HL with GameMode2 address
15EA	3610	LD	(HL),#10	; set game mode 2 to 10
15EC	C9	RET		; return after both players played this turn

GameMode2 = 10

0722 C2 15 ; (10) check if board method BOARDS and set GameMode2

en:

15C2	3A3662	LD	A,(#6236)	; load A with scoring method
15C5	FE00	CP	#00	; is scoring method BOARDS?
15C7	3E11	LD	A,#11	; load A with #11
15C9	2802	JR	Z,#15CD	; yes skip next step
;				
15CB	3E03	LD	A,#03	; load A with #03
;				
15CD	210A60	LD	HL,600A	; load HL with GameMode2 address
15D0	77	LD	(HL),A	; set game mode 2
15D1	C9	RET		; return

en:

GameMode2 = 11

0724 F2 12 ; (11) determine board winner and update board score

en:

12F2	3A3A62	LD	A,(#623A)	; load A with screen timer
12F5	3C	INC	A	; increment screen timer
12F6	323A62	LD	(#623A),A	; store new screen timer
;				
12F9	FE01	CP	#01	; is screen timer 01? – first time
12FB	CA1113	JP	Z,#1311	; yes, jump to label XXXX
;				
12FE	FEB0	CP	#B0	; is screen timer B0 – last time
1300	CACD13	JP	Z,13CD	; yes, jump to label YYYY
;				
1303	214176	LD	HL,#7641	; load HL with screen position for score
1306	11E0FF	LD	DE,#FFE0	; load DE with offset for column
1309	CB67	BIT	4,A	; check bit 3 of screen timer
130B	C29715	JP	NZ,#1597	; bit 3 is not set – clear score from screen
130E	C39F15	JP	#159F	; bit 3 is set – display score on screen
;				
1311	CD7408	CALL	#0874	; clear the screen and sprites – label XXXX
;				
1314	21867D	LD	HL,REG_PALETTE_A	; load HL with palette bank
1317	3600	LD	(HL),#00	; set palette bank selector
1319	23	INC	HL	; next palette bank
131A	3600	LD	(HL),#00	; clear palette bank selector
;				
; Check for winning score				
;				
131C	11B460	LD	DE,#60B4	; load DE with lowest byte of P1 score
131F	21B760	LD	HL,#60B7	; load HL with lowest byte of P2 score
1322	CDD215	CALL	#15D2	; compare scores – call routine ZZZZ
;				
1325	FE01	CP	#01	; Did P1 win?
1327	2813	JR	Z,#133C	; yes, jump to label BBBB
;				
1329	FE02	CP	#02	; Did P2 win?
132B	2820	JR	Z,#134D	; yes, jump to label CCCC
;				
; P1 and P2 scores are equal				
;				
132D	110603	LD	DE,#0306	; load task data for text #06 "HAS WON ..."
1330	CD9F30	CALL	#309F	; insert task to draw text
1333	213862	LD	HL,#6238	; load HL with BoardsWonP1
1336	34	INC	(HL)	; increase BoardsWonP1
1337	213962	LD	HL,#6239	; load HL with BoardsWonP2

133A	34	INC	(HL)	; increase BoardsWonP2
133B	C9	RET		; return
;				
; P1 has highest score				
;				
133C	110203	LD	DE,#0302	; load task data for text #2 "PLAYER 1" - label BBBB
133F	CD9F30	CALL	#309F	; insert task to draw text
1342	111A03	LD	DE,#031A	; load task data for text #1A "WON THIS BOARD"
1345	CD9F30	CALL	#309F	; insert task to draw text
;				
1348	213862	LD	HL,#6238	; load HL with BoardsWonP1
134B	34	INC	(HL)	; increase BoardsWonP1
134C	C9	RET		; return
;				
; P2 has highest score				
;				
134D	110303	LD	DE,#0303	; load task data for text #3 "PLAYER 2" – label CCCC
1350	CD9F30	CALL	#309F	; insert task to draw text
1353	111A03	LD	DE,#031A	; load task data for text #1A "WON THIS BOARD"
1356	CD9F30	CALL	#309F	; insert task to draw text
;				
1359	213962	LD	HL,#6239	; load HL with BoardsWonP2
135C	34	INC	(HL)	; increase BoardsWonP2
135D	C9	RET		; return
;				
135E	210A60	LD	HL,600A	; load HL with GameMode2 address – label YYYY
1361	3603	LD	(HL),#03	; set game mode 2 to 03
1363	3E00	LD	A,#00	; load A with 00
1365	323A62	LD	(#623A),A	; reset screen timer
;				
1358	C9	RET		; return

De volgende procedure wordt gebruikt voor het vaststellen welke score hoger is (of eventueel even hoog). De twee scores als input in DE en HL. A geeft het resultaat terug: 0 = scores zijn gelijk, 1 = score in DE is hoger, 2 = score in HL is hoger.

15D2	0603	LD	B,#03	; for B = 1 to 3 – label ZZZZ
15D4	1A	LD	A,(DE)	; load A with DE score byte
15D5	BE	CP	(HL)	; compare A with HL score byte
15D6	3809	JR	C,#15E1	; jump to HL higher -jump to label AAAA
15D8	200A	JR	NZ,#15E4	; jump to DE higher – jump to label BBBB
15DA	1B	DEC	DE	; get next byte of P1 score
15DB	2B	DEC	HL	; get next byte of HL score
15DC	10F6	DJNZ	#BBBB	; next B
;				
15DE	3E00	LD	A,#00	; scores are equal
15E0	C9	RET		; return
;				

15E1	3E02	LD	A,#02	; score in HL is higher – label	AAAA
15E3	C9	RET		; return	
;					
15E4	3E01	LD	A,#01	; score in DE is higher – label	BBBB
15E6	C9	RET		; return	

Dit is een aparte procedure omdat dit vergelijken later ook nog vaker nodig is om de board statistics bij te werken.

Nu versie v0.30 opgeslagen.

Nu nog de teksten aanpassen.

Tekst 1A aanpassen naar "WON THIS BOARD"

37B6: D2 76 27 1F 1E 10 24 18 19 23 10 12 1F 11 22 14 3F

En vrije tekst #6 gebruiken voor "EQUAL SCORES"

3657: DD 3F ; tekst 06 "EQUAL SCORES"

3FDD: B2 76 15 21 25 11 1C 10 23 13 1F 22 15 23 3F

Ja dat is nu goed.

Nu versie v0.31 opgeslagen.

Nu de eindafhandeling maken.

Dus bepalen wie er uiteindelijk gewonnen heeft. Bij scoring method BOARDS kijken naar de boards won en bij scoring method TOTAL kijken naar de scores. Wanneer gelijkspel dan kijken naar de hoogste score voor barrels board. Maar dat hebben we nog niet, dus eerst nog een stub daarvoor gebruiken. Dat was het idee, maar niet voldoende ruimte om dat nog allemaal toe te voegen. Dus volstaan met melding gelijkspel.

En boodschap plaatsen: PLAYER ONE en WON THIS GAME.

En dan eigenlijk kijken of het overzicht getoond kan worden. Daar ook de uitslag tonen.

Eerst even kijken welke GameMode2 dat wordt.

Dat wordt hier gezet:

0BF1	3A3362	LD	A,(#6233)	; load A with actual round number
0BF4	47	LD	B,A	; store A to B
0BF5	3A3562	LD	A,(#6235)	; load A with round max
0BF8	B8	CP	B	; A == B?
0BF9	2006	JR	NZ,0C01	; no, not end of the game – jump to label
;				

0BFB	210A60	LD	HL,600A	; load HL with GameMode2 address
0BFE	360A	LD	(HL),#0A	; set game mode 2 to 0A
0C00	C9	RET		; return

Maar 0A wordt al gebruikt. Dus daar een andere voor gaan nemen: GameMode2 = 14.

Dus aanpassen:

0BFE	3614	LD	(HL),#14	; set game mode 2 to 14
------	------	----	----------	-------------------------

GameMode2 = 14

072A	69 13			; (14) determine board winner and update board score
------	-------	--	--	--

en:

1369	3A3A62	LD	A,(#623A)	; load A with screen timer
136C	3C	INC	A	; increment screen timer
136D	323A62	LD	(#623A),A	; store new screen timer
;				
1370	FE01	CP	#01	; is screen timer 01? – first time
1372	CA7B13	JP	Z,#137B	; yes, jump to label XXXX
;				
1375	FEB0	CP	#B0	; is screen timer B0 – last time
1377	CACD13	JP	Z,13CD	; yes, jump to label YYYY
;				
137A	C9	RET		; return
;				
137B	CD7408	CALL	#0874	; clear the screen and sprites – label XXXX
;				
137E	21867D	LD	HL,REG_PALETTE_A	; load HL with palette bank
1381	3600	LD	(HL),#00	; set palette bank selector
1383	23	INC	HL	; next palette bank
1384	3600	LD	(HL),#00	; clear palette bank selector
;				
; Check for winning score				
;				
1386	3A3662	LD	A,(#6236)	; load A with scoring method
1389	FE01	CP	#01	; is scoring method TOTAL?
138B	280E	JR	Z,139B	; yes, jump to label AAAA
;				
; Check for winning score when scoring method BOARDS				
;				
138D	3A3862	LD	A,(#6238)	; load A with BoardsWonP1
1390	47	LD	B,A	; store BoardsWonP1 in B
1391	3A3962	LD	A,(#6239)	; load A with BoardsWonP2
1394	B8	CP	B	; is A > B?
1395	381C	JR	C,#13B3	; yes, P1 won, jump to label BBBB

```

1397  2027      JR    NZ,#13C0      ; no A < B, P2 won, jump to label CCCC
1399  1811      JR    #13AC        ; A = B, equal score, jump to label DDDD
;
; Check for winning score when scoring method TOTAL
;
139B  11B460    LD    DE,#60B4      ; load DE with lowest byte of P1 score- label AAAA
139E  21B760    LD    HL,#60B7      ; load HL with lowest byte of P2 score
13A1  CDD215    CALL  #15D2        ; compare scores – call routine ZZZZ
;
13A4  FE01      CP    #01          ; Did P1 win?
13A6  280B      JR    Z,#13B3      ; yes, jump to label BBBB
;
13A8  FE02      CP    #02          ; Did P2 win?
13AA  2814      JR    Z,#13C0      ; yes, jump to label CCCC
;
; Equal scores
;
13AC  110B03    LD    DE,#031A      ; load task data for text #0B "DRAW" – label DDDD
13AF  CD9F30    CALL  #309F        ; insert task to draw text
13B2  C9        RET                ; return
; P1 has highest score
;
13B3  110203    LD    DE,#0302      ; load task data for text #2 "PLAYER 1" - label BBBB
13B6  CD9F30    CALL  #309F        ; insert task to draw text
13B9  110803    LD    DE,#031A      ; load task data for text #08 "WON THIS DUEL"
13BC  CD9F30    CALL  #309F        ; insert task to draw text
13BF  C9        RET                ; return
;
; P2 has highest score
;
13C0  110303    LD    DE,#0303      ; load task data for text #3 "PLAYER 2" – label CCCC
13C3  CD9F30    CALL  #309F        ; insert task to draw text
13C6  110803    LD    DE,#031A      ; load task data for text #08 "WON THIS DUEL"
13C9  CD9F30    CALL  #309F        ; insert task to draw text
13CC  C9        RET                ; return
; Screen timer expired, reset GameMode1 and GameMode2
;
13CD  3E01      LD    A,#01          ; A := 1 – label YYYY
13CF  32827D    LD    (REG_FLIPSCREEN),A ; set screen flip setting
13D2  320560    LD    (GameMode1),A ; store into game mode1
13D5  320760    LD    (NoCredits),A ; set indicator for no credits
13D8  3E00      LD    A,#00          ; A := 0
13DA  320A60    LD    (GameMode2),A ; reset game mode2 to 0.
13DD  323A62    LD    (#623A),A      ; reset screen timer
;
13E0  C9        RET                ; return

```

Gaat wel goed. Maar toch nog een aantal zaken die voor de volgende game niet goed staan. De cursor in het optie menu staat te laag en gaat daardoor niet goed. En de score loopt door. Dus blijkbaar worden de boards won niet gereset.

Nu versie v0.32 opgeslagen.

Teksten nog aanpassen:

Tekst "WON THIS DUEL" opnemen als nog vrije tekst 08

365B: EC 3F

3FEC: B2 76 27 1F 1E 10 24 18 19 23 10 14 25 15 1C 3F

Tekst "DUEL ENDED IN DRAW" opnemen als nog vrije tekst 0B

3661: 6F 14

14 6F: 32 77 14 25 15 1C 10 15 1E 14 15 14 10 19 1E 10 11 10 14 22 11 27 3F

Gaat goed. Alleen de plek nog aanpassen. Moet plek van EQUAL SCORES tekst worden.

Is nu goed.

Nu versie v0.33 opgeslagen.

Voor het options menu, de optiecursor weer resetten en de board scores voor beide spelers resetten.

Onderstaande code wordt aangeroepen in GameMode2=0 (en dat is dus voor GameMode2 = 1 waarin het opties menu getoond wordt.

0BA5	CD1C01	CALL	#011C	; clear all sounds
0BA8	3E05	LD	A,#05	; load A with 05
0BAA	322962	LD	(#6229), A	; set level to 05
OBAD	3E03	LD	A,#03	; load A with 03
OBAF	323562	LD	(#6235),A	; set rounds max to 03
0BB2	C9	RET		; return

Aanpassen:

OBAF	CDE113	CALL	#13E1	; call additional routine
------	--------	------	-------	---------------------------

en:

13E1	323562	LD	(#6235),A	; set rounds max to 03
;				
13E4	3E00	LD	A,#00	; load A with 00


```

13E6  323762      LD      (#6237),A      ; reset selected menu option
13E9  323862      LD      (#6238),A      ; reset boards won P1
13EC  323962      LD      (#6239),A      ; reset boards won P2
;
13EF  C9          RET                      ; return

```

6232 (board order) en 6236 (scoring method) ook op 0 zetten?

Gaat goed, maar de board order wordt niet meteen getoond. Pas aan het einde van het eerste boars, wanneer de stand aangepast wordt. Dus hier alvast de stand meteen weer op nul zetten. Wanneer wordt de speler score gereset? Dat is meteen wanneer er naar het optiemenu gesprongen wordt.

Dus daar waar de scores van P1 en P2 gereset worden ook de board scores resetten en de board score tonen en meteen de reset doen van de selected menu option.

Dus terug naar versie v0.33.

De reset wordt ergens gedaan in de 2 Players start: #0919 t/m #095D.

Zo maken:

```

0951  CDE113      CALL    #13E1          ; call additional code

```

en:

```

13E1  CD9F30      CALL    #309F          ; insert task
;
13E4  3E00        LD      A,#00          ; load A with 00
13E6  323762      LD      (#6237),A      ; reset selected menu option
13E9  323862      LD      (#6238),A      ; reset boards won P1
13EC  323962      LD      (#6239),A      ; reset boards won P2
;
13EF  C38A15      JP      #158A          ; display board score

```

Gaat nu goed. Zet de board score bij starten van spel op nul en displayed de board score. Maar ook nog na het optie selectiescherm nogmaals de display board score routine aanroepen. Want als je scoring method TOTAL kiest, moet het display weer weggehaald worden (of juist weergegeven wanneer je eerst TOTAL gekozen had, maar nu voor BOARD kiest).

Dus meteen na het kiezen van de opties wordt GameMode2 = 3 gezet en worden de variabelen geïntialiseerd. Daar de boardscore routine aanroepen. Daar worden nu ook nog de board HS gereset. Maar dat wordt niet meer gebruikt. Dus dat kan weg.

```

0BC2  3E01        LD      A,#01          ; load A with 01
0BC4  323362      LD      (#6233),A      ; Set actual round to 01
0BC7  320F60      LD      (#600F),A      ; Set two player game
;

```

```

0BCA  3E00      LD      A,#00          ; load A with 00
0BCC  323462    LD      (#6234),A      ; Set actual board number to 00
;
0BCF  214062    LD      HL,#6240      ; load HL with start location for board HS
0BD2  061F      LD      B,#1F        ; For B = 1 to 1F
0BD4  77        LD      (HL),A        ; reset memory location
0BD5  23        INC      HL          ; next memory location
0BD6  10FC      DJNZ    #AAAA        ; next B
;
0BCF  CD8A15    CALL    #158A          ; display board score
;
0BD2 – 0BD7    NOP's
;
0BD8  210A60    LD      HL,600A        ; load HL with GameMode2 address
0BDB  3603      LD      (HL),#03        ; set game mode 2 to 3
;
0BDD  C9        RET                    ; return

```

Ja dat gaat nu ook goed.

Alleen de board score meteen bij het opstarten van het spel weergeven. Dus ergens helemaal aan het begin wordt het bord goedgezet en ook de P2 score getekend. Dat gebeurde in het originele spel niet, dus dat is al toegevoegd. Daar ook een aanroep voor het tekenen van de board score toevoegen.

Dat is als volgt aangepast: 2UP score tekenen naar voren geplaatst en het tekenen van de HIGH SCORE verwijderd.

```

01FA  CDB107    CALL    #07B1          ; call additional code – call label AAAA
01FD  CD9F30    CALL    #309F          ; insert task
0200  110002    LD      DE,#0200        ; load task #2, parameter 0 to display player 1 score
0203  CD9F30    CALL    #309F          ; insert task
0206  C9        RET                    ; return

```

en

```

07B1  CDEE09    CALL    #09EE          ; draw "2UP" on screen – label AAAA
07B4  110102    LD      DE,#0201        ; load task #2, parameter 1 to display player 2 score
07B7  C9        RET                    ; return

```

Aanpassen:

```

07B4  C3F213    JP      #13F2          ; jump to additional code – jump to label BBBB

```

en

```

13F2  CD8A15    CALL    #158A          ; display board score
13F5  110102    LD      DE,#0201        ; load task #2, parameter 1 to display player 2 score

```

13F8 C9 RET

Ja dat gaat goed.

Nu versie v0.34 opgeslagen.

De volgende aanpassingen nog doen:

- Updaten board weergave wanneer volgende board — nu te laat
- Aanpassen menu optie selectie, onthouden vorige keuze, doorlopen aantal ronden
- Checken screen flip in cocktail mode
- High score scherm aanpassen met pakkende teksten en design by Paul Goes melding
- Wachttijden aanpassen: te lang (begin eerste barrels), te kort (einde rivets)
- Muziekje wanneer bonus tijd bijna om en dan dood, blijft doorspelen
- Aanpassen tweede How High tekst naar CAN YOU GET HIGHER?
- Eventueel ongebruikte muziekje onder eindscherm met weergave resultaat duel
- Eventueel nog animatie onder eindscherm met weergave resultaat duel — niet gedaan!
- Goofy Kong een bord geven met P1 vs P2 of DUEL erop

Nu nog de default 3 als start waarde voor aantal rondes eerder zetten. En dan de waardes van het vorige spel aanhouden: met druk op jump meteen dezelfde settings gebruiken. Daarnaast de rondes ook laten rondgaan: dus van 9->1 en van 1->9.

Waar wordt dat nu gezet? Hier:

OBA5	CD1C01	CALL	#011C	; clear all sounds
OBA8	3E05	LD	A,#05	; load A with 05
OBAA	322962	LD	(#6229), A	; set level to 05
OBAD	3E03	LD	A,#03	; load A with 03
OBAF	323562	LD	(#6235),A	; set rounds max to 03
OBB2	C9	RET		; return

Dus dat niet meer doen:

OBAD t/m OBB1 5x NOP's

En dan wel helemaal aan begin van spel doen. Toevoegen aan de additonele code die we zojuist hebben toegevoegd:

13F2	CD8A15	CALL	#158A	; display board score
13F5	3E03	LD	A,#03	; load A with 03
13F7	323562	LD	(#6235),A	; set rounds max to 03
13FA	110102	LD	DE,#0201	; load task #2, parameter 1 to display player 2 score
13FD	C9	RET		

En de overloop van het wijzigen van de rounds max zodat 1 verlagen geeft 9 en 9 verhogen geeft 1.

0B82	3A3562	LD	A,(#6235)	; load A with rounds max – label CCCC
0B85	FE01	CP	#01	; A == 1?
0B87	2002	JR	NZ,#0B42	; no, skip next step
0B89	3E02	LD	A,#02	; load A with 02
0B89	3E0A	LD	A,#0A	; load A with 0A
0B8B	3D	DEC	A	; decrement A
0B8C	323562	LD	(#6235),A	; store in rounds max
0B8F	C39F0B	JP	#0B9F	; jump forward – jump to label EEEE
;				
0B92	3A3562	LD	A,(#6235)	; load A with rounds max – label DDDD
0B95	3C	INC	A	; increment A
0B96	FE0A	CP	#0A	; A == 0A?
0B98	2002	JR	NZ,#0B2D	; no, skip next step
0B9A	3E09	LD	A,#09	; load A with 09
0B9A	3E01	LD	A,#01	; load A with 01
0B9C	323562	LD	(#6235),A	; store in rounds max

Ja dat gaat nu goed.

Nu versie v0.35 opgeslagen.

Nu nog het juiste board updaten wanneer het volgende board gekozen is. Wordt nu pas gedaan wanneer het board gestart is. Maar in het 'wacht op P1/P2 button' scherm al meteen goed weergeven.

Waar wordt het nieuwe board bepaald? Dat gebeurt hier:

0C01	78	LD	A,B	; restore A from B – label CCCC
0C02	3C	INC	A	; find next round number
0C03	323362	LD	(#6233),A	; store next round number
0C06	0601	LD	B,#01	; reset board number
;				
0C08	78	LD	A,B	; restore A from B – label BBBB
0C09	323462	LD	(#6234),A	; store next board number
;				
0C0C	C9	RET		; return

Bij label CCCC wordt een nieuwe ronde gestart en het board gereset geselecteerd.

Bij label BBBB wordt het juiste (opgehoogd of bij nieuwe ronde gereset) board weggeschreven.

Waar wordt de ronde/boardinformatie weergegeven?

Dat begint bij #06C7 en eindigt in een RET. Dus dan met een CALL aanroepen.

Aanpassen:

0C09	CDFE13	CALL	#13FE	; call to additional code – call to label AAAA
------	--------	------	-------	--

en:

```
13FE  323462      LD      (#6234),A      ; store next board number
1401  C3C706      JP       #06C7        ; jump to code that draws "R=..."
```

Lijkt goed te gaan. Meteen de tekst "R=" aangepast na selectie in het options menu. Dan het wacht scherm voor P1 en het How High scherm voor P1, maar daarna gaat het spelen fout: springt naar een reset. Dus er gaat nog iets niet goed. Komt dat door het geklooi met de returns? Ja, ergens stond een JP, maar was een CALL: dan kloppen de returns niet.

Maar op een hoger niveau kan ook gekeken worden. De overkoepelende code waarin de "R=..." getekend wordt, is task #6. Misschien beter om dan de task #6 aan te roepen.

Aanpassen:

```
13FE  323462      LD      (#6234),A      ; store next board number
1401  110006      LD      DE,#0600        ; load task #6 parameter 0 to display "R=" and "L="
1404  CD9F30      CALL    #309F          ; insert task
1407  C9          RET                    ; return
```

Ja dat gaat goed.

Nu versie v0.36 opgeslagen.

Even checken met de Cocktail setup. Werkt het daar nog goed met het flippen van het scherm nu er extra schermen toegevoegd zijn?

Het wisselen tijdens het spelen gaat goed. Alleen de tussenschermen met resultaat wanneer scoring methode BOARD en het eindscherm worden na P2 ook omgekeerd getoond.

Keuze maken wat we willen. Je zou kunnen zeggen dat dat afhankelijk wordt van wie er gewonnen heeft. Of standaard die schermen altijd gewoon (niet omgekeerd) tonen. Keuze voor het laatste.

Dat zou je kunnen doen door aan het einde van een game altijd terug te keren naar de niet omgekeerde toestand.

GameMode2 = 0D : spelen board	GameMode2 = 0D : spelen board
GameMode2 = 16 : einde level animatie	GameMode2 = 0E : dood animatie
GameMode2 = 04	GameMode2 = 0F : einde dood
	GameMode2 = 04

Dus waar kan je het dan toevoegen? Doorloopt andere GameMode2 waarden.

Daarom toevoegen aan beide routines die de betreffende schermen tekenen.

De eerste routine is die BOARD WON afhandelt. Daar wordt de eerste keer de volgende code doorlopen:

```

1311  CD7408      CALL  #0874          ; clear the screen and sprites – label XXXX
;
1314  21867D      LD    HL,REG_PALETTE_A    ; load HL with palette bank
1317  3600        LD    (HL),#00          ; set palette bank selector
1319  23          INC    HL                ; next palette bank
131A  3600        LD    (HL),#00          ; clear palette bank selector

```

En de tweede routine is die DUEL WON afhandelt. Daar wordt de eerste keer de volgende code doorlopen:

```

137B  CD7408      CALL  #0874          ; clear the screen and sprites – label XXXX
;
137E  21867D      LD    HL,REG_PALETTE_A    ; load HL with palette bank
1381  3600        LD    (HL),#00          ; set palette bank selector
1383  23          INC    HL                ; next palette bank
1384  3600        LD    (HL),#00          ; clear palette bank selector

```

Dan beide clear screen CALLs vervangen door en CALL naar additionele code. En in de additionele code dan de clear screen doen en het flippen van het scherm.

Aanpassen:

```

1311  CD0814      CALL  #1408          ; call to additional code – call to label AAAA
137B  CD0814      CALL  #1408          ; call to additional code – call to label AAAA

```

en:

```

1408  3E01        LD    A,#01            ; load A with 01 – label AAAA
140A  32827D      LD    (REG_FLIPSCREEN),A ; set to no flipping
140D  CD7408      CALL  #0874          ; clear the screen and sprites
1410  C9          RET                    ; return

```

Dat gaat goed.

Nu versie v0.37 opgeslagen.

Kijken naar de vertraging bij het eerste bord spelen. Vreemd. Is alleen bij het eerste bord van het spel: 1-1, maar dan niet altijd: soms wel zonder vertraging. Maar daarna in ieder geval niet meer. Debuggen toont aan dat hij uit de interrupt routine springt op een gegeven moment en dan weer terugkeert naar waar hij mee bezig was. Vaak het knipperen van de 1UP.

Heeft dit te maken met de overgang van GameMode1 naar GameMode2 menu? Geen idee.

Daarnaast ook heel vreemd dat in de debugger dit altijd goed gaat. Nooit de vertraging. Hoe kun je dit dan debuggen?

Eerst eens kijken of dit wellicht met de P1 en P2 knoppen te maken heeft? Wat gebeurt er als we het 'wachtscherm' eventjes niet laten wachten op het indrukken van de knop? Dan gaan we meteen naar het How High scherm.

Dat zit hier:

```

14FB  3A2660      LD      A,(UprightCab) ; load A with Upright or Cocktail
14FE  FE00        CP      #00          ; Playing on CockTail?
1500  2817        JR      Z,#1519      ; yes, skip next steps – jump to label AAAA
;
1502  3A0D60      LD      A,(PlayerTurnA) ; load A with player that plays
1505  FE00        CP      #00          ; Is player 1 playing?
1507  2008        JR      NZ,#1511     ; no, skip next steps – jump to label BBBB
;
1509  3A007D      LD      A,(IN2)      ; load A with IN2
150C  FE04        CP      #04          ; is P1 button pressed?
150E  C0          RET      NZ          ; no, return – wait until pressed
150F  180F        JR      #1520        ; skip next steps
;
1511  3A007D      LD      A,(IN2)      ; load A with IN2 – label BBBB
1514  FE08        CP      #08          ; is P2 button pressed?
1516  C0          RET      NZ          ; no, return – wait until pressed
1517  1807        JR      #1520        ; skip next steps
;
1519  3A1060      LD      A,(InputState) ; load A with input – label AAAA
151C  CB7F        BIT      7,A         ; is jump button pressed?
151E  2805        RET      Z,          ; no, return - wait until pressed
;
1520  210A60      LD      HL,600A      ; load HL with GameMode2 address - CCCC
1523  3608        LD      (HL),#08     ; set game mode 2 to 08
;
1525  C9          RET                  ; return

```

Altijd Jump button checken door 1500 geen JR Z, maar JR te maken.

```

1500  1817        JR      #1519      ; yes, skip next steps – jump to label AAAA

```

Nee dat maakt niets uit. Nog steeds de extra wachttijd.

Eventjes debuggen en kijken waar hij op dat moment mee bezig is. Kijken of dat te achterhalen is. Bijvoorbeeld welke routine, stuk code... Maar dat werkt niet want in de debugger treedt de extra wachttijd niet op... Erg lastig zo.

Kijken bij de verschillende stukken code bij het spelen of daar een RST #18 aangeroepen wordt.

De board afhandeling zijn de volgende stukken code die vanuit de main met GameMode sprongen aangeroepen worden:

070A	91 0C		; (4) clear screen and setup	; 0C91
070C	3C 12		; (5) set sprite, lives and level	; 123C
070E	7A 19		; (6) play game	; 197A

Op #0C91 wordt een RST #18 gedaan:

```
; arrive here from #0701 when game mode = 9
; clears screen, update timers, draws current screen, sets background music,

0C91 DF      RST      #18          ; count down WaitTimerMSB and only continue when 0

; arrive here from #0776 during attract mode

0C92 CD7408  CALL     #0874        ; clears the screen and sprites
0C95 AF      XOR      A            ; A := 0
0C96 328C63  LD        (#638C),A    ; reset onscreen timer
```

Op #123C wordt een RST #18 gedaan:

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

123C DF      RST      #18          ; count down WaitTimerMSB and only continue when 0
123D 3A2762  LD        A, (#6227)   ; load a with screen number
1240 fe03    CP        #03          ; is this the elevators?
1242 0116E0  LD        BC, #e016    ; B := #E0, C := #16. used for X,Y coordinates
1245 cA4B12  JP        Z, #124B     ; if elevators skip next step
```

Op 197A wordt geen RST #18 gedaan.

Proberen #0C91 NOP maken. Lijkt het probleem te verhelpen. Wel alert blijven of dit geen bijeffecten heeft.

Nu versie v0.38 opgeslagen.

Inderdaad bij beëindigen van rivets wordt meteen na het stoppen van het draaien van de ogen het rivets veld verlaten en gesprongen naar een status scherm terwijl het muziekje nog niet afgelopen is.

GameMode2 = 16 is einde level animatie en daarna wordt er teruggesprongen naar GameMode2 = 4.

Aan het einde van een level wordt inderdaad de WaitTimerMSB gezet en dan de GameMode2 = 4 gezet. Dat is op twee plekken:

17B2	23	INC	HL	; HL := GameMode2
17B3	3604	LD	(HL), #04	; set game mode2 to 4
17B5	C9	RET		; return

en

```
1967  23          INC    HL          ; increase HL to GameMode2
1968  3604        LD     (HL),#04    ; set game mode2 to 4
196A  C9          RET                ; return
```

Maar in GameMode2 = 4 wordt geen RST #18 gedaan. Dus wellicht op te lossen door een extra GameMode tussen te voegen.

Aanpassen:

```
17B3  3617        LD     (HL),#17    ; set game mode2 to 17
1968  3617        LD     (HL),#17    ; set game mode2 to 17
```

en:

```
0730  11 14              ; (17) Waiting until animation ready
```

en:

```
1411  DF          RST    #18          ; count down WaitTimerMSB
1412  3E04        LD     A,#04        ; A := 04
1414  320A60      LD     (GameMode2),A ; reset game mode2 to 4.
;
1417  C9          RET
```

Ja dat werkt goed. Rivets blijft mooi wachten totdat het muziekje afgelopen is.

Nu versie v0.39 opgeslagen.

Kijken wat er gebeurt wanneer je dood gaat of het board finished met de bonus timer lager dan 1000 en het geluid daarvoor afgespeeld wordt. Eerder geconstateerd dat dit dan door blijft spelen.

Gaat nu bij alle boards goed voor zowel dood gaan als board finish. Kwam waarschijnlijk doordat toen de GameMode2 volgorde nog niet goed was en spelen halverwege onderbroken werd.

Ook nog eventjes testen wanneer de hamer actief is. Kijken of het hamer geluid ook niet doorspeelt. Gaat ook goed bij zowel hamer actief wanneer dood en wanneer level finish.

Geluid onder scherm met einde DUEL melding.

Additionele tekst CAN YOU GET HIGHER ?

De tekst wordt hier geplaatst:

```
154B  110703      LD     DE,#0307    ; load task data for text #7 "HOW HIGH"
154E  CD9F30      CALL  #309F        ; insert task to draw text
```

Het plaatsen dan afhankelijk maken of dit de eerste beurt of de tweede beurt is: #6237: 1 = eerste beurt; 2 = tweede beurt.

Er is nog een tekst slot vrij: #1D. Dus die daarvoor gebruiken.

Aanpassen:

```
154B  C31814      JP      #1418      ; jump to additional code
```

en:

```
1418  110703      LD      DE,#0307      ; load task data for text #7 "HOW HIGH CAN ..."
141B  3A3762      LD      A,(#6237)      ; load A with turn number
141E  FE01        CP      #01          ; is this the first turn?
1420  2803        JR      Z,#BBBB      ; yes skip next step
1422  111D03      LD      DE,#031D      ; load task data for text #1D "CAN YOU GET ..."
1425  C34E15      JP      #154E        ; jump back
```

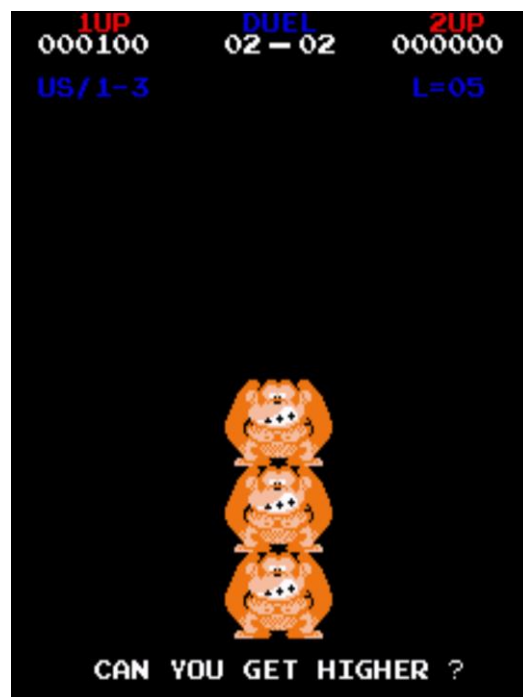
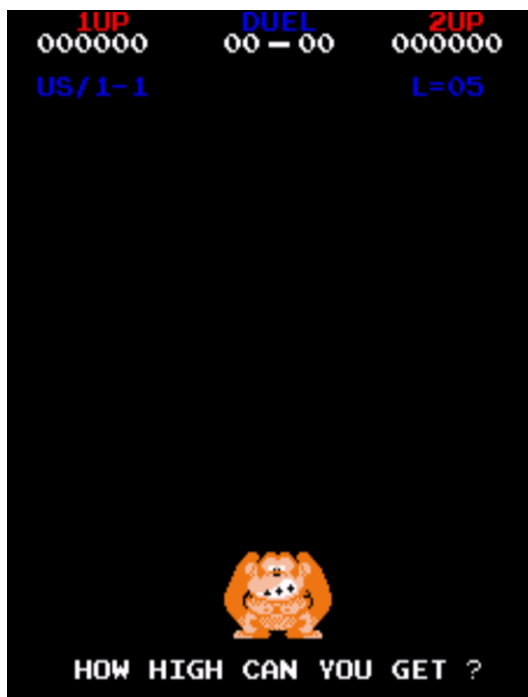
en:

```
3685  58 14              1D          : #YYYY "CAN YOU GET HIGHER ?"
```

en:

```
1458:  3E 77 13 11 1E 10 29 1F 25 10 17 15 24 10 18 19 17 18 15 22 10 FB 3F
```

Ja dat gaat goed.



Nu versie v0.40 opgeslagen.

Nu muziekje onder het einduitslagscherm. Eerst proberen met het muziekje wat normaal gespeeld wordt wanneer DK in het introscherm omhoog komt.

Dat is muziekje 01.

Het einduitslagscherm wordt hier voor het eerst opgebouwd:

1369	3A3A62	LD	A,(#623A)	; load A with screen timer
136C	3C	INC	A	; increment screen timer
136D	323A62	LD	(#623A),A	; store new screen timer
;				
1370	FE01	CP	#01	; is screen timer 01? – first time
1372	CA7B13	JP	Z,#137B	; yes, jump to label XXXX
;				
1375	FEB0	CP	#B0	; is screen timer B0 – last time
1377	CACD13	JP	Z,13CD	; yes, jump to label YYYY
;				
137A	C9	RET		; return
;				
137B	CD7408	CALL	#0874	; clear the screen and sprites – label XXXX
;				
137E	21867D	LD	HL,REG_PALETTE_A	; load HL with palette bank
1381	3600	LD	(HL),#00	; set palette bank selector
1383	23	INC	HL	; next palette bank
1384	3600	LD	(HL),#00	; clear palette bank selector

Maar daar is al additionele code aan toegevoegd:

1311	CD0814	CALL	#1408	; call to additional code – call to label AAAA
137B	CD0814	CALL	#1408	; call to additional code – call to label AAAA

en:

1408	3E01	LD	A,#01	; load A with 01 – label AAAA
140A	32827D	LD	(REG_FLIPSCREEN),A	; set to no flipping
140D	CD7408	CALL	#0874	; clear the screen and sprites
1410	C9	RET		; return

Dus nu aan de additionele code toevoegen:

Maar dat is voor zowel board einde (1311) als duel einde (137B). Maar alleen voor 137B regelen.

Aanpassen:

137B	CD2814	CALL	#1428	; call to additional code – call to label BBBB
------	--------	------	-------	--

en:

1428	218A60	LD	HL,#608A	; load HL with music buffer
142B	3601	LD	(HL),#01	; play music for end results screen
142D	23	INC	HL	; load HL with duration
142E	3603	LD	(HL),#03	; set duration to 3
1430	CD0814	CALL	#1408	; call to original additional code
1433	C9	RET		; return

Ja dat gaat goed. Nu nog het muziekje aanpassen.

500:	00	db \$00	; %00000000 0: No music
501:	20	db \$20	; %00100000 1: Music when DK climbs ladder
502:	10	db \$10	; %00010000 2: How high can you get?
503:	05	db \$05	; %00000101 3: Running out of time
504:	06	db \$06	; %00000110 4: Hammer music
505:	12	db \$12	; %00010010 5: Music after beating even-numbered rivet levels
506:	40	db \$40	; %01000000 6: Hammer hit
507:	16	db \$16	; %00010110 7: Music for completing a non-rivet stage
508:	01	db \$01	; %00000001 8: Music for barrel stage
509:	02	db \$02	; %00000010 9: Music for pie factory
50A:	00	db \$00	; %00000000 A: Music (or lack thereof) for elevator stage
50B:	04	db \$04	; %00000100 B: Music for rivet stage
50C:	14	db \$14	; %00010100 C: Music after beating odd-numbered rivet levels
50D:	FA	db \$fa	; %11111010 D: Used when rivet removed
50E:	1E	db \$1e	; %00011110 E: Music when DK is about to fall in rivet stage
50F:	2D	db \$2d	; %00101101 F: DK roars

Dus de regel op 501 vervangen door het nummer van de ongebruikte tune: pattern 12.

Aanpassen in s_3i_b.bin:

501:	12	db \$12	; music for end of duel
------	----	---------	-------------------------

Ja dat gaat goed. Alleen scherm iets langer laten staan...

Dat wordt hier geregeld:

1369	3A3A62	LD	A,(#623A)	; load A with screen timer
136C	3C	INC	A	; increment screen timer
136D	323A62	LD	(#623A),A	; store new screen timer
;				
1370	FE01	CP	#01	; is screen timer 01? – first time
1372	CA7B13	JP	Z,#137B	; yes, jump to label XXXX
;				
1375	FEB0	CP	#B0	; is screen timer B0 – last time
1377	CACD13	JP	Z,13CD	; yes, jump to label YYYY

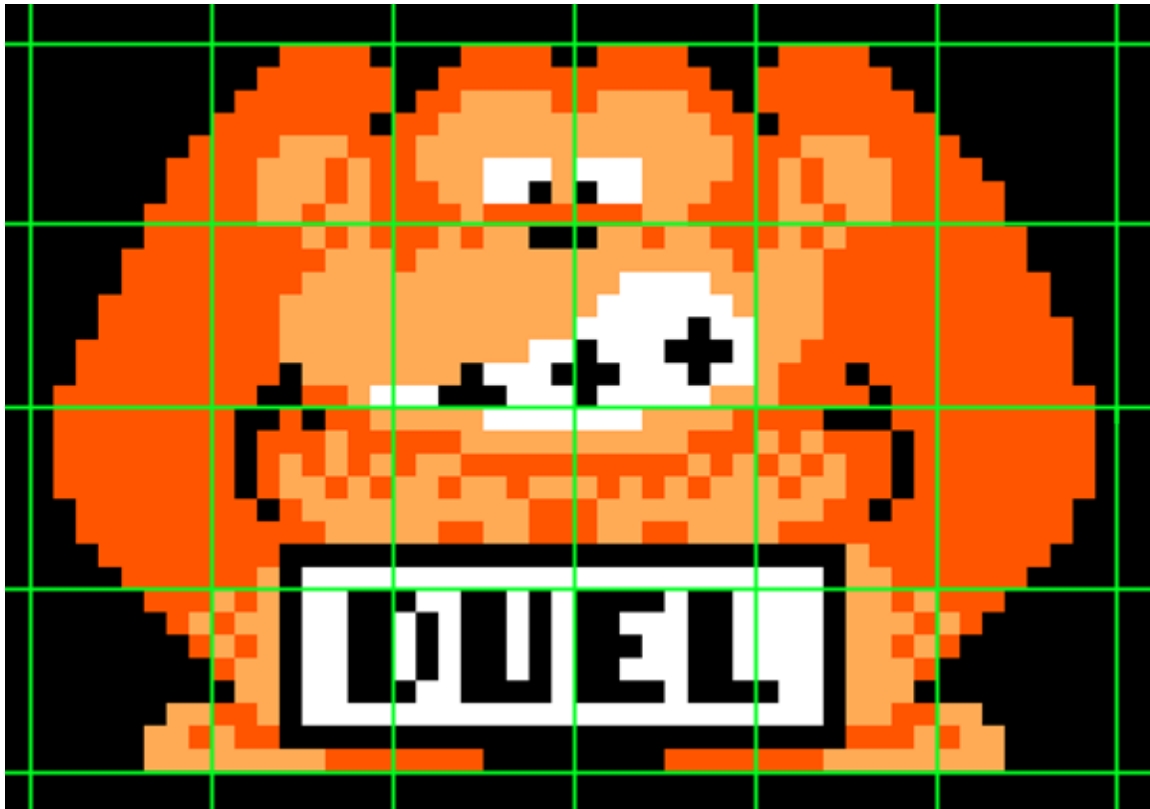
Aanpassen:

1375 FEF8 CP #F8 ; is screen time F8 – last time

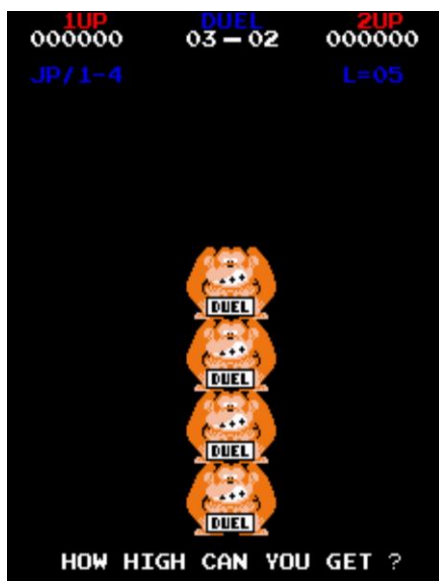
Ja dat is goed. Had nog langer gekund. Maar ook prima dat het overgaat naar het high score scherm.
Daar nog de teksten aanpassen.

Nu versie v0.41 opgeslagen.

De Goofy Kong graphic hierin veranderen:



Ziet er goed uit:



Nu versie v0.42 opgeslagen.

Kijken of het eindscherm toch nog langer getoond kan worden zodat het muziekje helemaal uit kan spelen.

Wordt hier geregeld:

```
1369 3A3A62 LD A,(#623A) ; load A with screen timer
136C 3C INC A ; increment screen timer
136D 323A62 LD (#623A),A ; store new screen timer
;
1370 FE01 CP #01 ; is screen timer 01? – first time
1372 CA7B13 JP Z,#137B ; yes, jump to label XXXX
;
1375 FEF8 CP #F8 ; is screen timer F8 – last time
1377 CACD13 JP Z,13CD ; yes, jump to label YYYY

...

;
; Screen timer expired, reset GameMode1 and GameMode2
;
13CD 3E01 LD A,#01 ; A := 1 – label YYYY
13CF 32827D LD (REG_FLIPSCREEN),A ; set screen flip setting
13D2 320560 LD (GameMode1),A ; store into game mode1
13D5 320760 LD (NoCredits),A ; set indicator for no credits
13D8 3E00 LD A,#00 ; A := 0
13DA 320A60 LD (GameMode2),A ; reset game mode2 to 0.
13DD 323A62 LD (#623A),A ; reset screen timer
;
13E0 C9 RET ; return
```

Eigenlijk na verstrijken van de screen timer een tweede starten, zolang die niet verlopen is gewoon RET blijven doen totdat de tweede verlopen is en dan pas echt eindigen. Voor de tweede timer geheugenplaats #6240 gebruiken. Die werd eerst voor de board HS gebruikt, maar dat wordt niet meer gedaan. Werd ook gereset, maar dat is ook verwijderd:

```
0BCF 214062 LD HL,#6240 ; load HL with start location for board HS
0BD2 061F LD B,#1F ; For B = 1 to 1F
0BD4 77 LD (HL),A ; reset memory location
0BD5 23 INC HL ; next memory location
0BD6 10FC DJNZ #AAAA ; next B
;
0BCF CD8A15 CALL #158A ; display board score
```

Aanpassen:

Reset wel weer doen, maar niet voor hele blok, maar alleen voor de #6240:

```
OBCF 3E00 LD A,#00 ; load A with 00
OBD1 324062 LD (#6240),A ; reset screen timer 2
OBD4 CD8A15 CALL #158A ; display board score
```

en:

```
1377 CA3414 JP Z,#1434 ; yes, jump to label AAAA
```

en

```
1434 3A3A62 LD A,(#623A) ; load A with screen timer – label AAAA
1437 3D DEC A ; decrement screen timer
1438 323A62 LD (#623A),A ; store new screen timer
;
143B 3A4062 LD A,(#6240) ; load A with screen timer 2
143E 3C INC A ; increment screen timer 2
143F 324062 LD (#6240),A ; store new screen timer 2
;
1442 FE8 CP #A8 ; is screen timer A8?
1444 C0 RET NZ ; no, return
;
1445 CACD13 JP Z,13CD ; yes, jump to label YYYY
```

Ja dat gaat goed.

Nu versie v0.43 opgeslagen.

Nu nog de high score table aanpassen naar een stuk pakkende tekst en design by melding.

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
```

Per regel maximaal 25 karakter posities.

```
BOARDS JP: 1-2-3-4 10 12 1F 11 22 14 23 10 1A 20 2E 10 10 10 10 10 01 2C 02 2C 03 2C 04
BOARDS US: 1-2-1-3-1-4 10 12 1F 11 22 14 23 10 25 23 2E 10 01 2C 02 2C 01 2C 03 2C 01 2C 04
MAXIMUM # OF ROUNDS: 9 10 1D 11 28 19 1D 25 1D 10 BB 10 1F 16 10 22 1F 25 1E 14 23 2E 10 09
BOARDS WON OR HIGH SCORE 12 1F 11 22 14 23 10 27 1F 1E 10 1F 22 10 18 19 17 18 10 23 13 1F 22 15
```

Op de oude plek van PLAYER1 PLAYER 2 de tekst PLAY WITH A FRIEND.

De oude titels staan op #379E:

92 77 22 11 1E 1B 10 10 23 13 1F 22 15 10 10 1E 11 1D 15 10 10 10 10 3F

Veranderen in:

11 77 20 1C 11 29 10 27 19 24 18 10 11 10 16 22 19 15 1E 14 10 10 10 3F

En de kleuren aanpassen van de regels.

Dit zit in het bestand v-5e.bpr:

BOARDS US (kolom 06) rood (04) maken en MAXIMUM # (kolom 08) en BOARDS WON (kolom 0A) groen (03) maken.

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

Ja de kleuren zijn nu goed. Ziet er nu zo uit:



Nu versie v0.44 opgeslagen.

Eventueel kijken andere kleuren jumpman.

Dat lijkt hier gezet te worden:

```
124B DD210062 LD IX,#6200 ; set IX to mario sprite array
124F 214C69 LD HL,#694C ; load HL with address for mario sprite X value
1252 DD360001 LD (IX+#00),#01 ; turn on sprite
1256 DD7103 LD (IX+#03),C ; store X position
1259 71 LD (HL),C ; store X position
125A 2C INC L ; next
125B DD360780 LD (IX+#07),#80 ; store sprite graphic
125F 3680 LD (HL),#80 ; store sprite graphic
1261 2C INC L ; next
1262 DD360802 LD (IX+#08),#02 ; store sprite color
1266 3602 LD (HL),#02 ; store sprite color
1268 2C INC L ; next
1269 DD7005 LD (IX+#05),B ; store Y position
126C 70 LD (HL),B ; store Y position
126D DD360F01 LD (IX+#0F),#01 ; turn this on (???)
```

De originele kleur #02.



De kleur #0C is wel aardig.



De kleur #0E is ook aardig.



Het vreemde is dat de kleuren: #03, #04, #05 en #06 allemaal hetzelfde zijn:



Heeft dat een reden? Of wordt er één gebruikt en kan een van de anderen hergebruikt worden om bijvoorbeeld een jumpman met groene kleding (ala Luigi) te maken?

035200 'palette'

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
10																
20																
30																
40																
50																
60																
70																
80																
90																
A0																
B0																
C0																
D0																
E0																
F0																

De kleuren van de sprites komen overeen met #80 t/m #BF:

80																
90																
A0																
B0																

Je ziet daar ook dat 3, 4, 5 en 6 hetzelfde zijn. Kunnen we dit patroon terugvinden in één van de romfiles? Van 256 bytes groot? Kleurcode op 80:

Ja dat komt overeen met de combinatie van de c-2j.bpr en de c-2k.bpr. Daarin gaan schuiven. En kijken effect op alle sprites en characters. En dan kijken of je een 'luigi' kan samenstellen.

Eventjes testen met aanpassen van sprite color #05: van #94 t/m #97.

Daar de kleuren van #02: van #88 t/m #8B neerzetten: de kleuren van jumpman en dan de rood van #96 vervangen door de groen van #0F. En dat in beide files doen: c-2j.bpr en c-2k.bpr.

De kleur van jumpman is nu inderdaad met groen. Maar het verziekt wel een aantal lijnen in het palette van girders en elevators. En nu blijkt ook dat de kleur van jumpman ook anders is op pies en rivets.

De twee files vormen de 4 palettes lijkt wel. Eventjes in plaatje aangeven met lijnen.

Het lijkt erop dat je dan dus de 05 in alle palettes moet aanpassen om overal jumpman groen te maken. Eventjes testen.

Ja dat klopt. Nu is jumpman in alle drie de palettes (de onderste 3 rijen uit het palette overzicht) met een groene kleur. Maar ook in alle drie de palettes zijn nu dezelfde twee lijnen niet de juiste kleur. Dat komt door het overall lijnen patroon voor een scherm.

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

Dat zijn waarschijnlijk de lijnen van de kolommen 07 en 0C. Wat als we daar ook #06 van maken?

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

Ja dat gaat goed.

Nu versie v0.45 opgeslagen.

Nu de hele felle kleur groen aanpassen naar een donkerder variant die meer op luigi's kleur lijkt.

Mooiere donkergroen: j/k: 0E BB: dus dat is op de #56, #96 en #D6.

Ja dat is de perfecte groen!

En dan nog ervoor zorgen dat de kleur van jumpman afhankelijk van de speler gezet wordt.

De speler wordt bepaald in GameMode2 = 5 en GameMode2 = 6.

GameMode2 = 5

070C 86 14 ; (5) Set Player1

en:

1486	3E00	LD	A,#00	; load A with 00
1488	320D60	LD	(PlayerTurnA),A	; set for player 1
148B	320E60	LD	(PlayerTurnB),A	; set for player 1
;				
148E	3E01	LD	A,#01	; load A with 01
1490	32827D	LD	(REG_FLIPSCREEN),A	; set to no flipping
;				
1493	CD7408	CALL	#0874	; clear screen and all sprites
1496	210A60	LD	HL,600A	; load HL with GameMode2 address
1499	3607	LD	(HL),#07	; set game mode 2 to 07
149B	C9	RET		; return

GameMode2 = 6

070E 9C 14 ; (6) Set Player2

en:

149C	3E01	LD	A,#01	; load A with 01
149E	320D60	LD	(PlayerTurnA),A	; set for player 2
14A1	320E60	LD	(PlayerTurnB),A	; set for player 2
;				
14A4	3A2660	LD	A,(UpRightCab)	; load A with UprightCab
14A7	32827D	LD	(REG_FLIPSCREEN),A	; flipping depends on dipswitches
;				
14AA	CD7408	CALL	#0874	; clear screen and all sprites
14AD	210A60	LD	HL,600A	; load HL with GameMode2 address
14B0	3607	LD	(HL),#07	; set game mode 2 to 07
14B2	C9	RET		; return

Dus daar goed gaan zetten.

Aanpassen:

1493	3E02	LD	A,#02	; load original jumpman color
1495	CDED15	CALL	#AAAA	; call additional code – call label AAAA
1498-149A		3x NOP		

en:

14AA	3E05	LD	A,#05	; load different jumpman color
14AC	CDED15	CALL	#15ED	; call additional code – call label AAAA
14AF-14B1		3x NOP		

en:

15ED	324E69	LD	(#694E),A	; set jumpman color to sprite – label AAAA
15F0	320862	LD	(#6208),A	; set jumpman color to data array
15F3	CD7408	CALL	#0874	; clear screen and all sprites
15F6	210A60	LD	HL,600A	; load HL with GameMode2 address
15F9	3607	LD	(HL),#07	; set game mode 2 to 07
15FB	C9	RET		; return XXXX

En de regels die de kleur zetten NOP maken. Maar dat gaat niet goed omdat de sprite waarden blijkbaar na het zetten hierboven gereset worden. Maar doordat het niet meer gezet wordt, wordt de kleur 00 gebruikt.

Met debugger kijken waar de jumpman sprite color allemaal aangepast wordt. Bij elke clear screen worden de sprites ook gewist. Dus aanpassing hierboven heeft geen zin.

Terug naar versie v0.45 en de kleur donkergroen terugzetten.

En dan dus de aanpassing hier doen:

1262	DD360802	LD	(IX+#08),#02	; store sprite color
1266	3602	LD	(HL),#02	; store sprite color

Afhankelijk maken van PlayerTurnA: #600D: = 00 (P1) of 01 (P2).

1262	CDED15	CALL	#15ED	; call additional code – call label AAAA
1265 t/m 1267		3x NOP		

en:

15ED	3A0D60	LD	A,(#600D)	; load A with PlayerTurnA – label AAAA
15F0	FE00	CP	#00	; A : = 00
15F2	3E02	LD	A,#02	; load A with 02 = original jumpman color
15F4	2802	JR	Z,#BBBB	; yes, skip next step
15F6	3E05	LD	A,#05	; load A with 05 = different jumpman color
;				
15F8	DD7708	LD	(IX+#08),A	; load jumpman data table with correct color
15FB	77	LD	(HL),A	; load jumpman sprite with correct color
15FC	C9	RET		; return

Dit lijkt goed te gaan.

Nu versie v0.46 opgeslagen.

Nog eventjes een keer testen met cocktail mode. Ja gaat nog allemaal goed.

Onsterfelijkheid weer uitzetten.

Nu versie v0.47 opgeslagen.

Dit is de final version v1.00.

Gaat toch nog iets niet goed.

Al eerder geconstateerd, maar dacht dat het opgelost was. Maar wanneer de speler doodgaat wanneer de bonus timer bijna afgelopen is en het muziekje daarvan speelt, dan blijft dat doorspelen in het 'start' scherm van de volgende speler. Dus bij doodgaan moet eigenlijk het geluid uitgezet worden.

Deze code wordt doorlopen aan het begin van een board einde en voor een duel einde.

1408	3E01	LD	A,#01	; load A with 01 – label AAAA
140A	32827D	LD	(REG_FLIPSCREEN),A	; set to no flipping
140D	CD7408	CALL	#0874	; clear the screen and sprites
1410	C9	RET		; return

Aanpassen:

140D	CDFD15	CALL	#15FD	; call additional code
------	--------	------	-------	------------------------

en:

15FD	CD1C01	CALL	#011C	; clear all sounds
1600	CD7408	CALL	#0874	; clear the screen and sprites
1603	C9	RET		; return

Maar moet eigenlijk ook aangeroepen worden vanuit het 'start' scherm. Dat is GameMode2 = 2.

Dat gebeurt op twee plekken:

1493	CD7408	CALL	#0874	; clear screen and all sprites
...				
14AA	CD7408	CALL	#0874	; clear screen and all sprites

Deze allebei aanpassen:

1493	CDFD15	CALL	#15FD	; call additional code
------	--------	------	-------	------------------------

...

14AA CDFD15 CALL #15FD ; call additional code

Nu de versie in het titelscherm aanpassen naar v1.01.

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

Aanpassen:

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

Nu versie v1.01 opgeslagen.

Door de bug fix van v1.01 is het muzikje aan het einde van het duel weggefallen. Waar werd dat gedaan?

Hier dus:

137B CD2814 CALL #1428 ; call to additional code – call to label BBBB

en:

1428	218A60	LD	HL,#608A	; load HL with music buffer
142B	3601	LD	(HL),#01	; play music for end results screen
142D	23	INC	HL	; load HL with duration
142E	3603	LD	(HL),#03	; set duration to 3
1430	CD0814	CALL	#1408	; call to original additional code
1433	C9	RET		; return

Maar dat is inderdaad vreemd omdat het muzikje eerst in #1428-142F aaangezet wordt, maar daarna de call gedaan wordt waarin alle muziek weer uitgezet wordt.

Aanpassen:

1428	CD0814	CALL	#1408	; call to original additional code
142B	218A60	LD	HL,#608A	; load HL with music buffer
142E	3601	LD	(HL),#01	; play music for end results screen
1430	23	INC	HL	; load HL with duration
1431	3603	LD	(HL),#03	; set duration to 3
1433	C9	RET		; return

Ja dat werkt.

Nu de versie in het titelscherm aanpassen naar v1.02.

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

Aanpassen:

#373E: 3D 76 26 01 2B 00 02 10 3F (V1.01)

Nu versie v1.02 opgeslagen.