

## Racked Off

By Richard Bayliss

### Programmed in ACME cross assembler

This is the improved source that uses IRQs outside the Kernal, unlike the version of this game I wrote and released on to CSDB when I used \$0314 and \$0315 for interrupts. The game data + source was done in realaunch64 and was assembled in ACME cross assembler. I thought I kindly contribute this piece of source code to the C64 Codebase and also a link for you to grab the binaries and source code (in .a format).

For this source, you'll need the following tools:

- ACME cross assembler - Pucrunch/Exomizer - WinVice/CCS64/Hoxs64

Assemble with: acme -v3 game.a

then compress with either of the two crunchers:

PuCrunch: pucrunch racked\_off.prg -s rackedoff.prg -x \$3000

or

Exomizer: exomizer sfx \$3000 racked\_off.prg -o rackedoff.prg -n

Please feel free to improve and modify this source code, to make the game work better if you have to. If you do, please let me know via email or via PM on the CSDB. Thanks

[Download source and binaries](#)

Here's the source code:

```
=====
;Racked Off by Richard Bayliss
;ACME cross assembler
;=====

;some logical labels
objpos = $0370      ;Expanded position for sprite object
sync = $0340       ;Synchronized delay to get routines working outside IRQ
colstore = $03f0   ;sprite to sprite collision table
enemy1_direction = $02 ;Direction pointer for enemy 1
enemy2_direction = $03 ;Direction pointer for enemy 2
enemy3_direction = $04 ;Direction pointer for enemy 3
enemy4_direction = $05 ;Direction pointer for enemy 4
enemy1_released = $06 ;Release pointer for enemy 1
enemy2_released = $07 ;Release pointer for enemy 2
enemy3_released = $08 ;Release pointer for enemy 3
enemy4_released = $09 ;Release pointer for enemy 4
```

```

enemy_anim_pointer = $0a ;Sprite animation pointer
enemy_anim_delay = $0b ;Sprite animation duration
player_anim_delay = $0c ;Player sprite animation duration
player_anim_pointer = $0d ;Animation pointer for player
food_counter = $0e ;Amount of food
level_screen = $0f ;The level counter

levelcolour1 = $10 ;Background colour 1
levelcolour2 = $11 ;Background colour 2
levelcolour3 = $12 ;Background colour 3
dirface = $13 ;Direction of player
xpause = $15 ; Smooth scroll pointer for title screen
bgrdelay = $16 ;Background delay
foodstore = $03e0 ; Value of food
                !to "racked_off.prg",cbm
;set up the title screen, initialize scroll text and much more
;=====
                ;Import Richard's font/gfx char
                * = $0800
                !binary "char.chr"
                ;Import Richard's game music
                * = $1000-2
                !binary "music.prg"
                ;Import Richard's game sprite
                * = $2000
                !binary "sprites.spr"
                ;Code and game jump address is here
                * = $3000
                jsr $c6dc ; Execute Jon Wells MSCK screen
                jsr $c708 ; data and code
                lda #$08 ;
                jsr $ffd2 ;

;=====
;Title screen code
;=====
title          sei
                lda #$37 ;Turn Kernal back on
                sta $01
                lda #$00
                sta bgrdelay
                lda #$00 ;Silence
                sta $d418
                sta $d01a ;No IRQ running
                lda #$81
                sta $dc0d
                ;Initialize the title scroll text
                lda #<scrolltext
                sta read+1
                lda #>scrolltext
                sta read+2
                lda #$00

```

```
    sta $d020
    sta $d021
    sta xpause ; Reset smooth pointer
    lda #$12
    sta $d018 ; Game charset
    lda #$18
    sta $d016 ; Multicolour is on
    lda #$0a
    sta $d022
    lda #$08
    sta $d023
    lda #$0a
    sta levelcolour1
    lda #$08
    sta levelcolour2
    lda #$09
    sta levelcolour3
    lda #$00
    sta $d020
    sta $d021
;Clear screen and add bitmap colours

    ldx #$00
clearscreen lda #$20
    sta $0400,x
    sta $0500,x
    sta $0600,x
    sta $06e8,x
    lda $5800,x
    sta $d800,x
    lda $5900,x
    sta $d900,x
    lda $5a00,x
    sta $da00,x
    lda $5ae8,x
    sta $dae8,x
    inx
    bne clearscreen
    ldx #$00
pastecredits lda titletext1,x
    sta $0590,x
    lda titletext2,x
    sta $0590+40,x
    lda titletext3,x
    sta $0590+80
    lda titletext4,x
    sta $0590+120,x
    lda titletext5,x
    sta $0590+160,x
    lda titletext6,x
    sta $0590+200,x
```

```
        lda tittletext7,x
        sta $0590+240,x
        lda tittletext8,x
        sta $0590+280,x
        lda tittletext9,x
        sta $0590+320,x
        lda tittletext10,x
        sta $0590+360,x
        lda tittletext11,x
        sta $0590+400,x
        inx
        cpx #$28
        bne pastecredits
colourize   ldx #$00
           lda #$01
           sta $d990,x
           sta $d990+40,x
           lda #$07
           sta $d990+120,x
           sta $d990+160,x
           lda #$0d
           sta $d990+240,x
           sta $d990+280,x
           lda #$0a
           sta $d990+360,x
           sta $d990+400,x
           inx
           cpx #$28
           bne colourize
           lda #$08
           sta $d016
           lda #$0b
           sta $d011
           lda #$00
           sta $02
           sta $03
wait_delay   inc $02
           lda $02
           cmp #$fc
           bne wait_delay
           lda #$00
           sta $02
           inc $03
           lda $03
           cmp #$fc
           bne wait_delay
           lda #$1b
           sta $d011
           lda #$35 ;Switch kernal on
           sta $01
           lda #$00 ;Title music initialized
```

```

        jsr $9000
        lda #<tirq
        sta $fffe
        lda #>tirq
        sta $ffff
        lda #$7f
        sta $dc0d
        sta $dd0d
        lda $dc0d
        lda $dd0d
        lda #$01
        sta $d01a
        lda #$00
        sta $d012
        lda #$1b
        sta $d011
        lda $dc0d
        lsr $d019
        cli
        ;jmp *
titleloop  lda #0
           sta sync
           lda sync
syncwaittit  cmp sync
           beq syncwaittit
           jsr scroller
           lda $dc90
           lsr
           lsr
           lsr
           lsr
           lsr
           bcs me
           jmp game
me          lda $dc91
           lsr
           lsr
           lsr
           lsr
           bcs maintit
           jmp game
maintit    jmp titleloop
;Routine for the smooth scroller and scroll text
;control
scroller   lda xpause
           sec
           sbc #$01
           and #$07
           sta xpause
           bcs endscroll

```

```
wrapscreen    ldx #$00
              lda $0799,x
              sta $0798,x
              lda #$0e
              sta $db98,x
              inx
              cpx #$28
              bne wrapscreen
read          lda $07bf
              cmp #$00
              bne nowrap
              lda #<scrolltext
              sta read+1
              lda #>scrolltext
              sta read+2
              jmp read
nowrap       sta $07bf
              inc read+1
              lda read+1
              cmp #$00
              bne endscroll
              inc read+2
endscroll    rts

;Main multiple IRQ for the title screen

tirq         pha
              txa
              pha
              tya
              pha
              lda $d019
              sta $d019
              lda #$2e
              sta $d012
              ldx #$0c
time1        dex
              bne time1
              lda #$03
              sta $dd00
              lda #$1b
              sta $d011
              lda #$12
              sta $d018
              lda xpause
              sta $d016
              ;inc $d020
              lda #<tirq2
              sta $fffe
              lda #>tirq2
              sta $ffff
```

```
                lda #$01
                sta sync
                jsr $9003
                pla
                tay
                pla
                tax
                pla
                rti
tirq2          pha
                txa
                pha
                tya
                pha
                lda $d019
                sta $d019
                lda #$72
                sta $d012
                ldx #$0c
time2         dex
                bne time2
                lda #$02
                sta $dd00
                lda #$3b
                sta $d011
                lda #$78
                sta $d018
                lda #$18
                sta $d016
                lda #<tirq3
                sta $fffe
                lda #>tirq3
                sta $ffff
                pla
                tay
                pla
                tax
                pla
                rti
tirq3          pha
                tya
                pha
                txa
                pha
                lda $d019
                sta $d019
                lda #$e4
                sta $d012
                ldx #$03
time3         dex
                bne time3
```

```

        lda #$03
        sta $dd00
        lda #$1b
        sta $d011
        lda #$12
        sta $d018
        lda #$08
        sta $d016
        lda #<tirq
        sta $fffe
        lda #>tirq
        sta $ffff
        pla
        tay
        pla
        tax
        pla
        rti
        ;jsr $e544
        jmp game ;jump to game for the timebeing

```

```

;=====
;we start to play the game so set up the levels and sprites priorities,
;start positions, colour, etc.
;=====

```

```

game          sei
              lda #$37
              sta $01
              lda #$81
              sta $dc0d
              lda #0
              sta $d01a
              sta $d418
              ldx #$00
copystat      lda statusdefault,x
              sta statusstore,x
              inx
              cpx #$28
              bne copystat
              lda #00
              sta level_screen
              lda #<level1ctr
              sta levelcount+1
              lda #>level1ctr
              sta levelcount+2
gameloop      sei
              lda #$37
              sta $01

```



```
        lda #$00
        sta $d01a
        lda #$81
        sta $dc0d

clearscrn
        ldx #$00
        lda #$20
        sta $0400,x
        sta $0500,x
        sta $0600,x
        sta $06e8,x
        inx
        bne clearscrn
        lda #$1b
        sta $d011
        lda #$12
        sta $d018
        lda #$18
        sta $d016
        ldx level_screen
        jsr $ce02
        lda levelcolour3
        sta $d021
        sta $d020
        lda levelcolour1
        sta $d022
        lda levelcolour2
        sta $d023
levelcount
        ldx #$00
        lda level1ctr,x
        sta foodstore,x
        inx
        cpx #$03
        bne levelcount
        lda #$07
        sta $d027
        lda #$01
        sta $d025
        lda #$00
        sta $d026
        lda #$80
        sta $07f8
        lda #$18
        sta objpos+$00
        lda #$60
        sta objpos+$01
        lda #$ff
        sta $d015
        sta $d01c
        lda #$86
```

```
sta $07f9
lda #$84
sta $07fa
lda #$88
sta $07fb
lda #$8a
sta $07fc
lda #$0a
sta $d028
lda #$0d
sta $d029
lda #$03
sta $d02a
lda #$0f
sta $d02b
lda #$0c
sta objpos+$02
lda #$90
sta objpos+$03
lda #$a0
sta objpos+$04
lda #$90
sta objpos+$05
lda #$58
sta objpos+$06
lda #$e0
sta objpos+$07
lda #$40
sta objpos+$09
lda #$58
sta objpos+$08
lda #$00
sta objpos+$0a
sta objpos+$0b
sta objpos+$0c
sta objpos+$0d
sta objpos+$0e
sta objpos+$0f
lda #$00
sta enemy1_released
sta enemy2_released
sta enemy3_released
sta enemy4_released
lda #$00
sta enemy1_direction
sta enemy3_direction
lda #$01
sta enemy2_direction
sta enemy4_direction
lda #$00
sta enemy_anim_pointer
```

```
        sta enemy_anim_delay
        sta player_anim_pointer
        sta player_anim_delay
whitetext    ldx #$00
            lda #1
            sta $d800,x
            lda statusstore,x
            sta $0400,x
            inx
            cpx #$28
            bne whitetext
            lda #$35
            sta $01
            lda #<irq
            sta $fffe
            lda #>irq
            sta $ffff
            lda #<nmi
            sta $fffa
            lda #>nmi
            sta $fffb
            lda #$7f
            sta $dc0d
            lda #$01
            sta $d01a
            lda #$00
            jsr $1000
            lda #$36
            sta $0424
            lda #$30
            sta $0425
            sta $0426
            sta $0427
            lda level_screen
            cmp #12
            beq fixscreen
            jmp clrflag
fixscreen    lda #$4f
            sta $0727
            sta $0727+$28
            lda #$00
            sta $db27
            sta $db27+$28
            lda $dd0d
            lsr $d019
clrflag     cli
mainloop    lda #$00
            sta sync
            lda sync
syncwait    cmp sync
            beq     syncwait
```

```

        jsr expand
        jsr joyread
        jsr backcollision
        jsr enemy1_routine
        jsr enemy2_routine
        jsr enemy3_routine
        jsr enemy4_routine
        jsr animate_enemies
        jsr arealldone
        jsr collision
        jsr animate_background
        jsr time
        jsr time
        jmp mainloop
irq
        pha
        txa
        pha
        tya
        pha
        lda $d019
        sta $d019
        lda #0
        sta $d012
        lda #1
        sta sync
        jsr $1003
        jsr colroll
        pla
        tay
        pla
        tax
        pla
nmi
        rti
;expand the size of the play field for all the game sprites

expand
expandloop
        ldx #$00
        lda objpos+$01,x
        sta $d001,x
        lda objpos+$00,x
        asl
        ror $d010
        sta $d000,x
        inx
        inx
        cpx #$10
        bne expandloop
        rts

joyread
        lda $dc90
        lsr
        bcs down

```

```
mu          jsr animate_player
            lda #$01
            sta dirface
            ldx objpos+$01
            dex
            dex
            cpx #$52
            bcs setup
            ldx #$52
setup       stx objpos+$01
            rts
down       lsr
            bcs left
md         jsr animate_player
            lda #$02
            sta dirface
            ldx objpos+$01
            inx
            inx
            cpx #$ce
            bcc setdown
            ldx #$ce
setdown    stx objpos+$01
            rts
left       lsr
            bcs right
ml         jsr animate_player
            lda #$03
            sta dirface
            ldx objpos+$00
            dex
            cpx #$12
            bcs setleft
            ldx #$12
setleft    stx objpos+$00
            rts
right      lsr
            bcs nojoy
mr         jsr animate_player
            lda #$04
            sta dirface
            ldx objpos+$00
            inx
            cpx #$9c
            bcc setright
            ldx #$9c
setright   stx objpos+$00
nojoy      rts
colroll    lda colours+$00
            sta colours+$28
```

```
wrapcols      ldx #$00
              lda colours+$01,x
              sta colours+$00,x
              lda colours+$00,x
              sta $d800,x
              lda colours+$10
              sta $dbc0,x
              inx
              cpx #$28
              bne wrapcols
              rts
;player sprite to background collision routines
backcollision: lda objpos+$01
              sec
              sbc #$32
              lsr
              lsr
              lsr
              tay
              lda scrlo,y
              sta $70
              lda scrhi,y
              sta $71
              lda objpos+$00
              sec
              sbc #$0a
              lsr
              lsr
              tay
              ldx #3
              sty _selfmod+1
_l1:          ;piny
              lda ($70),y
              cmp #65
              beq _hit
              lda ($70),y
              cmp #65 ;player touches the food,
              beq _hit; if touched the food is off screen
              cmp #66;
              beq _hit;
              cmp #67;
              beq _hit;
              cmp #68;
              beq _hit;
              iny
              lda ($70),y
              cmp #65
              beq _hit
              cmp #67
              beq _hit
              cmp #66
```

```

        beq _hit
        cmp #68
        beq _hit
        cmp #$56           ;player touches the switch char
        beq _switch       ;if touched, the switch will turn
        cmp #$57           ;all the water into food
        beq _switch
        cmp #$58
        beq _switch
        cmp #$59
        beq _switch
        cmp #$4a           ;player touches the water chars
        beq _water         ;if the player touches the water
        cmp #$4b           ;he will drown
        beq _water
        cmp #$4c
        beq _water
        cmp #$4d
        beq _water
        cmp #$4e
        beq _water
        cmp #$52           ;player touches the rock chars
        beq _rock          ;if touched, then the player
        cmp #$53           ;will stop according to the
        beq _rock          ;direction he moves
        cmp #$54
        beq _rock
        cmp #$55
        beq _rock
        ;iny
        jmp _selfmod

        lda ($70),y
        cmp #16
        bpl _hit

_selfmod:
mode2:   ldy #$00
        lda $70
        clc
        adc #$28 ;next row
        sta $70
        bcc _l2
        inc $71
_l2:     dex
        bne _l1
        rts
_water:  jsr player_drown ;because player touched water, he drowns
        rts
_hit:    lda #73
        sta ($70),y
        jsr score

```

```

        jsr addfood
        ;cmp #$00
        ;beq leveledone
no:     rts
_switch jsr water_to_food ;turn water into food
        rts
_rock   jsr stop_at_rock   ;stop player moving
        rts
stop_at_rock lda dirface
          cmp #$01
          beq stopatup
          cmp #$02
          beq stopatdown
          cmp #$03
          beq stopatleft
          cmp #$04
          beq stopatright
          rts
stopatup  ldx objpos+$01
          inx
          inx

stopatbgrd stx objpos+$01
          rts
stopatdown ldx objpos+$01
          dex
          dex
          stx objpos+$01
          rts
stopatleft ldx objpos+$00
          inx
          ;inx
          stx objpos+$00
          rts
stopatright ldx objpos+$00
          dex
          ;dex
          stx objpos+$00
          rts

player_drown ldx #$00
showd1  lda deathtext1,x
        sta $07c0,x
        lda #$02
        sta $dbc0,x
        inx
        cpx #$28
        bne showd1
        jmp player_is_hit

addfood   ;player eats food

```



```

        dec foodstore+2
        lda foodstore+2
        cmp #$2f
        beq nextbit
        rts
nextbit  lda #$39
        sta foodstore+2
        dec foodstore+1
        lda foodstore+1
        cmp #$2f
        beq lastbit
        rts
lastbit  lda #$39
        sta foodstore+1
        dec foodstore+0
        rts
arealldone  lda foodstore+2
        cmp #$30
        bne endcheck
        lda foodstore+1
        cmp #$30
        bne endcheck
        lda foodstore+0
        cmp #$30
        bne endcheck
levelisdone  lda #$02
        jsr $1000
        ldx #$00
showmess7  lda message2,x
        sta $07c0,x
        inx
        cpx #$28
        bne showmess7
        lda #$00
        sta $d015
loopit    lda $dc90
        lsr
        lsr
        lsr
        lsr
        lsr
        bcs loopit2
        jmp nextlev
loopit2  lda $dc01
        lsr
        lsr
        lsr
        lsr
        lsr
        bcs loopit
nextlev  inc level_screen

```

```
        inc $041d
        lda $041d
        cmp #$3a
        bne good
        lda #$30
        sta $041d
        inc $041c
good      ldx #$00
copybit2  lda $0400,x
          sta statusstore,x
          inx
          cpx #$28
          bne copybit2
          jsr checklevel
endcheck  rts

checklevel  lda level_screen
           cmp #$01 ;level 2
           bne notlev2
           lda #$04
           sta levelcolour1
           lda #$03
           sta levelcolour2
           lda #$06
           sta levelcolour3
           lda #<level2ctr
           sta levelcount+1
           lda #>level2ctr
           sta levelcount+2
           jmp gameloop
notlev2    lda level_screen
           cmp #$02
           bne notlev3
           lda #$08
           sta levelcolour1
           lda #$09
           sta levelcolour2
           lda #$0b
           sta levelcolour3
           lda #<level3ctr
           sta levelcount+1
           lda #>level3ctr
           sta levelcount+2
           jmp gameloop
notlev3    lda level_screen
           cmp #$03
           bne notlev4
           lda #$0f
           sta levelcolour1
           lda #$0c
           sta levelcolour2
```

```
        lda #$0b
        sta levelcolour3
        lda #<level4ctr
        sta levelcount+1
        lda #>level4ctr
        sta levelcount+2
        jmp gameloop
notlev4  lda level_screen
        cmp #$04
        bne notlev5
        lda #$07
        sta levelcolour1
        lda #$0a
        sta levelcolour2
        lda #$02
        sta levelcolour3
        lda #<level5ctr
        sta levelcount+1
        lda #>level5ctr
        sta levelcount+2
        jmp gameloop
notlev5  lda level_screen
        cmp #$05
        bne notlev6
        lda #$01
        sta levelcolour1
        lda #$0d
        sta levelcolour2
        lda #$05
        sta levelcolour3
        lda #<level6ctr
        sta levelcount+1
        lda #>level6ctr
        sta levelcount+2
        jmp gameloop
notlev6  lda level_screen
        cmp #$06
        bne notlev7
        lda #$03
        sta levelcolour1
        lda #$0e
        sta levelcolour2
        lda #$04
        sta levelcolour3
        lda #<level7ctr
        sta levelcount+1
        lda #>level7ctr
        sta levelcount+2
        jmp gameloop
notlev7  lda level_screen
        cmp #$07
```

```

    bne notlev8
    lda #$0c
    sta levelcolour1
    lda #$0f
    sta levelcolour2
    lda #$0b
    sta levelcolour3
    lda #<level8ctr
    sta levelcount+1
    lda #>level8ctr
    sta levelcount+2
    jmp gameloop
notlev8    lda level_screen
           cmp #$08
           bne notlev9
           lda #$0a
           sta levelcolour1
           lda #$08
           sta levelcolour2
           lda #$09
           sta levelcolour3
           lda #<level9ctr
           sta levelcount+1
           lda #>level9ctr
           sta levelcount+2
           jmp gameloop
notlev9    lda level_screen
           cmp #$09
           bne notlev10
           lda #$0e
           sta levelcolour1
           lda #$03
           sta levelcolour2
           lda #$09
           sta levelcolour3
           lda #<level10ctr
           sta levelcount+1
           lda #>level10ctr
           sta levelcount+2
           jmp gameloop
notlev10   lda level_screen
           cmp #$0a
           bne notlev11
           lda #$0f
           sta levelcolour1
           lda #$0c
           sta levelcolour2
           lda #$0b
           sta levelcolour3
           lda #<level11ctr
           sta levelcount+1
```

```
notlev11    lda #>level11ctr
            sta levelcount+2
            jmp gameloop
            lda level_screen
            cmp #$0b
            bne notlev12
            lda #$02
            sta levelcolour1
            lda #$08
            sta levelcolour2
            lda #$09
            sta levelcolour3
            lda #<level12ctr
            sta levelcount+1
            lda #>level12ctr
            sta levelcount+2
            jmp gameloop
notlev12    lda level_screen
            cmp #$0c
            bne notlev13
            lda #$0d
            sta levelcolour1
            lda #$03
            sta levelcolour2
            lda #$02
            sta levelcolour3
            lda #<level13ctr
            sta levelcount+1
            lda #>level13ctr
            sta levelcount+2
            jmp gameloop
notlev13    lda level_screen
            cmp #$0d
            bne notlev14
            lda #$0f
            sta levelcolour1
            lda #$0e
            sta levelcolour2
            lda #$0b
            sta levelcolour3
            lda #<level14ctr
            sta levelcount+1
            lda #>level14ctr
            sta levelcount+2
            jmp gameloop
notlev14    lda level_screen
            cmp #$0e
            bne notlev15
            lda #$0d
            sta levelcolour1
            lda #$05
```

```
notlev15    sta levelcolour2
            lda #$09
            sta levelcolour3
            lda #<level15ctr
            sta levelcount+1
            lda #>level15ctr
            sta levelcount+2
            jmp gameloop
            lda level_screen
            cmp #$0f
            bne gamefinished
            lda #$03
            sta levelcolour1
            lda #$0e
            sta levelcolour2
            lda #$0b
            sta levelcolour3
            lda #<level16ctr
            sta levelcount+1
            lda #>level16ctr
            sta levelcount+2
            jmp gameloop
gamefinished sei
            lda #$37
            sta $01
            jmp game_complete
water_to_food ldx #$00
showmessage lda message,x
            sta $07c0,x
            lda #$07
            sta $dbc0,x
            inx
            cpx #$28
            bne showmessage
convloop    ldx #$00
            jsr waterloop1
            jsr waterloop2
            jsr waterloop3
            jsr waterloop4
            jsr switchgone1
            jsr switchgone2
            jsr switchgone3
            jsr switchgone4
            inx
            bne convloop
            rts
waterloop1  lda $0400,x
            cmp #$4a
            beq conv1
            cmp #$4b
```

```

    beq conv2
    cmp #$4c
    beq conv3
    cmp #$4d
    beq conv4
    cmp #$4e
    beq conv1
    rts
conv1   lda #$41
        sta $0400,x
        rts
conv2   lda #$42
        sta $0400,x
        rts
conv3   lda #$43
        sta $0400,x
        rts
conv4   lda #$44
        sta $0400,x
        rts

waterloop2  lda $0500,x
            cmp #$4a
            beq conv1b
            cmp #$4b
            beq conv2b
            cmp #$4c
            beq conv3b
            cmp #$4d
            beq conv4b
            cmp #$4e
            beq conv1b
            rts
conv1b   lda #$41
        sta $0500,x
        rts
conv2b   lda #$42
        sta $0500,x
        rts
conv3b   lda #$43
        sta $0500,x
        rts
conv4b   lda #$44
        sta $0500,x
        rts

waterloop3  lda $0600,x
            cmp #$4a
            beq conv1c
            cmp #$4b
            beq conv2c
```

```
    cmp #$4c
    beq conv3c
    cmp #$4d
    beq conv4c
    cmp #$4e
    beq conv1c
    rts
conv1c    lda #$41
          sta $0600,x
          rts
conv2c    lda #$42
          sta $0600,x
          rts
conv3c    lda #$43
          sta $0600,x
          rts
conv4c    lda #$44
          sta $0600,x
          rts

waterloop4    lda $06e8,x
              cmp #$4a
              beq conv1d
              cmp #$4b
              beq conv2d
              cmp #$4c
              beq conv3d
              cmp #$4d
              beq conv4d
              cmp #$4e
              beq conv1d
              rts
conv1d    lda #$41
          sta $06e8,x
          rts
conv2d    lda #$42
          sta $06e8,x
          rts
conv3d    lda #$43
          sta $06e8,x
          rts
conv4d    lda #$44
          sta $06e8,x
          rts

leveldone    jmp $3000 ;next level

score        inc $040a
              ldx #$05
scloop       lda $0407,x
              cmp #$3a
```



```
        bne scok
        lda #$30
        sta $0407,x
        inc $0406,x
scok    dex
        bne scloop
        rts
switchgone1 lda $0400,x
        cmp #$56
        beq isblank1
        cmp #$57
        beq isblank1
        cmp #$58
        beq isblank1
        cmp #$59
        beq isblank1
        rts
isblank1  lda #$20
        sta $0400,x
        rts
switchgone2 lda $0500,x
        cmp #$56
        beq isblank2
        cmp #$57
        beq isblank2
        cmp #$58
        beq isblank2
        cmp #$59
        beq isblank2
        rts
isblank2  lda #$20
        sta $0500,x
        rts

switchgone3 lda $0600,x
        cmp #$56
        beq isblank3
        cmp #$57
        beq isblank3
        cmp #$58
        beq isblank3
        cmp #$59
        beq isblank3
        rts
isblank3  lda #$20
        sta $0600,x
        rts

switchgone4 lda $06e8,x
        cmp #$56
        beq isblank4
```

```

        cmp #$57
        beq isblank4
        cmp #$58
        beq isblank4
        cmp #$59
        beq isblank4
        rts
isblank4    lda #$20
            sta $06e8,x
            rts

```

;the routines for enemy 1 (the enemy on the right of the screen)

```

enemy1_routine:
            lda enemy1_released
            cmp #$01
            bne not_released1
            jmp release_enemy
not_released1:  lda objpos+$03
            cmp objpos+$01
            bne contmove1
            lda #$01
            sta enemy1_released
            rts
contmove1:    lda enemy1_direction
            cmp #$00
            beq enemy1_up
            cmp #$01
            beq enemy1_down
            rts

```

;because the enemy spots player 1. make it move across the screen  
;to the right

```

release_enemy  lda objpos+$02
            clc
            adc #$01
            cmp #$b2
            bcc notoffset
            lda #$0c
            sta objpos+$02
            lda #$e0
            sta objpos+$03
            lda #$00
            sta enemy1_released
            rts
notoffset     sta objpos+$02
            rts
enemy1_up     lda objpos+$03
            sec

```

```
    sbc #$01
    cmp #$40
    bcs set_eup1
    lda #$3f
    sta objpos+$03
    lda #$01
    sta enemy1_direction
    rts
set_eup1    sta objpos+$03
           rts
enemy1_down    lda objpos+$03
              clc
              adc #$01
              cmp #$e2
              bcc set_edown1
              lda #$00
              sta enemy1_direction
              rts
set_edown1    sta objpos+$03
           rts
enemy2_routine
           lda enemy2_released
           cmp #$01
           bne not_released2
           jmp release_enemy2
not_released2    lda objpos+$05
                cmp objpos+$01
                bne not_release2
                lda #$01
                sta enemy2_released
                rts
not_release2    lda enemy2_direction
                cmp #$00
                beq enemy2_up
                cmp #$01
                beq enemy2_down
                rts
release_enemy2    lda objpos+$04
                 sec
                 sbc #$01
                 cmp #$02
                 bcs contmove2
                 lda #$48
                 sta objpos+$05
                 lda #$a0
                 sta objpos+$04
                 lda #$00
                 sta enemy2_released
                 rts
contmove2    sta objpos+$04
```

```
enemy2_up      rts
               lda objpos+$05
               sec
               sbc #$01
               cmp #$48
               bcs set_eup2
               lda #$01
               sta enemy2_direction
               rts
set_eup2       sta objpos+$05
               rts
enemy2_down    lda objpos+$05
               clc
               adc #$01
               cmp #$e2
               bcc set_edown2
               lda #$00
               sta enemy2_direction
               rts
set_edown2     sta objpos+$05
               rts
enemy3_routine
               lda enemy3_released
               cmp #$01
               bne not_released3
               jmp release_enemy3
not_released3  lda objpos+$06
               cmp objpos+$00
               bne cont_enemy3
               lda #$01
               sta enemy3_released
               rts
cont_enemy3    lda enemy3_direction
               cmp #$00
               beq enemy3_left
               cmp #$01
               beq enemy3_right
               rts
enemy3_left    lda objpos+$06
               sec
               sbc #$01
               cmp #$0c
               bcs setleft3
               lda #$01
               sta enemy3_direction
               rts
setleft3      sta objpos+$06
               rts
enemy3_right   lda objpos+$06
               clc
```

```

        adc #$01
        cmp #$a0
        bcc setright3
        lda #$00
        sta enemy3_direction
        rts
setright3    sta objpos+$06
            rts
;the player has been spotted by enemy 3, so this enemy will move upscreen
release_enemy3
            lda objpos+$07
            sec
            sbc #$01
            cmp #$02
            bcs offset3
            lda #$e0
            sta objpos+$07
            lda #$98
            sta objpos+$06
            lda #$00
            sta enemy3_released
            rts
offset3     sta objpos+$07
            rts
enemy4_routine
            lda enemy4_released
            cmp #$01
            bne not_released4
            jsr release_enemy4
            rts
not_released4    lda objpos+$08
                cmp objpos+$00
                bne contene4
                lda #$01
                sta enemy4_released
                rts
contene4       lda enemy4_direction
                cmp #$00
                beq enemy4_left
                cmp #$01
                beq enemy4_right
                rts
enemy4_left    lda objpos+$08
                sec
                sbc #$01
                cmp #$0c
                bcs setleft4
                lda #$01
                sta enemy4_direction

```

```
setleft4      rts
              sta objpos+$08
              rts
enemy4_right  lda objpos+$08
              clc
              adc #$01
              cmp #$a0
              bcc setright4
              lda #$00
              sta enemy4_direction
              rts
setright4     sta objpos+$08
              rts
;the player has been spotted by enemy 4, so this enemy will move downscreen
release_enemy4
              lda objpos+$09
              clc
              adc #$01
              cmp #$e0
              bcc offset4
              lda #$48
              sta objpos+$09
              lda #$92
              sta objpos+$08
              lda #$00
              sta enemy4_released
              rts
offset4      sta objpos+$09
              rts

;animate those moving enemies
animate_enemies
              inc enemy_anim_delay
              lda enemy_anim_delay
              cmp #$04
              beq reset_delay_pointer
              rts
reset_delay_pointer
              lda #$00
              sta enemy_anim_delay
              ldx enemy_anim_pointer
              lda enemy1_frame,x
              sta $07fa
              lda enemy2_frame,x
              sta $07f9
              lda enemy4_frame,x
              sta $07fc
              lda enemy3_frame,x
              sta $07fb
```

```

                                inx
                                cpx #$04
                                beq reset_anim
                                inc enemy_anim_pointer
                                rts
reset_anim                      ldx #$00
                                stx enemy_anim_pointer
                                rts
animate_player                  inc player_anim_delay
                                lda player_anim_delay
                                cmp #$08
                                beq do_anim
                                rts
do_anim                         lda #$00
                                sta player_anim_delay
                                ldx player_anim_pointer
                                lda player_frame,x
                                sta $07f8
                                inx
                                cpx #$04
                                beq reset_panim
                                inc player_anim_pointer
                                rts
reset_panim                     ldx #$00
                                stx player_anim_pointer
                                rts
;enemy to player sprite collision
collision                       lda objpos+$00
                                sec
                                sbc #$06
                                sta colstore+$00
                                clc
                                adc #$0c
                                sta colstore+$01
                                lda objpos+$01
                                sec
                                sbc #$0c
                                sta colstore+$02
                                clc
                                adc #$18
                                sta colstore+$03
                                ldx #$00
enemycolloop                    lda objpos+$02,x
                                cmp colstore+$00
                                bcc noenemycollision
                                cmp colstore+$01
                                bcs noenemycollision
                                lda objpos+$03,x
                                cmp colstore+$02
```

```

                bcc noenemycollision
                cmp colstore+$03
                bcs noenemycollision
                ldx #$00
showd2         lda deathtext2,x
                sta $07c0,x
                lda #$02
                sta $dbc0,x
                inx
                cpx #$28
                bne showd2

                jmp player_is_hit
noenemycollision
                inx
                inx
                cpx #$0e
                bne enemycolloop
                rts
;the player is hit by one of the enemy creatures. one way to solve this
;problem. player loses a life.
player_is_hit  lda #$01
                sta $d015
                lda #$8d
                sta $07f8
                lda #$01
                jsr $1000
awit          lda #$00
                sta $fd
                lda #$00
                sta $fe
pause        inc $fd
                lda $fd
                cmp #$fd
                bne pause
                lda #$00
                sta $fd
                inc $fe
                lda $fe
                cmp #$fd
                bne pause
                dec $0414
                lda $0414
                cmp #$30
                beq game_over
                ldx #$00
copystat2    lda $0400,x
                sta statusstore,x
                inx
                cpx #$28

```



```

                                bne copystat2
                                jmp gameloop
game_over                       lda #$04
                                jsr $1000
                                lda #$00
                                sta $d015
                                sta $d020
                                sta $d021
                                ldx #$00
goclr                           lda #$20
                                sta $0428,x
                                sta $0500,x
                                sta $0600,x
                                sta $06e8,x
                                lda #$01
                                sta $d828,x
                                sta $d900,x
                                sta $da00,x
                                sta $dae8,x
                                inx
                                bne goclr
;show message
                                ldx #$00
gomess                          lda gameoverscreen,x
                                sta $05e0,x
                                inx
                                cpx #$78
                                bne gomess
waitfire                        lda $dc90
                                lsr
                                lsr
                                lsr
                                lsr
                                lsr
                                bcs waitfire2
waitfire2                       jmp title
                                lda $dc01
                                lsr
                                lsr
                                lsr
                                lsr
                                lsr
                                bcs waitfire
                                jmp title
time                             dec $0427
                                ldx #$03
timeloop                        lda $0424,x
                                cmp #$2f
                                bne timeok
                                lda #$39
                                sta $0424,x
```

```
timeok      dec $0423,x
            dex
            bne timeloop
            lda $0424
            cmp #$2f
            bne nox
            lda #$30
            sta $0427
            sta $0426
            sta $0425
            sta $0424
            ldx #$00
messagemad  lda deathtext3,x
            sta $07c0,x
            inx
            cpx #$28
            bne messagemad
            jmp player_is_hit
nox         rts
;The game is complete so now we show the end screen
;Yeah I know. It is just something simple :)
game_complete sei
            lda #$37
            sta $01
            lda #$00
            sta $d418
            sta $d01a
            lda #$81
            sta $dc0d
            lda #$00
            sta $d020
            sta $d021
            sta $d015
            ldx #$00
clearscreen2 lda endtext,x
            sta $0400,x
            lda endtext+$100,x
            sta $0500,x
            lda endtext+$200,x
            sta $0600,x
            lda endtext+$2e8,x
            sta $06e8,x
            lda #$0d
            sta $d800,x
            sta $d900,x
            sta $da00,x
            sta $dae8,x
            inx
            bne clearscreen2
            sei
            lda #$7f
```

```

        sta $dc0d
        sta $dd0d
        lda $dc0d
        lda $dd0d
        lda #$01
        sta $d01a
        ldx #$00
        sta $d012
        lda #$1b
        sta $d011
        lda #$35
        sta $01
        lda #<endirq
        sta $fffe
        lda #>endirq
        sta $ffff
        lda #$03
        jsr $1000
        lda #$08
        sta $d016
        lda $dc0d
        lsr $d019
        cli
endwait    lda $dc01
           cmp #$ef
           bne endwait
           jmp title

endirq     pha
           txa
           pha
           tya
           pha
           lda $d019
           sta $d019
           lda #$00
           sta $d012
           jsr $1003
           lda #$01
           sta sync
           pla
           tay
           pla
           tax
           pla
           rti

```

```

;Animate the water chars, so it looks as if the water in the game
;is flowing. Use a slow speed, because if it is too fast, it would
;look crap. Water must go downwards, so reverse the char positions :)

```

```
animate_background
```

```

        inc bgrdelay
        lda bgrdelay
        cmp #$04
        beq waterflow
        rts
waterflow    lda #$00
            sta bgrdelay
            ldx #6
wrapwaterchar    lda $0a50,x
            sta $0a51,x
            lda $0a58,x
            sta $0a59,x
            lda $0a60,x
            sta $0a61,x
            lda $0a68,x
            sta $0a69,x
            lda $0a70-1,x
            sta $0a70,x
            dex
            bne wrapwaterchar
            lda $0a57
            sta $0a51
            lda $0a5f
            sta $0a59
            lda $0a67
            sta $0a61
            lda $0a7f
            sta $0a79
            lda $0a77
            sta $0a71
            rts

```

;The score status & a copy of the same status for  
;after a level is complete or a life is lost.

```
statusdefault !scr "score:000000 lives:03 level:01 time:6000"
```

```
statusstore !scr "score:000000 lives:03 level:01 time:6000"
```

```
gamescreen
```

;All the text for the game over screen

```
gameoverscreen
```

```

!scr "          g a m e   o v e r          "
!scr "
!scr "          p r e s s   f i r e   t o   p l a y   a g a i n          "

```

;All the text for the intro/title screen

```

titletext1    !scr "    copyright 2007 the new dimension    "
titletext2    !scr "          all rights reserved          "
titletext3    !scr "
titletext4    !scr "    programming, graphics and music by    "
titletext5    !scr "          richard bayliss          "

```

```

titletext6      !scr "                "
titletext7      !scr "          bitmap logo graphics done by  "
titletext8      !scr "                johan janssen          "
titletext9      !scr "                "
titletext10     !scr "          use joystick in port 2, and    "
titletext11     !scr "                press fire to play      "

;Various 1 line text messages depending on what happens
;in the game

deathtext2      !scr "  ouch! i bet that hurt. they got you!  "
deathtext1      !scr "oh, come on, silly fool. you can't swim!"
deathtext3      !scr "don't wait so long. you ran out of time!"
message         !scr "  nice, you have hit the magic switch  "
message2        !scr "well done, level cleared.          -fire!-"

;Text for the end screen

endtext         !scr "                "
                !scr "                "
                !scr "                "
                !scr "congratulations. you have completed all "
                !scr "16 levels of =racked off=. barry is not "
                !scr "racked off with your hard efforts to "
                !scr "finally help him get all the fruit from "
                !scr "the 16 gardens. "
                !scr "                "
                !scr "barry finds himself to be literally "
                !scr "bloated and can't eat any more. so he "
                !scr "sets of back home and rests in his cosy "
                !scr "bed once more. "
                !scr "                "
                !scr "we do hope you have enjoyed playing this"
                !scr "fun game, courtesey with: "
                !scr "                "
                !scr "                the new dimension "
                !scr "                "
                !scr "                http://www.redesign.sk/tnd64 "
                !scr "                "
                !scr "                "
                !scr "                "
                !scr "                "
                !scr "                "

;Table for the famous Richy Bayliss colour washing
;routine.
colours         !byte $00,$00,$02,$02,$06,$06,$04,$04
                !byte $05,$05,$07,$07,$01,$01,$01,$01

```

```

!byte $01,$01,$01,$01,$01,$01,$01,$01
!byte $01,$01,$01,$01,$01,$01,$07,$07
!byte $05,$05,$04,$04,$06,$06,$02,$02
!byte $00
xtmp          !byte $00

killerchars:  !byte
$10,$11,$12,$13,$14,$15,$16,$17,$18,$19,$1a,$1b,$1c,$1d,$1e,$1f
!byte
$20,$21,$22,$23,$24,$25,$26,$27,$28,$29,$30,$31,$32,$33,$34,$35
!byte
$36,$37,$38,$39,$3a,$3b,$3c,$3d,$3e,$3f,$40,$41,$42,$43,$44,$45
!byte
$46,$47,$48,$49,$4a,$4b,$4c,$4d,$4e,$4f,$50,$51,$52,$53,$54,$55
!byte
$56,$57,$58,$59,$5a,$5b,$5c,$5d,$5e,$5f,$60,$61,$62,$63,$64,$65
!byte
$66,$67,$68,$69,$6a,$6b,$6c,$6d,$6e,$6f,$70,$71,$72,$73,$74,$75
!byte
$76,$77,$78,$79,$7a,$7b,$7c,$7d,$7e,$7f,$80,$81,$82,$83,$84,$85
!byte
$86,$87,$88,$90,$8a,$8b,$8c,$8d,$8e,$8f,$90,$91,$92,$93,$94,$95
!byte
$96,$97,$98,$99,$9a,$9b,$9c,$9d,$9e,$9f,$a0,$a1,$a2,$a3,$a4,$a5
!byte
$a6,$a7,$a8,$a9,$aa,$ab,$ac,$ad,$ae,$af,$b0,$b1,$b2,$b3,$b4,$b5
!byte
$b6,$b7,$b8,$b9,$ba,$bb,$bc,$bd,$be,$bf,$c0,$c1,$c2,$c3,$c4,$c5
!byte
$c6,$c7,$c8,$c9,$ca,$cb,$cc,$cd,$ce,$cf,$d0,$d1,$d2,$d3,$d4,$d5
!byte
$d6,$d7,$d8,$d9,$da,$db,$dc,$dd,$de,$df,$e0,$e1,$e2,$e3,$e4,$e5
!byte
$e6,$e7,$e8,$e9,$ea,$eb,$ec,$ed,$ee,$ef,$f0,$f1,$f2,$f3,$f4,$f5
!byte
$f6,$f7,$f8,$f9,$fa,$fb,$fc,$fd,$fe,$ff,$ff,$ff,$ff,$ff,$ff,$ff

;Data tables for the screen data for sprite to char collision
;all chars from $0400-$07e7

scrhi          !byte
$04,$04,$04,$04,$04,$04,$04,$05,$05,$05,$05,$05,$05,$06,$06,$06,$06,$06,$06,
$06,$07,$07,$07,$07,$07,$07
scrlo          !byte
$00,$28,$50,$78,$a0,$c8,$f0,$18,$40,$68,$90,$b8,$e0,$08,$30,$58,$80,$a8,$d0,
$f8,$20,$48,$70,$98,$c0,$e0

!byte $00
;Sprite animation frames
enemy1_frame  !byte $83,$83,$84,$84

```

```
!byte $00
enemy2_frame !byte $86,$86,$85,$85
!byte $00
enemy3_frame !byte $88,$87,$87,$88
!byte $00
enemy4_frame !byte $89,$8a,$8a,$89
!byte $00
player_frame !byte $80,$81,$82,$81
!byte $00
;The amount of food to eat per level

level1ctr !scr "260"
level2ctr !scr "160"
level3ctr !scr "264"
level4ctr !scr "240"
level5ctr !scr "144"
level6ctr !scr "256"
level7ctr !scr "176"
level8ctr !scr "182"
level9ctr !scr "144"
level10ctr !scr "288"
level11ctr !scr "504"
level12ctr !scr "464"
level13ctr !scr "258"
level14ctr !scr "564"
level15ctr !scr "180"
level16ctr !scr "500"

;The scroll text for the title screen

scrolltext !scr "... hi there and be warmly welcomed to "
!scr "... racked off ... copyright (c)2007 th"
!scr "e new dimension ... all programming by "
!scr "richard bayliss ... game graphics by ri"
!scr "chard bayliss ... bitmap logo by johan "
!scr "janssen (jsl) ... music arranged and co"
!scr "mposed by richard bayliss ... use a joy"
!scr "stick plugged in port 2 when playing .."
!scr ". help barry the bear safely around the "
!scr "screen, chomping all the fruit, planted "
!scr "in 16 different gardens ... only one pr"
!scr "oblem though ... mutant bugs do not like"
!scr " barry scoffing the fruit, as they want "
!scr "it first, therefore they are racked off"
!scr " ... if one of those bugs spot exactly "
!scr "where you are, they will try and stop y"
!scr "ou from gobbling the fruit ... one way "
!scr "to avoid those mutant bugs is by moving"
!scr " out their way, else there will be more"
!scr " trouble ... because if you get caught "
!scr "at any time by those bugs, you will los"
```

```
!scr "e a life ... you also will need to keep"
!scr " an eye out for time as well, because i"
!scr "f by any chance you are too slow, you wi"
!scr "ll risk losing a life ... not good is i"
!scr "t? ... heheh, i thought not ... also if"
!scr " a mutant bug hits you, you will also l"
!scr "ose a life ... can you complete all 16 "
!scr "of the crazy levels before barry gets e"
!scr "ven more racked off and give up on his "
!scr "quest to scoff the scrummy fruit? ... t"
!scr "here is only one way to find out ... pr"
!scr "ess the fire button or the spacebar to "
!scr "play ... good luck, you will need it! "
!scr " "
!byte 0
* = $5800-2
!binary "colram.prg"
* = $5c00-2
!binary "vidram.prg"
* = $6000-2
!binary "bitmap.prg"
    * = $8ffe
    !binary "tiletune2.prg"
    * = $9ffe
!binary "levels.prg"
```

From:  
<http://codebase64.org/> - **Codebase 64 wiki**

Permanent link:  
[http://codebase64.org/doku.php?id=base:racked\\_off](http://codebase64.org/doku.php?id=base:racked_off)

Last update: **2015-04-17 04:33**

