

1032

://cadaver.homeftp.net/games/mwsrc.zip/player.s

sta joystick

```

;////////////////////////////////////L
;>SCANKEYS                                >
;                                           >
;>Reads keyboard. Ripped from Jukka Tapanimäki's C64 game programming guide. >
;                                           >
;>Parameters: -                            >
;>Returns: -                                >
;>Modifies: A,X,Y                          >
;////////////////////////////////////Y

```

```

scankeys:      lda #$ff
               sta keyval
               ldy #$07
sk_rowloop:    lda rowbit,y
               sta $dc00
               ldx $dc01
               cpx #$ff
               bne sk_keyfound
               dey
               bpl sk_rowloop
               lda #$ff
               sta lastkey
               rts
sk_keyfound:   sty keytemp
               ldy #7
sk_getcol:     txa
               and colbit,y
               beq sk_colfound
               dey
               bpl sk_getcol
sk_colfound:   ldx keytemp
               tya
               clc
               adc rowval,x
               cmp lastkey
               beq sk_samekey
               sta keyval
               sta lastkey
sk_samekey:    rts

```

```

;////////////////////////////////////L

```

1032

://cadaver.homeftp.net/games/mwsrc.zip/player.s

```

;≥INITPLAYER
;≥INITNEXTLIFE
;≥
;≥Initializes player in different ways. INITPLAYER resets lives & score,
;≥INITNEXTLIFE inits energy.
;≥
;≥Parameters: -
;≥Returns: -
;≥Modifies: A
;¿fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffY

```

```

initplayer:    lda #$00
               ldx #prevscore-score-1
initplayerloop: sta score,x
               dex
               bpl initplayerloop
               sta shpd
               lda #$05
               sta extracmp
               lda #$03
               sta lives
               jmp inl_maxpower
initnextlife:  lda #SHP_PLRRUN
               sta shpt
               ldx #$ff
               stx shporg
               inx
               stx shpf
               stx shpfd
               stx shpsy
               stx shpbits
               stx firedelay
               stx drunk
               lda #150           ;Immortality for 3 seconds
               sta shpimm
inl_maxpower:  lda #PLR_MAXPOWER
               sta shpp
               rts

```

```

; /fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffL
;≥CHECKDEATH
;≥
;≥Initializes new life/ends game when out of lives.

```

1032

://cadaver.homeftp.net/games/mwsrc.zip/player.s

```

;≥
;≥Parameters: -
;≥Returns: -
;≥Modifies: A
;ΔfffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffY

```

```

isdead:      lda shpt
              beq dead
              cmp #SHP_PLRDEAD

dead:        rts

checkdeath:  jsr isdead
              beq chd_noskip

chd_notyet:  rts
chd_noskip:  inc deadcnt           ;Wait 200 frames after death
              lda deadcnt
              cmp #200
              bcc chd_notyet

chd_livesdec: dec lives           ;Decrease lives
              beq chd_gameover    ;Go to gameover if out of them
              jsr leveltune_noskip
              jsr initnextlife
              lda lastxl
              sta shpxl
              lda lastxh
              sta shpxh
              lda lastyl
              sta shpyl
              lda lastyh
              sta shpyh
              lda lastt
              sta shpt
              rts

chd_gameover: pla                 ;Jump directly out of the main loop
              pla
              jmp gameover

```

```

//fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffL
;≥MOVPLR
;≥
;≥Player move routines (running, jumping, ducking, climbing)
;≥

```

1032

://cadaver.homeftp.net/games/mwsrc.zip/player.s

```

;≥Parameters: X:actor number (0)
;≥Returns: -
;≥Modifies: A,Y
;¿fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffY

```

```

movplr_drunk:   lda drunk
                beq movplr_notdrunk
                dec drunkd
                bpl movplr_nodd
                lda #$08                ;If player character has
                sta drunkd              ;drunk alcohol, use random
                dec drunk                ;controls
                jsr random
                and #JOY_LEFT+JOY_RIGHT
                sta drunkrnd
movplr_nodd:   lda joystick
                beq movplr_notdrunk
                and #JOY_UP+JOY_DOWN+JOY_FIRE
                ora drunkrnd
                sta joystick
movplr_notdrunk: rts

movplr_run:   jsr movplr_drunk
                lda shpbits,x           ;Player jumping?
                and #$01
                beq movplr_nofly
                jmp movplr_fly
movplr_nofly: jsr getcharpos            ;Get char position
                jsr getcharinfo
                tay
                and #CI_WALK            ;Feet under ground?
                bne movplr_nofall       ;If not, fall
                lda #2
                sta shpsy,x
                jmp movplr_fall
movplr_nofall: lda joystick            ;Ducking?
                and #JOY_DOWN
                beq movplr_noduck
                tya                      ;Ladder?
                and #CI_CLIMB
                bne movplr_initclimb
                lda #SHP_PLRDUCK
                sta shpt,x

```

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.