::::::::::::::::::::::

:rem tron 3.0 :::

:rem by f.g.huerta :::

::::::::::::::::::::::

:rem ms->screen memory (normally 7680)

:rem mc->color memory (normally 38400)

:rem cm->character memory (normally 32768)

1 ms=peek(648):mc=37888+256\*(ms and 2):ms=ms\*256:cm=32768:poke36879,16

::::::::::::::::::::::

:rem music :::::::::::

:rem m1,m2,m3 music chanels

:rem n%(8) -> melodie

:rem max volumen

3 dim n%(8):m1=36874:m2=36875:m3=36876:ni=0:poke 36878,15

::::::::::::::::::::::

:rem players :::::::::

:rem p%(i, data) -> data player array

:rem i -> player id

:rem p%(i, 0)-> x pos

:rem p%(i, 1)-> y pos

:rem p%(i, 2)-> ix

:rem p%(i, 3)-> iy

:rem p%(i, 4)-> color

:rem p%(i, 5)-> status: 0->ok, 1->crashed

:rem p%(i, 6)-> current q

:rem p%(i, 7)-> last q

:rem p%(i, 8)-> last ix

:rem p%(i, 9)-> last iy

:rem p%(i,10)-> num crashes

:rem v ->velocity

:rem b ->background sprite 46

:rem nc -> num cols

:rem nr -> num rows

4 dim p%(1,10):v=1:nc=22:nr=22:b=102

:rem sprites

:rem s(0) -> up down 93

:rem s(1) -> up right 112

:rem s(2) -> up left 110

:rem s(3) -> left right 64

:rem s(4) -> down right 109

:rem s(5) -> down left 125

5 s0=93:s1=112:s2=110:s3=64:s4=109:s5=125

:rem 52 s$=chr$(93)+chr$(112)+chr$(110)+chr$(64)+chr$(109)+chr$(125)

:rem chr$(158) -> yellow

:rem chr$(30) -> green

:rem chr$(31) -> blue

::::::::::::::::::::::

:rem new game entry point

::::::::::::::::::::::

:rem reset scores

10 p%(0,10)=0:p%(1,10)=0

::::::::::::::::::::::

:rem new match entry point

::::::::::::::::::::::

:rem init melodie

11 for i=0 to 8:read n%(i):next

:rem init players

12 for i=0 to 1: for d=0 to 9: read p%(i,d): next:next

:rem clear screen cyan chr$(159): white 5: black 144

13 k$="":for i=1 to nc:k$=k$+chr$(166):next i

14 ?chr$(147)chr$(28):for i=0 to nr-1:?k$;:next

:rem paint scores

15 gosub 140

::::::::::::::::::::::

:::rem begin main loop

::::::::::::::::::::::

: rem debug - print main loop miliseconds

rem 20 ?chr$(19)+str$(timer-t)+" ";:t=timer

: rem match finished?

20 if p%(0,5) or p%(1,5) then 100

: rem game finished?

22 if p%(0,10)=9 or p%(1,10)=9 then 120

: rem store old velocities

23 p%(0,8)=p%(0,2):p%(0,9)=p%(0,3):p%(1,8)=p%(1,2):p%(1,9)=p%(1,3)

: rem read keyboard in any key

24 if peek(198)=0 then 30

:::::::::::: rem exit? 'x'

25 get k$:if k$="x" then 120

:::::::::rem player ia

26 if k$="a" then p%(0,2)=-v:p%(0,3)=0:goto 30

27 if k$="d" then p%(0,2)=v:p%(0,3)=0:goto 30

28 if k$="w" then p%(0,3)=-v:p%(0,2)=0:goto 30

29 if k$="s" then p%(0,3)=v:p%(0,2)=0:goto 30

:::::::::::rem computer ia

: rem inminent crash? and little random factor

rem 30 q=(p%(1,1)+p%(1,3))\*nc+p%(1,0)+p%(1,2):i=rnd(1)

30 d1=p%(1,1):d3=p%(1,3):d0=p%(1,0):d2=p%(1,2)

31 q=(d1+d3)\*nc+d0+d2:i=rnd(1)

32 if peek(ms+q)=b and i<.95 then 40

: rem change direction

33 d=v: if i<.5 then d=-d

34 if d2<>0 then q=(d1+d)\*nc+d0:goto 36

35 q=d1\*22+d0+d

: rem direction blocked? change direction again

36 if peek(ms+q)<>b then d=-d

: rem final decision

37 if d2<>0 then p%(1,2)=0:p%(1,3)=d: goto 40

38 p%(1,2)=d:p%(1,3)=0

:::::::::: rem phisics

40 for i=0to1

rem 40 i=0

:::::::::::: rem move - memory optimized

rem 41 d0=p%(i,0):d2=p%(i,2):p%(i,0)=d0+d2+22\*((d0=21andd2=1)-(d0=0andd2=-1))

rem 42 d1=p%(i,1):d3=p%(i,3):p%(i,1)=d1+d3+22\*((d1=21andd3=1)-(d1=0andd3=-1))

::::::::: rem move - speed optimized

41 d0=p%(i,0)+p%(i,2):d1=p%(i,1)+p%(i,3):p%(i,0)=d0:p%(i,1)=d1

42 if d0<0 then p%(i,0)=21

43 if d0>21 then p%(i,0)=0

44 if d1<0 then p%(i,1)=21

45 if d1>21 then p%(i,1)=0

46 next

rem 46 if i=0 then i=1:goto 41

:::::rem check crashes

:rem draw?

50 if p%(0,0)=p%(1,0) and p%(0,1)=p%(1,1) then p%(0,5)=1:p%(1,5)=1:goto 100

:rem crash?

51 for i=0 to 1

:rem if crashes ->

:rem - update status

:rem - increment crashes

52 p%(i,7)=p%(i,6):p%(i,6)=p%(i,1)\*nc+p%(i,0)

53 if peek(ms+p%(i,6))<>b then p%(i,5)=1: p%(i,10)=p%(i,10)+1

54 next

:::::::::::::rem paint

60 for i=0 to 1

:::::::::::::rem tail

:: rem get tail char

61 d2=p%(i,2):if d2 and d2=p%(i,8) then d=s3:goto 67

62 d3=p%(i,3):if d3 and d3=p%(i,9) then d=s0:goto 67

63 if (p%(i,9)=-v and p%(i,2)=-v) or (p%(i,8)= v and p%(i,3)= v) then d=s2:goto 67

64 if (p%(i,9)=-v and p%(i,2)= v) or (p%(i,8)=-v and p%(i,3)= v) then d=s1:goto 67

65 if (p%(i,9)= v and p%(i,2)=-v) or (p%(i,8)= v and p%(i,3)=-v) then d=s5:goto 67

66 if (p%(i,9)= v and p%(i,2)= v) or (p%(i,8)=-v and p%(i,3)=-v) then d=s4:goto 67

:: rem poke tail & head

67 d7=p%(i,7):poke ms+d7,d:poke mc+d7,p%(i,4):d6=p%(i,6):poke ms+d6,90:poke mc+d6,0

68 next

: rem play music

69 gosub 85

:::::rem end main loop

70 goto 20

::::::::::::::::::::::

:rem music :::::::::::

:rem prepare next note

85 ni=ni+1:if ni>8 then ni=0:poke m1,0

:rem play a note :::::

86 poke m1,n%(ni):poke m2,n%(ni):poke m3,n%(ni)

87 return

:::::::: rem music off

88 poke m1,0:poke m2,0:poke m3,0:ni=0

89 return

::::::::::::::::::::::

:::::::::rem end match

::::::::::::::::::::::

::::::::rem select msg

100 if p%(0,5)=1 then msg$="human crash!":q=p%(0,1)\*nc+p%(0,0):poke mc+q,2

101 if p%(1,5)=1 then msg$="computer crash!":q=p%(1,1)\*nc+p%(1,0):poke mc+q,2

102 if p%(0,5)+p%(1,5)=2 then msg$="draw!"

:::::::::rem music off

103 gosub 88

::::::::rem earthquake

104 for n=1 to 10:for m=10 to 14

105 poke 36864,m:poke 36877,240-m

106 next m: next n

107 poke 36877,0

:::::::::rem show info

110 k$=chr$(17):?chr$(19)k$k$k$k$k$k$k$k$k$k$k$k$

111 ?chr$(28)" "msg$:?" any key continue ";

::rem flush key buffer

112 get k$: if k$<>"" then 112

::::::::::::::rem wait

113 for i=1 to 1000: next

:rem read keyboard

115 get k$: if k$="" then 115

:::::::::::::rem exit?

116 if k$="x" then 120

::rem start new match

117 restore

118 goto 11

::::::::::::::::::::::

:::::: rem exit ::::::

::::::::::::::::::::::

: rem music off

120 gosub 88

: rem score screen

121 ?chr$(144)chr$(147):

122 ?"final score"

123 ?"----------------------"

124 ?"human : ";p%(1,10)

125 ?"computer : ";p%(0,10)

126 if p%(0,10)=p%(1,10) then msg$="draw!":goto 129

127 if p%(0,10)<p%(1,10) then msg$="human wins!":goto 129

128 msg$="computer wins!"

129 ?msg$

:::::: new game? :::::

130 ?:?"play again (y/n)?"

: rem flush key buffer

131 get k$: if k$<>"" then 131

::::::::::::: rem wait

132 for i=1 to 1000: next

133 get k$: if k$="" then 133

134 if k$="y" then restore:goto 10

135 ?"bye.":end

:::::::rem paint score

140 ?chr$(19);:for i=1 to nc:?chr$(17);:next:d0=p%(0,10):d1=p%(1,10)

141 ?chr$(31)" human:"chr$(158)str$(d1)chr$(31)" computer:"chr$(30)str$(d0);

::::::: rem background score

142 for i=0 to 1

143 for ir=0 to 7:d=peek(cm+(48+p%(1-i,10))\*8+ir)

144 for ic=7 to 0 step -1

145 d=d/2:d0=0:if d=int(d) then d0=p%(i,4)

146 d=int(d):poke mc+(ir+1)\*nc+ic+2+10\*i,d0

147 next:next:next

148 return

::::::::::::::::::::::

::::: rem data :::::::

::::::::::::::::::::::

: rem melody

200 data 215,201,187, 215,201,195, 215,201,187

: rem initial players values

210 data 5,10,1,0,7,0,225,224,0,1

220 data 16,10,-1,0,5,0,236,237,0,1

From: http://blog.nanomuelle.com/2012/03/08/tron-3-0/