Cargo Shuffle

A PICO-8 adaptation of a ZX Spectrum type-in

In 1983, the ZX Spectrum from Sinclair Research was only a year old, but publications were already continuing the well-established tradition of providing free software in print. In an August issue of that year, *Popular Computing Weekly* published one such type-in under the name **Barrels and Ladders**. Written by a David Millington, it had all the classic staples of a Spectrum BASIC listing, including character graphics and a little machine code. Now, over 40 years later, an adaptation is presented here for the PICO-8 fantasy console. The goal of the game is to bring the red barrels from the various platforms down to the blue loading dock on the bottom right of the play field; all while avoiding the little green creatures crisscrossing to and fro.

```
--cargo shuffle (2025) by amatkins
--based on barrels and ladders (1983) by david millington
poke(0x5f36,4)
poke(0x5600,unpack(split"8,8,10"))
poke(0x5708,unpack(split"4,4,12,12,12,0,12,0"))
poke(0x5778,unpack(split"32,48,24,12,6,3,1,0,63,33,33,49,4
9,49,63,0,8,8,8,12,12,12,12,0,63,33,32,63,3,3,63,0,31,17,16
,62,48,49,63,0,31,17,17,17,63,24,24,0,63,1,1,63,48,49,63,0,
63,33,1,63,49,49,63,0,63,32,32,48,48,48,48,0,30,18,18,63,49
,49,63,0,63,33,33,63,48,48,48,0,0,0,12,0,0,12,0,0"))
poke(0x57f8,unpack(split"62,32,32,62,6,0,6,0"))
poke(0x5908,unpack(split"30,18,18,63,35,35,35,0,31,17,17,6
3,35,35,63,0,63,33,1,3,3,35,63,0,63,33,33,35,35,35,63,0,63,
1,1,63,3,3,63,0,63,1,1,63,3,3,3,0,63,33,1,51,35,35,63,0,33,
33,33,63,35,35,35,0,4,4,4,12,12,12,12,0,16,16,16,48,48,49,6
3,0,17,17,17,63,35,35,35,0,1,1,1,3,3,3,63,0,63,41,41,43,43,
43,43,0,63,33,33,35,35,35,35,0,63,49,33,33,33,33,63,0,63,33
,33,63,3,3,3,0,63,33,33,33,33,57,63,0,31,17,17,63,35,35,35,
0,63,33,1,63,48,49,63,0,63,4,4,12,12,12,12,0,33,33,33,35,35
,35,63,0,35,35,35,51,18,18,30,0,41,41,41,43,43,43,63,0,33,3
3,33,30,35,35,35,0,33,33,33,63,12,12,12,0,63,33,32,63,3,35,
63,0"))
poke(0x5a00,unpack(split"255,255,255,255,255,255,255,255,2
51,251,251,0,223,223,223,0,54,36,126,153,126,28,38,96,189,1
89,90,60,24,56,100,6,17,68,17,68,17,68,17,0,4,12,124,62,31,
24,16,0,60,66,255,129,255,129,66,60,34,119,127,127,62,28,8,
0,42,28,54,119,54,28,42,0,28,28,62,93,28,20,20,0,8,28,62,12
7,62,42,58,0,60,60,25,126,152,28,38,96,108,36,126,153,126,5
6,100,6,24,120,8,8,8,15,7,0,62,99,107,99,62,65,62,0,8,20,42
,93,42,20,8,0,0,0,0,85,0,0,0,0,60,60,152,126,25,56,100,6,8,
28,127,28,54,34,0,0,127,34,20,8,20,34,127,0,189,189,90,60,2
4,28,38,96,0,10,4,0,80,32,0,0,17,42,68,0,17,42,68,0,62,107,
119,107,62,65,62,0,127,0,127,0,127,0,127,0,195,66,126,66,66
,66,126,66"))
poke(0x5f58,129)
poke(0x5f5c,unpack(split"8,0,255,8"))
local scale=split"c1,c#1,d1,d#1,e1,f1,f#1,g1,g#1,a1,a#1,b1
,c2,c#2,d2,d#2,e2,f2,f#2,g2,g#2,a2,a#2,b2"
local ap={[false]={[false]="■",[true]="■"},[true]={[fal
se]="■",[true]="■"}}
local am={[false]="■",[true]="■"}
local lg,lc,x,y,ti
function set_cell(x,y,g,c)
 lg[y][x],lc[y][x]=g or "",c or 7
 print("■",x*8-8,y*8-8,0)
 if(g)print(g,x*8-8,y*8-8,c or 7)
end
::init::
local pf,px,py,pg,pc,li,sc=false,12,14,"",12,3,0
::new::
local st,ba,pb,bg,bc,nx,ny=t(),6,false,"",7,16,14
cls()
--screen contents
lg,lc={},{}
for j=1,16 do
 add(lg,{}) add(lc,{})
 for i=1,16 do add(lg[j],"") add(lc[j],7) end
end
--platforms
for j=3,15,3 do
 for i=1,16 do set_cell(i,j,"▒",13) end
end
local hl=split"7,6,9,9,8,12,7,15"
for i=1,2 do
::new_hole::
 local j=flr(rnd(4))
 x,y=hl[j*2+1],hl[j*2+2]
 if(lc[y][x]==7)goto new_hole
 set_cell(x,y)
end
--loading dock
for i=13,16 do set_cell(i,15,"■",12) end
--ladders
for i,l in pairs(split"4,5,10,5,7,8,13,8,4,11,10,11") do
 if i%2==1 then x=l
 else for j=l,l+3 do set_cell(x,j,"‖",1) end
 end
end
--barrels
local bl=split"1,5,2,5,15,5,16,5,1,8,2,8,15,8,16,8,1,11,2,
11,15,11,16,11,1,14,2,14"
for i=3,28,2 do
 local j=flr(rnd((i+1)\2))*2+1
 bl[i],bl[i+1],bl[j],bl[j+1]=bl[j],bl[j+1],bl[i],bl[i+1]
end
for i,b in pairs(bl) do
 if i%2==1 then x=b
 else
  if(i<=12)set_cell(x,b,"●",8) else set_cell(x,b,"",11)
 end
end
--enemies
local mn,mt={},0
for y=5,14,3 do
 repeat x=flr(rnd(12))+3
 until lc[y][x]==7 and (y~=14 or x<=8)
 mn[y]={x,y~=14 and rnd()<0.5,rnd()<0.5,"",7,false,"",7}
 set_cell(x,y,"■",3)
end
--player
set_cell(px,py,"웃")
--lives
if(li<3)li+=1
for i=1,li do set_cell(i,2,"♥",8) end
print("ｘ:99",0,0,6)
print("●:"..sc,64,0)
::ready::
flip()
print("GET READY!",24,120,t()*5%5+8)
if(t()-st<1.5)goto ready
print("\#G■",0,120,0)
st=t()
::update::
flip()
ti=99-flr(t()-st)
print("\#3■\^J4O\Fb"..ti,16,0,0)
if(ti<=0)goto damage
--update monsters
mt=(mt+1)%8
for y,m in pairs(mn) do
 local x,d,f,g1,c1,b,g2,c2=unpack(m)
 -- pickup barrel
 if(x~=px or y~=py-1 or not pb or b)goto move_enemy
 ?"\A58X5A3.FA3.FA2."
 if(bc~=3)g1,c1=bg,bc
 m[6],b,pb,bg,bc=true,true,false,"",7
 print("●",x*8-8,y*8-16,8)
::move_enemy::
 if(mt~=0)goto next_enemy
 set_cell(x,y,g1,c1)
 if(b)set_cell(x,y-1,g2,c2)
 x+=(d and 1 or -1)
 if(x==1 or (y==14 and 8 or 16)==x)d=not d
 if(lc[y][x]~=11 or not b)goto update_enemy
 set_cell(x,y,"●",8)
 b,g2,c2=false,"",7
 m[6],m[7],m[8]=b,g2,c2
::update_enemy::
 m[1],m[2],m[3],m[4],m[5]=x,d,not f,lg[y][x],lc[y][x]
 if(b)m[7],m[8]=lg[y-1][x],lc[y-1][x]
 set_cell(x,y,am[f],3)
 if(b)print("●",x*8-8,y*8-16,8)
::next_enemy::
```

```
 end
 --update player
 x,y=px,py
 if(lc[y][x]==3)goto damage
 if(btnp(⬅️)and (lc[y+1][x-1]==1 or lc[y+1][x-
1]==13))x=max(x-1,1)
 if(btnp(➡️)and (lc[y+1][x+1]==1 or lc[y+1][x+1]==13))x=min
(x+1,16)
 if(btnp(⬆️)and lc[y-1][x]==1)y=max(y-1,4)
 if(btnp(⬇️)and lc[y+1][x]==1)y=min(y+1,14)
 if(lc[y+1][x]~=3 or pb or not mn[y+1][6])goto move_player
 ?"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
 pb,bg,bc=true,lg[y-1][x],lc[y-1][x]
 set_cell(x,y,mn[y+1][7],mn[y+1][8])
 mn[y+1][6],mn[y+1][7],mn[y+1][8]=false,"",7
 print("🅾️",px*8-8,py*8-16,8)
::move_player::
 if(x==px and y==py)goto update
 set_cell(px,py,pg,pc)
 if(pb)set_cell(px,py-1,bg,bc)
 pf,px,py,pg,pc=not pf,x,y,lg[y][x],lc[y][x]
 if(pb)bg,bc=lg[y-1][x],lc[y-1][x]
 if(lc[y][x]==3)goto damage
 ?"▒▒▒▒▒▒▒▒"..scale[25-y+5-x].."▒▒"..scale[25-y+5-x]
 --pick up barrel
 if(pc~=8 or pb)goto place_barrel
 ?"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
 pb,pg,pc=true,"",11
::place_barrel::
 if(pc~=12 or not pb)goto draw_player
 ?"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
 pb,ba,sc=false,ba-1,sc+1
 print("▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"..sc,80,0,0)
 set_cell(nx,ny,"🅾️",14)
 ny-=1 if(ny==12)nx,ny=nx-1,14
 if(ba==0)goto win_stage
::draw_player::
 set_cell(px,py,ap[pb][pf])
 if(pb)print("🅾️",px*8-8,py*8-16,8)
 goto update
::damage::
 ?"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
 for i=1,30 do
  set_cell(px,py,ap[pb][pf],t()*5%2+7)
  flip()
 end
 set_cell(li,2,"▒▒",0)
 li,x=li-1,0
 if(li==0)goto gameover
 if(pc~=3)mn[py][4],mn[py][5]=pg,pc
 if(not pb)goto replace_player
 if(ti<=0 or mn[py][6])goto lose_barrel
 mn[py][6],mn[py][7],mn[py][8]=true,bg,bc
 pb,bg,bc=false,"",7
 goto replace_player
::lose_barrel::
 set_cell(px,py-1,bg,bc)
 pb,bg,bc=false,"",7
 x=px-1 while lc[py][x]~=7 and lc[py][x]~=11 do x-=1 end
 set_cell(x,py,"🅾️",8)
::replace_player::
 px,py,pg,pc=12,14,"",12
 set_cell(px,py,ap[false][pf])
 goto update
::win_stage::
 for i=1,20 do flip() end
 x=0
 for i=ti,0,-1 do
  ?"▒▒▒▒▒▒▒▒"
  x+=1 if(x==12)sc,x=sc+1,0
  print("▒▒▒▒▒▒▒▒",0,0,0)
  print("❎:"..i,0,0,6)
  print("🅾️:"..sc,64,0)
  set_cell(px,py,"⬇️",t()*5%5+8)
  flip()
 end
 goto new
::gameover::
 print("TRY AGAIN? ❎/🅾️",8,120,t()%1<0.5 and 7 or 5)
 flip()
 if btnp(❎) then x=1
 elseif btnp(🅾️) then x=2
 end
 if(x==0)goto gameover
 if(x==1)goto init
 cls()
```

This program uses 1630 tokens (or about 17% compressed space), and at 6672 characters long, it's a bit of a lengthy text. The code is structured in a fashion similar to that of a BASIC program with goto jumps for flow control, color attributes for collision detection, and user-defined characters for graphics. If you're new to typing in PICO-8, the special graphical characters can be typed using shift and the letter keys. While inspired by the original listing, this version of the game has been heavily modified to be suitable for the different system. Regardless, we hope you'll enjoy typing and playing!