```
!-------------------------------------------------------
!- WR-Ball!
!- 2018, Zoltan Szoke
!-------------------------------------------------------

!- Compact list
0reade$,f$,z,d,p,a$,a,b$,c$,d$:dimb(p),n(p),l(p):w=7:h=9:deffna(u)=peek(u)and15:deffnp(i)=-(fna(a)=c):fori=0to15step3:pokea+i,62:next
1pokez+1,0:e=1318:w=w+1:?a$w-7;"{home}{down*3}"e$:s=.:r=.:t=.:fori=0tow:forj=0tow:a=e+i*p+j:pokea,160:poked+a,rnd(1)*(w/2-1)
+5:next:next:o=o+1
2pokez,0:f=198:pokef,.:?b$t0"+"stab(20)"{light gray}hi sc:"h:pokez-11,1:printmid$(c$,1,w+5)tab(20)"{white}O
{reverse off}"o:pokez+7,15:pokez-32,x*8+135:pokez-31,y*8+106
3poke2040,11:waitf,1:getk$:k=asc(k$):y=y-(k=17andy<w)+(k=145andy>.):u=d+e+x+y*p:on-(k<49ork>54oro=.)goto4:pokeu,k-44:o=o-1:goto2
4on-(k=13)goto8:x=x-(k=29andx<w)+(k=157andx>.):on-(k<>32)goto2:pokez+7,2:b=.:n=.:b(.)=u:n(.)=u:c=fna(u):on-(c=.)goto2:gosub7
5fori=0ton:v=n(i):l=.:forj=0tow:l(l)=peek(v):l=l-(fna(v)>0):v=v-p:j=j-p*(v<d+e):next:v=n(i):fork=0toj-p:pokev,-l(k)*(k<l):v=v-p
6next:next:s=s-(n-1)*n*(n>1):r=t/(w+1)/(w+1)*100:?f$tab(19)int(r)" % ":goto2:data"{down*11}   {reverse on}{yellow}{160}{127}{down}
{left}{162}{reverse off}{orange}Q{left*3}Q{up}{right}{light gray}N{up}N{up}N{light green}{165}{down}{left}T{down}{left}{red}Q"
7q=b(b):a=q-1:onfnp(.)gosub9:a=q+1:onfnp(.)gosub9:a=q+p:onfnp(.)gosub9:a=q-p:onfnp(.)gosub9:b=b-1:on-(b>-1)goto7:n=n-
1:return:data"{reverse off}{home}{yellow}"
8s=int(s*r/100):t0=t0+s:on-(w<12andr>89)goto1:g=h>t0:h=-h*g+t0*(g+1):printd$t0:waitf,1:w=7:t0=.:m=.:o=.:x=.:y=.:goto1:data53280,54272
9pokea,0:t=t+1:n(n)=a:n=n+1:b(b)=a:b=b+1:return:data40,"{black}{clear}{red}{reverse off}  site",704,"{home}{down}  {yellow}sc:","{home}
{down*2}  {reverse on}{green}1{blue}2{yellow}3{orange}4{brown}5","{clear}final score: "


!- Detailed code

0reade$,f$,z,d,p,a$,a,b$,c$,d$:dimb(p),n(p),l(p):w=7:h=9:deffna(u)=peek(u)and15:deffnp(i)=-(fna(a)=c):fori=0to15step3:pokea+i,62:next
!-------------------------------------------------------
!- Line 0 - Game initialization
!-------------------------------------------------------
!-  Read values: reade$,f$,p,z,d,a$,a,b$,c$,d$
!-      e$                              wrecking ball
!-      f$                              yellow color for line 6 length optimization
!-      z                               53280 - setting colors and sprites. Optimized for line 1 and 2, therefore not 53248.
!-      d                               54272 - offset for color table from screen codes
!-      p                               const for 40 - sparing 1 character every time
!-      a$                              clear screen, text 'site' in red colors
!-      a                               704, sprite starting
!-      b$                              score in yellow - line optim.
!-      c$                              color codes for repainting - keys 1-5
!-      d$                              text for final score

!-  Arrays: dimb(p),n(p),l(p)
!-      b(...)                          'fill' method buffer for recursion, temporary
!-      n(...)                          buffer for color matches, results
!-      l(...)                          temporary buffer for gravitation
!-  Variable setting: w=7:h=9
!-      w                               starting with and height is 7+1+1
!-      h                               starting high score is 9. Easy to beat...
!-  Definitions:
```

```
!-      deffna(u)=peek(u)and15          color code for pixel u
!-      deffnp(i)=-(fna(a)=c)           checks if color is c, used for on gosub jumps (no real parameters)
!-  Sprite 'cursor' definition: 001111110 every line.
!-      fori=0to15step3:pokea+i,62:next

1pokez+1,0:e=1318:w=w+1:?a$w-7;"{home}{down*3}"e$:s=.:r=.:t=.:fori=0tow:forj=0tow:a=e+i*p+j:pokea,160:poked+a,rnd(1)*(w/2-1)
+5:next:next:o=o+1
!-------------------------------------------------------
!-  Line 1: New construction site initialization
!-------------------------------------------------------
!-      pokez+1,0                       set border and background color for black -
 not in line 0 because of character limit optimalization, not speed
!-      e=1318                          constant for start of game area screen code
!-      w=w+1:                          increase width every level
!-      ?a$w-7;...e$                    print "site "(8-7, increasing every level), and add the wrecking ball (e$)
!-      s=.                             reset site (level) score
!-      r=.                             contract fulfillment percentage
!-      t=.                             number of 'pixels' cleared
!-
!-      fori=0tow:forj=0tow:a=e+i*p+j:pokea,160:poked+a,rnd(1)*(w/2-1)+5:next:next
!-                                      Draw game area (w+1*w+1), with random colored blocks
!-
!-      o=o+1                           add one 'pixel' repaint every level

2pokez,0:f=198:pokef,.:?b$t0"+"stab(20)"{light gray}hi sc:"h:pokez-11,1:printmid$(c$,1,w+5)tab(20)"{white}O
{reverse off}"o:pokez+7,15:pokez-32,x*8+135:pokez-31,y*8+106
!-------------------------------------------------------
!-  Line 2: Main status (and a few things missing from line 0 and 1)
!-------------------------------------------------------
!-      pokez,0                         set border and background color for black -
 not in line 0 because of character limit optimalization, not speed
!-      f=198                           keyboard buffer - used for WAITing for a key, and clear buffer (pokef,.)
!-      pokef,.                         clear keyboard buffer (enter is a dangerous key, don't press it twice)
!-      ?b$                             score text
!-      t0"+"s                          show score from previous sites and current site score
!-      tab(20)"{light gray}hi sc:"h    show high score
!-      pokez-11,1                      show cursor
!-      printmid$(c$,1,w+5)             color list, but only colors used on current site
!-      tab(20)"{white}O{reverse off}"o number of repaints left
!-      pokez+7,15                      change cursor color to white
!-      pokez-32,x*8+103                set cursor x coordinate
!-      pokez-31,y*8+82                 set cursor y coordinate

3poke2040,11:waitf,1:getk$:k=asc(k$):y=y-(k=17andy<w)+(k=145andy>.):u=d+e+x+y*p:on-(k<49ork>54oro=.)goto4:pokeu,k-44:o=o-1:goto2
!-------------------------------------------------------
!- Line 3: vertical cursor movement and repaint with keys 1-5
!-------------------------------------------------------
!-      poke2040,11:                    cursor shape 11*64=704
!-      waitf,1:getk$:k=asc(k$)         wait for one key, k$ char and k ascii code
```

```
!-      y=y-(k=17andy<w)+(k=145andy>0)  handle k=17 and k=145 (up and down key) if there are still rows to move
!-      u=d+e+x+y*p:                   current pixel poke code
!-      on-(k<49ork>54oro=0)goto4      go to next line if keypress is not between "1" and "5"
!-      pokeu,k-44:o=o-1:goto2         repaint current pixel (u), and decrease repaint count

4on-(k=13)goto8:x=x-(k=29andx<w)+(k=157andx>.):on-(k<>32)goto2:pokez+7,2:b=.:n=.:b(.)=u:n(.)=u:c=fna(u):on-(c=.)goto2:gosub7
!-------------------------------------------------------
!- Line 4: horizontal cursor movement, call game over/next level/wrecking ball
!-------------------------------------------------------
!-      on-(k=13)goto8                 if enter key pressed, go to line 8
!-      x=x-(k=29andx<w)+(k=157andx>0) handle k=29 and k=157 (left and right key) if there are still columns to move
!-      on-(k<>32)goto2                if it's not a space, then a key without function pressed, go to line 2
!-      pokez+7,2                      the cursor is red, demolition started
!-      b=.:n=.                        b and n: index for b(...) and n(...) arrays
!-      b(0)=u:n(0)=u                  first element of each buffer is the current pixel
!-      c=fna(u)                       get color of current pixel
!-      on-(c=0)goto2                  if color is black, no wrecking ball
!-      gosub7                         call wrecking subroutine

5fori=0ton:v=n(i):l=.:forj=0tow:l(l)=peek(v):l=l-(fna(v)>0):v=v-p:j=j-p*(v<d+e):next:v=n(i):fork=0toj-p:pokev,-l(k)*(k<l):v=v-p
!-------------------------------------------------------
!- Line 5-6: handle result of wrecking ball subroutine (line 7), plus wrecking ball image
!-------------------------------------------------------
!-      fori=0ton                      handle all (n) results
!-        v=n(i)                       current result item screen peek code
!-        l=.                          count for temporary buffer l(...)
!-
!-        forj=0tow                    loop until v<d+e (within game area)
!-              l(l)=peek(v)           current pixel
!-              l=l-(fna(v)>0)         increase l if color is not black (there is a pixel to move)
!-              v=v-p                  move up (decrease v peek value by 40)
!-              j=j-p*(v<d+e)          add 40 (end loop) if v<d+e
!-        next                         result: l array with every non-black value
!-
!-        v=n(i)                       restart loop
!-
!-        fork=0toj-p                  repaint every pixel above v
!-              pokev,-l(k)*(k<l)      put color from buffer
!-              v=v-p                  move screen code pointer (v) up (-40)
!-        next
!-      next

6next:next:s=s-(n-1)*n*(n>1):r=t/(w+1)/(w+1)*100:?f$tab(19)int(r)" % ":goto2:data"{down*11}   {reverse on}{yellow}{160}{127}{down}
{left}{162}{reverse off}{orange}Q{left*3}Q{up}{right}{light gray}N{up}N{up}N{light green}{165}{down}{left}T{down}{left}{red}Q"
!-------------------------------------------------------
!- Line 6: add score, show progress
!-------------------------------------------------------
!-      s=s-(n-1)*n*(n>1)              score: 0 if 2, 2*1 if 3 blocks, 3*2 if 4 blocks etc...
!-      r=t/(w+1)/(w+1)*100            calculate progress - pixels cleared / total starting pixels
```

```
!-      ?f$                             yellow color
!-      tab(19)int(r)" % "              show progress
!-      goto2                           restart loop
!-      data...                         wrecking ball

7q=b(b):a=q-1:onfnp(.)gosub9:a=q+1:onfnp(.)gosub9:a=q+p:onfnp(.)gosub9:a=q-p:onfnp(.)gosub9:b=b-1:on-(b>-1)goto7:n=n-
1:return:data"{reverse off}{home}{yellow}"
!-------------------------------------------------------
!- Line 7: add score, show progress
!-------------------------------------------------------
!-      q=b(b)                          get actual value from b(...) buffer
!-      a=q-1:onfnp(.)gosub9            check value on the left, if color equals, call line 9. (no such thing as on(condition)
return which would be great in line 9)
!-      a=q+1:onfnp(.)gosub9            check right
!-      a=q+p:onfnp(.)gosub9            check downwards
!-      a=q-p:onfnp(.)gosub9            check upwards
!-      b=b-1                           value finished, remove from b(...) buffer
!-      on-(b>-1)goto7                  loop while there are values left in the buffer
!-      n=n-1:return                    decrease result buffer length
!-      data...                         yellow text

8s=int(s*r/100):t0=t0+s:on-(w<12andr>89)goto1:g=h>t0:h=-h*g+t0*(g+1):printd$t0:waitf,1:w=7:t0=.:m=.:o=.:x=.:y=.:goto1:data53280,54272
!-------------------------------------------------------
!- Line 8: enter pressed, game over / next level
!-------------------------------------------------------
!-      s=int(s*r/100)                  s is corrected by site progress
!-      t0=t0+s                         'payment' received, added to accumulated score
!-      on-(w<12andr>89)goto1           if it's before level5 (w<12) and progress>89%, show next site
!-      g=h>t0                          s it a high score?
!-      h=-h*g+t0*(g+1)                 calculate new high score, h=max(h,t0)
!-      printd$t0                       show score
!-      waitf,1                         wait for a key
!-      w=7:t0=.:m=.:o=.:x=.:y=.:goto1  reset all values, restart game
!-      data...                         data for line 0 variables

9pokea,0:t=t+1:n(n)=a:n=n+1:b(b)=a:b=b+1:return:data40,"{black}{clear}{red}{reverse off}  site",704,"{home}{down}  {yellow}sc:","{home}
{down*2}  {reverse on}{green}1{blue}2{yellow}3{orange}4{brown}5","{clear}final score: "
!-------------------------------------------------------
!- Line 9: wrecking ball subfunction and text data
!-------------------------------------------------------
!-      pokea,0                         clear current pixel
!-      t=t+1                           increase cleared area counter
!-      n(n)=a: n=n+1                   push value to n buffer (result array, not overwritten if pixel handled)
!-      b(b)=a: b=b+1                   push value to b buffer (temporary buffer, overwritten if pixel handled)
!-      return                          return
!-      data...                         data for line 0 variables
```