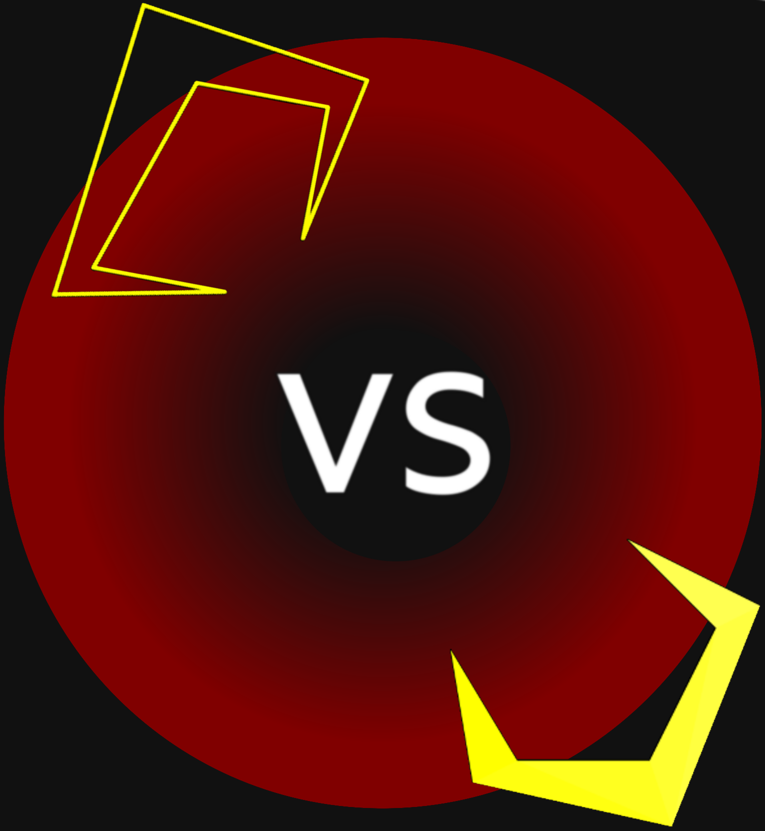
TEMPEST



TEMPEST

THE MAKING AND REMAKING OF ATARI'S ICONIC VIDEOGAME

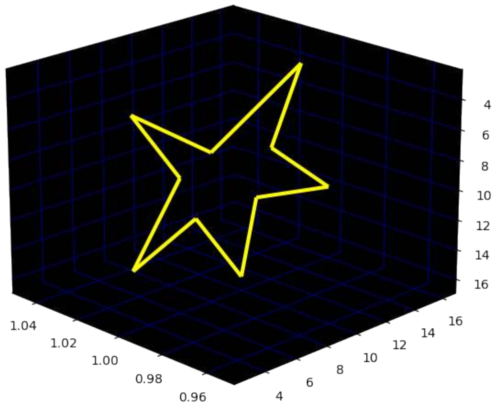
TEMPEST VS TEMPEST

Notes on the Source Code of Two Video Games

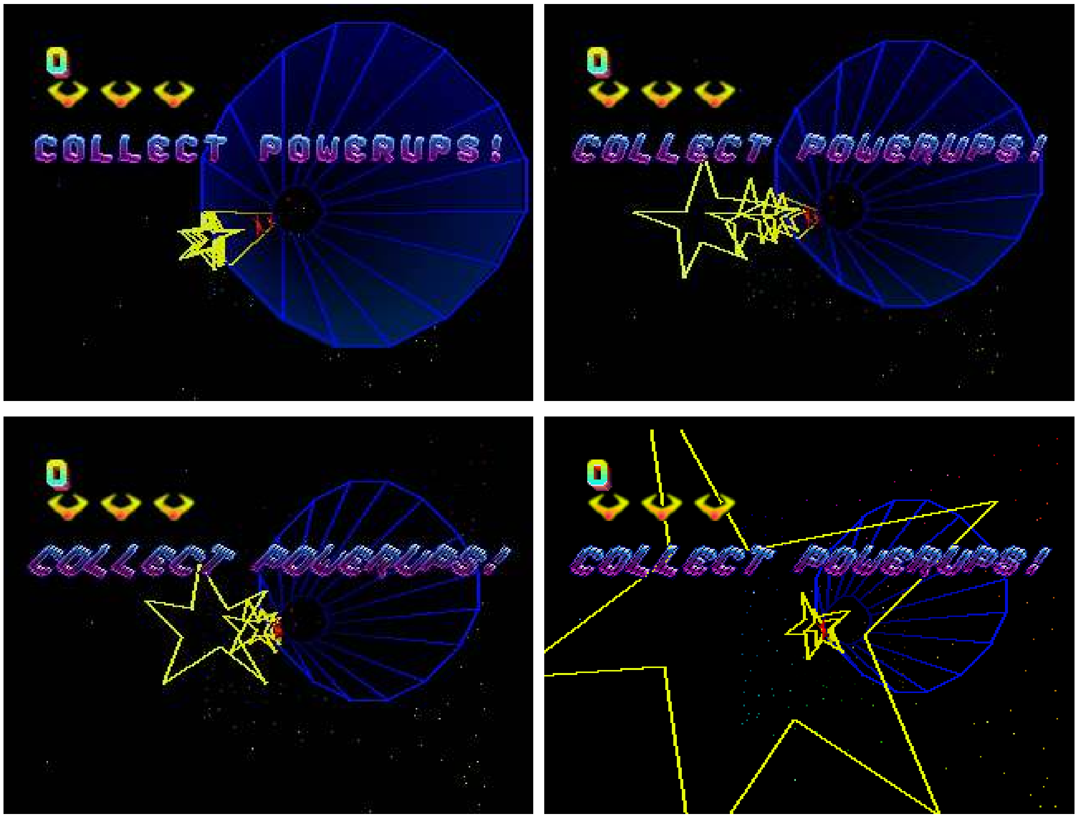
For Edna.

Contents

ouch



```
zap:
  dc.b 3,3,1    ; Co-ordinate 1
  dc.b 2,10,0   ; Connect co-ords 2 & 10
  dc.b 8,7,1    ; Co-ordinate 2
  dc.b 3,0      ; Connect to co-ord 3
  dc.b 14,3,1   ; Co-ordinate 3
  dc.b 4,0      ; Connect to co-ord 4
  dc.b 12,8,1   ; Co-ordinate 4
  dc.b 5,0      ; Connect to co-ord 5
  dc.b 16,12,1  ; Co-ordinate 5
  dc.b 6,0      ; Connect to co-ord 6
  dc.b 11,11,1  ; Co-ordinate 6
  dc.b 7,0      ; Connect to co-ord 7
  dc.b 10,16,1  ; Co-ordinate 7
  dc.b 8,0      ; Connect to co-ord 8
  dc.b 7,11,1   ; Co-ordinate 8
  dc.b 9,0      ; Connect to co-ord 9
  dc.b 3,13,1   ; Co-ordinate 9
  dc.b 10,0     ; Connect to co-ord 10
  dc.b 6,8,1    ; Co-ordinate 10
  dc.b 0,0      ; End of Data
```



```
; ********************************************************************
; ouch
; ********************************************************************
ouch:
  clr l_soltarg           ; Clear the solid background on the web.
  clr.l 20(a0)            ; Stop the claw moving.
  move.l #zap_player,routine ; Set update routine to 'zap_player'.
  move.l _zap,(a0)        ; Set address of the vector object to draw: _zap.
  clr 36(a0)              ; Set X centre to default.
  bsr clzapa              ; Clear more state.
  clr laser_type          ; Clear laser type.
  clr jenable             ; Disable jumping.
  clr bonum               ; Clear power-ups.
  move.l bshotspeed,shotspeed ; Reset shot speed.
  move #-17,38(a0)        ; Set color of object.
  move #3,44(a0)          ; Configure behaviour on rail.
  move #2,34(a0)          ; Draw routine in draw_vex is 'draw_z'.
  bra zapson              ; Set sound effect and return.
```

```
; ********************************************************************
; zap_player
; *******************************************************************
zap_player:
  bsr rightit              ; Remove any 'roll' from the player's viewpoint.
  move.l _claw,a0          ; Point a0 at the claw object..
  move.l _zap,(a0)         ; Set address of the vector object to draw: _zap.
  move.l 4(a0),vp_xtarg    ; Set player's X viewpoint from current X pos.
  move.l 8(a0),vp_ytarg    ; Set player's Y viewpoint from current X pos.
  add #12,28(a0)           ; Roll the object by 12 points.
  sub #$80,36(a0)          ; Increase the speed of the explosion moving nearer.
  sub.l #$11000,vp_z       ; Move the player viewpoint away from the web.
  sub.l #$4000,12(a0)      ; Move the explosion nearer the player.
  bpl vp_xform             ; If not reached player yet, update player viewpoint
    .
  ; Otherwise the explosion has reached the player viewpoint,
  ; so we can switch it off.
  clr 34(a0)               ; Turn off the object's drawroutine.
  bsr zapson               ; Play sound effect.
  bra setsnatch            ; End the player's life.
  ; Returns
```

`draw_z` draws four instances of an object, each one getting closer to the viewpoint of the player.

```
; *******************************************************************
; draw_z
; Draw  with a number of incrementally coloured layers.
; Used for vector objects in the activeobjects list.
; Called during the draw_objects sequence as a member of the draw_vex list.
; *******************************************************************
draw_z:
  bsr draw                 ; Draw the original object
  move 40(a6),-(a7)        ; Save the original color of the object.
  move #2,d0               ; Z images counter
  move 36(a6),d1           ; delta z
  ext.l d1                 ; Extend
  asl.l #8,d1              ; Convert to 16:16
  move 38(a6),d2           ; Delta for colour
  move.l 12(a6),d3         ; Z position

dr_z:
  add d2,40(a6)            ; Add the delta to the colour
  add.l d1,d3              ; Add the z delta to the z position.
  movem.l d0-d3,-(a7)      ; Stash the difference between z positions.

  move.l (a6),a1           ; Get object header
  lea in_buf+4,a0          ; Get GPU buffer
  move.l 4(a6),d0          ; Get the X pos
  sub.l vp_x,d0            ; Subtract the viewpoint
  move.l d0,(a0)+          ; Add to GPU buffer
  move.l 8(a6),d0          ; Get the Y pos
  sub.l vp_y,d0            ; Subtract the viewpoint
  move.l d0,(a0)+          ; Add to GPU buffer
  move.l d3,d0             ; Get the z position
```

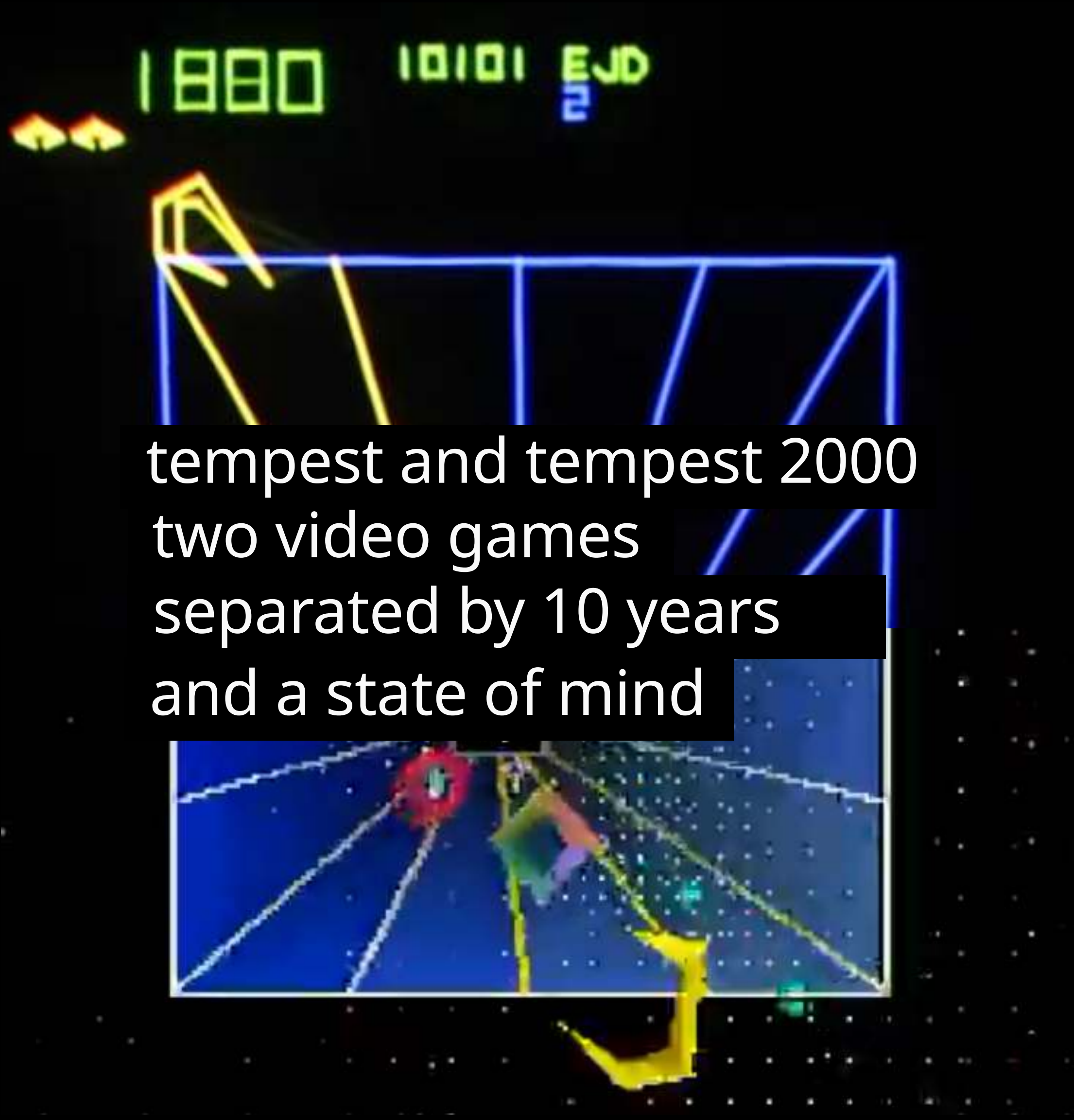
```
   bsr dra                  ; Run the GPU routine to draw the vectors.
   movem.l (a7)+,d0-d3      ; Retrieve the difference between z positions
   dbra d0,dr_z             ; Loop until we're done for all differences.

   move (a7)+,40(a6)        ; Get old colour back
   rts
```

```
; *******************************************************************
; Draw a vector without the above header information.
; *******************************************************************
dra:    sub.l vp_z,d0                 ; Subtract the camera viewpoint.
        move.l d0,(a0)+               ; Add to the GPU buffer.
dragg:  lea fastvector,a2             ; Load the GPU module in 'llama.gas'.
draaa:  move 28(a6),d0                ; Get the XZ orientation
druuu:  and.l #$ff,d0                 ; Only first byte.
        move.l d0,24(a1)              ; Copy to XY orientation
        move 30(a6),d0                ; Get Y rotation of object.
        and.l #$ff,d0                 ; Only first byte.
        move.l d0,28(a1)              ; Copy to XZ orientation
        move 32(a6),d0                ; Get Z rotation
        and.l #$ff,d0                 ; Only first byte.
        move.l d0,32(a1)              ; Copy to YZ orientation

        ; Copy the first 48 bytes of object to GPU buffer.
        move #11,d0                   ; Copy 48 bytes (12 * 4).
xhead:  move.l (a1)+,(a0)+            ; Copy 4 bytes to GPU input ram
        dbra d0,xhead                 ; Keep looping until 48 bytes copied.

        move 40(a6),d0                ; Get the colour
        and.l #$ff,d0                 ; Only first byte
        move.l d0,(a0)+               ; Add to the GPU buffer.
        move 42(a6),d0                ; Get the scale factor.
        ext.l d0                      ; Make it a long.
        move.l d0,scaler              ; Move to scaler.
        move.l a1,oopss+4             ; Stash the updated object.
        move.l (a6),oopss+8           ; Stash the original object.
godraa: move.l #2,gpu_mode            ; Op 2 is vect3d in llama.gas.
        move.l a2,a0                  ; Load the GPU shader in llama.gas.
        jsr gpurun                    ; Run the selected gpu routine: vect3d.
        jsr gpuwait                   ; Wait until finished.
        rts
```



tempest and tempest 2000
two video games
separated by 10 years
and a state of mind