

# Process & Decision Documentation

## Project/Assignment Decisions

So, for this assignment I mostly wanted to work with interaction and hitboxes. Originally my intention was to have my orb grow and shrink based on whether it was being hit but for dramatic effect I decided to just let it keep growing if it was hitting a spike to show festering anxiety. Code wise I made a few changes, aesthetically I changed the environment to more intense colours. For the player I used the blob script instead of a square that becomes more agitated if hit by spikes. The way I got this to work was by first understanding how the object array worked and then I added a hit collider that would trigger different effects such as radius increase and colour change.

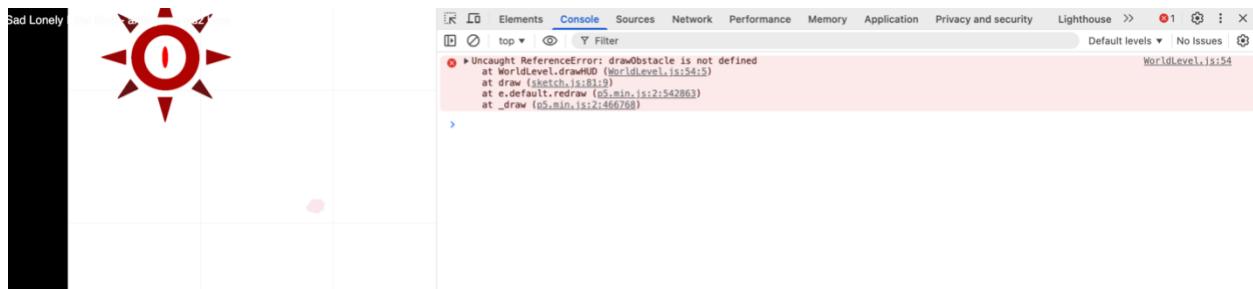
## Role-Based Process Evidence

When I first started with the obstacles I wanted to test how the script worked so I was figuring out hit boxes and how to call correctly to the obstacles array. Hit boxes arguably took the longest amount of time to figure out but I got it to work by checking whether the player was in any of the spike rectangles array (technically it looks like a circle, but the image is a rectangle). I managed to put this together by looking at the initial setup of how the obstacle was made then I grabbed the for loop and added a call to the player so I could grab and edit values as needed.

```
EXPLORER JS Player.js M ENEMY.png U sketch.js M world.json M style.css WorldLevel.js M

✓ OPEN EDITORS 1 unsaved
  Welcome
  jsconfig.json
  index.html
  Player.js
  ENEMY.png assets/images
  sketch.js
  world.json
  # style.css
  WorldLevel.js M

  WorldLevel.js > WorldLevel > drawObstacle
  1   class WorldLevel {
  35    drawHUD(player, camX, camY) {
  44      // ...
  45      // (player.y | 0) +
  46      // " Cam: " +
  47      // ((camX | 0) +
  48      // ", " +
  49      // ((camY | 0),
  50      // 12,
  51      // 40,
  52      // );
  53      fill(0);
  54      drawObstacle(player, this.obstacles);
  55    }
  56
  57    drawObstacle(player, obstacles) {
  58      if (player.x && player.y < obstacles.w * obstacles.h) {
  59        print("hi");
  60        You, 38 seconds ago + Uncommitted changes
  61      }
  62    }
  63  }
  64
```

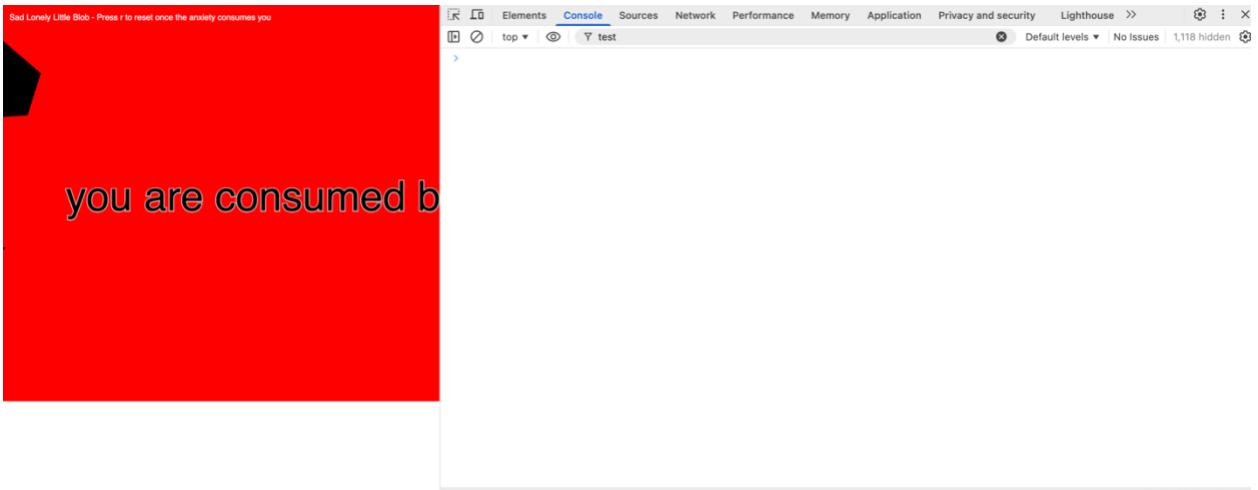


While I was working on my colour, I realized just having it stay one shade of red was boring, so I wanted to make it darker. I initially overcomplicated this process in my head and later found a way to change the colour darker once the threshold was met in one if loop. Down here I was testing to see if my while parameter was going off.

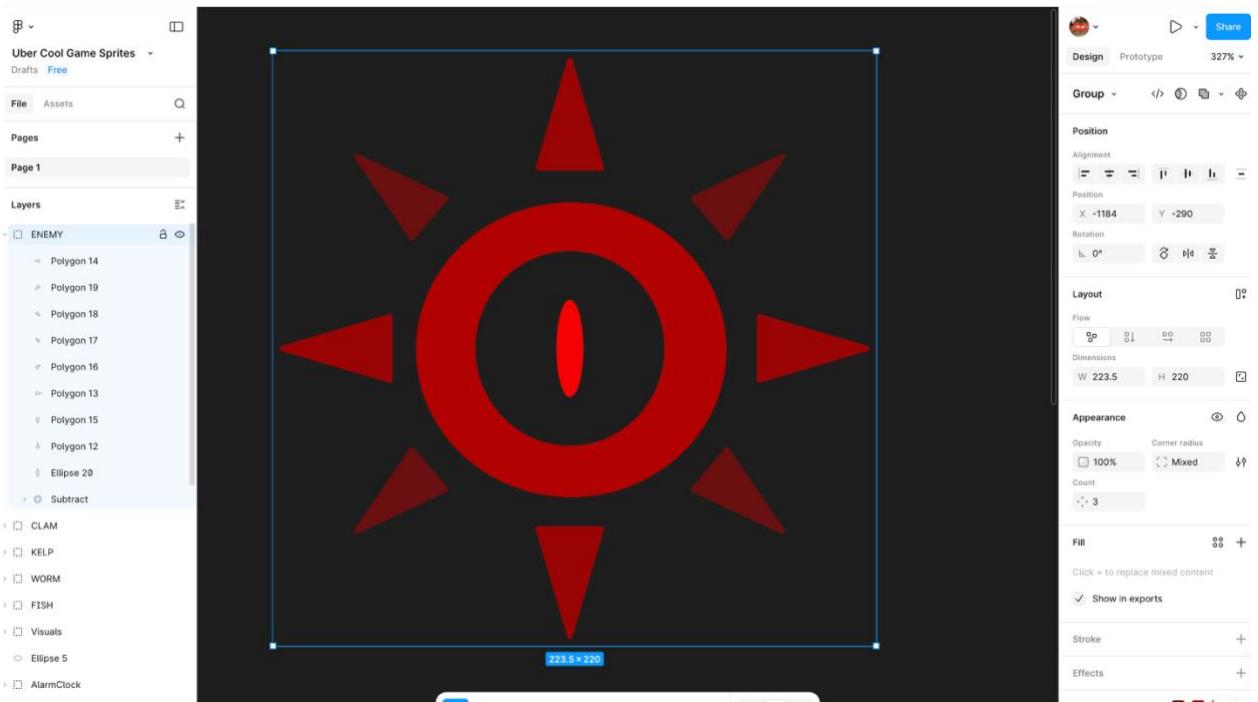
```

EXPLORER
OPEN EDITORS
  Welcome
    jsconfig.json
  index.html
  Player.js
    ENEMY.png assets/images
  sketch.js
    world.json
  # style.css
  WorldLevel.js M
    README.md
AAANTEQUE_SIDEQUEST_WS
  assets
  libraries
  types
  .gitattributes
  index.html
  jsconfig.json
  Player.js
    README.md
  sketch.js
  # style.css
  world.json
  WorldLevel.js M
  OUTLINE
  TIMEINFO

JS Player.js M ENEMY.png JS sketch.js # world.json JS WorldLevel.js M X README.md
  JS WorldLevel.js > WorldLevel > obstacleInteraction
  1 class WorldLevel {
  2   constructor(json) {
  3     this.w = json.world?.w ?? 1200;
  4     this.h = json.world?.h ?? 800;
  5     this.bg = json.world?.bg ?? [235, 235, 235];
  6     this.gridStep = json.world?.gridStep ?? 160;
  7
  8     this.obstacles = json.obstacles ?? [];
  9
 10    // NEW: camera tuning knob from JSON (data-driven)
 11    this.camLerp = json.camera?.lerp ?? 0.12;
 12  }
 13
 14  obstacleInteraction(player) {
 15    let ogRadius = 10;
 16    let ogC = 255;
 17
 18    for (const o of this.obstacles) {
 19
 20      if (player.x > o.x && player.x < o.x + o.w && player.y > o.y && player.y < o.y + o.h) {
 21
 22        player.r++;
 23        player.gc--;
 24        player.bc--;
 25        player.s = 10;
 26        player.wobbleFreq++;
 27        player.wobble++;
 28
 29        if (player.bc == 0 && player.gc == 0) {
 30          while (player.bC <= 17 && player.rC >= 81) {
 31            player.bC++;
 32            player.rC--;
 33            print("test");
 34          }
 35
 36          print("hi");
 37          print(player.r);
 38        }
 39
 40      }
 41    }
 42
 43  }
 44
 45  drawBackground() {
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525
 526
 527
 528
 529
 530
 531
 532
 533
 534
 535
 536
 537
 538
 539
 540
 541
 542
 543
 544
 545
 546
 547
 548
 549
 550
 551
 552
 553
 554
 555
 556
 557
 558
 559
 560
 561
 562
 563
 564
 565
 566
 567
 568
 569
 570
 571
 572
 573
 574
 575
 576
 577
 578
 579
 580
 581
 582
 583
 584
 585
 586
 587
 588
 589
 590
 591
 592
 593
 594
 595
 596
 597
 598
 599
 600
 601
 602
 603
 604
 605
 606
 607
 608
 609
 610
 611
 612
 613
 614
 615
 616
 617
 618
 619
 620
 621
 622
 623
 624
 625
 626
 627
 628
 629
 630
 631
 632
 633
 634
 635
 636
 637
 638
 639
 640
 641
 642
 643
 644
 645
 646
 647
 648
 649
 650
 651
 652
 653
 654
 655
 656
 657
 658
 659
 660
 661
 662
 663
 664
 665
 666
 667
 668
 669
 670
 671
 672
 673
 674
 675
 676
 677
 678
 679
 680
 681
 682
 683
 684
 685
 686
 687
 688
 689
 690
 691
 692
 693
 694
 695
 696
 697
 698
 699
 700
 701
 702
 703
 704
 705
 706
 707
 708
 709
 7010
 7011
 7012
 7013
 7014
 7015
 7016
 7017
 7018
 7019
 7020
 7021
 7022
 7023
 7024
 7025
 7026
 7027
 7028
 7029
 70210
 70211
 70212
 70213
 70214
 70215
 70216
 70217
 70218
 70219
 70220
 70221
 70222
 70223
 70224
 70225
 70226
 70227
 70228
 70229
 70230
 70231
 70232
 70233
 70234
 70235
 70236
 70237
 70238
 70239
 70240
 70241
 70242
 70243
 70244
 70245
 70246
 70247
 70248
 70249
 70250
 70251
 70252
 70253
 70254
 70255
 70256
 70257
 70258
 70259
 70260
 70261
 70262
 70263
 70264
 70265
 70266
 70267
 70268
 70269
 70270
 70271
 70272
 70273
 70274
 70275
 70276
 70277
 70278
 70279
 70280
 70281
 70282
 70283
 70284
 70285
 70286
 70287
 70288
 70289
 70290
 70291
 70292
 70293
 70294
 70295
 70296
 70297
 70298
 70299
 702100
 702101
 702102
 702103
 702104
 702105
 702106
 702107
 702108
 702109
 702110
 702111
 702112
 702113
 702114
 702115
 702116
 702117
 702118
 702119
 7021100
 7021101
 7021102
 7021103
 7021104
 7021105
 7021106
 7021107
 7021108
 7021109
 70211010
 70211011
 70211012
 70211013
 70211014
 70211015
 70211016
 70211017
 70211018
 70211019
 702110110
 702110111
 702110112
 702110113
 702110114
 702110115
 702110116
 702110117
 702110118
 702110119
 7021101110
 7021101111
 7021101112
 7021101113
 7021101114
 7021101115
 7021101116
 7021101117
 7021101118
 7021101119
 70211011110
 70211011111
 70211011112
 70211011113
 70211011114
 70211011115
 70211011116
 70211011117
 70211011118
 70211011119
 702110111110
 702110111111
 702110111112
 702110111113
 702110111114
 702110111115
 702110111116
 702110111117
 702110111118
 702110111119
 7021101111110
 7021101111111
 7021101111112
 7021101111113
 7021101111114
 7021101111115
 7021101111116
 7021101111117
 7021101111118
 7021101111119
 70211011111110
 70211011111111
 70211011111112
 70211011111113
 70211011111114
 70211011111115
 70211011111116
 70211011111117
 70211011111118
 70211011111119
 702110111111110
 702110111111111
 702110111111112
 702110111111113
 702110111111114
 702110111111115
 702110111111116
 702110111111117
 702110111111118
 702110111111119
 7021101111111110
 7021101111111111
 7021101111111112
 7021101111111113
 7021101111111114
 7021101111111115
 7021101111111116
 7021101111111117
 7021101111111118
 7021101111111119
 70211011111111110
 70211011111111111
 70211011111111112
 70211011111111113
 70211011111111114
 70211011111111115
 70211011111111116
 70211011111111117
 70211011111111118
 70211011111111119
 702110111111111110
 702110111111111111
 702110111111111112
 702110111111111113
 702110111111111114
 702110111111111115
 702110111111111116
 702110111111111117
 702110111111111118
 702110111111111119
 7021101111111111110
 7021101111111111111
 7021101111111111112
 7021101111111111113
 7021101111111111114
 7021101111111111115
 7021101111111111116
 7021101111111111117
 7021101111111111118
 7021101111111111119
 70211011111111111110
 70211011111111111111
 70211011111111111112
 70211011111111111113
 70211011111111111114
 70211011111111111115
 70211011111111111116
 70211011111111111117
 70211011111111111118
 70211011111111111119
 702110111111111111110
 7021101111111111111111
 7021101111111111111112
 7021101111111111111113
 7021101111111111111114
 7021101111111111111115
 7021101111111111111116
 7021101111111111111117
 7021101111111111111118
 7021101111111111111119
 70211011111111111111110
 702110111111111111111111
 702110111111111111111112
 702110111111111111111113
 702110111111111111111114
 702110111111111111111115
 702110111111111111111116
 702110111111111111111117
 702110111111111111111118
 702110111111111111111119
 7021101111111111111111110
 70211011111111111111111111
 70211011111111111111111112
 70211011111111111111111113
 70211011111111111111111114
 70211011111111111111111115
 70211011111111111111111116
 70211011111111111111111117
 70211011111111111111111118
 70211011111111111111111119
 702110111111111111111111110
 7021101111111111111111111111
 7021101111111111111111111112
 7021101111111111111111111113
 7021101111111111111111111114
 7021101111111111111111111115
 7021101111111111111111111116
 7021101111111111111111111117
 7021101111111111111111111118
 7021101111111111111111111119
 70211011111111111111111111110
 702110111111111111111111111111
 702110111111111111111111111112
 702110111111111111111111111113
 702110111111111111111111111114
 702110111111111111111111111115
 702110111111111111111111111116
 702110111111111111111111111117
 702110111111111111111111111118
 702110111111111111111111111119
 7021101111111111111111111111110
 70211011111111111111111111111111
 70211011111111111111111111111112
 70211011111111111111111111111113
 70211011111111111111111111111114
 70211011111111111111111111111115
 70211011111111111111111111111116
 70211011111111111111111111111117
 70211011111111111111111111111118
 70211011111111111111111111111119
 702110111111111111111111111111110
 70211011111111111111111111111111111
 70211011111111111111111111111111112
 70211011111111111111111111111111113
 70211011111111111111111111111111114
 70211011111111111111111111111111115
 70211011111111111111111111111111116
 70211011111111111111111111111111117
 70211011111111111111111111111111118
 70211011111111111111111111111111119
 702110111111111111111111111111111110
 70211011111111111111111111111111111111
 70211011111111111111111111111111111112
 70211011111111111111111111111111111113
 70211011111111111111111111111111111114
 70211011111111111111111111111111111115
 70211011111111111111111111111111111116
 70211011111111111111111111111111111117
 70211011111111111111111111111111111118
 70211011111111111111111111111111111119
 702110111111111111111111111111111111110
 70211011111111111111111111111111111111111
 70211011111111111111111111111111111111112
 70211011111111111111111111111111111111113
 70211011111111111111111111111111111111114
 70211011111111111111111111111111111111115
 70211011111111111111111111111111111111116
 70211011111111111111111111111111111111117
 70211011111111111111111111111111111111118
 70211011111111111111111111111111111111119
 702110111111111111111111111111111111111110
 702110111111111111111111111111111111111111111
 7021101111111111111111111111111111111111111112
 7021101111111111111111111111111111111111111113
 7021101111111111111111111111111111111111111114
 7021101111111111111111111111111111111111111115
 7021101111111111111111111111111111111111111116
 7021101111111111111111111111111111111111111117
 7021101111111111111111111111111111111111111118
 7021101111111111111111111111111111111111111119
 70211011111111111111111111111111111111111111110
 7021101111111111111111111111111111111111111111111
 7021101111111111111111111111111111111111111111112
 7021101111111111111111111111111111111111111111113
 7021101111111111111111111111111111111111111111114
 7021101111111111111111111111111111111111111111115
 7021101111111111111111111111111111111111111111116
 7021101111111111111111111111111111111111111111117
 7021101111111111111111111111111111111111111111118
 7021101111111111111111111111111111111111111111119
 70211011111111111111111111111111111111111111111110
 70211011111111111111111111111111111111111111111111111
 70211011111111111111111111111111111111111111111111112
 70211011111111111111111111111111111111111111111111113
```



Last but not least my beautiful spike Figma creation.



## Entry Header

Name: Alejandra Antequera

Role(s): Ultimate Supreme Designer and Programmer

Primary responsibility for this work: EVERYTHING

### *Goal of Work Session*

- Getting it done

## Tools, Resources, or Inputs Used

- Lecture Notes
- Prior drafts or code

## *GenAI Documentation*

No GenAI used for this task