

Process & Decision Documentation

Side Quest 4(Individual Work)

Project/Assignment Decisions

Again, I focused on learning how to revise the example code from professor to align adding levels to an existing game. The purpose for Side Quest #4 was to have the game was to understand how various loops interconnect.

GenAI Documentation

Date Used: February 9th,

Tool Disclosure: Claude Haiku 4.5 (extension in Visual Studios)

Purpose of Use: I used GenAI to help add levels and formality to the game. ADD more?

Summary of Interaction: As I provided the tool with guidelines to ensure the game meets Side Quest requirements, when things became complex various loops were added by GenAI.

Human Decision Point(s): Before using GenAI I played Example 5 and wanted to understand the constraints/affordances players can already achieve. As a new coder, I feel it is important for me to layout my ideas before asking GenAI to just create something generic. Therefore, writing the storyline out first helped me to refine my prompts and limit errors.

Like Celeste, I wanted the character (e.g. the blob) to go through challenges and time limits. Something that came to mind was a life count, I thought creating limited lives create this incentive for the player to use a strategic strategy. I manually changed the platform locations upwards (in the levels file) to help with added obstacles as well as fixing the sizing issues (e.g. of the obstacles and speed/duration)

Integrity & Verification Note: I checked GenAI's accuracy against the provided code example by cross-referencing comments, tutorial requirements, and created more refined prompts from previous Side Quests and only accepted code I understood (provided with comments)

Scope of GenAI Use: The tool helped clean up my thoughts, at various times I would change something manually and did not take into the changes that needed to go with it. GenAI helped with better efficiency throughout.

Limitations or Misfires: The tool did not understand proper platform placements therefore I manually changed those myself along obstacles sizing (e.g. tool did not understand the height the blob could jump). Once I completed level one, I could not play level two as platforms blocked the blobs ability to fit through/jump onto them, therefore I went in and eliminated a few platforms manually.

Summary of Process (Human + Tool)

- I broke down the prompts to prevent overcomplication and missed outputs for game functionality
- Went in and manually fixed any code (e.g obstacles sizing, platform reconfigurations, speed/duration)

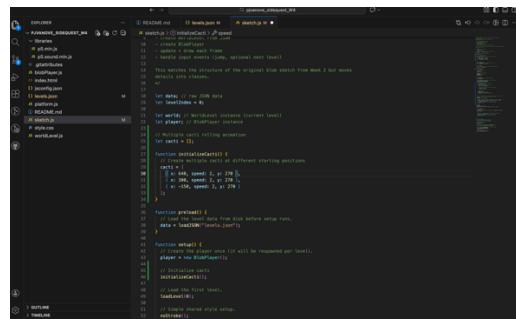
Decision Points & Trade-offs

I also accepted the tools addition of text evidence with the lives tallies, I could interpret them easily (e.g. size of text was legible). What I changed were the game element sizes, and speed/duration to allow the player use a strategic mindset/more challenging rather than no challenge at all. I also realized I had to ensure the blob could jump high enough to reach the new height of the platforms, I got the tool to put that in.

This personalization helped me reflect on the Celeste game in class, the games ability to challenge the player but also keep their attention (even when lives ran out) was my hope to this game. The text indicator created warnings for the player.

Prompt #3 (Speed of moving objects and sizes)

GenAI Output



```

function initializeCacti() {
    // Create multiple cacti at different starting positions
    cacti = [
        { x: 640, speed: 5, y: 270 },
        { x: 300, speed: 4.5, y: 270 },
        { x: -150, speed: 5, y: 270 },
        { x: -150, speed: 4.5, y: 270 },
        { x: -150, speed: 5, y: 270 },
        { x: -150, speed: 4.5, y: 270 },
        { x: -150, speed: 5, y: 270 },
        { x: -150, speed: 4.5, y: 270 }
    ];
}

function initializeSmalls() {
    // Create multiple smalls at different starting positions
    smalls = [
        { x: 640, speed: 5, y: 270 },
        { x: 300, speed: 4.5, y: 270 },
        { x: -150, speed: 5, y: 270 }
    ];
}

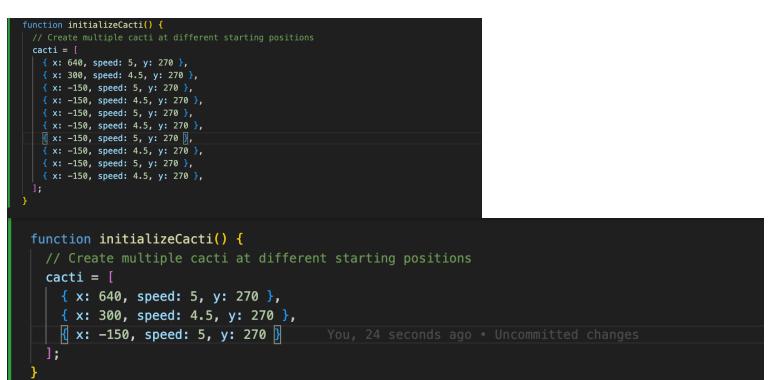
function gameLoop() {
    // Main game loop
    // ...
}

function main() {
    // Initialize game
    initializeCacti();
    initializeSmalls();
    gameLoop();
}

main();

```

My changes



```

function initializeCacti() {
    // Create multiple cacti at different starting positions
    cacti = [
        { x: 640, speed: 5, y: 270 },
        { x: 300, speed: 4.5, y: 270 },
        { x: -150, speed: 5, y: 270 }
    ];
}

function initializeSmalls() {
    // Create multiple smalls at different starting positions
    smalls = [
        { x: 640, speed: 5, y: 270 },
        { x: 300, speed: 4.5, y: 270 },
        { x: -150, speed: 5, y: 270 }
    ];
}

function gameLoop() {
    // Main game loop
    // ...
}

function main() {
    // Initialize game
    initializeCacti();
    initializeSmalls();
    gameLoop();
}

main();

```

Number of platforms level two

GenAI Output

```
7  },
8  {
9    "name": "Platform Rush",
10   "gravity": 0.7,      You, 15 minutes ago * Uncommitted changes
11   "jumpV": -16.0,
12   "theme": { "bg": "#E8F4F8", "platform": "#4A90E2", "blob": "#FF6B6B" },
13   "start": { "x": 320, "y": 280, "r": 26 },
14   "platforms": [
15     { "x": 0, "y": 320, "w": 640, "h": 40 },
16     { "x": 50, "y": 250, "w": 100, "h": 12 },
17     { "x": 200, "y": 200, "w": 100, "h": 12 },
18     { "x": 350, "y": 160, "w": 100, "h": 12 },
19     { "x": 500, "y": 120, "w": 100, "h": 12 },
20     { "x": 150, "y": 80, "w": 120, "h": 12 },
21     { "x": 400, "y": 30, "w": 150, "h": 12 }
22   ],
23 },
24 }
```

My changes

```
19  },
20  {
21    "name": "Platform Rush",
22    "gravity": 0.7,
23    "jumpV": -16.0,
24    "theme": { "bg": "#E8F4F8", "platform": "#4A90E2", "blob": "#FF6B6B" },
25    "start": { "x": 320, "y": 280, "r": 26 },
26    "platforms": [
27      { "x": 0, "y": 320, "w": 640, "h": 40 },
28      { "x": 200, "y": 200, "w": 100, "h": 12 },
29      { "x": 350, "y": 160, "w": 100, "h": 12 },
30      { "x": 500, "y": 120, "w": 100, "h": 12 },
31      { "x": 150, "y": 80, "w": 120, "h": 12 }
32    ],
33  },
```

Verification & Judgement

Explain how you evaluated whether your decision or change was appropriate:

- Cross-references code structure from this week's tutorial example
- I manually tested in Visual Studios for errors/ debugging needs
- I confirmed misfires and make manually changes through what I have learned so far (e.g. Prompt #2 size changes and how many objects were added) also adding my own refinements versus GenAI outputs displays my own judgments were added to better the game functionality

Appendix

Transcript

Prompt #1 (creating game layout)

I am creating a multi-level game that is continuous add one more level to this one already created. Therefore, I want a black wall at the top screen that brings the blob and player into the next level. I would like two new levels added.

Prompt #2

Please ensure the blob can jump higher and onto the platform and the black wall needs to be on right wall on the top (make it vertical) ensure it changes to the next level where the blob can jump through it after the third platform.

```
levels.json
{
  "schemaVersion": 1,
  "levels": [
    {
      "name": "Intro Steps",
      "gravity": "0.5",
      "jumpV": "-31.5",
      "jumpS": "-35.5",
      "theme": { "bg": "#F0F0F0", "platform": "#C8C8C8", "blob": "#1478FF" },
      "start": { "x": 80, "y": 220, "r": 26 },
      "platforms": [
        { "x": 100, "y": 300, "w": 640, "h": 36 },
        { "x": 120, "y": 264, "w": 120, "h": 12 },
        { "x": 380, "y": 156, "w": 96, "h": 12 },
        { "x": 380, "y": 112, "w": 96, "h": 12 },
        { "x": 520, "y": 264, "w": 96, "h": 12 }
      ]
    },
    {
      "name": "Platform Rush",
      "gravity": "0.7",
      "jumpV": "-30.5",
      "jumpS": "-34.5",
      "theme": { "bg": "#E6F0FA", "platform": "#A498E2", "blob": "#FF6B6B" },
      "start": { "x": 320, "y": 288, "r": 26 },
      "platforms": [
        { "x": 100, "y": 328, "w": 640, "h": 40 },
        { "x": 56, "y": 256, "w": 100, "h": 12 },
        { "x": 280, "y": 288, "w": 100, "h": 12 },
        { "x": 356, "y": 304, "w": 100, "h": 12 },
        { "x": 356, "y": 336, "w": 100, "h": 12 },
        { "x": 156, "y": 86, "w": 128, "h": 12 },
        { "x": 480, "y": 36, "w": 156, "h": 12 }
      ]
    },
    {
      "name": "Sky Climb",
      "gravity": "0.5",
      "jumpV": "-30.5",
      "jumpS": "-34.5",
      "theme": { "bg": "#87CEEB", "platform": "#FFD700", "blob": "#800080" },
      "start": { "x": 180, "y": 380, "r": 26 },
      "platforms": [
        { "x": 100, "y": 380, "w": 640, "h": 36 },
        { "x": 120, "y": 344, "w": 120, "h": 12 },
        { "x": 380, "y": 236, "w": 96, "h": 12 },
        { "x": 380, "y": 192, "w": 96, "h": 12 },
        { "x": 520, "y": 344, "w": 96, "h": 12 }
      ]
    }
  ]
}
```

```

{
  "levels": [
    {
      "platforms": [
        {
          "x": 150, "y": 120, "w": 80, "h": 12 },
        {
          "x": 150, "y": 80, "w": 80, "h": 12 },
        {
          "x": 400, "y": 30, "w": 150, "h": 12 }
      ],
      "name": "Sky Cacti",
      "gravity": 8.6,
      "jumpY": -18.5,
      "platforms": [
        {
          "x": 100, "y": 320, "w": 60, "h": 40 },
        {
          "x": 100, "y": 260, "w": 80, "h": 12 },
        {
          "x": 250, "y": 220, "w": 80, "h": 12 },
        {
          "x": 250, "y": 180, "w": 80, "h": 12 },
        {
          "x": 180, "y": 140, "w": 80, "h": 12 },
        {
          "x": 380, "y": 100, "w": 80, "h": 12 },
        {
          "x": 380, "y": 60, "w": 80, "h": 12 },
        {
          "x": 250, "y": 20, "w": 150, "h": 12 }
      ],
      "start": {
        "x": 100, "y": 300, "r": 26 },
      "platforms": [
        {
          "x": 100, "y": 320, "w": 60, "h": 40 },
        {
          "x": 100, "y": 260, "w": 80, "h": 12 },
        {
          "x": 250, "y": 220, "w": 80, "h": 12 },
        {
          "x": 250, "y": 180, "w": 80, "h": 12 },
        {
          "x": 180, "y": 140, "w": 80, "h": 12 },
        {
          "x": 380, "y": 100, "w": 80, "h": 12 },
        {
          "x": 380, "y": 60, "w": 80, "h": 12 },
        {
          "x": 250, "y": 20, "w": 150, "h": 12 }
      ]
    }
  ]
}

```

Prompt #3

Within the first level, please have  rolling on the ground coming from the right side and continuously looping back.

```

function setup() {
  // Create the player once (it will be respawned per level),
  player = new BlobPlayer();
  // Initialize cacti
  initialCacti();
  // Load the first level,
  loadLevel();
}

// Simple shared style setup.
noStroke();
textFont("sans-serif");
 textSize(14);

function draw() {
  // 1) Draw the world (background + platforms)
  world.drawWorld();
  // 2) Update and draw the player on top of the world
  player.updateWorld(platforms);
  player.drawWorld(theme.blob);

  // 3) Draw rolling cactus on level 1 only
  // 3.1) Draw cactus on level 1 only
  if (cactusX < 400) {
    cactusX = width + 20;
  }

  // Draw cactus emoji
  textSize(40);
  text(`", cactusX, cactusY);
  textSize(14); // Reset text size for HUD
  for (let i = 0; i < 3; i++) {
    // Update cactus position (move left)
    cactusX -= cactusSpeed;
    // Loop back to right side when it goes off screen
    if (cactusX < -400) {
      cactusX = width + 20;
    }
  }
}

```

```

1 6
2 7 This file orchestrates everything:
3 8   - create world
4 9   - create WorldLevel From JSON
5 10  - create BlobPlayer
6 11  - update + draw each frame
7 12  - handle input events (jump, optional next level)
8 13
9 14 This matches the structure of the original blob sketch from Week 2 but moves
10 15 details into classes.
11 16
12 17
13 18 let data; // raw JSON data
14 19 let levelIndex = 0;
15 20
16 21 let world; // WorldLevel instance (current level)
17 22 let player; // BlobPlayer instance
18 23
19 24 // Platform timer for level 2
20 25 let platformTimers = {};
21 26 let currentPlatformIndex = 0; // frames (about 3 seconds at 60fps)
22 27 let currentPlatformIndex = -1; // which platform player is on
23 28 let collapsedPlatforms = new Set(); // track which platforms have collapsed
24 29
25 30 // Multiple cacti rolling animation
26 31 let cacti = [];
27 32
28 33 function initializeCacti() {
29 34   // Create multiple cacti at different starting positions
30 35   cacti = [
31 36     { x: 640, speed: 5, y: 270 },
32 37     { x: 380, speed: 4.5, y: 270 },
33 38     { x: -150, speed: 5, y: 270 },
34 39     { x: -150, speed: 4.5, y: 270 },
35 40     { x: 150, speed: 5, y: 270 },
36 41     { x: 150, speed: 4.5, y: 270 },
37 42     { x: -150, speed: 5, y: 270 },
38 43     { x: -150, speed: 4.5, y: 270 },
39 44     { x: 150, speed: 5, y: 270 },
40 45     { x: -150, speed: 4.5, y: 270 },
41 46   ];
42 47 }
43 48
44 49 function preload() {
45 50   // Load the level data from disk before setup runs.
46 51   data = loadJSON("levels.json");
47 52 }
48 53
49 54 function setup() {
50 55   // Create the player once it will be respawned per level.
51 56   player = new BlobPlayer();
52 57 }
53 58
54 59 // Initialize cacti
55 60
56 61
57 62
58 63
59 64
60 65
61 66
62 67
63 68
64 69
65 70
66 71
67 72
68 73
69 74
70 75
71 76
72 77
73 78
74 79
75 80
76 81
77 82
78 83
79 84
80 85
81 86
82 87
83 88
84 89
85 90
86 91
87 92
88 93
89 94
90 95
91 96
92 97
93 98
94 99
95 100
96 101
97 102
98 103
99 104
100 105
101 106
102 107
103 108
104 109
105 110
106 111
107 112
108 113
109 114
110 115
111 116
112 117
113 118
114 119
115 120
116 121
117 122
118 123
119 124
120 125
121 126
122 127
123 128
124 129
125 130
126 131
127 132
128 133
129 134
130 135
131 136
132 137
133 138
134 139
135 140
136 141
137 142
138 143
139 144
140 145
141 146
142 147
143 148
144 149
145 150
146 151
147 152
148 153
149 154
150 155
151 156
152 157
153 158
154 159
155 160
156 161
157 162
158 163
159 164
160 165
161 166
162 167
163 168
164 169
165 170
166 171
167 172
168 173
169 174
170 175
171 176
172 177
173 178
174 179
175 180
176 181
177 182
178 183
179 184
180 185
181 186
182 187
183 188
184 189
185 190
186 191
187 192
188 193
189 194
190 195
191 196
192 197
193 198
194 199
195 200
196 201
197 202
198 203
199 204
200 205
201 206
202 207
203 208
204 209
205 210
206 211
207 212
208 213
209 214
210 215
211 216
212 217
213 218
214 219
215 220
216 221
217 222
218 223
219 224
220 225
221 226
222 227
223 228
224 229
225 230
226 231
227 232
228 233
229 234
230 235
231 236
232 237
233 238
234 239
235 240
236 241
237 242
238 243
239 244
240 245
241 246
242 247
243 248
244 249
245 250
246 251
247 252
248 253
249 254
250 255
251 256
252 257
253 258
254 259
255 260
256 261
257 262
258 263
259 264
260 265
261 266
262 267
263 268
264 269
265 270
266 271
267 272
268 273
269 274
270 275
271 276
272 277
273 278
274 279
275 280
276 281
277 282
278 283
279 284
280 285
281 286
282 287
283 288
284 289
285 290
286 291
287 292
288 293
289 294
290 295
291 296
292 297
293 298
294 299
295 300
296 301
297 302
298 303
299 304
299 305
300 306
301 307
302 308
303 309
304 310
305 311
306 312
307 313
308 314
309 315
310 316
311 317
312 318
313 319
314 320
315 321
316 322
317 323
318 324
319 325
320 326
321 327
322 328
323 329
324 330
325 331
326 332
327 333
328 334
329 335
330 336
331 337
332 338
333 339
334 340
335 341
336 342
337 343
338 344
339 345
340 346
341 347
342 348
343 349
344 350
345 351
346 352
347 353
348 354
349 355
350 356
351 357
352 358
353 359
354 360
355 361
356 362
357 363
358 364
359 365
360 366
361 367
362 368
363 369
364 370
365 371
366 372
367 373
368 374
369 375
370 376
371 377
372 378
373 379
374 380
375 381
376 382
377 383
378 384
379 385
380 386
381 387
382 388
383 389
384 390
385 391
386 392
387 393
388 394
389 395
390 396
391 397
392 398
393 399
394 400
395 401
396 402
397 403
398 404
399 405
399 406
400 407
401 408
402 409
403 410
404 411
405 412
406 413
407 414
408 415
409 416
410 417
411 418
412 419
413 420
414 421
415 422
416 423
417 424
418 425
419 426
420 427
421 428
422 429
423 430
424 431
425 432
426 433
427 434
428 435
429 436
430 437
431 438
432 439
433 440
434 441
435 442
436 443
437 444
438 445
439 446
440 447
441 448
442 449
443 450
444 451
445 452
446 453
447 454
448 455
449 456
450 457
451 458
452 459
453 460
454 461
455 462
456 463
457 464
458 465
459 466
460 467
461 468
462 469
463 470
464 471
465 472
466 473
467 474
468 475
469 476
470 477
471 478
472 479
473 480
474 481
475 482
476 483
477 484
478 485
479 486
480 487
481 488
482 489
483 490
484 491
485 492
486 493
487 494
488 495
489 496
490 497
491 498
492 499
493 500
494 501
495 502
496 503
497 504
498 505
499 506
499 507
500 508
501 509
502 510
503 511
504 512
505 513
506 514
507 515
508 516
509 517
510 518
511 519
512 520
513 521
514 522
515 523
516 524
517 525
518 526
519 527
520 528
521 529
522 530
523 531
524 532
525 533
526 534
527 535
528 536
529 537
530 538
531 539
532 540
533 541
534 542
535 543
536 544
537 545
538 546
539 547
540 548
541 549
542 550
543 551
544 552
545 553
546 554
547 555
548 556
549 557
550 558
551 559
552 560
553 561
554 562
555 563
556 564
557 565
558 566
559 567
560 568
561 569
562 570
563 571
564 572
565 573
566 574
567 575
568 576
569 577
570 578
571 579
572 580
573 581
574 582
575 583
576 584
577 585
578 586
579 587
580 588
581 589
582 590
583 591
584 592
585 593
586 594
587 595
588 596
589 597
590 598
591 599
592 600
593 601
594 602
595 603
596 604
597 605
598 606
599 607
599 608
600 609
601 610
602 611
603 612
604 613
605 614
606 615
607 616
608 617
609 618
610 619
611 620
612 621
613 622
614 623
615 624
616 625
617 626
618 627
619 628
620 629
621 630
622 631
623 632
624 633
625 634
626 635
627 636
628 637
629 638
630 639
631 640
632 641
633 642
634 643
635 644
636 645
637 646
638 647
639 648
640 649
641 650
642 651
643 652
644 653
645 654
646 655
647 656
648 657
649 658
650 659
651 660
652 661
653 662
654 663
655 664
656 665
657 666
658 667
659 668
660 669
661 670
662 671
663 672
664 673
665 674
666 675
667 676
668 677
669 678
670 679
671 680
672 681
673 682
674 683
675 684
676 685
677 686
678 687
679 688
680 689
681 690
682 691
683 692
684 693
685 694
686 695
687 696
688 697
689 698
690 699
691 700
692 701
693 702
694 703
695 704
696 705
697 706
698 707
699 708
699 709
700 710
701 711
702 712
703 713
704 714
705 715
706 716
707 717
708 718
709 719
710 720
711 721
712 722
713 723
714 724
715 725
716 726
717 727
718 728
719 729
720 730
721 731
722 732
723 733
724 734
725 735
726 736
727 737
728 738
729 739
730 740
731 741
732 742
733 743
734 744
735 745
736 746
737 747
738 748
739 749
740 750
741 751
742 752
743 753
744 754
745 755
746 756
747 757
748 758
749 759
750 760
751 761
752 762
753 763
754 764
755 765
756 766
757 767
758 768
759 769
760 770
761 771
762 772
763 773
764 774
765 775
766 776
767 777
768 778
769 779
770 780
771 781
772 782
773 783
774 784
775 785
776 786
777 787
778 788
779 789
780 790
781 791
782 792
783 793
784 794
785 795
786 796
787 797
788 798
789 799
790 800
791 801
792 802
793 803
794 804
795 805
796 806
797 807
798 808
799 809
800 810
801 811
802 812
803 813
804 814
805 815
806 816
807 817
808 818
809 819
810 820
811 821
812 822
813 823
814 824
815 825
816 826
817 827
818 828
819 829
820 830
821 831
822 832
823 833
824 834
825 835
826 836
827 837
828 838
829 839
830 840
831 841
832 842
833 843
834 844
835 845
836 846
837 847
838 848
839 849
840 850
841 851
842 852
843 853
844 854
845 855
846 856
847 857
848 858
849 859
850 860
851 861
852 862
853 863
854 864
855 865
856 866
857 867
858 868
859 869
860 870
861 871
862 872
863 873
864 874
865 875
866 876
867 877
868 878
869 879
870 880
871 881
872 882
873 883
874 884
875 885
876 886
877 887
878 888
879 889
880 890
881 891
882 892
883 893
884 894
885 895
886 896
887 897
888 898
889 899
890 900
891 901
892 902
893 903
894 904
895 905
896 906
897 907
898 908
899 909
900 910
901 911
902 912
903 913
904 914
905 915
906 916
907 917
908 918
909 919
910 920
911 921
912 922
913 923
914 924
915 925
916 926
917 927
918 928
919 929
920 930
921 931
922 932
923 933
924 934
925 935
926 936
927 937
928 938
929 939
930 940
931 941
932 942
933 943
934 944
935 945
936 946
937 947
938 948
939 949
940 950
941 951
942 952
943 953
944 954
945 955
946 956
947 957
948 958
949 959
950 960
951 961
952 962
953 963
954 964
955 965
956 966
957 967
958 968
959 969
960 970
961 971
962 972
963 973
964 974
965 975
966 976
967 977
968 978
969 979
970 980
971 981
972 982
973 983
974 984
975 985
976 986
977 987
978 988
979 989
980 990
981 991
982 992
983 993
984 994
985 995
986 996
987 997
988 998
989 999
990 999

```

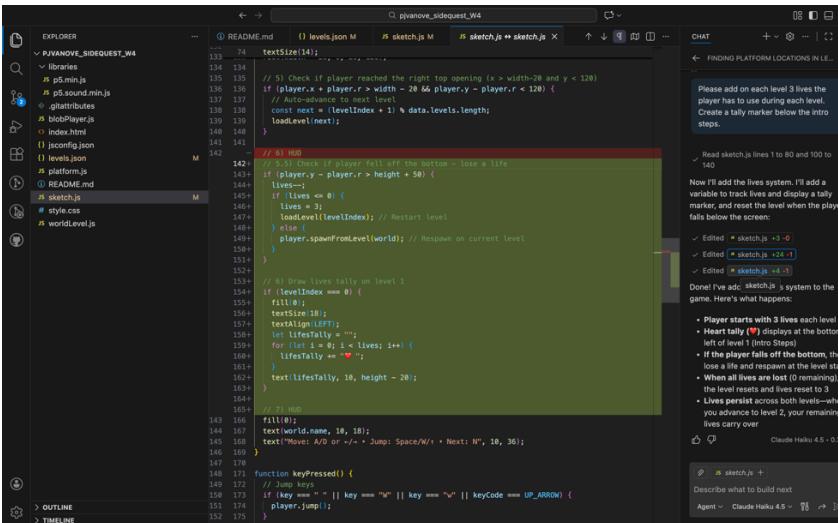
Prompt #4

Within the second level, please have a timer set on giving the player limited time complete this level

```

1 6
2 7 This file orchestrates everything:
3 8   - create world
4 9   - create WorldLevel From JSON
5 10  - create BlobPlayer
6 11  - update + draw each frame
7 12  - handle input events (jump, optional next level)
8 13
9 14 This matches the structure of the original blob sketch from Week 2 but moves
10 15 details into classes.
11 16
12 17
13 18 let data; // raw JSON data
14 19 let levelIndex = 0;
15 20
16 21 let world; // WorldLevel instance (current level)
17 22 let player; // BlobPlayer instance
18 23
19 24 // Platform timer for level 2
20 25 let platformTimers = {};
21 26 let currentPlatformIndex = 0; // frames (about 3 seconds at 60fps)
22 27 let currentPlatformIndex = -1; // which platform player is on
23 28 let collapsedPlatforms = new Set(); // track which platforms have collapsed
24 29
25 30 // Multiple cacti rolling animation
26 31 let cacti = [];
27 32
28 33 function initializeCacti() {
29 34   // Create multiple cacti at different starting positions
30 35   cacti = [
31 36     { x: 640, speed: 5, y: 270 },
32 37     { x: 380, speed: 4.5, y: 270 },
33 38     { x: -150, speed: 5, y: 270 },
34 39     { x: -150, speed: 4.5, y: 270 },
35 40     { x: 150, speed: 5, y: 270 },
36 41     { x: 150, speed: 4.5, y: 270 },
37 42     { x: -150, speed: 5, y: 270 },
38 43     { x: -150, speed: 4.5, y: 270 },
39 44     { x: 150, speed: 5, y: 270 },
40 45     { x: -150, speed: 4.5, y: 270 },
41 46   ];
42 47 }
43 48
44 49 function preload() {
45 50   // Load the level data from disk before setup runs.
46 51   data = loadJSON("levels.json");
47 52 }
48 53
49 54 function setup() {
50 55   // Create the player once it will be respawned per level.
51 56   player = new BlobPlayer();
52 57 }
53 58
54 59 // Initialize cacti
55 60
56 61
57 62
58 63
59 64
60 65
61 66
62 67
63 68
64 69
65 70
66 71
67 72
68 73
69 74
70 75
71 76
72 77
73 78
74 79
75 80
76 81
77 82
78 83
79 84
80 85
81 86
82 87
83 88
84 89
85 90
86 91
87 92
88 93
89 94
90 95
91 96
92 97
93 98
94 99
95 100
96 101
97 102
98 103
99 104
100 105
101 106
102 107
103 108
104 109
105 110
106 111
107 112
108 113
109 114
110 115
111 116
112 117
113 118
114 119
115 120
116 121
117 122
118 123
119 124
120 125
121 126
122 127
123 128
124 129
125 130
126 131
127 132
128 133
129 134
130 135
131 136
132 137
133 138
134 139
135 140
136 141
137 142
138 143
139 144
140 145
141 146
142 147
143 148
144 149
145 150
146 151
147 152
148 153
149 154
150 155
151 156
152 157
153 158
154 159
155 160
156 161
157 162
158 163
159 164
160 165
161 166
162 167
163 168
164 169
165 170
166 171
167 172
168 173
169 174
170 175
171 176
172 177
173 178
174 179
175 180
176 181
177 182
178 183
179 184
180 185
181 186
182 187
183 188
184 189
185 190
186 191
187 192
188 193
189 194
190 195
191 196
192 197
193 198
194 199
195 200
196 201
197 202
198 203
199 204
200 205
201 206
202 207
203 208
204 209
205 210
206 211
207 212
208 213
209 214
210 215
211 216
212 217
213 218
214 219
215 220
216 221
217 222
218 223
219 224
220 225
221 226
222 227
223 228
224 229
225 230
226 231
227 232
228 233
229 234
230 235
231 236
232 237
233 238
234 239
235 240
236 241
237 242
238 243
239 244
240 245
241 246
242 247
243 248
244 249
245 250
246 251
247 252
248 253
249 254
250 255
251 256
252 257
253 258
254 259
255 260
256 261
257 262
258 263
259 264
260 265
261 266
262 267
263 268
264 269
265 270
266 271
267 272
268 273
269 274
270 275
271 276
272 277
273 278
274 279
275 280
276 281
277 282
278 283
279 284
280 285
281 286
282 287
283 288
284 289
285 290
286 291
287 292
288 293
289 294
290 295
291 296
292 297
293 298
294 299
295 300
296 301
297 302
298 303
299 304
299 305
300 306
301 307
302 308
303 309
304 310
305 311
306 312
307 313
308 314
309 315
310 316
311 317
312 318
313 319
314 320
315 321
316 322
317 323
318 324
319 325
320 326
321 327
322 328
323 329
324 330
325 331
326 332
327 333
328 334
329 335

```



EXPLORER

JAVADOC LIBRARIES

p5.js

p5.sound.js

glitterattributes

blobPlayer.js

index.html

config.json

levels.json

platform.js

README.md

sketch.js

style.css

worldLevel.js

textSize(14);

// If player reached the right top opening (x > width-20 and y < 120)

if (player.x + player.r > width - 20 && player.y - player.r < 120) {

// Auto-slide to next level

const next = (levelIndex + 1) % data.levels.length;

loadLevel(next);

// (3) HMO

// (4) Check if player fell off the bottom - lose a life

if (player.y - player.r > height + 50) {

lives--;

if (lives === 0) {

lives = 3;

loadLevel(levelIndex); // Restart level

} else {

player.spawnFromLevel(world); // Respawn on current level

lives = 3;

}}

// (5) Draw lives tally on level 1

if (levelIndex === 0) {

fill(0);

textSize(18);

text("Lives", 10, 10);

text("Lives", 10, 30);

int lifeTally = 0;

for (let i = 0; i < lives; i++) {

lifeTally += ❤;

}

text(lifeTally, 10, height - 20);

}}

fill(0);

text(world.name, 10, 10);

text("Move: A/D or -/+ Jump; Space/M+ = Next; N", 10, 30);

146 169 }

147

148 171 function keypressed() {

149 172 // Jump keys

150 173 if (key === " " || key === "w" || key === "v" || keyCode === UP_ARROW) {

151 174 player.jump();

152 175 }

153 176 }

154 177

155 178

156 179

157 180

158 181

159 182

160 183

161 184

162 185

163 186

164 187

165 188

166 189

167 190

168 191

169 192

170 193

171 194

172 195

173 196

174 197

175 198

176 199

177 200

178 201

179 202

180 203

181 204

182 205

183 206

184 207

185 208

186 209

187 210

188 211

189 212

190 213

191 214

192 215

193 216

194 217

195 218

196 219

197 220

198 221

199 222

200 223

201 224

202 225

203 226

204 227

205 228

206 229

207 230

208 231

209 232

210 233

211 234

212 235

213 236

214 237

215 238

216 239

217 240

218 241

219 242

220 243

221 244

222 245

223 246

224 247

225 248

226 249

227 250

228 251

229 252

230 253

231 254

232 255

233 256

234 257

235 258

236 259

237 260

238 261

239 262

240 263

241 264

242 265

243 266

244 267

245 268

246 269

247 270

248 271

249 272

250 273

251 274

252 275

253 276

254 277

255 278

256 279

257 280

258 281

259 282

260 283

261 284

262 285

263 286

264 287

265 288

266 289

267 290

268 291

269 292

270 293

271 294

272 295

273 296

274 297

275 298

276 299

277 300

278 301

279 302

280 303

281 304

282 305

283 306

284 307

285 308

286 309

287 310

288 311

289 312

290 313

291 314

292 315

293 316

294 317

295 318

296 319

297 320

298 321

299 322

300 323

301 324

302 325

303 326

304 327

305 328

306 329

307 330

308 331

309 332

310 333

311 334

312 335

313 336

314 337

315 338

316 339

317 340

318 341

319 342

320 343

321 344

322 345

323 346

324 347

325 348

326 349

327 350

328 351

329 352

330 353

331 354

332 355

333 356

334 357

335 358

336 359

337 360

338 361

339 362

340 363

341 364

342 365

343 366

344 367

345 368

346 369

347 370

348 371

349 372

350 373

351 374

352 375

353 376

354 377

355 378

356 379

357 380

358 381

359 382

360 383

361 384

362 385

363 386

364 387

365 388

366 389

367 390

368 391

369 392

370 393

371 394

372 395

373 396

374 397

375 398

376 399

377 400

378 401

379 402

380 403

381 404

382 405

383 406

384 407

385 408

386 409

387 410

388 411

389 412

390 413

391 414

392 415

393 416

394 417

395 418

396 419

397 420

398 421

399 422

400 423

401 424

402 425

403 426

404 427

405 428

406 429

407 430

408 431

409 432

410 433

411 434

412 435

413 436

414 437

415 438

416 439

417 440

418 441

419 442

420 443

421 444

422 445

423 446

424 447

425 448

426 449

427 450

428 451

429 452

430 453

431 454

432 455

433 456

434 457

435 458

436 459

437 460

438 461

439 462

440 463

441 464

442 465

443 466

444 467

445 468

446 469

447 470

448 471

449 472

450 473

451 474

452 475

453 476

454 477

455 478

456 479

457 480

458 481

459 482

460 483

461 484

462 485

463 486

464 487

465 488

466 489

467 490

468 491

469 492

470 493

471 494

472 495

473 496

474 497

475 498

476 499

477 500

478 501

479 502

480 503

481 504

482 505

483 506

484 507

485 508

486 509

487 510

488 511

489 512

490 513

491 514

492 515

493 516

494 517

495 518

496 519

497 520

498 521

499 522

500 523

501 524

502 525

503 526

504 527

505 528

506 529

507 530

508 531

509 532

510 533

511 534

512 535

513 536

514 537

515 538

516 539

517 540

518 541

519 542

520 543

521 544

522 545

523 546

524 547

525 548

526 549

527 550

528 551

529 552

530 553

531 554

532 555

533 556

534 557

535 558

536 559

537 560

538 561

539 562

540 563

541 564

542 565

543 566

544 567

545 568

546 569

547 570

548 571

549 572

550 573

551 574

552 575

553 576

554 577

555 578

556 579

557 580

558 581

559 582

560 583

561 584

562 585

563 586

564 587

565 588

566 589

567 590

568 591

569 592

570 593

571 594

572 595

573 596

574 597

575 598

576 599

577 600

578 601

579 602

580 603

581 604

582 605

583 606

584 607

585 608

586 609

587 610

588 611

589 612

590 613

591 614

592 615

593 616

594 617

595 618

596 619

597 620

598 621

599 622

600 623

601 624

602 625

603 626

604 627

605 628

606 629

607 630

608 631

609 632

610 633

611 634

612 635

613 636

614 637

615 638

616 639

617 640

618 641

619 642

620 643

621 644

622 645

623 646

624 647

625 648

626 649

627 650

628 651

629 652

630 653

631 654

632 655

633 656

634 657

635 658

636 659

637 660

638 661

639 662

640 663

641 664

642 665

643 666

644 667

645 668

646 669

647 670

648 671

649 672

650 673

651 674

652 675

653 676

654 677

655 678

656 679

657 680

658 681

659 682

660 683

661 684

662 685

663 686

664 687

665 688

666 689

667 690

668 691

669 692

670 693

671 694

672 695

673 696

674 697

675 698

676 699

677 700

678 701

679 702

680 703

681 704

682 705

683 706

684 707

685 708

686 709

687 710

688 711

689 712

690 713

691 714

692 715

693 716

694 717

695 718

696 719

697 720

698 721

699 722

700 723

701 724

702 725

703 726

704 727

705 728

706 729

707 730

708 731

709 732

710 733

711 734

712 735

713 736

714 737

715 738

716 739

717 740

718 741

719 742

720 743

721 744

722 745

723 746

724 747

725 748

726 749

727 750

728 751

729 752

730 753

731 754

732 755

733 756

734 757

735 758

736 759

737 760

738 761

739 762

740 763

741 764

742 765

743 766

744 767

745 768

746 769

747 770

748 771

749 772

750 773

751 774

752 775

753 776

754 777

755 778

756 779

757 780

758 781

759 782

760 783

761 784

762 785

763 786

764 787

765 788

766 789

767 790

768 791

769 792

770 793

771 794

772 795

773 796

774 797

775 798

776 799

777 800

778 801

779 802

780 803

781 804

782 805

783 806

784 807

785 808

786 809

787 810

788 811

789 812

790 813

791 814

792 815

793 816

794 817

795 818

796 819

797 820

798 821

799 822

800 823

801 824

802 825

803 826

804 827

805 828

806 829

807 830

808 831

809 832

810 833

811 834

812 835

813 836

814 837

815 838

816 839

817 840

818 841

819 842

820 843

821 844

822 845

823 846

824 847

825 848

826 849

827 850

828 851

829 852

830 853

831 854

832 855

833 856

834 857

835 858

836 859

837 860

838 861

839 862

840 863

841 864

842 865

843 866

844 867

845 868

846 869

847 870

848 871

849 872

850 873

851 874

852 875

853 876

854 877

855 878

856 879

857 880

858 881

859 882

860 883

861 884

862 885

863 886

864 887

865 888

866 889

867 890

868 891

869 892

870 893

871 894

872 895

873 896

874 897

875 898

876 899

877 900

878 901

879 902

880 903

881 904

882 905

883 906

884 907

885 908

886 909

887 910

888 911

889 912

890 913

891 914

892 915

893 916

894 917

895 918

896 919

897 920

898 921

899 922

900 923

901 924

902 925

903 926

904 927

905 928

906 929

907 930

908 931

909 932

910 933

911 934

912 935

913 936

914 937

915 938

916 939

917 940

918 941

919 942

920 943

921 944

922 945

923 946

924 947

925 948

926 949

927 950

928 951

929 952

930 953

931 954

932 955

933 956

934 957

935 958

936 959

937 960

938 961

939 962

940 963

941 964

942 965

943 966

944 967

945 968

946 969

947 970

948 971

949 972

950 973

951 974

952 975

953 976

954 977

955 978

956 979

957 980

958 981

959 982

960 983

961 984

962 985

963 986

964 987

965 988

966 989

967 990

968 991

969 992

970 993

971 994

972 995

973 996

974 997

975 998

976 999

977 1000

CHAT

FINDING PLATFORM LOCATIONS

Please add each level 3 lives the player has to during each level. Create a tally marker below the intro steps.

Read sketch.js lines 1 to 80 and 100 to 1000

New level and the lives system. I'll add a variable to track lives and display a tally marker, and reset the level when the player falls below the screen

Edited sketch.js +

Edited sketch.js +

Edited sketch.js +

Done! I've added a tally marker to the system when the level ends. Here's what happens:

- Player starts with 3 lives each level
- Heart tally (♥) displays at the bottom left of the intro steps
- If the player falls off the bottom, they lose a life and restart at the level start
- When the level ends, the tally marker and the level ends and lives reset to 3
- Lives persist across both levels--when you advance to level 2, your remaining lives carry over

Claude Haku 4.5 - 0.93x

sketch.js +

Describe what to build next

Agent Claude Haku 4.5 - 0.93x