

The DS4 Equalizer

LAB 07

SECTION 5

ARYAN RAO

DATE- 11/2/2021

Problem

Use the DualShock 4 and determine using its roll and pitch whether it's being tipped left or right.

Analysis

The program should start by displaying graph roll. To display roll, user should press triangle button, while the X button is to be pushed for pitch.

Design

- Scan inputs from DS4
- Collect gyroscope values
- Print the character respectively
- Collect joystick data
- Associate that with circle button

Testing

Calculate the roll and pitch values manually and compare them with the values we get on the program.

Comments

Observing patterns and readings enables you to obtain good range of value for each movement. This is crucial for an accurate and a good working program.

Questions

1. How did you scale your values? Write an equation to justify it.

```
if (gz > 0.2 && gz <= 1)  
gz=gz*0.2
```

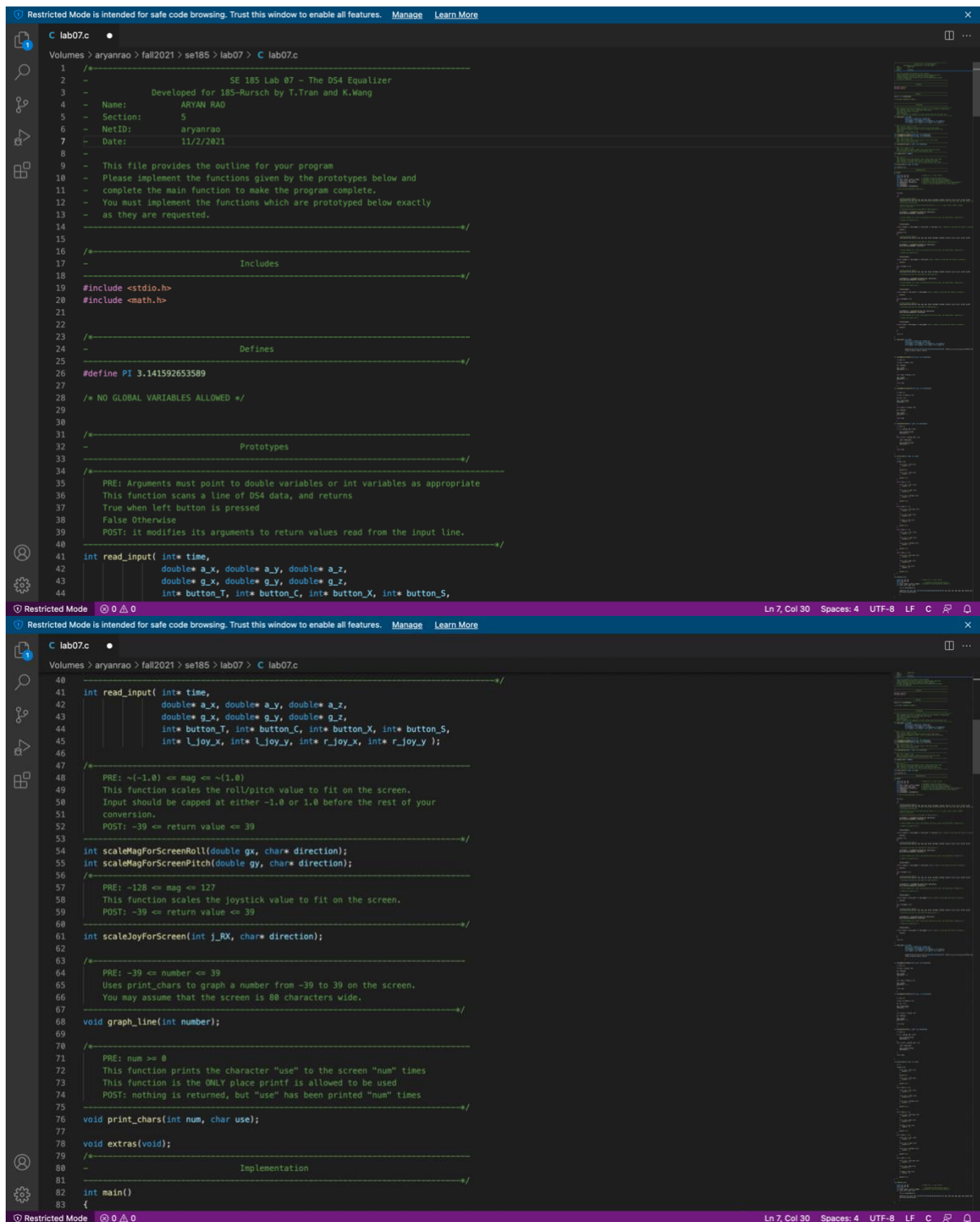
```
num = (gz/0.8)*40;
```

```
*direction = 'B';
```

2. As your experiment with the roll and pitch, what do you notice about the graph's behavior near the limits of its values?

The graph becomes more stable and constant near the limits of its value.

Source Code



```
1  /*
2  - SE 185 Lab 07 - The DS4 Equalizer
3  - Developed for 185-Rursch by T.Tran and K.Wang
4  - Name: ARIAN RAO
5  - Section: 5
6  - NetID: arianrao
7  - Date: 11/2/2021
8  -
9  - This file provides the outline for your program
10 - Please implement the functions given by the prototypes below and
11 - complete the main function to make the program complete.
12 - You must implement the functions which are prototyped below exactly
13 - as they are requested.
14 -
15 -
16 - Includes
17 -
18 -
19 #include <stdio.h>
20 #include <math.h>
21
22 -
23 - Defines
24 -
25 -
26 #define PI 3.141592653589
27
28 /* NO GLOBAL VARIABLES ALLOWED */
29
30 -
31 - Prototypes
32 -
33 -
34 /*
35 PRE: Arguments must point to double variables or int variables as appropriate
36 This function scans a line of DS4 data, and returns
37 True when left button is pressed
38 False otherwise
39 POST: it modifies its arguments to return values read from the input line.
40 */
41 int read_input( int* time,
42                double* a_x, double* a_y, double* a_z,
43                double* g_x, double* g_y, double* g_z,
44                int* button_T, int* button_C, int* button_X, int* button_S,
45                int* l_joy_x, int* l_joy_y, int* r_joy_x, int* r_joy_y );
46
47 /*
48 PRE: ~(-1.0) <= mag <= ~(1.0)
49 This function scales the roll/pitch value to fit on the screen.
50 Input should be capped at either -1.0 or 1.0 before the rest of your
51 conversion.
52 POST: ~39 <= return value <= 39
53 */
54 int scaleMagForScreenRoll(double gx, char* direction);
55 int scaleMagForScreenPitch(double gy, char* direction);
56
57 /*
58 PRE: ~128 <= mag <= 127
59 This function scales the joystick value to fit on the screen.
60 POST: ~39 <= return value <= 39
61 */
62 int scaleJoyForScreen(int j_RX, char* direction);
63
64 /*
65 PRE: ~39 <= number <= 39
66 Uses print_chars to graph a number from ~39 to 39 on the screen.
67 You may assume that the screen is 80 characters wide.
68 */
69 void graph_line(int number);
70
71 /*
72 PRE: num >= 0
73 This function prints the character "use" to the screen "num" times
74 This function is the ONLY place printf is allowed to be used
75 POST: nothing is returned, but "use" has been printed "num" times
76 */
77 void print_chars(int num, char use);
78
79 void extras(void);
80
81 - Implementation
82 -
83 int main()
84 {
```

```
1 // Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
2
3 C lab07.c
4
5 Volumes > aryanrao > fall2021 > se185 > lab07 > C lab07.c
6
7 81
82 int main()
83 {
84     double ax, ay, az;           /* Values of x, y, and z axis*/
85     double gx, gy, gz;
86     int t;                       /* Variable to hold the time value */
87     int b_Up, b_Down, b_Left, b_Right; /* Variables to hold the button statuses */
88     int j_LX, j_LY, j_RX, j_RY;   /* Variables to hold the joystick statuses */
89     int scaled_pitch, scaled_roll; /* Value of the roll/pitch adjusted to fit screen display */
90     int scaled_joy;              /* Value of joystick adjusted to fit screen display */
91     char direction = ' ';
92     int printNumRoll, printNumPitch;
93
94     /* Put pre-loop preparation code here */
95
96
97     while(1){
98     do
99     {
100
101         /* Scan a line of input */
102         read_input(&t,&ax,&ay,&az, &gx, &gy, &gz, &b_Up, &b_Right, &b_Down, &b_Left, &j_LX, &j_LY, &j_RX, &j_RY);
103         //scanf("%d, %lf, %lf, %lf, %lf, %lf, %lf, %lf, %lf, %d, %d, %d, %d, %d, %d, %d, %d, %d, %d, %d, %d, %d", &t, &ax, &ay, &az, &gx, &gy, &gz, &b_Up, &b_Left, &b_Down, &b_Right,
104         //&j_LX, &j_LY, &j_RX, &j_RY);
105
106         //printf("%d, %lf, %lf, %lf, %d, %d, %d, %d, %d, %d, %d, %d, %d, %d, %d, %d, %d, %d, %d, %d, %d, %d\n", t, x, y, z, b_Up, b_Left, b_Down, b_Right,
107         //j_LX, j_LY, j_RX, j_RY);
108
109         /* Calculate and scale for pitch AND roll AND joystick */
110
111         printNumRoll = scaleMagForScreenRoll(gx, &direction);
112         print_chars(printNumRoll, direction);
113
114
115         /* Switch between roll, pitch, and joystick with the up, down, and right button, respectively */
116
117         /* Output your graph line */
118
119         fflush(stdout);
120
121     } while (b_Down != 1 && b_Right != 1 && b_Left != 1 && b_Up != 1); /* Modify to stop when left button is pressed */
122
123     extras();
124
125
126
127
128
129 if(b_Up == 1){
130 do
131 {
132
133     /* Scan a line of input */
134     read_input(&t,&ax,&ay,&az, &gx, &gy, &gz, &b_Up, &b_Right, &b_Down, &b_Left, &j_LX, &j_LY, &j_RX, &j_RY);
135
136
137     /* Calculate and scale for pitch AND roll AND joystick */
138
139     printNumRoll = scaleMagForScreenRoll(gx, &direction);
140     print_chars(printNumRoll, direction);
141
142
143     /* Switch between roll, pitch, and joystick with the up, down, and right button, respectively */
144
145     /* Output your graph line */
146
147     fflush(stdout);
148
149 } while (b_Down != 1 && b_Right != 1 && b_Left != 1); /* Modify to stop when left button is pressed */
150
151 extras();
152
153
154
155
156
157 else if(b_Down == 1){
158 do
159 {
160
161     /* Scan a line of input */
162     read_input(&t,&ax,&ay,&az, &gx, &gy, &gz, &b_Up, &b_Right, &b_Down, &b_Left, &j_LX, &j_LY, &j_RX, &j_RY);
163
164
165     /* Calculate and scale for pitch AND roll AND joystick */
166
167     printNumPitch = scaleMagForScreenPitch(gz, &direction);
168     print_chars(printNumPitch, direction);
169
170
171     /* Switch between roll, pitch, and joystick with the up, down, and right button, respectively */
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
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
8
```

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

C lab07.c

Volumes > arianrao > fall2021 > se185 > lab07 > C lab07.c

```
160     printNumPitch = scaleMagForScreenPitch(gz, &direction);
161     print_chars(printNumPitch, direction);
162
163     /* Switch between roll, pitch, and joystick with the up, down, and right button, respectively */
164
165     /* Output your graph line */
166
167     fflush(stdout);
168
169 } while (b_Up != 1 && b_Left != 1 && b_Right != 1); /* Modify to stop when left button is pressed */
170
171 extras();
172 }
173
174
175 else if(b_Right == 1){
176 do
177 {
178     /* Scan a line of input */
179     read_input(&t,&ax,&ay,&az, &gx, &gy, &gz, &b_Up, &b_Right, &b_Down, &b_Left, &j_LX, &j_LY, &j_RX, &j_RY);
180
181     /* Calculate and scale for pitch AND roll AND joystick */
182
183
184     printNumPitch = scaleJoyForScreen(j_RX, &direction);
185     print_chars(printNumPitch, direction);
186
187     /* Switch between roll, pitch, and joystick with the up, down, and right button, respectively */
188
189     /* Output your graph line */
190
191     fflush(stdout);
192
193 } while (b_Up != 1 && b_Down != 1 && b_Right != 1); /* Modify to stop when left button is pressed */
194
195 extras();
196
197 }
198
199 }
200
201 return 0;
202
203 }
```

Ln 7, Col 30 Spaces: 4 UTF-8 LF C

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

C lab07.c

Volumes > arianrao > fall2021 > se185 > lab07 > C lab07.c

```
205 int read_input( int* time,
206                double* a_x, double* a_y, double* a_z,
207                double* g_x, double* g_y, double* g_z,
208                int* button_T, int* button_C, int* button_X, int* button_S,
209                int* l_joy_x, int* l_joy_y, int* r_joy_x, int* r_joy_y ){
210
211
212     scanf("%d,%lf,%lf,%lf,%lf,%lf,%d,%d,%d,%d,%d,%d,%d,%d", time,a_x,a_y,a_z,g_x,g_y,g_z,button_T,button_C,button_X,button_S,
213           l_joy_x,l_joy_y,r_joy_x,r_joy_y);
214
215 }
216
217
218 int scaleMagForScreenRoll(double gx, char* direction){
219
220     int num = 0;
221
222     if( gx >= -1 && gx < 0){
223
224         gx = fabs(gx);
225
226         num = gx*40;
227         *direction = 'R';
228     }
229
230     else if(gx > 0 && gx <= 1){
231
232         num = gx*40;
233         *direction = 'L';
234     }
235
236     return num;
237
238 }
239
240
241 int scaleMagForScreenPitch(double gz, char* direction){
242
243
244     int num = 0;
245
246     if( gz > 0.2 && gz <= 1){
247
248         gz = gz - 0.2;
```

Ln 7, Col 30 Spaces: 4 UTF-8 LF C

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

C lab07.c

Volumes > aryanrao > fall2021 > se185 > lab07 > C lab07.c

```
241 int scaleMagForScreenPitch(double gz, char* direction){
242
243
244     int num = 0;
245
246     if( gz > 0.2 && gz <= 1){
247
248         gz = gz - 0.2;
249
250         num = (gz/0.8)*40;
251         *direction = 'B';
252     }
253
254     else if(gz >= -1 && gz < 0){
255
256         gz = fabs(gz);
257
258         num = gz*40;
259         *direction = 'F';
260     }
261
262     return num;
263 }
264
265
266
267 int scaleJoyForScreen(int j_RX, char* direction){
268
269     int num = 0;
270
271     if (1 < j_RX && j_RX < 127){
272
273         num=(j_RX/127.0)*40;
274         *direction='R';
275     }
276
277     else if(-127 < j_RX && j_RX < -1){
278
279         j_RX = fabs(j_RX);
280
281         num=(j_RX/127.0)*40;
282         *direction='L';
283     }
284 }
```

Ln 7, Col 30 Spaces: 4 UTF-8 LF C

Restricted Mode

0 0 0

C lab07.c

Volumes > aryanrao > fall2021 > se185 > lab07 > C lab07.c

```
291 void print_chars(int num, char use){
292
293     int i;
294
295     if(num == 0){
296
297         for(i = 0; i < 40; i++){
298             printf(" ");
299         }
300
301         printf("0");
302
303         for(i = 0; i < 40; i++){
304             printf(" ");
305         }
306
307         printf("\n");
308     }
309
310     else if(use == 'R'){
311
312         for(i = 0; i < 40 ; i++){
313             printf(" ");
314         }
315
316         for(i = 0; i < num; i++){
317             printf("R");
318         }
319
320         for(i = 0; i < 40-num; i++){
321             printf(" ");
322         }
323
324         printf("\n");
325     }
326
327     else if(use == 'L'){
328
329         for(i = 0; i < 40- num; i++){
330             printf(" ");
331         }
332
333         for(i = 0; i < num; i++){
```

Ln 7, Col 30 Spaces: 4 UTF-8 LF C

Restricted Mode

0 0 0

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

C lab07.c

Volumes > arianrao > fall2021 > se185 > lab07 > C lab07.c

```
332     }
333
334     for(i = 0; i < num; i++){
335         printf("L");
336     }
337
338     for(i=0; i < 40; i++){
339         printf(" ");
340     }
341
342     printf("\n");
343 }
344
345 else if(use == 'P'){
346
347     for(i = 0; i < 40 ; i++){
348         printf(" ");
349     }
350
351     for(i = 0; i < num; i++){
352         printf("P");
353     }
354
355     for(i = 0; i < 40-num; i++){
356         printf(" ");
357     }
358
359     printf("\n");
360 }
361
362 else if(use == 'B'){
363
364     for(i = 0; i < 40- num; i++){
365         printf(" ");
366     }
367
368     for(i = 0; i < num; i++){
369         printf("B");
370     }
371
372     for(i=0; i < 40; i++){
373         printf(" ");
374     }
375 }
```

Ln 7, Col 30 Spaces: 4 UTF-8 LF C

C lab07.c

Volumes > arianrao > fall2021 > se185 > lab07 > C lab07.c

```
369     printf("B");
370 }
371
372     for(i=0; i < 40; i++){
373         printf(" ");
374     }
375
376     printf("\n");
377 }
378 }
379
380 void extras(void){
381
382     double ax, ay, az;          /* Values of x, y, and z axis*/
383     double gx, gy, gz;
384     int t = 0;                  /* Variable to hold the time value */
385     int b_Up, b_Down, b_Left, b_Right; /* Variables to hold the button statuses */
386     int j_LX, j_LY, j_RX, j_RY;
387
388     for(int i=0; i<100; i++){
389
390         scanf("%d %lf %lf %lf %lf %d %d %d %d %d %d", &t, &ax, &ay, &az, &gx, &gy, &gz, &b_Up, &b_Left, &b_Down, &b_Right,
391             &j_LX, &j_LY, &j_RX, &j_RY);
392     }
393 }
394
395 }
```

Ln 7, Col 30 Spaces: 4 UTF-8 LF C

OUTPUT

[illegible]