

# VG100 — Introduction to Engineering

## Project 1 Report (Team 3)

### Rubric

- Game Design (10 pts)
- Code Quality (50 pts)
- Readme (15 pts)
- Personal work (20 pts)

## 1 Game Design

Not included in this report.

## 2 Code Quality

Your total score of this part is 42/50.

All related information is listed below:

2 point(s) **deduction**, hard coded, in file [Block.elm](#), lines 156-305.

```
156     blockArray -5 3 7 14 1
157     -- 1
158     ++ blockArray -5 -5 4 2 1
159     --2
160     ++ blockArray 12 2 27 14 1
161     --3
162     ++ blockArray 14 -4 27 1 1
163     --4
164     ++ blockArray 17 -30 27 -5 1
165     --5
...
296     expList1 =
297         [ ( 6, -3 ), (14, -14), ( 4, 2), (6, -3), (15, -2) ]
298
299
300     expList2 =
301         [ ( 4, 4 ), ( 16, 3 ), ( 7, -2 ), ( 13, -3 ), ( 4, -8 ), ( 9, -9 ), ( 18, -3 ) ]
302
303
304     expList3 =
305         [ ( 12, 3 ), ( 14, 0 ), ( 13, -10 ), ( 6, 0 ), ( 9, -11 ) ]
```

1 point(s) **deduction**, useless let in, in file [Block.elm](#), lines 59-64.

```
59     rect =
60         { a = ( x, y )
61         , b = ( x, y + gameParams.xStep )
62         , c = ( x + gameParams.xStep, y + gameParams.yStep )
63         , d = ( x + gameParams.xStep, y )
64         }
```

1 point(s) deduction, hard coded, in file [Bob2.elm](#), lines 23-125.

```
23      [ path [ d "M1893.65,118c0,24.29,34,35-6.5,35S1747,142.31,1747,118s40.41-17,80.91-17,65.74-7.3,65.74,17", transform
    ↪      "translate(-233 -11)", fill "#ebffff" ]
24      []
25      ]
26      , path [ d "M836.07,426.76c4.85-100,43.15-176.44,59.18-204.37a15.11,15.11,0,0,1,19-6.43A15.61,15.61,0,0,1,923,235.05l-
    ↪      67.81,166.3", transform "translate(-233 -11)", fill "#ffdbc0"
    ↪      ]
27      []
28      , path [ d "M873.23,321.17l1.22,66c7.86,4.3,10.57,14.33,6.69,24.67l-15.79,41.88c-5.5,14.59-21.65,23.79-33.64,19.16l-3.24-1.25c-
    ↪      9.65-3.72-13.13-15.25-8.11-26.88l17.81-41.3c6.43-14.92,23.6-23.21,35.06-16.94", transform "translate(-233 -11)", fill
    ↪      "#7d6450" ]
29      []
30      , path [ d "M828.06,339.38c17.53,29,26.88,41.14,11.17,50.63s-22.38-4.94-39.9-34-16.62-20.7-.91-30.19,12.12-15.5,29.64,13.52",
    ↪      transform "translate(-233 -11)", fill "#ffdbc0" ]
31      []
32      , path [ d "M863.32,696.49c-13.14,144.06,13.13,128.39,22.14,110,1.43-2.91,1.41-15.41,1.7-18.64L910,621.49c.67-7.41-10.95-
    ↪      31.53-18.47-32.21l-8.83-.81a13.56,13.56,0,0,0-14.86,12.74Z", transform "translate(-233 -11)", fill "#874209"
    ↪      ]
...
116      []
117      , path [ d "M742.4,259.88c-31.2,26.63-39.71,66.49-19,89l113-96.46C815.7,229.91,773.61,233.24,742.4,259.88Z", transform
    ↪      "translate(-233 -11)", fill "#ffc90e" ]
118      []
119      , polygon [ points "622.07 239.8 490.7 353.27 482.36 344.29 613.72 230.82 622.07 239.8", fill "#ff7f27", stroke "#000",
    ↪      strokeMiterlimit "10" ]
120      []
121      , ellipse [ cx "831.16", cy "235.4", rx "22.35", ry "23.28", transform "translate(139.37 956.35) rotate(-73.73)", fill "#00a284",
    ↪      stroke "#000", strokeMiterlimit "10", opacity "0.36" ]
122      []
123      , ellipse [ cx "830.65", cy "236.39", rx "12.97", ry "17.33", transform "translate(-180.25 600.52) rotate(-41.63)", fill "#fff",
    ↪      stroke "#000", strokeMiterlimit "10" ]
124      []
125      ]
```

1 point(s) deduction, too long function, in file [GameView.elm](#), lines 52-188.

```
52  renderGame : Model -> Html Msg
53  renderGame model =
54      let
55          yStep =
56              gameParams.yStep
57
58          coverOpacity =
59              if model.currentLevel <= 3 && model.pad.y > negate 16 * yStep then
60                  0
61
62      ...
63
64      ...
65
66      ...
67
68      ...
69
70      ...
71
72      ...
73
74      ...
75
76      ...
77
78      ...
79      "Press Enter back to levels page"
80
81      ...
82      else
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      ...
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      ...
818
819      ...
820
821      ...
822
823      ...
824
825      ...
826
827      ...
828
829      ...
830
831      ...
832
833      ...
834
835      ...
836
837      ...
838
839      ...
840
841      ...
842
843      ...
844
845      ...
846
847      ...
848
849      ...
850
851      ...
852
853      ...
854
855      ...
856
857      ...
858
859      ...
860
861      ...
862
863      ...
864
865      ...
866
867      ...
868
869      ...
870
871      ...
872
873      ...
874
875      ...
876
877      ...
878
879      ...
880
881      ...
882
883      ...
884
885      ...
886
887      ...
888
889      ...
890
891      ...
892
893      ...
894
895      ...
896
897      ...
898
899      ...
900
901      ...
902
903      ...
904
905      ...
906
907      ...
908
909      ...
910
911      ...
912
913      ...
914
915      ...
916
917      ...
918
919      ...
920
921      ...
922
923      ...
924
925      ...
926
927      ...
928
929      ...
930
931      ...
932
933      ...
934
935      ...
936
937      ...
938
939      ...
940
941      ...
942
943      ...
944
945      ...
946
947      ...
948
949      ...
950
951      ...
952
953      ...
954
955      ...
956
957      ...
958
959      ...
960
961      ...
962
963      ...
964
965      ...
966
967      ...
968
969      ...
970
971      ...
972
973      ...
974
975      ...
976
977      ...
978
979      ...
980
981      ...
982
983      ...
984
985      ...
986
987      ...
988
989      ...
990
991      ...
992
993      ...
994
995      ...
996
997      ...
998
999      ...
1000
```

```

183         )
184         |> Svg.text
185     ]
186     , listenMusic model
187 ]
188 ]

```

1 point(s) **deduction**, duplicate code, in file [Geometry.elm](#), lines 198-227.

```

198     if nx_1 > 0 && ny_1 > 0 then
199         Intersect { quadrant = 1, lambda = lambda_1, axis = ( nx_1, ny_1 ) }
200
201     else if nx_1 < 0 && ny_1 > 0 then
202         Intersect { quadrant = 2, lambda = lambda_1, axis = ( nx_1, ny_1 ) }
203
204     else if nx_1 < 0 && ny_1 < 0 then
205         Intersect { quadrant = 3, lambda = lambda_1, axis = ( nx_1, ny_1 ) }
206
207     else if nx_1 > 0 && ny_1 < 0 then
208
209     ...
210
211     Intersect { quadrant = 2, lambda = lambda_2, axis = ( nx_2, ny_2 ) }
212
213     else if nx_2 < 0 && ny_2 < 0 then
214         Intersect { quadrant = 3, lambda = lambda_2, axis = ( nx_2, ny_2 ) }
215
216     else if nx_2 > 0 && ny_2 < 0 then
217         Intersect { quadrant = 4, lambda = lambda_2, axis = ( nx_2, ny_2 ) }
218
219     else
220         NotIntersect
221
222

```

1 point(s) **deduction**, duplicate code, in file [Geometry.elm](#), lines 392-414.

```

392     inter1 =
393         circleIntersectLine c1 line |> filterByQuarant 1
394
395     inter3 =
396         circleIntersectLine c3 line |> filterByQuarant 2
397
398     inter5 =
399         circleIntersectLine c5 line |> filterByQuarant 3
400
401     inter7 =
402         circleIntersectLine c7 line |> filterByQuarant 4
403
404     inter2 =
405         lineIntersectLine line l2
406
407     inter4 =
408         lineIntersectLine line l4
409
410     inter6 =
411         lineIntersectLine line l6
412

```

```

413     inter8 =
414         lineIntersectLine line l8

```

1 point(s) **deduction**, useless let in, in file [Geometry.elm](#), lines 422-427.

```

422     prior =
423         if toLambda prior_ <= 1 then
424             prior_
425
426     else
427         NotIntersect

```

1 point(s) **deduction**, duplicate code, in file [Store.elm](#), lines 111-134.

```

111 filledBlock : Html Msg
112 filledBlock =
113     div
114         [ style "background" "#AE01AB"
115         , style "border" "groove"
116         , style "width" "25px"
117         , style "height" "40px"
118         , style "margin" "auto"
119         , style "display" "inline-block"
120         ]
121     []
122
123
124 emptyBlock : Html Msg
125 emptyBlock =
126     div
127         [ style "border-color" "#AE01AB"
128         , style "border" "groove"
129         , style "width" "25px"
130         , style "height" "40px"
131         , style "margin" "auto"
132         , style "display" "inline-block"
133         ]
134     []

```

1 point(s) **deduction**, duplicate code and usage of continuous if instead of case, in file [Update.elm](#), lines 17-48.

```

17     if n == 1 then
18         { model
19             | block = map1__
20             , currentLevel = 1
21             , ball = initBall1
22             , pad = initPad
23             , waterLevel = 0.0
24             , waterLevelRiseSpeed = initWater 1
25             , targetPos = initTargetpos1
26         }
...
39     else
40         { model

```

```
41         | block = map3_  
42         , currentLevel = 3  
43         , ball = initBall3  
44         , pad = initPad  
45         , waterLevel = 0.0  
46         , waterLevelRiseSpeed = initWater 3  
47         , targetPos = initTargetpos3  
48     }
```

2 point(s) **bonus**, some documentations found.

### **3 Readme**

Not included in this report.

### **4 Personal work**

Not included in this report.