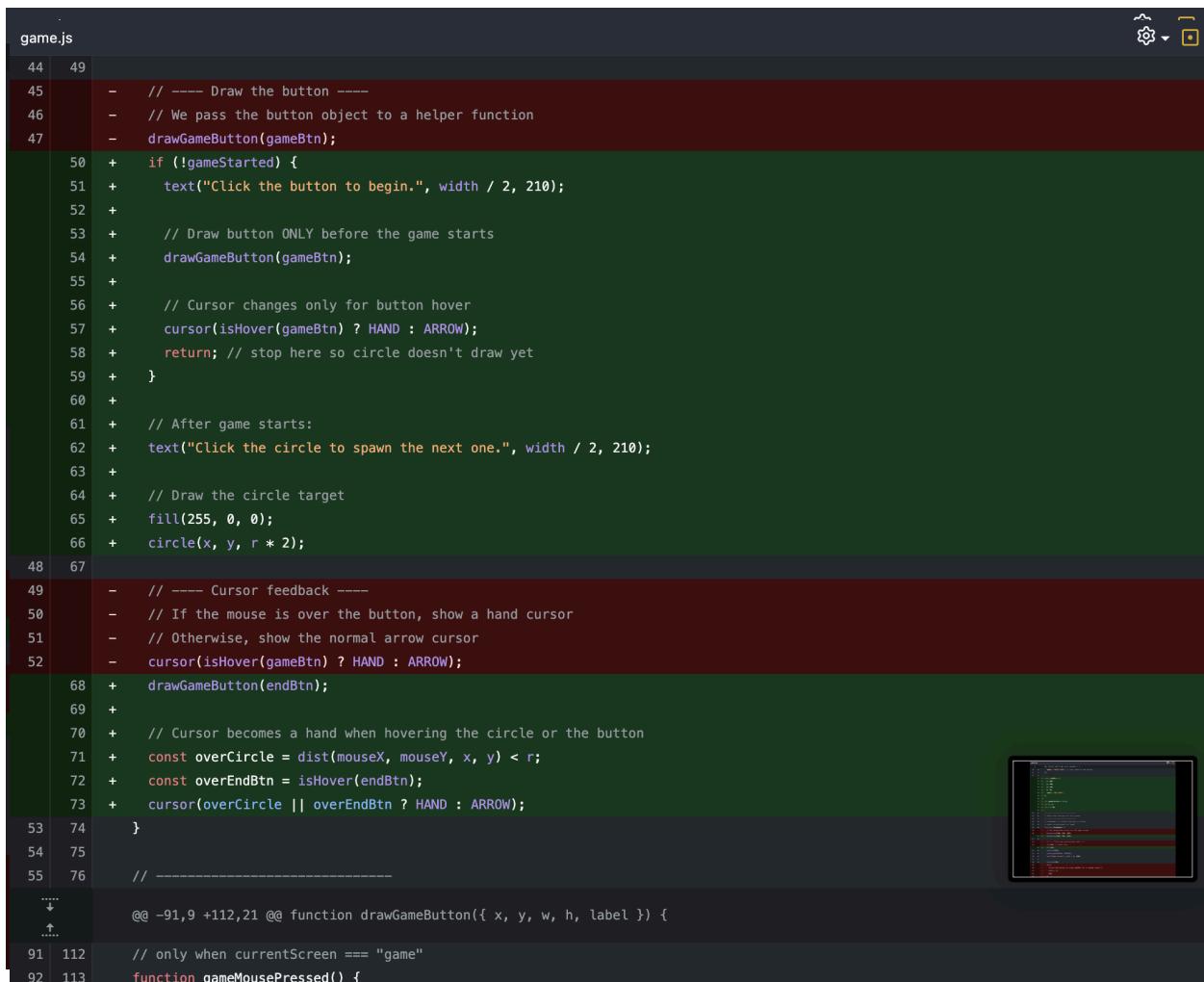


# Process & Decision Documentation

## Side Quests and A4 (Individual Work)

For my sidequest for this week, I focused on changing the functionality of the win/lose example to a version that allows users to play the game by detecting the circles as they spawn on the screen. Users can also end the game and ‘lose’ when they decide to.

One significant design decision I had to make was including a ‘lose’ functionality. Unlike the original example where losing or winning was based on a random number, my game’s functionality did not inherently have a way to lose or exit out of the game screen. Due to time constraints, I simply included “endbtn” similar to “gamebtn” that loaded up the ‘lose’ screen when pressed.



```
game.js
44 49 - // ---- Draw the button ----
45 - // We pass the button object to a helper function
46 - drawGameButton(gameBtn);
47
48 50 + if (!gameStarted) {
49 +   text("Click the button to begin.", width / 2, 210);
50 +
51 +   // Draw button ONLY before the game starts
52 +   drawGameButton(gameBtn);
53 +
54 +   // Cursor changes only for button hover
55 +   cursor(isHover(gameBtn) ? HAND : ARROW);
56 +   return; // stop here so circle doesn't draw yet
57 +
58 + }
59 +
60 +
61 + // After game starts:
62 + text("Click the circle to spawn the next one.", width / 2, 210);
63 +
64 + // Draw the circle target
65 + fill(255, 0, 0);
66 + circle(x, y, r * 2);
67
68 - // ---- Cursor feedback ----
69 -
70 - // If the mouse is over the button, show a hand cursor
71 - // Otherwise, show the normal arrow cursor
72 - cursor(isHover(gameBtn) ? HAND : ARROW);
73
74 + drawGameButton(endBtn);
75 +
76 + // Cursor becomes a hand when hovering the circle or the button
77 + const overCircle = dist(mouseX, mouseY, x, y) < r;
78 + const overEndBtn = isHover(endBtn);
79 + cursor(overCircle || overEndBtn ? HAND : ARROW);
80
81 +
82 +-----+
83 @@ -91,9 +112,21 @@ function drawGameButton({ x, y, w, h, label }) {
84
85   // only when currentScreen === "game"
86   function gameMousePressed() {
87
88     if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
89       if (isHover(gameBtn)) {
90         if (gameStarted) {
91           end();
92         } else {
93           gameStarted = true;
94           start();
95         }
96       } else {
97         if (gameStarted) {
98           end();
99         } else {
100          start();
101        }
102      }
103    }
104  }
105
106  // Draw the button
107  drawGameButton(gameBtn);
108
109  // Draw the circle target
110  fill(255, 0, 0);
111  circle(x, y, r * 2);
112
113  // only when currentScreen === "game"
114  function gameMousePressed() {
115
116    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
117      if (isHover(gameBtn)) {
118        if (gameStarted) {
119          end();
120        } else {
121          start();
122        }
123      } else {
124        if (gameStarted) {
125          end();
126        } else {
127          start();
128        }
129      }
130    }
131  }
132
133  // Draw the button
134  drawGameButton(gameBtn);
135
136  // Draw the circle target
137  fill(255, 0, 0);
138  circle(x, y, r * 2);
139
140  // only when currentScreen === "game"
141  function gameMousePressed() {
142
143    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
144      if (isHover(gameBtn)) {
145        if (gameStarted) {
146          end();
147        } else {
148          start();
149        }
150      } else {
151        if (gameStarted) {
152          end();
153        } else {
154          start();
155        }
156      }
157    }
158  }
159
160  // Draw the button
161  drawGameButton(gameBtn);
162
163  // Draw the circle target
164  fill(255, 0, 0);
165  circle(x, y, r * 2);
166
167  // only when currentScreen === "game"
168  function gameMousePressed() {
169
170    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
171      if (isHover(gameBtn)) {
172        if (gameStarted) {
173          end();
174        } else {
175          start();
176        }
177      } else {
178        if (gameStarted) {
179          end();
180        } else {
181          start();
182        }
183      }
184    }
185  }
186
187  // Draw the button
188  drawGameButton(gameBtn);
189
190  // Draw the circle target
191  fill(255, 0, 0);
192  circle(x, y, r * 2);
193
194  // only when currentScreen === "game"
195  function gameMousePressed() {
196
197    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
198      if (isHover(gameBtn)) {
199        if (gameStarted) {
200          end();
201        } else {
202          start();
203        }
204      } else {
205        if (gameStarted) {
206          end();
207        } else {
208          start();
209        }
210      }
211    }
212  }
213
214  // Draw the button
215  drawGameButton(gameBtn);
216
217  // Draw the circle target
218  fill(255, 0, 0);
219  circle(x, y, r * 2);
220
221  // only when currentScreen === "game"
222  function gameMousePressed() {
223
224    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
225      if (isHover(gameBtn)) {
226        if (gameStarted) {
227          end();
228        } else {
229          start();
230        }
231      } else {
232        if (gameStarted) {
233          end();
234        } else {
235          start();
236        }
237      }
238    }
239  }
240
241  // Draw the button
242  drawGameButton(gameBtn);
243
244  // Draw the circle target
245  fill(255, 0, 0);
246  circle(x, y, r * 2);
247
248  // only when currentScreen === "game"
249  function gameMousePressed() {
250
251    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
252      if (isHover(gameBtn)) {
253        if (gameStarted) {
254          end();
255        } else {
256          start();
257        }
258      } else {
259        if (gameStarted) {
260          end();
261        } else {
262          start();
263        }
264      }
265    }
266  }
267
268  // Draw the button
269  drawGameButton(gameBtn);
270
271  // Draw the circle target
272  fill(255, 0, 0);
273  circle(x, y, r * 2);
274
275  // only when currentScreen === "game"
276  function gameMousePressed() {
277
278    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
279      if (isHover(gameBtn)) {
280        if (gameStarted) {
281          end();
282        } else {
283          start();
284        }
285      } else {
286        if (gameStarted) {
287          end();
288        } else {
289          start();
290        }
291      }
292    }
293  }
294
295  // Draw the button
296  drawGameButton(gameBtn);
297
298  // Draw the circle target
299  fill(255, 0, 0);
300  circle(x, y, r * 2);
301
302  // only when currentScreen === "game"
303  function gameMousePressed() {
304
305    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
306      if (isHover(gameBtn)) {
307        if (gameStarted) {
308          end();
309        } else {
310          start();
311        }
312      } else {
313        if (gameStarted) {
314          end();
315        } else {
316          start();
317        }
318      }
319    }
320  }
321
322  // Draw the button
323  drawGameButton(gameBtn);
324
325  // Draw the circle target
326  fill(255, 0, 0);
327  circle(x, y, r * 2);
328
329  // only when currentScreen === "game"
330  function gameMousePressed() {
331
332    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
333      if (isHover(gameBtn)) {
334        if (gameStarted) {
335          end();
336        } else {
337          start();
338        }
339      } else {
340        if (gameStarted) {
341          end();
342        } else {
343          start();
344        }
345      }
346    }
347  }
348
349  // Draw the button
350  drawGameButton(gameBtn);
351
352  // Draw the circle target
353  fill(255, 0, 0);
354  circle(x, y, r * 2);
355
356  // only when currentScreen === "game"
357  function gameMousePressed() {
358
359    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
360      if (isHover(gameBtn)) {
361        if (gameStarted) {
362          end();
363        } else {
364          start();
365        }
366      } else {
367        if (gameStarted) {
368          end();
369        } else {
370          start();
371        }
372      }
373    }
374  }
375
376  // Draw the button
377  drawGameButton(gameBtn);
378
379  // Draw the circle target
380  fill(255, 0, 0);
381  circle(x, y, r * 2);
382
383  // only when currentScreen === "game"
384  function gameMousePressed() {
385
386    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
387      if (isHover(gameBtn)) {
388        if (gameStarted) {
389          end();
390        } else {
391          start();
392        }
393      } else {
394        if (gameStarted) {
395          end();
396        } else {
397          start();
398        }
399      }
400    }
401  }
402
403  // Draw the button
404  drawGameButton(gameBtn);
405
406  // Draw the circle target
407  fill(255, 0, 0);
408  circle(x, y, r * 2);
409
410  // only when currentScreen === "game"
411  function gameMousePressed() {
412
413    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
414      if (isHover(gameBtn)) {
415        if (gameStarted) {
416          end();
417        } else {
418          start();
419        }
420      } else {
421        if (gameStarted) {
422          end();
423        } else {
424          start();
425        }
426      }
427    }
428  }
429
430  // Draw the button
431  drawGameButton(gameBtn);
432
433  // Draw the circle target
434  fill(255, 0, 0);
435  circle(x, y, r * 2);
436
437  // only when currentScreen === "game"
438  function gameMousePressed() {
439
440    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
441      if (isHover(gameBtn)) {
442        if (gameStarted) {
443          end();
444        } else {
445          start();
446        }
447      } else {
448        if (gameStarted) {
449          end();
450        } else {
451          start();
452        }
453      }
454    }
455  }
456
457  // Draw the button
458  drawGameButton(gameBtn);
459
460  // Draw the circle target
461  fill(255, 0, 0);
462  circle(x, y, r * 2);
463
464  // only when currentScreen === "game"
465  function gameMousePressed() {
466
467    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
468      if (isHover(gameBtn)) {
469        if (gameStarted) {
470          end();
471        } else {
472          start();
473        }
474      } else {
475        if (gameStarted) {
476          end();
477        } else {
478          start();
479        }
480      }
481    }
482  }
483
484  // Draw the button
485  drawGameButton(gameBtn);
486
487  // Draw the circle target
488  fill(255, 0, 0);
489  circle(x, y, r * 2);
490
491  // only when currentScreen === "game"
492  function gameMousePressed() {
493
494    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
495      if (isHover(gameBtn)) {
496        if (gameStarted) {
497          end();
498        } else {
499          start();
500        }
501      } else {
502        if (gameStarted) {
503          end();
504        } else {
505          start();
506        }
507      }
508    }
509  }
510
511  // Draw the button
512  drawGameButton(gameBtn);
513
514  // Draw the circle target
515  fill(255, 0, 0);
516  circle(x, y, r * 2);
517
518  // only when currentScreen === "game"
519  function gameMousePressed() {
520
521    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
522      if (isHover(gameBtn)) {
523        if (gameStarted) {
524          end();
525        } else {
526          start();
527        }
528      } else {
529        if (gameStarted) {
530          end();
531        } else {
532          start();
533        }
534      }
535    }
536  }
537
538  // Draw the button
539  drawGameButton(gameBtn);
540
541  // Draw the circle target
542  fill(255, 0, 0);
543  circle(x, y, r * 2);
544
545  // only when currentScreen === "game"
546  function gameMousePressed() {
547
548    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
549      if (isHover(gameBtn)) {
550        if (gameStarted) {
551          end();
552        } else {
553          start();
554        }
555      } else {
556        if (gameStarted) {
557          end();
558        } else {
559          start();
560        }
561      }
562    }
563  }
564
565  // Draw the button
566  drawGameButton(gameBtn);
567
568  // Draw the circle target
569  fill(255, 0, 0);
570  circle(x, y, r * 2);
571
572  // only when currentScreen === "game"
573  function gameMousePressed() {
574
575    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
576      if (isHover(gameBtn)) {
577        if (gameStarted) {
578          end();
579        } else {
580          start();
581        }
582      } else {
583        if (gameStarted) {
584          end();
585        } else {
586          start();
587        }
588      }
589    }
590  }
591
592  // Draw the button
593  drawGameButton(gameBtn);
594
595  // Draw the circle target
596  fill(255, 0, 0);
597  circle(x, y, r * 2);
598
599  // only when currentScreen === "game"
600  function gameMousePressed() {
601
602    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
603      if (isHover(gameBtn)) {
604        if (gameStarted) {
605          end();
606        } else {
607          start();
608        }
609      } else {
610        if (gameStarted) {
611          end();
612        } else {
613          start();
614        }
615      }
616    }
617  }
618
619  // Draw the button
620  drawGameButton(gameBtn);
621
622  // Draw the circle target
623  fill(255, 0, 0);
624  circle(x, y, r * 2);
625
626  // only when currentScreen === "game"
627  function gameMousePressed() {
628
629    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
630      if (isHover(gameBtn)) {
631        if (gameStarted) {
632          end();
633        } else {
634          start();
635        }
636      } else {
637        if (gameStarted) {
638          end();
639        } else {
640          start();
641        }
642      }
643    }
644  }
645
646  // Draw the button
647  drawGameButton(gameBtn);
648
649  // Draw the circle target
650  fill(255, 0, 0);
651  circle(x, y, r * 2);
652
653  // only when currentScreen === "game"
654  function gameMousePressed() {
655
656    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
657      if (isHover(gameBtn)) {
658        if (gameStarted) {
659          end();
660        } else {
661          start();
662        }
663      } else {
664        if (gameStarted) {
665          end();
666        } else {
667          start();
668        }
669      }
670    }
671  }
672
673  // Draw the button
674  drawGameButton(gameBtn);
675
676  // Draw the circle target
677  fill(255, 0, 0);
678  circle(x, y, r * 2);
679
680  // only when currentScreen === "game"
681  function gameMousePressed() {
682
683    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
684      if (isHover(gameBtn)) {
685        if (gameStarted) {
686          end();
687        } else {
688          start();
689        }
690      } else {
691        if (gameStarted) {
692          end();
693        } else {
694          start();
695        }
696      }
697    }
698  }
699
700  // Draw the button
701  drawGameButton(gameBtn);
702
703  // Draw the circle target
704  fill(255, 0, 0);
705  circle(x, y, r * 2);
706
707  // only when currentScreen === "game"
708  function gameMousePressed() {
709
710    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
711      if (isHover(gameBtn)) {
712        if (gameStarted) {
713          end();
714        } else {
715          start();
716        }
717      } else {
718        if (gameStarted) {
719          end();
720        } else {
721          start();
722        }
723      }
724    }
725  }
726
727  // Draw the button
728  drawGameButton(gameBtn);
729
730  // Draw the circle target
731  fill(255, 0, 0);
732  circle(x, y, r * 2);
733
734  // only when currentScreen === "game"
735  function gameMousePressed() {
736
737    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
738      if (isHover(gameBtn)) {
739        if (gameStarted) {
740          end();
741        } else {
742          start();
743        }
744      } else {
745        if (gameStarted) {
746          end();
747        } else {
748          start();
749        }
750      }
751    }
752  }
753
754  // Draw the button
755  drawGameButton(gameBtn);
756
757  // Draw the circle target
758  fill(255, 0, 0);
759  circle(x, y, r * 2);
760
761  // only when currentScreen === "game"
762  function gameMousePressed() {
763
764    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
765      if (isHover(gameBtn)) {
766        if (gameStarted) {
767          end();
768        } else {
769          start();
770        }
771      } else {
772        if (gameStarted) {
773          end();
774        } else {
775          start();
776        }
777      }
778    }
779  }
780
781  // Draw the button
782  drawGameButton(gameBtn);
783
784  // Draw the circle target
785  fill(255, 0, 0);
786  circle(x, y, r * 2);
787
788  // only when currentScreen === "game"
789  function gameMousePressed() {
790
791    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
792      if (isHover(gameBtn)) {
793        if (gameStarted) {
794          end();
795        } else {
796          start();
797        }
798      } else {
799        if (gameStarted) {
800          end();
801        } else {
802          start();
803        }
804      }
805    }
806  }
807
808  // Draw the button
809  drawGameButton(gameBtn);
810
811  // Draw the circle target
812  fill(255, 0, 0);
813  circle(x, y, r * 2);
814
815  // only when currentScreen === "game"
816  function gameMousePressed() {
817
818    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
819      if (isHover(gameBtn)) {
820        if (gameStarted) {
821          end();
822        } else {
823          start();
824        }
825      } else {
826        if (gameStarted) {
827          end();
828        } else {
829          start();
830        }
831      }
832    }
833  }
834
835  // Draw the button
836  drawGameButton(gameBtn);
837
838  // Draw the circle target
839  fill(255, 0, 0);
840  circle(x, y, r * 2);
841
842  // only when currentScreen === "game"
843  function gameMousePressed() {
844
845    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
846      if (isHover(gameBtn)) {
847        if (gameStarted) {
848          end();
849        } else {
850          start();
851        }
852      } else {
853        if (gameStarted) {
854          end();
855        } else {
856          start();
857        }
858      }
859    }
860  }
861
862  // Draw the button
863  drawGameButton(gameBtn);
864
865  // Draw the circle target
866  fill(255, 0, 0);
867  circle(x, y, r * 2);
868
869  // only when currentScreen === "game"
870  function gameMousePressed() {
871
872    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
873      if (isHover(gameBtn)) {
874        if (gameStarted) {
875          end();
876        } else {
877          start();
878        }
879      } else {
880        if (gameStarted) {
881          end();
882        } else {
883          start();
884        }
885      }
886    }
887  }
888
889  // Draw the button
890  drawGameButton(gameBtn);
891
892  // Draw the circle target
893  fill(255, 0, 0);
894  circle(x, y, r * 2);
895
896  // only when currentScreen === "game"
897  function gameMousePressed() {
898
899    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
900      if (isHover(gameBtn)) {
901        if (gameStarted) {
902          end();
903        } else {
904          start();
905        }
906      } else {
907        if (gameStarted) {
908          end();
909        } else {
910          start();
911        }
912      }
913    }
914  }
915
916  // Draw the button
917  drawGameButton(gameBtn);
918
919  // Draw the circle target
920  fill(255, 0, 0);
921  circle(x, y, r * 2);
922
923  // only when currentScreen === "game"
924  function gameMousePressed() {
925
926    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
927      if (isHover(gameBtn)) {
928        if (gameStarted) {
929          end();
930        } else {
931          start();
932        }
933      } else {
934        if (gameStarted) {
935          end();
936        } else {
937          start();
938        }
939      }
940    }
941  }
942
943  // Draw the button
944  drawGameButton(gameBtn);
945
946  // Draw the circle target
947  fill(255, 0, 0);
948  circle(x, y, r * 2);
949
950  // only when currentScreen === "game"
951  function gameMousePressed() {
952
953    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
954      if (isHover(gameBtn)) {
955        if (gameStarted) {
956          end();
957        } else {
958          start();
959        }
960      } else {
961        if (gameStarted) {
962          end();
963        } else {
964          start();
965        }
966      }
967    }
968  }
969
970  // Draw the button
971  drawGameButton(gameBtn);
972
973  // Draw the circle target
974  fill(255, 0, 0);
975  circle(x, y, r * 2);
976
977  // only when currentScreen === "game"
978  function gameMousePressed() {
979
980    if (mouseX > x && mouseX < x + w && mouseY > y && mouseY < y + h) {
981      if (isHover(gameBtn)) {
982        if (gameStarted) {
983          end();
984        } else {
985          start();
986        }
987      } else {
988        if (gameStarted) {
989          end();
990        } else {
991          start();
992        }
993      }
994    }
995  }
996
997  // Draw the button
998  drawGameButton(gameBtn);
999
1000 // Draw the circle target
1001 fill(255, 0, 0);
1002 circle(x, y, r * 2);
1003
1004 // only when currentScreen === "game"
1005 function gameMousePressed() {
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428

```

game.js

```
94 -     if (isHover(gameBtn)) {
95 -         triggerRandomOutcome();
115 +     if (!gameStarted) {
116 +         if (isHover(gameBtn)) {
117 +             gameStarted = true;
118 +             newTarget();
119 +         }
120 +         return;
121     }
122 +
123 +     if (isHover(endBtn)) {
124 +         currentScreen = "lose";
125 +         return;
126 +     }
127 +
128 +     // After start: clicking the circle spawns the next one
129 +     checkhit(mouseX, mouseY);
130 }
131
132 // -----
133 @@ -101,28 +134,31 @@ function gameMousePressed() {
134 // -----
135 // Allows keyboard-only interaction (accessibility + design)
136 function gameKeyPressed() {
137     // ENTER key triggers the same behaviour as clicking the button
138     if (keyCode === ENTER) {
139         triggerRandomOutcome();
140
141         if (!gameStarted && keyCode === ENTER) {
142             gameStarted = true;
143             newTarget();
144             checkhit(mouseX, mouseY);
145         }
146     }
147
148 // -----
```



instructions.js

```
... @@ -11,8 +11,8 @@
11 11     // drawInstr() is called from main.js
12 12     // only when currentScreen === "instr"
13 13     function drawInstr() {
14 -     // Light neutral background
15 -     background(240);
14 +     // dusk blue background
15 +     background(63, 80, 112);
16 16
17 17     // ---- Screen title ----
18 18     fill(0);
19
20 @@ -26,7 +26,7 @@ function drawInstr() {
21 26     // \n creates a line break in the text
22 27     // This is useful for simple multi-line instructions
23 28     const lines =
24 -     "Press the game button.\n" + "You have a chance to win or lose!";
25 29 +     "Press the start button.\n" + "Find as many circles as possible!";
26 30
27 31     text(lines, width / 2, 160);
28 32
```

start.js

```
... @@ -12,13 +12,13 @@
12 12     // currentScreen === "start"
13 13     function drawStart() {
14 14         // Background colour for the start screen
15 15     - background(180, 225, 220); // soft teal background
16 16     + background(229, 107, 111); // light coral
17 17         // ---- Title text ----
18 18         fill(30, 50, 60);
19 19         textSize(46);
20 20         textAlign(CENTER, CENTER);
21 21     - text("Win or Lose", width / 2, 180);
22 22     + text("Lose or Lose", width / 2, 180);
23 23         // ---- Buttons (data only) ----
24 24         // These objects store the position/size/label for each button.
...

```



lose.js

```
... @@ -10,8 +10,8 @@
10 10     // drawLose() is called from main.js
11 11     // only when currentScreen === "lose"
12 12     function drawLose() {
13 13     -     // Red-tinted background to communicate failure
14 14     -     background(255, 210, 210);
15 15     +     // dusty lavender
16 16     +     background(109, 89, 122);
17 17         fill(0);
18 18         textAlign(CENTER, CENTER);
...

```



```
main.js
@@ -51,7 +51,7 @@ function draw() {
51 51     if (currentScreen === "start") drawStart();
52 52     else if (currentScreen === "instr") drawInstr();
53 53     else if (currentScreen === "game") drawGame();
54 54     - else if (currentScreen === "win") drawWin();
54 54     + // else if (currentScreen === "win") drawWin();
55 55     else if (currentScreen === "lose") drawLose();
56 56
57 57     // (Optional teaching note)
58 58
@@ -78,7 +78,7 @@ function mousePressed() {
78 78     else if (currentScreen === "game") gameMousePressed();
79 79     // The ?.() means "call this function only if it exists"
80 80     // This prevents errors if a screen doesn't implement a handler.
81 81     - else if (currentScreen === "win") winMousePressed?.();
81 81     + // else if (currentScreen === "win") winMousePressed?.();
82 82     else if (currentScreen === "lose") loseMousePressed?.();
83 83 }
84 84
@@ -97,7 +97,7 @@ function keyPressed() {
97 97     if (currentScreen === "start") startKeyPressed();
98 98     else if (currentScreen === "instr") instrKeyPressed();
99 99     else if (currentScreen === "game") gameKeyPressed?.();
100 100     - else if (currentScreen === "win") winKeyPressed?.();
100 100     + //else if (currentScreen === "win") winKeyPressed?.();
101 101     else if (currentScreen === "lose") loseKeyPressed?.();
102 102 }
103 103
....
```



## Game Screen

Click the circle to spawn the next one.



**END GAME**

## Appendix

what is wrong with my code-

```
GAME.JS // NOTE: Do NOT add setup() or draw() in this file // setup() and draw() live in main.js // This file only defines: // 1) drawGame() → what the game screen looks like // 2) input handlers → what happens when the player clicks or presses keys // 3) helper functions specific to this screen // ----- // Button data //
----- // This object stores all the information needed to draw // and interact with the button on the game screen. // Keeping this in one object makes it easier to move, // resize, or restyle the button later. const gameBtn = { x: 400, // x position (centre of the button) y: 550, // y position (centre of the button) w: 260, // width h: 90, // height label: "PRESS HERE", // text shown on the button }; //
----- // Main draw function for this screen // -----
drawGame() is called from main.js *only* // when currentScreen === "game" function drawGame() { // Set background colour for the game screen background(181, 101, 118); // ---- Title and instructions text --- fill(0); // black text textSize(32); textAlign(CENTER, CENTER); text("Game Screen", width / 2, 160); textSize(18); text("Click the button to begin. \n", width / 2, 210); // ---- Draw the button ---- // We pass the button object to a helper function drawGameButton(gameBtn); // ---- Cursor feedback ---- // If the mouse is over the button, show a hand cursor // Otherwise, show the normal arrow cursor cursor(isHover(gameBtn) ? HAND : ARROW); } //
----- // Button drawing helper // -----
// This function is responsible *only* for drawing the button. // It does NOT handle clicks or game logic. function drawGameButton({ x, y, w, h, label }) { rectMode(CENTER); // Check if the mouse is hovering over the button // isHover() is defined in main.js so it can be shared const hover = isHover({ x, y, w, h }); noStroke(); // Change button colour when hovered // This gives visual feedback to the player fill( hover ? color(180, 220, 255, 220) // lighter blue on hover : color(200, 220, 255, 190), // normal state ); // Draw the button rectangle rect(x, y, w, h, 14); // last value = rounded corners // Draw the button text fill(0); textSize(28); textAlign(CENTER, CENTER); text(label, x, y); } //
----- // Mouse input for this screen //
----- // This function is called from main.js // only when currentScreen === "game" function gameMousePressed() { // Only trigger the outcome if the button is clicked if (isHover(gameBtn)) { checkhit(mouseX, mouseY); } } let x, y, r = 40; //circle //let score = 0; function newTarget() { x = random(width); // new random x position y = random(height); // new random y position } //
----- // Keyboard input for this screen //
----- // Allows keyboard-only interaction (accessibility + design) function gameKeyPressed() { // ENTER key triggers the same behaviour as clicking the button if (keyCode === ENTER) { checkhit(mouseX, mouseY); } } //
```

```
----- // Game logic: click the circle ----- // This
function decides what happens next in the game. // It does NOT draw anything. function
checkhit(mx, my) { //check if the circle let d = dist(mx, my, x, y); if (d < r) { newTarget(); }
} MAIN.JS // ----- // main.js = the "router"
(traffic controller) for the whole game // -----
// Idea: this project has multiple screens (start, instructions, game, win, lose). // Instead
of putting everything in one giant file, each screen lives in its own // file and defines two
main things: // 1) drawX() → how that screen looks // 2) XMousePressed() /
XKeyPressed() → how that screen handles input // // This main.js file does 3 important
jobs: // A) stores the current screen in a single shared variable // B) calls the correct
draw function each frame // C) sends mouse/keyboard input to the correct screen
handler // ----- // Global game state // ----- // This
variable is shared across all files because all files run in the same // global JavaScript
scope when loaded in index.html. // // We store the "name" of the current screen as a
string. // Only one screen should be active at a time. let currentScreen = "start"; // "start"
| "instr" | "game" | "win" | "lose" // ----- // setup() runs ONCE at the
beginning // ----- // This is where you usually set canvas size and
initial settings. function setup() { createCanvas(800, 800); newTarget(); // Sets a default
font for all text() calls // (This can be changed later per-screen if you want.)
textFont("sans-serif"); } // ----- // draw() runs every frame (many times
per second) // ----- // This is the core "router" for visuals. // Depending
on currentScreen, we call the correct draw function. function draw() { fill(255, 0, 0);
circle(x, y, r * 2); // Each screen file defines its own draw function: // start.js →
drawStart() // instructions.js → drawInstr() // game.js → drawGame() // win.js →
drawWin() // lose.js → drawLose() if (currentScreen === "start") drawStart(); else if
(currentScreen === "instr") drawInstr(); else if (currentScreen === "game") drawGame();
else if (currentScreen === "win") drawWin(); else if (currentScreen === "lose")
drawLose(); // (Optional teaching note) // This "if/else chain" is a very common early
approach. // Later in the course you might replace it with: // - a switch statement, or // -
an object/map of screens } // ----- // mousePressed() runs once each
time the mouse is clicked // ----- // This routes mouse input to the
correct screen handler. function mousePressed() { // Each screen *may* define a mouse
handler: // start.js → startMousePressed() // instructions.js → instrMousePressed() //
game.js → gameMousePressed() // win.js → winMousePressed() // lose.js →
loseMousePressed() if (currentScreen === "start") startMousePressed(); else if
(currentScreen === "instr") instrMousePressed(); else if (currentScreen === "game")
gameMousePressed(); // The ?.() means "call this function only if it exists" // This
prevents errors if a screen doesn't implement a handler. else if (currentScreen ===
```

```

"win") winMousePressed?(); else if (currentScreen === "lose") loseMousePressed?(); }
// ----- // keyPressed() runs once each time a key is pressed //
----- // This routes keyboard input to the correct screen handler.

function keyPressed() { // Each screen *may* define a key handler: // start.js →
startKeyPressed() // instructions.js → instrKeyPressed() // game.js →
gameKeyPressed() // win.js → winKeyPressed() // lose.js → loseKeyPressed() if
(currentScreen === "start") startKeyPressed(); else if (currentScreen === "instr")
instrKeyPressed(); else if (currentScreen === "game") gameKeyPressed?(); else if
(currentScreen === "win") winKeyPressed?(); else if (currentScreen === "lose")
loseKeyPressed?(); } // ----- // Shared helper
function: isHover() // ----- // // Many screens
have buttons. // This helper checks whether the mouse is inside a rectangle. // //
Important: our buttons are drawn using rectMode(CENTER), // meaning x,y is the
CENTRE of the rectangle. // So we check mouseX and mouseY against
half-width/half-height bounds. // // Input: an object with { x, y, w, h } // Output: true if
mouse is over the rectangle, otherwise false function isHover({ x, y, w, h }) { return (
mouseX > x - w / 2 && // mouse is right of left edge mouseX < x + w / 2 && // mouse is
left of right edge mouseY > y - h / 2 && // mouse is below top edge mouseY < y + h / 2 // mouse
is above bottom edge ); }

```

Thought for 7s

Two big things are “wrong” here (and they’re the reason your circle/click logic feels broken):

## 1) You draw the circle in `main.js` before the game screen draws its background

In `draw()` you do:

```
fill(255, 0, 0);
```

```
circle(x, y, r * 2);
```

```
else if (currentScreen === "game") drawGame();
```

But inside `drawGame()` you call:

```
background(181, 101, 118);
```

So the circle gets drawn... then immediately gets covered by the background. That's why it looks like it's not spawning.

### Fix (best)

Move the circle drawing into `drawGame()` **after** the background:

```
function drawGame() {  
    background(181, 101, 118);  
  
    // draw target circle here (after background)  
    fill(255, 0, 0);  
    circle(x, y, r * 2);  
  
    // ...rest of your UI  
    drawGameButton(gameBtn);  
    cursor(isHover(gameBtn) ? HAND : ARROW);  
}
```

And **delete** the circle drawing from `main.js`.

---

## 2) Your click handler only checks the button, not the circle

Right now your `gameMousePressed()` does this:

```
if (isHover(gameBtn)) {  
    checkhit(mouseX, mouseY);  
}
```

So `checkhit()` only ever runs when you click the **button**, not when you click the **circle**.

### Fix (simple)

Let clicks anywhere on the game screen check the circle, but still keep the button for whatever purpose:

```
function gameMousePressed() {  
    // if you want the button to do something specific, do it here  
    if (isHover(gameBtn)) {  
        // example: start/reset, go to instructions, etc.  
        return;  
    }  
  
    // otherwise, clicking anywhere checks if you hit the circle  
    checkhit(mouseX, mouseY);  
}
```

Or if you want *only clicking the circle* to matter, you still need to call `checkhit()` even when not on the button.

---

## Bonus gotcha (common)

Make sure `index.html` loads files in an order where `newTarget()` exists before `setup()` runs.

If your script order is:

```
<script src="main.js"></script>  
<script src="game.js"></script>
```

Then `setup()` runs and calls `newTarget()` before `game.js` defines it → crash / undefined.

### Safe order:

```
<script src="game.js"></script>
```

```
<script src="main.js"></script>
```

(Or define `newTarget()` in `main.js`, since it's shared.)

---

If you tell me your `index.html` script order + what you want the button to do (start game? reset target? go to lose screen?), I can give you the cleanest final version.

Now create a end game button that takes us to the lose game screen

Perfect, this fits really cleanly into your existing screen system 

We'll add **an End Game button on the game screen** that switches `currentScreen` to "lose".

I'll keep this **simple, readable, and consistent** with how your other buttons work.

---

## ① Add an End Game button object (GAME.JS)

Put this **under your existing `gameBtn`:**

```
const endGameBtn = {  
  x: 400,  
  y: 680,  
  w: 260,  
  h: 70,  
  label: "END GAME",  
};
```

---

## ② Draw the End Game button in `drawGame()`

Inside `drawGame()`, **after** your main button and circle are drawn:

```
// draw end game button  
drawGameButton(endGameBtn);
```

Your `drawGame()` should now conceptually look like:

```
function drawGame() {  
    background(181, 101, 118);  
  
    // target circle  
    fill(255, 0, 0);  
    circle(x, y, r * 2);  
  
    // text  
    fill(0);  
    textSize(32);  
    textAlign(CENTER, CENTER);  
    text("Game Screen", width / 2, 160);  
  
    // buttons  
    drawGameButton(gameBtn);  
    drawGameButton(endGameBtn);  
  
    cursor(  
        isHover(gameBtn) || isHover(endGameBtn)  
        ? HAND  
        : ARROW  
    );  
}
```

---

### ③ Handle clicking the End Game button

Update `gameMousePressed()` in `game.js`:

```
function gameMousePressed() {  
    // end game button → go to lose screen  
    if (isHover(endGameBtn)) {  
        currentScreen = "lose";  
        return;  
    }  
  
    // optional: other button behaviour  
    if (isHover(gameBtn)) {  
        return;  
    }  
  
    // clicking anywhere else checks circle hit  
    checkhit(mouseX, mouseY);  
}
```

---

### ④ (Optional but recommended) Keyboard support

Inside `gameKeyPressed()`:

```
function gameKeyPressed() {  
    // ESC key ends the game  
    if (keyCode === ESCAPE) {
```

```
currentScreen = "lose";  
}  
  
if (keyCode === ENTER) {  
    checkhit(mouseX, mouseY);  
}  
}
```