

Overview

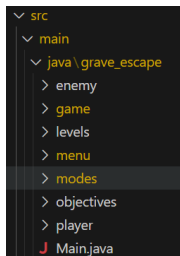
In this report, we have included and documented the smells that we have identified and the solutions for addressing the detected smells as well as the performed refactoring included with images. Our github activity shows the commits that we have made to make the appropriate changes to our code.

Code Smells

1. Badly Structured Project (All files)
2. Data Clumps (Player.java)
3. Large Class (Enemy.java)
4. Large Class (Player.java)
5. Badly Structured Project (Levels inside levels)
6. Badly Structured Project (Door.java & Wall.java)
7. Duplicate Code (GameFactory.java)
8. Duplicate Code (Grid.java, Level.java & GamePanel.java)

Refactoring Techniques to Solve the Code Smells

1. Organized Files
 - Made folders for each important component of our game rather than having all the files inside the java folder. This helps with improved readability and appropriate grouping.
2. Extract Class/Delete data
 - The class had initializations of variables that were not used so we moved/deleted unnecessary variables. This helps with the fact that we now have less lines of unnecessary code in our files for improved readability.



```
public class Player {
    private Position position;
    private int lives;
    private float fieldOfView;
    private Direction facing;
    private int matchPrevMove;
    public Player(Position position) {
```



```
public class Player {
    private Position position;
    private Direction facing;
    private int matchPrevMove;
    public Player(Position position) {
```

3. Extract Class
 - Moving over data and methods that have to do with getting the position of and Enemy Tracking to a MovingObject.java file. Removed method (code) duplication in the Enemy class.

```
import grave_escape.game.Position;
import grave_escape.game.Direction;

public class Enemy {
    private Position position;
    private Direction facing;
    private int matchPrevMove;

    public Enemy(Position position) {
        this.position = position;
    }

    public Position getPosition() {
        return position;
    }

    public void setPosition(Position p) {
        this.position = p;
    }

    public Direction getFacing() {
        return facing;
    }

    public void setFacing(Direction d) {
        this.facing = d;
    }

    public int getLives() {
        return lives;
    }

    public void setLives(int lives) {
        this.lives = lives;
    }

    public float getFieldOfView() {
        return fieldOfView;
    }

    public void setFieldOfView(float fov) {
        this.fieldOfView = fov;
    }

    public int getMatchPrevMove() {
        return matchPrevMove;
    }

    public void setMatchPrevMove(int matchPrevMove) {
        this.matchPrevMove = matchPrevMove;
    }
}
```



```
package grave_escape.enemy;

import grave_escape.game.Position;
import grave_escape.player.MovingObject;

public abstract class Enemy extends MovingObject {
    private Position position;
    private Direction facing;
    private int matchPrevMove;

    public Enemy(Position position) {
        super(position);
    }
}
```

```
import grave_escape.enemy;
import grave_escape.game.Position;
import grave_escape.game.Direction;

public class Enemy {
    private Position position;
    private Direction facing;
    private int matchPrevMove;

    public Enemy(Position position) {
        this.position = position;
    }

    public Position getPosition() {
        return position;
    }

    public void setPosition(Position p) {
        this.position = p;
    }

    public Direction getFacing() {
        return facing;
    }

    public void setFacing(Direction d) {
        this.facing = d;
    }

    public int getLives() {
        return lives;
    }

    public void setLives(int lives) {
        this.lives = lives;
    }

    public float getFieldOfView() {
        return fieldOfView;
    }

    public void setFieldOfView(float fov) {
        this.fieldOfView = fov;
    }

    public int getMatchPrevMove() {
        return matchPrevMove;
    }

    public void setMatchPrevMove(int matchPrevMove) {
        this.matchPrevMove = matchPrevMove;
    }
}
```

4. Extract Class
 - Moving over data and methods that have to do with getting the position of the player to a MovingObject.java file. Removed method (code) duplication in the Player class.
5. Organized Files
 - Added subfolders to the levels folder to specify the levels for each corresponding difficulty/progression. Having each level grouped makes it easier to make changes as well as improved readability.
6. Organized Files
 - Moved over the Door.java and Wall.java to a different "structure" folder since they did not really fit the "criteria" of the objectives folder. The objectives folder now only is left with the HighestResult.java and the Objective.java files. This helps for improved readability as well as more appropriate grouping.

7. Refactor to Factory Pattern

- Extracted logic for creating game levels and difficulties into a new GameFactory class.
- Consolidated Level object creation (e.g., LevelEasy, LevelNormal, etc.) based on difficulty and game level.
- Removed duplicate code across buttons by using GameFactory.createGame() to encapsulate game initialization logic.

```
src/main/java/grave_escape/modes/CampaignPanel.java
import grave_escape.levels.*;
import grave_escape.objectives.HighestResult;

// ...

121 117 difficultyPanel.setLayout(null);
122 118 difficultyPanel.setBorder(BorderFactory.createLineBorder(Color.white, 2));
123 119 difficultyPanel.setBounds((1280 / 2) - (300 / 2), 175, 300, 300);

120 + // Game factory class to create multiple game levels
121 + GameFactory gameFactory = new GameFactory(cardLayout, mainPanel, GameMode.CAMPAIGN);
122 +
123 +
124 124 // Draw difficulty buttons
125 125 easyButton = drawButton("Easy", new Rectangle(50, 50, 200, 50), 20);
126 126 difficultyPanel.add(easyButton);
127 127 easyButton.addActionListener(e -> {
128 - difficulty = Difficulty.EASY;
129 - Level level = new LevelEasy();
130 - Game game = new Game(cardLayout, mainPanel, difficulty, GameMode.CAMPAIGN, level);
131 + Game game = gameFactory.createGame(Difficulty.EASY, GameLevel.Level1);
132 - game.startGame();
133 131
134 132 normalButton = drawButton("Normal", new Rectangle(50, 125, 200, 50), 20);
135 133 difficultyPanel.add(normalButton);
136 134 normalButton.addActionListener(e -> {
137 - difficulty = Difficulty.NORMAL;
138 - Level level = new LevelNormal();
139 - Game game = new Game(cardLayout, mainPanel, difficulty, GameMode.CAMPAIGN, level);
140 + Game game = gameFactory.createGame(Difficulty.NORMAL, GameLevel.Level1);
141 - game.startGame();
142 136
143 137
144 138 hardButton = drawButton("Hard", new Rectangle(50, 200, 200, 50), 20);
145 139 difficultyPanel.add(hardButton);
146 140 hardButton.addActionListener(e -> {
147 - difficulty = Difficulty.HARD;
148 - Level level = new LevelHard();
149 - Game game = new Game(cardLayout, mainPanel, difficulty, GameMode.CAMPAIGN, level);
150 + Game game = gameFactory.createGame(Difficulty.HARD, GameLevel.Level1);
151 - game.startGame();
152 143
153 144
154 145
155 146
156 147
157 148
158 149
159 150
160 151
161 152
162 153
163 154
164 155
165 156
166 157
167 158
168 159
169 160
170 161
171 162
172 163
173 164
174 165
175 166
176 167
177 168
178 169
179 170
180 171
181 172
182 173
183 174
184 175
185 176
186 177
187 178
188 179
189 180
190 181
191 182
192 183
193 184
194 185
195 186
196 187
197 188
198 189
199 190
200 191
201 192
202 193
203 194
204 195
205 196
206 197
207 198
208 199
209 200
210 201
211 202
212 203
213 204
214 205
215 206
216 207
217 208
218 209
219 210
220 211
221 212
222 213
223 214
224 215
225 216
226 217
227 218
228 219
229 220
230 221
231 222
232 223
233 224
234 225
235 226
236 227
237 228
238 229
239 230
240 231
241 232
242 233
243 234
244 235
245 236
246 237
247 238
248 239
249 240
250 241
251 242
252 243
253 244
254 245
255 246
256 247
257 248
258 249
259 250
260 251
261 252
262 253
263 254
264 255
265 256
266 257
267 258
268 259
269 260
270 261
271 262
272 263
273 264
274 265
275 266
276 267
277 268
278 269
279 270
280 271
281 272
282 273
283 274
284 275
285 276
286 277
287 278
288 279
289 280
290 281
291 282
292 283
293 284
294 285
295 286
296 287
297 288
298 289
299 290
300 291
301 292
302 293
303 294
304 295
305 296
306 297
307 298
308 299
309 300
310 301
311 302
312 303
313 304
314 305
315 306
316 307
317 308
318 309
319 310
320 311
321 312
322 313
323 314
324 315
325 316
326 317
327 318
328 319
329 320
330 321
331 322
332 323
333 324
334 325
335 326
336 327
337 328
338 329
339 330
340 331
341 332
342 333
343 334
344 335
345 336
346 337
347 338
348 339
349 340
350 341
351 342
352 343
353 344
354 345
355 346
356 347
357 348
358 349
359 350
360 351
361 352
362 353
363 354
364 355
365 356
366 357
367 358
368 359
369 360
370 361
371 362
372 363
373 364
374 365
375 366
376 367
377 368
378 369
379 370
380 371
381 372
382 373
383 374
384 375
385 376
386 377
387 378
388 379
389 380
390 381
391 382
392 383
393 384
394 385
395 386
396 387
397 388
398 389
399 390
400 391
401 392
402 393
403 394
404 395
405 396
406 397
407 398
408 399
409 400
410 401
411 402
412 403
413 404
414 405
415 406
416 407
417 408
418 409
419 410
420 411
421 412
422 413
423 414
424 415
425 416
426 417
427 418
428 419
429 420
430 421
431 422
432 423
433 424
434 425
435 426
436 427
437 428
438 429
439 430
440 431
441 432
442 433
443 434
444 435
445 436
446 437
447 438
448 439
449 440
450 441
451 442
452 443
453 444
454 445
455 446
456 447
457 448
458 449
459 450
460 451
461 452
462 453
463 454
464 455
465 456
466 457
467 458
468 459
469 460
470 461
471 462
472 463
473 464
474 465
475 466
476 467
477 468
478 469
479 470
480 471
481 472
482 473
483 474
484 475
485 476
486 477
487 478
488 479
489 480
490 481
491 482
492 483
493 484
494 485
495 486
496 487
497 488
498 489
499 490
500 491
501 492
502 493
503 494
504 495
505 496
506 497
507 498
508 499
509 500
510 501
511 502
512 503
513 504
514 505
515 506
516 507
517 508
518 509
519 510
520 511
521 512
522 513
523 514
524 515
525 516
526 517
527 518
528 519
529 520
530 521
531 522
532 523
533 524
534 525
535 526
536 527
537 528
538 529
539 530
540 531
541 532
542 533
543 534
544 535
545 536
546 537
547 538
548 539
549 540
550 541
551 542
552 543
553 544
554 545
555 546
556 547
557 548
558 549
559 550
560 551
561 552
562 553
563 554
564 555
565 556
566 557
567 558
568 559
569 560
570 561
571 562
572 563
573 564
574 565
575 566
576 567
577 568
578 569
579 570
580 571
581 572
582 573
583 574
584 575
585 576
586 577
587 578
588 579
589 580
590 581
591 582
592 583
593 584
594 585
595 586
596 587
597 588
598 589
599 590
600 591
601 592
602 593
603 594
604 595
605 596
606 597
607 598
608 599
609 600
610 601
611 602
612 603
613 604
614 605
615 606
616 607
617 608
618 609
619 610
620 611
621 612
622 613
623 614
624 615
625 616
626 617
627 618
628 619
629 620
630 621
631 622
632 623
633 624
634 625
635 626
636 627
637 628
638 629
639 630
640 631
641 632
642 633
643 634
644 635
645 636
646 637
647 638
648 639
649 640
650 641
651 642
652 643
653 644
654 645
655 646
656 647
657 648
658 649
659 650
660 651
661 652
662 653
663 654
664 655
665 656
666 657
667 658
668 659
669 660
670 661
671 662
672 663
673 664
674 665
675 666
676 667
677 668
678 669
679 670
680 671
681 672
682 673
683 674
684 675
685 676
686 677
687 678
688 679
689 680
690 681
691 682
692 683
693 684
694 685
695 686
696 687
697 688
698 689
699 690
700 691
701 692
702 693
703 694
704 695
705 696
706 697
707 698
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
1000 991
1001 992
1002 993
1003 994
1004 995
1005 996
1006 997
1007 998
1008 999
1009 1000
1010 1001
1011 1002
1012 1003
1013 1004
1014 1005
1015 1006
1016 1007
1017 1008
1018 1009
1019 1010
1020 1011
1021 1012
1022 1013
1023 1014
1024 1015
1025 1016
1026 1017
1027 1018
1028 1019
1029 1020
1030 1021
1031 1022
1032 1023
1033 1024
1034 1025
1035 1026
1036 1027
1037 1028
1038 1029
1039 1030
1040 1031
1041 1032
1042 1033
1043 1034
1044 1035
1045 1036
1046 1037
1047 1038
1048 1039
1049 1040
1050 1041
1051 1042
1052 1043
1053 1044
1054 1045
1055 1046
1056 1047
1057 1048
1058 1049
1059 1050
1060 1051
1061 1052
1062 1053
1063 1054
1064 1055
1065 1056
1066 1057
1067 1058
1068 1059
1069 1060
1070 1061
1071 1062
1072 1063
1073 1064
1074 1065
1075 1066
1076 1067
1077 1068
1078 1069
1079 1070
1080 1071
1081 1072
1082 1073
1083 1074
1084 1075
1085 1076
1086 1077
1087 1078
1088 1079
1089 1080
1090 1081
1091 1082
1092 1083
1093 1084
1094 1085
1095 1086
1096 1087
1097 1088
1098 1089
1099 1090
1100 1091
1101 1092
1102 1093
1103 1094
1104 1095
1105 1096
1106 1097
1107 1098
1108 1099
1109 1100
1110 1101
1111 1102
1112 1103
1113 1104
1114 1105
1115 1106
1116 1107
1117 1108
1118 1109
1119 1110
1120 1111
1121 1112
1122 1113
1123 1114
1124 1115
1125 1116
1126 1117
1127 1118
1128 1119
1129 1120
1130 1121
1131 1122
1132 1123
1133 1124
1134 1125
1135 1126
1136 1127
1137 1128
1138 1129
1139 1130
1140 1131
1141 1132
1142 1133
1143 1134
1144 1135
1145 1136
1146 1137
1147 1138
1148 1139
1149 1140
1150 1141
1151 1142
1152 1143
1153 1144
1154 1145
1155 1146
1156 1147
1157 1148
1158 1149
1159 1150
1160 1151
1161 1152
1162 1153
1163 1154
1164 1155
1165 1156
1166 1157
1167 1158
1168 1159
1169 1160
1170 1161
1171 1162
1172 1163
1173 1164
1174 1165
1175 1166
1176 1167
1177 1168
1178 1169
1179 1170
1180 1171
1181 1172
1182 1173
1183 1174
1184 1175
1185 1176
1186 1177
1187 1178
1188 1179
1189 1180
1190 1181
1191 1182
1192 1183
1193 1184
1194 1185
1195 1186
1196 1187
1197 1188
1198 1189
1199 1190
1200 1191
1201 1192
1202 1193
1203 1194
1204 1195
1205 1196
1206 1197
1207 1198
1208 1199
1209 1200
1210 1201
1211 1202
1212 1203
1213 1204
1214 1205
1215 1206
1216 1207
1217 1208
1218 1209
1219 1210
1220 1211
1221 1212
1222 1213
1223 1214
1224 1215
1225 1216
1226 1217
1227 1218
1228 1219
1229 1220
1230 1221
1231 1222
1232 1223
1233 1224
1234 1225
1235 1226
1236 1227
1237 1228
1238 1229
1239 1230
1240 1231
1241 1232
1242 1233
1243 1234
1244 1235
1245 1236
1246 1237
1247 1238
1248 1239
1249 1240
1250 1241
1251 1242
1252 1243
1253 1244
1254 1245
1255 1246
1256 1247
1257 1248
1258 1249
1259 1250
1260 1251
1261 1252
1262 1253
1263 1254
1264 1255
1265 1256
1266 1257
1267 1258
1268 1259
1269 1260
1270 1261
1271 1262
1272 1263
1273 1264
1274 1265
1275 1266
1276 1267
1277 1268
1278 1269
1279 1270
1280 1271
1281 1272
1282 1273
1283 1274
1284 1275
1285 1276
1286 1277
1287 1278
1288 1279
1289 1280
1290 1281
1291 1282
1292 1283
1293 1284
1294 1285
1295 1286
1296 1287
1297 1288
1298 1289
1299 1290
1300 1291
1301 1292
1302 1293
1303 1294
1304 1295
1305 1296
1306 1297
1307 1298
1308 1299
1309 1300
1310 1301
1311 1302
1312 1303
1313 1304
1314 1305
1315 1306
1316 1307
1317 1308
1318 1309
1319 1310
1320 1311
1321 1312
1322 1313
1323 1314
1324 1315
1325 1316
1326 1317
1327 1318
1328 1319
1329 1320
1330 1321
1331 1322
1332 1323
1333 1324
1334 1325
1335 1326
1336 1327
1337 1328
1338 1329
1339 1330
1340 1331
1341 1332
1342 1333
1343 1334
1344 1335
1345 1336
1346 1337
1347 1338
1348 1339
1349 1340
1350 1341
1351 1342
1352 1343
1353 1344
1354 1345
1355 1346
1356 1347
1357 1348
1358 1349
1359 1350
1360 1351
1361 1352
1362 1353
1363 1354
1364 1355
1365 1356
1366 1357
1367 1358
1368 1359
1369 1360
1370 1361
1371 1362
1372 1363
1373 1364
1374 1365
1375 1366
1376 1367
1377 1368
1378 1369
1379 1370
1380 1371
1381 1372
1382 1373
1383 1374
1384 1375
1385 1376
1386 1377
1387 1378
1388 1379
1389 1380
1390 1381
1391 1382
1392 1383
1393 1384
1394 1385
1395 1386
1396 1387
1397 1388
1398 1389
1399 1390
1400 1391
1401 1392
1402 1393
1403 1394
1404 1395
1405 1396
1406 1397
1407 1398
1408 1399
1409 1400
1410 1401
1411 1402
1412 1403
1413 1404
1414 1405
1415 1406
1416 1407
1417 1408
1418 1409
1419 1410
1420 1411
1421 1412
1422 1413
1423 1414
1424 1415
1425 1416
1426 1417
1427 1418
1428 1419
1429 1420
1430 1421
1431 1422
1432 1423
1433 1424
1434 1425
1435 1426
1436 1427
1437 1428
1438 1429
1439 1430
1440 1431
1441 1432
1442 1433
1443 1434
1444 1435
1445 1436
1446 1437
1447 1438
1448 1439
1449 1440
1450 1441
1451 1442
1452 1443
1453 1444
1454 1445
1455 1446
1456 1447
1457 1448
1458 1449
1459 1450
1460 1451
1461 1452
1462 1453
1463 1454
1464 1455
1465 1456
1466 1457
1467 1458
1468 1459
1469 1460
1470 1461
1471 1462
1472 1463
1473 1464
1474 1465
1475 1466
1476 1467
1477 1468

```