



**PRT582 SOFTWARE ENGINEERING: PROCESS
AND TOOLS**
Assignment 2

Submitted by:

Student Name:	Student ID:
Shahil Jha	S368427

CONTENTS

Introduction.....	3
Programming Language and Unit Testing Tool	3
Process.....	3
1. Random Number Generation Function.....	4
2. Get Hint Function	7
3. Number comparision Function	10
4. Quit or Replay Option Function	14
5. Integretion Testing.....	17
Outcome of the Program.....	18
Conclusion	19
References	20

INTRODUCTION

This report provides a detailed explanation of the development of the number guessing program using Test Driven Development. The main intention of this program is to generate a score based on the number of times it takes the user to guess the randomly generated four-digit number. The program will then give some clues about the number. Once the player guesses the number correctly, the program will show the number of attempts taken.

The requirements of the program are:

- Randomly generate a four-digit number.
- The program will keep asking the user to guess the number until the player guesses it correctly or has quit.
- When the number is entered, the program will respond with hints using 'circle' and 'x' to show how accurate the guess was.
 - A 'circle' indicates that one digit is correct and is in the right spot.
 - A 'x' indicates that one digit is correct but in the wrong spot.
- Once the game is finished,
 - The number of attempts taken will be displayed.
 - The player will be asked to quit or to play again.
- Players can quit the game anytime.

PROGRAMMING LANGUAGE AND UNIT TESTING TOOL

Python is a potent programming language that's simple to comprehend and easy on the eyes. It showcases many traits shared by numerous other languages and proves its practicality in real-life scenarios. Moreover, it is freely available, comes with a standardized version, and enjoys the support of a sizable and affable community of passionate developers.

PyUnit (unittest): The foundation of the unittest unit testing framework drew its initial inspiration from JUnit and carried a comparable essence to prominent testing frameworks in various programming languages. It enables automated testing, sharing setup and teardown code for tests, grouping tests into collections, and ensuring that tests remain unaffected by the reporting system used (Python, 2023). The fact that this is a built-in module of python makes it more dependable.

PROCESS

Test-driven development (TDD) represents an agile approach to software development. Like other agile methodologies, TDD constructs software in incremental steps, commencing with creating an automated test, succeeded by a small segment of code tailored to fulfill that test's requirements. Subsequently, the code is refined to enhance its performance (Anwer et al., 2017). TDD starkly contrasts conventional software development practices that follow a sequence of design, coding, and testing. Instead, TDD inverts this sequence, initiating testing before coding even begins. The primary aim of TDD is to curtail the occurrence of defects and elevate the overall quality of code.

The phases of the number guessing program are as follows:

1. Test Cases and Initial Failing Test
2. Development to Pass the Test
3. Passing the Test

Unlike in other development approaches, in TDD the tests for any unit function are written first before the development of the business logic itself. Similarly, the test cases for the functionality of the number guessing game were written first and was expected to fail first.

1. RANDOM NUMBER GENERATION FUNCTION

The random number generation function is responsible for generating the random four-digit number that the player must guess.

```
import unittest
from unittest.mock import patch
from guess_number_game import guess_number_game

class TestGuessNumberGame(unittest.TestCase):
    def setUp(self):
        self.game = guess_number_game(1234)

    def test_generate_random_number_randomness(self):
        random_number1 = self.game.generate_random_number()
        random_number2 = self.game.generate_random_number()
        random_number3 = self.game.generate_random_number()
        random_number4 = self.game.generate_random_number()
        self.assertNotEqual(random_number1, random_number2)
        self.assertNotEqual(random_number1, random_number3)
        self.assertNotEqual(random_number1, random_number4)

    def test_generate_random_number_data_type(self):
        random_number = self.game.generate_random_number()
        result = isinstance(random_number, int)
        self.assertTrue(result)
        result = isinstance(random_number, float)
        self.assertFalse(result)
        result = isinstance(random_number, str)
        self.assertFalse(result)
```

FIGURE 1: TEST CASES FOR RANDOM NUMBER GENERATION

```

test > test_generate_random_number.py > ...
1 import unittest
2 from unittest.mock import patch
3 from guess_number_game import guess_number_game
4
5 class TestGuessNumberGame(unittest.TestCase):
6     def setUp(self):
7         self.game = guess_number_game(1234)
8
9     def test_generate_random_number(self, parameter):
10         random_number1 = self.game.generate_random_number()
11         random_number2 = self.game.generate_random_number()
12         random_number3 = self.game.generate_random_number()
13         random_number4 = self.game.generate_random_number()
14         self.assertNotEqual(random_number1, random_number2)
15         self.assertNotEqual(random_number1, random_number3)
16         self.assertNotEqual(random_number1, random_number4)
17
18     def test_generate_random_number_data_type(self):
19         random_number = self.game.generate_random_number()
20         result = isinstance(random_number, int)
21         self.assertTrue(result)
22         result = isinstance(random_number, float)
23         self.assertFalse(result)
24         result = isinstance(random_number, str)
25         self.assertFalse(result)
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

PS D:\Masters\Semester 2\PRITS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\Code> python -m unittest
FF
-----
FAIL: test_generate_random_number_data_type (test.test_generate_random_number.TestGuessNumberGame)
Traceback (most recent call last):
  File "D:\Masters\Semester 2\PRITS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\Code\test\test_generate_random_number.py", line 21, in test_generate_random_number_data_type
    self.assertTrue(result)
AssertionError: False is not true
=====
FAIL: test_generate_random_number_randomness (test.test_generate_random_number.TestGuessNumberGame)
Traceback (most recent call last):
  File "D:\Masters\Semester 2\PRITS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\Code\test\test_generate_random_number.py", line 14, in test_generate_random_number_randomness
    self.assertNotEqual(random_number1, random_number2)
AssertionError: None == None
=====
Ran 2 tests in 0.009s
FAILED (failures=2)
PS D:\Masters\Semester 2\PRITS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\Code>

```

FIGURE 2: FAILED TEST FOR RANDOM NUMBER GENERATION

```

# Generate a random four-digit number
def generate_random_number(self):
    digits = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
    # Shuffle the digits
    random.shuffle(digits)
    # Take the first 4 digits
    generated_value = int(''.join(map(str, digits[:4])))
    return generated_value

```

FIGURE 3: FUNCTION DEVELOPMENT OF RANDOM NUMBER GENERATION

The screenshot shows a code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project structure with files like `guess_number_game.py`, `test_generate_random_number.py`, `main.py`, `README.md`, and `rules.md`. The code editor displays the content of `test_generate_random_number.py`, which includes imports for `unittest` and `unittest.mock`, and a class `TestGuessNumberGame` with two test methods: `test_generate_random_number_randomness` and `test_generate_random_number_data_type`. The terminal shows the output of running the tests, indicating that all tests passed successfully.

```
test > test_generate_random_number.py
import unittest
from unittest.mock import patch
from guess_number_game import guess_number_game

class TestGuessNumberGame(unittest.TestCase):
    def setUp(self):
        self.game = guess_number_game(1234)

    def test_generate_random_number_randomness(self):
        random_number1 = self.game.generate_random_number()
        random_number2 = self.game.generate_random_number()
        random_number3 = self.game.generate_random_number()
        random_number4 = self.game.generate_random_number()
        self.assertEqual(random_number1, random_number2)
        self.assertNotEqual(random_number1, random_number3)
        self.assertNotEqual(random_number1, random_number4)

    def test_generate_random_number_data_type(self):
        random_number = self.game.generate_random_number()
        result = isinstance(random_number, int)
        self.assertTrue(result)
        result = isinstance(random_number, float)
        self.assertFalse(result)
        result = isinstance(random_number, str)
        self.assertFalse(result)

TERMINAL    PROBLEMS    OUTPUT    DEBUG CONSOLE

AssertionError: False is not true

FAIL: test_generate_random_number_randomness (test_generate_random_number.TestGuessNumberGame)
-----
Traceback (most recent call last):
  File "D:\Masters\Semester 2\PRISB2 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\Code\test\test_generate_random_number.py", line 14, in test_generate_random_number_randomness
    self.assertEqual(random_number1, random_number2)
AssertionError: None == None

Ran 2 tests in 0.010s

FAILED (failures=2)
PS D:\Masters\Semester 2\PRISB2 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\Code> python -m unittest -v
test_generate_random_number_data_type (test_generate_random_number.TestGuessNumberGame) ... ok
test_generate_random_number_randomness (test_generate_random_number.TestGuessNumberGame) ... ok

Ran 2 tests in 0.009s

OK
PS D:\Masters\Semester 2\PRISB2 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\Code>
```

FIGURE 4: PASSED TEST FOR RANDOM NUMBER GENERATION

2. GET HINT FUNCTION

This function is responsible for hint generation for the input number using circles and crosses.

```
import unittest
from unittest.mock import patch
from guess_number_game import guess_number_game

class TestGuessNumberGame(unittest.TestCase):
    def setUp(self):
        self.game = guess_number_game(1234)

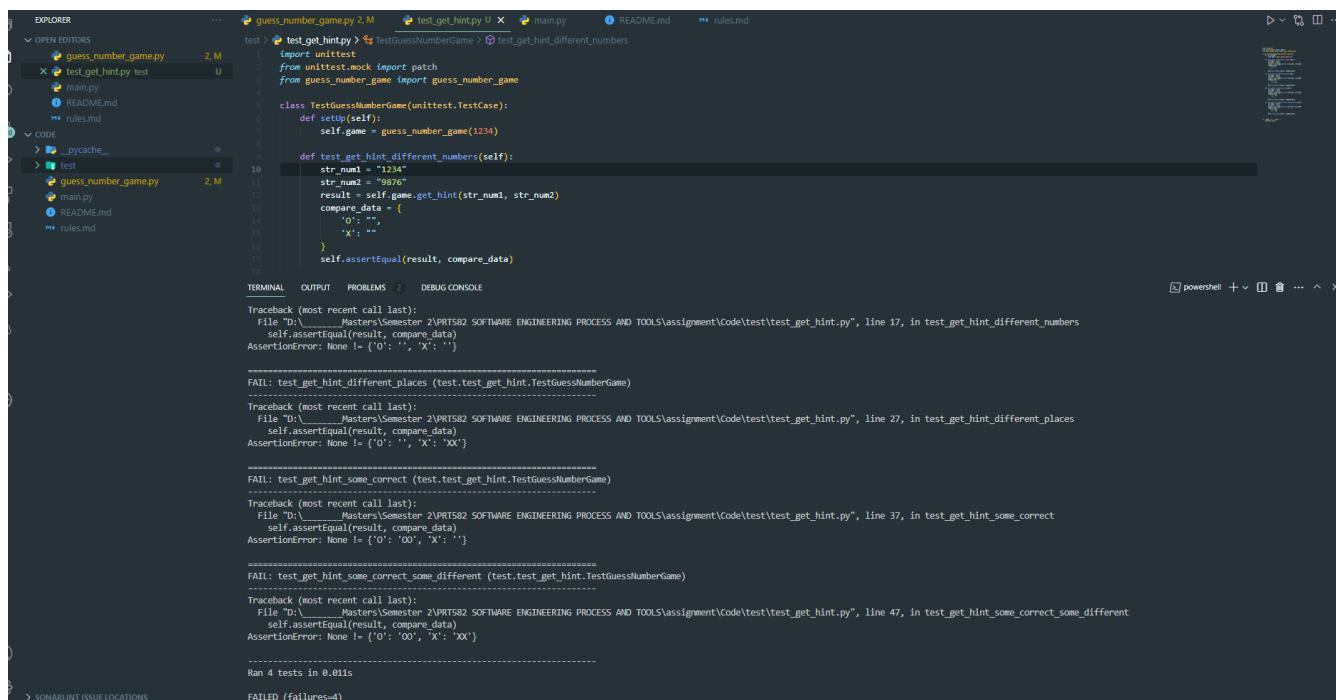
    def test_get_hint_different_numbers(self):
        str_num1 = "1234"
        str_num2 = "9876"
        result = self.game.get_hint(str_num1, str_num2)
        compare_data = {
            'O': "",
            'X': ""
        }
        self.assertEqual(result, compare_data)

    def test_get_hint_different_places(self):
        str_num1 = "1234"
        str_num2 = "4871"
        result = self.game.get_hint(str_num1, str_num2)
        compare_data = {
            'O': "",
            'X': "XX"
        }
        self.assertEqual(result, compare_data)

    def test_get_hint_some_correct(self):
        str_num1 = "1234"
        str_num2 = "1874"
        result = self.game.get_hint(str_num1, str_num2)
        compare_data = {
            'O': "00",
            'X': ""
        }
        self.assertEqual(result, compare_data)

    def test_get_hint_some_correct_some_different(self):
        str_num1 = "1234"
        str_num2 = "1324"
        result = self.game.get_hint(str_num1, str_num2)
        compare_data = {
            'O': "00",
            'X': "XX"
        }
        self.assertEqual(result, compare_data)
```

FIGURE 5: TEST CASES FOR GET HINT FUNCTION



```

test > test_get_hint.py > TestGuessNumberGame > test_get_hint_different_numbers
1 import unittest
2 from unittest.mock import patch
3 from guess_number_game import guess_number_game
4
5 class TestGuessNumberGame(unittest.TestCase):
6     def setUp(self):
7         self.game = guess_number_game(1234)
8
9     def test_get_hint_different_numbers(self):
10         str_num1 = "1234"
11         str_num2 = "9876"
12         result = self.game.get_hint(str_num1, str_num2)
13         compare_data = {
14             '0': '',
15             'X': ''
16         }
17         self.assertEqual(result, compare_data)
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Traceback (most recent call last):
 File "D:\...Masters\Semester 2\PRIS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\code\test\test_get_hint.py", line 17, in test_get_hint_different_numbers
 self.assertEqual(result, compare_data)
 AssertionError: None != {'0': '', 'X': ''}

FAIL: test_get_hint_different_numbers (test.test_get_hint.TestGuessNumberGame)

Traceback (most recent call last):
 File "D:\...Masters\Semester 2\PRIS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\code\test\test_get_hint.py", line 27, in test_get_hint_different_places
 self.assertEqual(result, compare_data)
 AssertionError: None != {'0': '', 'X': 'XX'}

FAIL: test_get_hint_different_places (test.test_get_hint.TestGuessNumberGame)

Traceback (most recent call last):
 File "D:\...Masters\Semester 2\PRIS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\code\test\test_get_hint.py", line 37, in test_get_hint_some_correct
 self.assertEqual(result, compare_data)
 AssertionError: None != {'0': '00', 'X': ''}

FAIL: test_get_hint_some_correct (test.test_get_hint.TestGuessNumberGame)

Traceback (most recent call last):
 File "D:\...Masters\Semester 2\PRIS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\code\test\test_get_hint.py", line 47, in test_get_hint_some_correct_some_different
 self.assertEqual(result, compare_data)
 AssertionError: None != {'0': '00', 'X': 'XX'}

FAIL: test_get_hint_some_correct_some_different (test.test_get_hint.TestGuessNumberGame)

Ran 4 tests in 0.011s
 FAILED (failures=4)

FIGURE 6: FAILED TESTS FOR GET HINT FUNCTION



```

# method to compare numbers and generate hint
def get_hint(self, str_num1, str_num2):
    crosses = ""
    circles = ""
    # iterate the string casted numver generate hints
    for i in range(4):
        if str_num1[i] == str_num2[i]:
            circles += "0"
        elif str_num1[i] in str_num2:
            crosses += "X"
    data = {
        '0': circles,
        'X': crosses
    }
    return data

```

FIGURE 7: FUNCTION DEVELOPMENT OF GET HINT FUNCTION


```

test > test_get_hint.py > testGuessNumberGame > test_get_hint_different_numbers
1 import unittest
2 from unittest.mock import patch
3 from guess_number_game import guess_number_game
4
5 class TestGuessNumberGame(unittest.TestCase):
6     def setUp(self):
7         self.game = guess_number_game(1234)
8
9     def test_get_hint_different_numbers(self):
10         str_num1 = "1234"
11         str_num2 = "8078"
12         result = self.game.get_hint(str_num1, str_num2)
13         compare_data = {
14             '0': '',
15             'X': ''
16         }
17         self.assertEqual(result, compare_data)
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
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
1001
1002
1003
1004
1005
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
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2
```

3. NUMBER COMPARISON FUNCTION

This function is responsible for comparing the randomly generated four-digit number and the input number by the user.

```
import unittest
from unittest.mock import patch
from guess_number_game import guess_number_game

class TestGuessNumberGame(unittest.TestCase):
    def setUp(self):
        self.game = guess_number_game(1234)

    def test_compare_guess_correct(self):
        self.assertTrue(self.game.compare_guess(1234))

    def test_compare_guess_incorrect(self):
        self.assertFalse(self.game.compare_guess(5678))

    def test_compare_guess_input_exception(self):
        with self.assertRaises(ValueError) as exception_context:
            self.game.compare_guess(guess_number="1234")
        self.assertEqual(
            str(exception_context.exception),
            "Both inputs must be integers"
        )

    def test_compare_guess_generated_number_exception(self):
        self.game.generated_number = "1234"
        with self.assertRaises(ValueError) as exception_context:
            self.game.compare_guess(guess_number=1234)
        self.assertEqual(
            str(exception_context.exception),
            "Both inputs must be integers"
        )

    def test_compare_guess_input_value_range_exception(self):
        with self.assertRaises(ValueError) as exception_context:
            self.game.compare_guess(guess_number=12345)
        self.assertEqual(
            str(exception_context.exception),
            "Both numbers must be 4-digit numbers"
        )

    def test_compare_guess_generated_number_value_range_exception(self):
        self.game.generated_number = 12345
        with self.assertRaises(ValueError) as exception_context:
            self.game.compare_guess(guess_number=1234)
        self.assertEqual(
            str(exception_context.exception),
            "Both numbers must be 4-digit numbers"
        )
```

FIGURE 9: TEST CASES FOR NUMBER COMPARISON FUNCTION

```

EXPLORER
  guess_number_game.py 2 M
  test_compare_guess.py test
  main.py
  README.md
  rules.md

CODE
  _pycache_
  test
  guess_number_game.py 2 M
  main.py
  README.md
  rules.md

test > test_compare_guess.py > testGuessNumberGame > test_compare_guess_input_exception
1 import unittest
2 from unittest.mock import patch
3 from guess_number_game import guess_number_game
4
5 class TestGuessNumberGame(unittest.TestCase):
6     def setUp(self):
7         self.game = guess_number_game(1234)
8
9     def test_compare_guess_correct(self):
10        self.assertTrue(self.game.compare_guess(1234))
11
12    def test_compare_guess_incorrect(self):
13        self.assertFalse(self.game.compare_guess(5678))
14
15    def test_compare_guess_input_exception(self):
16        with self.assertRaises(ValueError) as exception_context:
17            self.game.compare_guess(guess_number="1234")
18        self.assertEqual(
19
TERMINAL
  PROBLEMS OUTPUT DEBUG CONSOLE
  powershell + - - - - -
  FAIL: test_compare_guess_generated_number_exception (test.test_compare_guess.TestGuessNumberGame)
  -----
  Traceback (most recent call last):
    File "D:\MastersSemester 2\PARTS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignmentCode\test\test_compare_guess.py", line 26, in test_compare_guess_generated_number_exception
      self.game.compare_guess(guess_number=1234)
  AssertionError: ValueError not raised
  =====
  FAIL: test_compare_guess_generated_number_value_range_exception (test.test_compare_guess.TestGuessNumberGame)
  -----
  Traceback (most recent call last):
    File "D:\MastersSemester 2\PARTS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignmentCode\test\test_compare_guess.py", line 43, in test_compare_guess_generated_number_value_range_exception
      self.game.compare_guess(guess_number=1234)
  AssertionError: ValueError not raised
  =====
  FAIL: test_compare_guess_input_exception (test.test_compare_guess.TestGuessNumberGame)
  -----
  Traceback (most recent call last):
    File "D:\MastersSemester 2\PARTS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignmentCode\test\test_compare_guess.py", line 17, in test_compare_guess_input_exception
      self.game.compare_guess(guess_number="1234")
  AssertionError: ValueError not raised
  =====
  FAIL: test_compare_guess_input_value_range_exception (test.test_compare_guess.TestGuessNumberGame)
  -----
  Traceback (most recent call last):
    File "D:\MastersSemester 2\PARTS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignmentCode\test\test_compare_guess.py", line 34, in test_compare_guess_input_value_range_exception
      self.game.compare_guess(guess_number=12345)
  AssertionError: ValueError not raised
  =====
  Ran 6 tests in 0.012s
  > SONARLint ISSUE LOCATIONS

```

FIGURE 10: FAILED TEST NUMBER COMPARISION FUNCTION

```

# Compare the player's guess with the generated number
def compare_guess(self, guess_number):
    # Check input validity
    if not (
        isinstance(self.generated_number, int) and isinstance(guess_number, int)
    ):
        raise ValueError("Both inputs must be integers")

    # Convert numbers to strings for comparison
    temp_str_number1 = str(self.generated_number) if self.generated_number != 0 else
"0000"
    temp_str_number2 = str(guess_number) if guess_number != 0 else "0000"

    # resolve length issue if the front of any number is 0
    str_number1 = temp_str_number1 if len(temp_str_number1) == 4 else ('0' +
temp_str_number1)
    str_number2 = temp_str_number2 if len(temp_str_number2) == 4 else ('0' +
temp_str_number2)

    # check the numbers are in range and are of 4 digits
    if not (len(str_number1) == 4 and len(str_number2) == 4):
        raise ValueError("Both numbers must be 4-digit numbers")

    # Increment the number of attempts
    self.attempts += 1
    if str_number1 == str_number2:
        return True

    # get hint
    hint = self.get_hint(str_number1, str_number2)

    # add the hint to table
    self.table.add_row(temp_str_number2, hint['X'], hint['O'])
    return False

```

FIGURE 11: FUNCTION DEVELOPMENT OF NUMBER COMPARISON FUNCTION

The screenshot shows a VS Code editor with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The file explorer shows a project structure with files like `guess_number_game.py`, `test_compare_guess.py`, `main.py`, `README.md`, and `rules.md`. The code editor displays the content of `test_compare_guess.py`, which includes imports for `unittest` and `unittest.mock`, and a class `TestGuessNumberGame` with several test methods. The terminal shows the output of running the tests, including failure messages for `test_compare_guess_input_exception` and `test_compare_guess_input_value_range_exception`, and a final summary indicating that 6 tests passed.

```

test_compare_guess.py 3 TestGuessNumberGame
import unittest
from unittest.mock import patch
from guess_number_game import guess_number_game

class TestGuessNumberGame(unittest.TestCase):
    def setUp(self):
        self.game = guess_number_game(1234)

    def test_compare_guess_correct(self):
        self.assertTrue(self.game.compare_guess(1234))

    def test_compare_guess_incorrect(self):
        self.assertFalse(self.game.compare_guess(5678))

    def test_compare_guess_input_exception(self):
        with self.assertRaises(ValueError) as exception_context:
            self.game.compare_guess(guess_number="1234")
        self.assertEqual(exception_context.exception.args[0], "Invalid input")

    def test_compare_guess_input_value_range_exception(self):
        with self.assertRaises(ValueError) as exception_context:
            self.game.compare_guess(guess_number=12345)
        self.assertEqual(exception_context.exception.args[0], "Value out of range")

if __name__ == '__main__':
    unittest.main()

```

```

=====
FAIL: test_compare_guess_input_exception (test.test_compare_guess.TestGuessNumberGame)
Traceback (most recent call last):
  File "D:\Masters\Semester 2\PRTS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\code\test\test_compare_guess.py", line 17, in test_compare_guess_input_exception
    self.game.compare_guess(guess_number="1234")
AssertionError: ValueError not raised
=====
FAIL: test_compare_guess_input_value_range_exception (test.test_compare_guess.TestGuessNumberGame)
Traceback (most recent call last):
  File "D:\Masters\Semester 2\PRTS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\code\test\test_compare_guess.py", line 34, in test_compare_guess_input_value_range_exception
    self.game.compare_guess(guess_number=12345)
AssertionError: ValueError not raised
=====
Ran 6 tests in 0.012s

FAILED (failures=5)
PS D:\Masters\Semester 2\PRTS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\code> python -m unittest -v
test_compare_guess_correct (test.test_compare_guess.TestGuessNumberGame) ... ok
test_compare_guess_generated_number_exception (test.test_compare_guess.TestGuessNumberGame) ... ok
test_compare_guess_generated_number_value_range_exception (test.test_compare_guess.TestGuessNumberGame) ... ok
test_compare_guess_incorrect (test.test_compare_guess.TestGuessNumberGame) ... ok
test_compare_guess_input_exception (test.test_compare_guess.TestGuessNumberGame) ... ok
test_compare_guess_input_value_range_exception (test.test_compare_guess.TestGuessNumberGame) ... ok
Ran 6 tests in 0.011s

OK
PS D:\Masters\Semester 2\PRTS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\code>

```

FIGURE 12: PASSED TEST NUMBER COMPARISON FUNCTION

4. QUIT OR REPLAY OPTION FUNCTION

```
import unittest
from unittest.mock import patch
from guess_number_game import guess_number_game

class TestGuessNumberGame(unittest.TestCase):
    def setUp(self):
        self.game = guess_number_game(1234)

    def test_check_game_finish_input_option_quit(self):
        self.assertFalse(self.game.check_game_finish_input_option("q"))

    def test_check_game_finish_input_option_replay(self):
        self.assertTrue(self.game.check_game_finish_input_option("r"))

    def test_check_game_finish_input_option_quit_case_insensitive(self):
        self.assertFalse(self.game.check_game_finish_input_option("Q"))

    def test_check_game_finish_input_option_replay_case_insensitive(self):
        self.assertTrue(self.game.check_game_finish_input_option("R"))

    def test_check_game_finish_input_option_invalid_then_quit(self):
        with self.assertRaises(ValueError) as exception_context:
            self.game.check_game_finish_input_option(input='a')
        self.assertEqual(
            str(exception_context.exception),
            "Value should be either Q or R (in lowercase or uppercase).")
    )
```

FIGURE 13: TEST CASES FOR QUIT OR REPLAY OPTION FUNCTION

```

test > test_check_game_finish.py > ...
1 from guess_number_game import guess_number_game
2
3 class TestGuessNumberGame(unittest.TestCase):
4     ...def setUp(self):
5         self.game = guess_number_game(1234)
6     ...
7     ...def test_check_game_finish_input_option_quit(self):
8         self.assertFalse(self.game.check_game_finish_input_option("q"))
9     ...
10
11     ...def test_check_game_finish_input_option_replay(self):
12         self.assertTrue(self.game.check_game_finish_input_option("r"))
13     ...
14
15     ...def test_check_game_finish_input_option_quit_case_insensitive(self):
16         self.assertFalse(self.game.check_game_finish_input_option("Q"))
17     ...
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
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
1001
1002
1003
1004
1005
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
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2
```

```

EXPLORER
  OPEN EDITORS
    guess_number_game.py
    test_check_game_finish.py test
    main.py
    README.md
    rules.md
  CODE
    _pycache_
    test
      guess_number_game.py
      main.py
      README.md
      rules.md
  TERMINAL
    powerhell
    =====
    FAIL: test_check_game_finish_input_option_replay (test.test_check_game_finish.TestGuessNumberGame)
    -----
    Traceback (most recent call last):
      File "D:\Masters\Semester 2\PRITS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\Code\test\test_check_game_finish.py", line 14, in test_check_game_finish_input_option_replay
        self.assertTrue(self.game.check_game_finish_input_option("r"))
    AssertionError: None is not true

    =====
    FAIL: test_check_game_finish_input_option_replay_case_insensitive (test.test_check_game_finish.TestGuessNumberGame)
    -----
    Traceback (most recent call last):
      File "D:\Masters\Semester 2\PRITS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\Code\test\test_check_game_finish.py", line 22, in test_check_game_finish_input_option_replay_case_insensitive
        self.assertTrue(self.game.check_game_finish_input_option("R"))
    AssertionError: None is not true

    =====
    Ran 5 tests in 0.011s

    FAILED (failures=2)
    PS D:\Masters\Semester 2\PRITS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\Code> python -m unittest -v
    test_check_game_finish_input_option_invalid_then_quit (test.test_check_game_finish.TestGuessNumberGame) ... ok
    test_check_game_finish_input_option_quit (test.test_check_game_finish.TestGuessNumberGame) ... ok
    test_check_game_finish_input_option_quit_case_insensitive (test.test_check_game_finish.TestGuessNumberGame) ... ok
    test_check_game_finish_input_option_replay (test.test_check_game_finish.TestGuessNumberGame) ... ok
    test_check_game_finish_input_option_replay_case_insensitive (test.test_check_game_finish.TestGuessNumberGame) ... ok

    =====
    Ran 5 tests in 0.011s

    OK
    PS D:\Masters\Semester 2\PRITS82 SOFTWARE ENGINEERING PROCESS AND TOOLS\assignment\Code>
  
```

FIGURE 16: PASSED TEST FOR QUIT OR REPLAY OPTION FUNCTION

5. INTEGRATION TESTING

There are a total of 4 functions for unit testing and a total of 17 test cases.

Performing all tests together.

```

EXPLORER
  OPEN EDITORS
    guess_number_game.py X test_get_hint.py test_generate_random_number.py test_compare_guess.py test_check_game_finish.py main.py README.md rules.md
  CODE
    guess_number_game.py
      25 # Print game rules from a markdown file
      26 def print_rules(self):
      27     # Read the rules.md file and assign to variable
      28     with open("rules.md") as readme:
      29         markdown = Markdown(readme.read())
      30         self.console.print(markdown)
      31
      32 # Generate a random four-digit number
      33 def generate_random_number(self):
      34     digits = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
      35     # Shuffle the digits
      36     random.shuffle(digits)
      37     # Join the digits & digits
      38     generated_value = int(''.join(map(str, digits[:4])))
      39     return generated_value
      40
      41 # Method to compare numbers and generate hint
      42 def get_hint(self, str num1, str num2):
      43
  test.py
    1
    2
    3
    4
    5
    6
    7
    8
    9
    10
    11
    12
    13
    14
    15
    16
    17
    18
    19
    20
    21
    22
    23
    24
    25
    26
    27
    28
    29
    30
    31
    32
    33
    34
    35
    36
    37
    38
    39
    40
    41
    42
    43
    44
    45
    46
    47
    48
    49
    50
    51
    52
    53
    54
    55
    56
    57
    58
    59
    60
    61
    62
    63
    64
    65
    66
    67
    68
    69
    70
    71
    72
    73
    74
    75
    76
    77
    78
    79
    80
    81
    82
    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
    1001
    1002
    1003
    1004
    1005
    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
    1429
    1430
    1431
    1432
    1433
    1434
    1435
    1436
    1437
    1438
    1439
    1440
    1441
    1442
    1443
    1444
    1445
    1446
    1447
    1448
    1449
    1450
    1451
    1452
    1453
    1454
    1455
    1456
    1457
    1458
    1459
    1460
    1461
    1462
    1463
    1464
    1465
    1466
    1467
    1468
    1469
    1470
    1471
    1472
    1473
    1474
    1475
    1476
    1477
    1478
    1479
    1480
    1481
    1482
    1483
    1484
    1485
    1486
    1487
    1488
    1489
    1490
    1491
    1492
    1493
    1494
    1495
    1496
    1497
    1498
    1499
    1500
    1501
    1502
    1503
    1504
    1505
    1506
    1507
    1508
    1509
    1510
    1511
    1512
    1513
    1514
    1515
    1516
    1517
    1518
    1519
    1520
    1521
    1522
    1523
    1524
    1525
    1526
    1527
    1528
    1529
    1530
    1531
    1532
    1533
    1534
    1535
    1536
    1537
    1538
    1539
    1540
    1541
    1542
    1543
    1544
    1545
    1546
    1547
    1548
    1549
    1550
    1551
    1552
    1553
    1554
    1555
    1556
    1557
    1558
    1559
    1560
    1561
    1562
    1563
    1564
    1565
    1566
    1567
    1568
    1569
    1570
    1571
    1572
    1573
    1574
    1575
    1576
    1577
    1578
    1579
    1580
    1581
    1582
    1583
    1584
    1585
    1586
    1587
    1588
    1589
    1590
    1591
    1592
    1593
    1594
    1595
    1596
    1597
    1598
    1599
    1600
    1601
    1602
    1603
    1604
    1605
    1606
    1607
    1608
    1609
    1610
    1611
    1612
    1613
    1614
    1615
    1616
    1617
    1618
    1619
    1620
    1621
    1622
    1623
    1624
    1625
    1626
    1627
    1628
    1629
    1630
    1631
    1632
    1633
    1634
    1635
    1636
    1637
    1638
    1639
    1640
    1641
    1642
    1643
    1644
    1645
    1646
    1647
    1648
    1649
    1650
    1651
    1652
    1653
    1654
    1655
    1656
    1657
    1658
    1659
    1660
    1661
    1662
    1663
    1664
    1665
    1666
    1667
    1668
    1669
    1670
    1671
    1672
    1673
    1674
    1675
    1676
    1677
    1678
    1679
    1680
    1681
    1682
    1683
    1684
    1685
    1686
    1687
    1688
    1689
    1690
    1691
    1692
    1693
    1694
    1695
    1696
    1697
    1698
    1699
    1700
    1701
    1702
    1703
    1704
    1705
    1706
    1707
    1708
    1709
    1710
    1711
    1712
    1713
    1714
    1715
    1716
    1717
    1718
    1719
    1720
    1721
    1722
    1723
    1724
    1725
    1726
    1727
    1728
    1729
    1730
    1731
    1732
    1733
    1734
    1735
    1736
    1737
    1738
    1739
    1740
    1741
    1742
    1743
    1744
    1745
    1746
    1747
    1748
    1749
    1750
    1751
    1752
    1753
    1754
    1755
    1756
    1757
    1758
    1759
    1760
    1761
    1762
    1763
    1764
    1765
    1766
    1767
    1768
    1769
    1770
    1771
    1772
    1773
    1774
    1775
    1776
    1777
    1778
    1779
    1780
    1781
    1782
    1783
    1784
    1785
    1786
    1787
    1788
    1789
    1790
    1791
    1792
    1793
    1794
    1795
    1796
    1797
    1798
    1799
    1800
    1801
    1802
    1803
    1804
    1805
    1806
    1807
    1808
    1809
    1810
    1811
    1812
    1813
    1814
    1815
    1816
    1817
    1818
    1819
    1820
    1821
    1822
    1823
    1824
    1825
    1826
    1827
    1828
    1829
    1830
    1831
    1832
    1833
    1834
    1835
    1836
    1837
    1838
    1839
    1840
    1841
    1842
   
```

OUTCOME OF THE PROGRAM

```

D:\Masters\Semester 1\

RULES

1 A random number will be generated at the start of the game.
2 The main theme of the game is to guess the randomly generated number.
3 When the number is entered, the program will respond with hints using "O" and "X" to show how accurate
the guess was

• A 'O' indicates that one digit is correct and is in the right spot.
• A 'X' indicates that one digit is correct but in the wrong spot.
• The number of such symbol represents the number of digits corresponding to the symbol.

You can Enter 'q' or 'Q' any time to quit the game.
Please Enter to Start the Game.

Input your guessed number : 1234
HISTORY



| Guesses | Crosses | Circles |
|---------|---------|---------|
| 1234    | XX      |         |



Input your guessed number : 1134
HISTORY



| Guesses | Crosses | Circles |
|---------|---------|---------|
| 1234    | XX      |         |
| 1134    | X       |         |



Input your guessed number : 2134
HISTORY



| Guesses | Crosses | Circles |
|---------|---------|---------|
| 1234    | XX      |         |
| 1134    | X       |         |
| 2134    | XX      |         |



Input your guessed number : 1324
HISTORY
  
```

FIGURE 18: EXECUTABLE PROGRAM OF THE GAME

CONCLUSION

This report details how Test-Driven Development can be used to develop a Number Guessing Game. There were many requirements for the applications, such as random four-digit number generation, the input of user-guessed numbers until a correct number is guessed or the program is quit, using circles and crosses to show hints and the ability for the player to quit the game at any time. Overall, developing the program was relatively easy, but it was not easy as well. Python was used to create this program because it makes it easier to understand and maintain code, is fast, and, most importantly, comes pre-installed with the testing tools. In the beginning, it took time to write test cases and develop the application. Later, the familiarity of the tool increased, making it easier to implement. Most of the testing and debugging was done during the development phase, making the program less prone to bugs and issues. Therefore, this is the first time I have developed a program by writing tests first and then coding, and this assignment taught me a new way to do programming.

Link to the GitHub repository: <https://github.com/ShahilJha/s368427-PRT582-Assignment-2.2>

Executable Program File: *guess_number_game.exe* in the same file directory.

REFERENCES

Anwer, F., Aftab, S., Waheed, U., & Muhammad, S. S. (2017). Agile software development models tdd, fdd, dsdm, and crystal methods: A survey. *International journal of multidisciplinary sciences and engineering*, 8(2), 1-10.

Python, W. (2021). Python. Python Releases for Windows, 24.

Python. (2023, August 24). unittest — Unit testing framework. In Python Documentation.

<https://docs.python.org/3/library/unittest.html>