

Helena Trent & Olivia Atheus
CYSE 250 Networking and Programming

Dec 7, 2022

Abstract: This report covers a group project between two individuals using python and techniques learned in class.

Background / Introduction

Python was designed by a dutch man named Guido Rossum in 1991. Python started as a hobby for Rossum as he wanted to develop a language that was more simple than the ones he used at the time. His main goals were increasing the readability and decreasing the amount of code needed to accomplish projects (*History of Python*, 2022). The first version of python consisted of functions, classes, modules, and core data types. Since the first release, newer versions of python have been released to add more functions making it more accessible. Over the years python has grown into one of the most popular programming languages of our time. Programmers who are proficient in python are very valuable in the professional world (Ostrowska, 2022).

In CYSE 250 we focused on essential parts of the python programming to kickstart our understanding of the language. Throughout the semester we covered servers, dictionaries, functions, loops, and more. Using this knowledge we have been tasked with formulating a project to show what we have learned throughout the semester.

Problem Statement

A client wants to create a game to guess words. The client would like the game to be multiplayer and have a select set of words. The other aspects of the game are up to the programmers discretion.

Materials

- Hardware
 - Dell Inspiron
 - MacBook Air
- Software
 - Pycharm version python 3.8
 - Wing 101
 - Zoom

The programming portion of the project was completed using pycharm version python 3.8 and wing 101. Pycharm was used on a dell inspiration and wing was used on a MacBook Air. We used Zoom video conferencing to communicate with each other about the project. As we edited the code we sent the files back and forth through email. A shared google doc was used to complete the report. We also utilized canvas to access supplementary materials from class.

Procedure

First we had to agree on a project idea. This project is not our original idea but we decided that it would be more obtainable for our skill set. Then we discussed the requirements for the project and how we could implement them in our code. This includes lists, dictionaries, loops, and functions. From there we started working on the

code and getting our game to run smoothly. After we were satisfied with the outcome we started on the report, documenting our project.

To write this code we used the techniques used in class and referred back to the powerpoints, textbook chapters, and lectures for guidance. We focused on the requirements for the project to fulfill them with our game.

Techniques Implemented

For our project we implemented an array of functions, loops and dictionaries in the code to execute a successful python program.

Below is a highlight of defined functions presented in the code:

choose() : the program chooses a word

mix() : Shuffles the characters of the chosen word.

final() : Showing the final scores of both players.

winner() : Declares who wins the game

start() : starts the game.

The list is represented by the words used in the game. The loops keep the game going until the user inserts the needed characters to end it.

Further Development/Uses

If further developed this code can be used to create a more complex game that could be used for fun or educational purposes. In our game we used terms from the networking portion of class. If someone else were to use this game they could input any words they wanted. There isn't a limit on how many words can be used either. This game could be improved by adding more players. With more time we could also add more features to the game to lengthen it. Further development could also mean adding a tie breaker. That being said, this game is a solid starting point.

Roles

For our group dynamic we decided to split the projects into two parts. One of us was in charge of the report and the other was in charge of the code. While we did do our part on both elements of the project we took a leadership role in one task. Over the course of the project we checked each other's work to keep ourselves on track. We also decided to split the presentation.

Conclusion and Project Overview

Overall this project was a learning experience. Python has so many amazing uses and we have only scratched the surface. Through this project we have been exposed to the difficulty of programming. While neither of us intend to be programmers. This experience has provided us with a further understanding of programming and respect for those who complete complex projects.

Appendix

These pictures are screenshots of our code with the line numbers labeled. The pictures are on a split screen.

```

1  # For python jumble terms game.
2
3  # import random
4  import random
5
6  # function for randomized terms.
7  def choose():
8      # lists of randomized terms
9      terms = ['Application', 'Presentation', 'Transport', 'Network',
10              'Data Link', 'Physical', 'Session', 'TCP',
11              'OSI']
12
13      pick = random.choice(terms)
14
15      return pick
16
17
18  # Function for shuffling the characters of the chosen word.
19  def mix(word):
20      # shuffles the characters of the word
21      random_word = random.sample(word, len(word))
22
23      # join() method join the elements of the list with character
24      mixed = ''.join(random_word)
25
26      return mixed
27
28
29  # Function for the final score.
30  def final(p1n, p2n, p1, p2):
31      print(p1n, 'Your score is:', p1)
32      print(p2n, 'Your score is:', p2)
33
34  # winner() function calling
35  winner(p1n, p2n, p1, p2)
36
37  print("Thanks for playing, see you next time!")
38
39
40  # Function for the winner
41  def winner(player1, player2, p1score, p2score):
42      if p1score > p2score:
43          print("The winner is:", player1)
44      elif p2score > p1score:
45          print("The winner is:", player2)
46      else:
47          print("Draw! Well done players")
48
49
50  # Function for starting the game.
51  def start():
52      # enter player1 and player2 name
53      p1name = input("Player 1, Please enter your name:")
54      p2name = input("Player 2, Please enter your name: ")
55
56      # calculating score
57      p1 = 0
58      p2 = 0
59
60      # counting turns
61      turn = 0
62
63      # loop
64      while True:
65

```

Stack Data

Search:

Replace:

☐ Case sensitive ☐ Whole words ☐ In Selection

Options

Debug (IO | Python Shell #1 | Python Shell #2 | Python Shell #3)

Waiting for keyboard input

```

Your Score is: 1
press 1 to continue and 0 to quit:1
The jumbled word is: eetPansotirn
lena It is now your Turn.
What is the word? Transportation
Better luck next time!
trant It is now your turn.
What is the word? Transportation
Better luck next time. The correct word is: Presentation
press 1 to continue and 0 to quit:

```

Line 148 Col 3

The screenshot displays a code editor with two Python files open. The left file, `complexnumbers.py`, defines a game logic with functions for choosing words, mixing them, and checking turns. The right file, `pythonscramblegame.py`, serves as the main driver, handling user input and game flow. The bottom panel shows the output of the program, indicating the user's score and the correct word.

```

# complexnumbers.py
60 # counting turns
61 turn = 0
62
63 # loop
64 while True:
65     # choose() function calling
66     picked_word = choose()
67
68     # jumble() function calling
69     qn = mix(picked_word)
70     print("The jumbled word is:", qn)
71
72     # checking turn is odd or even
73     if turn % 2 == 0:
74
75         # if turn no. is even
76         # player1 turn
77         print(pname, 'It is now your Turn.')
78
79         ans = input("What is the word? ")
80
81     # checking ans is equal to picked_word or not
82     if ans == picked_word:
83
84         # incremented by 1
85         pp1 += 1
86
87         print('Your score is:', pp1)
88         turn += 1
89
90     else:
91         print("Better luck next time!")
92
93 # pythonscramblegame.py
128 print("Better luck next time!")
129 print(pname, 'It is now your turn.')
130 ans = input("What is the word? ")
131
132 if ans == picked_word:
133     pp1 += 1
134     print("Your Score is:", pp1)
135
136 else:
137     print("Better luck next time. The correct word is:", picked_word)
138
139 c = int(input("press 1 to continue and 0 to quit:"))
140
141 if c == 0:
142     # final() function calling
143     final(pname, p2name, pp1, pp2)
144     break
145
146 c = int(input("press 1 to continue and 0 to quit:"))
147 if c == 0:
148     # final() function calling
149     final(p1name, p2name, pp1, pp2)
150     break
151
152 # Driver code
153 if __name__ == '__main__':
154     # start() function calling
155     start()
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

```

References

History of Python. (2022, November 26). GeeksforGeeks. Retrieved December 6, 2022, from <https://www.geeksforgeeks.org/history-of-python/>

Ostrowska, K. (2022, June 20). *A Brief History of Python*. LearnPython.com. Retrieved December 6, 2022, from <https://learnpython.com/blog/history-of-python/>