



Lab 2 AI

Connect 4

Presented to

Dr Marwan el torki

Ta eng.mohamed el habibi

OMAR ELSAMMAK

Overview:

Connect 4 is a two-player game in which the players first choose a color and then take turns dropping their colored discs from the top into a grid. The pieces fall straight down, occupying the next available space within the column. The objective of the game is to connect-four of one's own discs of the same color next to each other vertically, horizontally, or diagonally. The two players keep playing until the board is full. The winner is the player having greater number of connected-fours.

Functions used:

1. Draw_gui
This function draws the array to form a gui to make communication easier
2. Array_2_int
This function takes the array that represents the board and stores it in a Long int to store the game state (it represents each column state in 9 bits)
3. Int_2_array
This function takes the stored long int and transfers it into a 6*7 array to represent the state of the board
4. Get_playable_row
This function takes the column we want to play in and the state of the board as a long int and gets which row is the last free one so that we can drop the piece in it
5. Get_playable_columns
This function takes the board state as a long int and checks which columns are available
6. Drop_checker
This functions take the row and column we want to play in and which turn is it and it places the checker piece in the desired place
7. Print_arr
Takes the state as and array and prints it just to illustrate and make sure that the gui is
Working (just a check not important)
8. Calculate_score
Calculates the weight of the given node so that we can use it in the minimax algorithm
9. Get_fours
This functions traverses through the array to get how many combinations of fours are there for a certain player (either player 1 or 2) to be used

when calculating the score of a certain node , also used to check which player is the winner

10. Get_threes

This functions traverses through the array to get how many combinations of threes are there for a certain player (either player 1 or 2) to be used when calculating the score of a certain node

11. Get_twos

This functions traverses through the array to get how many combinations of twos are there for a certain player (either player 1 or 2) to be used when calculating the score of a certain node

12. Minimax

the main algorithm used for making the computer play against a user it is a recursive algorithm ,it also uses recursion to search through the game-tree.

two players play the game, one is called MAX and other is called MIN Both Players of the game are opponent of each other, where MAX will select the maximized value and MIN will select the minimized value

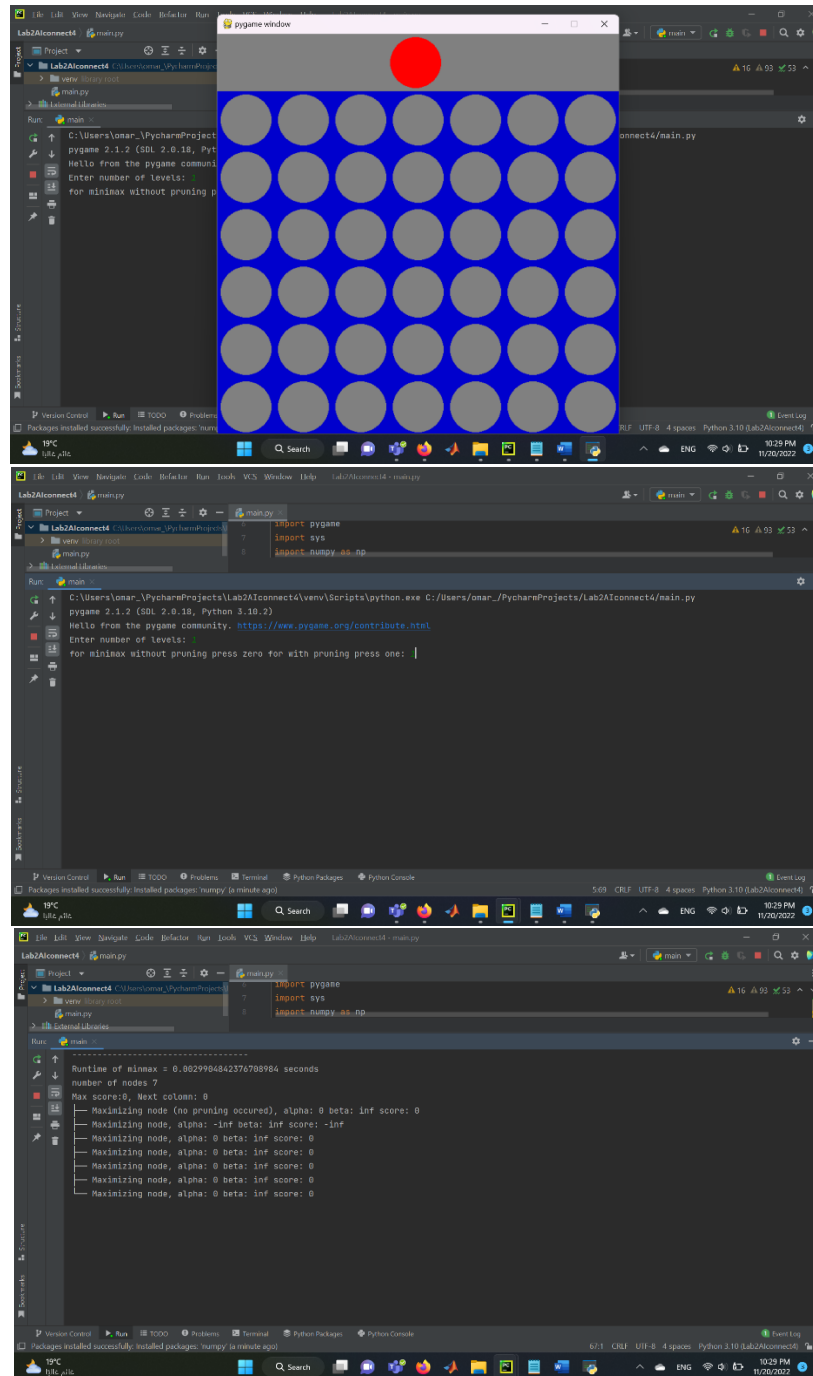
This algorithm applies DFS, so in this game-tree, we have to go all the way through the leaves to reach the terminal nodes

This function takes the depth of the tree from the user as an input K

And whether we want to use it using alpha-beta pruning or not

13. If_game_ened

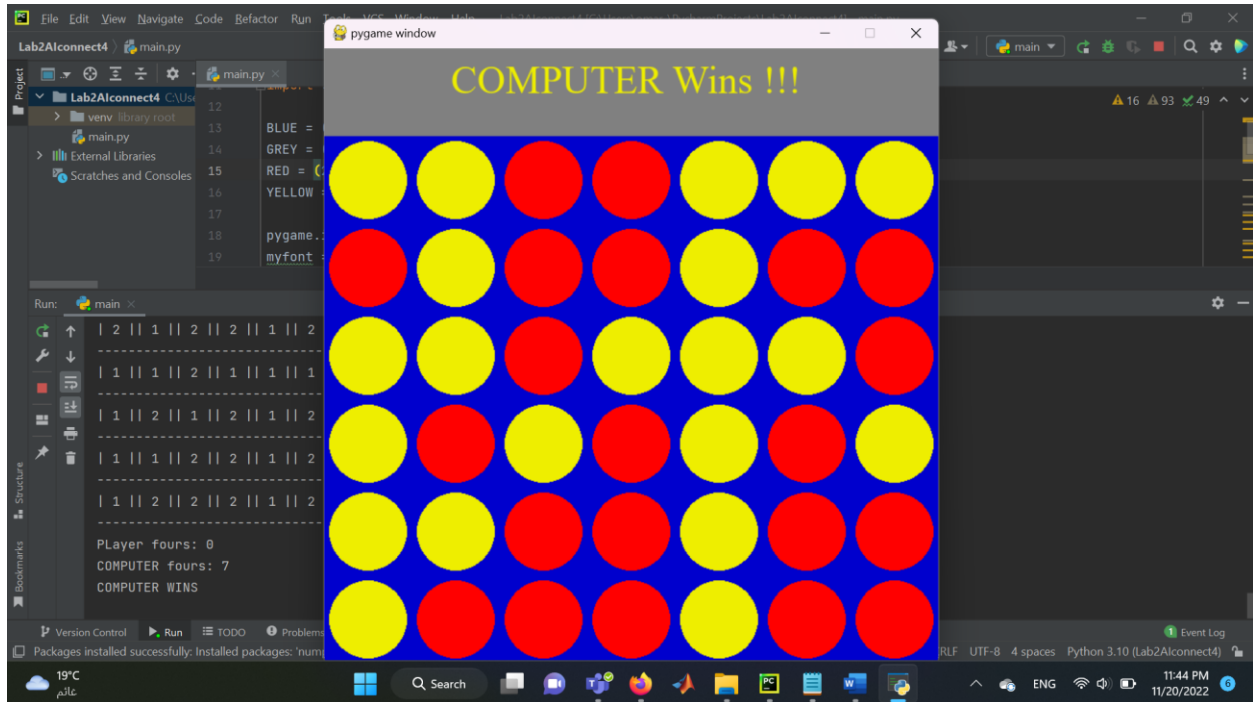
This function checks if all the columns are full to see if the game ended or not



This is a sample run

The computer asks for the depth and whether we want to play using alpha-beta pruning or without, Then we can start playing using the gui

After every move the number of nodes expanded are printed, and the search tree is also printed



Here is a sample run using depth one and with alpha beta pruning

Now I will demonstrate more by showing multiple test cases for multiple depths with and without alpha beta pruning

Without alpha beta pruning :

Depth 1

```
Lab2AIconnect4 - main.py
Project: Lab2AIconnect4
main.py
import pygame
import sys
import numpy as np

Run: main
-----
Runtime of minmax = 0.00240206718446242 seconds
number of nodes 7
Max score:0, Next column: 0
-----
Maximizing node, score: 0
Maximizing node, score: -inf
Maximizing node, score: 0
Maximizing node, score: 0
Maximizing node, score: 0
Maximizing node, score: 0
Maximizing node, score: 0
-----
Packages installed successfully: installed packages: 'numpy' (9 minutes ago)
250.1 CRLF UTF-8 4 spaces Python 3.10 (lab2AIconnect4)
10:36 PM 11/20/2022
```

```
Lab2AIconnect4 - main.py
Project: Lab2AIconnect4
main.py
import pygame
import sys
import numpy as np

Run: main
-----
Runtime of minmax = 0.0026395320892333984 seconds
number of nodes 7
Max score:3, Next column: 0
-----
Maximizing node, score: 3
Maximizing node, score: -inf
Maximizing node, score: 3
Maximizing node, score: 3
Maximizing node, score: 3
Maximizing node, score: 3
Maximizing node, score: 3
-----
Packages installed successfully: installed packages: 'numpy' (9 minutes ago)
188.1 CRLF UTF-8 4 spaces Python 3.10 (lab2AIconnect4)
10:36 PM 11/20/2022
```

```
Lab2AIconnect4 - main.py
Project: Lab2AIconnect4
main.py
import pygame
import sys
import numpy as np

Run: main
-----
Runtime of minmax = 0.0029959678649902344 seconds
number of nodes 7
Max score:0, Next column: 0
-----
Maximizing node, score: 0
Maximizing node, score: -inf
Maximizing node, score: 0
Maximizing node, score: 0
Maximizing node, score: 0
Maximizing node, score: 0
Maximizing node, score: 0
-----
Packages installed successfully: installed packages: 'numpy' (8 minutes ago)
128.1 CRLF UTF-8 4 spaces Python 3.10 (lab2AIconnect4)
10:36 PM 11/20/2022
```

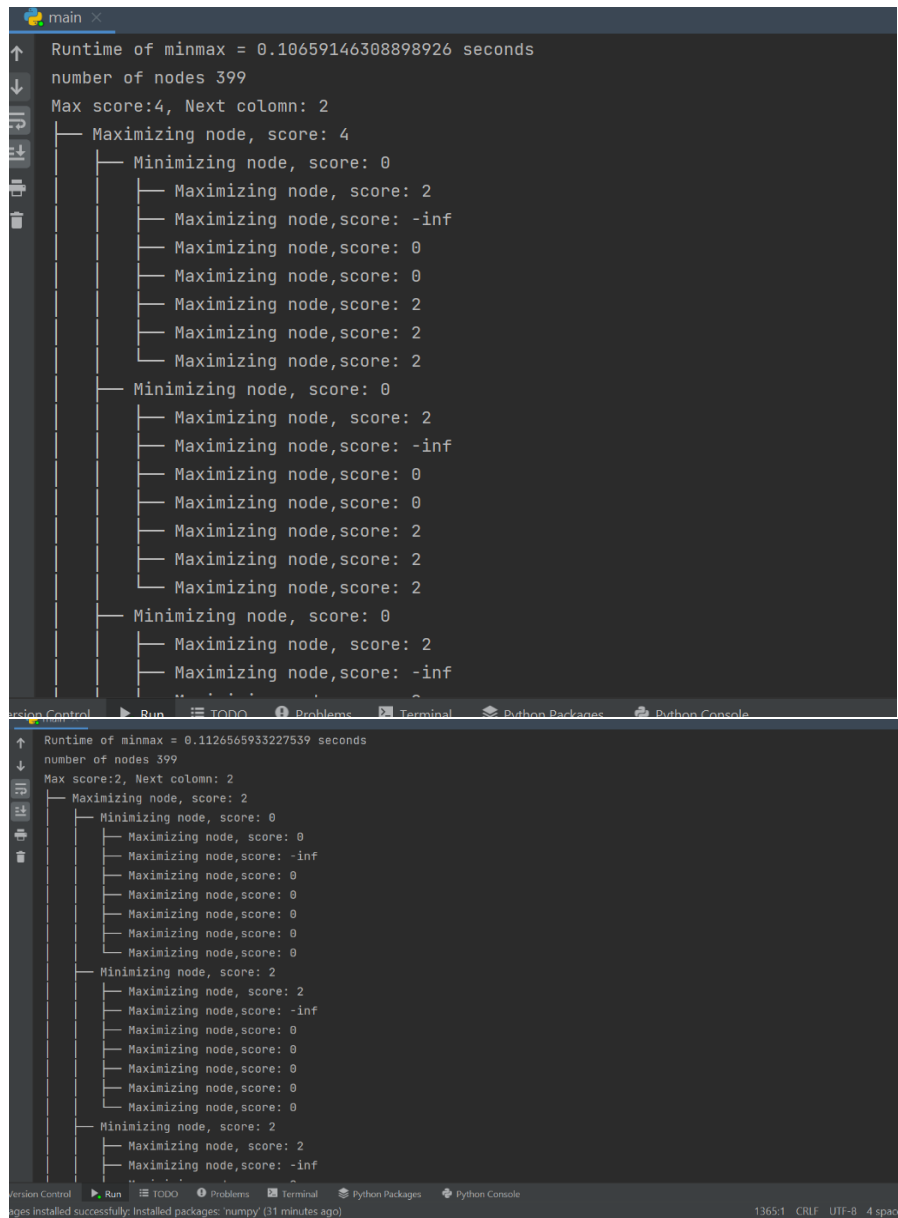
```
File Edit View Navigate Code Refactor Run Tools VCS Window Help Lab2AIconnect4 - main.py
Project main.py
Lab2AIconnect4 C:\Users\omer\PycharmProjects\Lab2AIconnect4
venv library root
main.py
External Libraries
Run: main
| | | | | | | |
| 2 | | | | | |
Runtime of minmax = 0.0019931793212890625 seconds
number of nodes 7
Max score:0, Next column: 0
  Maximizing node, score: 0
  Maximizing node, score: -inf
  Maximizing node, score: 0
  Maximizing node, score: 0
  Maximizing node, score: 0
  Maximizing node, score: 0
  Maximizing node, score: 0
Version Control Run TOODO Problems Terminal Python Packages Python Console
Packages installed successfully. Installed packages: numpy (8 minutes ago)
33:15 CRJF UTF-8 4 spaces Python 3.10 (Lab2AIconnect4)
19°C 10:36 PM 11/20/2022
```

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help Lab2AIconnect4 - main.py
Project main.py
Lab2AIconnect4 C:\Users\omer\PycharmProjects\Lab2AIconnect4
venv library root
main.py
External Libraries
Run: main
Runtime of minmax = 0.001982450685229492 seconds
number of nodes 7
Max score:14, Next column: 1
  Maximizing node, score: 14
  Maximizing node, score: -inf
  Maximizing node, score: 13
  Maximizing node, score: 14
  Maximizing node, score: 14
  Maximizing node, score: 14
  Maximizing node, score: 14
Version Control Run TOODO Problems Terminal Python Packages Python Console
Packages installed successfully. Installed packages: numpy (9 minutes ago)
33:15 CRJF UTF-8 4 spaces Python 3.10 (Lab2AIconnect4)
19°C 10:36 PM 11/20/2022
```

Depth 2:

[illegible]

Depth 3:



```
main x
↑
↓
Runtime of minmax = 0.10659146308898926 seconds
number of nodes 399
Max score:4, Next column: 2
├── Maximizing node, score: 4
│   ├── Minimizing node, score: 0
│   │   ├── Maximizing node, score: 2
│   │   ├── Maximizing node, score: -inf
│   │   ├── Maximizing node, score: 0
│   │   ├── Maximizing node, score: 0
│   │   ├── Maximizing node, score: 2
│   │   ├── Maximizing node, score: 2
│   │   └── Maximizing node, score: 2
│   ├── Minimizing node, score: 0
│   │   ├── Maximizing node, score: 2
│   │   ├── Maximizing node, score: -inf
│   │   ├── Maximizing node, score: 0
│   │   ├── Maximizing node, score: 0
│   │   ├── Maximizing node, score: 2
│   │   ├── Maximizing node, score: 2
│   │   └── Maximizing node, score: 2
│   └── Minimizing node, score: 0
│       ├── Maximizing node, score: 2
│       └── Maximizing node, score: -inf
└── ...

Runtime of minmax = 0.1126565933227539 seconds
number of nodes 399
Max score:2, Next column: 2
├── Maximizing node, score: 2
│   ├── Minimizing node, score: 0
│   │   ├── Maximizing node, score: 0
│   │   ├── Maximizing node, score: -inf
│   │   ├── Maximizing node, score: 0
│   │   ├── Maximizing node, score: 0
│   │   ├── Maximizing node, score: 0
│   │   ├── Maximizing node, score: 0
│   │   └── Maximizing node, score: 0
│   ├── Minimizing node, score: 2
│   │   ├── Maximizing node, score: 2
│   │   ├── Maximizing node, score: -inf
│   │   ├── Maximizing node, score: 0
│   │   ├── Maximizing node, score: 0
│   │   ├── Maximizing node, score: 0
│   │   ├── Maximizing node, score: 0
│   │   └── Maximizing node, score: 0
│   └── Minimizing node, score: 2
│       ├── Maximizing node, score: 2
│       └── Maximizing node, score: -inf
└── ...
```

Version Control Run TODO Problems Terminal Python Packages Python Console

ages installed successfully: installed packages: 'numpy' (31 minutes ago)

1365:1 CRLF UTF-8 4 space

Depth 5:

```
main.py x
Lab2AIconnect4 C 421
> venv library root 422
main.py 423
External Libraries 424
Scratches and Cons 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
1
```

As shown the time taken by minmax increases as the depth increases because the number of nodes searched increase exponentially

With alpha beta pruning:

Depth 1

The image displays two screenshots of a PyCharm IDE, showing the execution of a game tree search algorithm. The top screenshot shows the initial state of the search, with a score of 0. The bottom screenshot shows the state after a move, with a score of 5 and a visual representation of the game board.

Top Screenshot:

- File Explorer: `Lab2AIconnect4` (422), `venv` (423), `main.py` (424), `External Libraries` (425), `Scratches and Console`.
- Code Editor: `main.py` (422-425).

```
stringg = f"Max score:{score}, Next column: {nextcol}"
search_tree.update_node(0, tag=stringg)
search_tree.show()
if nextcol == None:
```
- Run Console: `main`.
 - Runtime of minmax = 0.0029544830322265625 seconds
 - number of nodes 7
 - Max score:0, Next column: 0
 - Maximizing node (no pruning occurred), alpha: 0 beta: inf score: 0
 - Maximizing node, alpha: -inf beta: inf score: -inf
 - Maximizing node, alpha: 0 beta: inf score: 0
 - Maximizing node, alpha: 0 beta: inf score: 0
 - Maximizing node, alpha: 0 beta: inf score: 0
 - Maximizing node, alpha: 0 beta: inf score: 0
 - Maximizing node, alpha: 0 beta: inf score: 0

Bottom Screenshot:

- File Explorer: `Lab2AIconnect4` (422), `venv` (423), `main.py` (424), `External Libraries` (425), `Scratches and Console`.
- Code Editor: `main.py` (422-425).

```
stringg = f"Max score:{score}, Next column: {nextcol}"
search_tree.update_node(0, tag=stringg)
search_tree.show()
if nextcol == None:
```
- Run Console: `main`.
 - Runtime of minmax = 0.002890348434448242 seconds
 - number of nodes 7
 - Max score:5, Next column: 0
 - Maximizing node (no pruning occurred), alpha: 5 beta: inf score: 5
 - Maximizing node, alpha: -inf beta: inf score: -inf
 - Maximizing node, alpha: 5 beta: inf score: 5
 - Maximizing node, alpha: 5 beta: inf score: 5
 - Maximizing node, alpha: 5 beta: inf score: 5
 - Maximizing node, alpha: 5 beta: inf score: 5
 - Maximizing node, alpha: 5 beta: inf score: 5

Depth 2:

The image shows a code editor with a Go board and a terminal window. The Go board is a 9x9 grid with pieces placed at (4,4), (4,5), (5,4), (5,5), (6,4), (6,5), (7,4), (7,5), (8,4), (8,5), (8,6), and (9,4). The terminal window displays the output of a minmax search for Depth 2.

```
stringg = f"Max score:{score}, Next col"
search_tree.update_node(0, tag=stringg)
search_tree.show()
if nextcol == None:
    if (K>0 and (prune==1) or (prune... while True for event in pygame.event.get() if event.type == pygame...
```

Run: main

```
Runtime of minmax = 0.01159214973449707 seconds
number of nodes 41
Max score:-2, Next column: 3
| Maximizing node (no pruning occurred), alpha: -2 beta: inf score: -2
| | Minimizing node, alpha: -2 beta: 0 score: 0
| | Minimizing node, alpha: -2 beta: inf score: inf
| | Minimizing node (pruning occurred), alpha: -2 beta: -2 score: -2
| Maximizing node, alpha: -2 beta: inf score: -2
| | Minimizing node, alpha: -2 beta: 0 score: 0
| | Minimizing node, alpha: -2 beta: inf score: inf
| | Minimizing node (pruning occurred), alpha: -2 beta: -2 score: -2
| Maximizing node, alpha: -2 beta: inf score: -2
| | Minimizing node, alpha: -2 beta: 0 score: 0
| | Minimizing node, alpha: -2 beta: inf score: inf
| | Minimizing node (pruning occurred), alpha: -2 beta: -2 score: -2
| Maximizing node, alpha: -5 beta: inf score: -5
| | Minimizing node, alpha: -5 beta: 0 score: 0
| | Minimizing node, alpha: -5 beta: 0 score: 0
```

Version Control Run TODO Problems Terminal Python Packages Python Console

```
if (K>0 and (prune==1) or (prune... while True for event in pygame.event.get() if event.type == pygame...
```

main

```
Runtime of minmax = 0.012321233749389648 seconds
number of nodes 42
Max score:-2, Next column: 1
| Maximizing node (no pruning occurred), alpha: -2 beta: inf score: -2
| | Minimizing node, alpha: -2 beta: 0 score: 0
| | Minimizing node, alpha: -2 beta: 0 score: 0
| | Minimizing node, alpha: -2 beta: inf score: inf
| | Minimizing node (pruning occurred), alpha: -2 beta: -2 score: -2
| Maximizing node, alpha: -2 beta: inf score: -2
| | Minimizing node, alpha: -2 beta: 0 score: 0
| | Minimizing node, alpha: -2 beta: 0 score: 0
| | Minimizing node, alpha: -2 beta: inf score: inf
| | Minimizing node (pruning occurred), alpha: -2 beta: -2 score: -2
| Maximizing node, alpha: -2 beta: inf score: -2
| | Minimizing node, alpha: -2 beta: 0 score: 0
| | Minimizing node, alpha: -2 beta: 0 score: 0
| | Minimizing node, alpha: -2 beta: 0 score: 0
| | Minimizing node, alpha: -2 beta: inf score: inf
```

Version Control Run TODO Problems Terminal Python Packages Python Console

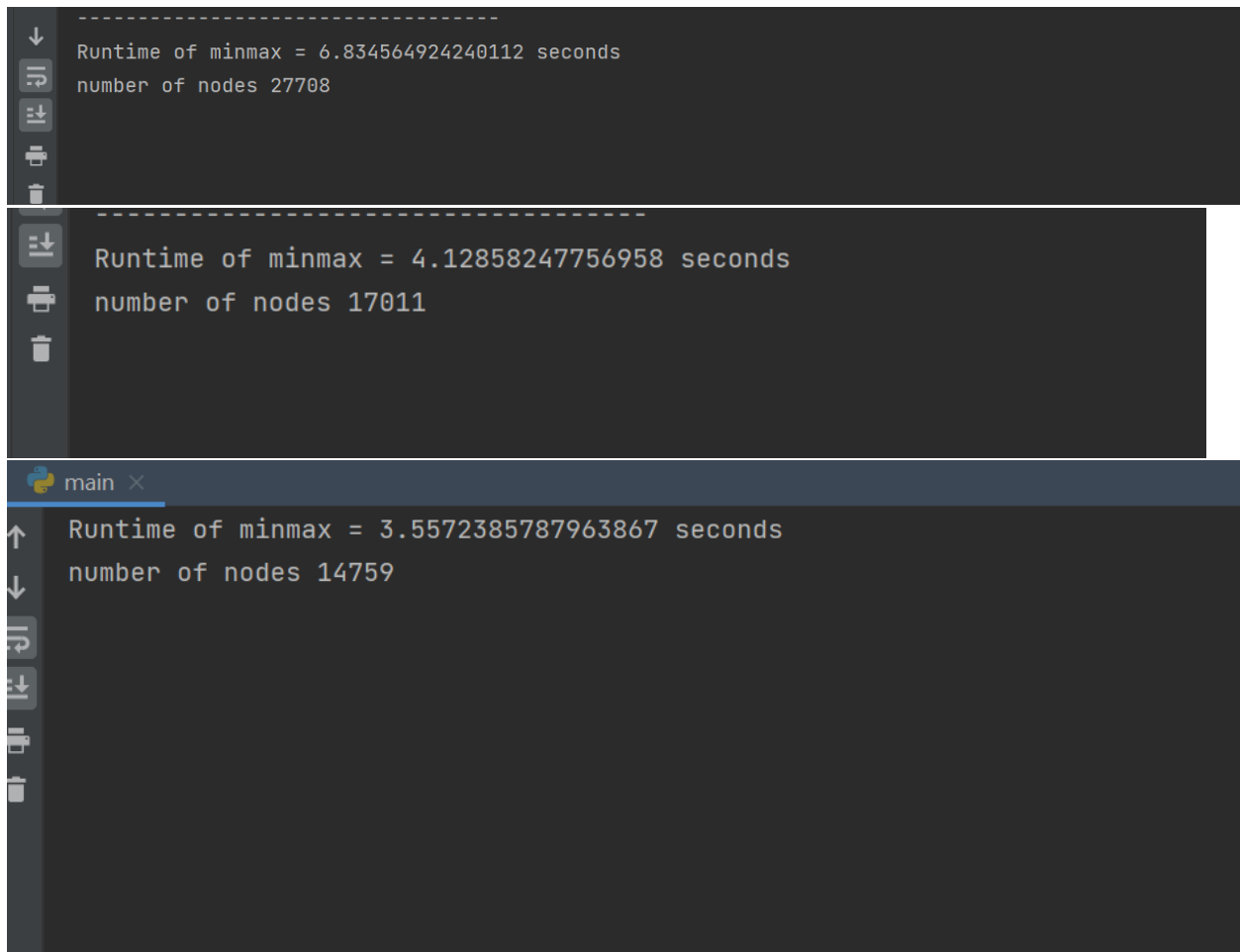
Depth 4

```
395 sys.exit()
396
397 if event.type == pygame.MOUSEMOTION:
398     pygame.draw.rect(screen, GREY, (0, 0, width, SQUARESIZE))
    if (K>0 and (prune==1) or (prune... > while True > for event in pygame.event.get() > if event.type == pygan

Run: main ×
-----
Runtime of minmax = 0.23752403259277344 seconds
number of nodes 967
Max score:-2, Next column: 4
|-----|
| Maximizing node (no pruning occurred), alpha: -2 beta: inf score: -2
| |-----|
| | Minimizing node, alpha: -2 beta: inf score: inf
| | |-----|
| | | Maximizing node (no pruning occurred), alpha: 0 beta: inf score: 0
| | | |-----|
| | | | Minimizing node, alpha: 0 beta: 2 score: 2
| | | | Minimizing node, alpha: 0 beta: 2 score: 2
| | | | Minimizing node, alpha: 0 beta: 2 score: 2
| | | | Minimizing node, alpha: 0 beta: inf score: inf
| | | | Minimizing node (pruning occurred), alpha: 0 beta: -6 score: -6
| | | |-----|
| | | | Maximizing node, alpha: -2 beta: inf score: -3
| | | | |-----|
| | | | | Minimizing node, alpha: -2 beta: 2 score: 2
| | | | | Minimizing node, alpha: -2 beta: 2 score: 2
| | | | | Minimizing node, alpha: -2 beta: 2 score: 2
| | | | | Minimizing node, alpha: -2 beta: inf score: inf
| | | | | Minimizing node (pruning occurred), alpha: -2 beta: -6 score: -6
| | | | |-----|
| | | | | event.type == pygame.MOUSEMOTION:
| | | | | pygame.draw.rect(screen, GREY, (0, 0, width, SQUARESIZE)
    if (K>0 and (prune==1) or (prune... > while True > for event in pygame.event.get() > if event.type ==

Run: main ×
| 1 | 2 | 2 | 1 | 2 |
|-----|
Runtime of minmax = 0.1890861988067627 seconds
number of nodes 800
Max score:-2, Next column: 1
|-----|
| Maximizing node (no pruning occurred), alpha: -2 beta: inf score: -2
| |-----|
| | Minimizing node, alpha: -2 beta: inf score: inf
| | |-----|
| | | Maximizing node (no pruning occurred), alpha: 0 beta: inf score: 0
| | | |-----|
| | | | Minimizing node (pruning occurred), alpha: 0 beta: 0 score: 0
| | | |-----|
| | | | Maximizing node, alpha: -2 beta: inf score: -2
| | | | |-----|
| | | | | Minimizing node, alpha: -2 beta: inf score: inf
| | | | | Minimizing node (pruning occurred), alpha: -2 beta: -2 score: -2
| | | | |-----|
| | | | | Maximizing node, alpha: -2 beta: inf score: -2
| | | | | |-----|
| | | | | | Minimizing node, alpha: -2 beta: 0 score: 0
| | | | | | Minimizing node, alpha: -2 beta: 0 score: 0
| | | | | | Minimizing node, alpha: -2 beta: inf score: inf
| | | | | | Minimizing node (pruning occurred), alpha: -2 beta: -2 score: -2
| | | | | |-----|
| | | | | | Maximizing node, alpha: -2 beta: inf score: -2
```

Depth 6:

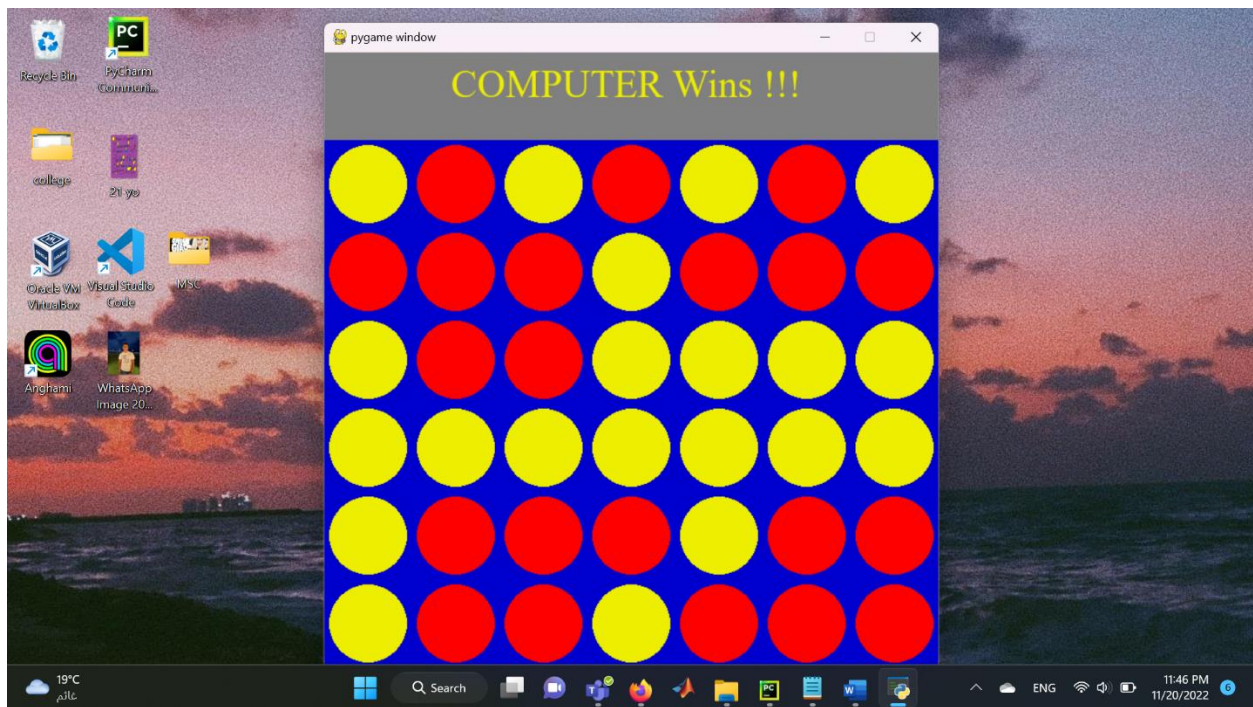
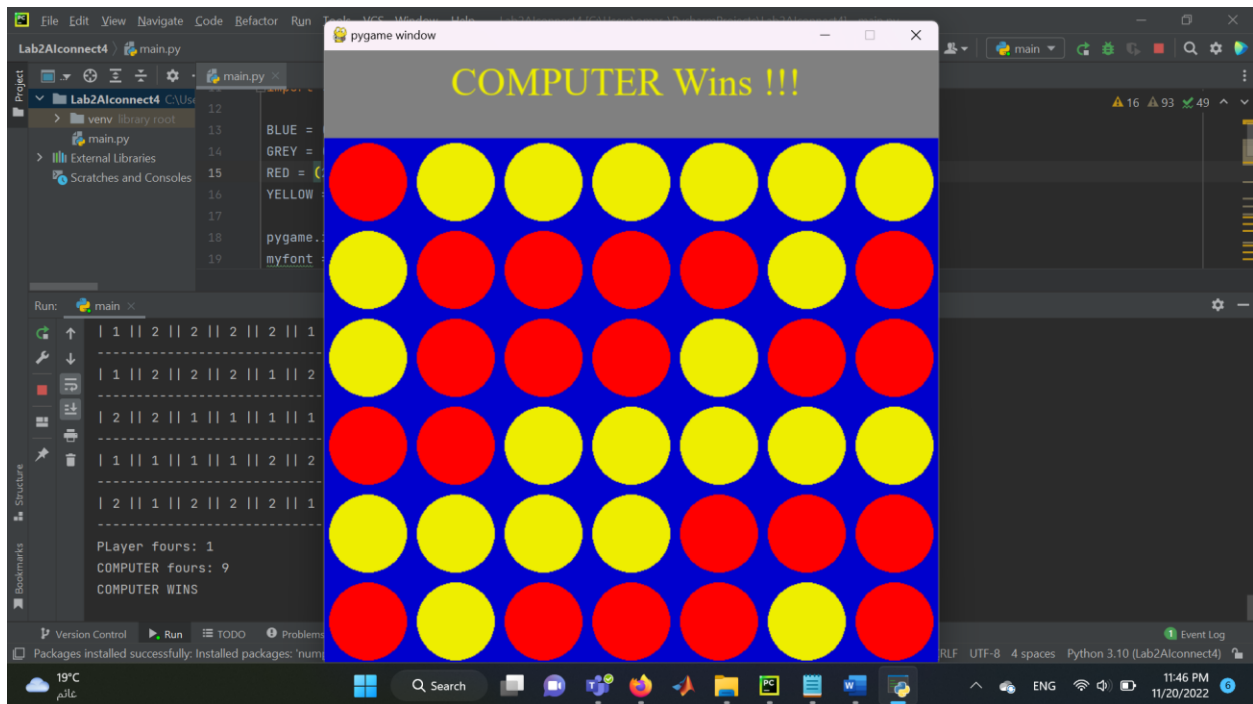


```
-----  
Runtime of minmax = 6.834564924240112 seconds  
number of nodes 27708  
  
-----  
Runtime of minmax = 4.12858247756958 seconds  
number of nodes 17011  
  
main ×  
Runtime of minmax = 3.5572385787963867 seconds  
number of nodes 14759
```

As show the time increases with depth increasing but when we increase the depth

But on average even with depth 6 using alpha beta pruning will be better than using depth 5 without alpha beta because the alpha beta doesn't always have to search the whole tree

Here are two full games first is using alpha beta and the other is without
And depth 4



Node expanded in different depths in above testcases

Depth	Alpha beta	No alpha beta
1	7 nodes and 0.029 secs	7 nodes and 0.0026 secs
2	41 nodes and 0.01 secs	56 nodes and 0.014 secs
3	119 nodes and 0.03 secs	399 nodes an 0.1 secs
4	976 nodes and 0.2 secs	2800 nodes and 0.83 secs
5	2315 nodes and 0.613 secs	19607 nodes and 5.4 secs
6	17011 nodes and 4.1 secs	137255 nodes and 41.25 secs