

Class Game:

Στην κλάση Game περιέχεται η μέθοδος main του προγράμματος. Αρχικά, μέσα στη main δημιουργούμε ορισμένα αντικείμενα που θα χρειαστούμε και δηλώνουμε μία μεταβλητή και δύο HashSet με τα οποία θα χειριζόμαστε τις διαθέσιμες κινήσεις των δύο παικτών. Στη συνέχεια, ζητάμε είσοδο από τον χρήστη μέσω της κλάσης Scanner. Έχοντας ορίσει ο αριθμός 1 να αντιπροσωπεύει τα άσπρα πούλια και ο αριθμός 2 τα μαύρα, απαιτούμε από τον χρήστη να πληκτρολογήσει είτε 1 είτε 2. Μετά την επιλογή του χρώματος από τον χρήστη, του ζητάμε να επιλέξει το επίπεδο δυσκολίας του παιχνιδιού ελέγχοντας κάθε φορά η είσοδος από τον χρήστη να είναι ένας θετικός αριθμός. Έπειτα με την κλήση της μεθόδου getUser() της κλάσης Board ελέγχουμε το χρώμα των πουλιών που επέλεξε ο χρήστης και ανάλογα αυτό δίνουμε στον υπολογιστή το άλλο χρώμα. Επιπλέον, αρχικοποιούμε τα δύο HashSet με την κλήση των μεθόδων availableMovesWhite() και availableMovesBlack() της κλάσης Board και με βάση το χρώμα που είχε επιλέξει ο χρήστης θέτουμε στη μεταβλητή playerTurn είτε την τιμή True είτε την τιμή False. Εφόσον, την πρώτη κίνηση την έχουν τα μαύρα αν ο παίκτης έχει τα μαύρα η μεταβλητή playerTurn θα πάρει την τιμή True, διαφορετικά θα πάρει την τιμή False. Μόλις το παιχνίδι ξεκινήσει εκτυπώνουμε την αρχική κατάσταση του πίνακα και ο παίκτης που έχει τα μαύρα πούλια κάνει την πρώτη κίνηση. Για όσο τουλάχιστον ένας από τους δύο παίκτες έχει διαθέσιμη κίνηση εκτελείται ο βρόχος μέσα στην while. Αυτό το πετυχαίνουμε ελέγχοντας κάθε φορά αν το HashSet του κάθε παίκτη είναι κενό. Αν είναι η σειρά του χρήστη να κάνει κίνηση, εκτυπώνουμε τις διαθέσιμες κινήσεις του ώστε να επιλέξει μία από αυτές, ελέγχοντας κάθε φορά αν οι συντεταγμένες της κίνησης που θέλει να κάνει είναι έγκυρες μέσω της μεθόδου isValidMove() της κλάσης Board. Αφού σιγουρευτούμε ότι πληκτρολόγησε έγκυρη κίνηση, εκτελούμε την κίνηση μέσω της μεθόδου MakeMove() της κλάσης Board. Μετά από κάθε κίνηση εκτυπώνουμε την κατάσταση που προέκυψε, "καθαρίζουμε" τα δύο HashSet με την clear(), τα ξανά αρχικοποιούμε και αλλάζουμε την τιμή της μεταβλητής playerTurn. Αν ήταν True και ο αντίπαλος έχει διαθέσιμες κινήσεις την κάνουμε False, ενώ αν ήταν False και ο αντίπαλος έχει διαθέσιμες κινήσεις την κάνουμε True. Στην περίπτωση που έχει σειρά ο υπολογιστής εκτυπώνουμε τις διαθέσιμες κινήσεις του με το αντίστοιχο score και ύστερα καλούμε την μέθοδο minimax() μέσω της οποίας θα γίνει η κίνηση του υπολογιστή. Τέλος, αν η συνθήκη της while δεν ικανοποιείται πια, τότε το παιχνίδι έχει τελειώσει και εκτυπώνουμε το τελικό score.

Class Board:

Η κλάση Board προσομοιώνει τον πίνακα του παιχνιδιού και όλες τις λειτουργίες του. Αρχικά, δηλώνουμε μερικές private μεταβλητές που θα χρειαστούμε όπως την winner η οποία παίρνει την τιμή 1 αν νικήσουν τα άσπρα πούλια και την τιμή 2 αν νικήσουν τα μαύρα, τις user και ai οι οποίες παίρνουν τις τιμές 1, 2 και αντιπροσωπεύουν το χρώμα των πουλιών που έχουν ο χρήστης και ο υπολογιστής και τέλος τις scoreBlack, scoreWhite οι οποίες κρατούν το score του κάθε παίκτη. Επίσης, εκχωρούμε την αρχική κατάσταση του πίνακα στην μεταβλητή board. Η κλάση Board διαθέτει δύο κατασκευαστές ο πρώτος δημιουργεί ένα αντίγραφο της τρέχουσας κατάστασης του πίνακα και ο δεύτερος αρχικοποιεί το score των δύο παικτών στο 0. Στην συνέχεια υπάρχουν μέθοδοι get και set. Οι μέθοδοι getScoreBlack(), getScoreWhite() υπολογίζουν και επιστρέφουν το score των αντίστοιχων πουλιών ενώ οι getBoard(), getUser(), getAi() επιστρέφουν τον πίνακα και τα αναγνωριστικά του παίκτη και του υπολογιστή αντίστοιχα. Από την άλλη υπάρχουν και οι αντίστοιχες μέθοδοι set μέσω των οποίων μπορούμε να αλλάξουμε τις τιμές των παραπάνω μεταβλητών. Στη συνέχεια έχουμε τις μεθόδους getWinner(), isValidMove(). Η πρώτη υπολογίζει το score των δύο παικτών, τα συγκρίνει και επιστρέφει το αναγνωριστικό του νικητή (σε περίπτωση ισοπαλίας το 0), ενώ η δεύτερη καλή την getWinner() και με βάση τον αναγνωριστικό του νικητή επιστρέφει μία boolean τιμή. Οι μέθοδοι availableMovesBlack(), availableMovesWhite() εκτελούν την ίδια ακριβώς λειτουργία. Ορίζουμε ένα HashSet στο οποίο εκχωρούνται οι διαθέσιμες κινήσεις, στην availableMovesBlack() για τα μαύρα ενώ στην availableMovesWhite() για τα άσπρα και έπειτα για κάθε κενή θέση στον πίνακα κρατάμε τις συντεταγμένες της και ελέγχουμε προς κάθε κατεύθυνση αν υπάρχουν πούλια αντιπάλου. Αν υπάρχουν (είτε 1 είτε περισσότερα) τότε προχωράμε προς τη συγκεκριμένη κατεύθυνση μέχρι να συναντήσουμε ίδιου χρώματος πούλι (μαύρο για την availableMovesBlack() και άσπρο για την availableMovesWhite()). Σε περίπτωση που δεν συναντήσουμε τότε με την εντολή continue επιστρέφουμε στην for και ψάχνουμε για άλλη κενή θέση, διαφορετικά τοποθετούμε την κενή θέση στις διαθέσιμες κινήσεις για το αντίστοιχο χρώμα και όταν ελεγχθεί όλος ο πίνακας επιστρέφουμε το αντίστοιχο HashSet. Ακολουθεί η μέθοδος MakeMove() μέσω της οποίας εκτελείται μία διαθέσιμη κίνηση. Παίρνει ως όρισμα ένα αντικείμενο τύπου Checker και μέσα από αυτό ελέγχει το αναγνωριστικό του παίκτη που κάλεσε την MakeMove() για να πραγματοποιήσει μία του κίνηση και το εκχωρεί στη μεταβλητή player ενώ το αναγνωριστικό που μένει το εκχωρεί στην μεταβλητή enemy. Μέσα από το αντικείμενο παίρνει και τις συντεταγμένες της κίνησης που θέλει να κάνει ο παίκτης και τις εκχωρεί στις μεταβλητές i, j. Τοποθετεί το αναγνωριστικό του παίκτη στην θέση την οποία επιθυμεί να γίνει η κίνηση και κρατά τις συντεταγμένες της θέσης αυτής στις μεταβλητές a, b. Ύστερα σαρώνει τον πίνακα προς κάθε κατεύθυνση από την θέση της κίνησης που πραγματοποιήσε και για όσο συναντά εχθρικό πούλι προχωρά. Την στιγμή που θα συναντήσει δικό του πούλι θα επιστρέψει στην αρχική θέση (την οποία έχουμε κρατήσει στις μεταβλητές a, b) και αντιμεταθέτει όλα τα εχθρικά πούλια. Είτε συναντήσει εχθρικό πούλι είτε όχι (προς την κατεύθυνση που ψάχνει), μόλις φτάσει στα όρια του πίνακα για αυτή την κατεύθυνση επιστρέφει στην αρχική θέση (δηλαδή στις a, b) και σαρώνει προς κάποια άλλη κατεύθυνση μέχρι να τις ελέγξει όλες. Τέλος, υπάρχουν οι μέθοδοι isValidMove() και displayBoard(). Η πρώτη ελέγχει αν είναι έγκυρη η κίνηση σαρώνοντας το αντίστοιχο HashSet και η δεύτερη εκτυπώνει την τρέχουσα κατάσταση του πίνακα.

->Checker()

Η κλάση αυτή αναπαριστά κάθε πουλι στον πίνακα. Για να συμβεί αυτό κρατάμε χαρακτηριστικά όπως την θέση του στον πίνακα (x,y) καθώς και το χρώμα του το οποίο συμβολίζουμε με 1 για λευκό και 2 για μαύρο, τα οποία αρχικοποιούμε μέσω των 3 κατασκευαστών που έχουμε φτιάξει. Επίσης, η κλάση Checker περιέχει τις μεθόδους get οι οποίες μας δίνουν την θέση που έχει το πουλι στον πίνακα και το χρώμα του καθώς και την μέθοδο equals η οποία επιβεβαιώνει τα χαρακτηριστικά που έχει το πουλι.

->AI()

Στην κλάση αυτή έχουμε την μέθοδο minimax η οποία εκτελεί τον αλγόριθμο α-β pruning για να βρούμε αναδρομικά την καλύτερη κίνηση που μπορεί να κάνει ο υπολογιστής. Μέσω της minimax καλούμε τις μεθόδους evaluatePosition(), getMax(), getMin() οι οποίες βοηθάνε αντίστοιχα στην αξιολόγηση των δυνατών κινήσεων, την επιλογή της καλύτερης κίνησης για τον υπολογιστή και στον υπολογισμό της καλύτερης κίνησης για τον παίκτη. Ειδικότερα μόλις η getMax() εντοπίζει την καλύτερη κίνηση την κάνει και στον αρχικό πίνακα. Επίσης, έχουμε την μέθοδο evaluatePosition η οποία αποτελεί την ευρετική συνάρτηση που χρησιμοποιούμε για την επιλογή της καλύτερης κίνησης. Η συγκεκριμένη ευρετική αξιοποιεί τα βάρη θέσεων τα οποία είναι περισσότερο σημαντικά για κάποια σημεία του πίνακα και λιγότερο σημαντικά για άλλα στοιχεία του πίνακα. Επίσης, στην κλάση εκτελούμε και κάποιες μεθόδους get οι οποίες μας επιστρέφουν τις συντεταγμένες της καλύτερης θέσης και το βάθος της αναζήτησης.

Παραδείγματα εισόδων-εξόδων του προγράμματος

```
eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help

<terminated> Game (4) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (13 Νοε 2018, 6:53:01 μ.μ.)
Give 1 if you want white or 2 for black: 2
Please choose difficulty: 20
The game starts.....!!

 0 1 2 3 4 5 6 7
0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0
3 0 0 0 1 2 0 0 0
4 0 0 0 2 1 0 0 0
5 0 0 0 0 0 0 0 0
6 0 0 0 0 0 0 0 0
7 0 0 0 0 0 0 0 0

Please choose one of the moves below:
You can do this move: (2 ,3)
You can do this move: (3 ,2)
You can do this move: (4 ,5)
You can do this move: (5 ,4)
Steps to go horizontal:2
Steps to go vertical:3
After player Move our board is.....

 0 1 2 3 4 5 6 7
0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0
2 0 0 0 2 0 0 0 0
3 0 0 0 2 2 0 0 0
4 0 0 0 2 1 0 0 0
5 0 0 0 0 0 0 0 0
6 0 0 0 0 0 0 0 0
7 0 0 0 0 0 0 0 0

Computer can do this move: (2 ,2)with score: 0.51
Computer can do this move: (2 ,4)with score: 0.07
Computer can do this move: (4 ,2)with score: 0.04
Thinking.....
After Computer Move our board is.....
```

```
eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help

<terminated> Game (4) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (13 Νοε 2018, 6:53:01 μ.μ.)
You can do this move: (3 ,4)
Steps to go horizontal:2
Steps to go vertical:3
After player Move our board is.....

 0 1 2 3 4 5 6 7
0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0
2 0 0 0 2 0 0 0 0
3 0 0 0 2 2 0 0 0
4 0 0 0 2 1 0 0 0
5 0 0 0 0 0 0 0 0
6 0 0 0 0 0 0 0 0
7 0 0 0 0 0 0 0 0

Computer can do this move: (2 ,2)with score: 0.51
Computer can do this move: (2 ,4)with score: 0.07
Computer can do this move: (4 ,2)with score: 0.04
Thinking.....
After Computer Move our board is.....

 0 1 2 3 4 5 6 7
0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0
2 0 0 1 2 0 0 0 0
3 0 0 0 1 2 0 0 0
4 0 0 0 2 1 0 0 0
5 0 0 0 0 0 0 0 0
6 0 0 0 0 0 0 0 0
7 0 0 0 0 0 0 0 0

Please choose one of the moves below:
You can do this move: (2 ,1)
You can do this move: (3 ,2)
You can do this move: (4 ,5)
You can do this move: (5 ,4)
4
Steps to go horizontal:2
```

```
eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help

<terminated> Game (4) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (13 Nov 2018, 6:53:01 μμ.)

Steps to go horizontal:2
Steps to go vertical:1
After player Move our board is.....

  0 1 2 3 4 5 6 7
0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0
2 0 2 2 2 0 0 0
3 0 0 0 1 2 0 0
4 0 0 0 2 1 0 0
5 0 0 0 0 0 0 0
6 0 0 0 0 0 0 0
7 0 0 0 0 0 0 0

Computer can do this move: (1 ,1)with score: 1.81
Computer can do this move: (1 ,3)with score: 0.27
Computer can do this move: (2 ,4)with score: 0.07
Computer can do this move: (3 ,5)with score: 0.04
Computer can do this move: (4 ,2)with score: 0.04
Computer can do this move: (5 ,3)with score: 0.07
Thinking.....
After Computer Move our board is.....

  0 1 2 3 4 5 6 7
0 0 0 0 0 0 0 0
1 0 1 0 0 0 0 0
2 0 2 1 2 0 0 0
3 0 0 0 1 2 0 0
4 0 0 0 2 1 0 0
5 0 0 0 0 0 0 0
6 0 0 0 0 0 0 0
7 0 0 0 0 0 0 0

Please choose one of the moves below:
You can do this move: (0 ,1)
You can do this move: (3 ,2)
You can do this move: (4 ,5)|
<
```

```
eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help

<terminated> Game (4) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (13 Nov 2018, 6:53:01 μμ.)

Steps to go horizontal:7
Steps to go vertical:4
After player Move our board is.....

  0 1 2 3 4 5 6 7
0 1 1 1 1 1 1 1
1 1 2 1 1 2 1 1
2 1 2 1 1 1 1 2
3 1 2 1 1 1 1 2
4 1 1 1 1 2 2 1
5 1 1 1 1 2 1 2
6 2 2 2 2 2 2 2
7 2 2 2 0 2 0 0

Computer can do this move: (7 ,3)with score: 0.43
Computer can do this move: (7 ,5)with score: 0.99
Computer can do this move: (7 ,6)with score: 3.03
Computer can do this move: (7 ,7)with score: 16.16
Thinking.....
After Computer Move our board is.....

  0 1 2 3 4 5 6 7
0 1 1 1 1 1 1 1
1 1 2 1 1 2 1 1
2 1 2 1 1 1 1 2
3 1 2 1 1 1 1 2
4 1 1 1 1 2 2 1
5 1 1 1 1 2 1 2
6 2 2 1 1 1 2 2
7 2 2 2 1 2 0 0

Please choose one of the moves below:
You can do this move: (7 ,5)
1
Steps to go horizontal:7
Steps to go vertical:5
After player Move our board is.....
<
```

```
eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help

<terminated> Game (4) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (13 Nov 2018, 6:53:01 μ.μ.)
You can do this move: (7,5)
1
Steps to go horizontal:7
Steps to go vertical:5
After player Move our board is.....

 0 1 2 3 4 5 6 7
0 1 1 1 1 1 1 1
1 1 2 1 1 2 1 1
2 1 2 1 1 1 1 2
3 1 2 1 1 1 1 2
4 1 1 2 1 2 2 1
5 1 1 1 2 2 1 2
6 2 2 1 1 2 2 2
7 2 2 2 1 2 2 0

Computer can do this move: (7,6)with score: 3.03
Computer can do this move: (7,7)with score: 16.16
Thinking.....
After Computer Move our board is.....

 0 1 2 3 4 5 6 7
0 1 1 1 1 1 1 1
1 1 2 1 1 2 1 1
2 1 2 1 1 1 1 2
3 1 2 1 1 1 1 2
4 1 1 2 1 2 2 1
5 1 1 1 2 1 1 2
6 2 2 1 1 2 1 2
7 2 2 2 1 1 1 0

Please choose one of the moves below:
You can do this move: (7,7)
1
Steps to go horizontal:7
Steps to go vertical:7
After player Move our board is.....
```

```
eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help

<terminated> Game (4) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (13 Nov 2018, 6:53:01 μ.μ.)
5 1 1 1 2 2 1 2
6 2 2 1 1 2 2 2
7 2 2 2 1 2 2 0

Computer can do this move: (7,6)with score: 3.03
Computer can do this move: (7,7)with score: 16.16
Thinking.....
After Computer Move our board is.....

 0 1 2 3 4 5 6 7
0 1 1 1 1 1 1 1
1 1 2 1 1 2 1 1
2 1 2 1 1 1 1 2
3 1 2 1 1 1 1 2
4 1 1 2 1 2 2 1
5 1 1 1 2 1 1 2
6 2 2 1 1 2 1 2
7 2 2 2 1 1 1 0

Please choose one of the moves below:
You can do this move: (7,7)
1
Steps to go horizontal:7
Steps to go vertical:7
After player Move our board is.....

 0 1 2 3 4 5 6 7
0 1 1 1 1 1 1 1
1 1 2 1 1 2 1 1
2 1 2 1 1 1 1 2
3 1 2 1 1 1 1 2
4 1 1 2 1 2 2 1
5 1 1 1 2 1 1 2
6 2 2 1 1 2 1 2
7 2 2 2 2 2 2 2

Scores are: Black ->26 , White ->38
```