Αλέξανδρος Βεντούρας (ΑΜ: 3160013)
Νίκος Δούσης (ΑΜ: 3160030)
Δικαία Σωτηροπούλου (ΑΜ: 3160172,
email: dikeasot@hotmail.com)

Reversi/Othello

Για την υλοποίηση του παιχνιδιού *Reversi* ή *Othello* δημιουργήσαμε 4 κλάσεις με ονόματα *OurMain*, *MoveValidation* HumanPlayer και *ComputerPlayer*.

Αρχικά, η κλάση ***OurMain*** είναι η κύρια κλάση του προγράμματος μας. Κατά την εκτέλεση ζητείται από τον χρήστη να επιλέξει ποιος θα παίξει πρώτος, (εκείνος ή ο υπολογιστής) και στη συνέχει να επιλέξει το επίπεδο δυσκολίας στο οποίο θέλει να αντιμετωπίσει τον αντίπαλο του (δηλαδή τον υπολογιστή). Έπειτα, υπάρχει μια δομή επανάληψης που επιτρέπει στους δύο αντιπάλους να παίζουν εναλλάξ. Όταν δεν υπάρχουν διαθέσιμες κινήσεις, το παιχνίδι τερματίζει.

Στην κλάση ***MoveValidation*** ελέγχουμε αν μια κίνηση είναι έγκυρη ή όχι. Η μέθοδος *isValidMove* ελέγχει τις κατευθύνσεις γύρω από το κελί που θέλουμε να βάλουμε ένα πούλι. Πιο συγκεκριμένα:

- *Direction2:* έλεγχος στο κελί που βρίσκεται πάνω από το κελί που ελέγχουμε,
- *Direction7:* έλεγχος στο κελί που βρίσκεται κάτω από το κελί που ελέγχουμε,
- *Direction4:* έλεγχος στο κελί που βρίσκεται αριστερά από το κελί που ελέγχουμε,
- *Direction5:* έλεγχος στο κελί που βρίσκεται δεξιά από το κελί που ελέγχουμε,
- *Direction1:* έλεγχος στο κελί που βρίσκεται πάνω και αριστερά από το κελί που ελέγχουμε,
- *Direction3:* έλεγχος στο κελί που βρίσκεται πάνω και δεξιά από το κελί που ελέγχουμε,
- *Direction6:* έλεγχος στο κελί που βρίσκεται κάτω και αριστερά από το κελί που ελέγχουμε,
- *Direction8:* έλεγχος στο κελί που βρίσκεται κάτω και δεξιά από το κελί που ελέγχουμε.

Στη συνέχεια, οι μέθοδοι *moveAftermath* και *changeColour* αλλάζουν τα πούλια στον πίνακα του παιχνιδιού.

Αλέξανδρος Βεντούρας (ΑΜ: 3160013)
Νίκος Δούσης (ΑΜ: 3160030)
Δικαία Σωτηροπούλου (ΑΜ: 3160172,
email: dikeasot@hotmail.com)

Η κλάση ***HumanPlayer*** κάνει τις κινήσεις του χρήστη. Η μέθοδος *createArray* δημιουργεί έναν δισδιάστατο πίνακα αρχικοποιώντας τον με 4 πούλια τοποθετημένα στο κέντρο (αρχική κατάσταση παιχνιδιού). Η μέθοδος *selectSquare* ζητάει από τον χρήστη τη θέση στην οποία θέλει να παίξει, εμφανίζει κατάλληλα μηνύματα σχετικά με την εγκυρότητα της θέσης αυτής και αν είναι έγκυρη κάνει την κίνηση αυτή. Η *printArray* τυπώνει το ταμπλό με τα πούλια που έχουν τοποθετηθεί μέχρι εκείνη τη στιγμή. Η *finalCount* υπολογίζει τον αριθμό των πουλιών που έχει κάθε παίκτης στο τέλος του παιχνιδιού, ενώ η μέθοδος *count* μετράει πόσα υπάρχουν κατά τη διάρκεια του παιχνιδιού. Τέλος, η *isTerminal* παίρνει την τιμή true όταν κανένας από τους δύο παίκτες δε μπορεί να κάνει κάποια κίνηση.

Η κλάση ***ComputerPlayer*** κάνει τις κινήσεις του υπολογιστή. Δημιουργούμε μια κλάση *Node* η οποία περιέχει τον τρέχον πίνακα και το σκορ του (πόσο καλά παίζει ο υπολογιστής). Στη μέθοδο *heuristic* υλοποιούνται 4 ευρετικές, μια για τις γωνίες του πίνακα, μια για τις σειρές και τις στήλες που βρίσκονται στα άκρα του πίνακα, μια για το εσωτερικό τετράγωνο (4x4) του πίνακα και μια για τις κινήσεις που δε θεωρούνται καλές. Επιπλέον, έχουμε τη μέθοδο *getChildren*, όπου δημιουργούνται τα παιδιά (μελλοντικές καταστάσεις) είτε του τωρινού, είτε μελλοντικών πινάκων/καταστάσεων. Έπειτα υλοποιείται ο αλγόριθμος *MiniMax* καλώντας αναδρομικά τις συναρτήσεις *min* και *max* βρίσκοντας ποια είναι η καλύτερη δυνατή κίνηση για τον υπολογιστή.

Ακολουθούν φωτογραφίες με την αρχή του παιχνιδιού και ένα ενδεικτικό τέλος του προγράμματος:

Αλέξανδρος Βεντούρας (ΑΜ: 3160013)
Νίκος Δούσης (ΑΜ: 3160030)
Δικαία Σωτηροπούλου (ΑΜ: 3160172,
email: dikeasot@hotmail.com)

```
C:\Users\Nikolas98\Desktop\ergasia>java OurMain
  1 2 3 4 5 6 7 8
1 - - - - - - - -
2 - - - - - - - -
3 - - - - - - - -
4 - - - O X - - -
5 - - - X O - - -
6 - - - - - - - -
7 - - - - - - - -
8 - - - - - - - -
Who do you want to play first?
Computer (O) or You (X)?

X
Please choose difficulty for the game.

4
Now it is your (X) turn!
Please select the square you want to place your piece:
Select the row:
4
Select the column:
3
You: 4 VS Computer: 1
  1 2 3 4 5 6 7 8
1 - - - - - - - -
2 - - - - - - - -
3 - - - - - - - -
4 - - X X X - - -
5 - - - X O - - -
6 - - - - - - - -
7 - - - - - - - -
8 - - - - - - - -
Computer is thinking, please wait...

You: 3 VS Computer: 3
  1 2 3 4 5 6 7 8
1 - - - - - - - -
2 - - - - - - - -
3 - - O - - - - -
4 - - X O X - - -
5 - - - X O - - -
6 - - - - - - - -
7 - - - - - - - -
8 - - - - - - - -
Now it is your (X) turn!
Please select the square you want to place your piece:
Select the row:
```

Αλέξανδρος Βεντούρας (ΑΜ: 3160013)
Νίκος Δούσης (ΑΜ: 3160030)
Δικαία Σωτηροπούλου (ΑΜ: 3160172,
email: dikeasot@hotmail.com)

```
Now it is your (X) turn!
Please select the square you want to place your piece:
Select the row:
7
Select the column:
7
You: 37 VS Computer: 27
  1 2 3 4 5 6 7 8
1 X O O O O O O O
2 X O O X O O O O
3 X O X X X X O O
4 X O X X X O X O
5 X X O X X O X O
6 X X O O X X X O
7 X X X X X X X O
8 X X X X X X X O
Computer is thinking, please wait...

Computer can make no moves, and so it passes.
  1 2 3 4 5 6 7 8
1 X O O O O O O O
2 X O O X O O O O
3 X O X X X X O O
4 X O X X X O X O
5 X X O X X O X O
6 X X O O X X X O
7 X X X X X X X O
8 X X X X X X X O
Now it is your (X) turn!
You cannot play; the computer plays again.
You: 37 VS Computer: 27
The game is over.
Congratulations! You won, with 37 vs 27 pieces.
```