

Στρατηγική Ομάδας 004

Βασικές κλάσεις της στρατηγικής μας αποτελούν οι κλάσεις `Node` και `CombinationTree`. Η λογική μας είναι η εξής: Ο `smartplayer` καθώς παίρνει τα φύλλα στα χέρια του, δημιουργεί στο μυαλό του όλους τους πιθανούς συνδυασμούς – δρόμους που μπορεί να ακολουθήσει κατά την διάρκεια ενός γύρου. Έχοντας δημιουργήσει όλα τα πιθανά πακέτα συνδυασμών, αξιολογεί το κάθε συνδυασμό κάθε πακέτου. Μια παράμετρος `px` που λαμβάνεται υποψιν είναι πόσοι συνδυασμοί μπορούν να δημιουργηθούν σε μία τράπουλα και έχουν μεγαλύτερη αξία από τον το κάθε συνδυασμό του πακέτου. Για παράδειγμα αν δημιουργήσει μία μονοφυλλία με (K) ρήγα, τότε αντιλαμβάνεται ότι ο συνδυασμός αυτός μπορεί να χάσει μόνο από κάποιον άσσο ή από κάποιο `SpecialCard` (αν δεν έχουν παιχτεί αυτά τα φύλλα).

Έτσι, κάθε πακέτο (το οποίο αποτελείται από κάποιο πλήθος συνδυασμών) λαμβάνει μία αξιολόγησή. Το πακέτο με την μεγαλύτερη αξιολόγηση επιλέγεται από τον `smartplayer`. Για να δημιουργηθούν τα πακέτα συνδυασμών χρησιμοποιείται η κλάση `Node` που έχει όλα τα χαρακτηριστικά ενός κομβού (`parent`, `child`, `addChild`) και έχει αντικείμενα ένα `hand` που είναι το `hand` του πατέρα ελλειπόμενο κατά τις κάρτες ενός συνδυασμού και ένα πίνακα συνδυασμών που έχει τους συνδυασμούς του πατέρα + 1

`CreateTree`:

Το δέντρο αναζήτησης ξεκινάει με ένα αρχικό κόμβο, την ρίζα (`root - Aragorn`) που περιλαμβάνει το χέρι του παίκτη τη δεδομένη στιγμή. Για να βρούμε όλους τους πιθανούς συνδυασμούς ψάχνει στη ρίζα ένα συνδυασμό (`pair`, 3 of a kind, straight, κλπ). Για κάθε συνδυασμό που βρίσκει (μία φορά για τους διαφορετικούς συνδυασμούς) δημιουργεί έναν καινούριο κόμβο με χέρι αυτό του πατέρα (εδώ `root`) χωρίς τα φύλλα του συνδυασμού και προσθέτει ως συνδυασμό, αυτόν που βρήκε. Επειτα ο αλγόριθμος καλεί αναδρομικά τον εαυτό του θέτοντας ως πατέρα το κάθε παιδί που βρίσκει. Όταν δεν μπορούμε να βρούμε άλλους συνδυασμούς, προσθέτουμε τις μονοφυλλίες μας σαν συνδυασμούς ως μονάδα και ονομάζουμε αυτόν τον κόμβο φύλλο (`leaf`). Τα φύλλα τα αποθηκεύουμε σε έναν πίνακα `leaves`. Στη συνέχεια αξιολογούμε κάθε φύλλο με μία αξία βάσει του πόσο "δυνατός" είναι ο κάθε συνδυασμός καθώς και πόσοι συνδυασμοί μπορούν να παιχτούν πάνω σε αυτόν (προκύπτει κρατώντας ιστορικό των φύλλων που έχουν παιχτεί και κατόπιν πιθανοκρατικής ανάλυσης). Το φύλλο με την μεγαλύτερη αξιολόγηση επιλέγεται από τον `smartplayer`.

Αξιολόγηση φύλλων:

Η αξιολόγηση των φύλλων είναι το άθροισμα της αξιολόγησης που δημιουργήσαμε για κάθε συνδυασμό. Λαμβάνεται υπόψιν η αξία του συνδυασμού, ένας αυθαίρετος συντελεστής με βάση την αξία του συνδυασμού (ώστε πχ μια βομβά από 4 να έχει μεγαλύτερη αξία από μια διφυλλία από 5αρια) και το σύνολο των συνδυασμών που μπορούν παιχτούν πάνω πάνω στον συνδυασμό με βάση τα εναπομείναντα φύλλα. Το τελευταίο υπολογίζεται με την βοήθεια 2 κλάσεων της `deckplayed` που αποθηκεύει ποια φύλλα που έχουν παιχτεί και που ο παίκτης έχει στα χέρια του σε ένα 2d Boolean πίνακα (και σε ένα 1d bool τα σπεςιαλ) και της `handTable`. Η `handTable` είναι η ίδια κλάση με τη `hand` με κάποιες εξτρα συναρτήσεις. Με βάση τους bool πίνακες βάζει στον εαυτό της τις κάρτες που δεν έχουν παιχτεί. Στη συνέχεια με τη συναρτηση `findCombination(Combination* comb)` βρίσκονται όλοι οι συνδυασμοί που μπορούν να παιχτούν πάνω στο `comb` αξιοποιώντας συναρτήσεις του `hand`.

Η δημιουργία δέντρων και η αξιολόγηση επαναλαμβάνεται κάθε φορά που έρχεται η σειρά του smartplayer.

setPhoenixValue:

Ο τρόπος υλοποίησης του smartplayer βασίζεται στις πιθανότητες ύπαρξης συνδυασμών του ίδιου τύπου που μπορούν να χτυπήσουν το κάθε συνδυασμό του smartplayer. Η συνάρτηση αυτή καλείται σε περίπτωση που ο παίκτης μας έχει τον φοίνικα στα χέρια του. Σε αυτήν την περίπτωση ο φοίνικας παίρνει όλες τις πιθανές τιμές που μπορεί να έχει στο παιχνίδι, δημιουργεί δηλαδή 14 διαφορετικά χέρια στον παίκτη που διαφέρουν μόνο στην τιμή του φοίνικα, και μέσα από την δημιουργία του CombinationTree (όλων των πιθανών τρόπων παιξίματος των καρτών που υπάρχουν στα χέρια του παίκτη) βρίσκει την ιδανικότερη τιμή. Επειδή ήταν χρονοβόρο επιλέχθηκαν αυθαίρετες τιμές από τις οποίες θα ξεκινά ο φοίνικας για πολλά φύλλα

Decideandplay:

Ο smartplayer έχει παρόμοια λογική με την νοημοσύνη ενός αληθινού παίκτη. Έτσι, σκανάρει το τραπέζι για πόντους, δεν χτυπάει τον συμπαίκτη του με συνδυασμούς που έχουν μεγάλη διαφορά στο value, και προσπαθεί να ξεφορτωθεί συνδυασμούς με χαμηλή αξία κάθε φορά που κερδίζει ένα TRICK. Ο έξυπνος παίκτης παίζει άλλοτε παθητικά και άλλοτε επιθετικά. Αυτό εξαρτάται από τα φύλλα που υπάρχουν στο τραπέζι. Αν παρατηρήσει την ύπαρξη πολλών πόντων στο τραπέζι ενεργοποιείται το Emergency instinct του και επιτίθεται με τους καλύτερους του συνδυασμούς. Πάντα προσπαθεί να παίξει τους συνδυασμούς που έχει το καλύτερο leaf.

Watch:

Είναι για να γεμίζει το deckPlayed

CalcAggresiveness

Αυτή η συναρτήση δίνει τιμές σε κάποιες global μεταβλητές όπως το emergency, για το τι να κάνει σε μερικές ειδικές περιπτώσεις και το aggression. Όταν ο παίκτης φτάνει στη τελική απόφαση δηλαδή έχει βρει πιο συνδυασμό θέλει να παίξει υπολογίζει (με βάση bestLeaf και αν ταυτίζεται με κάποιο playableCombination) ποια είναι η αξία του leaf χωρίς των συνδυασμό που θέλει να παίξει. Αν μειωθεί πολύ δεν θα παίξει (πχ ώστε να μην παίξει 2 ασσους πανω σε 2 τεσσера). Το πότε θα παίξει έχει σχέση με το aggression του παίκτη (όσο πιο χαμηλό τόσο πιο επιθετικός παίκτης – ναι είναι αναποδα). Αν το newValue είναι μεγαλύτερο από το (παλιο value)*(aggression) τότε παίζει.

Το αν θα κερδίσουμε βασίζεται πολύ στις παραμετρους της αξιολόγηση των leaves, στις ειδικές συνθήκες και στο aggression. Ωστόσο επειδή δεν είχαμε πολύ χρόνο (τη δευτερα πρωι 8/6 ξεκινήσαμε την υλοποίηση του smart player, τις προηγούμενες μέρες φτιάχναμε τις άλλες συναρτήσεις-κλασεις, επρεπε να διαβασουμε πως δουλευουν τα δεντρα σε crr και βλεπαμε αν λειτουργουσαν σωστά το οποίο μας πήρε πολυ καιρό) για να βρούμε τις καλύτερες παραμέτρους η επιλογή κάποιων έγινε αυθαίρετα και αμα κερδιζαμε τον random θεωρούσαμε οτι ήταν καλές. Δεν προλαβαμε να τεστάρουμε πολλές παραμέτρους, να βάλουμε αντίπαλες ομάδες με διαφορετικές παραμέτρους, ενω σε κάποια σημεια δεν είχαμε και τις απαραίτητες γνωσεις (πχ να υπολογισουμε τις πιθανοτητες να παιχτει ενας συνδυασμος πανω στο δικο μας με βαση τα εναπομείναντα φύλλα)

Αποτελέσματα κατόπιν 100 γύρων με rand και ίδιες παραμέτρους:

87 νίκες(60 1-2), 11 ηττες(με πολυ μικρή διαφορά πχ 45-55), 2 ισοπαλίες