

Το πρόγραμμα μεταγλωττίζεται με την εντολή `make` και τρέχει με την εντολή `./game`.

Από τα δωθέντα αρχεία χρησιμοποιήθηκε το `DeckBuilder`, που τροποποιήθηκε, όπου τροποποιήθηκαν οι κλήσεις των `constructors` των καρτών (`Acher`, `Bow`, κλπ.), ώστε να δέχονται εκτός από το όνομα της κάρτας, 2 ints, ο πρώτος για τον γενικό τύπο πράσινης ή μαύρης κάρτας (`t`) (`Follower`, `Item`, `Personality`, `Holding`) και ο δεύτερος για τον ειδικό τύπο (`Attacker`, `Defender`, κλπ) (`spT`).

Επίσης χρησιμοποιούνται τα δωθέντα `enums`, σαν `public` στοιχείο του `Card` για να μπορούν να χρησιμοποιηθούν και στις άλλες κλάσεις σχετικές με κάρτες.

Σχεδιαστικές επιλογές:

- Υλοποιήθηκε η βοηθητική συνάρτηση `initializeGameBoard`, όπου παράγονται τυχαίοι αριθμοί παικτών (2-8) και ελάχιστων και μέγιστων καρτών (ίδιο για όλα τα είδη, μεταξύ τυχαία ορισμένων ορίων) για την παρτίδα. Δημιουργούνται τα γενικά `decks` με τις κάρτες, οι παίκτες, μία λίστα όπου αποθηκεύονται και ένα αντικείμενο `GameBoard` με τα προηγούμενα δεδομένα. Τυπώνονται τα στατιστικά του παιχνιδιού και επιστρέφεται η λίστα των παικτών στη `main`, για να μπορεί να χρησιμοποιηθεί αργότερα.
- Το παιχνίδι συνεχίζεται μέσω της κλάσης `Gameplay`, που περιλαμβάνει την εκτέλεση των 5 φάσεων του παιχνιδιού, μία συνάρτηση που ταξινομεί μια λίστα παικτών (`select sort`) κατά φθίνουσα σειρά `honour` και μία συνάρτηση που ελέγχει τα `winning conditions`.
- Οι παίκτες αντιπροσωπεύονται από μία κλάση `Player`, η οποία περιέχει όλα τα δεδομένα και `decks` που έχει ένας παίκτης. Οι παίκτες αντιπροσωπεύονται από λίστα στο υπόλοιπο πρόγραμμα για ευκολότερη διαχείρισή τους.
- Τα `decks` δημιουργούνται ως εξής: Δημιουργούνται ένα `green` και ένα `black` στην `initializeGameBoard`, τα οποία περιέχουν όλες τις κάρτες του παιχνιδιού. Ανακατεύονται και περνούνται στους παίκτες (ως όρισμα στον `constructor` τους), όπου μοιράζονται στους παίκτες. Γίνεται κάθε φορά `pop`, ώστε να μην μοιράζονται οι ίδιες κάρτες. Στην περίπτωση που
- Τα `decks` αντιπροσωπεύονται από τις κλάσεις `Deck`, `FateDeck` και `DynastyDeck`. Η καθεμία αντιπροσωπεύει μία λίστα από δείκτες σε κάρτες, κάθε είδους, πράσινες και μαύρες αντίστοιχα. Η `Deck` κληρονομείται από τις άλλες δύο και οι σχετικές τους συναρτήσεις υλοποιούνται εκεί.
- Οι κάρτες, δηλαδή οι κλάσεις `Card`, `GreenCard` και `BlackCard`, όλες, οι πράσινες και οι μαύρες αντίστοιχα. Η `Card` κληρονομείται από τις άλλες δύο και οι γενικές τους συναρτήσεις υλοποιούνται εκεί. Οι `constructors` παίρνουν σαν ορίσματα το όνομα και ένα `enum int` για το `type`, σύμφωνα με τις προδιαγραφές της εκφώνησης, καθώς και άλλο ένα για τον συγκεκριμένο τύπο. Η δημιουργία γίνεται από το `DeckBuilder`, ξεκινώντας από τις `subclasses`, οι οποίες καλούν κάθε φορά την `upperclass` τους, μέχρι την `Card`, ώστε να

πάρουν όλες το name και τον int του type. Τα υπόλοιπα δεδομένα μίας κάρτας δεν αρχικοποιούνται στις κατώτερες κλάσεις (Mine, Katana, κλπ).

- Αντίθετα, η GreenCard έχει τις υποκλάσεις Follower και Item, οι οποίες έχουν αντίστοιχα υποκλάσεις τα είδη των καρτών αυτών (πχ. Archer, Bow, κλπ) και η BlackCard τις Personality και Holding, με υποκλάσεις αντίστοιχα των ειδών τους (πχ. Attacker, Mine, κλπ). Οι 4 αυτές κλάσεις – είδη καρτών, δέχονται σαν ορίσματα από τον constructor τους όλα τους τα δεδομένα που δεν είναι σταθερά. Οι υποκλάσεις Bow, Katana, κλπ τους περνούν τα στοιχεία τους.
- Οι Archer, Mine, κλπ δεν αποθηκεύουν κάποιο από τα type. Οι Follower, Item, Personality και Holding, αποθηκεύουν το συγκεκριμένο type της κάρτας, το οποίο αντιστοιχεί στα enums Followers, Items, Personalities και Holdings. Τέλος, οι Card, BlackCard και GreenCard αποθηκεύουν τον γενικό τύπο κάρτας που αντιστοιχεί στο enum Type.
- Τα Stronghold δημιουργούνται μέσω της κλάσης Player, μέσω της συνάρτησης GenerateRandomStronghold, η οποία παίρνει τυχαίες τιμές μεταξύ τυχαίων ορίων για τα πεδία του Stronghold, δημιουργεί το αντικείμενο με αυτές τις τιμές και το επιστρέφει. Για το όνομα δημιουργήθηκε ένα vector με 8 ονόματα (για να υποστηρίζονται 8 παίκτες), επιλέγεται κάθε φορά ένα τυχαίο και διαγράφεται για να μην επιλεγεί ξανά.
- Γενικώς, οι περισσότερες λειτουργίες του προγράμματος γίνονται μέσα στην Gameplay και την Player, από όπου καλούνται και οι συναρτήσεις των άλλων κλάσεων.
- Το παιχνίδι εκτελείται μέχρι να απομείνει μόνο ένας παίκτης ζωντανός και να νικήσει.
- Για περισσότερες λεπτομέρειες σχετικά με την υλοποίηση, υπάρχουν αναλυτικά σχόλια πάνω στον κώδικα.