



## Γενικές Οδηγίες και Περιγραφή των Παραδοτέων της Εργασίας

### Φάσεις

- **Φάση Α: 10% τελικού βαθμού** (ρήτρα 5 για επιτυχία στο μάθημα)
  - ο 21 Νοεμβρίου - 3 Δεκεμβρίου 2022
- **Φάση Β: 15% τελικού βαθμού** (ρήτρα 5 για επιτυχία στο μάθημα)
  - ο 4 Δεκεμβρίου - 23 Δεκεμβρίου 2023 (bonus 5%) ή
  - ο 4 Δεκεμβρίου - 7 Ιανουαρίου 2023 (χωρίς bonus)

**Παρατάσεις μπορεί να δοθούν μόνο αν κριθεί σκόπιμο και το πολύ 3 μέρες για τη Φάση Α και για τη Φάση Β!**

### Φάση Α

Σε αυτή τη φάση πρέπει να γίνει ο σχεδιασμός της εφαρμογής βάσει των ιδεών και των αρχών του αντικειμενοστρεφούς προγραμματισμού που έχετε διδαχθεί. Αποτέλεσμα αυτής της φάσης είναι ο καθορισμός των αντικειμένων, των χαρακτηριστικών και της συμπεριφοράς τους που απαιτούνται για να αναπαραστήσουν τις καταστάσεις και τις λειτουργίες του θέματος της εργασίας, όπως έχουν περιγραφεί στην εκφώνηση.

Παραδοτέα αυτής της φάσης είναι :

- **Γραπτή αναφορά** (όχι σε greeklish) η οποία θα περιγράφει τα παραπάνω στοιχεία και θα παρουσιάζει το σχέδιο υλοποίησης της προγραμματιστικής εργασίας, έτσι ώστε να είναι έτοιμο το πέρασμα στην επόμενη φάση της υλοποίησης. Θα πρέπει να συμπεριλαμβάνονται και **UML class diagrams**.
- **Πηγαίνος κώδικας** που περιλαμβάνει τις **διεπαφές** (interfaces) και το περίγραμμα των κλάσεων (class outline) Java του προγράμματος σας, συνοδευόμενες από τα απαραίτητα **javadoc** σχόλια, τα οποία θα καθοδηγήσουν την υλοποίηση της επόμενης φάσης.

Επιγραμματικά, οι σημαντικότερες εργασίες που πρέπει να γίνουν σε αυτή τη φάση είναι:

- Αναγνώριση των κλάσεων και διεπαφών για κάθε μικρή και μεγάλη συνιστώσα (component) του προγράμματος. Αναγνώριση των ευθυνών κάθε κλάσης και των πιθανών σχέσεων της με άλλες.
- Εύρεση των χαρακτηριστικών και των μεθόδων κάθε κλάσης.
- Εύρεση της συμπεριφοράς (behaviour) κάθε κλάσης και διεπαφής, καθώς και της επικοινωνίας μέσω μηνυμάτων (method calls) που χρειάζεται να έχουν μεταξύ τους.
- Οργάνωση των κλάσεων σε ιεραρχίες με στόχο την μέγιστη δυνατή επαναχρησιμοποίηση του κώδικα σας και τον πολυμορφισμό
- Για κάθε κλάση που υλοποιεί μια διεπαφή δώστε τις υπογραφές (signatures) για όλες τις μεθόδους και τις εκ των προτέρων, εκ των υστέρων και αμετάβλητες συνθήκες (preconditions, postconditions, invariants) που τις διέπουν σε μορφή javadoc σχολίων.

**ΣΗΜΑΝΤΙΚΟ:** Σημειώστε ότι όσο πληρέστερη και αναλυτικότερη δουλειά κάνετε στην σχεδίαση (Φάση Α) τόσο πιο σωστή, εύκολη και επιτυχημένη θα είναι η υλοποίηση (Φάση Β). Επειδή τα προηγούμενα χρόνια παρατηρήσαμε ότι όσοι δεν είχαν κάνει καλή Φάση Α σπάνια έκαναν καλή Φάση Β, τα τελευταία χρόνια υπάρχει η ρήτρα 5 και στη Φάση Α, ελπίζοντας ότι αυτό θα σας βοηθήσει να

**κάνετε καλύτερη κατανομή του χρόνου σας και να μεγιστοποιήσετε τα οφέλη από το μάθημα και τον τελικό βαθμό που θα πετύχετε.**

## UML Plugins

Μπορείτε να συμβουλευτείτε το link που βρίσκεται στην σελίδα του μαθήματος στο section Project : “Πληροφορίες για UML tools/plugins”

### Αναλυτική βαθμολόγηση για τη Α φάση

Παραδοτέα αυτής της φάσης είναι :

| Παραδοτέα   | Μονάδες |
|---|---------|
| Γραπτή αναφορά (όχι σε greeklish) η οποία θα περιγράφει τα παραπάνω στοιχεία και θα παρουσιάζει το σχέδιο υλοποίησης της προγραμματιστικής εργασίας, έτσι ώστε να είναι έτοιμο το πέρασμα στην επόμενη φάση της υλοποίησης.<br>Στο moodle υπάρχει ενδεικτικό template για τη γραπτή αναφορά της εργασίας σας.   | 20      |
| Θα πρέπει να συμπεριλαμβάνονται και UML class diagrams και να εξηγείται τι απεικονίζει το κάθε διάγραμμα.   | 10      |
| Χωρισμός εργασίας σύμφωνα με το μοντέλο MVC (Model View Controller)   | 15      |
| Για κάθε κλάση που υλοποιεί μια διεπαφή δώστε τις υπογραφές (signatures) για όλες τις μεθόδους και τις εκ των προτέρων, εκ των υστέρων και τις αμετάβλητες συνθήκες (pre-conditions, post-conditions και invariants αντίστοιχα) που τις διέπουν σε μορφής javadoc σχολίων.  | 15      |
| Οργάνωση των κλάσεων σε ιεραρχίες με στόχο την μέγιστη δυνατή επαναχρησιμοποίηση των κώδικα. Χρησιμοποίηση abstract κλάσεων ή/και interfaces για τις βασικές κλάσεις Player και Piece.  | 15      |
| Να υπάρχουν οι κλάσεις και οι απαραίτητες μέθοδοι (25%)<br>Board, Player, Controller, View<br>Πρέπει να υπάρχουν οι μέθοδοι για τα βασικά στοιχεία του παιχνιδιού   |         |
| Ενδεικτικά κάποιες βασικές μέθοδοι ( <b>Θα πρέπει να ορίσετε και αρκετές ακόμα</b> ):<br><ul style="list-style-type: none"> <li>• Μέθοδος για έλεγχο αν η κίνηση του πιονιού είναι επιτρεπτή</li> <li>• Μέθοδος για το ποιο πιόνι κερδίζει όταν έχουμε επίθεση</li> <li>• Μέθοδοι που φτιάχνουν τα πιόνια, τους παίκτες κλπ</li> <li>• Μέθοδος που καθορίζει τη σειρά (ποιος παίκτης παίζει)</li> </ul> | 25      |
| <b>Σύνολο</b>   | 100     |

Αναφορές που κάνουν μόνο copy paste τον κώδικα από netbeans/eclipse με τα σχόλια θα παίρνουν χαμηλό βαθμό (μισό βαθμό και λιγότερο ανάλογα την περίπτωση) για να μην αδικούνται οι φοιτητές που έκαναν καλή αναφορά. Μια καλή αναφορά πρέπει να έχει προφανώς τα στοιχεία σας και να είναι οργανωμένη σε ενότητες (καλό είναι να έχει αριθμημένες σελίδες και πίνακα περιεχομένων στην αρχή).

## Φάση Β

Σε αυτή τη φάση πρέπει να γίνει η κυρίως υλοποίηση της εφαρμογής, βάσει της σχεδίασης που έχει προηγηθεί (φάση Α). Δεν επιβάλλεται να χρησιμοποιηθεί αυτούσια η σχεδίαση της φάσης Α, καθώς κάποιες σχεδιαστικές επιλογές αποδεικνύεται στην πορεία ότι χρειάζονται αναθεώρηση. Εντούτοις, η τελική βαθμολογία θα εξαρτηθεί και από τη συνέπεια της τελικής υλοποίησης ως προς την αρχική σχεδίαση.

Σε αυτή τη φάση, παραδοτέα είναι :

- ο **πηγαίος κώδικας** που υλοποιεί την εργασία
- αναλυτικές οδηγίες για το πώς μεταγλωττίζεται και πώς τρέχει το πρόγραμμά σας (README, Ant, Maven κλπ)
- αναφορά, στην οποία θα αναλύετε :
  - ο η τελική σχεδίαση της εφαρμογής,
  - ο ποιες αλλαγές έγιναν σε σχέση με τη σχεδίαση της Α' φάσης (και γιατί),
  - οι αλγόριθμοι που χρησιμοποιήθηκαν
  - τυχόν διαφοροποιήσεις στους κανόνες σε σχέση με τους κανόνες που δίνονται παραπάνω
  - οι σχεδιαστικές και προγραμματιστικές αποφάσεις που ελήφθησαν και πώς αυτό αντανακλάται στον τελικό χρήστη (π.χ. ευκολία/δυσκολία χειρισμού)
  - τα προβλήματα που αντιμετωπίστηκαν
  - τα JUnit tests που φτιάχτηκαν για τον έλεγχο της ορθότητας
  - γενικά ότι άλλο κρίνετε απαραίτητο να αναφερθεί (όπως για παράδειγμα τι ενδεχομένως δεν κάνατε).

## Βαθμολογία Εργασίας

Για τη βαθμολογία της εργασίας σας θα συνεκτιμηθούν:

- εάν (και πόσο) η σχεδίαση της εφαρμογής εφαρμόζει τις έννοιες και τεχνικές του αντικειμενοστρεφούς προγραμματισμού που διδαχθήκατε στο μάθημα
- εάν (και πόσο) υλοποιήθηκαν οι υποχρεωτικές λειτουργίες της εφαρμογής
- η πληρότητα της τελικής αναφοράς, η οποία θα καταγράφει και θα τεκμηριώνει την σχεδίαση και υλοποίηση της εφαρμογής

Ο τρόπος βαθμολόγησης περιγράφεται αναλυτικά στην επόμενη σελίδα.

Για διευκρινίσεις σχετικά με την παραπάνω εργασία μπορείτε να στέλνετε μηνύματα με απορίες σας στο σχετικό **forum** στην ιστοσελίδα του moodle. Ερωτήσεις που στέλνονται στην λίστα του μαθήματος [hy252-list@csd.uoc.gr](mailto:hy252-list@csd.uoc.gr) **δε θα απαντώνται**.

## Αναλυτική βαθμολόγηση για τη Β φάση

Η αναλυτική βαθμολόγηση για τη Β φάση του Project θα είναι η εξής:

| Παραδοτέα (20μ)   | Μονάδες    |
|---|------------|
| Αναφορά (Ανανεωμένη από την Φάση Α)   | 8          |
| Σχόλια Javadoc (Pre – Post Conditions)  | 4          |
| JUnit Tests   | 4          |
| UML Diagrams με βάση την τελική έκδοση και επεξήγηση τους   | 4          |
| Λειτουργικότητα-Γραφικά (80μ +5μ bonus)   |            |
| <ul style="list-style-type: none"><li>Τήρηση σειράς και τοποθέτηση πιονιών (10%)</li><li>Σωστή Κίνηση πιονιών και ανιχνευτή (12%)</li><li>Επίθεση πιονιών (10%)</li><li>Επίθεση εξολοθρευτή-δράκου (7%)</li><li>Εξουδετέρωση παγίδας από νάνο (5%)</li><li>Διάσωση πιονιού που βγήκε από το παιχνίδι (10%)</li><li>Τέλος παιχνιδιού-κατάκτηση σημαίας (6%)</li><li>Μενού πληροφοριών στο πλάι (12%)</li><li>Game mode: Μειωμένος Στρατός (8%)</li><li>Game mode: Καμία Υποχώρηση (Bonus 5%)</li></ul> | 80         |
| <b>Σύνολο</b>   | <b>100</b> |

**ΣΗΜΑΝΤΙΚΟ:** Η **ελάχιστη** υλοποίηση που χρειάζεται κάποιος για να πάρει βαθμό 45/100 είναι να υπάρχει γραφικό περιβάλλον, τοποθέτηση των πιονιών στο ταμπλό και να κινούνται τα πιόνια στο ταμπλό. Τα παραπάνω προϋποθέτουν πως υπάρχει επικοινωνία της γραφικής διεπαφής με το μοντέλο μέσω του controller.

**Καλή Εργασία**



# Project Stratego

## Εκπαιδευτικοί Στόχοι

Προδιαγραφή και σχεδίαση συστήματος

Προδιαγραφή Αφαιρετικών Τύπων Δεδομένων (ΑΤΔ) που απαιτούνται για την επιτυχή ολοκλήρωσή του συστήματος

Υλοποίηση εξαρτημάτων (ΑΤΔ) του συστήματος των οποίων η προδιαγραφή δίνεται

Χρήση κληρονομικότητας και πολυμορφισμού

Δημιουργία Γραφικής Διεπαφής

Απεξάρτηση του πυρήνα του συστήματος από τη Γραφική Διεπαφή

Επαναχρησιμοποίηση διεπαφών και κλάσεων

Χρήση JFC (Java Collection Framework)

Τεκμηρίωση, Έλεγχος

## Σύντομη Περιγραφή Εργασίας

**Στόχος.** Στην εργασία αυτή καλείστε να σχεδιάσετε και να υλοποιήσετε μία παραλλαγή του κλασικού παιχνιδιού Stratego με τίτλο Stratego Ice vs Fire.

Η εργασία είναι ατομική και απαγορεύεται η χρήση κώδικα που δεν έχετε γράψει οι ίδιοι. Σε περίπτωση εντοπισμού αντιγραφής η εργασία θα μηδενίζεται.

Αν κάποιος δεν γνωρίζει αυτό το παιχνίδι, καλό είναι να δει το παρακάτω video:

<https://www.youtube.com/watch?v=3R7d0A9ymwQ>.

(πατήστε το CC αν θέλετε για υπότιτλους)

## Περιγραφή και Σκοπός του Παιχνιδιού.

Οι δύο παίκτες έχουν ένα στρατό που αποτελείται από 30 πιόνια, συμπεριλαμβανομένης της σημαίας. Ο σκοπός του παιχνιδιού είναι ο παίχτης να αιχμαλωτίσει τη σημαία του αντιπάλου του και να προστατεύσει τη δικιά του. Ο πρώτος από τους 2 παίχτες που τα καταφέρνει ανακηρύσσεται νικητής του παιχνιδιού. Ο κάθε παίκτης τοποθετεί τα πιόνια του στις τρεις πρώτες σειρές που έχει μπροστά του, ενώ κάθε πιόνι έχει ένα όνομα και ένα βαθμό κατάταξης. Όσο μεγαλύτερος είναι ο βαθμός τόσο ισχυρότερο είναι το πιόνι. Αν ένα πιόνι υψηλότερου βαθμού επιτεθεί σε ένα πιόνι χαμηλότερου βαθμού, το πιόνι με τον υψηλότερο βαθμό κερδίζει και το άλλο βγαίνει από το παιχνίδι. Η Σημαία μπορεί να κατακτηθεί από οποιοδήποτε πιόνι του αντιπάλου μπορεί να μετακινηθεί.

## Αναλυτική Περιγραφή της Εργασίας

### Περιεχόμενα παιχνιδιού

Τα παιχνίδι αποτελείται από :

- 1 ταμπλό με διαστάσεις 8x10 (8 σειρές-10 στήλες)
- 30 κόκκινα πιόνια (στρατός Volcandria)
- 30 μπλε πιόνια (στρατός Everwinter)

## Προετοιμασία του παιχνιδιού

Κάθε ένας από τους 2 παίκτες τοποθετεί τα 30 πιόνια<sup>1</sup> του στις 3 πρώτες σειρές που βρίσκονται μπροστά του. Κάθε πιόνι μπορεί να τοποθετηθεί σε οποιαδήποτε θέση από αυτές τις 30 όπως φαίνεται και στην Εικόνα 1. Για λόγους απλότητας και ταχύτητας, μπορείτε να τοποθετείτε random τα πιόνια, έτσι ώστε να μπορείτε να ελέγχεται πιο γρήγορα τους κανόνες και να μην καθυστερείτε με την τοποθέτηση τους.

Στο ταμπλό υπάρχουν 8 τετράγωνα (τα κίτρινα) που ονομάζονται απαγορευμένες ζώνες. Δε μπορεί κάποιο πιόνι να παραμείνει σε αυτά τα τετράγωνα.



Εικόνα 1

## Πιόνια του Παιχνιδιού

Τα συνολικά πιόνια για κάθε παίχτη είναι 30, ενώ τα μοναδικά πιόνια είναι 12, μιας και τα περισσότερα εμφανίζονται πάνω από 1 φορά:

| Όνομασία               | Κατάταξη | Διαθεσιμότητα |
|------------------------|----------|---------------|
| Δράκος (Dragon)        | 10       | 1             |
| Μάγος (Mage)           | 9        | 1             |
| Ιππότης (Knight)       | 8        | 2             |
| Αναβάτης (Beast Rider) | 7        | 3             |
| Μάγισσα (Sorceress)    | 6        | 2             |

<sup>1</sup> Στη παραλλαγή αυτή τα πιόνια είναι 30 αντί για 40.

|  |       |   |
|--|-------|---|
| Θηρίο Λάβας (Lava Beast) (κόκκινο) / Yeti (μπλε) | 5     | 2 |
| Ξωτικό (Elf)                                     | 4     | 2 |
| Νάνος (Dwarf)                                    | 3     | 5 |
| Ανιχνευτής (Scout)                               | 2     | 4 |
| Εξολοθρευτής (Slayer)                            | 1 (S) | 1 |
| Παγίδα   | -     | 6 |
| Σημαία   | -     | 1 |

## Κλασικοί κανόνες Stratego (80 %)

Στο κλασικό παιχνίδι, τα πιόνια με το μεγαλύτερο βαθμό μπορούν να αιχμαλωτίσουν τα πιόνια με το μικρότερο βαθμό. Τα μοναδικά πιόνια που έχουν κάποιες ειδικές δυνάμεις όσον αφορά είτε την κίνηση τους, είτε την επίθεση τους είναι οι **ανιχνευτές**, οι **νάνοι** και οι **εξολοθρευτές**. Παρόλο που χρησιμοποιούνται οι κλασικοί κανόνες, εσείς θα χρησιμοποιήσετε 30 πιόνια για κάθε παίχτη και το μέγεθος του ταμπλό θα είναι 8 γραμμές x 10 στήλες όπως στην Εικόνα 1.

Οι οδηγίες στα ελληνικά για το κλασικό παιχνίδι βρίσκονται στο <https://patrasstratego.gr/stratego/>.

## Ενέργειες που γίνονται στη σειρά κάθε παίχτη

Στη σειρά του κάθε παίκτης πρέπει:

- Είτε να κινήσει ένα από τα πιόνια του
- Είτε να κάνει επίθεση στον αντίπαλο.
- Ο ανιχνευτής είναι το μόνο πιόνι που μπορεί να κάνει κίνηση και επίθεση στον ίδιο γύρο
- Σε περίπτωση που ένας παίκτης δε μπορεί να κάνει κάποια κίνηση ή επίθεση, το παιχνίδι τελειώνει και ανακηρύσσεται νικητής ο αντίπαλος

Επίσης κάθε φορά που παίζει ένας παίκτης θα πρέπει να φαίνεται η πίσω όψη της κάρτας του αντιπάλου όπως στις παρακάτω εικόνες. Στην μία περίπτωση παίζει ο παίκτης με τα μπλε και στην άλλη ο παίκτης με τα κόκκινα. Και τους δύο παίκτες τους χειρίζεται ο χρήστης (δεν ζητείται να παίζει ο χρήστης εναντίον του υπολογιστή).



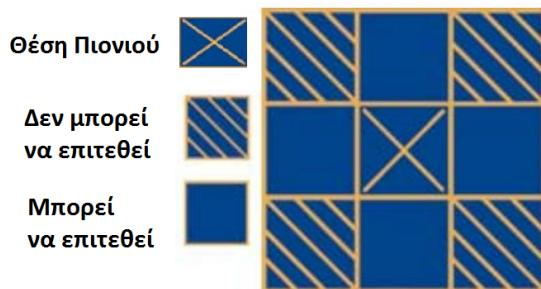
## 1 Κανόνες Κίνησης

Όλα τα πιόνια μπορούν να μετακινηθούν, εκτός από τη σημαία και τις παγίδες, τα οποία μένουν στην ίδια θέση που τοποθετούνται από την αρχή του παιχνιδιού. Τα υπόλοιπα πιόνια κινούνται σύμφωνα με τους παρακάτω κανόνες:

- Σε κάθε γύρο μόνο ένα πιόνι μπορεί να μετακινηθεί
- Κάθε πιόνι μετακινείται ένα τετράγωνο κάθε φορά
  - Τα πιόνια μετακινούνται μπροστά, πίσω, δεξιά ή αριστερά αλλά όχι διαγώνια
    - Εξαιρείται ο ανιχνευτής ο οποίος μπορεί να μετακινηθεί όσες κενές θέσεις θέλει μπροστά πίσω δεξιά ή αριστερά. Απλώς δε μπορεί να περάσει από τις απαγορευμένες ζώνες και να μετακινηθεί διαγώνια
- Δε γίνεται 2 πιόνια να βρίσκονται στην ίδια θέση.
- Δε γίνεται ένα πιόνι να περάσει από τις απαγορευμένες ζώνες

## 2 Κανόνες Επίθεσης

Όλα τα πιόνια μπορούν να κάνουν επίθεση, εκτός από τη Σημαία η την Παγίδα. Ένα πιόνι μπορεί να επιτεθεί σε οποιοδήποτε πιόνι αντιπάλου που βρίσκεται 1 τετράγωνο μπροστά, πίσω, δεξιά ή αριστερά όπως δείχνει η παρακάτω εικόνα



Όταν ένα πιόνι επιτίθεται, τότε το πιόνι με το χαμηλότερο βαθμό (κατάταξη) από τα 2 αφαιρείται από το ταμπλό.

- Αν το πιόνι που αφαιρείται είναι του αμυνόμενου, τότε το πιόνι του επιτιθέμενου παίρνει τη θέση του.
- Αν το πιόνι που αφαιρείται είναι του επιτιθέμενου, τότε το πιόνι του αμυνόμενου μένει στην ίδια θέση
- Αν τα πιόνια έχουν τον ίδιο βαθμό, τότε αφαιρούνται και τα δύο

Ο Ανιχνευτής μπορεί να επιτεθεί από όποια απόσταση θέλει, αρκεί τα τετράγωνα μεταξύ αυτού και του αμυνόμενου πιονιού να είναι ελεύθερα (άδεια).

## 3 Κατάταξη κάθε πιονιού για την επίθεση

Οι κανόνες που ισχύουν για την επίθεση είναι οι εξής:

- Ο δράκος (10) κερδίζει το Μάγο (9) και όλα τα χαμηλότερα σε βαθμό πιόνια. Ο Μάγος (9) κερδίζει τον Ιππότη(8) και όλα τα χαμηλότερα σε βαθμό πιόνια. Αυτός ο κανόνας συνεχίζεται μέχρι το τελευταίο πιόνι.
- Οποιοδήποτε πιόνι επιτεθεί στην παγίδα χάνει αυτόματα, εκτός από τον νάνο (3), που εξουδετερώνει την παγίδα και την βγάζει εκτός παιχνιδιού, ενώ παίρνει και τη θέση που είχε η παγίδα αυτή.
- Η μικρότερη αξία ανήκει στο πιόνι εξολοθρευτή (1 ή s). Οποιοδήποτε πιόνι επιτεθεί σε αυτό κερδίζει. Όμως αυτό το πιόνι έχει μια εξαιρετική δυνατότητα. Αν και μόνο αν επιτεθεί πρώτο στο Δράκο (10), τον βγάζει εκτός παιχνιδιού και παίρνει τη θέση του στο ταμπλό.  
(ΠΡΟΣΟΧΗ: Αν επιτεθεί πρώτος ο Δράκος στον εξολοθρευτή τότε υπερισχύει ο Δράκος)
- Οποιοδήποτε πιόνι μπορεί να επιτεθεί και να κατακτήσει (κερδίσει) τη Σημαία.

## 4 Διάσωση πιονιού που έχει βγει εκτός παιχνιδιού

Αν ένα πιόνι φτάσει στην πρώτη σειρά του αντιπάλου, τότε μπορεί να επαναφέρει στο παιχνίδι ένα πιόνι από τα δικά του που έχουν βγει εκτός παιχνιδιού και να το τοποθετήσει σε όποια θέση θέλει από τις 30 που ανήκουν στο στρατό του (3 πρώτες σειρές του παίκτη).

Περιορισμοί

- Οι ανιχνευτές δε μπορούν να κάνουν διάσωση
- Δε μπορεί να επαναφερθεί στο παιχνίδι μία παγίδα
- Κάθε παίκτης μπορεί να κάνει μέχρι δύο διασώσεις.
- Δε γίνεται το ίδιο πιόνι να κάνει δύο διασώσεις.

## 5 Τέλος του παιχνιδιού

Νικητής του παιχνιδιού είναι ο παίκτης που θα καταφέρει να κατακτήσει τη Σημαία του αντιπάλου του. Νίκη επιτυγχάνεται επίσης και όταν ο αντίπαλος παίκτης είναι ανίκανος να κινηθεί επειδή είτε του έχουν απομείνει μόνο η Σημαία και οι Παγίδες ή επειδή οι Παγίδες εμποδίζουν την κίνηση των τελευταίων πιονιών που μπορούν να μετακινηθούν.

## Εναλλακτικοί Κανόνες και Μενού Πληροφοριών (20% +5% bonus)

Πριν την αρχή κάθε παιχνιδιού θα εμφανίζεται ένα pop up menu στο οποίο ο παίκτης θα επιλέγει αν θα ισχύουν εναλλακτικοί κανόνες stratego.

Οι εναλλακτικοί που πρέπει να υλοποιηθούν είναι οι εξής:

1. Μειωμένος στρατός (8%): Με αυτήν τη επιλογή το παιχνίδι ξεκινάει με τα πιόνια να είναι μειωμένα κατά το μισό. Πιόνια που υπήρχαν 1 φορά παραμένουν όπως είναι. Για πιόνια με μονό αριθμό γίνεται στρογγυλοποίηση προς τα κάτω πχ. 5 Νάνοι -> 2 Νάνοι.
2. Καμία Υποχώρηση (5% - Bonus): Τα πιόνια δεν μπορούν ποτέ να κινηθούν προς τα πίσω παρα μόνο για επίθεση

**Προσοχή:** Οι κανόνες είναι ανεξαρτητοί μεταξύ τους οπότε το παιχνίδι πρέπει να παίζει κανονικά με οποιοδήποτε συνδυασμό αυτών , πχ (Καμία Υποχώρηση & Μειωμένος Στρατός) κτλ.

## Μενού Πληροφοριών (12%)

Δίπλα από το ταμπλο θα υπάρχει ένα μενού όπως φαίνεται το design στη φωτογραφία :

The screenshot shows a 10x10 grid board with various cards representing different units and traps. A green highlight box surrounds the 4x4 central area of the board. A card labeled "ELF" with a value of "4" is placed in the center of this highlighted area. To the right of the board are three main sections: "Ενεργοί Κανόνες" (Active Rules) with checkboxes for "Μειωμένος στρατός" (Reduced Army) and "Καμία Υποχώριση" (No Retreat); "Στατιστικά" (Statistics) showing "Player 2 turn", "Ποσοστό επιτ. επίθεσης: 78%", "Διασώσεις: 5", and "Γύρος: 9"; and "Αιχμαλωτίσεις" (Captures) showing a 3x3 grid of icons representing captured units.

Στο μενού αυτό θα φαίνεται στατιστικά του παιχνιδιού και οι κανόνες που έχουν επιλεγεί στην αρχή. Πιο συγκεκριμένα:

- Ενεργοί Κανόνες:** ένα tick box το οποίο θα είναι “ticked” αν ο κανόνας έχει επιλεγεί.
- Στατιστικά:**
  - Ποσοστό επιτυχίας επίθεσης: επιτυχημένες επιθέσεις / συνολικές επιθέσεις.
  - Διασώσεις (πόσες φορές έχει γίνει διάσωση από τον παίκτη).
  - Γύρος (αριθμός γύρου)
- Αιχμαλωτίσεις:** θα υπάρχει η εικόνα του κάθε τύπου πιονιού και με έναν αριθμό θα συμβολίζεται πόσες φορές ο παίκτης έχει αιχμαλωτίσει ένα τέτοιο πιόνι.
  - επίσης θα φαίνονται και οι συνολικές αιχμαλωτίσεις.

**Προσοχή:** τα στατιστικά και οι αιχμαλωτίσεις αλλάζουν κάθε γύρο εφόσον αφορούν άλλο παίκτη. (εκτός τον αριθμό γύρων)

## Πίνακας με τις δυνάμεις κάθε πιονιού

| Όνομασία                              | Δυνάμεις που υποστηρίζει                            | Πιόνια που κερδίζει                                 | Πιόνια που χάνει                                  |
|---------------------------------------|---|---|---|
| 10: Δράκος (Dragon)                   |   | 1(αν επιτεθεί πρώτος)<br>2,3,4,5,6,7,8,9,<br>Σημαία | Παγίδα,<br>1 (αν είναι σε άμυνα)                  |
| 9: Μάγος (Mage)                       |   | 1,2,3,4,5,6,7,8,<br>Σημαία                          | Παγίδα, 10  |
| 8: Ιππότης (Knight)                   |   | 1,2,3,4,5,6,7, Σημαία                               | Παγίδα, 10,9                                      |
| 7: Αναβάτης (Beast Rider)             |   | 1,2,3,4,5,6, Σημαία                                 | Παγίδα, 10,9,8                                    |
| 6: Μάγισσα (Sorceress)                |   | 1,2,3,4,5, Σημαία                                   | Παγίδα, 10,9,8,7                                  |
| 5: Lava Beast (Κόκκινο) / Yeti (μπλε) |   | 1,2,3,4, Σημαία                                     | Παγίδα, 10,9,8,7,6                                |
| 4: Ξωτικό (Elf)                       |   | 1,2,3, Σημαία                                       | Παγίδα, 10,9,8,7,6,5                              |
| 3: Νάνος (Dwarf)                      | Εξουδετέρωση<br>Παγίδας                             | 1,2, Παγίδα, Σημαία                                 | 10,9,8,7,6,5,4                                    |
| 2: Ανιχνευτής (Scout)                 | Μπορεί να μετακινηθεί παραπάνω από 1 ελεύθερα κελιά | 1, Σημαία   | Παγίδα, 10,9,8,7,6,5,4,3                          |
| 1: Εξολοθρευτής (Slayer)              | Μπορεί να εξουδετερώσει το Δράκο                    | 10 (αν επιτεθεί πρώτος), Σημαία                     | 10 (αν είναι σε άμυνα)<br>9,8,7,6,5,4,3,2, Παγίδα |
| Παγίδα                                | -   | 1-2, 4-10   | 3   |
| Σημαία                                | -   | -   | 1-10  |

### Υποδείξεις για Σχεδίαση/Υλοποίηση

Κατά τη διάρκεια της σχεδίασης του παιχνιδιού (και εν γένει οποιουδήποτε λογισμικού ή τεχνικού έργου), επιδιώκεται η αποσύνθεση του συστήματος σε μικρότερα τμήματα, με στόχο την ανάθεση σαφώς ορισμένων και καθορισμένων αρμοδιοτήτων σε κάθε τμήμα και την επικύρωση ότι όλα τα τμήματα μαζί επιτυγχάνουν τους σκοπούς του συστήματος. Επομένως, η σχεδίαση είναι μια διαδικασία επίλυσης και κατακερματισμού του αρχικού προβλήματος σε επιμέρους μικρότερα και ευκολότερα επιλύσιμα υπο-προβλήματα που θα ικανοποιούν τις λειτουργικές απαιτήσεις και θα υπόκεινται σε συγκεκριμένες αρχές καλής σχεδίασης.

Μια τέτοια αρχή αποτελεί η αποσύνδεση του μοντέλου (model), που περιγράφει τα δεδομένα, τη συμπεριφορά τους και το σύνολο των κανόνων που τα διέπει, από την απεικόνιση τους (view). Βασικός στόχος μιας τέτοιας αποσύνδεσμένης προσέγγισης είναι η ελαχιστοποίηση των απαιτούμενων επεμβάσεων σε κώδικα που μπορούν να έχουν μελλοντικές αλλαγές είτε στο μοντέλο είτε στο τρόπο/μέσο απεικόνισης. Αυτό οδηγεί σε καλύτερη ποιότητα κώδικα, με μικρότερο κόστος συντήρησης, επέκτασης και επαναχρησιμοποίησης. Για να γίνει πιο κατανοητό αυτό θεωρήστε το ακόλουθο σενάριο. Υποθέστε ότι αρχικά έχετε σχεδιάσει μια εφαρμογή που για διάφορούς λόγους εκτυπώνει τα αποτελέσματα της στην κονσόλα. Έχοντας ακολουθήσει μια τέτοια αρχιτεκτονική 'χαλαρής' σύνδεσης η μετάβαση σε ένα γραφικό παραθυρικό περιβάλλον γίνεται ομαλά, απλώς τροποποιώντας κατάλληλα (επεκτείνοντας) τις διαδικασίες εκείνες που ήταν υπεύθυνες για την αποτύπωση του μοντέλου στην κονσόλα.

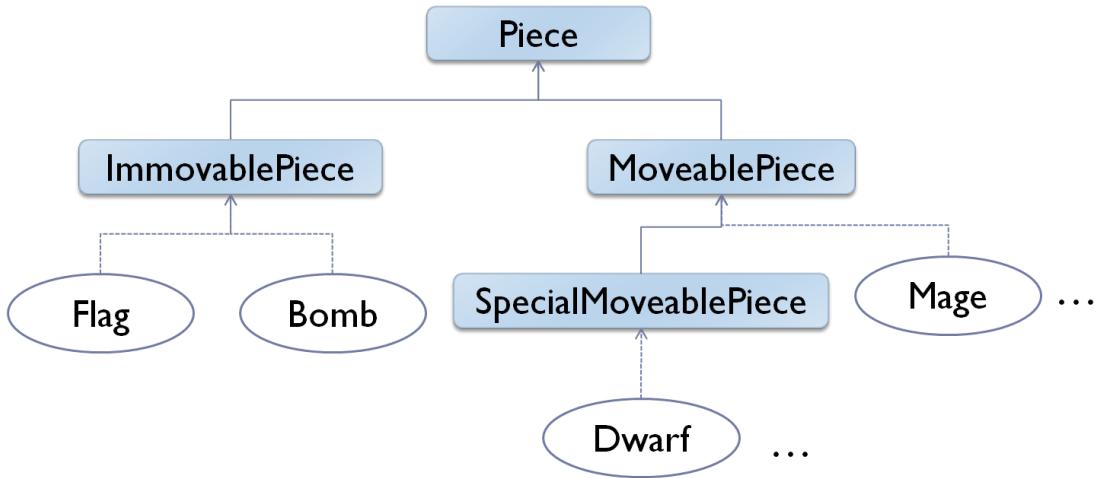
Στη συνέχεια ακολουθεί μια προτεινόμενη σχεδίαση της εφαρμογής, την οποία μπορείτε να ακολουθήσετε και να επεκτείνετε κατάλληλα. Η συγκεκριμένη στρατηγική σχεδίασης δεν είναι μοναδική, καθώς ένα αντικειμενοστραφές σύστημα λογισμικού μπορεί προφανώς να δομηθεί με πάρα πολλούς τρόπους. Αρκεί να αναλογιστεί κανείς τον αριθμό των δυνατών κλάσεων, των λειτουργιών που μπορούν να περιλαμβάνουν και των δυνατών συσχετίσεων μεταξύ τους. Συνεπώς, είστε ελεύθεροι να προτείνετε τη δική σας σχεδίαση εφόσον ακολουθεί τις αρχές που περιγράψαμε και είναι κατάλληλα τεκμηριωμένη.

### MVC (Model-View-Controller) pattern

Βασική αρχή στην ανάπτυξη της παρούσας εργασίας θα πρέπει να είναι το MVC (Model-View-Controller) pattern. Σύμφωνα με αυτό το πρότυπο θα πρέπει να διαχωρίζεται η ανάπτυξη της γραφικής διεπαφής του παιχνιδιού (View) από τον πυρήνα του παιχνιδιού που περιέχει όλη την πληροφορία κατάστασης (Model) και από τον μηχανισμό διαχείρισης και ενημέρωσης των ενεργειών του παιχνιδιού με τη γραφική του απεικόνιση (Controller). Πιο συγκεκριμένα το Model μπορεί να θεωρηθεί ότι αποτελείται από οτιδήποτε σχετίζεται με τα δεδομένα του παιχνιδιού. Υπό αυτήν την έννοια τα «Πιόνια», το «ταμπλό», οι «Παίχτες» αποτελούν μέρος του μοντέλου του παιχνιδιού καθώς περιγράφουν τα εκάστοτε δεδομένα που καλείται να διαχειρίσεται ο Controller του παιχνιδιού. Ο Controller επιφορτίζεται με τη διαχείριση της αλληλεπίδρασης της γραφικής διεπαφής με το μοντέλο. Το MVC pattern σας δίνει την δυνατότητα να διαχωρίσετε την υλοποίηση των παραπάνω και κατά συνέπεια καταστούν το πρόγραμμα σας ευκολότερο και στην ανάπτυξη και στην αποσφαλμάτωση (το κάθε συστατικό μπορεί να αναπτυχθεί και να δοκιμαστεί ξεχωριστά).

### Προτεινόμενες Κλάσεις – Model

Μια βασική οντότητα/κλάση του παιχνιδιού είναι η abstract κλάση Piece. Η κλάση αυτή μοντελοποιεί ένα πιόνι του παιχνιδιού και ορίζει τα γνωρίσματα της (μεταβλητές στιγμιότυπων, π.χ. θέση πιονιού στο ταμπλό, βαθμός κατάταξης) και τις χρήσιμες μεθόδους της (πχ. μέθοδος attack). Μία πιθανή ιεραρχία είναι η ακόλουθη:



Για παράδειγμα η σημαία και η βόμβα δε μπορούν να μετακινηθούν και να επιτεθούν οπότε θα μπορούσαν να **είναι στιγμιότυπα** της κλάσης Immovable Piece. Όλα τα υπόλοιπα που μπορούν να μετακινηθούν ανήκουν στην κατηγορία Moveable Piece που θα έχει μία μέθοδο που θα μπορούσε να επιστρέψει τις πιθανές θέσεις που μπορεί να πάει ένα πιόνι.

Θα μπορούσατε επίσης να σχεδιάσετε την λειτουργικότητα μιας κάρτας σαν διεπαφή (interface) και όχι απλώς σαν κλάση. Σε αυτή την περίπτωση, όλα τα διαφορετικά είδη καρτών θα πρέπει να υπακούν και να υλοποιούν το συμβόλαιο που έχει ορίσει η διεπαφή. Σε κάθε περίπτωση όμως θα πρέπει να συμπεριλάβετε στη σχεδίαση σας την τεχνική του πολυμορφισμού (π.χ. τη μέθοδο match). Η πολυμορφική αυτή μέθοδος είναι κοινή για όλες τις κλάσεις της ιεραρχίας, αλλά έχει πολλές διαφορετικές μορφές και υλοποιήσεις. Αρκεί να σκεφτούμε ότι η υλοποίηση της μεθόδου match διαφέρει από κάρτα σε κάρτα.

Μια σημαντική οντότητα του παιχνιδιού είναι η **Board**, η οποία προσομοιώνει το ταμπλό του παιχνιδιού. Το ταμπλό του παιχνιδιού αποτελείται από 80 τετράγωνα, όπου 8 από αυτά είναι απαγορευμένες ζώνες.

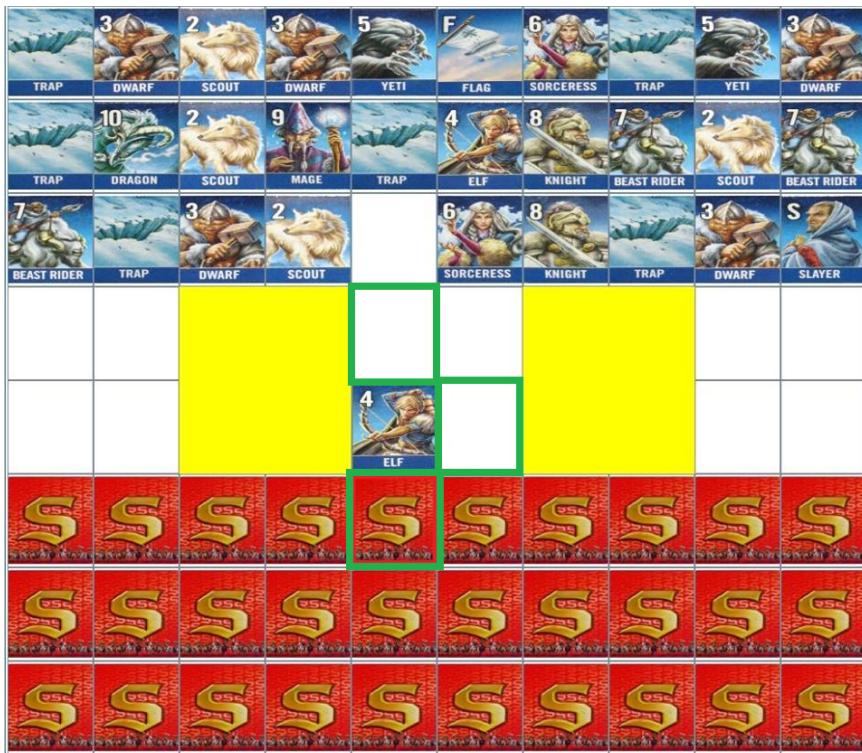
Μια από τις πιο στοιχειώδεις ίσως οντότητες του παιχνιδιού είναι αυτή του **Player**. Όπως υποδηλώνει και το όνομα της, η κλάση αυτή προσομοιώνει τον παίκτη του παιχνιδιού. Κάθε παίκτης διατηρεί ένα σύνολο από ενεργά πιόνια που βρίσκονται στο ταμπλό.

## Controller

Ο Controller έχει πρόσβαση στα δεδομένα του παιχνιδιού και στην διεπαφή. Από την διεπαφή λαμβάνει πληροφορία για τις κινήσεις των παιχτών και με αυτήν θα πρέπει να ανανεώσει το μοντέλο του παιχνιδιού. Αντίστοιχα η ανανέωση του μοντέλου του παιχνιδιού θα πρέπει να χρησιμοποιείται από τον Controller για την ανανέωση της διεπαφής (ως αποτέλεσμα της κίνησης του παίχτη). Η κλάση αυτή αποτελεί το «εγκέφαλο» του παιχνιδιού, καθώς ενορχηστρώνει όλες τις δυνατές λειτουργίες του και παρέχει προς τα έξω τις απαραίτητες μεθόδους ώστε να επικοινωνεί ο πυρήνας του παιχνιδιού (το σύνολο των κανόνων) με τη γραφική του αναπαράσταση. Συγκεκριμένα, ελέγχει για την αρχικοποίηση του, για την ομαλή διεξαγωγή του, την σειρά που θα παίζουν οι παίκτες, την ορθότητα των κινήσεων, τον τερματισμό του παιχνιδιού και την ανάδειξη του νικητή. Σκεφτείτε αναλυτικά τη ροή του παιχνιδιού και συμπεριλάβετε στη σχεδίαση σας όλες τις απαραίτητες λειτουργίες της κλάσης.

## Γραφική Απεικόνιση

Η γραφική διεπαφή του παιχνιδιού πρέπει να παρουσιάζει το ταμπλό και να δίνει την δυνατότητα στους παίχτες να παίξουν ταυτόχρονα το παιχνίδι σε έναν υπολογιστή. Ένα σχεδιαστικό παράδειγμα υλοποίησης της εν λόγω διεπαφής παρουσιάζεται στην Εικόνα 2. Σε αυτήν την εικόνα κάθε πλευρά της διεπαφής αντιστοιχεί στα πιόνια ενός παίκτη ενώ όταν πατάει κάποιος ένα πιόνι, φαίνονται οι πιθανές θέσεις που μπορεί είτε να πάει είτε να επιτεθεί. Η γραφική απεικόνιση αυτή είναι ενδεικτική. Οποιαδήποτε διαφοροποίηση που δεν μειώνει την λειτουργικότητα του παιχνιδιού είναι αποδεκτή. Για παράδειγμα κάποιος θα μπορούσε να προσθέσει οτιδήποτε του αρέσει που να κάνει το παιχνίδι πιο ευχάριστο όπως πληροφορία για το ποια πιόνια έχουν φύγει για κάθε παίχτη, ονόματα παιχτών, ήχους κλπ.



Εικόνα 2

### Προτεινόμενες Κλάσεις – View

Τέλος το View είναι η γραφική διεπαφή με την οποία αλληλεπιδρούν οι τελικοί χρήστες του παιχνιδιού. Όσον αφορά την διεπαφή η χρήση κληρονομικότητας μπορεί να διευκολύνει την ανάπτυξη μέσω της δημιουργίας νέων τύπων UI συστατικών τα οποία επεκτείνουν τη λειτουργικότητα των υπαρχόντων (για παράδειγμα η δημιουργία ενός τύπου PieceButton μπορεί να σας βοηθήσει στον χειρισμό των πιονιών ως Buttons αλλά με τα χαρακτηριστικά που εσείς επιθυμείτε να έχει ένα «Πιόνι» του παιχνιδιού). Μπορείτε να χρησιμοποιήσετε Grid Layout το μιας και το ταμπλό του παιχνιδιού ουσιαστικά είναι ένα grid διαστάσεων 8x10.

### Βοηθητικά

Παρακάτω σας δίνονται κάποια ενδεικτικά links τα οποία μπορούν να σας βοηθήσουν στην υλοποίηση:

### Επίσημες Οδηγίες στα Αγγλικά:

<http://www.hasbro.com/common/instruct/45068.pdf>

### Σύντομο επεξηγηματικό υλικό για το παιχνίδι μπορεί να βρεθεί εδώ:

<https://www.youtube.com/watch?v=3R7d0A9ymwQ>

**Ενδεικτικό βοήθημα για το MVC μοντέλο σε Java**

<http://www.newthinktank.com/2013/02/mvc-java-tutorial/>

Οι φωτογραφίες των καρτών έχουν ανέβει στο Elearn!

***Καλή Εργασία***