



Γενικές Οδηγίες και Περιγραφή των Παραδοτέων της Εργασίας

Φάσεις

- **Φάση A: 10% τελικού βαθμού** (ρήτρα 5 για επιτυχία στο μάθημα)
 - 16 Νοεμβρίου - 4 Δεκεμβρίου 2020
- **Φάση B: 15% τελικού βαθμού** (ρήτρα 5 για επιτυχία στο μάθημα)
 - 5 Δεκεμβρίου - 27 Δεκεμβρίου 2020 (bonus 5%) ή
 - 28 Δεκεμβρίου - 8 Ιανουαρίου 2021 (χωρίς bonus)

Παρατάσεις μπορεί να δοθούν μόνο αν κριθεί σκόπιμο και το πολύ 3 μέρες για τη Φάση A και για τη Φάση B!

Φάση A

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**.
- **Πηγαίος 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 A) τόσο πιο σωστή, εύκολη και επιτυχημένη θα είναι η υλοποίηση (Φάση B). Επειδή τα προηγούμενα χρόνια παρατηρήσαμε ότι όσοι δεν είχαν κάνει καλή Φάση A σπάνια έκαναν καλή Φάση B, τα τελευταία χρόνια υπάρχει η ρήτρα 5 και στη Φάση A,

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 επιτύχετε.

UML Plugins

Για το *Eclipse* μπορείτε να χρησιμοποιήσετε το ObjectAid. Θα πρέπει να κάνετε register για να το χρησιμοποιήσετε: <https://www.objectaid.com/obtain-license>. Οδηγίες εγκατάστασης μπορείτε να βρείτε εδώ: <https://www.objectaid.com/install-objectaid>. Για class diagrams δείτε το <http://www.objectaid.com/class-diagram>.

Για *Netbeans* (Version >= 8.0) αναλυτικές οδηγίες για το plugin easyUML υπάρχουν εδώ <https://elearn.uoc.gr/mod/resource/view.php?id=54784>. Εναλλακτικά, για το Netbeans μπορεί κάποιος να χρησιμοποιήσει και το visual paradigm <http://www.visual-paradigm.com/product/vpuml/tutorials/modelinginnetbeans.jsp>.

Για το IntelliJ, μπορεί κάποιος να βρει αναλυτικές οδηγίες στο link: https://www.jetbrains.com/help/idea/class-diagram.html#manage_class_diagram, ή να χρησιμοποιήσει το UML Design Tool Plugin <https://plugins.jetbrains.com/plugin/8394-uml-design-tool-plugin>.

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. 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 μορφή javadoc σχολίων.	15
Οργάνωση των κλάσεων σε ιεραρχίες με στόχο την μέγιστη δυνατή επαναχρησιμοποίηση του κώδικα. Χρησιμοποίηση abstract κλάσεων ή/και interfaces για τις βασικές κλάσεις Player και Tile.	15
Να υπάρχουν οι κλάσεις και οι απαραίτητες μέθοδοι (25%) Tile, Board, Player, Character, Controller, View Πρέπει να υπάρχουν οι μέθοδοι για τα βασικά στοιχεία του παιχνιδιού Ενδεικτικά κάποιες βασικές μέθοδοι (θα πρέπει να ορίσετε και αρκετές ακόμα): <ul style="list-style-type: none">• Μέθοδοι για την αρχικοποίηση του παιχνιδιού (ταμπλό, παίκτες, πλακίδια)• Μέθοδος που καθορίζει τη σειρά (ποιος παίκτης παίζει)• Μέθοδοι για την παραλαβή πλακιδίων• Μέθοδος που ελέγχει αν τελείωσε το παιχνίδι/νικητής• Μέθοδος που υπολογίζει το σκορ κάθε παίκτη	25
Σύνολο	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 αναλύεται :
 - η τελική σχεδίαση της εφαρμογής,
 - ποιές 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 ό,τι 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 δε θα απαντώνται.

Αναλυτική βαθμολόγηση για τη Β φάση

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

Παραδοτέα	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	
Αρχικοποίηση Παικτών-Ταμπλό-Στοιβάς Καρτών	10
Τήρηση Σειράς	7
Τράβηγμα πλακιδίων από σακούλα και τοποθέτηση στις αντίστοιχες περιοχές	8
Τράβηγμα πλακιδίων από το ταμπλό	8
Χρήση καρτών χαρακτήρα	8
Υπολογισμός πόντων για μωσαϊκά	8
Υπολογισμός πόντων για σκελετού	8
Υπολογισμός πόντων για αμφορείς	9
Υπολογισμός πόντων για αγάλματα	9
Έλεγχος για περιοχή κατολίσθησης - Τέλος παιχνιδιού	5
Παίξε Ενάντια στον Υπολογιστή (Bonus 5%)	5
Αποθήκευση Παιχνιδιού (Bonus 5%)	5
Σύνολο	110

ΣΗΜΑΝΤΙΚΟ: Η **ελάχιστη** υλοποίηση που χρειάζεται κάποιος για να πάρει βαθμό **50/100** είναι να υπάρχει αναφορά, σχόλια, JUnit tests, UML Diagrams, γραφικό περιβάλλον, σωστή αρχικοποίηση του ταμπλό, των παικτών και των πλακιδίων και να γίνεται έστω κάποια κίνηση από τον κάθε παίκτη μαζί με τήρηση σειράς.

Καλή Εργασία



Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών

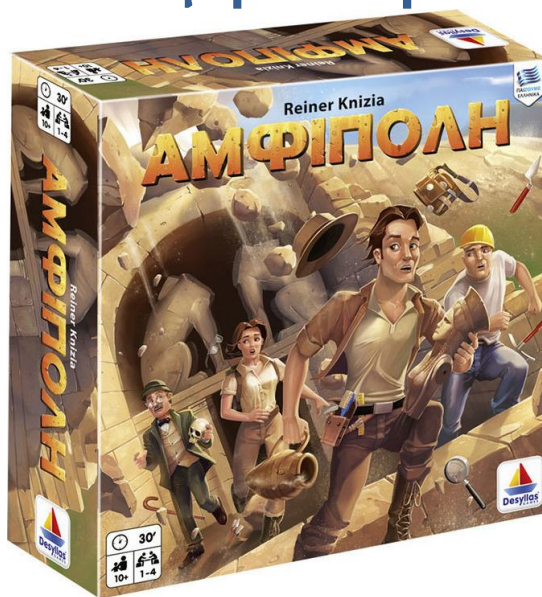
HY252 – Αντικειμενοστρεφής Προγραμματισμός (Χειμερινό εξάμηνο 2020-2021)

Διδάσκων: Γιάννης Τζίτζικας

Θέμα: **Ατομική Εργασία (project) μαθήματος**

Βοηθοί: Χ.Νίκας (επικεφαλής) και οι λοιποί βοηθοί

Αμφίπολη



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 το επιτραπέζιο παιχνίδι “Αμφίπολη”.

Η εργασία είναι ατομική και απαγορεύεται η χρήση κώδικα που δεν έχετε γράψει οι ίδιοι, είτε από το διαδίκτυο είτε από κάποιο συμφοιτητή σας. Σε περίπτωση εντοπισμού αντιγραφής η εργασία θα μηδενίζεται.

Σκοπός του Παιχνιδιού

Η ανασκαφή στον αρχαιολογικό χώρο της Αμφίπολης απειλείται από κατολισθήσεις. Στο παιχνίδι παίρνετε το ρόλο του αρχηγού μιας ομάδας αρχαιολόγων. Ο σκοπός του παιχνιδιού είναι να μαζέψετε όσους περισσότερους πόντους μπορείτε συλλέγοντας αρχαιολογικά ευρήματα. Το παιχνίδι παίζεται με 4 παίκτες.

Ένα σχετικό βίντεο μπορείτε να δείτε στη διεύθυνση:
<https://www.youtube.com/watch?v=4dqPWdZO4vk>

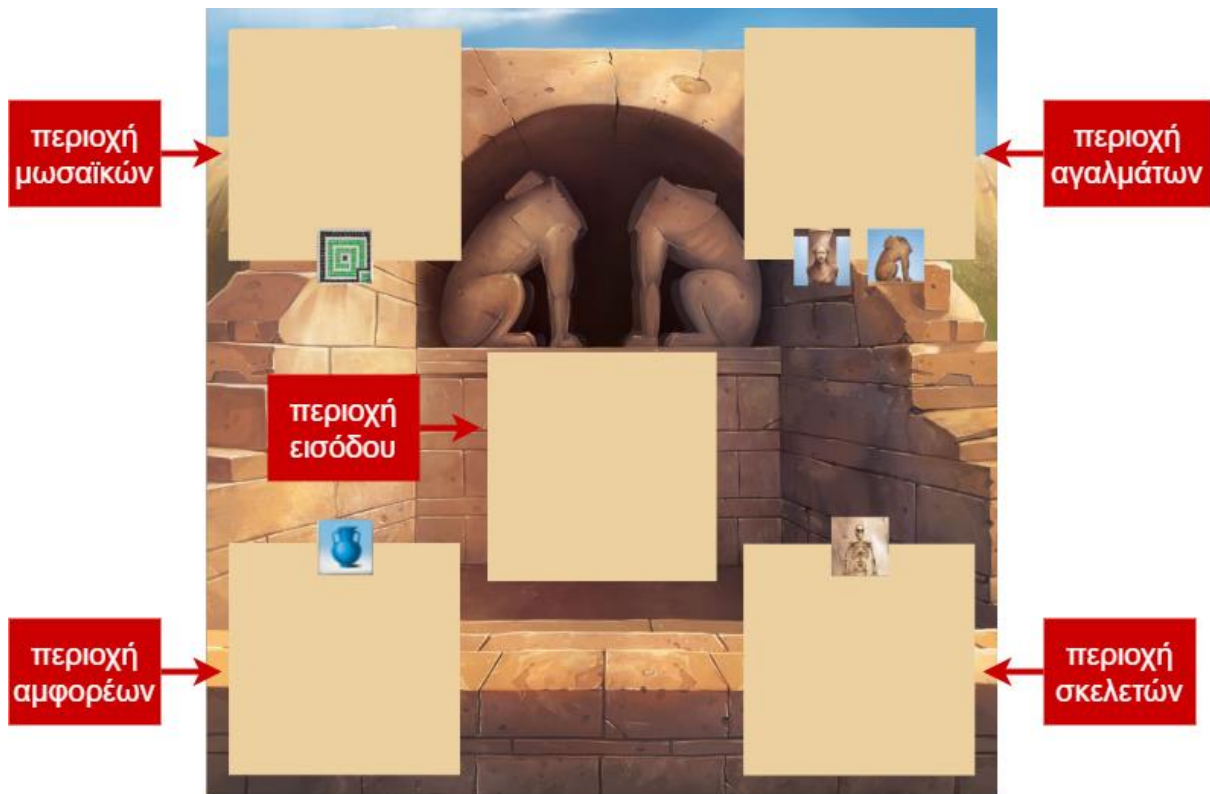
Σε σχέση με την επίσημη έκδοση του παιχνιδιού, υπάρχουν κάποιες παραλλαγές, οπότε θα πρέπει να βασιστείτε στην παρούσα εκφώνηση για τους κανόνες.

Αναλυτική Περιγραφή της Εργασίας

Περιεχόμενα παιχνιδιού

Τα παιχνίδι που αποτελείται από :

- 1 ταμπλό που περιλαμβάνει
 - Την περιοχή εισόδου στον αρχαιολογικό χώρο, όπου τοποθετούνται τα πλακίδια (tiles) κατολίσθησης
 - Τις περιοχές τοποθέτησης πλακιδίων ευρημάτων. Υπάρχουν 4 κατηγορίες πλακιδίων ευρημάτων: μωσαϊκά, αμφορείς, αγάλματα, και σκελετοί
- 16 κάρτες χαρακτήρων (4 για κάθε παίκτη)
- 27 πλακίδια μωσαϊκών (9 πράσινα, 9 κόκκινα, και 9 κίτρινα)
- 24 πλακίδια αγαλμάτων (12 καρυάτιδες και 12 σφίγγες)
- 24 πλακίδια κατολίσθησης
- 30 πλακίδια σκελετών. Ένα πλακίδιο σκελετού μπορεί να είναι το πάνω ή κάτω μέρος ενός σκελετού, και να ανήκει σε μικρό ή μεγάλο σκελετό
 - 10 πάνω μέρη μεγάλου σκελετού
 - 10 κάτω μέρη μεγάλου σκελετού
 - 5 πάνω μέρη μικρού σκελετού
 - 5 κάτω μέρη μικρού σκελετού
- 30 πλακίδια αμφορέων
 - 5 από κάθε χρώμα
 - τα χρώματα είναι: μπλε, καφέ, κόκκινο, πράσινο, κίτρινο, και μωβ
- μία σακούλα που περιέχει όλα τα πλακίδια στην αρχή του παιχνιδιού



Εικόνα 1: Το ταμπλό του παιχνιδιού

Προετοιμασία του παιχνιδιού

Η αρχικοποίηση του παιχνιδιού περιλαμβάνει:

- την αρχικοποίηση του ταμπλό
- την αρχικοποίηση όλων των πλακιδίων στη σακούλα
- την αρχικοποίηση των παικτών
- την ανάθεση ενός χρώματος (κίτρινο, κόκκινο, μπλε, μαύρο) και των 4 καρτών χαρακτήρων αυτού του χρώματος σε κάθε παίκτη
- την τοποθέτηση ενός πλακιδίου ευρήματος από κάθε κατηγορία από την σακούλα στην αντίστοιχη περιοχή τοποθέτησης
- την επιλογή του ποιος παίκτης θα παίξει πρώτος (π.χ. random).

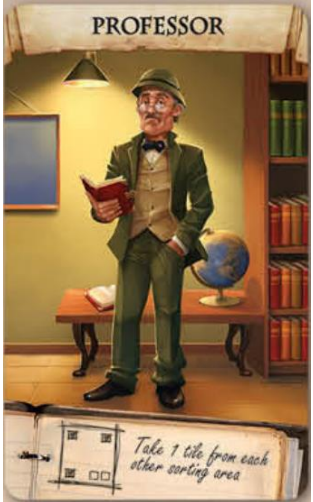
Ποιες κινήσεις γίνονται στη σειρά κάθε παίκτη

Στη σειρά του ο κάθε παίκτης:

1. Παίρνει 4 πλακίδια από τη σακούλα, και τα τοποθετεί στις αντίστοιχες περιοχές τοποθέτησης. Αν τραβήξει πλακίδιο κατολίσθησης, καλύπτει με αυτό μία κενή θέση στην περιοχή εισόδου στον αρχαιολογικό χώρο
2. Παίρνει μέχρι 2 πλακίδια από **μόνο μία** περιοχή τοποθέτησης
3. Χρησιμοποιεί **προαιρετικά** μία κάρτα χαρακτήρα. Οι κάρτες χαρακτήρα μπορούν να χρησιμοποιηθούν μόνο μία φορά σε ένα παιχνίδι

Ιδιότητες χαρακτήρων

 <p>ASSISTANT</p> <p>Take 1 tile from any sorting area</p>	<p>Βοηθός: Ο παίκτης παίρνει 1 πλακίδιο από οποιαδήποτε περιοχή τοποθέτησης</p>
 <p>ARCHAEOLOGIST</p> <p>Take 2 tiles from any other sorting area</p>	<p>Αρχαιολόγος: Ο παίκτης παίρνει μέχρι 2 πλακίδια από οποιαδήποτε περιοχή τοποθέτησης, εκτός από αυτή που διάλεξε νωρίτερα στη σειρά του</p>
 <p>DIGGER</p> <p>Take 2 tiles from the same sorting area</p>	<p>Εκσκαφέας: Ο παίκτης παίρνει μέχρι 2 πλακίδια από την περιοχή που διάλεξε νωρίτερα στη σειρά του</p>



	<p>Καθηγητής: Ο παίκτης παίρνει ένα πλακίδιο από κάθε περιοχή, εκτός από αυτή που διάλεξε νωρίτερα στη σειρά του</p>
---	---

Τέλος παιχνιδιού

Το παιχνίδι τελειώνει όταν όλες (16) οι θέσεις στην περιοχή εισόδου στον αρχαιολογικό χώρο καλυφθούν από πλακίδια κατολίσθησης. Νικητής είναι ο παίκτης με τους περισσότερους πόντους.

Υπολογισμός πόντων

Οι πόντοι υπολογίζονται σύμφωνα με τους παρακάτω κανόνες:

- **Μωσαϊκά:** Ένα ολοκληρωμένο μωσαϊκό αποτελείται από 4 πλακίδια μωσαϊκών.
 - Τα μωσαϊκά που αποτελούνται από 4 πλακίδια του ίδιου χρώματος κερδίζουν **4 πόντους**

 - Τα μωσαϊκά που αποτελούνται από διαφορετικά χρώματα κερδίζουν **2 πόντους**

 - Τα μωσαϊκά που δεν είναι ολοκληρωμένα δεν κερδίζουν πόντους
- **Σκελετοί:** Υπάρχουν 2 κατηγορίες σκελετών. Οι μεγάλοι, που ανήκουν σε ενηλίκους και οι μικροί που ανήκουν σε παιδιά. Ένας ολοκληρωμένος σκελετός αποτελείται από 2 πλακίδια (πάνω μέρος και κάτω μέρος).

- Μία οικογένεια, δηλαδή 2 ολοκληρωμένοι σκελετοί ενηλίκων και ένας ολοκληρωμένος σκελετός παιδιού κερδίζει **6 πόντους**
- Οποιοσδήποτε ολοκληρωμένος σκελετός δεν ανήκει σε οικογένεια κερδίζει **1 πόντο**
- Οι μη ολοκληρωμένοι σκελετοί δεν κερδίζουν πόντους
- **Αμφορείς:** Ένα σύνολο από πλακίδια αμφορέων κερδίζει πόντους με βάση τον αριθμό των διαφορετικών χρωμάτων από τα οποία αποτελείται το σύνολο.

Αριθμός διαφορετικών χρωμάτων	Πόντοι
6 	6
5 	4
4 	2
3 	1

Οι παραπάνω πόντοι ισχύουν για **οποιονδήποτε συνδυασμό διαφορετικών χρωμάτων**, όχι μόνο αυτών που απεικονίζονται.

- **Αγάλματα:** Υπάρχουν 2 τύποι αγαλμάτων που βαθμολογούνται ανεξάρτητα, οι καρυάτιδες και οι σφίγγες. Για κάθε μία από τις 2 κατηγορίες αγαλμάτων, οι παίκτες που έχουν τα περισσότερα αγάλματα παίρνουν από 6 πόντους ο κάθε ένας και οι παίκτες που έχουν τα λιγότερα αγάλματα παίρνουν όλοι 0 πόντους. Οι υπόλοιποι παίρνουν 3 πόντους.

Bonus 1 (5%)

Δυνατότητα αποθήκευσης του παιχνιδιού και μετέπειτα συνέχισης του σε επόμενο στιγμιότυπο εκτέλεσης της εφαρμογής. Συγκεκριμένα, θα πρέπει να δίδεται στο χρήστη μέσω του μενού μια επιλογή αποθήκευσης και εξόδου από την εφαρμογή. Αυτή η επιλογή θα έχει ως αποτέλεσμα τη μόνιμη αποθήκευση (persistent storage) της κατάστασης (state) του παιχνιδιού εκείνη τη δεδομένη χρονική στιγμή. Αφήνεται στη δική σας ευχέρεια, ως σχεδιαστές της εφαρμογής, να ορίσετε τι θα πρέπει να αποθηκεύσετε και πως. Έπειτα, δεδομένου του ότι έχει προηγηθεί μια τέτοια ενέργεια αποθήκευσης, θα πρέπει η εφαρμογή σας να είναι σε θέση πάλι μέσω του μενού εκτέλεσης να προσφέρει τη δυνατότητα συνέχισης του συγκεκριμένου παιχνιδιού.

Bonus 2 (5%)

Παιχνίδι για έναν παίκτη. Το “Αμφίπολη” μπορεί να παιχτεί από έναν μόνο παίκτη με τις παρακάτω αλλαγές:

- Στην αρχή του παιχνιδιού, τοποθετούνται 8 πλακίδια κατολίσθησης στην περιοχή εισόδου στον αρχαιολογικό χώρο.
- Στον αρχαιολογικό χώρο υπάρχει ένας κλέφτης. Όποτε ο παίκτης τραβάει πλακίδιο κατολίσθησης, το τοποθετεί κανονικά στην περιοχή εισόδου και τελειώνει αμέσως τη σειρά του. Αμέσως μετά ο κλέφτης παίρνει όλα τα πλακίδια από όλες τις περιοχές τοποθέτησης.
- Στο τέλος του παιχνιδιού ο παίκτης κερδίζει αν έχει περισσότερους πόντους από τον κλέφτη.

Δώστε στην αρχή του παιχνιδιού την επιλογή το παιχνίδι να παιχτεί με 1 η 4 παίκτες, και προσαρμόστε τους κανόνες ανάλογα.

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 απεικόνισή τους (view). Βασικός στόχος μιας τέτοιας αποσυνδεδεμένης 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 διάφορους 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

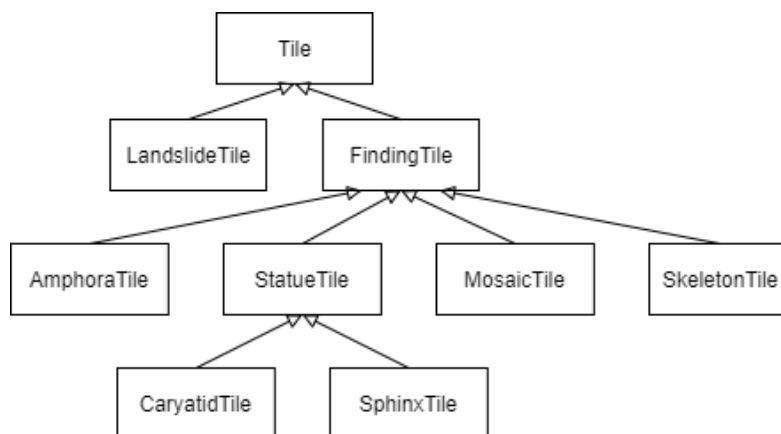
ενεργειών του παιχνιδιού με τη γραφική του απεικόνιση (Controller). Δείτε ένα video για να το καταλάβετε πλήρως: <http://www.newthinktank.com/2013/02/mvc-java-tutorial/>.

Πιο συγκεκριμένα το Model μπορεί να θεωρηθεί ότι αποτελείται από οτιδήποτε σχετίζεται με τα δεδομένα του παιχνιδιού. Υπό αυτήν την έννοια τα «πλακίδια», το «ταμπλό», οι «παίκτες» αποτελούν μέρος του μοντέλου του παιχνιδιού καθώς περιγράφουν τα εκάστοτε δεδομένα που καλείται να διαχειριστεί 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

Το model θα πρέπει να περιλαμβάνει όλα τα περιεχόμενα του παιχνιδιού, δηλαδή θα πρέπει να υπάρχουν κλάσεις για το ταμπλό, τα πλακίδια, τη σακούλα των πλακιδίων, τους παίκτες, τις κάρτες χαρακτήρα κλπ.

Πιο συγκεκριμένα, μία βασική κλάση είναι η abstract κλάση **Tile**. Η κλάση αυτή μοντελοποιεί ένα πλακίδιο του παιχνιδιού και ορίζει τα γνωρίσματα του, όπως κατηγορία, χρώμα κλπ. Επειδή υπάρχουν πολλοί τύποι πλακιδίων, και κάθε τύπος λειτουργεί διαφορετικά στον υπολογισμό των πόντων, και τοποθετείται σε διαφορετική περιοχή τοποθέτησης, είναι καλό να χωριστεί η κλάση Tile σε υποκλάσεις. Ένας τρόπος να χωριστεί η κλάση Tile σε υποκλάσεις φαίνεται στην εικόνα 2. Για τα πλακίδια θα είναι επίσης χρήσιμη μια κλάση **Bag**, που θα μοντελοποιεί τη σακούλα που περιέχει τα πλακίδια στην αρχή του παιχνιδιού.



Εικόνα 2: Υποκλάσεις της Tile

Μία βασική κλάση είναι η **Character**, που μοντελοποιεί τις κάρτες χαρακτήρα. Για μία κάρτα χαρακτήρα είναι σημαντικό να ξέρουμε αν έχει χρησιμοποιηθεί, σε ποιόν παίκτη ανήκει, και τη λειτουργικότητα της όταν χρησιμοποιηθεί. Επίσης η Character μπορεί να χωριστεί σε υποκλάσεις με βάση τον χαρακτήρα (Assistant, Archaeologist, Digger, Professor)

Απαραίτητες κλάσεις του παιχνιδιού είναι αυτές του παίκτη. Η κλάση **Player** προσομοιώνει τον παίκτη του παιχνιδιού. Κάθε παίκτης έχει ένα όνομα, και τα πλακίδια που έχει μαζέψει από το

ταμπλό. Επίσης θα έχει μία μέθοδο που στο τέλος του παιχνιδιού θα υπολογίζει τους πόντους του, με βάση τα πλακίδια που έχει μαζέψει.

Επίσης είναι σημαντική μία κλάση **Board**, όπου θα μοντελοποιεί το ταμπλό του παιχνιδιού. Η κλάση Board θα πρέπει να περιέχει γνωρίσματα για τις 5 περιοχές του ταμπλό και τα πλακίδια που περιέχουν αυτές. Επίσης, μπορεί να είναι υπεύθυνη για την αρχικοποίηση του παιχνιδιού και να περιέχει μεθόδους για τον έλεγχο της σειράς των παικτών, και τον έλεγχο για το τέλος του παιχνιδιού.

Controller

Ο Controller έχει πρόσβαση στα δεδομένα του παιχνιδιού και στην διεπαφή. Από την διεπαφή λαμβάνει πληροφορία για τις κινήσεις των παικτών και με αυτή 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 παίκτη). **Η κλάση αυτή αποτελεί τον «εγκέφαλο επικοινωνίας και συντονισμού» 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**, 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. Συγκεκριμένα, είναι σημαντικό να σκεφτείτε και να σχεδιάσετε το πώς θα επικοινωνούν τα διαφορετικά κομμάτια του model και του controller μεταξύ τους, π.χ. ποια θα είναι η ροή όταν έρθει η σειρά ενός παίκτη, όταν χρησιμοποιήσει μια κάρτα, ποιες μέθοδοι (που θα βρίσκονται σε διαφορετικές κλάσεις) και με ποια σειρά θα πρέπει να χρησιμοποιηθούν.

Γραφική Απεικόνιση

Η γραφική διεπαφή του παιχνιδιού πρέπει να παρουσιάζει το ταμπλό και να δίνει τη δυνατότητα στους παίκτες να παίζουν ταυτόχρονα το παιχνίδι σε έναν υπολογιστή. Ένα σχεδιαστικό παράδειγμα υλοποίησης της εν λόγω διεπαφής παρουσιάζεται στην Εικόνα 3. 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 τα ονόματα των παικτών, ήχους (π.χ. όταν ένα πιόνι κινείται), κλπ.

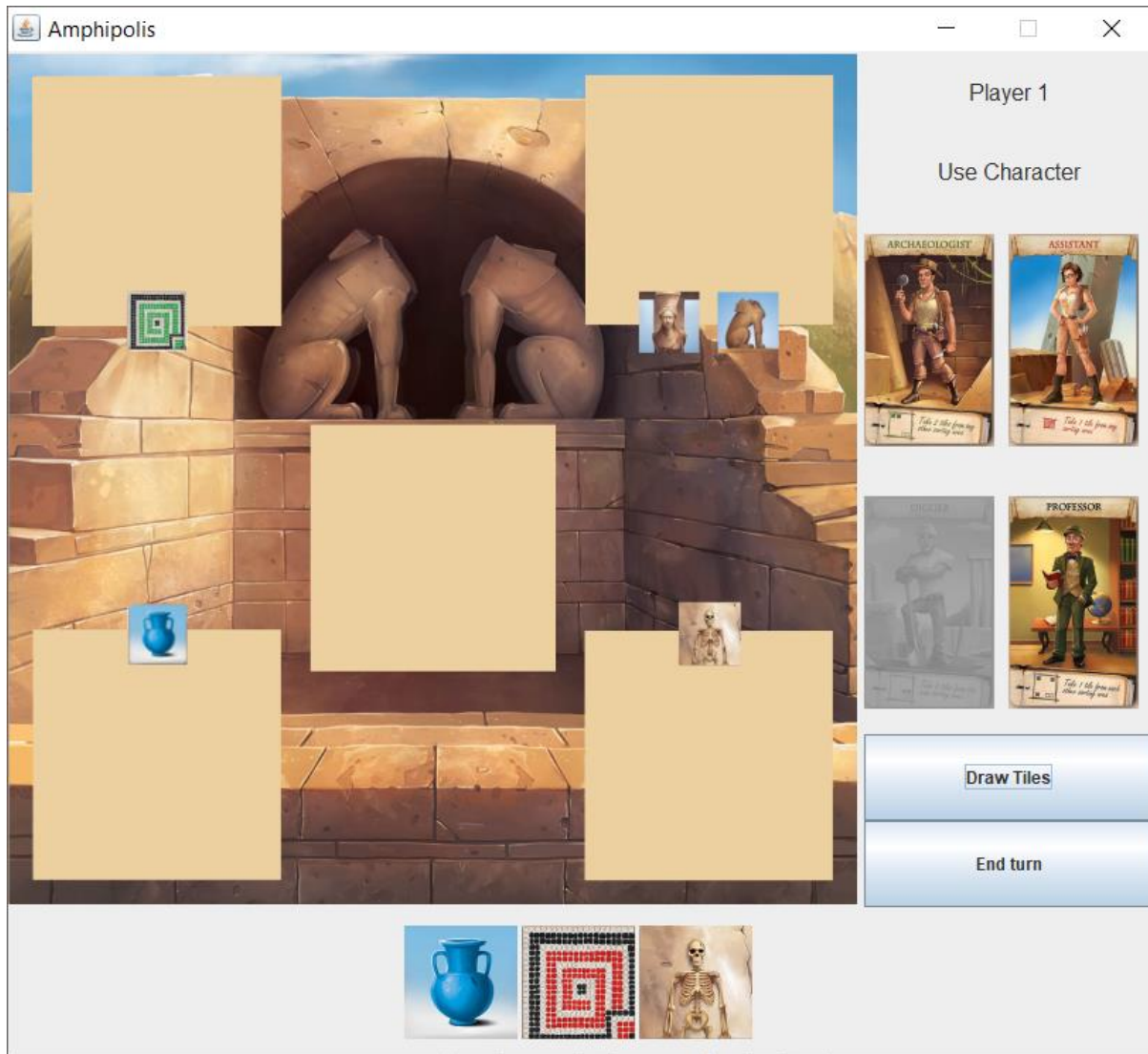
Προτεινόμενες Κλάσεις – View

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.

Στην εικόνα 3 βλέπουμε μια ενδεικτική υλοποίηση της γραφικής διεπαφής. Πάνω αριστερά υπάρχει το ταμπλό. Για το ταμπλό μπορείτε να χρησιμοποιήσετε την κλάση `JLayeredPane`, όπου κάποιος μπορεί να τοποθετήσει ένα `component` (π.χ., κουμπί, ετικέτα, κείμενο) πάνω σε ένα άλλο και κάθε `component` έχει συγκεκριμένες συντεταγμένες. Για παράδειγμα, στο ταμπλό μπορούμε να έχουμε ένα βασικό `background` (`JLayeredPane`), και πάνω σε αυτό να φτιάξουμε κάποια πλακίδια που φτιάχτηκαν μέσω στιγμιότυπων `JLabel`. Πάνω δεξιά φαίνεται το όνομα του παίκτη ο οποίος έχει σειρά να παίξει και οι ενέργειες που μπορεί να πραγματοποιήσει. Κάνοντας κλικ σε μια κάρτα χαρακτήρα μπορεί να τη χρησιμοποιήσει. Όταν μία κάρτα χαρακτήρα έχει ήδη χρησιμοποιηθεί μπορεί να φαίνεται ως `disabled` (<https://docs.oracle.com/javase/7/docs/api/javax/swing/GrayFilter.html>), ή να αφαιρείται από την γραφική διεπαφή. Με το κουμπί “draw tiles” ο παίκτης τραβάει πλακίδια από τη σακούλα, και με το κουμπί “end turn” τελειώνει τη σειρά του. Τέλος, στο κάτω μέρος φαίνονται τα πλακίδια που έχει τραβήξει ο παίκτης.

Επίσης, στο τέλος του παιχνιδιού θα πρέπει να ανακοινώνεται ο νικητής και οι πόντοι κάθε παίκτη. Αυτό μπορεί να γίνει μέσω `Java Dialog`, όπου μπορείτε να χρησιμοποιήσετε οποιοδήποτε τύπο `Java Dialogs` σας βολεύει (δείτε <http://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html>).

Δεν υπάρχει περιορισμός στην υλοποίηση των γραφικών, ουσιαστικά προσπαθήστε να χρησιμοποιήσετε την υλοποίηση που σας φαίνεται προτιμότερη.



Εικόνα 3: Ενδεικτικό GUI παιχνιδιού

Βοηθητικά για να Ξεκινήσετε (SOS)

Επίσημες Οδηγίες και Βίντεο στα Αγγλικά. **Προσοχή, οι οδηγίες μπορεί να είναι λίγο διαφορετικές από την εκφώνηση. Εσείς πάντα στηρίζετε στις οδηγίες της εκφώνησης!!!**

Οδηγίες: στο elearn

Video: <https://www.youtube.com/watch?v=4dqPWdZO4vk> (έχει αγγλικούς υπότιτλους)

Παρακάτω αναγράφονται κάποια πράγματα που μπορούν να σας βοηθήσουν στην υλοποίηση:

- Οι φωτογραφίες για όλα τα περιεχόμενα του παιχνιδιού και template για την αναφορά της εργασίας σας έχουν ανέβει στο elearn!
- Ενδεικτικό βοήθημα για το MVC (Model View Controller) μοντέλο σε Java <http://www.newthinktank.com/2013/02/mvc-java-tutorial> .
- Επίσης **θα σας δοθεί ένα project προηγούμενης χρονιάς** ώστε να αποκτήσετε από νωρίς μια αίσθηση του πως περίπου θα είναι η τελική σας υλοποίηση.

- **(SOS)** Τέλος μετά το φροντιστήριο των γραφικών, ανεβαίνουν παραδείγματα με γραφικά που σας βοηθούν να ξεκινήσετε! Μπορείτε να ξεκινήσετε με αυτά τα παραδείγματα και να τα αλλάξετε έτσι ώστε να φτιάξετε το γραφικό περιβάλλον της εργασίας σας.

Καλή Εργασία!!!