

ΗΥ-252 Αντικειμενοστρεφής Προγραμματισμός

Προγραμματιστική Εργασία Χειμερινού Εξαμήνου 2006

Υπάρχουν περισσότερα από 400 είδη πασιέντζας. Όλα όμως είναι παρόμοια στην φιλοσοφία και διαφέρουν μόνο σε επιμέρους κανόνες του παιχνιδιού. Σε αυτήν την εργασία θα σχεδιάσετε σε μια πρώτη φάση μια απλή βιβλιοθήκη χρησιμοποιώντας τεχνικές αντικειμενοστρεφούς προγραμματισμού έτσι ώστε να διευκολύνεται η υλοποίηση πολλών παραλλαγών της πασιέντζας. Στην συνέχεια, θα χρησιμοποιήσετε αυτή την βιβλιοθήκη για να υλοποιήσετε σε Java δύο παραλλαγές του παιχνιδιού, την “Klondike” (που είναι διαθέσιμη στα Windows) και την “FreeCell”.

1 Πασιέντζα

Ένα παιχνίδι πασιέντζας χρησιμοποιεί μία ή περισσότερες τράπουλες των 52 φύλλων. Κάθε τράπουλα περιέχει τέσσερα χρώματα φύλλων:

♠ Μπαστούνι, ♣ Σπαθί, ♥ Κούπα, ♦ Καρό

Για κάθε ένα από αυτά τα χρώματα υπάρχουν στην τράπουλα φύλλα με τους ακόλουθους βαθμούς: ace, two, three, four, five, six, seven, eight, nine, ten, jack, queen και king. Η παραπάνω ακολουθία λέγεται οικογένεια (family) εφόσον όλα τα φύλλα της ανήκουν στο ίδιο χρώμα. Τα περισσότερα παιχνίδια πασιέντζας επικεντρώνονται στην μετακίνηση των φύλλων της τράπουλας από έναν σωρό σε κάποιο άλλο, έτσι ώστε να συμπληρωθούν και οι τέσσερις οικογένειες της τράπουλας. Στη συνέχεια θα περιγράψουμε αναλυτικά τις δύο παραλλαγές του παιχνιδιού που καλείστε να υλοποιήσετε.

1.1 Klondike

Στο παιχνίδι “Klondike” υπάρχουν τέσσερις βασικοί σωροί φύλλων που περιγράφονται στην συνέχεια (βλέπε σχήμα 1):

1. Talon (απόθεμα): Στην αρχή του παιχνιδιού μοιράζονται μερικά φύλλα από την τράπουλα (deck). Όσα δεν μοιραστούν αποτελούν το απόθεμα από το οποίο μπορούμε να τραβάμε χαρτιά στην συνέχεια του παιχνιδιού.
2. Waste Pile (σωρός κατανάλωσης): Κάθε φορά που τραβάμε ένα φύλλο από το απόθεμα (Talon) το προσθέτουμε στο σωρός κατανάλωσης. Το φύλλο στην κορυφή του αυτού του σωρού μπορεί να προστεθεί στους υπόλοιπους σωρούς του παιχνιδιού (Foundations, Playing Piles). Όταν το απόθεμα (Talon) αδειάζει, τότε για να τραβήξουμε ένα φύλλο θα πρέπει να ξαναμεταφέρουμε όλα τα φύλλα από τον σωρό κατανάλωσης (Waste Pile) στο απόθεμα (Talon).
3. Foundations (βάσεις): Είναι σωροί που κρατάνε τα φύλλα κάθε χρώματος σε αυξανόμενο βαθμό. Αρχικά οι τέσσερις βάσεις (Foundations) είναι άδειες. Ξεκινώντας από τον άσο προσθέτουμε διαδοχικά φύλλα τα οποία τραβούμε από τις στοίβες παιχνιδιού (Playing piles), ή τον σωρό κατανάλωσης (Waste pile) με απώτερο

σκοπό να συμπληρώσουμε και τις τέσσερις οικογένειες της τράπουλας.

4. **Playing Piles (στοίβες):** Αυτές είναι επτά στοίβες φύλλων στις οποίες ουσιαστικά διαδραματίζεται όλο το παιχνίδι. Αρχικά, οι στοίβες παιχνιδιού έχουν από 0 έως 6 κρυμμένα φύλλα και στην κορυφή ένα φανερό. Μπορούμε να μεταφέρουμε ένα φύλλο πάνω σε κάποια στοίβα παιχνιδιού (Playing Pile) αν έχει στην κορυφή ένα φανερό φύλλο του οποίου το χρώμα είναι διαφορετικό ενώ ο βαθμός του είναι κατά ένα μεγαλύτερος από αυτό που πάμε να προσθέσουμε. Έτσι σχηματίζονται εναλλασσόμενες ακολουθίες (δηλ. τα φύλλα εναλλάσσονται σε χρώμα, κόκκινο, μαύρο). Μπορούμε αντίστοιχα να μετακινήσουμε ολόκληρες τέτοιες εναλλασσόμενες ακολουθίες αρκεί το κορυφαίο φύλλο του σωρού προορισμού να έχει διαφορετικό χρώμα και βαθμό κατά ένα μεγαλύτερο από το πρώτο φύλλο της ακολουθίας που επιχειρούμε να προσθέσουμε. Όταν μια στοίβα έχει μόνο κρυμμένα φύλλα τότε μπορούμε να φανερώσουμε το πρώτο (αυτό, για διευκόλυνση, μπορεί να γίνεται και αυτόματα όταν μείνει μια στοίβα χωρίς φανερά φύλλα). Αν πάλι μια στοίβα δεν έχει καθόλου φύλλα, τότε μπορούμε να προσθέσουμε σε αυτήν μόνο ακολουθίες που ξεκινάνε με ρήγα. Από ένα σωρό μπορούμε να επιχειρήσουμε να μεταφέρουμε οποιαδήποτε ακολουθία φανερών φύλλων αρκεί αυτή να ξεκινάει από το πρώτο φύλλο.

Στήσιμο Παιχνιδιού: Όπως μπορούμε να δούμε στο Σχήμα 1, αρχικά μοιράζουμε την τράπουλα σε όλες τις στοίβες παιχνιδιού (Playing Piles) και ότι περισσέψει το αφήνουμε στο απόθεμα φύλλων (Talon). Σε όλες τις αρχικές θέσεις τα φύλλα θα πρέπει να εισάγονται με τυχαίο τρόπο (ανακατεμένα).



Σχήμα 1. Klondike.

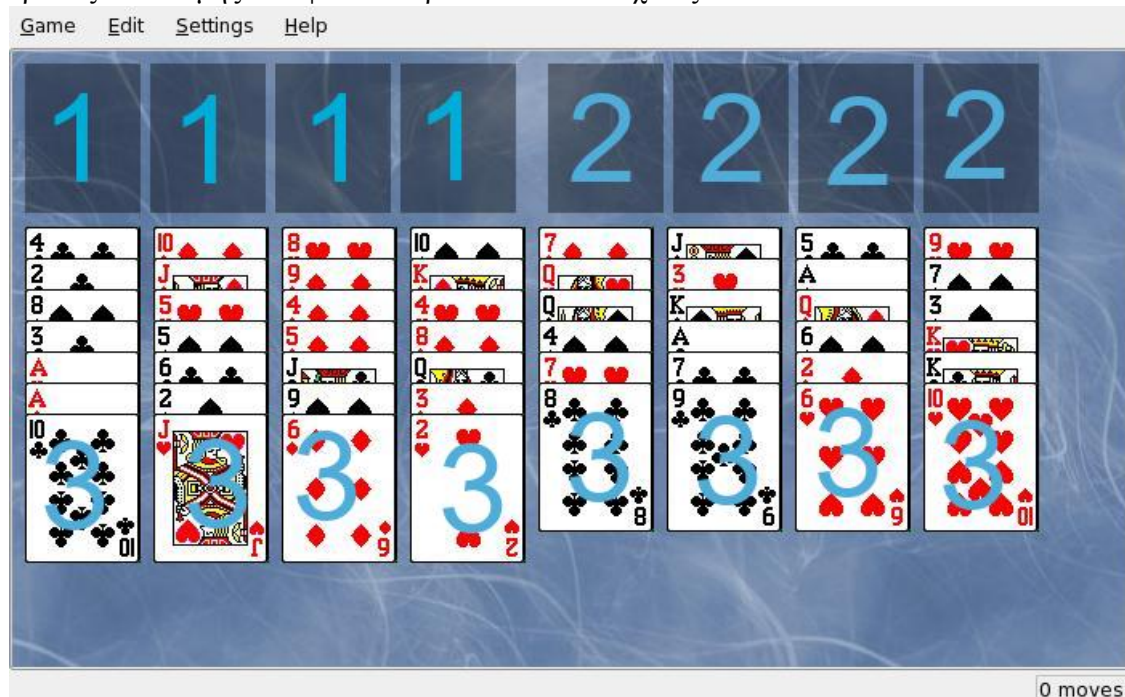
Στόχος: Στόχος του παιχνιδιού είναι να συμπληρωθούν οι τέσσερις βάσεις (Foundations) (μία για κάθε χρώμα) χρησιμοποιώντας τα διαθέσιμα φύλλα του παιχνιδιού. Για να επιτευχθεί αυτό ο παίχτης θα πρέπει να συνδυάζει φύλλα από τις στοίβες παιχνιδιού (Playing Piles) και το απόθεμα φύλλων (Talon) και να αποκαλύπτει σταδιακά κρυμμένα φύλλα από τους σωρούς.

1.2 FreeCell

Σε αυτό το παιχνίδι υπάρχουν τρεις βασικές σωροί φύλλων. Σε αντίθεση με το Klondike, όλα τα φύλλα μοιράζονται στην αρχή του παιχνιδιού. Οι σωροί περιγράφονται παρακάτω (βλέπε σχήμα 2).

1. Free Cells (κυψέλες): Αποτελούν τέσσερις προσωρινές θέσεις στις οποίες μπορούμε να βάζουμε οποιοδήποτε φύλλο, αλλά ένα κάθε φορά.
2. Foundations (βάσεις): Έχουν ακριβώς τον ίδιο ρόλο με το παιχνίδι Klondike.
3. Playing Piles (στοίβες): Οι στοίβες παιχνιδιού διαφέρουν λίγο από το Klondike. Υπάρχουν οκτώ τέτοιες στοίβες στις οποίες μοιράζονται όλα τα φύλλα στην αρχή του παιχνιδιού. Όλα τα φύλλα είναι φανερά. Μπορούμε και πάλι να μεταφέρουμε ένα φύλλο πάνω σε κάποια στοίβα παιχνιδιού (Playing Pile) αν έχει στην κορυφή ένα φανερό φύλλο του οποίου το χρώμα είναι διαφορετικό και ο βαθμός του είναι κατά ένα μεγαλύτερος από αυτό που επιχειρούμε να προσθέσουμε. Παρόλ'αυτά δεν μπορούμε να μεταφέρουμε ολόκληρες ακολουθίες. Η μεταφορά ακολουθιών γίνεται μόνο με την χρήση των προσωρινών θέσεων (Free Cells). Όταν μια στοίβα είναι άδεια μπορούμε να προσθέσουμε σε αυτή οποιοδήποτε φύλλο θέλουμε (όχι μόνο ρήγα).

Στήσιμο Παιχνιδιού: Όπως μπορούμε να δούμε στο Σχήμα 2, αρχικά μοιράζουμε ολόκληρη την τράπουλα στις οκτώ στοίβες (Playing Piles) με όλα τα φύλλα φανερά. Ο τρόπος κατανομής των φύλλων πρέπει να είναι τυχαίος.



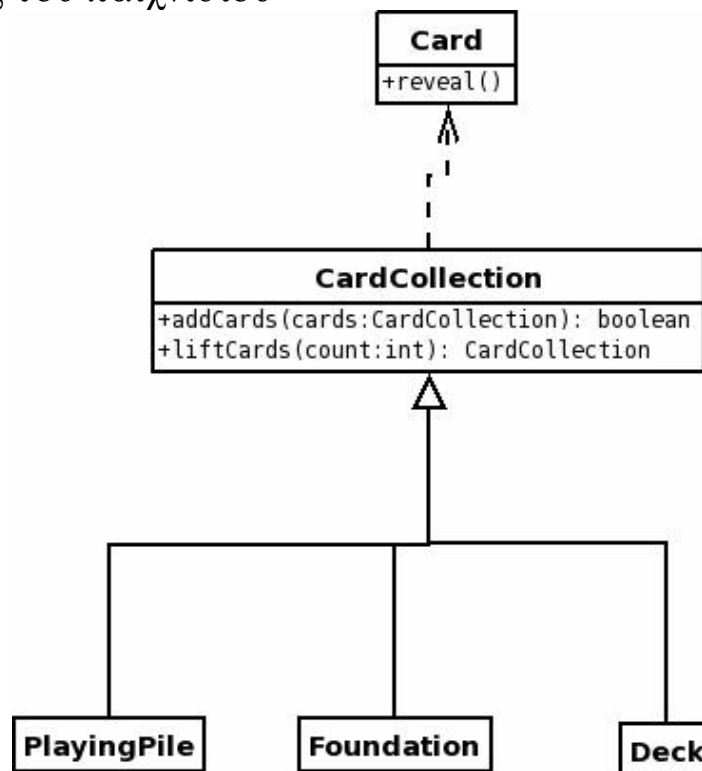
Σχήμα 2. FreeCell.

Στόχος: Στόχος του παιχνιδιού, όπως και στο Klondike, είναι να συμπληρωθούν οι τέσσερις βάσεις (Foundations) (μία για κάθε χρώμα) χρησιμοποιώντας τα χαρτιά που είναι διαθέσιμα. Για να επιτευχθεί αυτό θα πρέπει ο παίχτης να δημιουργεί όσο το δυνατόν μεγαλύτερες ακολουθίες και να αφήνει ελεύθερους σωρούς για μετακινήσεις. Σε αντίθεση με το Klondike σχεδόν όλες οι δυνατές μοιρασιές φύλων οδηγούν σε επιτυχή ολοκλήρωση του παιχνιδιού και είναι στο χέρι του παίχτη να το καταφέρει!

2. Ζητούμενα της εργασίας

Στόχος της εργασίας είναι να σχεδιάσετε και να υλοποιήσετε μία βιβλιοθήκη (δηλ ένα σύνολο κλάσεων και διεπαφών) για παιχνίδια πασιέντζας καθώς και να αναπτύξετε τα δύο παιχνίδια Klondike και FreeCell χρησιμοποιώντας αυτή την βιβλιοθήκη. Λόγω της πληθώρας των παιχνιδιών πασιέντζας, θα πρέπει η φιλοσοφία της εργασίας σας να εστιάζει στην δυνατότητα επαναχρησιμοποίησης και επέκτασης της λειτουργικότητας των κλάσεων που σχεδιάζετε. Αυτό απαιτεί την χρήση αφαιρετικών κλάσεων, διεπαφών, κληρονομικότητας και πολυμορφισμού. Στην συνέχεια, χρησιμοποιώντας τις ενότητες λογισμικού που ήδη έχουν αναπτυχθεί, η υλοποίηση κάποιας άλλης παραλλαγής θα είναι ιδιαίτερα απλή.

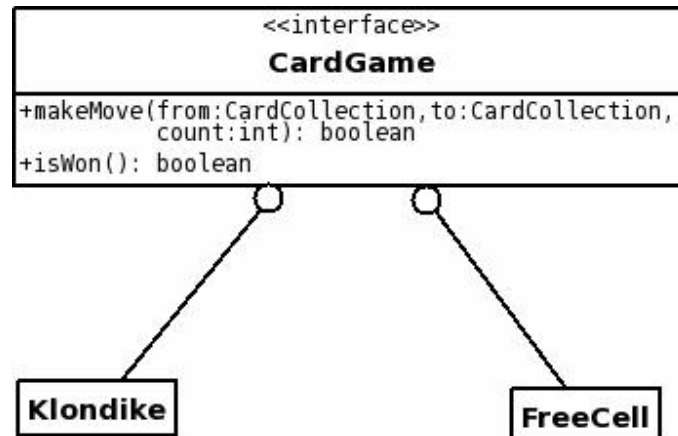
2.1. Πυρήνας του παιχνιδιού



Σχήμα 4. Παράδειγμα τμήματος της ιεραρχίας των κλάσεων.

Η βιβλιοθήκη σας θα πρέπει να περιλαμβάνει κλάσεις για την αναπαράσταση των διάφορων οντοτήτων του παιχνιδιού όπως είναι τα φύλλα της τράπουλας, καθώς και συλλογών φύλλων όπως είναι οι στοίβες παιχνιδιού (Foundations). Μια πρώτη κατεύθυνση για το πώς θα πρέπει να οργανωθούν οι κλάσεις σας δίνεται στο σχήμα 3.

Οποιοδήποτε παιχνίδι πασιέντζας εξελίσσεται σε διακριτά βήματα που είναι οι μετακινήσεις των φύλλων. Κάθε τέτοιο βήμα αλλάζει την κατάσταση του παιχνιδιού. Θα πρέπει να ορίσετε στην βιβλιοθήκη σας μια βασική διεπαφή παιχνιδιού αντίστοιχη με αυτή που σας δίνεται στο σχήμα 4. Οι υλοποιήσεις αυτής της διεπαφής θα χειρίζονται όλους τους σωρούς φύλλων και την κατάσταση τους σε κάποιο βήμα του παιχνιδιού. Επιπλέον σε αυτές θα πρέπει να εκτελούνται οι μετακινήσεις από ένα σωρό φύλλων σε μια κάποιο άλλο και θα γίνεται ο έλεγχος εάν η πασιέντζα έχει ολοκληρωθεί επιτυχώς.



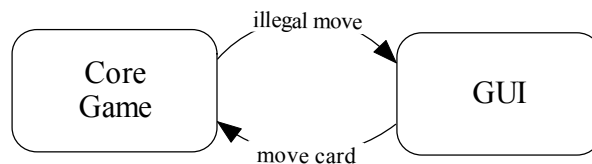
Σχήμα 4. Η βασική διεπαφή ενός παιχνιδιού.

Οι περιορισμοί που συναντάμε σε παιχνίδια πασιέντζας ανήκουν σε δύο είδη. Υπάρχουν οι περιορισμοί προέλευσης προορισμού και οι περιορισμοί ομάδας. Οι πρώτοι απαγορεύουν την μετακίνηση από έναν σωρό σε κάποιο άλλο (π.χ. από τα playing piles στο waste pile). Οι δεύτεροι περιορίζουν για κάθε σωρό ξεχωριστά ποια και πόσα φύλλα μπορούν να προστεθούν ή να αφαιρεθούν. Πού όμως θα πρέπει να επιβάλλονται αυτοί οι περιορισμοί; Μία λύση θα ήταν η αντίστοιχη συνάρτηση **makeMove()** της κάθε υλοποίησης να συμπεριλαμβάνει όλους τους κανόνες. Αυτό όμως απαιτεί κάθε παραλλαγή του παιχνιδιού να υλοποιεί εκ νέου όλους τους κανόνες. Όπως όμως είδαμε στην πρώτη ενότητα οι σωροί που χρησιμοποιούνται και στα δύο παιχνίδια είναι παρόμοιοι. Θα πρέπει επίσης να παρέχεται η δυνατότητα **undo** και **redo** των κινήσεων που έχουν ήδη γίνει σε ένα παιχνίδι. Για να το πετύχετε αυτό, θα πρέπει να ορίσετε μία κλάση για την μετακίνηση και να κρατάτε μία στοίβα από μετακινήσεις φύλλων που έχουν γίνει σε ένα παιχνίδι. Θα πρέπει τέλος να υπάρχουν αντίστοιχες μέθοδοι για την αναίρεση ή την επανεκτέλεση κινήσεων. Σημειώστε ότι στο **Klondike** η εμφάνιση ενός κρυμμένου φύλλου σε κάποια στοίβα παιχνιδιού (**Playing Pile**) δεν αναιρείται (άρα σε αυτή την περίπτωση η στοίβα κινήσεων αδειάζει).

2.2 Γραφική Διεπαφή Χρήσης

Με την υλοποίηση της διεπαφής του σχήματος 4 (δεδομένου ότι έχετε υλοποιήσει και όλες τις απαραίτητες κλάσεις της βιβλιοθήκης) η πασιέντζα μπορεί να τρέξει. Αλλά για να ολοκληρωθεί η υλοποίηση του παιχνιδιού θα πρέπει ο χρήστης να βλέπει επίσης τα χαρτιά και να κάνει κινήσεις. Εδώ ξεκινάει η σχεδίαση της γραφική διεπαφής χρήσης (**GUI**). Αν και είναι εφικτός ο συνδυασμός του λειτουργικού ρόλου των κλάσεων της βιβλιοθήκης σας με την πιθανή τους χρήση στην **GUI**, αυτή η λύση θα πρέπει να αποφευχθεί γιατί δυσχεραίνει την (ανεξάρτητη) επέκταση και των δύο συνιστωσών. Η

σωστή σχεδίαση συνίσταται στην όσο το δυνατόν μεγαλύτερη διαφοροποίηση των δύο συνιστωσών, είτε χρησιμοποιώντας κληρονομικότητα (π.χ. οι κλάσεις της GUI να επεκτείνουν τις κλάσεις της βιβλιοθήκης) είτε συσχέτιση (π.χ. το γραφικό κομμάτι ενός Foundation να κρατάει ένα reference στο Foundation του παιχνιδιού). Έτσι κρατάμε τις κλάσεις που αναπαριστούν την λογική του παιχνιδιού εντελώς ανεξάρτητες από την υλοποίηση του γραφικού περιβάλλοντος. Για παράδειγμα θα μπορεί η GUI να ενημερώνει την μηχανή του παιχνιδιού (core game) για το ποια κίνηση επιθυμεί να κάνει ο παίκτης και η μηχανή να επιστρέφει στην GUI π.χ. αν μια κίνηση είναι παράνομη η αν θα πρέπει τα γραφικά να ανανεωθούν όπως φαίνεται στο παρακάτω σχήμα 5.



Σχήμα 5. Βασική επικοινωνία διεπαφής και παιχνιδιού.

Προσέξτε ότι ακόμα και η GUI προσφέρει πολλά περιθώρια για επαναχρησιμοποίηση, τα οποία και θα πρέπει να εκμεταλλευτείτε. Όπως είπαμε, στόχος μας είναι να κάνουμε την υλοποίηση καινούριων παιχνιδιών όσο το δυνατόν πιο απλή υπόθεση. Στο σχήμα 6 βλέπουμε μια προσέγγιση στην εμφάνιση του GUI.

```

class CardCollectionGUI{
    int x,y;
    CardCollection cardcollection;
    void display(Graphics g){...}
    protected void drawCard(Graphics g,int x,int y, Card card){...}
}

class SpreadCardCollectionGUI extends CardCollectionGUI{
    void display(g){...}
}

class genericGUI{
    Collection<CardCollectionGUI> guielements;

    display(Graphics g){
        drawbackground(g);
        for each(CardCollectionGUI c in guielements){
            c.display(g);
        }
    }

    void addGuiElement(CardCollectionGUI c){...}
}

class SomeGame{
    Talon talon;
    Foundation[] foundations;
    PlayingPiles[] playingpiles;
}

class SomeGameGUI extends genericGUI{
    SomeGame somegame;

    createGuiElements(){
        this.addGuiElement(new CardCollectionGUI(this.getTalon()))
        ...
    }
}
  
```

Σχήμα 6. Ένας πιθανός σκελετός για την διεπαφή

2.3 Γενικά χαρακτηριστικά

Το γραφικό περιβάλλον των δύο παιχνιδιών θα πρέπει να σχεδιαστεί με βάση τα σχήματα 1 και 2. Το κάθε παιχνίδι θα πρέπει να ξεκινάει με μια τράπουλα ήδη μοιρασμένη αλλά θα πρέπει επίσης να δίνετε την δυνατότητα (π.χ. με ένα menu bar) να ξαναμοιραστούν τα φύλλα από την αρχή. Ακόμα θα πρέπει να προβλεφθούν επιλογές για undo, redo των μετακινήσεων των φύλλων. Ιδιαίτερη σημασία δίνουμε στην σχεδίαση ενός αποδοτικού τρόπου μοιράσματος των φύλλων στους διάφορους σωρούς με πραγματικά τυχαίο τρόπο. Η μετακίνηση φύλλων μπορεί να γίνει με δύο τρόπους: είτε θα κάνετε click σε ένα φύλλο, το οποίο θα φαίνεται ως επιλεγμένο, και μετά θα κάνετε click στο φύλλο προορισμού, είτε θα έχετε την δυνατότητα να κάνετε drag and drop το φύλλο ή μια ακολουθία φύλλων πάνω σε κάποιο άλλο φύλλο. Στην πρώτη περίπτωση θα πρέπει να δίνεται η δυνατότητα στον παίχτη με δεξί click να απενεργοποιεί την επιλογή ενός φύλλου. Όταν ο παίχτης επιχειρεί μία κίνηση που δεν επιτρέπεται θα πρέπει με κάποιο (ηχητικό ή οπτικό) μήνυμα να ειδοποιείται ότι η κίνηση αυτή δεν είναι δυνατή. Τέλος θα πρέπει ελέγχονται οι συνθήκες επιτυχούς ολοκλήρωσης του παιχνιδιού.

3. Σχεδιασμός και Υλοποίηση

Η εργασία απευθύνεται σε ομάδες των 2 ατόμων και χωρίζεται σε 2 φάσεις:

Φάση 1η – Σχεδιασμός

Σε αυτή τη φάση πρέπει να γίνει ο σχεδιασμός της εφαρμογής βάσει των αρχών και της μεθοδολογίας του αντικειμενοστρεφούς προγραμματισμού που έχετε διδαχθεί στο μάθημα. Αποτέλεσμα αυτής της φάσης είναι ο καθορισμός των αντικείμενων, των χαρακτηριστικών και της συμπεριφοράς τους, που απαιτούνται για να αναπαραστήσουν την κατάσταση και τις λειτουργίες του παιχνιδιού, όπως έχουν περιγραφεί προηγουμένως.

Προσοχή: τα παραδείγματα που σας έχουν δοθεί στις προηγούμενες ενότητες είναι μόνο ενδεικτικά της λειτουργικότητας του παιχνιδιού και σε καμία περίπτωση δεν είναι πλήρη. Παρέχονται για να σας δώσουν μια πρώτη ιδέα ως προς το σχεδιασμό που απαιτείται.

Παραδοτέα σε αυτή τη φάση είναι :

- Μία αναφορά η οποία θα περιγράφει τα παραπάνω στοιχεία και θα παρουσιάζει το σχέδιο υλοποίησης της εργασίας σας έτσι ώστε να είναι έτοιμο το πέραςμα στην επόμενη φάση της υλοποίησης.
- Οι διεπαφές και το περίγραμμα (outline) των κλάσεων Java (πηγαίος κώδικας) συνοδευόμενα από τα απαραίτητα javadoc σχόλια, στα οποία θα βασιστεί η υλοποίηση του παιχνιδιού στην επόμενη φάση.
- Μια περιγραφή για το πώς μπορεί να χρησιμοποιηθεί η βιβλιοθήκη σας για να δημιουργηθεί ένα καινούριο παιχνίδι πασιέντζας.

Επιγραμματικά, οι σημαντικότερες εργασίες κατά την διάρκεια αυτής της φάσης είναι:

- Αναγνώριση των κλάσεων και διεπαφών για κάθε μικρή και μεγάλη συνιστώσα του παιχνιδιού. Αναγνώριση των ευθυνών κάθε κλάσης και των πιθανών σχέσεών της με άλλες.
- Εύρεση των μεταβλητών στιγμιοτύπων κάθε κλάσης.

- Οργάνωση των κλάσεων σε ιεραρχίες με στόχο την μέγιστη δυνατή επαναχρησιμοποίηση του κώδικα σας όπως π.χ. κοινά μέλη στοιβών φύλλων που μπορούν να σχηματίσουν μία υπερκλάση «συλλογή καρτών» την οποία κληρονομούν στην συνέχεια όλες οι δυνατές εξειδικεύσεις της .
- Εύρεση της συμπεριφοράς (behavior) κάθε κλάσης και διεπαφής του προγράμματός σας, καθώς και της μεταξύ τους επικοινωνίας μέσω μηνυμάτων (method calls). Για κάθε κλάση και διεπαφή δώστε τις υπογραφές (signatures), το είδος (constructors, accessors, transformers) και τις εκ των προτέρων και εκ των υστέρων καθώς και τις αμετάβλητες συνθήκες (preconditions, postconditions, invariants) που διέπουν όλες τις μεθόδους τους.

Φάση 2η - Υλοποίηση

Σε αυτή τη φάση θα πρέπει να γίνει η κυρίως υλοποίηση της εφαρμογής, βάσει της σχεδίασης που έχει προηγηθεί (φάση 1). Μολονότι δεν επιβάλλεται να χρησιμοποιηθεί αυτούσια η σχεδίαση της 1ης φάσης, καθότι κάποιες σχεδιαστικές επιλογές αποδεικνύονται στην πορεία άκυρες και χρειάζονται αναθεώρηση, εντούτοις η τελική βαθμολογία θα εξαρτηθεί από την συνέπεια της τελικής υλοποίησης ως προς την αρχική σχεδίαση.

Σε αυτή τη φάση, παραδοτέα είναι :

- ο πηγαίος κώδικας που υλοποιεί το παιχνίδι και η δυνατότητα εκτέλεσής του σαν εφαρμογή και μικροεφαρμογή (Applet) Java.
- αναλυτικές οδηγίες για το πώς μεταγλωττίζεται και πώς τρέχει το πρόγραμμα (README, Makefile κ.λ.π.).
- αναφορά, στην οποία θα αναλύεται :
 - η τελική σχεδίαση της εφαρμογής,
 - ποιες αλλαγές έγιναν σε σχέση με τη σχεδίαση της 1ης φάσης (και γιατί),
 - οι αλγόριθμοι που χρησιμοποιήθηκαν,
 - τα προβλήματα που αντιμετωπίστηκαν,
 - οι σχεδιαστικές ή προγραμματιστικές αποφάσεις που ελήφθησαν και πώς αυτό αντανακλάται στον τελικό χρήστη (π.χ. ευκολία / δυσκολία χειρισμού),
 - ... γενικά ό,τι άλλο κρίνετε απαραίτητο να αναφερθεί.

Βαθμολόγηση της εργασίας

Για την βαθμολογία της εργασίας θα συνεκτιμηθούν:

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

Για περισσότερες διευκρινήσεις σχετικά με την παραπάνω εργασία, μπορείτε να στέλνετε ηλεκτρονικά τις απορίες σας στη λίστα του μαθήματος hy252-list.