



## Γενικές Οδηγίες και Περιγραφή των Παραδοτέων της Εργασίας

### Φάσεις

- **Φάση A: 10% τελικού βαθμού** (ρήτρα 5 για επιτυχία στο μάθημα)
  - ο 26 Νοεμβρίου - 11 Δεκεμβρίου 2021
- **Φάση B: 15% τελικού βαθμού** (ρήτρα 5 για επιτυχία στο μάθημα)
  - ο 12 Δεκεμβρίου - 9 Ιανουαρίου 2022 (bonus 5%) ή
  - ο 12 Δεκεμβρίου - 16 Ιανουαρίου 2022 (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 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 επιτύχετε.

## 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. 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"> <li>• Μέθοδοι για την αρχικοποίηση του παιχνιδιού (ταμπλό, παίκτες, πλακίδια)</li> <li>• Μέθοδος που καθορίζει τη σειρά (ποιος παίκτης παίζει)</li> <li>• Μέθοδοι για την παραλαβή πλακιδίων</li> <li>• Μέθοδος που ελέγχει αν τελείωσε το παιχνίδι/νικητής</li> <li>• Μέθοδος που υπολογίζει το σκορ κάθε παίκτη</li> </ul>	25
<b>Σύνολο</b>	<b>100</b>

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 :
  - ο η τελική σχεδίαση της εφαρμογής,
  - ο ποιές αλλαγές έγιναν σε σχέση με τη σχεδίαση της Α' φάσης (και γιατί),
  - ο οι αλγόριθμοι που χρησιμοποιήθηκαν
  - ο τυχόν διαφοροποιήσεις στους κανόνες σε σχέση με τους κανόνες που δίνονται παραπάνω
  - ο οι σχεδιαστικές και προγραμματιστικές αποφάσεις που ελήφθησαν και πώς αυτό αντανακλάται στον τελικό χρήστη (π.χ. ευκολία/δυσκολία χειρισμού)
  - ο τα προβλήματα που αντιμετωπίστηκαν
  - ο τα JUnit tests που φτιάχτηκαν για τον έλεγχο της ορθότητας
  - ο 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](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 (Ανανεωμένα 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
<b>Λειτουργικότητα-Γραφικά (80μ)</b>	
Αρχικοποίηση ταμπλό και καρτών	10
Τήρηση σειράς	7
Κίνηση πιονιών και ζάρι	10
Σωστή χρήση καρτών και θέσης Μηνύματος	15
Σωστή χρήση καρτών και θέσης Συμφωνίας	7
Χρήση Ειδικών θέσεων του ταμπλό (+ Jackpot, Sunday Football Day, Thursday rule)	16
Σωστή υλοποίηση της θέσης PayDay	5
Χρήση Δανείου	5
Τέλος παιχνιδιού- Νικητής	5
<b>Σύνολο</b>	<b>100</b>

**ΣΗΜΑΝΤΙΚΟ:** Η **ελάχιστη** υλοποίηση που χρειάζεται κάποιος για να πάρει βαθμό 50/100 είναι να υπάρχει γραφικό περιβάλλον, αρχικοποίηση του ταμπλό, των πιονιών και των καρτών και να κινούνται τα πιόνια στο ταμπλό με τη χρήση του ζαριού. Τα παραπάνω προϋποθέτουν πως υπάρχει επικοινωνία της γραφικής διεπαφής με το μοντέλο μέσω του controller

***Καλή Εργασία***



Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών

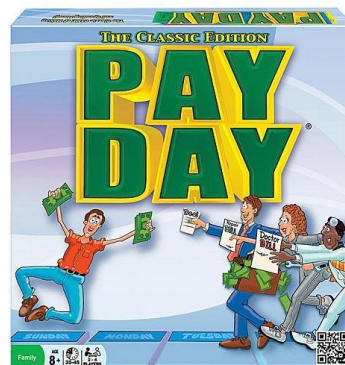
**HY252 – Αντικειμενοστρεφής Προγραμματισμός (Χειμερινό εξάμηνο 2021-2022)**

Διδάσκων: Γιάννης Τζίτζικας

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

Βοηθοί: Μάνος Σμυρνάκης και οι λοιποί βοηθοί

## Project



### 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 κλασσικό παιχνίδι **Ημέρα Πληρωμής** με... ελληνικά στοιχεία.

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

### **Περιγραφή και Σκοπός του Παιχνιδιού.**

Το παιχνίδι που θα υλοποιήσετε αποτελείται από 2 παίκτες και το ταμπλό του παιχνιδιού θυμίζει ένα ημερολόγιο όπου κάθε θέση στο ταμπλό αναπαριστά μία ημερομηνία. Σε κάθε θέση (ημερομηνία) του ταμπλό, ο παίκτης μπορεί να κερδίσει χρήματα (π.χ., να πληρωθεί από τη δουλειά του) ή να ξοδέψει χρήματα (π.χ., λόγω κάποιου λογαριασμού της ΔΕΗ). Ο σκοπός του παιχνιδιού είναι να καταφέρει ο παίκτης να έχει περισσότερα χρήματα από τον αντίπαλο του στο τέλος του παιχνιδιού.

## **Αναλυτική Περιγραφή της Εργασίας**

### **Περιεχόμενα παιχνιδιού**

Τα παιχνίδια αποτελούνται από :

- 2 παίκτες
- 1 ταμπλό με διαστάσεις 5x7 (5 σειρές-7 στήλες)
- 1 εξάεδρο ζάρι
- 1 στοίβα με 48 κάρτες μηνυμάτων
- 1 στοίβα με 20 κάρτες συμφωνίας

### **Προετοιμασία του παιχνιδιού**

Η αρχικοποίηση του παιχνιδιού περιλαμβάνει

- την επιλογή του ποιος παίκτης θα παίξει πρώτος (random),
- πόσους "μήνες" θα διαρκέσει το παιχνίδι (από 1 έως 3)
- την αρχικοποίηση των 31 θέσεων και της θέσης jackpot,
- την αρχικοποίηση και το ανακάτεμα των καρτών
- την αρχικοποίηση των παικτών
- την πληρωμή 3500 ευρώ σε κάθε παίκτη



Εικόνα 1

## Ενέργειες που γίνονται στη σειρά κάθε παίκτη

Στη σειρά του κάθε παίκτης:

- Πρέπει να ρίξει το ζάρι και να μετακινηθεί όσες μέρες έδειξε το ζάρι στο ημερολόγιο
- Μπορεί να πάρει δάνειο που πρέπει να είναι πολλαπλάσιο των 1000 ευρώ.
- Να ακολουθήσει τις οδηγίες που περιέχει η θέση στην οποία πήγε  
Η σειρά του παίκτη τελειώνει μόνο όταν έχει ολοκληρώσει τις ενέργειες που του ζητήθηκαν

Οπότε για κάθε παίκτη χρειαζόμαστε να κρατάμε τρία είδη χρημάτων

- Χρήματα που διαθέτει
- Λογαριασμοί που δεν έχει ξεπληρώσει
- Δάνειο που έχει πάρει και δεν έχει ξεπληρώσει

Επίσης πρέπει να κρατάμε τις κάρτες συμφωνίας που έχει αγοράσει.

Παρακάτω εμφανίζονται αναλυτικά οι διαφορετικές κάρτες και οι διαφορετικές θέσεις που υπάρχουν στο ταμπλό και ποιες ενέργειες πρέπει να γίνουν σε κάθε περίπτωση.



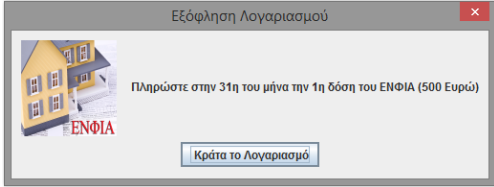
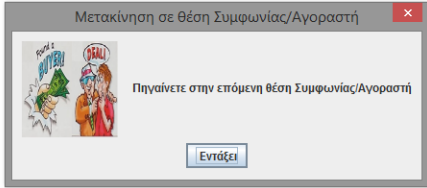
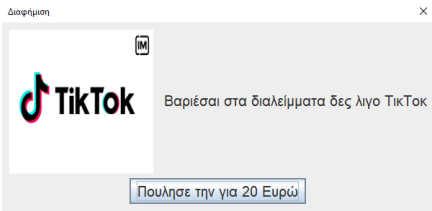
## Κάρτες του Παιχνιδιού

### Κάρτες Μηνυμάτων (48 Κάρτες)



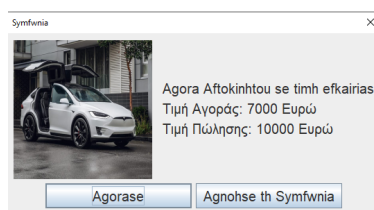
Κάρτα	Παράδειγμα
<p><b>Πλήρωσε τον γείτονα (Pay the Neighbor) - 8 κάρτες</b></p> <p>Με αυτήν την κάρτα πρέπει ο παίκτης να πληρώσει άμεσα στον αντίπαλο το ποσό που αναγράφεται εκεί. Αν χρειαστεί πρέπει να πάρει και δάνειο.</p>	
<p><b>Πάρε λεφτά από το γείτονα - 8 κάρτες</b></p> <p>Με αυτήν την κάρτα πρέπει ο παίκτης να πάρει άμεσα από τον αντίπαλο του το ποσό που αναγράφεται στην κάρτα. Αν χρειαστεί πρέπει να πάρει ο αντίπαλος δάνειο.</p>	
<p><b>Φιλανθρωπία (Charity) - 8 κάρτες</b></p> <p>Με αυτήν την κάρτα το ποσό που αναγράφεται δίνεται στο Jackpot. Αν χρειαστεί πρέπει να πάρει ο παίκτης και δάνειο.</p>	
<p><b>Εξόφληση Λογαριασμού (Bill) - 8 κάρτες</b></p> <p>Με αυτήν την κάρτα ο παίκτης πληρώνει στην τράπεζα το ποσό που αναγράφεται</p>	



<p>στην κάρτα στο τέλος του μήνα. Η κάρτα μένει στην κατοχή του μέχρι να την πληρώσει.</p>	
<p><b>Μετακίνηση στην πλησιέστερη θέση Συμφωνίας/Αγοραστή. (Move to Deal/Buyer) - 8 κάρτες</b></p> <p>Με αυτήν την κάρτα πάς κατευθείαν στην πλησιέστερη θέση Συμφωνίας/Αγοραστή .</p>	
<p><b>Διαφήμιση (Advertisement) - 8 κάρτες</b></p> <p>Αυτή η κάρτα περιέχει μία διαφήμιση και μπορείς να την πουλήσεις για 20 ευρώ.</p>	

Για τις κάρτες αυτές σας δίνεται μία κλάση όπου κάποιος διαβάζει τις 56 κάρτες από ένα αρχείο και τις εμφανίζει σε γραφικό περιβάλλον. Μπορείτε να αλλάξετε αν θέλετε τα μηνύματα και να βάλετε τα δικά σας :)

### Κάρτες Συμφωνίας (20 Κάρτες)



Αυτές οι κάρτες είναι προαιρετικές, δηλαδή ένας παίκτης μπορεί να αγοράσει ένα προϊόν μόνο εάν το επιθυμεί. Κάθε κάρτα έχει δύο τιμές, την τιμή αγοράς και την τιμή πώλησης. Ο παίκτης μπορεί να αγοράσει ένα προϊόν στην τιμή αγοράς μέσω μιας τέτοιας κάρτας. Έπειτα, αν είναι τυχερός και πέσει σε θέση Αγοραστή(Buyer), μπορεί να πουλήσει το προϊόν αυτό στην τιμή πώλησης. Αν όμως δεν καταφέρει να πουλήσει ένα ή παραπάνω προϊόντα από αυτά που έχει αγοράσει, τότε δε θα κερδίσει κάτι από αυτές τις κάρτες. Αν χρειαστεί να πληρώσει και δεν έχει το ποσό πρέπει να πάρει δάνειο.

Για τις κάρτες αυτές σας δίνεται μία κλάση όπου κάποιος διαβάσει τις 20 κάρτες από ένα αρχείο και τις εμφανίζει σε γραφικό περιβάλλον. Μπορείτε να αλλάξετε αν θέλετε τα μηνύματα και να βάλετε τα δικά σας :)

## Θέσεις στο Ταμπλό

### Θέση μηνύματος (8 θέσεις)

Σε αυτήν την περίπτωση, πρέπει ο παίκτης να τραβήξει τον αριθμό των καρτών που αναγράφεται στη θέση αυτή. Σε περίπτωση που η ενέργεια πρέπει να γίνει άμεσα, ο παίκτης πληρώνει/πληρώνεται αμέσως. Αλλιώς τα χρήματα μεταφέρονται στους λογαριασμούς που δεν έχει ξεπληρώσει. Στις περιπτώσεις που δε χρειάζεται η κάρτα να μείνει στην κατοχή του παίκτη τοποθετείται στη στοίβα απόρριψης. Αλλιώς, κρατείται μέχρι να την ξεπληρώσει ( στην περίπτωση μόνο της κάρτας λογαριασμού)

Θα πρέπει να φτιάξετε 4 θέσεις με την κάρτα ενός μηνύματος και 4 θέσεις με την κάρτα 2 μηνυμάτων.

Σε περίπτωση που πρέπει να πάρετε 2 κάρτες και η πρώτη είναι η 'Μετακίνηση στην πλησιέστερη θέση Συμφωνίας/Αγοραστή', διαβάστε και την δεύτερη και μετά πηγαίνετε στη θέση που σας λείει η κάρτα.

Σε περίπτωση που τελειώσουν οι κάρτες, χρησιμοποιείται ξανά αυτές που έχουν απορριφθεί (τις ανακατεύεται ξανά).



### Θέση Συμφωνίας (Deal) (5 Θέσεις)

Σε αυτήν την περίπτωση, πρέπει ο παίκτης να τραβήξει μία κάρτα τύπου Συμφωνίας (Deal). Έπειτα, αποφασίζει εάν θα αγοράσει το προϊόν ή όχι. Σε περίπτωση που το αγοράσει, η κάρτα κρατείται από τον παίκτη, αλλιώς απορρίπτεται. Ο παίκτης μπορεί οποιαδήποτε στιγμή να δει ποιες κάρτες έχει στην κατοχή του.

Σε περίπτωση που τελειώσουν οι κάρτες, χρησιμοποιείται ξανά αυτές που έχουν απορριφθεί (τις ανακατεύεται ξανά).



### Θέση Λαχείο (SWEEPSTAKES) (2 θέσεις)

Αν πέσει ο παίκτης σε αυτή θέση, τότε ρίχνει ξανά το ζάρι και κερδίζει 1000 Χ τον αριθμό που έφερε με το ζάρι σε ευρώ.



### Θέση Λοταρίας (3 θέσεις)

Αν πέσει ο παίκτης σε αυτή θέση, τότε η τράπεζα προσφέρει 1000 ευρώ σε ένα μόνο παίκτη μέσω κλήρωσης. Κάθε παίκτης διαλέγει έναν αριθμό από το 1-6, ξεκινώντας από τον παίκτη που βρίσκεται σε αυτό το τετράγωνο. Έπειτα γίνεται κλήρωση με το ζάρι μέχρι να έρθει ένας από τους 2 αριθμούς που επιλέχθηκαν. Μόλις γίνει αυτό, ο παίκτης που κερδίζει παίρνει το ποσό.



### Θέση Διαγωνισμός στο Ραδιόφωνο (2 θέσεις)

Αν πέσει ο παίκτης σε αυτή θέση, τότε η τράπεζα προσφέρει 1000 ευρώ στον παίκτη που θα φέρει τον μεγαλύτερο αριθμό. Σε περίπτωση ισοπαλίας, οι παίκτες ρίχνουν το ζάρι ξανά. Πρώτος ξεκινάει να ρίχνει το ζάρι ο παίκτης που πήγε σε αυτό το τετράγωνο.



### Θέση Αγοραστής (Buyer) (6 θέσεις)

Σε αυτή τη θέση, κάποιος πρέπει να πουλήσει ένα μόνο προϊόν που έχει αγοράσει πριν μέσω μίας κάρτας συμφωνίας. Οπότε αν έχει ένα ή παραπάνω προϊόντα, πρέπει να του δίνεται η επιλογή να επιλέξει ποιο θέλει να πουλήσει. Αν το κάνει αυτό, μεταφέρονται τα χρήματα στο λογαριασμό του και η κάρτα φεύγει από την κατοχή του παίκτη.



### Θέση Βραδιά Οικογενειακού Καζίνο (Family Casino Night) (2 θέσεις)

Εάν ο παίκτης έφτασε σε αυτή τη θέση φέρνοντας περιττό αριθμό στο ζάρι (1, 3 ή 5) τότε δίνει 500 ευρώ στο Jackpot.

Όμως, εάν ο παίκτης έφτασε σε αυτή τη θέση φέρνοντας άρτιο αριθμό στο ζάρι (2, 4 ή 6) τότε παίρνει 500 ευρώ από την τράπεζα.



## Θέση Αγορά με Έκπτωση - YARD SALE (2 θέσεις)

Ο παίκτης ρίχνει το ζάρι ξανά και πληρώνει την τράπεζα 100 x τον αριθμό που έφερε σε ευρώ. Έπειτα παίρνει την πρώτη κάρτα που βρίσκεται στη στοίβα των καρτών συμφωνίας χωρίς να πληρώσει άλλα χρήματα.



## Θέση Ημέρα Πληρωμής- PAY DAY (1 θέση-τελευταία)

Εδώ ο παίκτης είναι αναγκασμένος να σταματήσει ακόμα και αν το ζάρι έδειξε παραπάνω. Έπειτα κάνει τα παρακάτω.

1. Παίρνει μισθό 3500 ευρώ από την τράπεζα.
2. Πληρώνει τους λογαριασμούς που δεν έχει πληρώσει και απορρίπτει τις κάρτες αυτές.
3. Για το δάνειο που δεν έχει ξεπληρώσει αυτή τη στιγμή, πληρώνει άμεσα φόρο 10%. Για παράδειγμα αν έχει πάρει δάνειο 2000 ευρώ, πρέπει να πληρώσει άμεσα 200 ευρώ. Αν δεν έχει τα λεφτά πρέπει να πάρει και άλλο δάνειο.
4. Εκτός από το φόρο, ο παίκτης μπορεί να πληρώσει το δάνειο ή μέρος του δανείου. Πρέπει το ποσό που θα πληρώσει να είναι πολλαπλάσιο του 1000. Σε περίπτωση που δε πληρώσει το δάνειο ή μέρος του δανείου, το αρνητικό είναι ότι για τα χρήματα που δεν έχει ξεπληρώσει θα πρέπει να πληρώσει ξανά φόρο και τον επόμενο μήνα (ή τους επόμενους μήνες).
5. Αν είναι ο τελευταίος μήνας τότε ο παίκτης δε παίζει ξανά και τοποθετεί όλες τις κάρτες συμφωνίας που έχει στη στοίβα απόρριψης.
6. Αν δεν είναι ο τελευταίος μήνας, τότε το πιόνι του παίκτη πάει ξανά στην αρχή.



Η τοποθέτηση των θέσεων θα πρέπει να γίνεται random, απλώς θα πρέπει πάντα στην τελευταία θέση να είναι η θέση payday.

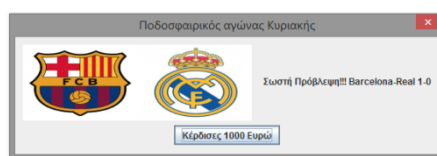
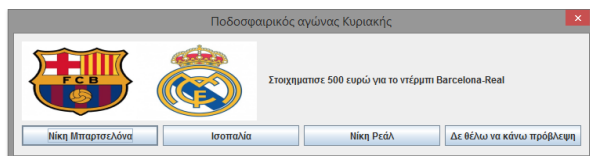
## Jackpot



Όποιος φέρει 6 στη σειρά του (όχι όμως όταν ρίχνει το ζάρι για διαγωνισμούς, λοταρίες, λαχείο κλπ.), παίρνει όλο το ποσό που βρίσκεται στο Jackpot εκείνη τη στιγμή! Έπειτα, συνεχίζει κανονικά τις ενέργειες που πρέπει να κάνει ανάλογα με τη θέση που θα πάει.

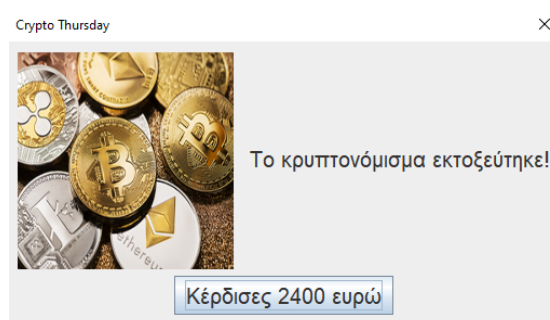
## Ποδοσφαιρικός Αγώνας Κυριακής - Sunday Football Day

Όποιος πάει σε θέση όπου είναι Κυριακή, έχει τη δυνατότητα να στοιχηματίσει αν θέλει 500 ευρώ σε ένα αγώνα ποδοσφαίρου και υπάρχουν 3 ενδεχόμενα α) νίκη γηπεδούχου β) ισοπαλία γ) νίκη φιλοξενούμενου. Ανάλογα με το τι θα φέρει το ζάρι και με την επιλογή του παίκτη μπορεί να διπλασιάσει ή να χάσει τα χρήματα που πόνταρε. Ο παίκτης διπλασιάζει τα χρήματα που πόνταρε εάν α) στοιχημάτισε σε νίκη γηπεδούχου και το ζάρι έφερε 1 ή 2 β) στοιχημάτισε σε ισοπαλία και το ζάρι έφερε 3 ή 4 στη σειρά και γ) στοιχημάτισε σε νίκη φιλοξενούμενου και το ζάρι έφερε 5 ή 6. Σε κάθε άλλη περίπτωση χάνει τα χρήματα του. Μπορείτε να θεωρήσετε ότι η κλήρωση γίνεται μέσα στο πρόγραμμα σας (χωρίς να χρειαστεί να ρίξει ο παίκτης το ζάρι), οπότε αυτόματα ενημερώνεται για το εάν κέρδισε ή όχι.

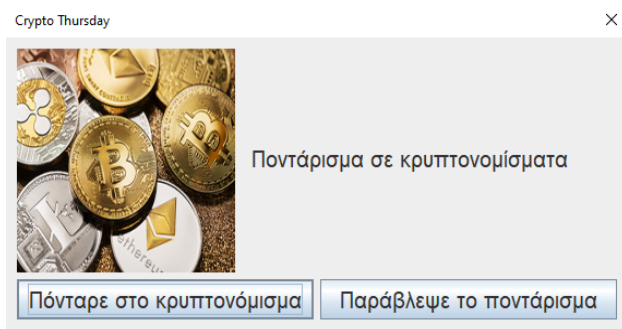


## Άνοδος αξίας κρυπτονομισμάτων – Thursday-Rise in the value of cryptos

Όποιος πάει σε θέση όπου είναι Πέμπτη, έχει τη δυνατότητα να στοιχηματίσει αν θέλει 300 ευρώ σε ένα κρυπτονόμισμα και υπάρχουν 3 ενδεχόμενα α) πτώση αξίας β) σταθεροποίηση αξίας γ) άνοδος αξίας. Ανάλογα με το τι θα φέρει το ζάρι και με την επιλογή του παίκτη μπορεί να διπλασιάσει, να χάσει τα χρήματα που πόνταρε ή να τα πάρει πίσω. Ο παίκτης χάνει τα χρήματα που πόνταρε εάν α) με το ζάρι έφερε 1 ή 2 β) παίρνει τα λεφτά του πίσω αν με το ζάρι έφερε 3 ή 4 στη σειρά και γ) κερδίζει τα διπλά αν με το ζάρι έφερε 5 ή 6. Μπορείτε να θεωρήσετε ότι η κλήρωση γίνεται μέσα στο πρόγραμμα σας (χωρίς να χρειαστεί να ρίξει ο παίκτης το ζάρι), οπότε αυτόματα ενημερώνεται για το εάν κέρδισε ή όχι.







## Περί Δανείων (και βαθμολόγησης)

Η χρήση του δανείου παίρνει 5% του βαθμού. Σε περίπτωση που κάποιος δε το υλοποιήσει, τότε θεωρούμε πως ο παίκτης μπορεί να έχει είτε θετικό είτε αρνητικό ποσό στα χρήματα του. Σε κάθε περίπτωση, οι λογαριασμοί πληρώνονται στο τέλος του μήνα.

## Τέλος του παιχνιδιού

Νικητής του παιχνιδιού είναι ο παίκτης που θα έχει τα περισσότερα χρήματα όταν θα έχουν φτάσει στη θέση payday του τελευταίου μήνα όλοι οι παίκτες. Από το ποσό αφαιρούνται τα χρήματα των δανείων. Ακόμα και σε περίπτωση αρνητικού ποσού, νικητής είναι αυτός που έχει το μικρότερο χρέος.

## Διευκρινίσεις

- ✓ Όταν ο παίκτης πρέπει να πάρει δάνειο για να πληρώσει άμεσα, τότε αυτό μπορεί να γίνεται αυτόματα μέσω του κώδικα και να μη χρειάζεται κάποια είσοδος για τον παίκτη που θα πληρώσει. Για παράδειγμα, έστω ότι πρέπει ο αντίπαλος να δώσει στον άλλο παίκτη 1500 ευρώ και δεν έχει καθόλου χρήματα. Τότε, μέσω του κώδικα θα πάρει δάνειο 2000 ευρώ και θα δώσει στον αντίπαλο τα 1500. Φυσικά, θα πρέπει να αλλάξουν οι μεταβλητές στον κώδικα και στο γραφικό περιβάλλον και να τυπώνεται μήνυμα ότι έγινε αυτή η συναλλαγή.
- ✓ Αν ο ένας παίκτης έχει τελειώσει το παιχνίδι και ο άλλος όχι, τότε ο άλλος συνεχίζει κανονικά μέχρι να φτάσει στο τέλος του τελευταίου μήνα. Επίσης, ο παίκτης που έχει τελειώσει είναι υποχρεωμένος να παίζει σε διαγωνισμούς και να πληρώνει τον αντίπαλο.
- ✓ Αν έρθει η κάρτα "Μετακίνηση στην Πλησιέστερη θέση Συμφωνίας/Αγοραστή" και μέχρι το τέλος του μήνα δεν υπάρχει άλλη διαθέσιμη τέτοια θέση, τότε μπορείτε να την αγνοήσετε. Εναλλακτικά, μπορείτε στη θέση 30 να έχετε πάντα μία θέση Αγοραστή ή Συμφωνίας.

## Bonus (5%)

Δυνατότητα αποθήκευσης του παιχνιδιού και μετέπειτα συνέχισης του σε επόμενο στιγμιότυπο εκτέλεσης της εφαρμογής. Συγκεκριμένα, θα πρέπει να δίνεται στον χρήστη μέσω του μενού μια επιλογή αποθήκευσης και εξόδου από την εφαρμογή. Αυτή η επιλογή θα έχει ως

αποτέλεσμα την μόνιμη αποθήκευση (persistent storage) της κατάστασης (state) του παιχνιδιού εκείνη την 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.

***Εναλλακτικό Bonus Παίξε με τον υπολογιστή.***

## **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 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). Πιο συγκεκριμένα το Model μπορεί να θεωρηθεί ότι αποτελείται από οτιδήποτε σχετίζεται με τα δεδομένα του παιχνιδιού. Υπό αυτή την έννοια τα «Πιόνια», το «ταμπλό», οι «Παίχτες» 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 πρόγραμμά 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 **Position**. Η κλάση αυτή μοντελοποιεί μία θέση του παιχνιδιού και ορίζει τα γνωρίσματα της (μεταβλητές στιγμιότυπων, π.χ., αριθμός στο ταμπλό, μέρα της εβδομάδας) και τις χρήσιμες μεθόδους της (πχ. performAction). Μία πιθανή ιεραρχία είναι η ακόλουθη:



Για παράδειγμα οι θέσεις μηνύματος και συμφωνίας μπορούν να βρίσκονται στην κλάση CardPosition και να έχουν τη μέθοδο getCards(Card C), ενώ η κλάση DicePosition να έχει τα πεδία (money, diceNumber) και τη μέθοδο performAction(Player p, int diceNumber). Επίσης, κάποιος μπορεί να εξειδικεύσει την κλάση DicePosition σε BothPlayersDicePosition για τις θέσεις όπου τα χρήματα διεκδικούν όλοι οι παίκτες μέσω του ζαριού (στην εργασία μας 2 παίκτες). Τέλος, μπορεί να έχει ξεχωριστή κλάση για την θέση PayDay η οποία εξειδικεύει την κλάση Position.

Θα μπορούσατε επίσης να σχεδιάσετε τη λειτουργικότητα μιας κάρτας σαν διεπαφή (interface) και όχι ως κλάση. Σε αυτήν 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 σχεδίασή σας την τεχνική του πολυμορφισμού (π.χ. τη μέθοδο action). 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 performAction διαφέρει από θέση σε θέση.

Παρομοίως, μπορείτε να κάνετε ιεραρχία και για τις κάρτες, ξεκινώντας από μία κλάση Card κι επεκτείνοντας την σε MailCard και DealCard. Για παράδειγμα, όλες οι κάρτες έχουν κάποιο αριθμό, κάποια εικόνα κλπ. Όμως, ανάλογα με τον τύπο της κάρτας, υπάρχουν διαφορετικά πεδία, όπως το κόστος αγοράς και πώλησης των καρτών Συμφωνίας.

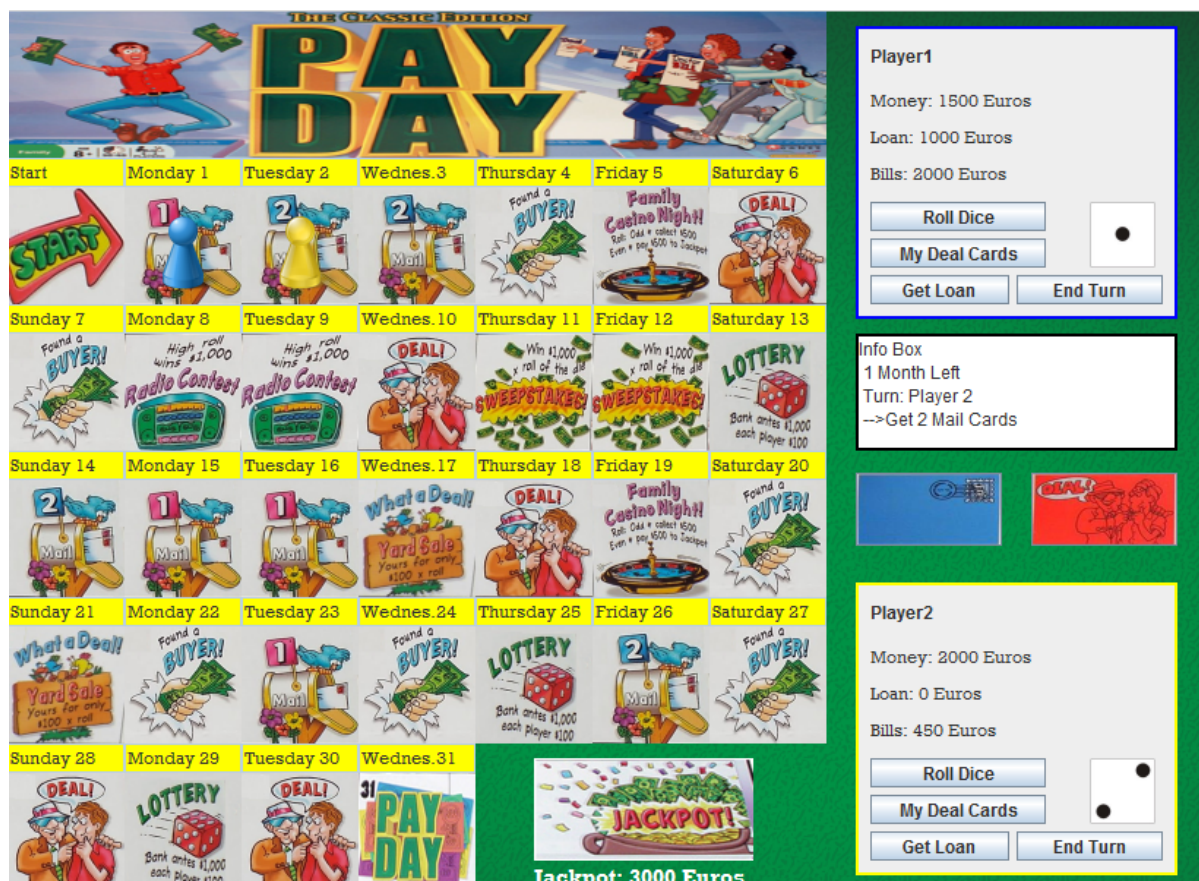
Μια από τις πιο στοιχειώδεις ίσως οντότητες του παιχνιδιού είναι αυτή του **Player**. Όπως υποδηλώνει και το όνομα της, η κλάση αυτή προσομοιώνει τον παίκτη του παιχνιδιού. Κάθε παίκτης διατηρεί χρήματα, λογαριασμούς, κάρτες, θέση στο ταμπλό κλπ.

## Controller

Ο Controller έχει πρόσβαση στα δεδομένα του παιχνιδιού και στην διεπαφή. Από την διεπαφή λαμβάνει πληροφορία για τις κινήσεις των παιχτών 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» 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 παίξουν 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 τη δυνατότητα στους παίκτες να παίξουν ταυτόχρονα το παιχνίδι σε έναν υπολογιστή. Ένα σχεδιαστικό παράδειγμα υλοποίησης της εν λόγω διεπαφής παρουσιάζεται στην Εικόνα 2. Σε αυτήν την εικόνα έχουμε ένα ταμπλό, δεξιά πληροφορίες για τα χρήματα των 2 παικτών, τις στοίβες καρτών, το κουμπί δάνειο και τα χρήματα του jackpot. 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 παιχτών, ήχους (πχ για το ζάρι) 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 DiceButton μπορεί να σας βοηθήσει στον χειρισμό των θέσεων ως Buttons αλλά με τα χαρακτηριστικά που εσείς επιθυμείτε να έχει το ζάρι του παιχνιδιού).

Μπορείτε να χρησιμοποιήσετε την κλάση JLayeredPane, όπου κάποιος μπορεί να τοποθετήσει ένα component (πχ, κουμπί, ετικέτα, κείμενο) πάνω σε ένα άλλο και κάθε component έχει συγκεκριμένες συντεταγμένες. Το ταμπλό του παιχνιδιού, μπορεί να φτιαχτεί είτε με κουμπιά, είτε με JLabels. Σε κάθε περίπτωση, είναι σημαντικό δείχνετε πάντα το που βρίσκονται τα πιόνια στο γραφικό περιβάλλον. Είναι σημαντικό το ζάρι να είναι ένα κουμπί, όπου όταν κάποιος το πατάει να εμφανίζει τυχαία 1 αριθμό από το 1 έως το 6 και να μεταφέρει το πιόνι του παίκτη στη σωστή θέση. Επίσης πρέπει να υπάρχουν τα κουμπιά που φαίνονται στην Εικόνα 2 (πχ. για τις κάρτες, για το δάνειο, κλπ), ενώ για τις κάρτες μπορείτε να χρησιμοποιήσετε Java Dialogs. Εσείς, μπορείτε να χρησιμοποιήσετε οποιοδήποτε τύπο Java Dialogs σας βολεύει για την κάθε μία περίπτωση

κάρτας (δείτε

<http://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html>).

Εναλλακτικά, μπορεί κάποιος να χρησιμοποιήσει το Grid Layout για τις θέσεις, μιας και το ταμπλό του παιχνιδιού ουσιαστικά είναι ένα grid διαστάσεων 5x7.

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

### **Βοηθητικά**

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

**Επίσημες Οδηγίες στα Αγγλικά. Προσοχή, μερικές οδηγίες είναι λίγο διαφορετικές από την εκφώνηση. Εσείς πάντα στηρίζετε στις οδηγίες της εκφώνησης.**

[http://www.gamingcorner.nl/rules/boardgames/payday\\_uk.pdf](http://www.gamingcorner.nl/rules/boardgames/payday_uk.pdf)

**Ενδεικτικό βοήθημα για το MVC (Model View Controller) μοντέλο σε Java**

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

Οι φωτογραφίες για όλα τα περιεχόμενα του παιχνιδιού (κάρτες, θέσεις, πιόνι, ζάρι κλπ) **έχουν ανέβει** στο Elearn!

Επίσης θα σας δοθεί ένα project προηγούμενης χρονιάς ώστε να αποκτήσετε από νωρίς αίσθηση του πως περίπου θα είναι η τελική σας υλοποίηση.

***Καλή Εργασία***