



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Πτυχιακή Εργασία
Προγραμματισμός Παιχνιδιών σε Java



Του φοιτητή
Σουβατζόπουλου Κωνσταντίνου
Αρ. Μητρώου 032271

Επιβλέπων καθηγητής
Αμπατζόγλου Απόστολος

Θεσσαλονίκη 2010

Πρόλογος

Η ανάπτυξη παιχνιδιών είναι μια διαδικασία ανάπτυξης λογισμικού από την οποία το παράγωγο είναι κάποιο παιχνίδι. Είναι ένας αναπτυσσόμενος κλάδος της ανάπτυξης λογισμικού καθώς χρόνο με το χρόνο δημιουργούνται νέες τεχνολογίες και εργαλεία τα οποία υποβοηθούν την εκτέλεση των παιχνιδιών τόσο από άποψης ενδιαφέροντος όσο και από άποψης γραφικών.

Η πτυχιακή εργασία αυτή θα ασχοληθεί με την ανάπτυξη ενός παιχνιδιού ηλεκτρονικού υπολογιστή γραμμένο σε Java. Πιο συγκεκριμένα θα γίνει πλήρης ανάλυση, σχεδίαση, υλοποίηση, καθώς και αξιολόγηση του παιχνιδιού αυτού. Το παιχνίδι, που πρόκειται να αναπτυχθεί, είναι ένα παιχνίδι Poker τύπου Limit Texas Holdem. Πρόκειται για τον πιο διαδεδομένο και δημοφιλή τύπο παιχνιδιού Poker στις μέρες μας, ιδιαίτερα στην Ευρώπη και στις ΗΠΑ. Υπάρχουν, σήμερα, δεκάδες εφαρμογές που επιτρέπουν σε χρήστες να παίζουν Texas Holdem διαδικτυακά, παίζοντας είτε με πραγματικά είτε με εικονικά χρήματα, τα οποία επιτρέπουν σε ανθρώπους που δε θέλουν να τζογάρουν να παίζουν χωρίς να διακινδυνεύουν τα χρήματά τους.

Η Java έχει δώσει λύση στον κλάδο της ανάπτυξης παιχνιδιών, όχι από την μεριά των γραφικών ή από την άποψη δημιουργίας πολύπλοκων, πλούσιων και μεγάλων παιχνιδιών, αλλά από την ευκολία που προσφέρει στην ανάπτυξη απλών και φορητών παιχνιδιών. Για αυτό κιόλας τα περισσότερα παιχνίδια για κινητά πλέον είναι γραμμένα σε Java. Στο Internet, επίσης, μπορεί κανείς να βρει μεγάλη ποικιλία open source παιχνιδιών τα οποία είναι γραμμένα σε Java.

Η επιλογή της συγκεκριμένης πτυχιακής εργασίας ήταν λόγω της προσωπικής έφεσης του συγγραφέα στον προγραμματισμό. Επίσης, ήταν η μεγαλύτερη, μέχρι τώρα, πρόκληση για τον ίδιο λόγω του μεγέθους του λογισμικού καθώς και της προσπάθειας για ανάπτυξη καλής ποιότητας λογισμικού σχεδιάζοντας, δηλαδή, μια καλή μηχανική.

Περίληψη

Η πτυχιακή εργασία αυτή ασχολείται με την ανάλυση, σχεδίαση και ανάπτυξη ενός παιχνιδιού πόκερ στη γλώσσα προγραμματισμού Java.

Η ανάλυση του συστήματος περιλαμβάνει τις εργασίες της ανακάλυψης και καταγραφής των περιπτώσεων χρήσης. Οι περιπτώσεις χρήσης για το σύστημά μας σχεδιάστηκαν σε ένα διάγραμμα και καταγράφηκαν οι τεκμηριώσεις τους. Στη συνέχεια, σχεδιάστηκε το εννοιολογικό μοντέλο με τις έννοιες που απαρτίζουν το σύστημα.

Για τη φάση της σχεδίασης σχεδιάστηκε το διάγραμμα των κλάσεων του συστήματος μαζί με τις ιδιότητες, τις λειτουργίες και τις συσχετίσεις που έχουν μεταξύ τους. Για κάθε περίπτωση χρήσης, που καθορίστηκε στη φάση της ανάλυσης, σχεδιάστηκε ένα διάγραμμα ακολουθίας συστήματος που περιγράφει τα μηνύματα που στέλνουν οι κλάσεις μεταξύ τους προκειμένου την ολοκλήρωση της.

Η φάση της ανάπτυξης περιλαμβάνει τη συγγραφή του κώδικα. Σε αυτό το κείμενο θα δούμε μια παρουσίαση της διαδικασίας αυτής καθώς και μια περιγραφή των λειτουργιών και των ιδιοτήτων των κλάσεων που απαρτίζουν το σύστημα.

Στη συνέχεια, ακολουθεί μια παρουσίαση της αναδόμησης του συστήματος. Η αναδόμηση αυτή αποσκοπεί στη βελτίωση του κώδικα και του έργου ως σύνολο. Η βελτίωση στοχεύει στο να κατασταθεί το έργο πιο εξελίξιμο και ευέλικτο σε αλλαγές.

Τέλος, γίνεται η αξιολόγηση του συστήματος με τη χρήση γνωστών μετρικών αντικειμενοστραφούς σχεδίασης. Οι μετρικές αυτές μας δείχνουν, από διαφορετικές οπτικές γωνίες, τις δυνατότητες και αδυναμίες του συστήματος.

Abstract

The aim of this project is the analysis, design and development of a poker game written in the programming language Java.

The system analysis includes the work of discovery and recording of the use cases. The use cases for our system were designed in a diagram and the documentation was recorded. Then, we designed the domain model with the concepts that are included in the system.

For the design phase, a class diagram was designed for the system along with their attributes, functions and the associations each other share. For each use case, set at the analysis phase, a sequence diagram was designed that describes the messages sent between classes in order to complete the use case.

The development phase includes the writing of the code. In this document we will make a presentation of the process and a description of the functions and properties of the classes that are included in the system.

Then, follows a presentation of the system refactoring process. The refactoring generally aims at improving the code and the project so it becomes more flexible to change.

Finally, we evaluate the system using known object oriented design metrics. The metrics show us different perspectives of the strengths and weaknesses of the system.

Ευρετήριο περιεχομένων

| | |
|--|----|
| Πρόλογος..... | 3 |
| Περίληψη..... | 4 |
| Abstract..... | 5 |
| Ευρετήριο περιεχομένων..... | 6 |
| Ευρετήριο σχημάτων και πινάκων..... | 9 |
| Εισαγωγή..... | 10 |
| Rational Unified Process (RUP)..... | 11 |
| Κεφάλαιο 1 Game Development..... | 14 |
| 1.1 Εισαγωγή..... | 14 |
| 1.2 Η ιστορία του game development..... | 14 |
| 1.3 Η ομάδα ανάπτυξης παιχνιδιού..... | 15 |
| 1.3.1 Ο σχεδιαστής..... | 15 |
| 1.3.2 Ο προγραμματιστής..... | 15 |
| 1.3.3 Σχέσεις σχετικές με τον ήχο..... | 16 |
| 1.3.4 Ο καλλιτέχνης..... | 17 |
| 1.3.5 Ο παραγωγός..... | 17 |
| 1.4 Java και Game development..... | 17 |
| 1.4.1 Java3D..... | 18 |
| 1.4.2 Τρόποι που μπορεί να χρησιμοποιηθεί η Java στην ανάπτυξη παιχνιδιών..... | 19 |
| 1.4.3 Φορητότητα..... | 19 |
| 1.4.4 Εν κατακλείδι..... | 20 |
| 1.5 Το πόκερ..... | 20 |
| 1.5.1 Η ιστορία του πόκερ..... | 21 |
| 1.5.2 Εκδοχές του πόκερ..... | 22 |
| 1.6 Επίλογος..... | 23 |
| Κεφάλαιο 2 Ανάλυση..... | 24 |
| 2.1 Εισαγωγή..... | 24 |
| 2.2 Ανάλυση απαιτήσεων..... | 24 |
| 2.3 Οι κανόνες του Limit Texas Holdem..... | 25 |
| 2.4 Διάγραμμα περιπτώσεων χρήσης..... | 28 |
| 2.4.1 Το διάγραμμα περιπτώσεων χρήσης για το σύστημά μας..... | 29 |
| 2.4.2 Τεκμηρίωση περιπτώσεων χρήσης..... | 29 |
| 2.4.2.1 New Game..... | 29 |
| 2.4.2.2 Play..... | 30 |
| 2.4.2.3 Bet..... | 31 |
| 2.4.2.4 Fold..... | 32 |
| 2.4.2.5 New Round..... | 32 |
| 2.4.2.6 Save Game..... | 33 |
| 2.4.2.7 Load Game..... | 33 |

| | | |
|------------|--|----|
| 2.5 | Εννοιολογικό μοντέλο..... | 34 |
| 2.5.1 | Το εννοιολογικό μοντέλο για το σύστημά μας..... | 35 |
| 2.6 | Επίλογος..... | 36 |
| Κεφάλαιο 3 | Σχεδίαση..... | 37 |
| 3.1 | Εισαγωγή..... | 37 |
| 3.2 | Διάγραμμα κλάσεων..... | 37 |
| 3.2.1 | Το διάγραμμα κλάσεων για το σύστημά μας..... | 38 |
| 3.3 | Διάγραμμα ακολουθίας συστήματος..... | 40 |
| 3.3.1 | Τα διαγράμματα ακολουθίας συστήματος για το σύστημά μας..... | 40 |
| 3.3.1.1 | New Game..... | 41 |
| 3.3.1.2 | Play..... | 42 |
| 3.3.1.3 | Bet..... | 42 |
| 3.3.1.4 | Fold..... | 43 |
| 3.3.1.5 | New Round..... | 44 |
| 3.3.1.6 | Save Game..... | 45 |
| 3.3.1.7 | Load Game..... | 45 |
| 3.4 | Επίλογος..... | 46 |
| Κεφάλαιο 4 | Υλοποίηση..... | 47 |
| 4.1 | Εισαγωγή..... | 47 |
| 4.2 | Εργαλεία που χρησιμοποιήθηκαν..... | 47 |
| 4.3 | Η διεπιφάνεια του χρήστη..... | 48 |
| 4.3.1 | Το αρχικό παράθυρο..... | 48 |
| 4.3.2 | Το παράθυρο έναρξης νέου παιχνιδιού..... | 49 |
| 4.3.3 | Το παράθυρο παιχνιδιού..... | 50 |
| 4.4 | Οι ιδιότητες των κλάσεων..... | 53 |
| 4.4.1 | Game..... | 53 |
| 4.4.2 | Player..... | 53 |
| 4.4.3 | Person..... | 54 |
| 4.4.4 | Computer..... | 54 |
| 4.4.5 | Card..... | 54 |
| 4.4.6 | Deck..... | 54 |
| 4.5 | Οι μέθοδοι των κλάσεων..... | 54 |
| 4.5.1 | Game..... | 54 |
| 4.5.2 | Player..... | 56 |
| 4.5.3 | Person..... | 56 |
| 4.5.4 | Computer..... | 56 |
| 4.5.5 | Card..... | 57 |
| 4.5.6 | Deck..... | 57 |
| 4.6 | Ο υπολογισμός της δύναμης του φύλλου..... | 57 |
| 4.7 | Η τεχνητή νοημοσύνη του παιχνιδιού..... | 58 |
| 4.8 | Επίλογος..... | 58 |
| Κεφάλαιο 5 | Συντήρηση/Επέκταση..... | 60 |
| 5.1 | Εισαγωγή..... | 60 |
| 5.2 | Έκδοση 2.0..... | 60 |
| 5.2.1 | Το διάγραμμα κλάσεων..... | 60 |
| 5.2.2 | Οι ιδιότητες των κλάσεων..... | 62 |
| 5.2.2.1 | Game..... | 62 |

| | |
|--|----|
| 5.2.2.2 Player..... | 62 |
| 5.2.2.3 Person..... | 62 |
| 5.2.2.4 Computer..... | 62 |
| 5.2.2.5 Card και Deck..... | 62 |
| 5.2.2.6 Trick..... | 63 |
| 5.2.2.7 ConfidenceCalculator..... | 63 |
| 5.2.2.8 PlayPolicy..... | 63 |
| 5.2.3 Οι μέθοδοι των κλάσεων..... | 63 |
| 5.2.3.1 Game..... | 63 |
| 5.2.3.2 Player..... | 64 |
| 5.2.3.3 Person..... | 64 |
| 5.2.3.4 Computer..... | 64 |
| 5.2.3.5 Card και Deck..... | 65 |
| 5.2.3.6 Trick..... | 65 |
| 5.2.3.7 ConfidenceCalculator..... | 65 |
| 5.2.3.8 PlayPolicy..... | 65 |
| 5.3 Έκδοση 3.0..... | 65 |
| 5.3.1 Το διάγραμμα κλάσεων..... | 66 |
| 5.3.2 Οι ιδιότητες των κλάσεων..... | 66 |
| 5.3.3 Οι μέθοδοι των κλάσεων..... | 68 |
| 5.4 Επίλογος..... | 68 |
| Κεφάλαιο 6 Αξιολόγηση..... | 70 |
| 6.1 Εισαγωγή..... | 70 |
| 6.2 Μετρικές αντικειμενοστραφούς σχεδίασης..... | 70 |
| 6.2.1 Attribute Hiding Factor (AHF) | 70 |
| 6.2.2 Coupling Between Objects (CBO) | 71 |
| 6.2.3 Cyclomatic Complexity (CC) | 71 |
| 6.2.4 Coupling Factor (CF) | 72 |
| 6.2.5 Lines Of Code (LOC) | 73 |
| 6.2.6 Lack Of Cohesion Of Methods 1 (LOCOM1) | 73 |
| 6.2.7 Polymorphism Factor (PF) | 74 |
| 6.2.8 Response For Class (RFC) | 74 |
| 6.2.9 Weighted Methods Per Class 1 (WMPC1) | 75 |
| 6.3 Πρότυπα σχεδίασης που εφαρμόστηκαν..... | 76 |
| 6.3.1 Έκδοση 1.0..... | 76 |
| 6.3.2 Έκδοση 2.0..... | 76 |
| 6.3.3 Έκδοση 3.0..... | 77 |
| 6.4 Επίλογος..... | 78 |
| Συμπεράσματα..... | 79 |
| Βιβλιογραφία..... | 80 |

Ευρετήριο σχημάτων και εικόνων

| | |
|--|----|
| Σχήμα 2.1 Διάγραμμα περιπτώσεων χρήσης..... | 29 |
| Σχήμα 2.2 Εννοιολογικό μοντέλο..... | 36 |
| Σχήμα 3.1 Διάγραμμα κλάσεων πρώτης έκδοσης..... | 39 |
| Σχήμα 3.2 Διάγραμμα ακολουθία συστήματος New Game..... | 41 |
| Σχήμα 3.3 Διάγραμμα ακολουθία συστήματος Play..... | 42 |
| Σχήμα 3.4 Διάγραμμα ακολουθία συστήματος Bet..... | 43 |
| Σχήμα 3.5 Διάγραμμα ακολουθία συστήματος Fold..... | 43 |
| Σχήμα 3.6 Διάγραμμα ακολουθία συστήματος New Round..... | 44 |
| Σχήμα 3.7 Διάγραμμα ακολουθία συστήματος Save Game..... | 45 |
| Σχήμα 3.8 Διάγραμμα ακολουθία συστήματος Load Game..... | 46 |
| Εικόνα 4.1 Το αρχικό παράθυρο..... | 48 |
| Εικόνα 4.2 Ο επιλογέας αρχείου..... | 49 |
| Εικόνα 4.3 Φόρμα εκκίνησης νέου παιχνιδιού..... | 50 |
| Εικόνα 4.4 Η διεπιφάνεια του παιχνιδιού..... | 51 |
| Εικόνα 4.5 Ο επιλογέας αρχείου για το σώσιμο του παιχνιδιού..... | 51 |
| Εικόνα 4.6 Τα χαρτιά του χρήστη..... | 52 |
| Εικόνα 4.7 Άνοιγμα των κοινών φύλλων..... | 52 |
| Σχήμα 5.1 Το διάγραμμα κλάσεων της δεύτερης έκδοσης..... | 61 |
| Σχήμα 5.2 Το διάγραμμα κλάσεων της τρίτης έκδοσης..... | 67 |
| Σχήμα 6.1 Attribute Hiding Factor..... | 71 |
| Σχήμα 6.2 Coupling Between Objects..... | 71 |
| Σχήμα 6.3 Cyclomatic Complexity / 1..... | 72 |
| Σχήμα 6.4 Cyclomatic Complexity / 2..... | 72 |
| Σχήμα 6.5 Coupling Factor..... | 73 |
| Σχήμα 6.6 Lines Of Code..... | 73 |
| Σχήμα 6.7Lack Of Cohesion Of Methods 1..... | 74 |
| Σχήμα 6.8 Polymorphism Factor..... | 74 |
| Σχήμα 6.9 Response For Class..... | 75 |
| Σχήμα 6.10 Weighted Methods Per Class 1 / 1..... | 75 |
| Σχήμα 6.11 Weighted Methods Per Class 1 / 2..... | 76 |

Εισαγωγή

Σε αυτήν την πτυχιακή εργασία θα αναπτυχθεί ένα παιχνίδι poker για ηλεκτρονικό υπολογιστή το οποίο θα είναι γραμμένο στη γλώσσα προγραμματισμού Java. Πιο συγκεκριμένα, θα γίνει ανάλυση και λεπτομερής σχεδίαση του λογισμικού πριν ξεκινήσει η υλοποίησή του. Στόχος είναι η ανάπτυξη υψηλής ποιότητας λογισμικού, το οποίο θα είναι ευέλικτο, επεκτάσιμο και εξελίξιμο. Κατά τη φάση της ανάλυσης, θα γίνει η εξαγωγή των απαιτήσεων του λογισμικού. Θα γίνει, δηλαδή, μια καταγραφή των λειτουργιών που πρέπει να καλύπτει και τις προδιαγραφές που θα πρέπει να πληρεί. Αυτή η καταγραφή είναι απαραίτητη πριν ξεκινήσει η σχεδίαση γιατί βάση αυτής θα δημιουργηθεί το σχέδιο του λογισμικού. Στη συνέχεια, θα προχωρήσουμε στη λεπτομερή σχεδίαση του λογισμικού χρησιμοποιώντας τα τεχνουργήματα της UML. Μετά το τέλος της φάσης της σχεδίασης, θα ξεκινήσει η ανάπτυξη του κώδικα. Τέλος, αφού ολοκληρωθεί η υλοποίηση και ο έλεγχος της σωστής λειτουργίας του λογισμικού θα γίνει μια γενική αξιολόγηση αυτού, χρησιμοποιώντας μετρικές αντικειμενοστραφούς σχεδίασης και μελέτη για την ύπαρξη προτύπων σχεδίασης. Έτσι, θα δούμε κατά πόσο το σχέδιο και η υλοποίηση του λογισμικού ανταποκρίνονται στα πρότυπα που θέτει η επιστήμη της μηχανικής λογισμικού για ένα σύγχρονο αντικειμενοστραφές πρόγραμμα.

Στο πρώτο κεφάλαιο θα πούμε κάποια πράγματα για το Game Development. Θα γίνει λόγος για κάποιες ιστορικές αναφορές καθώς και για την εξέλιξή του μέχρι σήμερα. Επίσης, σε αυτό το κεφάλαιο θα πούμε μερικά πράγματα για την ιστορία και την εξέλιξη του Poker.

Στο δεύτερο κεφάλαιο, θα γίνει η ανάλυση του συστήματος. Δε θα γίνει πλήρης αντικειμενοστραφής ανάλυση του προς ανάπτυξη λογισμικού καθώς αυτή δε προβλέπεται από τους στόχους της πτυχιακής εργασίας αυτής. Θα περιοριστούμε στην απλή αναφορά των βασικών λειτουργιών του επιθυμητού λογισμικού και μερικών προδιαγραφών που θα πρέπει να πληρεί. Αυτή η διαδικασία ονομάζεται ανάλυση απαιτήσεων. Ακόμα θα δούμε το διάγραμμα περιπτώσεων χρήσης και το εννοιολογικό μοντέλο. Για κάθε περίπτωση χρήσης θα υπάρχει και η τεκμηρίωσή της.

Στο τρίτο κεφάλαιο, θα γίνει η σχεδίαση του προγράμματος. Για τη σχεδίαση αυτή θα χρησιμοποιηθούν διάφορα τεχνουργήματα της UML, όπως το διάγραμμα ακολουθίας συστήματος και το διάγραμμα κλάσεων.

Στο τέταρτο κεφάλαιο, θα γίνει μια περιγραφή της τρίτης φάσης ανάπτυξης του λογισμικού, δηλαδή της γραφής του κώδικα. Θα γίνει μια παρουσίαση της διεπιφάνειας του χρήστη, της διαδικασίας ανάπτυξης του κώδικα, καθώς και μερικών λεπτομερειών αυτού.

Στο πέμπτο κεφάλαιο, θα γίνει μελέτη για την συντήρηση και την πιθανή επέκταση του προγράμματος. Επίσης, θα γίνει παρουσίαση της προσπάθειας βελτίωσης της ποιότητας της πρώτης έκδοσης του λογισμικού, όπου ποιότητα λογισμικού εννοούμε την ευκολία επέκτασης και συντήρησης του, το οποίο φαίνεται μέσα από τις μετρικές αντικειμενοστραφούς ανάλυσης και των προτύπων σχεδίασης.

Στο έκτο και τελευταίο κεφάλαιο, θα γίνει η αξιολόγηση του λογισμικού. Δηλαδή, θα γίνει μια μελέτη για την ύπαρξη προτύπων σχεδίασης, καθώς και μια περιγραφή αυτών σε περίπτωση που υπάρχουν. Επίσης, θα υπολογιστούν οι τιμές των μετρικών αντικειμενοστραφούς σχεδίασης για να βοηθήσουν στην διαδικασία αξιολόγησης.

Rational Unified Process (RUP)

Η RUP είναι μία επαναληπτική διαδικασία ανάπτυξης λογισμικού. Λέγοντας επαναληπτική εννοούμε ότι οι φάσεις ανάπτυξης λογισμικού που ορίζει η RUP επαναλαμβάνονται ξανά και ξανά μέχρι να πετύχουμε το τελικό αποτέλεσμα που θέλουμε.

Η Rational Unified Process είναι ένα προϊόν της διαδικασίας ανάπτυξης λογισμικού που αρχικά αναπτύχθηκε από την Rational Software, τμήμα της IBM. Μέχρι το 1997, η Rational είχε εξαγοράσει της εταιρείες Verdix, Objectory, Requisite, SQA, Performance Awareness, και Pure-Atria. Συνδυάζοντας τις εμπειρίες αυτών των εταιρειών, η Rational εξέδωσε έξι πρακτικές για τη σύγχρονη μηχανική λογισμικού:

- Επαναληπτική ανάπτυξη
- Διαχείριση προδιαγραφών

- Χρησιμοποίηση αρχιτεκτονικής βάσει συστατικού
- Εικονική απεικόνιση μοντέλου
- Επαναληπτική αξιολόγηση
- Έλεγχος αλλαγών

Η RUP έχει χωρίσει τη διάρκεια ζωής του έργου σε τέσσερις φάσεις. Το τέλος του κύκλου ζωής σημαίνει ότι ο στόχος του έχει επιτευχθεί κι ότι μπορούμε να προχωρήσουμε στην επόμενη επανάληψη αν αυτή χρειάζεται. Οι τέσσερις φάσεις του κύκλου ζωής του λογισμικού είναι οι εξής:

- **Η φάση της σύλληψης:** Η φάση αυτή περιλαμβάνει τη μελέτη σχετικά με το κόστος του κύκλου ανάπτυξης και μια εκτίμηση του κέρδους. Γενικά, γίνεται μια προσεκτική μελέτη για να αποφασιστεί αν το έργο αξίζει, συνήθως οικονομικά, να προχωρήσει ή αν θα σταματήσει εδώ κρίνοντας ότι δε θα επιφέρει κάποιο οικονομικό ή άλλο κέρδος. Κατά τη φάση αυτή, θα δημιουργηθούν κάποια μοντέλα και διαγράμματα, όπως ένα αρχικό διάγραμμα περιπτώσεων χρήσης, τα οποία θα υποβοηθήσουν την όλη μελέτη και την εξαγωγή της απόφασης. Επίσης, στη φάση αυτή γίνεται και μια πρώτη εκτίμηση των κινδύνων που κρύβει το έργο.
- **Η φάση της σχεδίασης:** Ο βασικός στόχος αυτής της φάσης είναι η ανίχνευση των βασικών σημείων κινδύνων που μπορεί να εμφανιστούν. Σε αυτή τη φάση, πρέπει να γίνει μελέτη για τα σημεία που μπορεί να εμφανίσουν κινδύνους καθώς και να βρεθούν τρόποι με τους οποίους μπορούν αυτοί να αντιμετωπιστούν. Όταν σε κάποιο σημείο υπάρχει περίπτωση να εμφανιστεί κάποιος κίνδυνος τότε υπάρχουν τρία ενδεχόμενα. Το πρώτο είναι να γίνουν ενέργειες ώστε να αποφύγουμε τον κίνδυνο. Δεύτερον, σε περιπτώσεις όπου ο κίνδυνος δε μπορεί να αποφευχθεί τότε θα πρέπει να βρεθεί τρόπος επίλυσης του προβλήματος και ανάκαμψης από αυτό. Τέλος, σε ακραία περίπτωση που μια επικίνδυνη κατάσταση δεν επιφέρει μόνιμη και ουσιαστική λύση θα πρέπει να γίνουν ενέργειες ώστε ο κίνδυνος αυτός να κάνει τη μικρότερη δυνατή ζημιά. Σε αυτή τη φάση, επίσης, γίνεται η ανάλυση και η σχεδίαση του συστήματος. Είναι το σημείο στο οποίο το έργο και η αρχιτεκτονική του έργου παίρνουν την πρώτη τους μορφή. Στο τέλος της φάσης αυτής πρέπει να περάσουν από συνεδρίαση τα παρακάτω:

- Διάγραμμα περιπτώσεων χρήσης.
 - Μία περιγραφή της αρχιτεκτονικής του συστήματος
 - Μία εκτελέσιμη έκδοση της αρχιτεκτονικής του συστήματος που θα απεικονίζει τις βασικές περιπτώσεις χρήσης.
 - Ένας κατάλογος των κινδύνων
 - Ένα σχέδιο ανάπτυξης για το σύνολο του έργου
 - Πρωτότυπα τα οποία παρουσιάζουν κάθε αναγνωρισμένο τεχνικό ρίσκο.
-
- **Η φάση της κατασκευής:** Στόχος αυτής της φάσης είναι η ανάπτυξη του έργου λογισμικού. Σε αυτή τη φάση κατασκευάζονται τα δομικά στοιχεία, καθώς και άλλα χαρακτηριστικά του συστήματος. Το μεγαλύτερο μέρος της φάσης αυτής περιλαμβάνει τη γραφή του κώδικα. Το παράγωγο αυτής της φάσης είναι μια εκτελέσιμη έκδοση του λογισμικού το οποίο θα πρέπει να συμφωνεί με το σχέδιο και τις προδιαγραφές που έχουν τεθεί.
-
- **Η φάση της μετάβασης:** Στόχος της φάσης αυτής είναι η μετάβαση του έργου από την ανάπτυξη στην παραγωγή. Είναι η φάση κατά την οποία το πρόγραμμα θα φτάσει στον τελικό χρήστη και θα πρέπει να φροντίσουμε αυτό να είναι κατανοητό από τον ίδιο. Οι δραστηριότητες αυτής της φάσης περιλαμβάνουν την κατάρτιση των τελικών χρηστών και των συντηρητών στο σύστημα καθώς και την επικύρωση των προδιαγραφών του συστήματος από τους χρήστες. Στο έργο, επίσης, ελέγχεται το επίπεδο ποιότητάς του με βάση αυτού που καθορίστηκε στη πρώτη φάση.

Κεφάλαιο 1 Game development

1.1 Εισαγωγή

Τα ηλεκτρονικά παιχνίδια έχουν αλλάξει. Ότι κάποτε ήταν απλά περιέργεια τώρα έχει γίνει μια από τους πιο δημοφιλείς τρόπους ψυχαγωγίας και διασκέδασης για όλες τις ηλικίες. Πρόκειται για μια συνεχώς αναπτυσσόμενη βιομηχανία η οποία για να επιβιώσει θα πρέπει συνεχώς να βρίσκονται καινούργιες ιδέες, τεχνολογίες και καινοτόμες δημιουργίες που να μπορούν να διεγείρουν και να κρατούν σταθερό το ενδιαφέρον των “Gamers”.

Σε αυτό το κεφάλαιο, θα μιλήσουμε γενικά για τον κλάδο του game development καθώς και για το πόκερ με το οποίο ασχολείται η συγκεκριμένη πτυχιακή.

1.2 Η ιστορία του game development

Η ιστορία της επιστήμης της ανάπτυξης παιχνιδιών ξεκινάει με την δημιουργία του πρώτου παιχνιδιού το 1952 που λεγόταν ΟΧΟ. Το ΟΧΟ γράφτηκε από τον προγραμματιστή Alexander S. Douglas σαν εργασία για το διδακτορικό του. Το παιχνίδι αυτό είναι η γνωστή σε όλους μας τρίλιζα. Ήταν το πρώτο παιχνίδι με υποτυπώδη ψηφιακά γραφικά που έτρεξε σε ηλεκτρονικό υπολογιστή. Το δεύτερο ηλεκτρονικό παιχνίδι που δημιουργήθηκε το 1958 ήταν ένα παιχνίδι που λεγόταν Tennis for Two, το οποίο για οθόνη χρησιμοποιούσε ένα παλμοσκόπιο. Το 1961 μία ομάδα προγραμματιστών του ινστιτούτου τεχνολογίας της Μασαχουσέτης (MIT) προγραμμάτισε το Spacewars, στο οποίο δύο διαστημόπλοια τα οποία πολεμούν μεταξύ τους προσπαθώντας ταυτόχρονα να αποφύγουν το βαρυτικό πεδίο ενός κοντινού αστεριού.

Ο κλάδος της ανάπτυξης παιχνιδιών άρχισε να ανθίζει όταν δημιουργήθηκαν οι πρώτες κονσόλες ηλεκτρονικών παιχνιδιών. Η πρώτη ήταν η Magnavox Odyssey και δόθηκε στην αγορά το 1972. Τότε, άρχισαν οι εταιρείες να δουλεύουν πάνω στη σχεδίαση και ανάπτυξη ηλεκτρονικών παιχνιδιών. Οι τότε υπάρχουσες κονσόλες και τεχνολογίες δεν είχαν να προσφέρουν πολλές καινοτόμες ιδέες σε ποικιλία παιχνιδιών αφού λίγο ή πολύ πολλά έμοιαζαν μεταξύ τους. Πολλά παιχνίδια προγραμματίζονταν από ερασιτέχνες προγραμματιστές αφού οι περιορισμοί σε τεχνολογίες δεν επέτρεπαν για υψηλή ποιότητα. Έκτοτε η συνεχόμενη ανάπτυξη των γραφικών και επεξεργαστικών δυνατοτήτων σε

υπολογιστές και κονσόλες καθώς και η αύξηση των απαιτήσεων του κοινού κατέστησε αδύνατη την ανάπτυξη ενός παιχνιδιού από έναν μόνο προγραμματιστή, πράγμα που έφερε την ανάγκη για τη δημιουργία ομάδων ανάπτυξης λογισμικού.

Σήμερα, ο κλάδος απαιτεί μεγάλες εταιρείες να ξοδεύουν budget εκατομμυρίων για την ανάπτυξη ενός και μόνο παιχνιδιού καθώς και την απασχόληση μεγάλου ανθρώπινου δυναμικού το οποίο περιλαμβάνει σχεδιαστές, προγραμματιστές, οικονομικούς εκτιμητές και αναρίθμητους άλλους εργαζόμενους.

1.3 Η ομάδα ανάπτυξης παιχνιδιού

Η ανάπτυξη ενός παιχνιδιού είναι μια διαδικασία κατά την οποία συνδυάζονται πολλά στοιχεία, όπως εικόνα, ήχος και κώδικας σε ένα σύνολο, το οποίο είναι το τελικό παιχνίδι, που θα είναι μια ευχάριστη εμπειρία για τον τελικό χρήστη. Μία ομάδα ανάπτυξης παιχνιδιού συνήθως αποτελείται από προγραμματιστές, μουσικούς, σχεδιαστές και άλλους εργαζόμενους απαραίτητους για την ολοκλήρωση του έργου. Παρακάτω θα αναφέρουμε μερικά πράγματα για τις βασικές θέσεις της ομάδας ανάπτυξης.

1.3.1 Ο σχεδιαστής

Κάθε ομάδα ανάπτυξης έχει έναν βασικό σχεδιαστή ο οποίος είναι υπεύθυνος για την ροή του παιχνιδιού. Αυτός καθορίζει πιο θα είναι το σενάριο και πως θα εξελίσσεται το παιχνίδι γενικά. Για παράδειγμα, να επιλέξει πόσες θα είναι οιπίστες ενός παιχνιδιού και τι θα περιλαμβάνει η καθεμιά από αυτές. Στην αρχή θα πρέπει να γράψει κάποια σενάρια σχετικά με την ιστορία του παιχνιδιού και το πώς αυτή θα ξεδιπλώνεται κατά τη διάρκειά του. Ο σχεδιαστής αργότερα θα συντάξει το σχέδιο του παιχνιδιού. Αυτό περιγράφει κάθε σημείο του παιχνιδιού και θα εξελίσσεται κατά τη διάρκεια της ανάπτυξης.

1.3.2 Ο προγραμματιστής

Οι προγραμματιστές παιχνιδιών είναι προγραμματιστές λογισμικού που παίρνουν τις ιδέες, την τέχνη και την μουσική και τα συνδυάζουν όλα σε ένα λογισμικό. Ο προγραμματιστής είναι υπεύθυνος να συνδυάσει τα γραφικά, τη μουσική, το artwork και όλα τα άλλα στοιχεία του έργου σε ένα λογισμικό που να

τρέχει σωστά και να ικανοποιούν τις προσδοκίες όλων. Οι προγραμματιστές, επίσης, πολλές φορές προσπαθούν να γράψουν τον κώδικά τους ώστε να είναι επαναχρησιμοποιήσιμος για μελλοντικά έργα, καθώς επίσης προσπαθούν να τον βελτιώσουν ώστε το πρόγραμμα να είναι όσο το δυνατόν πιο γρήγορο.

Πολλές φορές σε ένα έργο δουλεύουν προγραμματιστές που ειδικεύονται ο καθένας σε διαφορετικό τομέα. Ο προγραμματιστής γραφικών ασχολείται με την αποθήκευση των γραφικών και με τον καθορισμό του τρόπου με τον οποίο αυτά θα εμφανίζονται στην οθόνη. Ο προγραμματιστής τεχνητής νοημοσύνης καθορίζει τον τρόπο με τον οποίο οι αντίπαλοι του παίχτη θα αντιδρούν σε κάποιο συμβάν και γενικά πως θα δρουν ρεαλιστικά. Ο προγραμματιστής ήχου εργάζεται με τους υπεύθυνους για τη μουσική ώστε να δημιουργείται ένα ηχητικό περιβάλλον που να ταιριάζει σε αυτό του παιχνιδιού. Ο προγραμματιστής εργαλείων γράφει λογισμικό που χρησιμοποιείται από τους σχεδιαστές, τους μουσικούς και από άλλα μέλη.

1.3.3 Θέσεις σχετικές με τον ήχο

Η μουσική και τα ηχητικά εφέ είναι σημαντικά σε ένα ηλεκτρονικό παιχνίδι αφού μπορούν να βελτιώσουν κατά πολύ την ποιότητα της εμπειρίας του παιχνιδιού. Δυστυχώς, πολλές φορές τα budget της παραγωγής των παιχνιδιών δεν επιτρέπουν μεγάλη ποικιλία και ποιότητα σε αυτόν τον τομέα. Επίσης, ρίσκο αποτελεί και η ανεπαρκής δουλειά πάνω σε αυτό το μέρος αφού κακή μουσική που δεν ταιριάζει και δεν είναι ευχάριστη για τον τελικό χρήστη μπορεί να απομακρύνει τους αγοραστές. Οι δύο κύριες θέσεις εργασίας σε αυτόν τον τομέα είναι αυτές του μουσικού και του υπεύθυνου ηχητικών εφέ. Ο μουσικός, συνήθως, σε μια ομάδα ανάπτυξης παιχνιδιού έχει την λιγότερη δουλειά. Αυτό φυσικά δε σημαίνει ότι δεν είναι σημαντική. Ο μουσικός είναι υπεύθυνος για την μουσική επένδυση του παιχνιδιού. Ο υπεύθυνος ηχητικών εφέ είναι αυτός που θα επενδύσει το παιχνίδι με εφέ που θα το κάνουν πιο ρεαλιστικό. Για παράδειγμα, ήχους πυροβολισμών για ένα παιχνίδι FPS (First Person Shooter) ή ήχους μηχανών αυτοκινήτου για παιχνίδια τύπου ράλι. Την δουλειά αυτή πολλές φορές μπορεί να την αναθέσουν σε έναν μουσικό για λόγους οικονομίας. Επίσης μπορεί να χρησιμοποιηθούν έτοιμα ηχητικά εφέ από άλλα έργα ή πηγές.

1.3.4 Ο καλλιτέχνης

Οι καλλιτέχνες είναι υπεύθυνοι για τα γραφικά στοιχεία του παιχνιδιού. Συνήθως ειδικεύονται σε συγκεκριμένες δουλειές, όπως τα 3D γραφικά ή 2D artwork. Πολλές φορές έρχονται σε σύγκρουση με τους προγραμματιστές, οι οποίοι θέλουν το παιχνίδι να τρέχει γρήγορα και ομαλά χωρίς να κολάει, ενώ οι καλλιτέχνες θέλουν πιο εμπλουτισμένα γραφικά για να φαίνεται καλή η δουλειά τους. Είναι υπεύθυνοι να σχεδιάζουν γραφικά που να υπακούν στους τεχνολογικούς περιορισμούς που τους δίνονται.

1.3.5 Ο παραγωγός

Ο παραγωγός επιβλέπει όλο το έργο και φροντίζει αυτό να προχωράει ομαλά. Πρέπει να επιβλέπει το έργο ώστε να φροντίσει οι εργασίες να γίνονται εντός χρονοδιαγράμματος και εντός οικονομικού προϋπολογισμού. Πολλές φορές είναι ο τελικός κριτής ανάμεσα σε διαφωνίες που μπορεί να έχουν άλλα μέλη της ομάδας.

1.4 Java και game development

Η Java σαν γλώσσα προγραμματισμού έχει τα τελευταία χρόνια γνωρίζει μεγάλη άνθηση αφού πλέον υπάρχουν επαρκείς διαθέσιμες τεχνολογίες, όπως η Java 3D, η Java for OpenGL, η JInput, ο Sun Server for Online Games. Συμφέρει από άποψη οικονομικού κόστους αφού τα έσοδα ξεπερνούν κατά πολύ τα έξοδα της επένδυσης σε ένα παιχνίδι. Επίσης, η Java είναι γνωστή για τις δυνατότητες που παρέχει για web εφαρμογές. Αυτό της δίνει ένα ακόμα ατού παίρνοντας υπόψη και την άνθιση του online gaming.

Η Java προσφέρει σε υψηλό βαθμό μεγάλη σταθερότητα και αξιοπιστία. Έχει πολύ λίγα bugs σε σύγκριση με άλλες γλώσσες προγραμματισμού. Αυτό έχει διασταυρωθεί από τους εκατομμύρια προγραμματιστές που χρόνια τώρα τη χρησιμοποιούν. Η Sun Microsystems έχει πολλά tutorials τα οποία είναι online διαθέσιμα και βοηθούν πολύ τους προγραμματιστές. Ακόμα στο διαδίκτυο υπάρχει μεγάλη διαθεσιμότητα σε εργαλεία και εφαρμογές, δωρεάν και μη, τα οποία υποβοηθούν την ανάπτυξη παιχνιδιών. Παραδείγματα τέτοιων εργαλείων είναι το NetBeans, το Eclipse καθώς και πολλά ακόμα έτοιμα προγράμματα και πακέτα άλλων προγραμματιστών που τα διαθέτουν δωρεάν. Για όλα αυτά και για άλλους

ακόμα λόγους, η Java είναι μια γλώσσα προγραμματισμού η οποία προσφέρει μεγάλη ευκολία και ευελιξία σε εταιρείες ανάπτυξης παιχνιδιών αλλά και σε αυτόνομους προγραμματιστές που θέλουν να ασχοληθούν με το αντικείμενο.

1.4.1 Java3D

Το ενσωματωμένο πακέτο της Java το Java3D είναι μια επιλογή που παρέχει στους προγραμματιστές τη δυνατότητα υλοποίησης τρισδιάστατων γραφικών. Το Java3D δίνει πρόσβαση σε υλοποιήσεις OpenGL και DirectX που παρέχονται από το λειτουργικό σύστημα. Με το Java3D, ο προγραμματιστής σχεδιάζει μια σκηνή σε δομή δέντρου και δίνει τον έλεγχο της απόδοσής του σε μία βιβλιοθήκη. Τα πλεονεκτήματα της χρήσης του Java3D είναι τα εξής:

- Μειωμένο σημαντικά το κόστος ανάπτυξης. Αφού μεγάλο κομμάτι του κώδικα είναι έτοιμο και γραμμένο στο πακέτο, απαιτείται λιγότερος χρόνος από τον προγραμματιστή να σχεδιάσει τη σκηνή που θέλει.
- Ενσωματωμένη υποστήριξη για πολυνηματικό προγραμματισμό. Το Java3D είναι γραμμένο σε πολυνηματική μορφή ώστε να αξιοποιεί συστήματα πολλών επεξεργαστών.
- Ενσωματωμένη υποστήριξη από δείγματα εικονικής πραγματικότητας.
- Ο κώδικας γίνεται εκατό τις εκατό φορητός μεταξύ DirectX και OpenGL σε πολλαπλές πλατφόρμες.

Τα μειονεκτήματα της χρήσης του Java3D είναι τα εξής:

- Σημαντικά μειωμένη ποιότητα γραφικών. Η ποιότητα των γραφικών του Java3D είναι κατά πολύ μικρότερη αυτής των OpenGL και DirectX.
- Η Java3D δημιουργήθηκε σαν κοινός παρανομαστής στα τρισδιάστατα γραφικά. Αυτό σημαίνει ότι οι τελευταίες τεχνολογίες τρισδιάστατων γραφικών δεν είναι διαθέσιμες στους Java3D προγραμματιστές.
- Είναι συχνό φαινόμενο οι οδηγοί υλικού για τρισδιάστατα γραφικά να παρουσιάζουν bugs. Ακόμα το Java3D θεωρεί ότι οι οδηγοί λειτουργούν σωστά. Οι προγραμματιστές θα πρέπει να δουλέψουν με αυτό σαν δεδομένο και να προσπαθήσουν να βρουν λύσεις.

1.4.2 Τρόποι που μπορεί να χρησιμοποιηθεί η Java σε παιχνίδια

Όταν χρησιμοποιείται η Java στην ανάπτυξη παιχνιδιών υπάρχουν τρεις βασικοί τρόποι που μπορεί να γίνει αυτό:

- Ο πρώτος είναι το παιχνίδι να είναι μία καθαρή εφαρμογή Java. Να χρησιμοποιεί μόνο τεχνολογίες και βιβλιοθήκες που είναι ενσωματωμένες στη Java. Αυτά τα παιχνίδια προσφέρουν μεγάλη ευκολία φορητότητας, αλλά θέτουν και περιορισμούς σε διάφορες περιπτώσεις, όπως όταν χρειάζεται πρόσβαση στο σύστημα.
- Εφαρμογές Java μικτού κώδικα. Αυτές συνδυάζουν κώδικα C++ με κώδικα Java ανάλογα με τις ανάγκες της εφαρμογής. Για παράδειγμα, θα μπορούσαμε να αναθέσουμε όλη τη δημιουργία και χειρισμό των γραφικών σε κώδικα C++ και ο έλεγχος των διαδικασιών να γραφεί σε Java.
- Χρησιμοποίηση της Java σαν script. Αυτό συνήθως σημαίνει ότι το παιχνίδι είναι γραμμένο σε C++ που χρησιμοποιεί scripts γραμμένα σε Java. Αυτό συμβαίνει διότι η Java είναι πολύ γρηγορότερη από άλλα scripts που θα μπορούσε κάποιος να γράψει.

1.4.3 Φορητότητα

Ένας καλός λόγος για να χρησιμοποιήσει κάποιος Java για να γράψει ένα παιχνίδι, είναι η πλήρης φορητότητα των προγραμμάτων γραμμένων σε Java.

Σε υπολογιστές γραφείου, η φορητότητα είναι πλήρης μόνο για τα προγράμματα που είναι αποκλειστικά γραμμένα σε Java. Παρόλα, αυτά όταν κάποιος χρησιμοποιεί κάθε φορά την τελευταία έκδοση της Java η πλήρης φορητότητα είναι διαθέσιμη μόνο για τα λειτουργικά συστήματα Windows, Sun Solaris και Linux. Στα υπόλοιπα θα πρέπει οι αντίστοιχες εταιρείες πρώτα να ολοκληρώσουν τις πόρτες τους. Επιπλέον, σε περιπτώσεις που χρησιμοποιούνται προαιρετικά πακέτα και τοπικές βιβλιοθήκες, τότε θα δουλέψει μόνο σε λειτουργικά συστήματα που έχουν υλοποιηθεί οι αντίστοιχες πόρτες. Η Java δε μπορεί να εγγυηθεί ότι αυτές οι πόρτες υπάρχουν.

Σε κονσόλες παιχνιδιών, ή αλλιώς παιχνιδομηχανές, τα πράγματα είναι λίγο πιο δύσκολα. Η μόνη κονσόλα στην οποία το Java Virtual Machine είναι διαθέσιμο είναι το Sega Dreamcast. Για άλλες μεγάλες διάσημες κονσόλες, όπως το X-Box

και το Sony-Playstation, το Java Virtual Machine δεν είναι διαθέσιμο. Το ότι η φορητότητα θα γινόταν πρόβλημα στην ανάπτυξη παιχνιδιών σε Java θεωρήθηκε κάτι το ανεπίτρεπτο. Αυτό ανάγκασε την Sun Microsystems και την Sony στην ανάπτυξη του Java Game Profile. Αυτό είναι ένα προφίλ για το Java Micro Edition το οποίο επιτρέπει παιχνίδια γραμμένα σε Java να τρέχουν σε κονσόλες.

1.4.4 Εν κατακλείδι

Η Java εφαρμογές είναι ελαφρώς πιο αργές από εφαρμογές γραμμένες σε C++. Το βασικό της πλεονέκτημα όμως είναι ότι παρέχει υψηλά επίπεδα για ατομική δημιουργικότητα. Συμφέρει από άποψη κόστους, για χαμηλού προφίλ παιχνίδια, αφού οι έτοιμες βιβλιοθήκες της Java μειώνουν κατά πολύ τον χρόνο συγγραφής του κώδικα. Για παιχνίδια υψηλού προφίλ, η Java είναι καλύτερο να χρησιμοποιείται όταν δεν απαιτείται μεγάλη απόδοση γραφικών. Αυτό γιατί η Java δεν έχει να προσφέρει πολλά σε τέτοιες περιπτώσεις. Το σημαντικό μειονέκτημα είναι ότι για συγκεκριμένες βιβλιοθήκες σε συγκεκριμένα λειτουργικά συστήματα μερικές πόρτες δεν είναι διαθέσιμες. Επίσης υπάρχει και το θέμα με τις κονσόλες παιχνιδιών. Με την συνεχόμενη όμως ανάπτυξη και διάδοση της Java αυτά τα προβλήματα είναι θέμα χρόνου να εκλείψουν οριστικά αφού οι εταιρείες θα πέμψουν να την υποστηρίξουν.

1.5 Το πόκερ

Το πόκερ είναι μία οικογένεια παιχνιδιών με χαρτιά που βασίζεται σε κανόνες πονταρίσματος αλλά και ισχύος φύλλου στο χέρι. Υπάρχουν διάφοροι τύποι παιχνιδιών πόκερ στα οποία τη διαφορά τη βρίσκουμε στον τρόπο με τον οποίο μοιράζονται τα φύλλα, στη σειρά και τον τρόπο πονταρίσματος και στο πιο φύλλο είναι πιο δυνατό από το άλλο. Σε γενικές γραμμές, οι κοινοί κανόνες είναι ότι κάθε παίχτης πρέπει να ισοφαρίσει το μέγιστο ποντάρισμα για να μείνει στο παιχνίδι ή ακόμα μπορεί να το ανεβάσει προσπαθώντας να βγάλει τους αντιπάλους του από το παιχνίδι. Στο τέλος, νικάει ο παίχτης ο οποίος έχει το καλύτερο φύλλο ή ο παίχτης που είναι ο μόνος που δεν έχει πάει πάσο στην παρτίδα.

1.5.1 Η ιστορία του πόκερ

Η ιστορία του πόκερ είναι ένα θέμα για το οποίο υπάρχουν διάφορες εκδοχές, γνώμες και διαφορές. Η μία εκδοχή υποστηρίζει ότι το πρώτο παιχνίδι, με κοντινούς κανόνες με το πόκερ, που περιέχει χαρτιά, ποντάρισμα και μπλόφα, ήταν το γερμανικό Rochspiel, που εμφανίστηκε τον δέκατο πέμπτο αιώνα. Μία άλλη εκδοχή λέει ότι το πόκερ είναι μεταγενέστερο παιχνίδι του περσικού As Nas. Αυτή είναι και η επικρατέστερη εκδοχή σύμφωνα με ιστορικούς. Υπάρχουν κι άλλα προγενέστερα παιχνίδια που είχαν κοινά στοιχεία με το πόκερ, όπως το γαλλικό roque και το αγγλικό brag, τα οποία περιείχαν μπλόφα. Μια λογική θεωρία λέει ότι απλά το πόκερ επηρεάστηκε από όλα αυτά για να δημιουργηθεί η τελική εκδοχή του.

Άλλοι λένε ότι το πόκερ δεν έχει να κάνει με αυτά αφού παίζεται με 52 φύλλα, ενώ το As Nas με 20, και έχει σαν βάση του το ποντάρισμα. Μία βασική αρχή του πόκερ είναι να χρησιμοποιούνται πάντα και τα 52 φύλλα, ενώ τα άλλα παιχνίδια είχαν κανόνες ώστε να περικόπτονται φύλλα όταν λιγοστεύουν οι παίκτες. Σύμφωνα με αυτήν την εκδοχή η επινοήση του παιχνιδιού χρονολογείται στον δέκατο όγδοο αιώνα σε περιορισμένες περιοχές της Αμερικής και εξαπλώθηκε σε όλη την περιοχή του ποταμού Μισισσιπή μέχρι το 1800.

Σύμφωνα με το βιβλίο «An Exposure of the Arts and Miseries of Gambling» του συγγραφέα Jonathan Green, το παιχνίδι εξαπλώθηκε μέσω των ποταμόπλοιων του Μισισσιπή, όπου τζογαδόροι έπαιζαν μέσα για να περάσουν την ώρα τους στα μεγάλα ταξίδια. Έτσι, το παιχνίδι διαδόθηκε στις γύρω περιοχές του ποταμού και με την πάροδο του χρόνου πέρα από αυτές μέχρι που έφτασε να είναι δημοφιλές σε όλη την χώρα.

Μετά από την εξαπλωση του παιχνιδιού, άρχισαν να δημιουργούνται καινούργιες εκδοχές του παιχνιδιού. Πριν το 1850 είχε επινοηθεί το draw poker. Κατά τη διάρκεια του Αμερικανικού εμφυλίου πολέμου πρωτοεμφανίστηκαν τα stud poker και straight poker. Αργότερα προστέθηκαν κι άλλες μορφές πόκερ στην Αμερική όπως το wildcard γύρω στο 1875, το lowball και το split-pot poker κοντά στο 1900 και το community card poker κάπου στο 1925.

Το πόκερ γνώρισε μεγάλη δημοτικότητα στις αρχές του εικοστού πρώτου αιώνα όταν πρωτοεμφανίστηκε το online-poker. Παιχνίδι πόκερ, δηλαδή, μέσω του διαδικτύου. Ακόμα μεγαλύτερη δημοτικότητα του παιχνιδιού ήρθαν να προσθέσουν μεγάλα τουρνουά πόκερ όπως το αμερικάνικο World Poker Tour

(WPT), το European Poker Tour (EPT) και το Poker Stars, τα οποία μαγνητοσκοπούσαν τηλεοπτικά κανάλια και μπορούσαν να τα παρακολουθήσουν άνθρωποι σε όλο τον κόσμο.

1.5.2 Εκδοχές του πόκερ

Το πόκερ έχει πάρα πολλές εκδοχές. Όλες χρησιμοποιούν το ίδιο πρότυπο παιχνιδιού και πάνω κάτω τις ίδιες ιεραρχίες φύλλων. Οι διαφορές είναι στο μοίρασμα, στο ποντάρισμα και σε κάποιες άλλες λεπτομέρειες που αφορούν τους κανόνες με τους οποίους παίζεται το παιχνίδι. Όλες οι εκδοχές του πόκερ χωρίζονται σε τέσσερις βασικές οικογένειες:

- **Straight Poker.** Πέντε κλειστά φύλλα μοιράζονται σε κάθε παίχτη. Υπάρχει ένας γύρος πονταρίσματος στον οποίο η αύξηση και η κόντρα αύξηση αυτού επιτρέπεται ελεύθερα. Το καλύτερο φύλλο στο χέρι κερδίζει.
- **Stud Poker.** Τα φύλλα μοιράζονται πότε κλειστά και πότε ανοιχτά και ένα τη φορά. Αυτό κανονίζεται από πριν είτε με τη συμφωνία των παιχτών, είτε με την απόφαση αυτού που μοιράζει. Μετά το μοίρασμα του κάθε φύλλου υπάρχει και ένας γύρος πονταρίσματος. Επίσης τα πέντε φύλλα μπορούν να είναι συνδυασμός κλειστών και ανοιχτών χαρτιών.
- **Draw Poker.** Πέντε κλειστά φύλλα μοιράζονται σε κάθε παίχτη. Ακολουθεί ένας γύρος πονταρίσματος και μετά ο κάθε παίχτης μπορεί να αλλάξει από κανένα έως και τέσσερα φύλλα από το χέρι του. Έπειτα υπάρχει ένας ακόμα γύρος πονταρίσματος και μετά κρίνεται ο νικητής.
- **Community Card Poker.** Σε κάθε παίχτη μοιράζεται ένα μη ολοκληρωμένο χέρι κλειστών φύλλων ενώ στο κέντρο του τραπέζιου ανοίγονται επιπλέον φύλλα με τα οποία ο κάθε παίχτης θα συμπληρώσει την πεντάδα του χεριού και θα βγάλει το καλύτερο δυνατόν κόλπο. Οι δύο πιο γνωστές εκδοχές αυτής της οικογένειας είναι το Omaha και το Texas Holdem με το οποίο ασχολείται και αυτή η πτυχιακή εργασία.

Αυτές ήταν οι βασικές οικογένειες παιχνιδιών του πόκερ. Κάθε οικογένεια έχει επιπλέον εκδοχές οι οποίες έχουν η καθεμιά τις δικές τους διαφοροποιήσεις. Υπάρχουν κι άλλες ονομασίες παιχνιδιών που, όμως, διαφέρουν στον τρόπο διεξαγωγής παιχνιδιού και όχι στους κανόνες. Για παράδειγμα, το Video Poker

είναι ένα ηλεκτρονικό παιχνίδι πόκερ όπου ένας παίχτης παίζει ενάντια σε υπολογιστή με τεχνητή νοημοσύνη, όπως το πρόγραμμα αυτής της πτυχιακής εργασίας. Το Red Dog Poker είναι παρόμοιο με το Black Jack όπου παίχτες ποντάρουν ενάντια στη «μάννα» με δύο κλειστά φύλλα στο χέρι στοχεύοντας το επόμενο φύλλο που θα τραβήξουν να είναι ανάμεσα στα δύο πρώτα.

1.6 Επίλογος

Σε αυτό το κεφάλαιο είδαμε ότι ο κλάδος της ανάπτυξης παιχνιδιών είναι συνεχώς αναπτυσσόμενος και εξελίξιμος. Νέες τεχνολογίες, νέες προδιαγραφές προστίθενται συνεχώς σε αυτήν την προσπάθεια πράγμα αναγκαίο αφού και οι απαιτήσεις του κοινού επίσης αυξάνονται και ο ανταγωνισμός στην αγορά είναι μεγάλος. Η Java σαν τεχνολογία ανάπτυξης παιχνιδιών δεν μπορεί να προσφέρει τόσο ώστε να δημιουργηθούν παιχνίδια μεγάλου σκέλους, αλλά προτιμάται από πολλούς για ανάπτυξη παιχνιδιών χαμηλού προφίλ τα οποία σαφώς είναι και αυτά δημοφιλή στο κοινό. Το πόκερ τα τελευταία χρόνια έχει γνωρίσει μεγάλη δημοτικότητα. Υπάρχουν πλέον πάρα πολλές εφαρμογές στο διαδίκτυο που επιτρέπουν στον χρήστη να παίζει με αληθινούς παίχτες από όλο τον κόσμο είτε με εικονικά χρήματα είτε με κανονικά λεφτά. Εταιρείες διεξάγουν διαδικτυακά τουρνουά τα οποία είναι ανοιχτά σε όλους. Η δημοτικότητά του ανέβηκε απότομα στις αρχές του εικοστού πρώτου αιώνα και δείχνει ότι έχει ακόμα να γνωρίσει μεγάλη 'δόξα'.

Στο επόμενο κεφάλαιο αρχίζει η διαδικασία ανάπτυξης του συστήματός μας. Στην πρώτη φάση της ανάπτυξης θα γίνει η ανάλυση του τρέχοντος συστήματος. Για αυτό το έργο, η φάση αυτή, θα περιλαμβάνει την ανάλυση των απαιτήσεων και την καταγραφή και τεκμηρίωση των περιπτώσεων χρήσης.

Κεφάλαιο 2 Ανάλυση

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει η ανάλυση του λογισμικού που θα αναπτύξουμε. Στη διαδικασία της ανάλυσης ενός συστήματος συμπεριλαμβάνονται μια σειρά διαδικασιών και βημάτων όπως η στρατηγική μελέτη, η μελέτη σκοπιμότητας, ο λογικός καθορισμός συστήματος και άλλα, με τη βοήθεια των οποίων στο τέλος παράγουμε μια σειρά μοντέλων και εγγράφων, που μας βοηθούν να ορίσουμε το 'τι' θα κάνει το σύστημα χωρίς να περιγράψουμε το 'πώς'. Η ανάλυση αποσκοπεί στο να γίνει μια προσπάθεια συνεννόησης ανάμεσα στην ομάδα ανάπτυξης και στον πελάτη ώστε να καταλήξουν σε μια κοινή απόφαση σχετικά με τους σκοπούς του προς ανάπτυξη λογισμικού και γενικά με το πώς θα μοιάζει το σύστημα και τι θα πρέπει να κάνει αυτό. Για αυτό το σκοπό, δε χρησιμοποιούνται επιστημονικοί όροι και μοντέλα τα οποία δε θα είναι κατανοητά από τον πελάτη, που ενδεχομένως να είναι άσχετος με την επιστήμη της μηχανικής λογισμικού. Σε αυτήν την πτυχιακή εργασία, η πλήρης και λεπτομερής ανάλυση του λογισμικού δεν είναι μέσα στους σκοπούς της. Εδώ θα ασχοληθούμε μόνο με την ανάλυση απαιτήσεων, την ανακάλυψη και καταγραφή, δηλαδή, της επιθυμητής λειτουργίας και συμπεριφοράς του συστήματος, καθώς και θα δούμε το διάγραμμα περιπτώσεων χρήσης για τις απαιτήσεις αυτές. Στην ανάλυση συστημάτων αυτό είναι μέρος της φυσικής ανάλυσης συστήματος που αποτελεί ένα βήμα του κύκλου της ανάπτυξης συστημάτων.

2.2 Ανάλυση απαιτήσεων

Το παρόν λογισμικό αποτελεί ένα παιχνίδι ηλεκτρονικού υπολογιστή και συγκεκριμένα ένα παιχνίδι Poker τύπου Texas Holdem. Το παιχνίδι αυτό πρέπει να μπορεί να τρέχει σε προσωπικό ηλεκτρονικό υπολογιστή και να δίνει τη δυνατότητα στο χρήστη να παίζει το συγκεκριμένο παιχνίδι.

Ο χρήστης μόλις ξεκινάει ένα νέο παιχνίδι πρέπει να έχει μια φόρμα μπροστά του στην οποία μπορεί να επιλέγει τις διάφορες παραμέτρους του παιχνιδιού. Οι παράμετροι αυτές περιλαμβάνουν τον αριθμό των επιπλέον παιχτών, οι οποίοι δε μπορούν να ξεπερνούν τους πέντε αφού ένα τραπέζι poker μπορεί να έχει μέχρι και έξι παίχτες, τον αριθμό των μαρκών με τον οποίο ξεκινούν

οι παίχτες στα χέρια τους το παιχνίδι και το ποσό των small και big blinds. Αφού ο χρήστης επιλέξει αυτές τις παραμέτρους μπορεί να ξεκινήσει το παιχνίδι.

Ο χρήστης παίζει ενάντια σε παίχτες τους οποίους χειρίζεται το πρόγραμμα και που λειτουργούν με τεχνητή νοημοσύνη. Η τεχνητή νοημοσύνη πρέπει να περιέχει και τεχνικές ώστε οι παίχτες, που χειρίζεται το πρόγραμμα, να δίνουν την εντύπωση πραγματικού παίχτη. Για παράδειγμα, πρέπει ο παίχτης να καθυστερεί τεχνητά να παίρνει αποφάσεις αφού στην πραγματικότητα μπορεί να αποφασίσει για την κίνησή του σε χρόνο μικρότερο από ένα δευτερόλεπτο.

Το παιχνίδι πρέπει να έχει μια διεπιφάνεια φιλική προς το χρήστη με γραφικές εικόνες οι οποίες απεικονίζουν ένα πραγματικό περιβάλλον διεξαγωγής ενός παιχνιδιού poker και κάνουν τη χρήση του προγράμματος πιο ευχάριστη.

Κατά τη διάρκεια ενός παιχνιδιού το πρόγραμμα πρέπει να ακολουθεί πιστά και με ακρίβεια τους κανόνες του Limit Texas Holdem. Οι κανόνες αυτοί είναι καθορισμένοι από αρμόδια αρχή και είναι προσβάσιμοι από το κοινό μέσω του διαδικτύου ή δημοσιευμένων εγχειριδίων και βάση αυτών λειτουργεί το πρόγραμμα.

2.3 Οι κανόνες του Limit Texas Holdem Poker

Το Limit Texas Holdem Poker είναι μία εκδοχή του Texas Holdem Poker της οικογένειας Community Card Poker. Οι κανόνες διεξαγωγής του παιχνιδιού είναι καθορισμένοι και γνωστοί σε όσους ασχολούνται. Αναφέρονται εδώ γιατί στην ανάλυση απαιτήσεων αναφέρεται ότι το πρόγραμμα πρέπει να ακολουθεί τους κανόνες αυτούς.

Ένα παιχνίδι Limit Texas Holdem Poker αποτελείται από ένα τραπέζι στο οποίο συμμετέχουν από δύο έως έξι παίχτες. Κάθε παίχτης έχει στην κατοχή του έναν αριθμό από μάρκες που αντιπροσωπεύουν το χρηματικό ποσό το οποίο μπορεί να ποντάρει κατά τη διάρκεια του παιχνιδιού και που στην ουσία του δίνει τη δυνατότητα να συμμετέχει στις παρτίδες. Σε κάθε παρτίδα ένας παίχτης παίρνει τον ρόλο του dealer, αυτού δηλαδή που μοιράζει τα φύλλα. Ο πρώτος παίχτης στα αριστερά του dealer ποντάρει υποχρεωτικά έναν αριθμό από μάρκες, το λεγόμενο small blind, το οποίο το ποντάρει πριν πάρει φύλλα στα χέρια του. Το ποσό αυτό είναι προκαθορισμένο. Ο επόμενος παίχτης στα αριστερά ποντάρει, κατά την ίδια διαδικασία, τα διπλάσια του προηγούμενου κι αυτός πριν δει τα φύλλα του. Αυτό

είναι το λεγόμενο big blind. Οι παίχτες που ποντάρουν τα blinds αλλάζουν καθώς αλλάζει σε κάθε παρτίδα και ο dealer τη δουλειά του οποίου αναλαμβάνει κάθε παίχτης με τη σειρά, κινούμενοι κατά τη φορά του ρολογιού. Αφού τοποθετηθούν τα blinds ο dealer μοιράζει από δύο φύλλα στον κάθε παίχτη ξεκινώντας από αυτόν που βρίσκεται στα αριστερά του και κινούμενος σύμφωνα με τη φορά του ρολογιού μοιράζοντας ένα ένα τα φύλλα. Στο Limit Texas Holdem Poker χρησιμοποιείται μία πλήρης τράπουλα χωρίς τους μπαλαντέρ, δηλαδή πενήντα δύο φύλλα. Αφού ολοκληρωθεί το μοίρασμα των φύλλων οι παίχτες μπορούν να κοιτάξουν τα φύλλα τους, χωρίς να τα δουν οι αντίπαλοί τους. Σε αυτό το σημείο ξεκινά ο πρώτος γύρος πονταρίσματος. Πρώτος παίρνει λόγο ο παίχτης που βρίσκεται στα αριστερά αυτού που πόνταρε το big blind. Για να συνεχίσει το παιχνίδι πρέπει να ποντάρει τουλάχιστον τόσο όσο είναι το ποσό του big blind, που είναι αυτή τη στιγμή το μεγαλύτερο ποντάρισμα στο παιχνίδι. Μπορεί επίσης να αυξήσει κι άλλο το ποντάρισμα, αλλά η αύξηση αυτή πρέπει να είναι ακριβώς το διπλάσιο του μεγαλύτερου πονταρίσματος που υπάρχει εκείνη τη στιγμή στο παιχνίδι. Σε οποιαδήποτε φάση πονταρίσματος δεν μπορούν να γίνουν πάνω από τέσσερις αυξήσεις στο ποντάρισμα. Αν δεν επιθυμεί να συνεχίσει την παρτίδα γιατί προφανώς δε θεωρεί το φύλλο του αρκετά δυνατό μπορεί να πάει πάσο και να βγει από την παρτίδα. Όλοι οι παίχτες ακολουθούν την ίδια διαδικασία ο καθένας με τη σειρά του κινούμενοι με τη φορά του ρολογιού. Όταν έρθει η σειρά του παίχτη που πόνταρε το big blind και κανένας πριν δεν ανέβασε το ποντάρισμα πάνω από αυτό τότε ο ίδιος μπορεί είτε να ανεβάσει κι άλλο το ποντάρισμα, ή να συμφωνήσει σε αυτό το ποντάρισμα και το παιχνίδι να συνεχιστεί. Το παιχνίδι συνεχίζεται όταν όλοι οι παίχτες, που δεν έχουν πάει πάσο από την παρτίδα, συμφωνήσουν στο ίδιο ποντάρισμα. Ο συνολικός αριθμός των μαρκών που πονταρίστηκε συγκεντρώνεται στη μέση του τραπέζιού και η στοίβα που δημιουργείται λέγεται pot. Οι μάρκες που πάνε στο pot θεωρούνται χαμένες από τους παίχτες και θα τις κερδίσει ο νικητής της παρτίδας. Αν κάποιος αργότερα πάει πάσο χάνει τις μάρκες που πόνταρε μέχρι τώρα. Μόλις συμφωνήσουν, τότε ο dealer βγάζει εκτός παιχνιδιού το πάνω πάνω φύλλο της τράπουλας και ανοίγει στη μέση του τραπέζιού τρία φύλλα, τα οποία είναι ορατά και κοινά για όλους τους παίχτες. Μόλις εμφανιστούν τα τρία πρώτα κοινά φύλλα, το λεγόμενο flop, τότε ξεκινά ο δεύτερος γύρος πονταρίσματος. Η διαδικασία είναι ακριβώς η ίδια με τον πρώτο γύρο μόνο που τώρα δεν υπάρχει κάποιος που να έχει ποντάρει απ' την

αρχή κάποιο blind. Το πρώτο ποντάρισμα πρέπει να είναι όσο το ποσό του big blind. Επίσης, οι παίχτες μπορούν να συμφωνήσουν να συνεχίσουν στην επόμενη φάση της παρτίδας χωρίς να ποντάρει κανείς κάποιο ποσό, πηγαίνοντας δηλαδή όλοι check. Το μικρότερο ανέβασμα πονταρίσματος, ή αλλιώς raise, πρέπει να είναι τουλάχιστον το διπλάσιο του μεγαλύτερου πονταρίσματος αυτή τη στιγμή στο παιχνίδι όπως ακριβώς και στον πρώτο γύρο πονταρίσματος. Όταν οι παίχτες συμφωνήσουν ξανά στο ποντάρισμα ο dealer βγάζει ένα ακόμα φύλλο από την τράπουλα εκτός παιχνιδιού και τοποθετεί ένα τέταρτο φύλλο στα κοινά. Αυτή είναι η φάση του turn. Μετά αρχίζει ο τρίτος γύρος πονταρίσματος η διαδικασία του οποίου είναι ακριβώς η ίδια με τον προηγούμενο. Αφού τελειώσει κι αυτός ο γύρος πονταρίσματος ο dealer βγάζει ένα ακόμη φύλλο από το παιχνίδι και ανοίγει το τελευταίο κοινό φύλλο. Αυτή η φάση λέγεται river. Ακολουθεί ο τελευταίος γύρος πονταρίσματος με την ίδια ακριβώς διαδικασία που ακολουθήθηκε στους προηγούμενους. Σε οποιοδήποτε γύρο πονταρίσματος, αν μείνει ένας παίχτης μόνο που δεν πήγε πάσο, τότε αυτός είναι ο νικητής της παρτίδας και κερδίζει το pot. Στη τελευταία φάση της παρτίδας, οι παίχτες που έχουν παραμείνει ανοίγουν τα κρυφά φύλλα τους και τότε ελέγχεται ποιος έχει τον καλύτερο συνδυασμό στα χέρια του, ο οποίος θα κρίνει και τον νικητή της παρτίδας. Ο συνδυασμός υπολογίζεται χρησιμοποιώντας τα πέντε καλύτερα φύλλα από τα επτά που έχει ο καθένας στη διάθεσή του, δηλαδή από τα δυο κρυφά και τα πέντε κοινά. Οι συνδυασμοί είναι οι παρακάτω ταξινομημένοι από τον δυνατότερο προς τον πιο αδύναμο.

- **Flush Royal:** Ότι και το Straight Flush ξεκινώντας τη σειρά από τον Άσσο.
- **Straight Flush:** Πέντε φύλλα ίδιου σχήματος και τα νούμερα τους στη σειρά.
- **Four of a kind:** Τέσσερα από τα πέντε φύλλα έχουν ίδιο νούμερο.
- **Flush:** Και τα πέντε φύλλα έχουν ίδιο σχήμα.
- **Full House:** Τρία από τα πέντε φύλλα έχουν το ίδιο νούμερο και τα υπόλοιπα δύο επίσης έχουν ίδιο νούμερο.
- **Straight:** Πέντε φύλλα που έχουν τα νούμερα στη σειρά.
- **Three of a kind:** Τρία από τα πέντε φύλλα έχουν το ίδιο νούμερο.
- **Two pairs:** Υπάρχουν δύο ζεύγη φύλλων που έχουν το ίδιο νούμερο.
- **Two of a kind:** Δύο από τα πέντε φύλλα έχουν το ίδιο νούμερο.

2.4 Διάγραμμα περιπτώσεων χρήσης

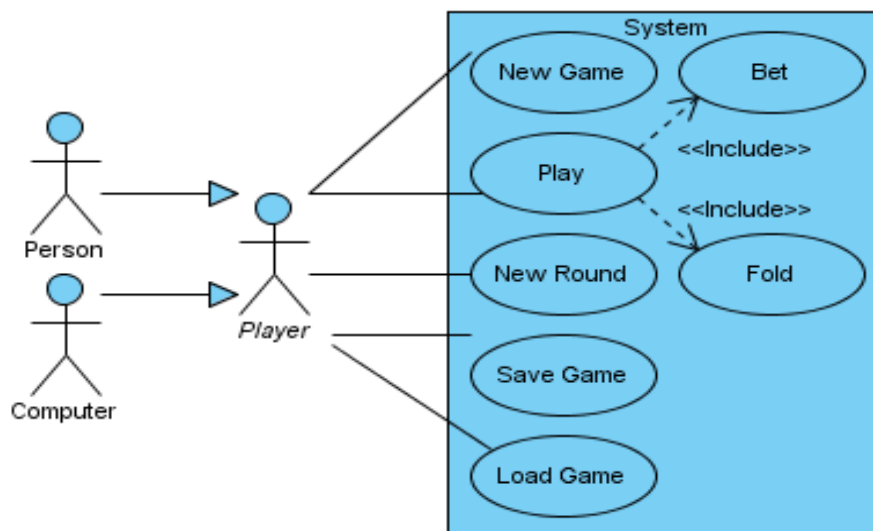
Οι περιπτώσεις χρήσης είναι η κάθε μια μία λειτουργία του συστήματος. Το σύνολό τους συνιστούν τη συμπεριφορά του συστήματος. Το διάγραμμα περιπτώσεων χρήσης χρησιμοποιείται για τη μοντελοποίηση της συμπεριφοράς του συστήματος όπως αυτή γίνεται αντιληπτή από τον εξωτερικό χρήστη. Ο χρήστης απεικονίζεται σαν ένα σχηματικό ανθρωπάκι ,που λέγεται ηθοποιός (actor), ενώ η περίπτωση χρήσης, ως ένα σύνολο λειτουργιών που έχουν κάποιο νόημα για το χρήστη, απεικονίζεται με μία έλλειψη. Η συσχέτιση μεταξύ ενός ηθοποιού και μιας περίπτωσης χρήσης σημαίνει ότι ο συγκεκριμένος ηθοποιός εκτελεί την περίπτωση χρήσης. Αυτή η συσχέτιση απεικονίζεται με μία ευθεία γραμμή που ενώνει τον ηθοποιό με την περίπτωση χρήσης. Εναλλακτικά μπορεί να χρησιμοποιηθεί και κατευθυνόμενη γραμμή με το βελάκι προς τη μεριά της περίπτωσης χρήσης, αλλά πολλές φορές αυτό παραλείπεται για λόγους απλότητας.

2.4.1 Το διάγραμμα περιπτώσεων χρήσης για το σύστημά μας

Για το συγκεκριμένο σύστημα ορίζονται επτά περιπτώσεις χρήσης. Η New Game αφορά την εκκίνηση ενός καινούργιου παιχνιδιού. Η Play εκτελείται όταν έρθει η σειρά του χρήστη για να παίξει. Σε αυτή την περίπτωση χρήσης περιέχονται δύο άλλες περιπτώσεις χρήσης η Bet και η Fold. Η πρώτη εκτελείται όταν ο παίχτης αποφασίζει να θέσει κάποιο ποντάρισμα στο τραπέζι. Η δεύτερη όταν αποφασίζει να πάει πάσο και να βγει από την συγκεκριμένη παρτίδα. Η New Round περίπτωση χρήσης ξεκινάει μια νέα παρτίδα παιχνιδιού όταν η προηγούμενη έχει τελειώσει. Η Save Game σώζει το παιχνίδι σε ένα αρχείο ώστε ο παίχτης να μπορεί να σταματήσει το παιχνίδι στη μέση και συνεχίσει κάποια άλλα στιγμή. Τέλος, Η Load Game φορτώνει ένα προηγουμένως σωσμένο παιχνίδι και συνεχίζει από το σημείο που έμεινε.

Οι ηθοποιοί στο διάγραμμα θα είναι τρεις. Ο Player ο οποίος αναπαριστά έναν παίχτη στο τραπέζι του παιχνιδιού. Οι άλλοι δύο είναι οι Person και Computer που είναι οι επεκτάσεις του Player. Ο Person αναπαριστά τον άνθρωπο παίχτη που παίζει, τον τελικό χρήστη δηλαδή του προγράμματος, ενώ ο Computer

αναπαριστά τον παίχτη που χειρίζεται ο ηλεκτρονικός υπολογιστής που παίζει με τεχνητή νοημοσύνη. Το διάγραμμα περιπτώσεων χρήσης φαίνεται στο παρακάτω σχήμα.



2.1 Διάγραμμα περιπτώσεων χρήσης

2.4.2 Τεκμηρίωση περιπτώσεων χρήσης

Για κάθε περίπτωση χρήσης πρέπει να υπάρχει και η τεκμηρίωσή της. Η τεκμηρίωση αυτή περιγράφει μια σειρά γεγονότων που ακολουθούνται για την επίτευξη της επιθυμητής συμπεριφοράς. Η περιγραφή αυτή θα πρέπει να είναι πλήρης και να εστιάζει στο 'τι' θα πρέπει να κάνει το σύστημα χωρίς να αναφέρει το 'πώς'. Η τεκμηρίωση περιπτώσεων χρήσης γράφεται σε μορφή απλού κειμένου και αναφέρει κατά βάση μια σύντομη περιγραφή της επιθυμητής συμπεριφοράς του συστήματος, τις προ-συνθήκες που πρέπει να ισχύουν πριν την έναρξη της περίπτωσης χρήσης, την κύρια ροή γεγονότων, που περιλαμβάνει μια περιγραφή της ακολουθίας γεγονότων που λαμβάνουν χώρα για την υλοποίηση της επιθυμητής συμπεριφοράς, την εναλλακτική ροή γεγονότων, που αναφέρει εξαιρέσεις και σφάλματα που μπορεί να προκύψουν κατά την εκτέλεση της κύριας ροής γεγονότων και τις μετά-συνθήκες που πρέπει να ισχύουν μετά την ολοκλήρωση της περίπτωσης χρήσης.

Παρακάτω, για κάθε περίπτωση χρήσης του συστήματός μας, καταγράφεται η τεκμηρίωσή της.

2.4.2.1 New Game

Σύντομη περιγραφή

Δημιουργία ενός νέου παιχνιδιού από τον χρήστη.

Προ-συνθήκες

Ο χρήστης έχει στον υπολογιστή του το αρχείο του προγράμματος και έχει εγκατεστημένο το Java Runtime Environment.

Κύρια ροή γεγονότων

1. Ο χρήστης ανοίγει το πρόγραμμα του παιχνιδιού.
2. Ο χρήστης επιλέγει την επιλογή New Game από το user interface.
3. Εμφανίζεται στην οθόνη η φόρμα δημιουργίας νέου παιχνιδιού.
4. Ο χρήστης συμπληρώνει όλα τα απαραίτητα πεδία και πατάει το κουμπί New Game.
5. Εμφανίζεται η διεπιφάνεια του παιχνιδιού και ο χρήστης μπορεί να ξεκινήσει το παιχνίδι.

Εναλλακτική ροή γεγονότων

- 2α. Ο χρήστης επιλέγει Load Game αντί για New Game.
 1. Φόρτωση προηγούμενου παιχνιδιού: include Load Game
- 2β. Ο χρήστης επιλέγει Quit αντί για New Game.
 1. Κλείσιμο του προγράμματος.
- 4α. Ο χρήστης δεν έχει συμπληρώσει όλα τα απαραίτητα πεδία πριν πατήσει το κουμπί Start.
 - 1.Ενημέρωση του χρήστη ότι δεν έχει συμπληρώσει όλα τα πεδία και επανάληψη του βήματος.

Μετά-συνθήκες

Έχει εμφανιστεί στην οθόνη η διεπιφάνεια του παιχνιδιού και ο χρήστης μπορεί να ξεκινήσει να παίζει.

2.4.2.2 Play

Σύντομη περιγραφή

Έρχεται η σειρά του χρήστη να παίζει σε μία παρτίδα.

Προ-Συνθήκες

Έχει ξεκινήσει ένα παιχνίδι, είναι σε εξέλιξη μία παρτίδα, είναι γύρος πονταρίσματος και ο χρήστης έχει σειρά να παίζει.

Κύρια ροή γεγονότων

1. Εμφανίζονται στην οθόνη οι επιλογές που έχει ο χρήστης.

2. Ο χρήστης πρέπει να επιλέξει το ποσό που θα ποντάρει : include Bet

3. Ολοκληρώνεται η σειρά του χρήστη

Εναλλακτική ροή γεγονότων.

2. Ο χρήστης πάει πάσο.

1. Ο χρήστης βγαίνει από το παιχνίδι : include Fold

Μετά-συνθήκες

Έχει τελειώσει η σειρά του χρήστη και περιμένει μέχρι να ξαναέρθει. Παίζει ο επόμενος παίχτης.

2.4.2.3 Bet

Σύντομη περιγραφή

Ο χρήστης θέλει να ποντάρει ένα ποσό στη σειρά του κατά τη διάρκεια κάποιου γύρου πονταρίσματος.

Προ-συνθήκες

Είναι η σειρά του χρήστη να παίξει.

Κύρια ροή γεγονότων

1. Το σύστημα ενημερώνει τον χρήστη πιο είναι το ελάχιστο ποσό που πρέπει να ποντάρει ο χρήστης ώστε να συνεχίσει και πιο είναι το ελάχιστο ποσό που πρέπει να ποντάρει αν θέλει να ανεβάσει το ποντάρισμα.

2. Ο χρήστης επιλέγει το ποσό που θέλει να ποντάρει και πατάει το κουμπί Bet.

3. Το ποσό του πονταρίσματος αφαιρείται από το σύνολο των μαρκών του παίχτη και τοποθετείται στο pot.

Εναλλακτική ροή γεγονότων

2α. Ο χρήστης δεν έχει αρκετές μάρκες για να ισοφαρίσει το ποντάρισμα των άλλων παιχτών.

1. Ο χρήστης ποντάρει όσες μάρκες του απομένουν.

Μετά-συνθήκες

Ο χρήστης μπορεί και παραμένει ενεργός στην παρτίδα και παίζει ο επόμενος παίχτης.

2.4.2.4 Fold

Σύντομη περιγραφή

Ο χρήστης πιστεύει ότι δεν έχει αρκετά καλό φύλλο για να κερδίσει και επιθυμεί να πάει πάσο από την παρτίδα.

Προ-συνθήκες

Ο χρήστης είναι στη σειρά του να παίξει.

Κύρια ροή γεγονότων

1. Ο χρήστης πατάει το κουμπί Fold.
2. Το σύστημα αποσύρει τον χρήστη από τη συγκεκριμένη παρτίδα.

Μετά-συνθήκες

Ο χρήστης δε συνεχίζει να παίζει σε αυτήν την παρτίδα και περιμένει ανενεργός μέχρι να τελειώσει.

2.4.2.5 New Round

Σύντομη περιγραφή

Δημιουργείται μια νέα παρτίδα στο παιχνίδι.

Προ-συνθήκες

Υπάρχει ένα παιχνίδι σε εξέλιξη, έχει τελειώσει μια παρτίδα και ο χρήστης έχει επιλέξει να ξεκινήσει μια νέα παρτίδα.

Κύρια ροή γεγονότων

1. Η τράπουλα ανακατεύεται
2. Ορίζεται ποιός παίχτης έχει σειρά να παίξει το ρόλο του dealer.
3. Τοποθετούνται τα ποσά των blinds από τους παίχτες που πρέπει.
4. Μοιράζονται τα δύο κρυφά φύλλα στα χέρια των παιχτών.
5. Παίζει ο παίχτης που έχει σειρά.

Εναλλακτική ροή γεγονότων

- 3α. Ο παίχτης δεν έχει αρκετές μάρκες ώστε να τοποθετήσει το ποσό του blind
 1. Ο παίχτης τοποθετεί όσες μάρκες του περισσεύουν σαν ποσό του blind

Μετά-Συνθήκες

Τα blinds έχουν πονταριστεί, ο χρήστης βλέπει τα φύλλα του και ο παίχτης που έχει σειρά μπορεί να παίξει.

2.4.2.6 Save Game

Σύντομη περιγραφή

Ο χρήστης σώζει την πρόοδο του παιχνιδιού έως τώρα.

Προ-συνθήκες

Έχει ολοκληρωθεί μια παρτίδα και ο χρήστης έχει επιλέξει να σώσει την πρόοδο του παιχνιδιού μέχρι τώρα.

Κύρια ροή γεγονότων

1. Εμφανίζεται στην οθόνη το παράθυρο για την επιλογή του ονόματος του αρχείου που θα είναι σωσμένο το παιχνίδι.
2. Ο χρήστης επιλέγει το όνομα του αρχείου που θα είναι σωσμένο το παιχνίδι και πατάει το κουμπί Save.
3. Δημιουργείται ένα αρχείο με το όνομα που επιλέχτηκε από τον χρήστη και αποθηκεύονται σε αυτό όλες οι απαραίτητες πληροφορίες ώστε να μπορεί ο χρήστης να συνεχίσει το παιχνίδι κάποια άλλη στιγμή από το σημείο που το έσωσε.
4. Εμφανίζεται στην οθόνη το Frame που είχε την επιλογή Save.

Εναλλακτική ροή γεγονότων

2α. Το όνομα του αρχείου υπάρχει.

1. Εμφανίζεται στην οθόνη μια προειδοποίηση στον χρήστη ότι υπάρχει αρχείο με αυτό το όνομα.

2. Ο χρήστης μπορεί να επιλέξει να σώσει με αυτό το όνομα διαγράφοντας το προηγούμενο αρχείο ή να επιλέξει άλλο όνομα επαναλαμβάνοντας τη διαδικασία.

Μετά-συνθήκες

Έχει δημιουργηθεί ένα αρχείο με το όνομα που επέλεξε ο χρήστης, το οποίο θα περιέχει της απαραίτητες πληροφορίες ώστε να συνεχίσει, ο χρήστης, το παιχνίδι κάποια άλλη στιγμή.

2.4.2.7 Load Game

Σύντομη περιγραφή

Ο χρήστης θέλει να συνεχίσει ένα παιχνίδι που ξεκίνησε άλλη φορά από το σημείο που το είχε σώσει.

Προ-συνθήκες

Ο χρήστης έχει ξεκινήσει το πρόγραμμα του παιχνιδιού έχει πατήσει την επιλογή Load Game και υπάρχει αρχείο save στον ειδικό φάκελο του παιχνιδιού.

Κύρια ροή γεγονότων

1. Εμφανίζεται στην οθόνη μια λίστα με τα ονόματα των αρχείων των προηγούμενων σωσμένων παιχνιδιών.
2. Ο χρήστης επιλέγει το όνομα του αρχείου του παιχνιδιού που θέλει να συνεχίσει και πατάει το κουμπί Load.
3. Εμφανίζεται στην οθόνη η διεπιφάνεια του παιχνιδιού και φορτώνονται στο παιχνίδι όλα τα απαραίτητα στοιχεία για να συνεχίσει το παιχνίδι από το σημείο που το είχε σώσει ο χρήστης.

Μετά-συνθήκες

Ο χρήστης μπορεί και παίξει ένα παιχνίδι το οποίο είχε ξεκινήσει και σώσει κάποια άλλη στιγμή παλαιότερα.

2.5 Εννοιολογικό μοντέλο

Το εννοιολογικό μοντέλο είναι ένα εργαλείο που μας βοηθάει να απεικονίσουμε όλες τις βασικές οντότητες και τις συσχετίσεις μεταξύ τους σε ένα διάγραμμα. Οι έννοιες που απεικονίζονται είναι οντότητες του πραγματικού κόσμου που έχουν σχέση με το σύστημα και θεωρούμε ότι το απαρτίζουν. Στο εννοιολογικό μοντέλο θα πρέπει να απεικονίζονται οι βασικές οντότητες και συσχετίσεις. Δε θα πρέπει να το γεμίζουμε με όλες τις έννοιες που μπορούμε να σκεφτούμε γιατί έτσι το μοντέλο θα γίνει πολύπλοκο, ακατανόητο και δυσανάγνωστο. Το εννοιολογικό μοντέλο βοηθάει σε μεγάλο βαθμό στην κατανόηση μιας πρώτης δομής του συστήματος από την ομάδα ανάπτυξης λογισμικού αλλά και από τον πελάτη. Έτσι μπορούν όλοι οι εμπλεκόμενοι στο έργο να συμφωνήσουν σε μια κοινή γραμμή για το σύστημα το οποίο θέλουν να υλοποιηθεί.

Κάθε έννοια στο μοντέλο απεικονίζεται με ένα ορθογώνιο παραλληλόγραμμο το οποίο χωρίζεται στη μέση με μία ενδιάμεση οριζόντια γραμμή. Στο πάνω μέρος αναγράφεται το όνομα της έννοιας και στο κάτω οι παράμετροί της. Οι παράμετροι πρώτα έχουν την ένδειξη ορατότητάς τους. Ένα σύμβολο αφαίρεσης (-) αν είναι private, μία δίεση (#) αν είναι protected και το σύμβολο της πρόσθεσης (+) αν είναι public. Ακολουθεί το όνομα της παραμέτρου

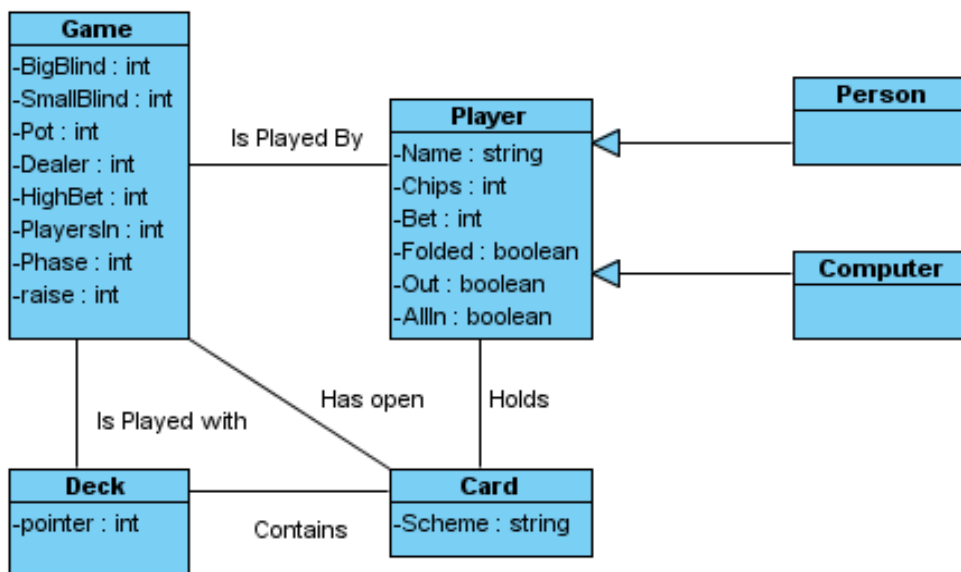
μία άνω και κάτω τελεία (:) και ο τύπος δεδομένων της (int, double, String κ.τ.λ.). Οι συσχετίσεις απεικονίζονται με μία γραμμή που ενώνει τις συσχετιζόμενες έννοιες. Πάνω στην πορεία της γραμμής αναγράφεται και ένα ρήμα που περιγράφει τη συσχέτιση μεταξύ των εννοιών.

Οι έννοιες που περιέχει το εννοιολογικό μοντέλο αργότερα θα εξεταστούν και οι περισσότερες από αυτές, το πιθανότερο, θα γίνουν κλάσεις στον πηγαίο κώδικα. Η διαδικασία παραγωγής πηγαίου κώδικα από μοντέλα λέγεται 'πρόσω μηχανική' (forward engineering). Η αντίστροφη διαδικασία παράγει μοντέλα παίρνοντας σαν είσοδο πηγαίο κώδικα και λέγεται 'αντίστροφη μηχανική' (reverse engineering). Περισσότερα για της κλάσεις και το διάγραμμά τους θα πούμε στο επόμενο κεφάλαιο της σχεδίασης.

2.5.1 Το εννοιολογικό μοντέλο για το σύστημά μας

Για το εννοιολογικό μοντέλο του παιχνιδιού που αναπτύσσουμε εδώ ορίζονται οι παρακάτω έννοιες. Η έννοια Game που είναι το παιχνίδι που παίζουμε, Η Card τα χαρτιά με τα οποία παίζεται το παιχνίδι, Η Deck που απαρτίζει τα χαρτιά της τράπουλας και τον Player τον παίχτη που παίζει. Η έννοια Player είναι αφηρημένη και κληρονομείται από τις έννοιες Person και Computer, όπως είδαμε και στο διάγραμμα περιπτώσεων χρήσης, που αντιπροσωπεύουν τον άνθρωπο και υπολογιστή παίχτη αντίστοιχα. Οι συνδέσεις που έχουμε είναι μεταξύ Player και Game με την έννοια ότι ο παίχτης παίζει το παιχνίδι, την σύνδεση μεταξύ Card και Deck αφού η τράπουλα απαρτίζεται από κάρτες, τη σύνδεση μεταξύ της έννοιας Card με τις έννοιες Player και Game και τη σύνδεση της Deck με την Game που την χειρίζεται.

Αυτό το εννοιολογικό μοντέλο είναι το αρχικό. Αργότερα στο πέμπτο κεφάλαιο θα δούμε την αναδόμηση του σχεδίου με σκοπό τη βελτίωση αυτού. Το εννοιολογικό μοντέλο μας φαίνεται στο παρακάτω σχήμα.



2.2 Εννοιολογικό μοντέλο

Βάσει αυτού του μοντέλου στη φάση της σχεδίασης θα γίνει το διάγραμμα κλάσεων. Λεπτομέρειες θα δούμε στο επόμενο κεφάλαιο.

2.6 Επίλογος

Με την ανάλυση που έγινε παραπάνω καθορίσαμε τη λειτουργία που θα έχει το πρόγραμμα και καθορίσαμε τις προδιαγραφές που αυτό θα πρέπει να πληρεί, σχετικά δηλαδή με θέματα όπως η διεπιφάνεια του χρήστη, τους κανόνες του παιχνιδιού και άλλα. Περιγράψαμε το 'τί' θα κάνει το πρόγραμμα χωρίς να ασχοληθούμε καθόλου με το 'πώς' θα το κάνει αυτό, πράγμα που είναι και ο σκοπός του βήματος της ανάλυσης. Καθορίστηκαν οι περιπτώσεις χρήσης και καταγράφηκαν οι τεκμηριώσεις τους. Ακόμα αποφασίστηκαν ποιες είναι οι οντότητες του συστήματος, τί σχέσεις θα έχουν μεταξύ τους και σχεδιάστηκαν σε ένα εννοιολογικό μοντέλο. Εφόσον ολοκληρώθηκε το κομμάτι αυτό μπορούμε να προχωρήσουμε στο επόμενο βήμα του κύκλου ανάπτυξης λογισμικού, που είναι η σχεδίαση της αρχιτεκτονικής του συστήματος. Με αυτό το θέμα θα ασχοληθούμε στο επόμενο κεφάλαιο, στο οποίο θα γίνει λεπτομερής σχεδίαση του προγράμματος που θα αναπτύξουμε.

Κεφάλαιο 3 Σχεδίαση

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει η σχεδίαση του προγράμματος. Για να γίνει αυτό θα χρησιμοποιηθούν διάφορα τεχνουργήματα της UML. Συγκεκριμένα θα γίνουν ένα διάγραμμα περιπτώσεων χρήσης, η τεκμηρίωση των περιπτώσεων χρήσης αυτών, τα διαγράμματα ακολουθίας συστήματος, το εννοιολογικό μοντέλο και το διάγραμμα των κλάσεων που θα χρησιμοποιηθούν. Η UML, ή αλλιώς στα ελληνικά η Ενοποιημένη Γλώσσα Μοντελοποίησης, είναι μια γλώσσα, η οποία χρησιμοποιείται για τον προσδιορισμό, την οπτικοποίηση, ανάπτυξη και τεκμηρίωση των κατασκευασμάτων ενός συστήματος λογισμικού. Δεν είναι γλώσσα προγραμματισμού. Παρέχει γενικά πληροφορίες για τη δομή και την συμπεριφορά του συστήματος. Η UML έχει αποδειχθεί ιδιαίτερα χρήσιμη στην προσπάθεια σχεδίασης συστημάτων, ειδικά αντικειμενοστραφών αφού πλέον υπάρχουν εργαλεία τα οποία μπορούν να παράγουν αυτόματα κώδικα από συγκεκριμένα είδη διαγραμμάτων της UML. Εδώ θα χρησιμοποιηθούν τα τεχνουργήματα που αναφέρθηκαν παραπάνω προκειμένου να γίνει μια μοντελοποίηση του προγράμματος που πρόκειται να αναπτύξουμε.

3.2 Διάγραμμα κλάσεων

Το διάγραμμα κλάσεων είναι ένα τεχνουργήμα της UML που μας επιτρέπει να αναπαραστήσουμε τις κλάσεις ενός σχεδίου λογισμικού μαζί με λεπτομέρειες σχετικά με τις ιδιότητες και τις μεθόδους που τις απαρτίζουν, καθώς και τις συζεύξεις των κλάσεων μεταξύ τους και τις σχέσεις κληρονομικότητας.

Οι κλάσεις είναι ο πυρήνας των αντικειμενοστραφών συστημάτων. Μία κλάση αναπαριστά μια διακριτή έννοια της εφαρμογής, έννοια που συνήθως είναι μια οντότητα του πραγματικού κόσμου. Τα αντικείμενα κάθε κλάσης που θα δημιουργηθούν θα έχουν λειτουργία και συμπεριφορά που θα ορίζεται από την κλάση. Η κατάσταση των αντικειμένων αυτών ορίζεται από τις ιδιότητές του. Οι ιδιότητες αυτές μπορεί να είναι το όνομα του ανθρώπου ή το φύλλο του ή ότι μπορεί κανείς να σκεφτεί για μια έννοια. Οι τιμές των ιδιοτήτων διαφοροποιούν τα αντικείμενα μεταξύ τους. Η συμπεριφορά των αντικειμένων ορίζεται από τις μεθόδους του. Οι μέθοδοι αποτελούν λειτουργίες που μπορεί να εκτελέσει μια

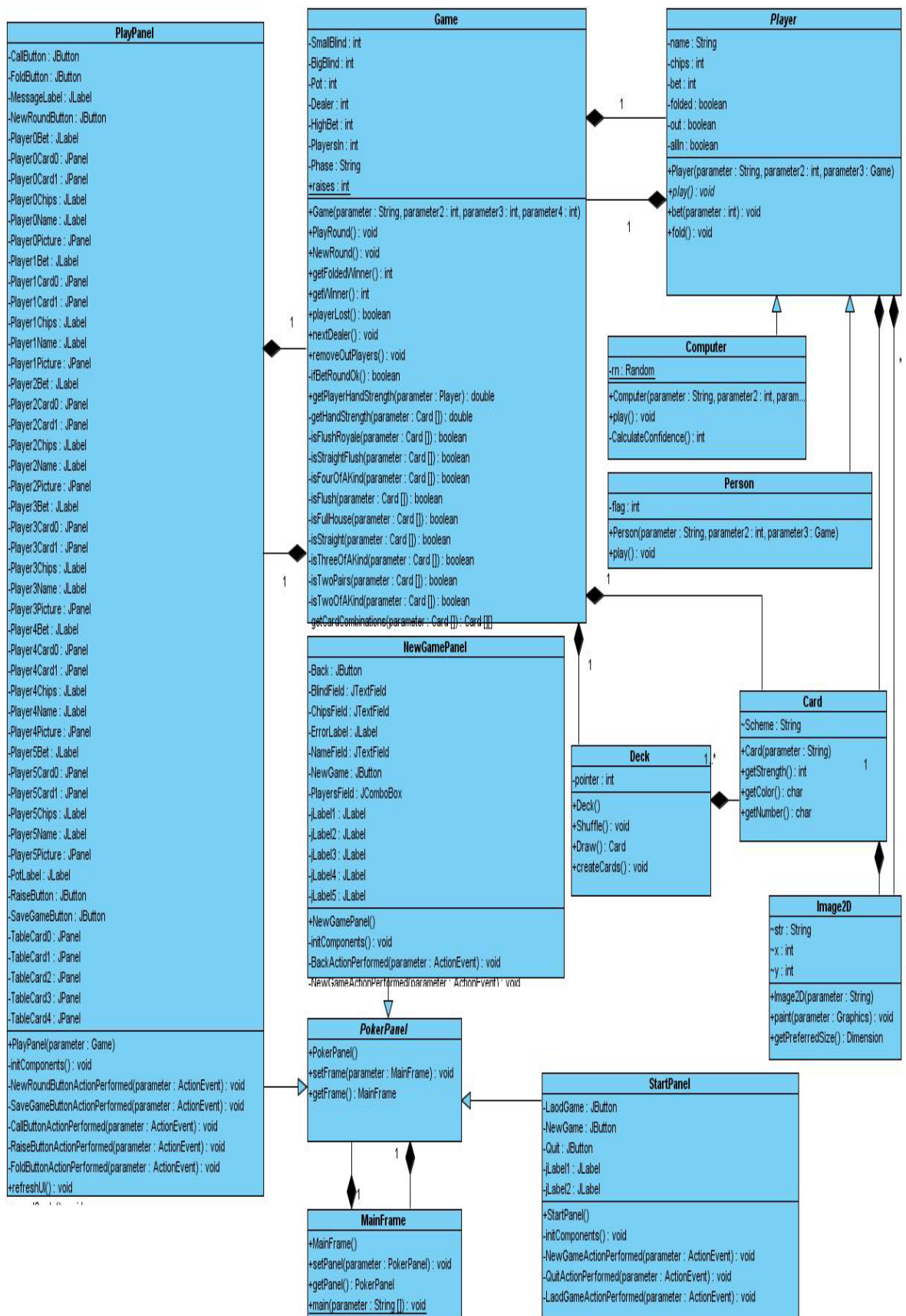
κλάση. Τις περισσότερες φορές αφορούν υπολογισμούς σχετικούς με το κάθε αντικείμενο.

Το διάγραμμα κλάσεων μοιάζει πάρα πολύ με το εννοιολογικό μοντέλο. Η διαφορά είναι ότι στο διάγραμμα κλάσεων οι συσχετίσεις μεταξύ των κλάσεων σχηματίζονται με διάφορες γραμμές ανάλογα με το είδος της σχέσης. Στις σχέσεις αναγράφουμε και την πολλαπλότητα της και για κάθε κλάση αναγράφονται οι μέθοδοί της. Οι μέθοδοι όπως και οι ιδιότητες έχουν αναγνωριστικό ορατότητας και τύπο δεδομένου επιστροφής ή την ένδειξη void αν δεν επιστρέφουν κάποια τιμή. Όταν μια κλάση ή μια μέθοδος είναι αφηρημένη τότε το όνομά της γράφεται με πλάγιους χαρακτήρες. Όταν μία ιδιότητα ή μια μέθοδος είναι στατική, δηλαδή κοινή για όλα τα αντικείμενα αυτής της κλάσης, τότε το όνομα είναι υπογραμμισμένο.

3.2.1 Το διάγραμμα κλάσεων για το σύστημά μας

Το διάγραμμα κλάσεων για το τρέχον έργο φαίνεται στο σχήμα 3.1 . Για το σχέδιο λογισμικού ορίζονται οι κλάσεις Game, Player, Person, Computer, Deck, Card. Αυτές τις κλάσεις τις είδαμε σαν έννοιες στο εννοιολογικό μοντέλο. Επίσης στο διάγραμμα φαίνονται και οι κλάσεις Image2D, PokerPanel, NewGamePanel, PlayPanel, StartPanel, MainFrame. Αυτές οι κλάσεις δημιουργούν την διεπιφάνεια του χρήστη. Περισσότερα για αυτό θα δούμε στο επόμενο κεφάλαιο της υλοποίησης.

Σε αυτές τις κλάσεις ορίζονται και μέθοδοι που εκτελούν τις βασικές περιπτώσεις χρήσης που ορίστηκαν στη φάση της ανάλυσης. Η περίπτωση χρήσης New Game υλοποιείται από την κλάση NewGamePanel και την μέθοδο NewGameActionPerformed(). Η Play υλοποιείται διαφορετικά από τις κλάσεις Person και Computer ενώ δηλώνεται σαν αφηρημένη μέθοδο στην κλάση Player. Οι περιπτώσεις χρήσης Bet και Fold είναι μέθοδοι στην κλάση Player γιατί είναι κοινή λειτουργία για τις Person και Computer. Η New Round υλοποιείται από την μέθοδο NewRound() στην κλάση Game και οι Load Game και Save Game από τις μεθόδους LoadGameActionPerformed() και SaveGameActionPerformed() στις



3.1 Διάγραμμα κλάσεων πρώτης έκδοσης

κλάσεις StartPanel και PlayPanel αντίστοιχα. Αυτό το διάγραμμα είναι της πρώτης έκδοσης του λογισμικού. Στο πέμπτο κεφάλαιο θα αναφερθούν βελτιώσεις του λογισμικού και του διαγράμματος κλάσεων.

3.3 Διάγραμμα ακολουθίας συστήματος

Το διάγραμμα ακολουθίας συστήματος είναι ένα ακόμα τεχνούργημα της UML που μας βοηθάει να σχεδιάσουμε και να απεικονίσουμε πως τα αντικείμενα του συστήματος επικοινωνούν και στέλνουν μηνύματα μεταξύ τους για να εκτελεστεί μία περίπτωση χρήσης. Τα μηνύματα αυτά πολλές φορές αποτελούν το αίτημα ενός αντικειμένου σε ένα άλλο να εκτελέσει μια μέθοδό του και να του επιστρέψει, ή όχι, μία τιμή.

Το διάγραμμα ακολουθίας συστήματος απεικονίζει σαν κάθετες γραμμές ,που ονομάζονται γραμμές ζωής, διάφορα αντικείμενα τα οποία ‘ζουν’ ταυτόχρονα. Στα σημεία που ένα αντικείμενο δε ζει η γραμμή είναι διακεκομμένη ενώ σε αντίθετη περίπτωση είναι διπλή συνεχόμενη. Στην κορυφή κάθε γραμμής ζωής υπάρχει ένα τετράγωνο παραλληλόγραμμο το οποίο περιέχει το όνομα του αντικειμένου ή ένα σχεδιασμένο ανθρωπάκι αν πρόκειται για ηθοποιό. Ο ηθοποιός όπως και στο διάγραμμα περιπτώσεων χρήσης αναπαριστά έναν χρήστη του συστήματος. Τα μηνύματα που στέλνουν τα αντικείμενα μεταξύ τους απεικονίζονται με ένα οριζόντιο κατευθυνόμενο βέλος που κατευθύνεται προς το αντικείμενο που πάει το μήνυμα και πάνω από αυτό αναγράφεται σύντομα, συνήθως με μία ή δύο λέξεις, η περιγραφή του μηνύματος. Τα βέλη τοποθετούνται στον κάθετο άξονα του διαγράμματος ανάλογα με τη χρονική στιγμή που αποστέλλονται. Επιπλέον, κάποια αντικείμενα μπορεί να στείλουν κάποιο μήνυμα στον εαυτό τους. Τέτοιες περιπτώσεις συμβαίνουν ,για παράδειγμα, όταν μέσα στο σώμα μιας μεθόδου καλείται μία άλλη μέθοδος της ίδιας κλάσης. Αυτά τα μηνύματα απεικονίζονται με ένα βέλος που ξεκινάει και καταλήγει στο ίδιο αντικείμενο. Εκτός από τα μηνύματα ένα διάγραμμα ακολουθίας συστήματος μπορεί να περιέχει και την δυναμική δημιουργία αντικειμένων. Αυτό απεικονίζεται με ένα βέλος που καταλήγει στο ορθογώνιο παραλληλόγραμμο που περιέχει το όνομα του αντικειμένου που δημιουργείται ή με την κλήση του δομητή του προς δημιουργία αντικειμένου.²

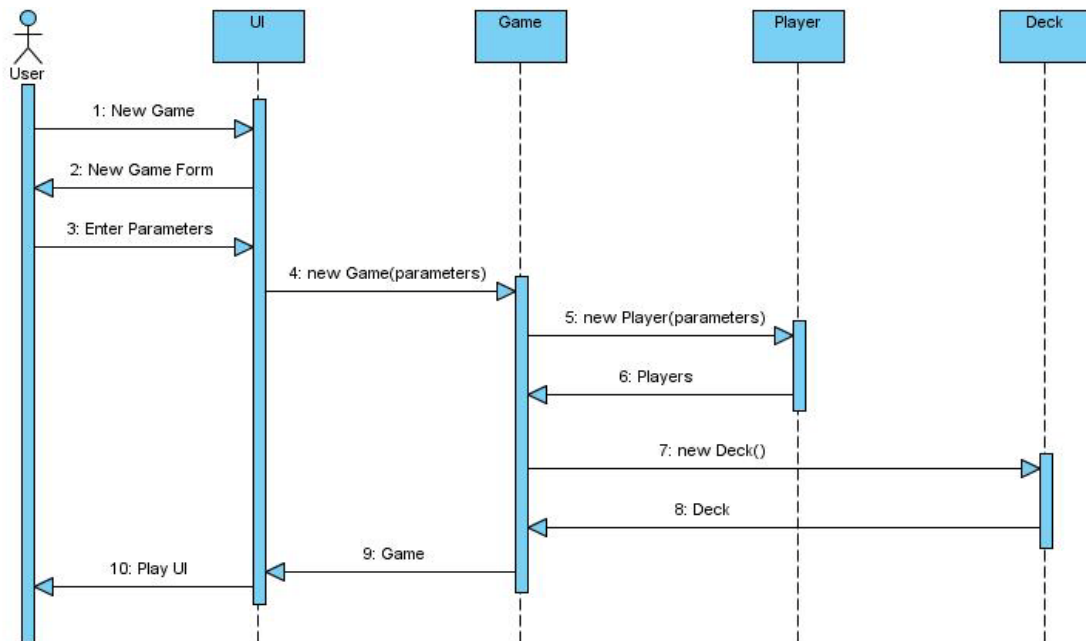
Το διάγραμμα ακολουθίας συστήματος δείχνει με τη σειρά την αποστολή μηνυμάτων που απαιτούνται για την ολοκλήρωση μιας περίπτωσης χρήσης και για αυτό σχεδιάζονται με βάση την τεκμηρίωσή τους αλλά και το διάγραμμα κλάσεων. Αντικείμενα δεν μπορούν να ανταλλάξουν μηνύματα αν δεν υπάρχει κάποια σύζευξη μεταξύ τους ή αν δεν δημιουργείται κάποιο στιγμιότυπο του ενός αντικειμένου στο άλλο. Είναι εξαιρετικά χρήσιμα για τον εντοπισμό λειτουργιών σε κάποιες κλάσεις. Όταν ένα αντικείμενο στέλνει ένα μήνυμα αυτό μπορεί να σημαίνει την κλήση μιας μεθόδου στον παραλήπτη, γεγονός που στην φάση της υλοποίησης μας οδηγεί στην συγγραφή της μεθόδου αυτής.

3.3.1 Τα διαγράμματα ακολουθίας συστήματος για το σύστημά μας

Παρακάτω θα δούμε τα διαγράμματα ακολουθίας συστήματος για το παιχνίδι που αναπτύσσουμε σε αυτήν την εργασία.

3.3.1.1 New Game

Σε αυτό το διάγραμμα, όπως φαίνεται παρακάτω, αλληλεπιδρούν οι κλάσεις Game, Player, Deck οι κλάσεις της διεπιφάνειας χρήστη και ο χρήστης.

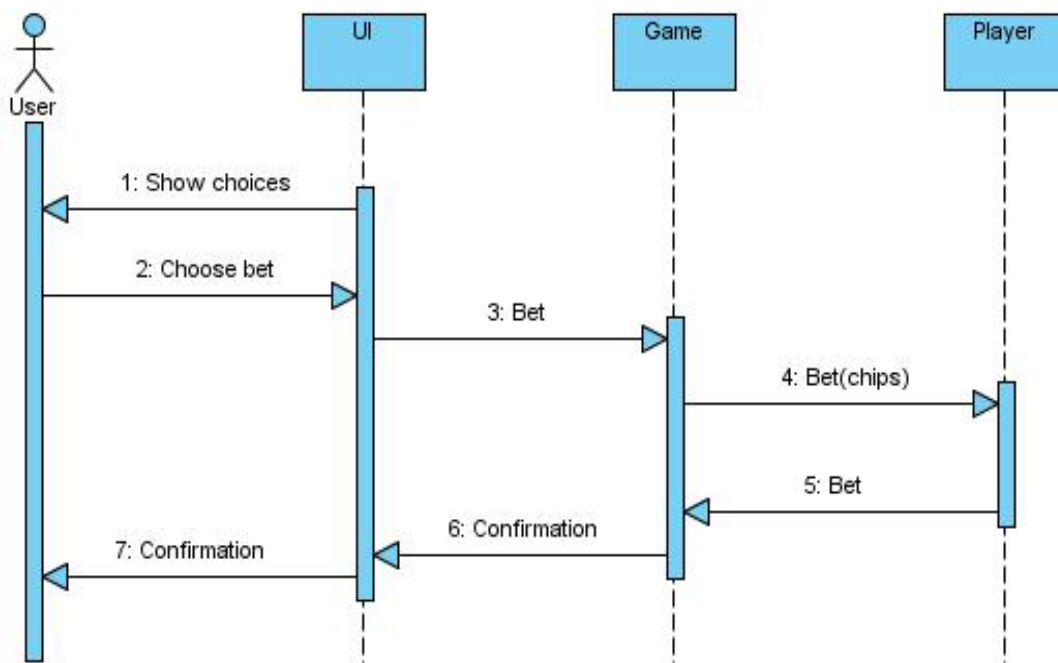


3.2 Διάγραμμα ακολουθίας συστήματος New Game

Ο χρήστης πατάει το κουμπί New Game και το σύστημα του δίνει τη φόρμα έναρξης νέου παιχνιδιού. Αυτός εισάγει τα στοιχεία και πατάει το κουμπί New Game. Αμέσως δημιουργούνται τα αντικείμενα του παιχνιδιού, των παιχτών και της τράπουλας και το σύστημα δείχνει στον χρήστη τη διεπιφάνεια του παιχνιδιού.

3.3.1.2 Play

Σε αυτή την περίπτωση χρήσης συμμετέχουν οι κλάσεις Game, Player, ο χρήστης και η διεπιφάνεια.

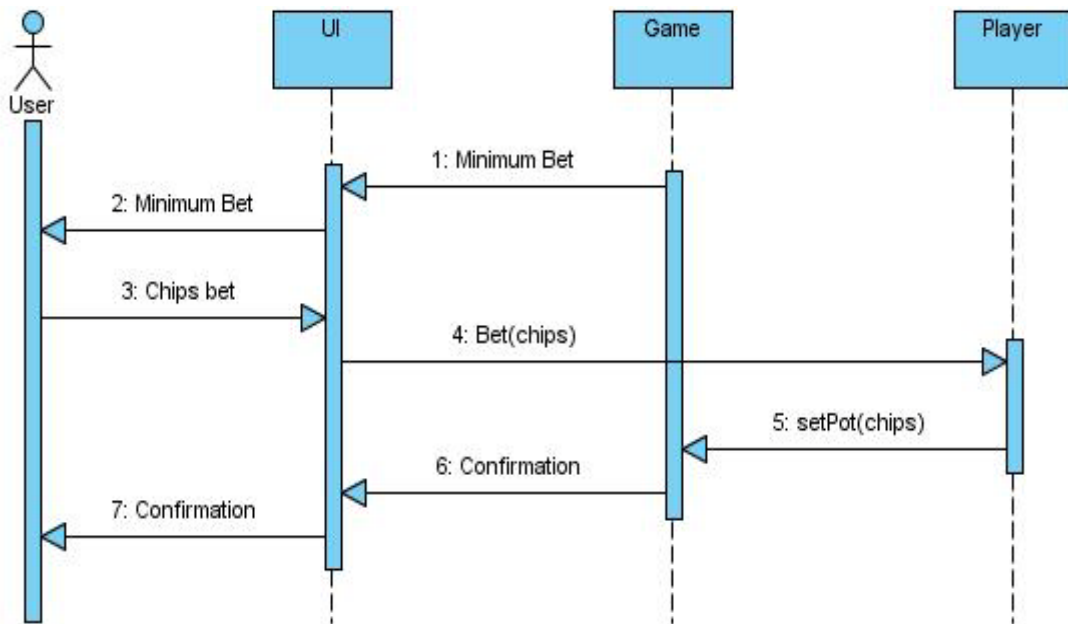


3.3 Διάγραμμα ακολουθίας συστήματος Play

Η διεπιφάνεια δίνει στον χρήστη τις επιλογές του. Στην κύρια ροή γεγονότων αυτός επιλέγει να ποντάρει. Ενσωματώνεται η περίπτωση χρήσης Bet, η κίνηση επιβεβαιώνεται και η λειτουργία τερματίζεται. Εδώ φαίνεται ότι υπάρχει η ανάγκη υλοποίησης μιας μεθόδου bet(int chips) στην κλάση Player.

3.3.1.3 Bet

Η περίπτωση χρήσης αυτή καλείται κατά την ομαλή λειτουργία της Play και συμμετέχουν οι ίδιες κλάσεις με αυτή.

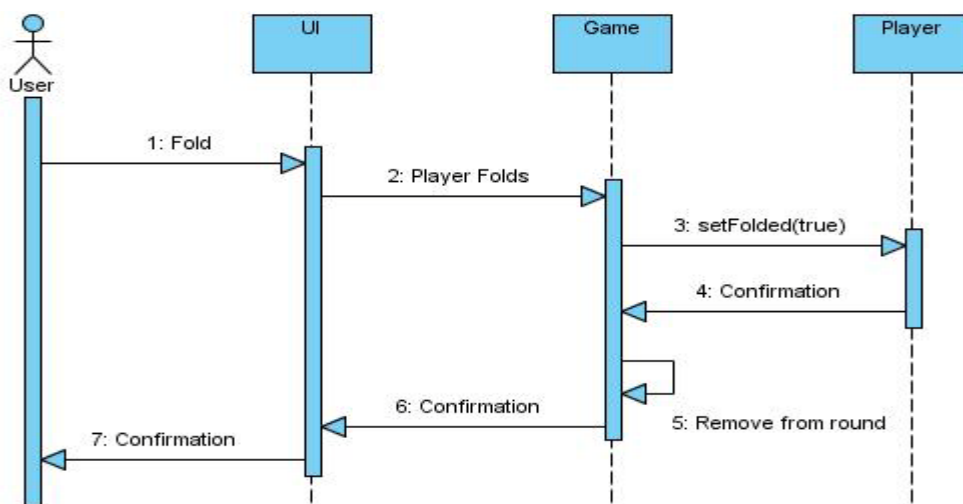


3.4 Διάγραμμα ακολουθίας συστήματος Bet

Ο χρήστης πρώτα ενημερώνεται για το ποσό που θα πρέπει να ποντάρει για να μείνει στην παρτίδα και το ποσό που θα πρέπει να ποντάρει για να αυξήσει το ποντάρισμα. Επιλέγει ποιο ποσό θέλει να ποντάρει και ενημερώνονται οι κλάσεις Game και Player.

3.3.1.4 Fold

Η Fold εκτελείται στην εναλλακτική ροή της περίπτωσης χρήσης Play όταν ο παίχτης αποφασίζει να πάει πάσο από τη παρτίδα και να βγει από αυτήν. Συμμετέχουν ξανά οι ίδιες κλάσεις.

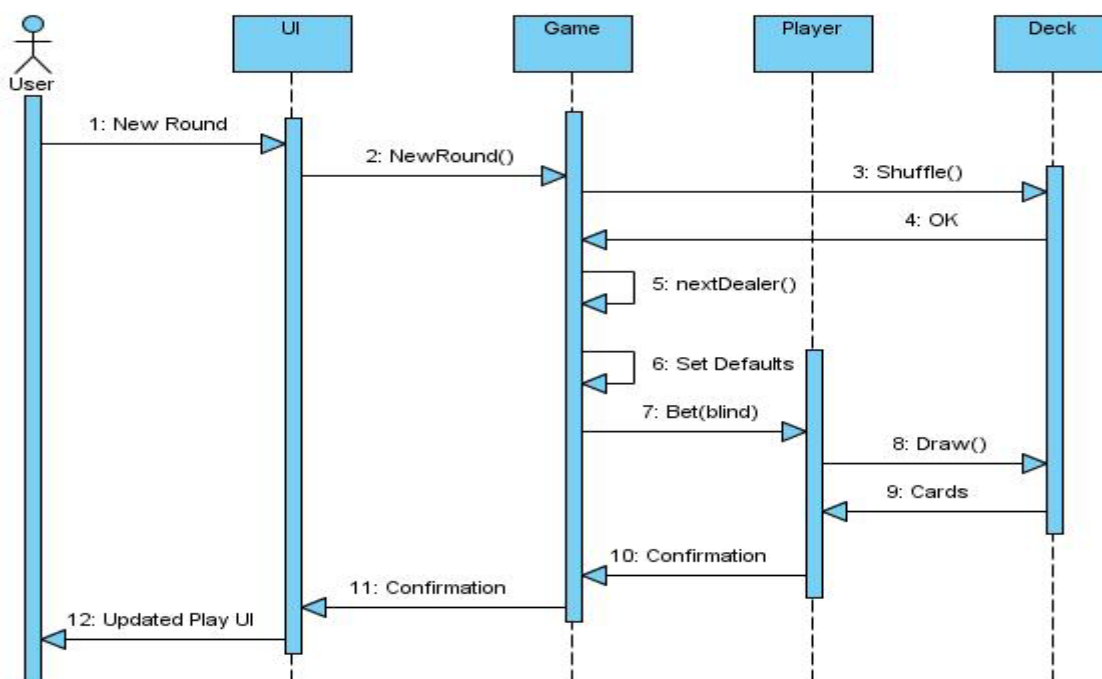


3.5 Διάγραμμα ακολουθίας συστήματος Fold

Ομοίως με την περίπτωση της Bet μόνο που εδώ ο χρήστης επιλέγει το κουμπί Fold. Ενημερώνεται το αντικείμενο του παίχτη ότι ο συγκεκριμένος έχει πάει πάσο, το παιχνίδι τον αποσύρει από την συγκεκριμένη παρτίδα και ο χρήστης ενημερώνεται για την ολοκλήρωση της λειτουργίας.

3.3.1.5 New Round

Η λειτουργία κάνει όλες τις απαιτούμενες ενέργειες για την έναρξη μίας νέας παρτίδας. Αφορά τις κλάσεις Game, Player, Deck, την διεπιφάνεια χρήστη και τον χρήστη.

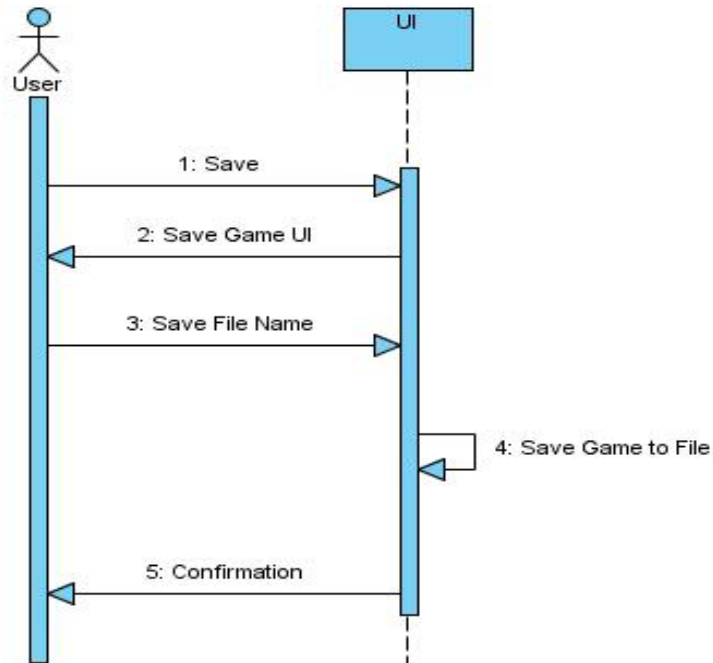


3.6 Διάγραμμα ακολουθίας συστήματος New Round

Ο χρήστης μετά το τέλος μιας παρτίδας επιλέγει να ξεκινήσει καινούργια και πατάει το κουμπί New Round. Η NewRound() θα πρέπει να υλοποιηθεί σαν μέθοδος στην κλάση Game. Ακόμα πρέπει να υπάρχουν μέθοδοι για το ανακάτεμα της τράπουλας, για την επιλογή του επόμενου 'Dealer' και για το τράβηγμα μιας κάρτας από την τράπουλα. Στη συνέχεια μηδενίζονται οι τιμές που πρέπει και μοιράζονται τα δύο κρυφά φύλλα στους παίχτες.

3.3.1.6 Save Game

Αυτή η λειτουργία σώζει ένα παιχνίδι σε ένα αρχείο. Η δυνατότητα αυτή δίνεται στον χρήστη μόνο μετά το τέλος μιας παρτίδας και ποτέ στη μέση αυτής.

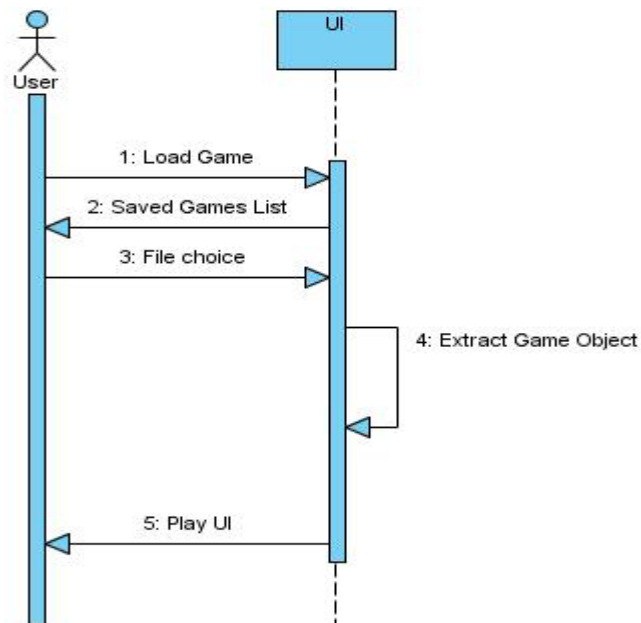


3.7 Διάγραμμα ακολουθίας συστήματος Save Game

Ο χρήστης μετά το τέλος μιας παρτίδας πατάει το κουμπί Save Game. Το σύστημα ανοίγει ένα παράθυρο επιλογής αρχείου. Το σύστημα σώζει στο αρχείο που επέλεξε ο χρήστης το αντικείμενο του παιχνιδιού και ενημερώνει τον χρήστη. Η λειτουργία θα υλοποιηθεί από μέθοδο στην κλάση PlayPanel.

3.3.1.7 Load Game

Η περίπτωση χρήσης Load Game φορτώνει ένα παιχνίδι από σωσμένο αρχείο που σώθηκε από την Save Game. Ο χρήστης επιλέγει το κουμπί Load Game αντί για το New Game και εμφανίζεται το παράθυρο επιλογής αρχείου. Ο χρήστης επιλέγει το αρχείο στο οποίο είχε παλιότερα σώσει κάποιο παιχνίδι και το σύστημα φορτώνει το αντικείμενο του παιχνιδιού και εμφανίζει την διεπιφάνεια για να παίξει. Η λειτουργία θα υλοποιηθεί από μέθοδο στην κλάση StartPanel.



3.8 Διάγραμμα ακολουθίας συστήματος Load Game

3.4 Επίλογος

Σε αυτό το κεφάλαιο είδαμε τη διαδικασία σχεδίασης της αρχιτεκτονικής του λογισμικού που αναπτύσσουμε. Με βάση το εννοιολογικό μοντέλο από τη φάση της ανάλυσης σχεδιάσαμε το διάγραμμα κλάσεων για το σύστημα, στο οποίο απεικονίστηκαν λεπτομερώς οι ιδιότητες κάθε κλάσης, οι λειτουργίες που θα υλοποιεί η καθεμιά από αυτές και τις σχέσεις που έχουν μεταξύ τους. Στα διαγράμματα ακολουθίας συστήματος, τα οποία σχεδιάστηκαν με βάση το διάγραμμα περιπτώσεων χρήσης και τις τεκμηριώσεις τους, είδαμε τα μηνύματα που θα πρέπει να ανταλλάσσουν τα αντικείμενα μεταξύ τους για να ολοκληρώνεται κάθε περίπτωση χρήσης

Η επόμενη φάση της ανάπτυξης λογισμικού είναι η φάση της υλοποίησης, στην οποία γίνεται η συγγραφή του κώδικα. Στο επόμενο κεφάλαιο θα γίνει μια παρουσίαση της φάσης αυτής για το τρέχον έργο.

Κεφάλαιο 4 Υλοποίηση

4.1 Εισαγωγή

Κατά τη φάση της υλοποίησης γίνεται η συγγραφή του κώδικα με βάση το σχέδιο λογισμικού που αναπτύχθηκε στη φάση της σχεδίασης. Σε μεγάλα έργα για αυτήν τη διαδικασία απασχολούνται πολλοί προγραμματιστές στους οποίους κατανέμονται οι εργασίες με τρόπο ώστε ο καθένας να υλοποιεί διαφορετικές αυτόνομες λειτουργίες του συστήματος ώστε να μην χρειάζεται η άμεση συνεργασία μεταξύ τους. Αφού ολοκληρωθούν οι ξεχωριστές λειτουργίες, ακολουθεί η διαδικασία της συνένωσης των δομικών στοιχείων στο τελικό σύστημα.

Το τρέχον έργο είναι μικρό σε έκταση και είναι εύκολο να συγγραφεί από έναν μόνο προγραμματιστή.

Σε αυτό το κεφάλαιο θα γίνει μια παρουσίαση της φάσης της υλοποίησης του λογισμικού. Θα δούμε την διεπιφάνεια του χρήστη, μία περιγραφή της λειτουργίας κάθε μεθόδου στο σύστημα, τον τρόπο υπολογισμού της δύναμης κάθε κόλπου στο χέρι των παιχτών και τον τρόπο λειτουργίας της τεχνητής νοημοσύνης.

4.2 Εργαλεία που χρησιμοποιήθηκαν

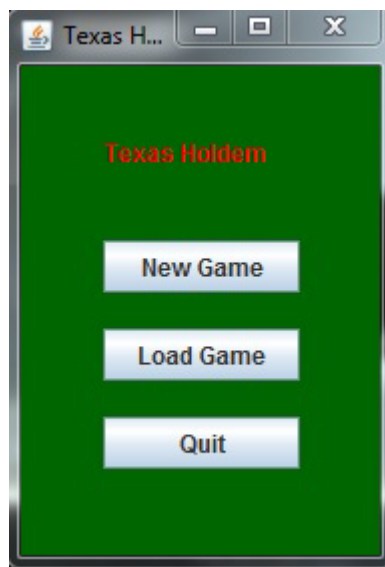
Το λογισμικό αναπτύχθηκε στη γλώσσα προγραμματισμού Java. Η έκδοση του Java Development Kit που χρησιμοποιήθηκε είναι η 6u18 για Windows και το πρόγραμμα στο οποίο έγινε η συγγραφή του κώδικα είναι το NetBeans IDE 6.9.1. Η τεχνολογία που χρησιμοποιήθηκε για τη διεπιφάνεια χρήσης είναι το ενσωματωμένο πακέτο της Java javax.swing. Για τη σχεδίαση της διεπιφάνειας του χρήστη χρησιμοποιήθηκε μια ενσωματωμένη λειτουργία του NetBeans που επιτρέπει τη σχεδίαση ενός παραθύρου με παλέτα συστατικών του πακέτου javax.swing. Οι εικόνες των παιχτών στη διεπιφάνεια του παιχνιδιού, σχεδιάστηκαν με το Southpark studio, ιστοσελίδα στο διαδίκτυο που επιτρέπει τη σχεδίαση χαρακτήρων όμοιων με αυτούς της τηλεοπτικής σειράς κινουμένων σχεδίων Southpark.

4.3 Η διεπιφάνεια του χρήστη

Εδώ θα γίνει η παρουσίαση της διεπιφάνειας του χρήστη. Για αυτή χρησιμοποιήθηκαν κλάσεις που κληρονομούν τις κλάσεις JFrame και JPanel του πακέτου javax.swing. Η κεντρική ιδέα είναι να υπάρχει μόνο ένα Frame το MainFrame, όπως φαίνεται και στο διάγραμμα κλάσεων, και αυτό να περιέχει τα διάφορα Panels που θα εναλλάσσονται μεταξύ τους ανάλογα με το πιο χρειάζεται κάθε φορά. Για αυτό όλα τα Panel θα κληρονομούν την αφηρημένη κλάση PokerPanel, η οποία κληρονομεί την JPanel και θα είναι συνδεδεμένη με την MainFrame. Η MainFrame υλοποιεί την μέθοδο setPanel(PokerPanel p) η οποία αλλάζει στο παράθυρο το Panel που πρέπει να δείξει.

4.3.1 Το αρχικό παράθυρο

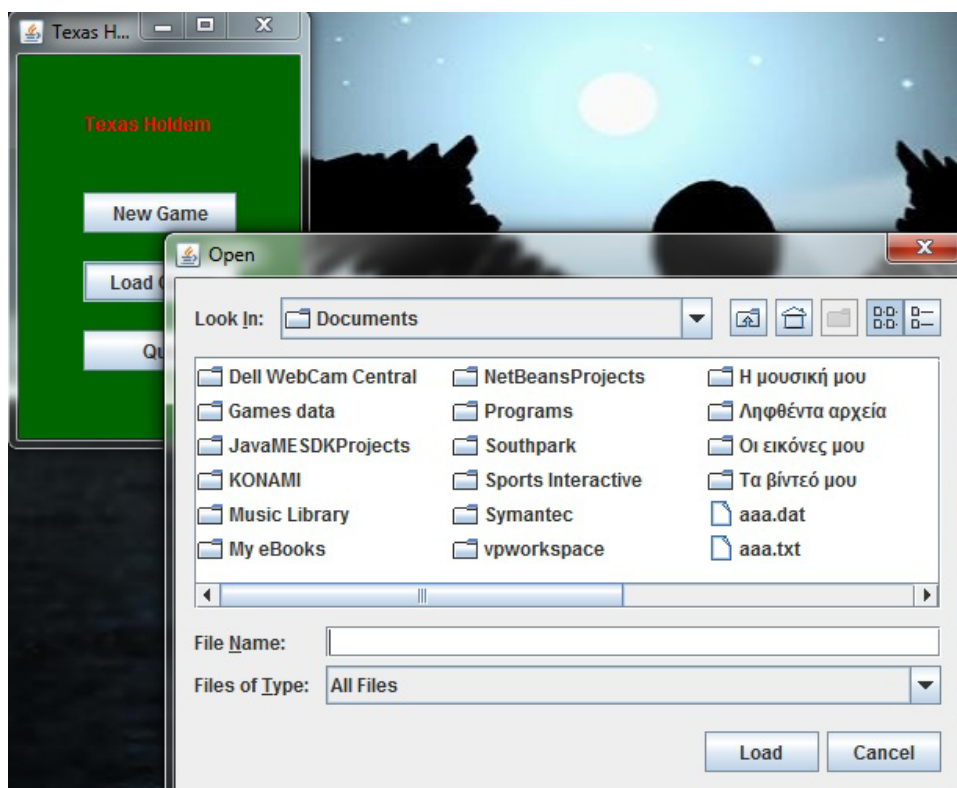
Το αρχικό παράθυρο που εμφανίζεται όταν ο χρήστης ξεκινάει την εφαρμογή. Όπως είπαμε και πριν το παράθυρο υλοποιείται με την κλάση MainFrame ενώ το περιεχόμενό του είναι υποκλάση της κλάσης PokerPanel. Το παράθυρο φαίνεται στην παρακάτω εικόνα.



4.1 Το αρχικό παράθυρο

Το πρώτο παράθυρο όπως φαίνεται παραπάνω περιέχει τον τίτλο του παιχνιδιού και τρία κουμπιά τα New Game, Load Game και Quit. Πατώντας το Quit τερματίζεται το πρόγραμμα. Στο παράθυρο έχει τεθεί να μη γίνεται τίποτα όταν ο

χρήστης πατήσει το 'X'. Το κουμπί Load Game εμφανίζει το παράθυρο επιλογής αρχείου όπως φαίνεται στην παρακάτω εικόνα.



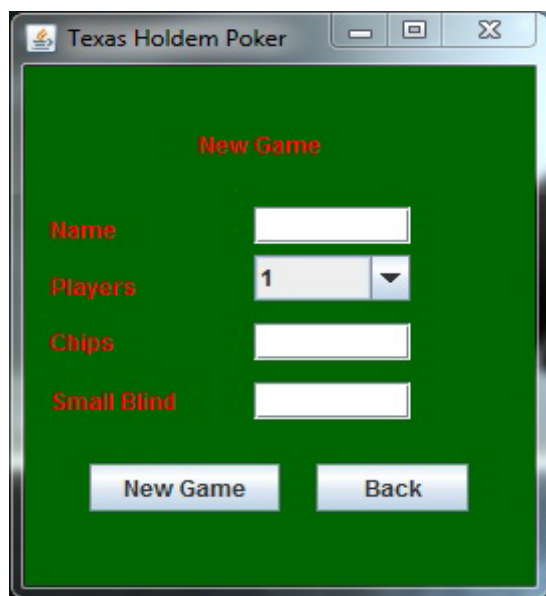
4.2 Ο επιλογέας αρχείου

Από εδώ ο χρήστης μπορεί να επιλέξει το αρχείο στο οποίο είχε σώσει κάποια προηγούμενη στιγμή το παιχνίδι του. Για τη διαδικασία φόρτωσης του παιχνιδιού καλείται η μέθοδος `LoadGameActionPerformed()`. Με την επιτυχή ανάκτηση του αντικειμένου από το αρχείο ανοίγει το παράθυρο παιχνιδιού με όλα στοιχεία του σωσμένου αντικειμένου.

4.3.2 Το παράθυρο έναρξης νέου παιχνιδιού

Πατώντας το κουμπί New Game στο αρχικό παράθυρο εμφανίζεται το παράθυρο έναρξης νέου παιχνιδιού, που υλοποιείται μέσω της κλάσης `NewGamePanel`. Το παράθυρο φαίνεται στην εικόνα 4.3. Σε αυτό το παράθυρο ο χρήστης περνάει τις παραμέτρους του παιχνιδιού όπως αυτές ορίστηκαν από τον καθορισμό απαιτήσεων στη φάση της ανάλυσης. Οι παράμετροι αυτοί είναι το όνομα του παίχτη, ο αριθμός των αντιπάλων, οι αρχικές μάρκες και το ποσό του small blind. Μόλις οι παράμετροι περαστούν, ο χρήστης μπορεί να ξεκινήσει το

παιχνίδι πατώντας το κουμπί New Game. Σε περίπτωση που ο χρήστης κάνει λάθος εισαγωγή στοιχείων όπως να μην συμπληρώσει κάποιο πεδίο ή να βάλει χαρακτήρες στο πεδίο των μαρκών το σύστημα του δίνει ένα μήνυμα λάθους και δε ξεκινάει το παιχνίδι.



4.3 Φόρμα εκκίνησης παιχνιδιού

Πατώντας το κουμπί New Game καλείται η μέθοδος `NewGameActionPerformed` η οποία δημιουργεί ένα αντικείμενο της κλάσης `Game` περνώντας σαν παραμέτρους στο δομητή τα στοιχεία που δόθηκαν και φορτώνεται το παράθυρο του παιχνιδιού.

4.3.3 Το παράθυρο παιχνιδιού

Το Panel του παραθύρου παιχνιδιού είναι το `PlayPanel`. Η εικόνα που βλέπει ο χρήστης μόλις ξεκινήσει το παιχνίδι φαίνεται στην εικόνα 4.4. Όλες οι γραφικές εικόνες που φαίνονται στο παράθυρο είναι αντικείμενα της κλάσης `Image2D` που κληρονομεί την `JComponent`. Ο δομητής της `Image2D` παίρνει σαν είσοδο το μονοπάτι ενός αρχείου εικόνας και δείχνει τη συγκεκριμένη εικόνα στο `container` που θα μπει. Το παράθυρο έχει πέντε κουμπιά, δύο κάτω δεξιά και τρία κάτω αριστερά. Τα κάτω αριστερά ενεργοποιούνται όταν έρθει η σειρά του παίχτη να παίξει. Είναι στην ουσία οι τρεις επιλογές που έχει ο παίχτης στη σειρά του. Ή θα δει το ποντάρισμα (`Call`) ή θα ανεβάσει το ποντάρισμα (`Raise`) ή θα πάει πάσο (`Fold`). Τα κάτω δεξιά κουμπιά είναι τα `New Round` και `Save and Exit` τα οποία

ενεργοποιούνται ανάμεσα στις παρτίδες. Το πρώτο καλεί την μέθοδο `PlayRound()` της `Game` η οποία παίζει μία ολόκληρη παρτίδα. Το δεύτερο σώζει το παιχνίδι εκτελώντας τη μέθοδο `SaveGameActionPerformed()`. Η μέθοδος αυτή εμφανίζει ένα παράθυρο επιλογής αρχείου όπως φαίνεται στην εικόνα 4.5 και σώζει το αντικείμενο του παιχνιδιού στο επιλεγμένο αρχείο.



4.4 Η διεπιφάνεια του παιχνιδιού

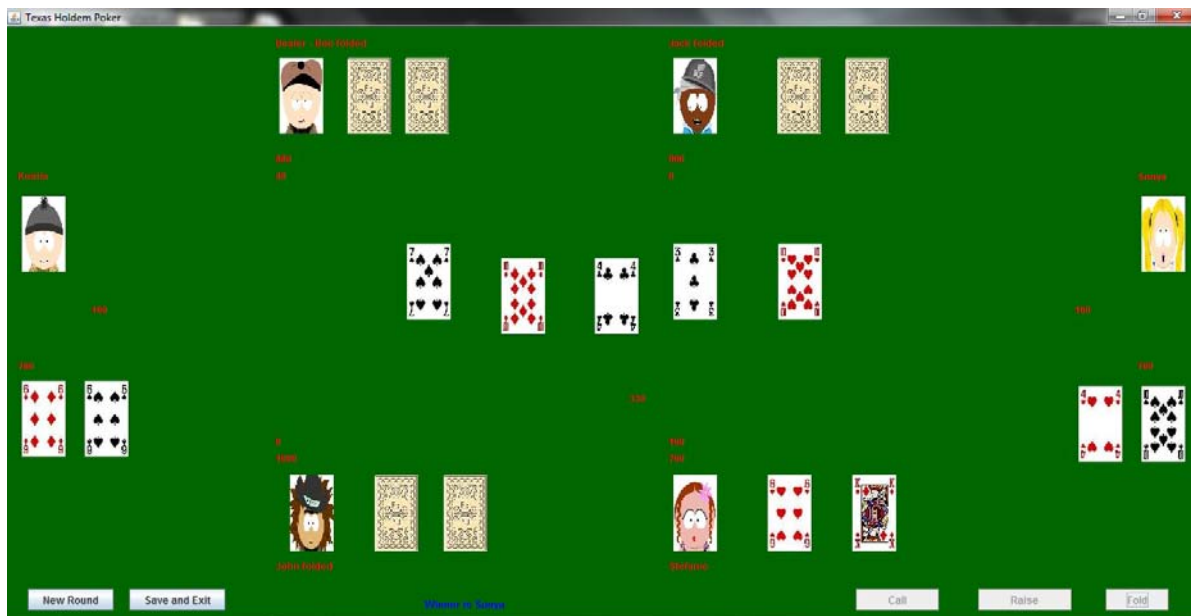


4.5 Ο επιλογέας αρχείου για σώσιμο παιχνιδιού

Όταν ο χρήστης πατήσει το κουμπί New Round ξεκινάει μία νέα παρτίδα. Μηδενίζονται οι τιμές και ο χρήστης βλέπει τα φύλλα του (εικόνα 4.6). Στους υπόλοιπους παίχτες τα φύλλα είναι κλειστά και ανοίγουν μόνο στο τέλος της παρτίδας για να φανεί ποιος έχει το καλύτερο φύλλο και κερδίζει. Στο κέντρο του παραθύρου ανοίγονται τα κοινά φύλλα για όλους τους παίχτες. (εικόνα 4.7).



4.6 Τα χαρτιά του χρήστη



4.7 Άνοιγμα των κοινών φύλλων

Πάνω ή κάτω από κάθε παίχτη, αναλόγως την θέση του στο παράθυρο, αναγράφεται το όνομά του. Από την άλλη μεριά φαίνεται ο αριθμός των διαθέσιμων μαρκών που έχει ο παίκτης στην κατοχή του ενώ δίπλα σε αυτό φαίνεται το τρέχον ποντάρισμα του. Κάτω από τα κοινά φύλλα φαίνεται το συνολικό ποσό του Pot που διεκδικούν οι παίχτες. Στην ετικέτα του ονόματος μπορεί να αναγράφεται και η λέξη Dealer αν ο παίκτης έχει τον ρόλο του dealer στην παρτίδα ή και η λέξη folded αν ο παίκτης έχει πάει πάσο από αυτήν.

Σε αυτήν την κλάση υλοποιούνται και οι μέθοδοι refreshUI() και revealCards(). Η πρώτη ανανεώνει κάποια συστατικά του παραθύρου όπως τις μάρκες των παιχτών, του πονταρίσματος και του pot, η οποία καλείται αρκετά συχνά όταν χρειάζεται ο χρήστης να ενημερωθεί σχετικά. Η δεύτερη καλείται στο τέλος της παρτίδας και φανερώνει τα φύλλα των αντιπάλων.

4.4 Οι ιδιότητες των κλάσεων

Σε αυτό το σημείο θα γίνει μια σύντομη περιγραφή των ιδιοτήτων κάθε κλάσης.

4.4.1 Game

- **int SmallBlind:** το ποσό του small blind.
- **int BigBlind:** το ποσό του big blind.
- **int Pot:** μεταβλητή που κρατάει το ποσό του pot.
- **int Dealer:** δείκτης του παίχτη που έχει τον ρόλο του dealer στον πίνακα των παιχτών.
- **int HighBet:** μεταβλητή που κρατάει το ποσό του μεγαλύτερου πονταρίσματος.
- **Int PlayersIn:** ο αριθμός των παιχτών στο παιχνίδι.
- **String Phase:** παίρνει τις τιμές Start, Flop, Turn, Phase ανάλογα με τη φάση του παιχνιδιού.
- **static int raises:** ο αριθμός των αυξήσεων πονταρίσματος σε ένα γύρο πονταρίσματος.

4.4.2 Player

- **String name:** το όνομα του παίχτη.
- **int chips:** ο αριθμός των διαθέσιμων μαρκών του παίχτη.

- **int bet:** ο αριθμός των πονταρισμένων μαρκών του παίχτη.
- **boolean folded:** ένδειξη για το αν ο παίχτης έχει πάει πάσο από μια παρτίδα ή όχι.
- **boolean out:** ένδειξη για το αν ο παίχτης έχει χάσει ή όχι.
- **boolean AllIn:** ένδειξη για το αν ο παίχτης έχει ποντάρει όλες του τις μάρκες ή όχι.

4.4.3 Person

- **int flag:** βοηθητική μεταβλητή που παίρνει της τιμές 1, 2 ή 3 ανάλογα με την επιλογή του παίχτη στη σειρά του.

4.4.4 Computer

- **Random rn:** αντικείμενο της κλάσης Random του πακέτου java.util για την παραγωγή τυχαίων αριθμών.

4.4.5 Card

- **String Scheme:** μεταβλητή που κρατάει το νούμερο και το χρώμα της κάρτας.

4.4.6 Deck

- **int pointer:** βοηθητική μεταβλητή που λειτουργεί σαν δείκτης για το επόμενο φύλλο που θα τραβηχτεί από την τράπουλα.

4.5 Οι μέθοδοι των κλάσεων

Παρακάτω καταγράφονται οι μέθοδοι για κάθε κλάση μαζί με τις παραμέτρους τις και μια περιγραφή της λειτουργίας τους.

4.5.1 Game

- **void PlayRound():** χειρίζεται τις διαδικασίες για την ολοκλήρωση μιας παρτίδας. Καλείται με το πάτημα του κουμπιού New Round και ολοκληρώνεται με το τέλος της παρτίδας.
- **void NewRound():** η υλοποίηση της αντίστοιχης περίπτωσης χρήσης. Ανακατεύει την τράπουλα, μοιράζει τα κρυφά φύλλα στους παίχτες και θέτει τις αρχικές εξ ορισμού τιμές.

- **int getFoldedWinner():** υπολογίζει αν υπάρχει νικητής στην παρτίδα ο οποίος προέκυψε επειδή όλοι οι άλλοι πήγαν πάσο. Επιστρέφει τον δείκτη του νικητή στον πίνακα των παιχτών ή -1 αν δεν υπάρχει.
- **int getWinner():** υπολογίζει τον νικητή της παρτίδας ανάλογα με το ποιός έχει το καλύτερο κόλπο στο χέρι του και επιστρέφει τον δείκτη του στον πίνακα των παιχτών.
- **boolean PlayerLost():** επιστρέφει true ή false ανάλογα αν ο χρήστης έχει χάσει στο παιχνίδι ή όχι.
- **void nextDealer():** θέτει τον δείκτη του παίχτη που έχει σειρά να πάρει τον ρόλο του dealer στο παιχνίδι.
- **void removeOutPlayers():** υπολογίζει αν κάποιος παίχτης έχει χάσει και μετακινεί στις πρώτες θέσεις του πίνακα παιχτών τους εναπομείναντες και μειώνει ανάλογα την τιμή της μεταβλητής PlayersIn.
- **boolean ifBetRoundOk():** επιστρέφει true ή false ανάλογα για τον αν έχει ολοκληρωθεί ο γύρος πονταρίσματος ή αν κάποιος παίχτης πρέπει να πάρουν αποφάσεις.
- **double getPlayerHandStrength(Player):** υπολογίζει μια τιμή για κάποιον παίχτη που υποδηλώνει το πόσο δυνατό είναι το κόλπο του.
- **double getHandStrength(Card[]):** υπολογίζει μια τιμή ανάλογα για μία πεντάδα ή δυάδα καρτών που υποδηλώνει το πόσο δυνατό είναι το φύλλο του συνδυασμού.
- **boolean isFlushRoyale(Card[]):** επιστρέφει true ή false ανάλογα αν η πεντάδα φύλλων που δέχεται είναι κόλπο τύπου Flush royale.
- **boolean isStraightFlush(Card[]):** επιστρέφει true ή false ανάλογα αν η πεντάδα φύλλων που δέχεται είναι κόλπο τύπου Straight flush.
- **boolean isFourOfAKind(Card[]):** επιστρέφει true ή false ανάλογα αν η πεντάδα φύλλων που δέχεται είναι κόλπο τύπου Four of a kind.
- **boolean isFlush(Card[]):** επιστρέφει true ή false ανάλογα αν η πεντάδα φύλλων που δέχεται είναι κόλπο τύπου Flush.
- **boolean isFullHouse(Card[]):** επιστρέφει true ή false ανάλογα αν η πεντάδα φύλλων που δέχεται είναι κόλπο τύπου Full house.
- **boolean isStraight(Card[]):** επιστρέφει true ή false ανάλογα αν η πεντάδα φύλλων που δέχεται είναι κόλπο τύπου Straight.

- **boolean isThreeOfAKind(Card[]):** επιστρέφει true ή false ανάλογα αν η πεντάδα φύλλων που δέχεται είναι κόλπο τύπου Three of a kind.
- **boolean isTwoPairs(Card[]):** επιστρέφει true ή false ανάλογα αν η πεντάδα φύλλων που δέχεται είναι κόλπο τύπου Two pairs.
- **boolean isTwoOfAKind(Card[]):** επιστρέφει true ή false ανάλογα αν η πεντάδα φύλλων που δέχεται είναι κόλπο τύπου Two of a kind.
- **Card[][] getCardCombinations(Card[]):** δέχεται σαν είσοδο έναν πίνακα από πέντε έως επτά φύλλα και επιστρέφει σε δισδιάστατο πίνακα καρτών τους συνδυασμούς πεντάδων. Βοηθάει για να ανακαλυφτεί ο καλύτερος συνδυασμός σε κάθε φάση.
- **Card[] SortHandByStrength(Card[]):** δέχεται έναν πίνακα με φύλλα και τον επιστρέφει ταξινομημένο κατά φθίνουσα σειρά της ισχύος του φύλλου.

4.5.2 Player

- **abstract void play():** αφηρημένη μέθοδος που πρέπει να υλοποιηθεί από τις κλάσεις Person και Computer.
- **void bet(int):** δέχεται σαν παράμετρο τον αριθμό των μαρκών που ποντάρει ο παίχτης, τον προσθέτει στην μεταβλητή bet και τον αφαιρεί από την μεταβλητή chips. Είναι μέρος της αντίστοιχης περίπτωσης χρήσης.
- **void fold():** θέτει true την μεταβλητή folded. Είναι μέρος της αντίστοιχης περίπτωσης χρήσης.

4.5.3 Person

- **void play():** υλοποίηση της περίπτωσης χρήσης play για τον χρήστη. Η μέθοδος περιμένει μέχρι να πατηθεί κάποιο από τα κουμπιά Call, Raise ή Fold τα οποία αλλάζουν την τιμή του flag. Καλείται ανάλογα η μέθοδος bet() ή fold() και τερματίζει η μέθοδος.

4.5.4 Computer

- **int CalculateConfidence():** υπολογίζει μία τιμή από το 1 έως το 300 που υποδεικνύει πόσο ο παίχτης πιστεύει στο φύλλο του. Υπολογίζεται ανάλογα με το κόλπο του, τη φάση του παιχνιδιού και τα φύλλα που κρατάει στο χέρι.

- **void play():** ο παίχτης του υπολογιστή κάνει την κίνησή του. Η απόφαση που θα πάρει είναι ανάλογη, κυρίως, της τιμής του confidence.

4.5.5 Card

- **int getStrength():** επιστρέφει μία τιμή που υποδεικνύει τη θέση του νούμερου που έχει το φύλλο στην ιεραρχία των καρτών. Για παράδειγμα το '10' θα επιστρέψει τιμή μεγαλύτερη του '7'.
- **char getColor():** επιστρέφει τον χαρακτήρα του χρώματος του φύλλου. 'D' για καρό, 'H' για καρδιά, ομοίως και για τα υπόλοιπα.
- **char getNumber():** επιστρέφει τον χαρακτήρα του νούμερου του φύλλου. 'K' για παπά, '10' για το 10, ομοίως και για τα υπόλοιπα.

4.5.6 Deck

- **void Shuffle():** ανακατεύει την τράπουλα.
- **Card Draw():** τραβάει το πάνω φύλλο της τράπουλας και το επιστρέφει.
- **void createCards():** βοηθητική μέθοδος που δημιουργεί τις κάρτες της τράπουλας.

4.6 Ο υπολογισμός της δύναμης του φύλλου

Η δύναμη του κόλπου μιας πεντάδας φύλλων υπολογίζεται σε μια τιμή strength. Αυτή η τιμή μας βοηθάει στο να υπολογίσουν οι παίχτες του υπολογιστή το τι κίνηση θα κάνουν στη σειρά τους, αλλά κατά κύριο λόγο στο να βρεθεί ο νικητής μιας παρτίδας.

Η τιμή strength είναι δεκαδική τιμή και κυμαίνεται μεταξύ του 0 και του 10'000. Όταν μια πεντάδα φύλλων δεν βγάζει κάποιο κόλπο, όπως ένα ζευγάρι ή μία κέντα (Straight) τότε η τιμή strength θα είναι πάντα κάτω από 1000. Η ακριβής τιμή υπολογίζεται ανάλογα με τα φύλλα. Για παράδειγμα, όταν δυο παίχτες, χωρίς κάποιο κόλπο, διεκδικούν τη νίκη τότε κερδίζει αυτός που έχει το ψηλότερο φύλλο. Αν κι εκεί υπάρχει ισοπαλία, τότε μετράται το δεύτερο ψηλότερο φύλλο και ούτω καθεξής. Έτσι, η τιμή strength υπολογίζεται από τον τύπο $(\text{δύναμη}_1 \cdot 10'000 + \text{δύναμη}_2 \cdot 1'000 + \text{δύναμη}_3 \cdot 100 + \text{δύναμη}_4 \cdot 10 + \text{δύναμη}_5) / 1000$, όπου δύναμη_η η δύναμη κάθε φύλλου στην ιεραρχία κατά φθίνουσα σειρά ώστε το ψηλότερο φύλλο

να έχει μεγαλύτερη βαρύτητα και στην σύγκριση των δύο strengths να κάνει από μόνο του τη διαφορά.

Όταν το κόλπο είναι Two of a kind το strength θα είναι τουλάχιστον 1'000 και ποτέ πάνω 2'000. Η ακριβής τιμή υπολογίζεται θέτοντας μέγιστη βαρύτητα στο νούμερο που έχει το ζευγάρι και ακολουθούν τα υπόλοιπα ξανά κατά φθίνουσα σειρά ισχύος. Κατά όμοιο τρόπο, κάθε κόλπο καταλαμβάνει το εύρος μιας χιλιάδας τιμών ανάλογα με την ισχύ του καταλήγοντας στο Flush royale που έχει καθορισμένο strength ίσο με 10'000.

4.7 Η τεχνητή νοημοσύνη του παιχνιδιού

Κάθε παίχτης που χειρίζεται ο υπολογιστής παίρνει τις αποφάσεις του σύμφωνα με μια μέθοδο τεχνητής νοημοσύνης. Η βάση αυτής είναι μία τιμή που λέγεται confidence και υπολογίζεται με την μέθοδο CalculateConfidence() στην κλάση Computer. Η τιμή αυτή είναι ένας ακέραιος αριθμός και κυμαίνεται μεταξύ 0 και 300 η οποία υπολογίζεται κάθε φορά που είναι η σειρά του παίχτη να παίξει και πρέπει να πάρει κάποια απόφαση. Η ακριβής τιμή εξαρτάται από τα φύλλα που κρατάει ο παίχτης στο χέρι, από την φάση του παιχνιδιού, από το κόλπο που βγάζει και τον αριθμό των αυξήσεων πονταρίσματος που έχουν γίνει.

Η απόφαση που θα πάρει ο παίχτης για την κίνηση του καθορίζεται από την τιμή του confidence και έναν τυχαίο αριθμό μεταξύ 0 και 99. Όταν το confidence είναι κάτω από 100, τότε βγαίνει ένας τυχαίος αριθμός μεταξύ 0 και 99 και αν αυτός είναι μεγαλύτερος του confidence, τότε ο παίχτης πάει πάσο, αλλιώς βλέπει το ποντάρισμα. Αν είναι μεταξύ 100 και 199 τότε γίνεται η ίδια διαδικασία μόνο που ο παίχτης στην μία περίπτωση θα δει απλώς το ποντάρισμα ενώ στην άλλη θα το αυξήσει περαιτέρω. Όταν το confidence είναι πάνω από 200 τότε υπάρχουν μικρές πιθανότητες να μην αυξήσει το ποντάρισμα. Όταν η διαδικασία έχει σαν αποτέλεσμα να πάει ο παίχτης πάσο, τότε υπολογίζεται μία πιθανότητα ο παίχτης να καταλήξει σε μπλόφα βλέποντας ή ακόμα και αυξάνοντας το ποντάρισμα.

4.8 Επίλογος

Σε αυτό το κεφάλαιο, είδαμε τις κλάσεις που απαρτίζουν το έργο, τις ιδιότητες και τις λειτουργίες τους καθώς και κάποιες ειδικές μηχανικές του έργου. Μετά την ολοκλήρωση υπολογίστηκαν οι μετρικές αντικειμενοστραφούς σχεδίασης

για το έργο αυτό και τα αποτελέσματα δεν ήταν ιδιαίτερα ικανοποιητικά. Λεπτομέρειες για τις μετρικές αυτές και για την γενική αξιολόγηση του έργου θα δούμε στο έκτο κεφάλαιο τις αξιολόγησης.

Στο επόμενο κεφάλαιο, θα γίνει μια παρουσίαση σχετικά με τις προσπάθειες βελτίωσης του κώδικα ώστε να είναι πιο ευέλικτος σε αλλαγές και προσθήκες.

Κεφάλαιο 5 Συντήρηση/Επέκταση

5.1 Εισαγωγή

Μετά τον υπολογισμό των μετρικών του αρχιτεκτονικού σχεδίου είδαμε ότι το έργο απαρτίζεται από πολλές μεθόδους με υψηλά επίπεδα πολυπλοκότητας, υψηλή σύζευξη μεταξύ των κλάσεων και από χαμηλές προοπτικές εξέλιξης.

Σε αυτό το κεφάλαιο, θα δούμε τις δύο επόμενες εκδόσεις του λογισμικού με βελτιωμένο κώδικα που θα μειώνει τα προαναφερθέντα προβλήματα. Αυτές οι προσπάθειες δεν αφορούν αλλαγές στη διεπιφάνεια του χρήστη ή στις κλάσεις της, οπότε δε θα γίνει κάποιος λόγος για αυτή. Ακόμα δε γίνεται καμία αλλαγή στις περιπτώσεις χρήσης και στα διαγράμματα ακολουθίας συστήματος για αυτές.

5.2 Έκδοση 2.0

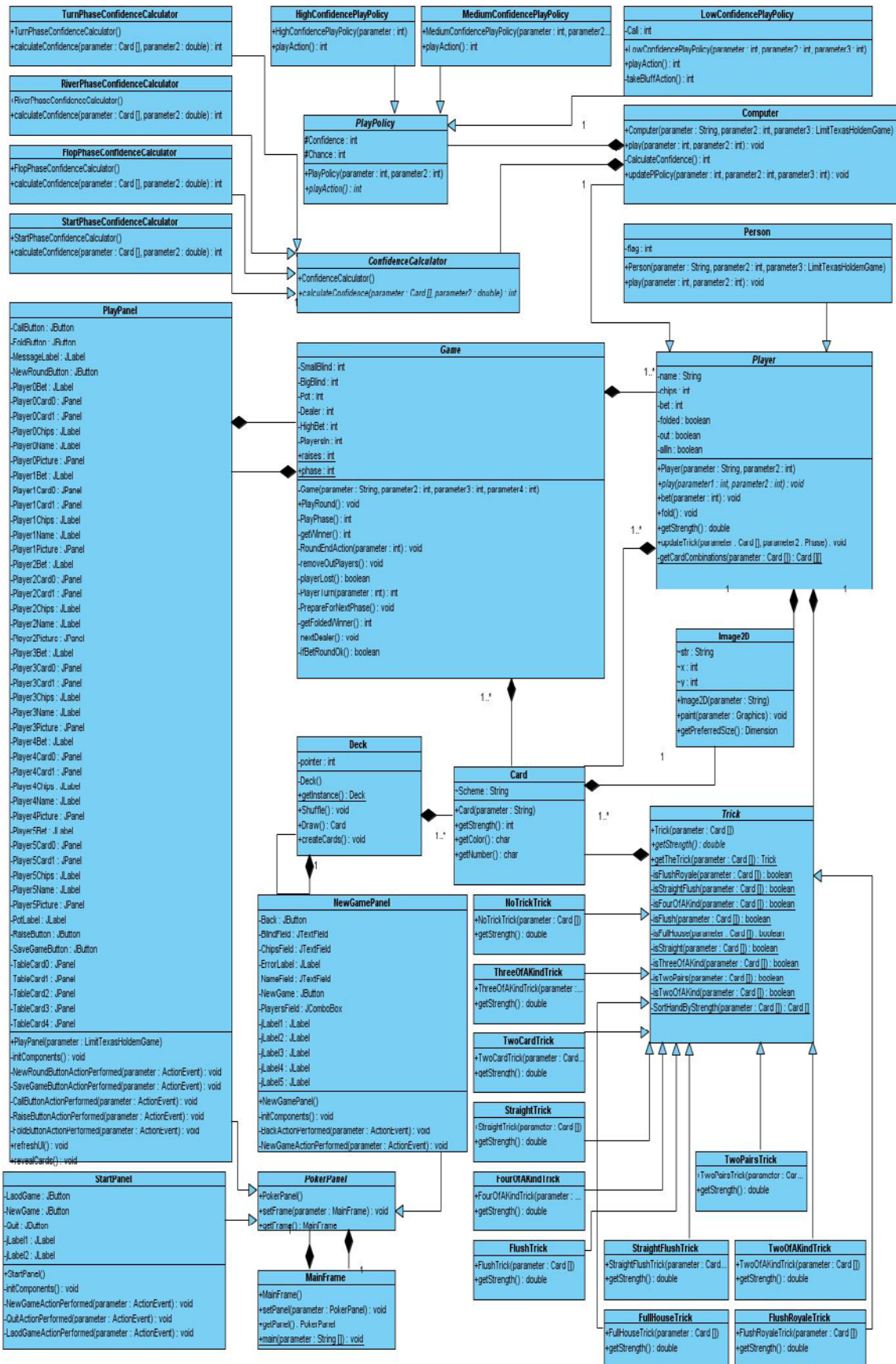
Σε αυτήν την έκδοση οι αλλαγές αφορούν κυρίως προσπάθειες μείωσης της πολυπλοκότητας κάποιων μεθόδων. Οι αλλαγές που έγιναν επεξηγούνται παρακάτω.

5.2.1 Το διάγραμμα κλάσεων

Στο σχήμα 5.1 φαίνεται το διάγραμμα κλάσεων για την δεύτερη έκδοση του λογισμικού. Σε αυτό προστίθενται αρκετές νέες κλάσεις.

Καταρχήν, θεωρήθηκε ότι ο υπολογισμός του strength του κόλπου κάθε παίχτη δεν πρέπει να είναι ευθύνη της κλάσης Game. Πλέον το κόλπο κάθε παίχτη θα απεικονίζεται με την αφηρημένη κλάση Trick. Την κλάση αυτή θα κληρονομούν έντεκα κλάσεις, μία για κάθε ένα από τα εννιά κόλπα που υπάρχουν, μία για την περίπτωση που δεν βγαίνει κάποιο κόλπο και μία για την πρώτη φάση που κάθε παίχτης έχει στη διάθεσή του μόνο τα δύο κρυφά φύλλα στο χέρι.

Η μέθοδος CalculateConfidence() χαρακτηρίστηκε από πολύ υψηλή πολυπλοκότητα, μέσω της μετρικής 'κυκλωματική πολυπλοκότητα', και έτσι γεννήθηκε η ανάγκη για βελτιστοποίηση. Στο διάγραμμα κλάσεων φαίνεται η αφηρημένη κλάση ConfidenceCalculator που συνθέτεται μέσα στην κλάση Computer. Την κλάση αυτή κληρονομούν τέσσερις κλάσεις μία για κάθε φάση του παιχνιδιού, οπότε διαχωρίζουμε τον τρόπο υπολογισμού του confidence ανάλογα με την φάση του παιχνιδιού.



5.1 Διάγραμμα κλάσεων δεύτερης έκδοσης

Επιπλέον, η μέθοδος `play()` στην κλάση `Computer` είχε υψηλή πολυπλοκότητα λόγω των πολλών εμφωλευμένων `'if'` στον κώδικα της μεθόδου. Αυτό καταπολεμήθηκε δημιουργώντας την αφηρημένη κλάση `PlayPolicy`, την οποία κληρονομούν τρεις κλάσεις ανάλογα με την τιμή του `confidence`. Οι τρεις κλάσεις χειρίζονται την πολιτική υπολογισμού της απόφασης ανάλογα με το ύψος του `confidence`.

Τέλος, στο διάγραμμα φαίνεται ότι η κλάση `Deck` συνδέεται πλέον με τον εαυτό της. Η σύνδεση αυτή είναι λόγω της εφαρμογής του προτύπου σχεδίασης μοναδιαίο. Περισσότερα για αυτό θα πούμε στο επόμενο κεφάλαιο.

5.2.2 Οι ιδιότητες των κλάσεων

Παρακάτω θα γίνει μια αναφορά στις αλλαγές και στις προσθήκες των ιδιοτήτων των κλάσεων, καθώς και μια περιγραφή αυτών.

5.2.2.1 Game

Σε αυτήν την κλάση αλλάζει η ιδιότητα `phase` από `String` σε `int` ώστε η φάση να αποθηκεύεται σαν αριθμός και γίνεται και ένα ανάλογο `enumeration`.

5.2.2.2 Player

Εδώ δεν υπάρχουν διαφορές μόνο που πλέον κρατάει και ένα αντικείμενο τύπου `Trick` που αντιπροσωπεύει το κόλπο που βγάζει ο παίχτης.

5.2.2.3 Person

Για αυτήν την κλάση δεν υπάρχουν αλλαγές στις ιδιότητες της.

5.2.2.4 Computer

Δεν έχει καθόλου ιδιότητες παρά μόνο δύο αντικείμενα τύπου `PlayPolicy` και `CalculateConfidence`.

5.2.2.5 Card και Deck

Σε αυτές τις κλάσεις δεν γίνονται αλλαγές στις ιδιότητές τους.

5.2.2.6 Trick

Η κλάση Trick και όλες οι υποκλάσεις της παίρνουν σαν όρισμα στον δομητή τους έναν πίνακα καρτών. Αυτός αντιπροσωπεύει την πεντάδα καρτών που αποτελούν το κόλπο ή την δυάδα για την περίπτωση της υποκλάσης TwoCardTrick. Αυτός ο πίνακας αποθηκεύεται σαν ιδιότητα στο αντικείμενο. Οι υποκλάσεις δεν έχουν επιπλέον ιδιότητες.

5.2.2.7 ConfidenceCalculator

Η κλάση ConfidenceCalculator και οι υποκλάσεις της δεν έχουν δικές τους ιδιότητες.

5.2.2.8 PlayPolicy

Η κλάση PlayPolicy και οι υποκλάσεις της έχουν δύο ιδιότητες τις confidence και chance. Η πρώτη είναι το confidence του παίχτη και η δεύτερη η τυχαία τιμή για τον καθορισμό της απόφασης. Και οι δύο είναι ακέραιοι αριθμοί.

5.2.3 Οι μέθοδοι των κλάσεων

Παρακάτω παρουσιάζονται οι νέοι μέθοδοι των κλάσεων και κάποιες παλιές που μετακινήθηκαν σε άλλες κλάσεις ή παρέμειναν ίδιες.

5.2.3.1 Game

- **void PlayRound():** η μέθοδος αυτή λόγω πολύ υψηλής πολυπλοκότητας διασπάστηκε σε πολλές μεθόδους οι οποίες φαίνονται παρακάτω. Η τελική της λειτουργία παρέμεινε ίδια.
- **int PlayPhase():** χειρίζεται τις διαδικασίες διεξαγωγής μίας φάσης της παρτίδας και επιστρέφει την τιμή -1 ή 0 ανάλογα με το αν πρέπει να τερματιστεί η PlayRound() ή όχι.
- **int getWinner():** όπως και στην πρώτη έκδοση.
- **void RoundEndAction(int):** δέχεται σαν παράμετρο τον δείκτη του νικητή στον πίνακα παιχτών και εκτελεί όλες τις λειτουργίες για τον τερματισμό της παρτίδας, όπως την πρόσθεση του rot στο σύνολο των μαρκών του νικητή.
- **void removeOutPlayers():** όπως και στην πρώτη έκδοση.
- **boolean playerLost():** όπως και στην πρώτη έκδοση.

- **int PlayerTurn(int):** δέχεται σαν παράμετρο τον δείκτη του παίχτη που έχει σειρά να παίξει και χειρίζεται την σειρά του.
- **void PrepareForNextPhase():** εκτελεί όλες τις απαραίτητες λειτουργίες πριν ξεκινήσει η επόμενη φάση, όπως το άνοιγμα του επόμενου κοινού φύλλου και τον μηδενισμό των τιμών bet στους παίχτες.
- **int getFoldedWinner():** όπως και στην πρώτη έκδοση.
- **void nextDealer():** όπως και στην πρώτη έκδοση.
- **boolean ifBetRoundOk():** όπως και στην πρώτη έκδοση.

5.2.3.2 Player

- **abstract void play(int,int):** η μέθοδος πλέον δέχεται δύο παραμέτρους και οι δύο ακέραιοι. Η πρώτη είναι το HighBet και η δεύτερη το BigBlind από την Game. Αυτό γιατί πλέον η κλάση Player δεν κρατάει αναφορά προς το παιχνίδι οπότε και παίρνει αυτά τα στοιχεία σαν παραμέτρους.
- **void bet(int):** όπως και στην πρώτη έκδοση.
- **void fold():** όπως και στην πρώτη έκδοση.
- **double getStrength():** καλεί την μέθοδο getStrength() της κλάσης Trick.
- **void updateTrick():** ανανεώνει το στιγμιότυπο της ιδιότητας trick στο τρέχον αντικείμενο με τα νέα δεδομένα.
- **Card[][] getCardCombinations(Card[]):** η ίδια μέθοδος της κλάσης Game που μεταφέρθηκε εδώ.

5.2.3.3 Person

Σε αυτήν την κλάση δεν έγιναν αλλαγές στις μεθόδους της.

5.2.3.4 Computer

- **void play():** η απόφαση πλέον παίρνεται με την βοήθεια της μεθόδου playAction() της κλάσης PlayPolicy.
- **int CalculateConfidence():** καλεί την μέθοδο CalculateConfidence() της κλάσης ConfidenceCalculator.
- **void updatePIPolicy():** ανανεώνει το στιγμιότυπο της ιδιότητας PIPolicy στο τρέχον αντικείμενο ανάλογα με τη φάση του παιχνιδιού.

5.2.3.5 Card και Deck

Σε αυτές τις κλάσεις δεν γίνονται αλλαγές στις μεθόδους τους.

5.2.3.6 Trick

Σε αυτήν την κλάση γίνεται ένα enumeration για κάθε κόλπο που υπάρχει. Οι εννέα μέθοδοι της κλάσης Game, που υπολόγιζαν μία πεντάδα φύλλων αν είναι κάποιο από τα κόλπα, υλοποιούνται πλέον στην κλάση Trick.

- **int getTheTrick(Card[]):** δέχεται σαν παράμετρο μία πεντάδα φύλλων και επιστρέφει το νούμερο του κόλπου που βγάζουν σύμφωνα με το enumeration που έγινε.
- **double getStrength():** αφηρημένη μέθοδος που υλοποιείται από κάθε υποκλάση και υπολογίζει το strength των φύλλων που έχει σαν ιδιότητα το αντικείμενο.

5.2.3.7 ConfidenceCalculator

- **int CalculateConfidence(Card[],double):** αφηρημένη μέθοδος που υλοποιείται από κάθε υποκλάση. Δέχεται σαν παραμέτρους τα δύο φύλλα που έχει ο παίχτης στα χέρια και το strength του κόλπου του και υπολογίζει την τιμή του confidence.

5.2.3.8 PlayPolicy

- **int playAction():** αφηρημένη μέθοδος που υλοποιείται από όλες τις υποκλάσεις. Επιστρέφει 0 όταν η απόφαση είναι ο παίχτης να πάει πάσο, 1 όταν είναι να δει το ποντάρισμα και 2 όταν το αυξήσει. Η κλάση LowConfidencePlayPolicy υλοποιεί και την μέθοδο takeBluffAction που κάνει τους υπολογισμούς της μπλόφας.

5.3 Έκδοση 3.0

Στην τρίτη έκδοση του λογισμικού οι βελτιώσεις που έγιναν ήταν για την περεταίρω μείωση της πολυπλοκότητας κάποιων μεθόδων αλλά κυρίως για να γίνει το λογισμικό πιο ανοιχτό σε επεκτάσεις. Για να πετύχουμε αυτό έγινε εφαρμογή γνωστών προτύπων σχεδίασης όπου ήταν αυτό δυνατόν. Περιγραφή για τα πρότυπα σχεδίασης θα γίνει στο επόμενο κεφάλαιο της αξιολόγησης. Παρακάτω, θα δούμε τις αλλαγές που έγιναν σε αυτήν την έκδοση.

5.3.1 Το διάγραμμα κλάσεων

Στο διάγραμμα κλάσεων της νέας έκδοσης προστέθηκαν μερικές ακόμα κλάσεις. Η ιδιότητα της κλάσης Game που καταχωρεί τη φάση του παιχνιδιού δεν είναι πλέον ακέραιος αριθμός, αλλά αντικείμενο μίας νέας κλάσης που λέγεται Phase. Η κλάση Phase είναι αφηρημένη και κληρονομείται από τέσσερις κλάσεις μία για κάθε φάση του παιχνιδιού. Αυτή η αλλαγή έγινε για να μη χρειάζεται πλέον να υπάρχει enumeration για τις φάσεις.

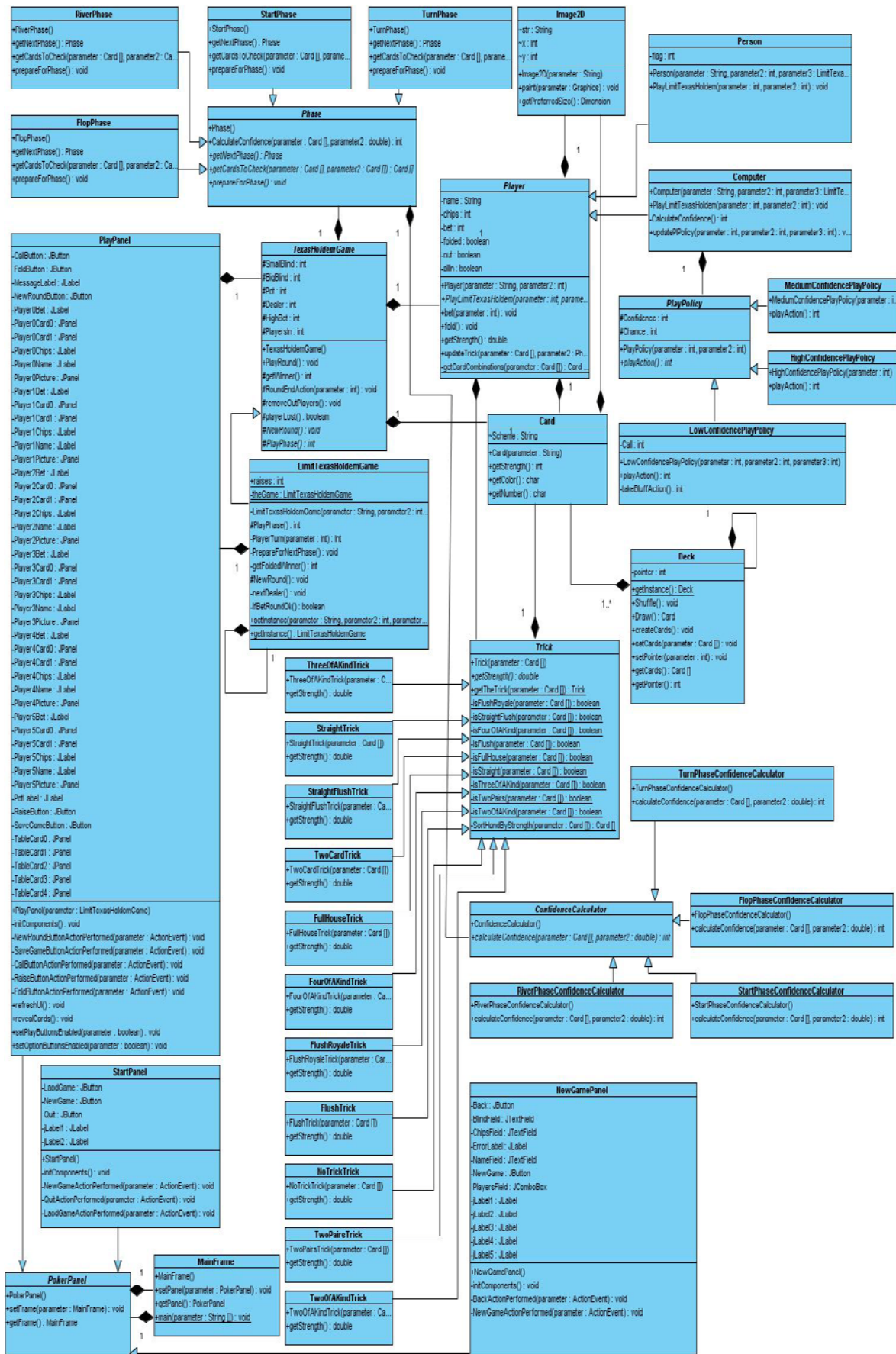
Η κλάση Game μετονομάζεται σε LimitTexasHoldemPoker που είναι και το όνομα του παιχνιδιού. Προστίθεται η αφηρημένη κλάση TexasHoldemPoker που κληρονομείται από την LimitTexasHoldemPoker. Ο λόγος της δημιουργίας αυτής της συσχέτισης είναι το να δοθεί μελλοντικά η δυνατότητα υλοποίησης ενός παιχνιδιού No Limit Texas Holdem, της δεύτερης παραλλαγής του Texas Holdem.

Μία ακόμα αλλαγή που φαίνεται στο διάγραμμα κλάσεων είναι ότι η κλάση ConfidenceCalculator συνδέεται πλέον με την κλάση Phase. Αυτό έγινε για λόγους ευκολίας υπολογισμού του confidence αφού ο βασικός τρόπος υπολογισμού του εξαρτάται από την φάση του παιχνιδιού.

Τέλος, η κλάση LimitTexasHoldemPoker συνδέεται με τον εαυτό της λόγω της χρήσης του προτύπου σχεδίασης μοναδιαίο. Το διάγραμμα κλάσεων για την Τρίτη έκδοση του λογισμικού φαίνεται στο σχήμα 5.2.

5.3.2 Οι ιδιότητες των κλάσεων

Δεν έχουμε πολλές αλλαγές στις ιδιότητες των κλάσεων αυτής της έκδοσης. Στην κλάση TexasHoldemPoker μεταφέρονται όλες οι ιδιότητες της πρώην κλάσης Game εκτός από την ιδιότητα raises. Οι δύο παραλλαγές του Texas Holdem δεν έχουν διαφορές σε αυτές τις ιδιότητες και για αυτό γίνεται η μεταφορά στην υπερκλάση. Οι υπόλοιπες κλάσεις δεν έχουν διαφορές στις ιδιότητές τους. Η κλάση Phase δεν έχει ιδιότητες παρά μόνο μία αναφορά προς ένα αντικείμενο τύπου ConfidenceCalculator.



5.2 Διάγραμμα κλάσεων τρίτης έκδοσης

5.3.3 Οι μέθοδοι των κλάσεων

Οι μέθοδοι `PlayRound()`, `getWinner()`, `RoundEndAction()`, `removeOutPlayers()` και `playerLost()` της κλάσης `LimitTexasHoldemPoker` μεταφέρονται στην υπερκλάση της λόγω κοινής συμπεριφοράς στις δύο παραλλαγές του παιχνιδιού. Οι υπόλοιπες δηλώνονται σαν αφηρημένες μέθοδοι στην υπερκλάση και παραμένουν ως είναι στην υποκλάση.

Η μέθοδος `calculateConfidence()` της κλάσης `Computer` πλέον καλεί την αντίστοιχη μέθοδο της `ConfidenceCalculator` μέσω της κλάσης `Phase` με την οποία πλέον συνδέεται.

Η μέθοδος `play()` της κλάσης `Player` μετονομάζεται σε `PlayLimitTexasHoldem()` λόγω εφαρμογής του προτύπου σχεδίασης γέφυρα. Η λειτουργίες της μεθόδου παραμένουν ίδιες.

Η νέα κλάση `Phase` έχει τέσσερις μεθόδους οι οποίες περιγράφονται παρακάτω:

- **`int calculateConfidence(Card[], double)`**: καλεί την αντίστοιχη μέθοδο της κλάσης `ConfidenceCalculator` η λειτουργία της οποίας παραμένει ίδια.
- **`Phase getNextPhase()`**: αφηρημένη μέθοδος που υλοποιείται από κάθε υποκλάση. Επιστρέφει ένα αντικείμενο μιας υποκλάσης της `Phase` ανάλογα με το ποια είναι η επόμενη φάση.
- **`Card[] getCardsToCheck(Card[], Card[])`**: βοηθητική αφηρημένη μέθοδος. Δέχεται σαν παράμετρο έναν πίνακα καρτών με τα φύλλα που έχει στα χέρια ένας παίχτης και έναν πίνακα καρτών με τα ανοιχτά φύλλα στο κέντρο του τραπέζιου. Επιστρέφει έναν πίνακα με πέντε έως επτά φύλλα, τα οποία είναι τα φύλλα που έχει ο κάθε παίχτης στη διάθεση του για να κάνει τον καλύτερο δυνατό συνδυασμό.
- **`void PrepareForPhase()`**: εκτελεί όλες τις απαραίτητες λειτουργίες για την έναρξη της συγκεκριμένης φάσης.

5.4 Επίλογος

Σε αυτό το κεφάλαιο, περιγράψαμε τις προσπάθειες για τη βελτιστοποίηση του κώδικα για το έργο που αναπτύσσουμε. Στις δύο νέες αυτές εκδόσεις πάρθηκαν, όπως και στην πρώτη, μετρικές αντικειμενοστραφούς σχεδίασης.

Στο επόμενο κεφάλαιο, θα γίνει η αξιολόγηση του έργου. Θα δούμε τις μετρικές αυτές όπως εξελίχτηκαν στις τρεις εκδόσεις και τι σημαίνουν αυτές για την ποιότητα του λογισμικού, καθώς και τα πρότυπα σχεδίασης που χρησιμοποιήθηκαν για το αρχιτεκτονικό σχέδιο και ποια είναι τα πλεονεκτήματα που προσφέρει η χρήση τους.

Κεφάλαιο 6 Αξιολόγηση

6.1 Εισαγωγή

Στα προηγούμενα κεφάλαια παρουσιάστηκε η διαδικασία συγγραφής του κώδικα του λογισμικού και οι προσπάθειες βελτιστοποίησης αυτού. Κάθε έκδοση του τρέχοντος έργου πέρασε από αξιολόγηση για να δούμε την ποιότητά του.

Η αξιολόγηση ενός έργου γίνεται με τον υπολογισμό των μετρικών αντικειμενοστραφούς σχεδίασης. Οι μετρικές αυτές αξιολογούν διάφορες πτυχές του έργου, όπως την πολυπλοκότητα των μεθόδων, το ύψος της σύζευξης μεταξύ των κλάσεων και άλλα. Σκοπός της αναδόμησης ενός λογισμικού είναι η βελτιστοποίηση των μετρικών αυτών όπου είναι δυνατόν.

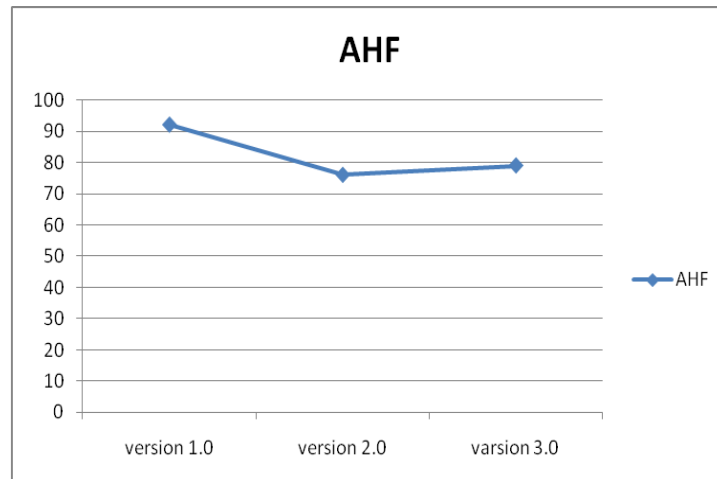
Σε αυτό το κεφάλαιο, θα δούμε τις μετρικές των τριών εκδόσεων του έργου όπως αυτές υπολογίστηκαν από τον κώδικα αυτού. Θα δούμε τα κύρια σημεία βελτίωσης με την χρήση γραφημάτων που θα μας δείχνουν πως αυτές εξελίχτηκαν από έκδοση σε έκδοση. Ακόμα, θα δούμε ποια πρότυπα σχεδίασης χρησιμοποιήθηκαν στις τρεις εκδόσεις και πως αυτά βελτιώνουν την ευελιξία του λογισμικού.

6.2 Μετρικές αντικειμενοστραφούς σχεδίασης

Υπάρχουν πολλές μετρικές οι οποίες αξιολογούν η καθεμιά διαφορετικές πτυχές του λογισμικού. Σε αυτό το σημείο θα παρουσιαστούν κάποιες από αυτές που υπολογίστηκαν για τις τρεις εκδόσεις αυτού του έργου.

6.2.1 Attribute Hiding Factor (AHF)

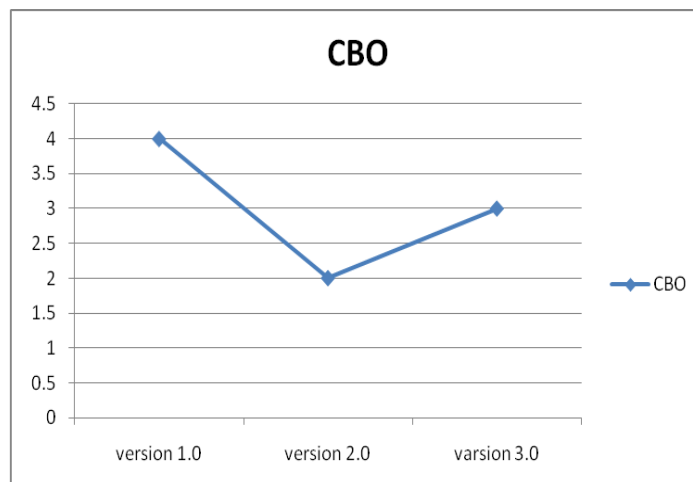
Αυτή η μετρική μετράει το κατά πόσο οι ιδιότητες των κλάσεων είναι αόρατες σε άλλες κλάσεις. Είναι μία γενική μετρική για όλο το έργο που δείχνει κατά μέσο όρο το ποσοστό μη ορατότητας των ιδιοτήτων. Το ζητούμενο είναι το ποσοστό να είναι υψηλό για πετύχουμε υψηλή ενθυλάκωση. Όπως φαίνεται στο σχήμα 6.1 η μετρική αυτή έχει πέσει από την πρώτη έκδοση στην δεύτερη και έγινε μια μικρή αύξηση στην τρίτη. Αυτό έγινε επειδή εισήχθησαν κάποιες public και protected μεταβλητές που στην πρώτη έκδοση δεν υπήρχαν. Παρόλα αυτά, το ποσοστό είναι ακόμα αρκετά υψηλό και δεν επηρεάζει αισθητά η μείωσή της.



6.1 Attribute Hiding Factor

6.2.2 Coupling Between Objects (CBO)

Αυτή η μετρική μετράει τις συσχετίσεις που έχει μία κλάση με άλλες κλάσεις. Οι συσχετίσεις περιλαμβάνουν συζεύξεις, πέρασμα παραμέτρων, επιστροφές τύπων και άλλες σχέσεις που μπορεί να συνδέουν δύο κλάσεις. Όταν μία κλάση έχει υψηλή αυτή την μετρική τότε είναι λιγότερο ανεξάρτητη και είναι πιο δύσκολο να επαναχρησιμοποιηθεί σε κάποιο άλλο έργο. Στο σχήμα 6.2 φαίνεται ο μέσος όρος τις μετρικής αυτής για τις τρεις εκδόσεις. Παρατηρούμε και εδώ μια πολύ καλή βελτίωση στην δεύτερη έκδοση με μια μικρή αύξηση στην τρίτη.

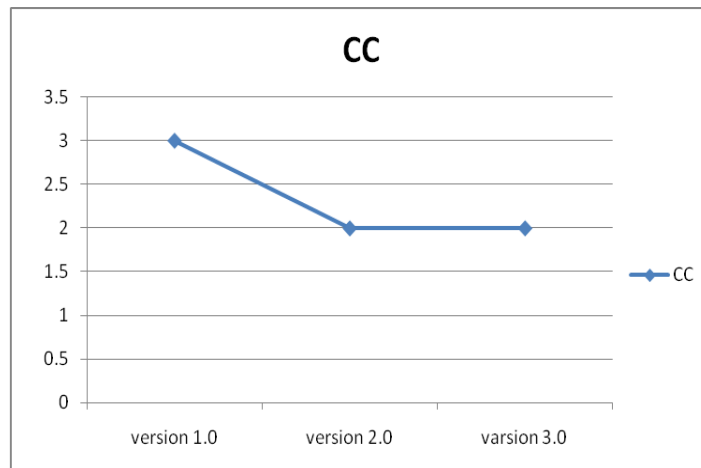


6.2 Coupling Between Objects

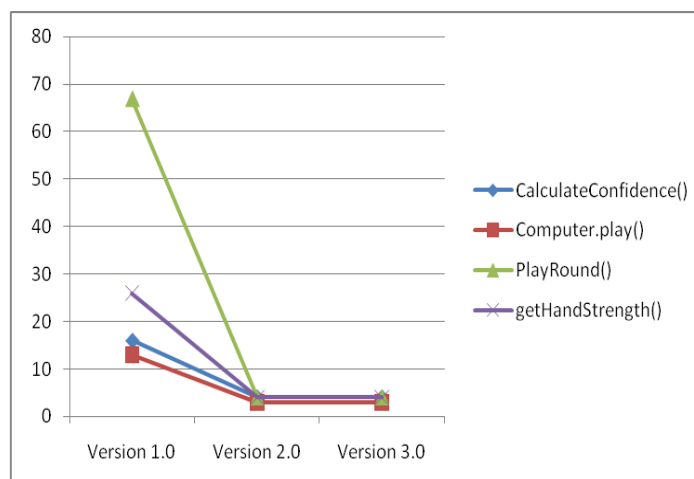
6.2.3 Cyclomatic Complexity (CC)

Πολύ σημαντική μετρική που μετράει την πολυπλοκότητα μιας μεθόδου. Η ακριβής τιμή της μετρικής είναι ο αριθμός των μονοπατιών που έχει μία μέθοδος

στο διάγραμμα ροής της. Όταν μία μέθοδος έχει πολλά μονοπάτια θεωρείται ιδιαίτερα πολύπλοκη. Στο τρέχον έργο είχαμε σημαντικές βελτιώσεις από έκδοση σε έκδοση. Στο σχήμα 6.3 βλέπουμε την εξέλιξη του μέσου όρου της μετρικής για το σύστημα, ενώ στο σχήμα 6.4 την εξέλιξη της μετρικής για τις τέσσερις πιο πολύπλοκες μεθόδους της πρώτης έκδοσης.



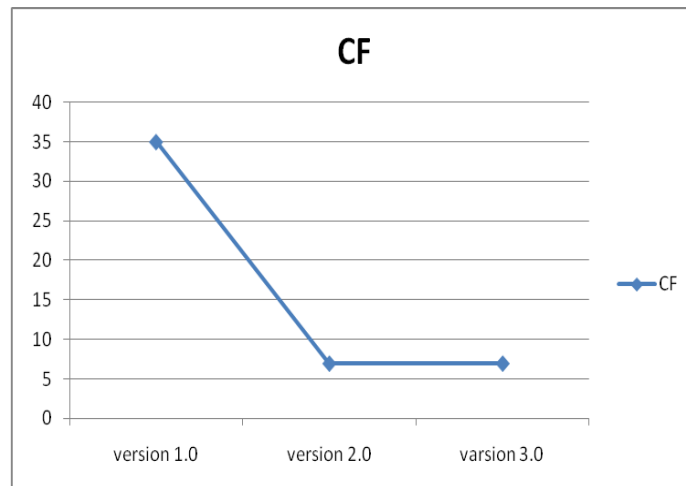
6.3 Cyclomatic Complexity / 1



6.4 Cyclomatic Complexity / 2

6.2.4 Coupling Factor (CF)

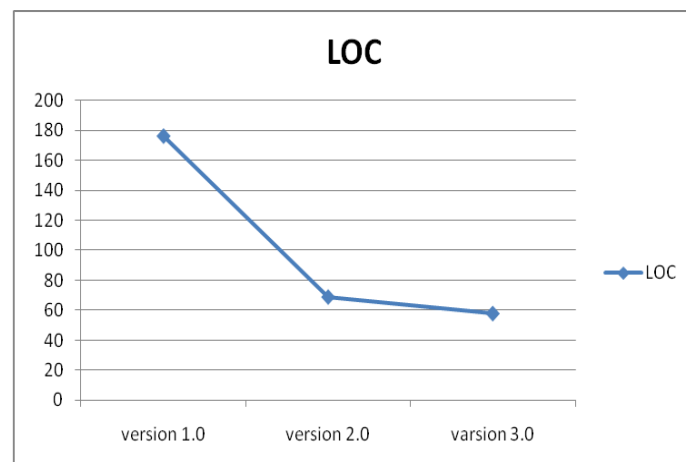
Η μετρική μετράει το ύψος της σύζευξης στο διάγραμμα κλάσεων εξαιρώντας τις σχέσεις κληρονομικότητας. Σκοπός ενός σχεδίου λογισμικού είναι η επίτευξη όσο το δυνατόν μικρότερης σύζευξης για να είναι το λογισμικό πιο ευέλικτο σε αλλαγές και προσθήκες. Στο τρέχον έργο η βελτίωση στις δύο τελευταίες εκδόσεις είναι εντυπωσιακή σε σχέση με την πρώτη. Αυτό οφείλεται στην αύξηση του αριθμού των κλάσεων. Η μετρική για της τρεις εκδόσεις φαίνεται στο σχήμα 6.5.



6.5 Coupling Factor

6.2.5 Lines Of Code (LOC)

Μετράει τις γραμμές του πηγαίου κώδικα. Στο σχήμα 6.6 φαίνεται ο μέσος όρος των γραμμών του κώδικα ανά κλάση. Η μείωση του σημαίνει πως στην πρώτη έκδοση υπήρχαν πολλές γραμμές περιττού κώδικα και πολλές γραμμές κώδικα σε λίγες κλάσεις, οι οποίες μπορούσαν να διασπαστούν όπως και έγινε.

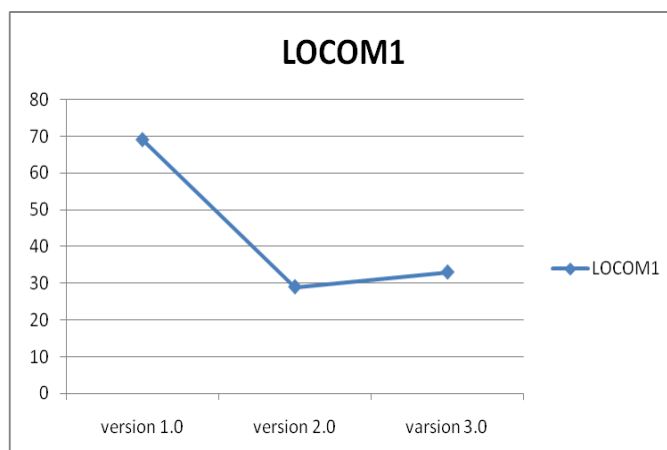


6.6 Lines Of Code

6.2.6 Lack Of Cohesion Of Methods 1 (LOCOM1)

Η μετρική μετράει την έλλειψη της συνεκτικότητας μεταξύ των μεθόδων μιας κλάσης. Δύο μέθοδοι είναι συνεκτικές όταν έχουν πρόσβαση σε κοινά δεδομένα της κλάσης. Η ακριβής τιμή είναι ο αριθμός των μη συνεκτικών ζευγών μείων τον αριθμό των συνεκτικών. Όταν η μετρική είναι υψηλή η κλάση θεωρείται ότι έχει χαμηλή συνεκτικότητα μεταξύ των μεθόδων της. Στο σχήμα 6.7 φαίνεται ο μέσος

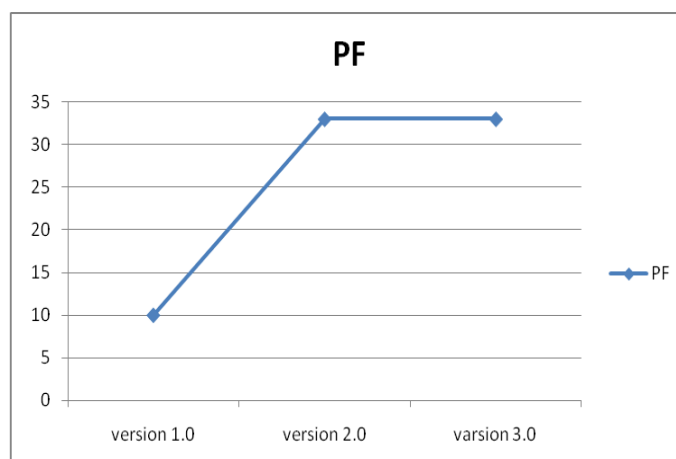
όρος της συνεκτικότητας των μεθόδων των κλάσεων. Έχουμε πολύ καλή βελτίωση της μετρικής από την πρώτη στη δεύτερη έκδοση με μια μικρή αύξηση στην τρίτη.



6.7 Lack Of Cohesion Of Methods

6.2.7 Polymorphism Factor (PF)

Η μετρική αυτή μετράει το επίπεδο πολυμορφισμού του συνολικού έργου. Υπολογίζεται σαν ποσοστό όπου ο αριθμητής είναι ο αριθμός των μεθόδων που υπερβαίνουν άλλες και ο παρανομαστής είναι ο αριθμός των πιθανών πολυμορφικών καταστάσεων. Το υψηλό ποσοστό της μετρικής σημαίνει υψηλό επίπεδο πολυμορφισμού. Η μετρική για το τρέχον έργο φαίνεται στο σχήμα 6.8 όπου φαίνεται η αύξηση του πολυμορφισμού στις μετέπειτα εκδόσεις.

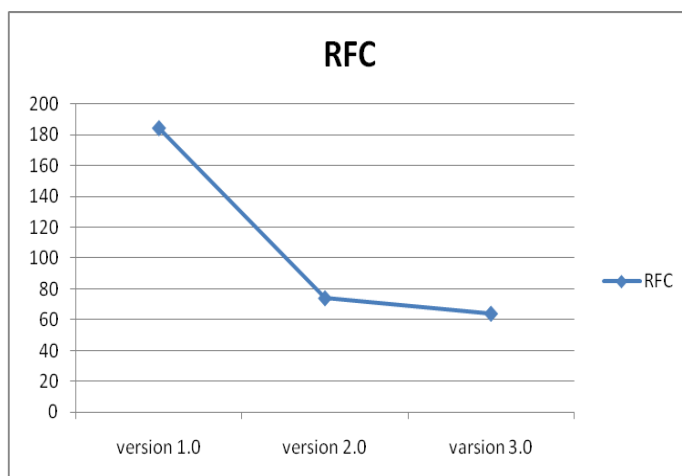


6.8 Polymorphism Factor

6.2.8 Response For Class (RFC)

Η μετρική αυτή μετράει πόσο πολύπλοκη και πόσο δύσκολο είναι μια κλάση να ελεγχθεί. Μεγάλη τιμή της μετρικής σημαίνει ότι η κλάση γενικά θέλει μεγαλύτερες προσπάθειες για έναν επιτυχή έλεγχο. Ο μέσος όρος της μετρικής

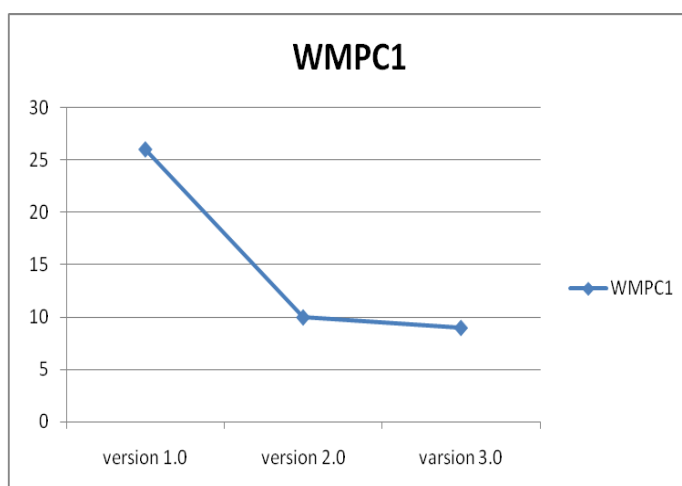
αυτής για κάθε έκδοση φαίνεται στο σχήμα 6.9 όπου βλέπουμε πολύ ικανοποιητική βελτίωση από έκδοση σε έκδοση.



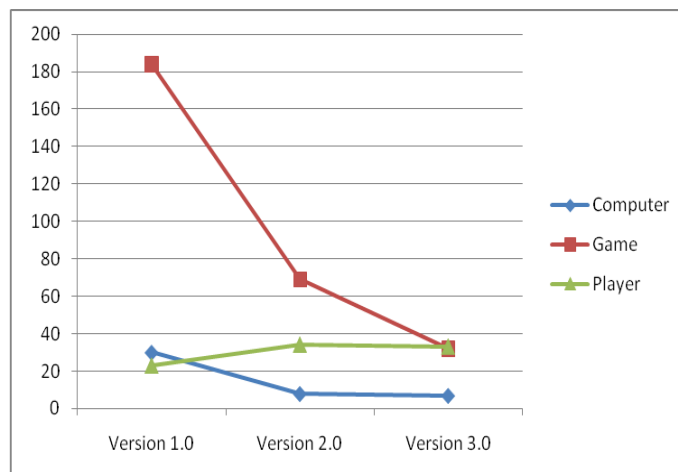
6.9 Response For Class

6.2.9 Weighted Methods Per Class 1 (WMPC1)

Μετράει την πολυπλοκότητα μιας κλάσης προσθέτοντας όλες τις μετρικές CC για κάθε μέθοδο της κλάσης. Στο σχήμα 6.10 φαίνεται ο μέσος όρος της μετρικής για το σύστημα στις τρεις εκδόσεις, όπου φαίνεται εξαιρετική βελτίωση. Στο σχήμα 6.11 βλέπουμε τη μετρική για τις κλάσεις Game, Computer και Player όπου και είχαμε τις πιο σημαντικές διαφορές. Ένα μειονέκτημα είναι η αύξηση της μετρικής για την κλάση Player.



6.10 Weighted Methods Per Class 1 / 1



6.11 Weighted Methods Per Class 1 /2

6.3 Πρότυπα σχεδίασης που εφαρμόστηκαν

Στις τρεις εκδόσεις του έργου εφαρμόστηκαν ορισμένα πρότυπα σχεδίασης. Σε αυτό το μέρος παρουσιάζονται τα πρότυπα αυτά, ο τρόπος με τον οποίο βελτιώνουν την ποιότητα του έργου και οι δυνατότητες που προσφέρει η εφαρμογή τους.

6.3.1 Έκδοση 1.0

Στην πρώτη έκδοση εφαρμόστηκε ένα μόνο στιγμιότυπο του προτύπου σχεδίασης 'στρατηγική'. Το συγκεκριμένο πρότυπο σχεδίασης ορίζει ότι κάποιες λειτουργίες μπορούν να υλοποιηθούν με διαφορετικούς τρόπους. Έτσι, υπάρχει μία αφηρημένη μέθοδος που υποδηλώνει μια λειτουργία και οι μέθοδοι που την υπερβαίνουν είναι οι διαφορετικοί τρόποι υλοποίησης της.

Στο τρέχον έργο, το πρότυπο εφαρμόζεται στην μέθοδο `play()` της κλάσης `Player` που υλοποιείται διαφορετικά για τις κλάσεις `Person` και `Computer`. Αυτό μας βοηθάει στο να μπορεί η κλάση `Game` να καλεί αυτήν την μέθοδο για κάθε παίχτη με τη σειρά χωρίς να την ενδιαφέρει αν είναι άνθρωπος ή υπολογιστής. Το ποιά από τις δύο υλοποιήσεις θα εκτελεστεί εξαρτάται από το στιγμιότυπο του αντικειμένου.

6.3.2 Έκδοση 2.0

Στην δεύτερη έκδοση χρησιμοποιήθηκαν περισσότερα πρότυπα σχεδίασης. Χρησιμοποιήθηκαν τέσσερα στιγμιότυπα του προτύπου 'στρατηγική' και ένα του προτύπου 'μοναδιαίο'. Το πρότυπο 'μοναδιαίο' χρησιμοποιείται σε κλάσεις για τις

οποίες ξέρουμε από την αρχή ότι χρειάζεται ένα μόνο αντικείμενο. Ορίζεται μία αναφορά σε αντικείμενο της κλάσης μέσα στην κλάση με `private` ορατότητα και δηλωμένο `static`, ορίζεται `private` και ο δομητής της κλάσης για να δημιουργήσει το στιγμιότυπο μέσα στην κλάση και ορίζεται και μία μέθοδος που θα επιστρέφει το στιγμιότυπο.

Στο έργο μας τα τέσσερα στιγμιότυπα του προτύπου 'στρατηγική' εφαρμόζονται για την μέθοδο `play()` όπως και στην πρώτη έκδοση, για την μέθοδο `getStrength()` της κλάσης `Trick` που καλείται από την `Player`, τη μέθοδο `calculateConfidence()` της κλάσης `ConfidenceCalculator` που καλείται από την `Computer` και τη μέθοδο `playAction()` της `PlayPolicy` που καλείται επίσης από την `Computer`. Όλες αυτές οι μέθοδοι δηλώνονται σαν αφηρημένες στις υπερκλάσεις και υλοποιούνται σε κάθε υποκλάση με διαφορετικό τρόπο.

Το στιγμιότυπο του προτύπου 'μοναδιαίο' εφαρμόζεται στην κλάση `Deck`. Από τις απαιτήσεις συστήματος είναι προφανές ότι για τη διεξαγωγή του παιχνιδιού χρειάζεται μία και μόνο τράπουλα.

6.3.3 Έκδοση 3.0

Στην τρίτη έκδοση εφαρμόστηκαν τα ίδια πρότυπα σχεδίασης με την δεύτερη με επιπλέον ένα στιγμιότυπο του προτύπου 'μέθοδος υπόδειγμα', ένα στιγμιότυπο του 'μοναδιαίο' και ένα ακόμα του 'στρατηγική'. Το πρότυπο 'μέθοδος υπόδειγμα' εφαρμόζεται όταν μια μέθοδος μιας υπερκλάσης καλεί μία μέθοδο της ίδιας κλάσης που υλοποιείται διαφορετικά για κάθε υποκλάση της. Αυτό σημαίνει ότι η πρώτη μέθοδος έχει κοινές λειτουργίες για όλες τις υποκλάσεις, αλλά κάποιο μέρος της, δηλαδή η μέθοδος που καλείται μέσα σε αυτήν, διαφέρει από υποκλάση σε υποκλάση. Έτσι, η μέθοδος που καλείται μέσα στην πρώτη δηλώνεται σαν αφηρημένη στην υπερκλάση και υπερβαίνεται στις υποκλάσεις.

Το νέο στιγμιότυπο 'στρατηγική' είναι η μέθοδος `prepareForPhase()` της κλάσης `Phase` που υλοποιείται διαφορετικά σε κάθε υποκλάση. Η μέθοδος `calculateConfidence()` καλείται πλέον από αυτήν την κλάση αλλά το πρότυπο μένει το ίδιο.

Το δεύτερο στιγμιότυπο του 'μοναδιαίο' αφορά την κλάση `LimitTexasHoldemPoker`. Χρειάζεται μόνο ένα στιγμιότυπο αντικειμένου αυτής της κλάσης αφού ο παίχτης δε μπορεί να παίζει ταυτόχρονα δύο παιχνίδια.

Το πρότυπο 'μέθοδος υπόδειγμα' εφαρμόζεται στην κλάση `TexasHoldemPoker`. Η μέθοδος `PlayRound()` υλοποιείται σε αυτήν την κλάση γιατί έχει κοινές λειτουργίες στις δύο παραλλαγές του Texas Holdem, ενώ κάποιες μέθοδοι που καλεί δηλώνονται αφηρημένες στην κλάση και υλοποιούνται από την `LimitTexasHoldemPoker`.

Σε περίπτωση που υλοποιηθεί ένα παιχνίδι No Limit Texas Holdem και η αντίστοιχη κλάση που κληρονομεί την `TexasHoldemPoker`, τότε στην κλάση `Player` και τις δύο υποκλάσεις πρέπει να υλοποιηθεί η μέθοδος `PlayNoLimitTexasHoldem`. Το σχέδιο λογισμικού αυτό υλοποιεί το πρότυπο 'γέφυρα'. Αυτό διαχωρίζει τις οντότητες από τις διάφορες υλοποιήσεις που έχουν. Όταν θα προστίθεται μία κλάση παιχνιδιού στο σχέδιο θα προστίθεται και μία μέθοδος στην κλάση `Player` για την υλοποίηση της κίνησης παιχνιδιού του παίχτη για το παιχνίδι αυτό. Αυτό το πρότυπο διευκολύνει σε εξαιρετικό βαθμό την επεκτασιμότητα του έργου.

6.4 Επίλογος

Η αξιολόγηση του έργου έδειξε ότι το έργο που υλοποιήσαμε ξεκίνησε όχι τις καλύτερες προοπτικές αλλά με την αναδόμηση του αρχιτεκτονικού του σχεδίου αποκτήσαμε ένα λογισμικό το οποίο μπορεί να ανταποκριθεί άριστα σε προσθήκες και επεκτάσεις. Η πολυπλοκότητα που το χαρακτηρίζει μειώθηκε κατά πολύ στην τρίτη έκδοση που σημαίνει ότι ένας αναγνώστης του κώδικα θα καταβάλει λιγότερο κόπο για την κατανόησή του.

Στη συνέχεια, ακολουθούν τα συμπεράσματα που εξάγαμε από την εκπόνηση αυτής της εργασίας.

Συμπεράσματα

Η εργασία αυτή στοχεύει στην ανάπτυξη λογισμικού που υλοποιεί ένα παιχνίδι πόκερ με την χρήση της γλώσσας προγραμματισμού Java. Η χρήση της γλώσσας αυτής αποδείχτηκε ιδιαίτερα χρήσιμη για τον λόγο του ότι χρησιμοποιήθηκαν ενσωματωμένα εργαλεία για την απόδοση των γραφικών που ήταν δύο διαστάσεων.

Η Java προσφέρει υψηλή φορητότητα στην εφαρμογή. Κάθε υπολογιστής που έχει εγκατεστημένο το Java Runtime Environment μπορεί να τρέξει την εφαρμογή χωρίς κάποια εγκατάστασή της στον υπολογιστή. Αρκούν τα .class αρχεία ή το .jar με όλες τις βιβλιοθήκες και τα αρχεία που χρησιμοποιεί.

Σαν αντικειμενοστραφής γλώσσα, επιτρέπει την εύκολη προσθήκη λειτουργιών στο σύστημα ή ακόμα και αλλαγές σε αυτό. Ο καλός διαχωρισμός των λειτουργιών των κλάσεων απλοποιεί τη διαδικασία συντήρησης του λογισμικού, όταν παρουσιάζεται κάποια ανάγκη για αλλαγή ή προσθήκη στον κώδικα.

Όλα αυτά καθιστούν απαραίτητη τη λεπτομερή ανάλυση του συστήματος και την «καλή» σχεδίαση της αρχιτεκτονικής, ώστε όλα τα πλεονεκτήματα της αντικειμενοστραφούς ανάπτυξης να φέρουν αποτελέσματα.

Στο συγκεκριμένο έργο, είδαμε ότι η σωστή αναδόμηση του λογισμικού (refactoring) βελτιώνει σημαντικά την ποιότητα του λογισμικού. Το αρχικό έργο είχε εξαιρετικά υψηλή πολυπλοκότητα στον κώδικα, όπως έδειξαν οι μετρικές. Μετά την αναδόμησή του στις δύο επόμενες εκδόσεις είδαμε τις μετρικές να βελτιώνονται σε σημαντικό επίπεδο. Με την προσθήκη περισσότερων κλάσεων μειώσαμε την πολυπλοκότητα τους διαμοιράζοντας σε αυτές τις αρμοδιότητες τηρώντας περισσότερο την αρχή της μοναδικής αρμοδιότητας. Στην τρίτη έκδοση μειώθηκαν και οι μη κληρονομούμενες συζεύξεις μεταξύ των κλάσεων όπως φαίνεται από την μετρική CF. Επίσης, παρατηρήθηκε ότι υπήρχαν πολλές περιττές συζεύξεις στην πρώτη έκδοση γεγονός που καθιστούσε το έργο κλειστό σε επεκτάσεις και δύσκολο να ελεγχθεί.

Εν κατακλείδι, η τήρηση των αρχών της μηχανικής λογισμικού κατά την ανάπτυξη ενός έργου είναι απαραίτητη, ώστε αργότερα να είναι δυνατή η αλλαγή, επέκταση και επαναχρησιμοποίηση του κώδικα.

Βιβλιογραφία

1. Chatzigeorgiou, A., (2005), Object Oriented Designing, pp 27-82.
2. Badcoe, I., (2000), The Computer Game Industry: Current State of Play, Infogrames, Sheffield, House, Sheffield, South Yorkshire, UK, pp 204-209.
3. Crooks, C., (2004), Awesome 3D Game Development, pp 2-6.
4. Jacob, M. , (2002), Evaluating Java for Game Development, Department of Computer Science, University of Copenhagen, Denmark, pp 75-87.
5. http://en.wikipedia.org/wiki/Game_development
6. <http://en.wikipedia.org/wiki/Poker>
7. <http://en.wikipedia.org/wiki/Rup>
8. <http://www.igda.org/Committees/education.htm>
9. <https://www.poker.gr/html/ent/008/ent.1008.asp>