



Πτυχιακή Εργασία

**«Προσομοίωση του επιτραπέζιου  
εκπαιδευτικού παιχνιδιού "Εναντίον του  
AIDS"»**

```
public void run() {  
    String tempStr = new String();  
    MainPanel mainPanel = frame.getMainPanel();  
    mainPanel.setGreenJokerNum(greenJokerNum);  
    Card topCard;  
    int luckyPlayer;  
    running = true;  
    mainPanel.repaint();  
    pauseTime(1500);  
}
```

Του φοιτητή  
Χατζηαναγνώστου Σωτήριου  
Αρ. Μητρώου : 1353

Επιβλέπων καθηγητής  
Παναγιώτης Αδαμίδης

## Περίληψη

Η εργασία «μεταφέρει» το επιτραπέζιο εκπαιδευτικό παιχνίδι "AIDS Counter AIDS" στον υπολογιστή. Πρόκειται για ένα παιχνίδι που προβάλλει όλα τα μέτρα πρόληψης για τον ιό HIV/AIDS ενθαρρύνοντας την συζήτηση και καταρρίπτοντας τις προκαταλήψεις και τους μύθους. Στόχος της παρούσας εργασίας είναι να δείξει πρωταρχικά τον τρόπο ανάλυσης, σχεδιασμού και ανάπτυξης λογισμικού και έπειτα να επικεντρωθεί στη γλώσσα Java, η οποία και χρησιμοποιείται στην υλοποίηση. Ο αναγνώστης μπορεί να δει τεχνικές προγραμματισμού γραφικών τόσο με το πακέτο *swing* αλλά και με *awt* και πως εφαρμόζονται στην ανάπτυξη ενός παιχνιδιού. Επιπλέον παρουσιάζεται η χρήση νημάτων (threads) αλλά και στοιχεία του Java Media Framework, όπως η αναπαραγωγή βίντεο.

## Abstract

This project implements the educational board game "AIDS Counter AIDS". It is a game that teaches ways of prevention from HIV/AIDS, encourages discussion and shoots down the myths and prejudices. The aim of this project is primarily to show software analysis, design and development, focusing on Java programming language. The reader can see graphics programming techniques with *swing* and *awt* packages and how they are being used at game developing. Additionally it is presented the thread usage and elements of Java Media Framework such as video playing.

## Περιεχόμενα

<b>1. ΕΙΣΑΓΩΓΗ</b>	<b>8</b>
1.1 Περιγραφή των κεφαλαίων	9
<b>2. ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΣΤΑ ΗΛΕΚΤΡΟΝΙΚΑ ΠΑΙΧΝΙΔΙΑ.</b>	<b>10</b>
<b>3. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ</b>	<b>13</b>
3.1 Συστατικά του παιχνιδιού.	13
3.2 Επεξήγηση καταστάσεων και εννοιών.	14
3.3 Περιγραφή του παιχνιδιού και των κανόνων του.	14
<b>4. ΠΡΟΔΙΑΓΡΑΦΕΣ ΑΠΑΙΤΗΣΕΩΝ</b>	<b>16</b>
4.1 Εισαγωγή	16
4.1.1 Σκοπός	16
4.1.2 Γενική άποψη	16
4.1.3 Ορισμοί - Γλωσσάριο	16
4.2 Γενική περιγραφή	19
4.2.1 Προοπτική του προϊόντος	19
4.2.2 Λειτουργίες του προϊόντος	19
4.2.3 Χαρακτηριστικά χρηστών	20
4.3 Ειδικές απαιτήσεις	20
4.3.1 Απαιτήσεις εξωτερικών διεπαφών	20
4.3.2 Προδιαγραφές περιπτώσεων χρήσης βασικού προϊόντος	23
4.3.3 Προδιαγραφές περιπτώσεων χρήσης δευτερεύοντος προϊόντος.	34
4.3.4 Προδιαγραφές επιδόσεων	36
4.3.5 Παράρτημα	37
<b>5. ΑΝΑΛΥΣΗ</b>	<b>38</b>
5.1 Ανάλυση του SafeGame	38
5.1.1 Σχεδιασμός διεπαφής	38
5.1.2 Αρχική διάκριση κλάσεων	39
5.1.3 Βασικές κλάσεις μηχανής παιχνιδιού και γραφικών	41
5.1.4 Ανάλυση περιπτώσεων χρήσης	43
5.1.5 Συμπληρωματική ανάλυση στοιχείων	55
5.2 Ανάλυση του συντάκτη SafeGameEditor	56
<b>6. ΣΧΕΔΙΑΣΗ</b>	<b>58</b>
6.1 Σχεδίαση του παιχνιδιού	58
6.1.1 Κλάσεις παιχτών, καρτών και δεδομένων	58
6.1.2 Κύρια κλάση του παιχνιδιού	68

6.1.3	Παραθυρικές κλάσεις	72
6.1.4	Μηχανή του παιχνιδιού και κλάσεις animation	93
6.1.5	Σχεδίαση περιπτώσεων χρήσης	134
<b>6.2</b>	<b>Σχεδίαση του συντάκτη SafeGameEditor</b>	<b>141</b>
<b>7.</b>	<b>ΥΛΟΠΟΙΗΣΗ</b>	<b>145</b>
<b>7.1</b>	<b>Υλοποίηση του SafeGame</b>	<b>145</b>
7.1.1	Κλάση NewGameDialog	145
7.1.2	Κλάση MainPanel και κλάσεις animation	146
7.1.3	Κλάση VideoDialog	148
7.1.4	Κλάση GlassQuestionShow	149
<b>7.2</b>	<b>Υλοποίηση του SafeGameEditor</b>	<b>150</b>
7.2.1	Πίνακας καρτών και κλάση CardTableData	150
<b>8.</b>	<b>ΠΡΟΤΑΣΕΙΣ - ΣΥΜΠΕΡΑΣΜΑΤΑ</b>	<b>152</b>
	<b>ΒΙΒΛΙΟΓΡΑΦΙΑ</b>	<b>153</b>
	<b>ΠΑΡΑΡΤΗΜΑ Α. - ΟΔΗΓΟΣ ΧΡΗΣΗΣ ΛΟΓΙΣΜΙΚΟΥ</b>	<b>154</b>
A.1	Οδηγός χρήσης του SafeGame	154
A.2	Οδηγός χρήσης του SafeGameEditor	158

## Ευρετήριο Εικόνων

Εικόνα 4-1 Βασική δομή προϊόντος	19
Εικόνα 4-2 Κύρια ροή γεγονότων στην κατά το τράβηγμα μη ασφαλούς κάρτας	31
Εικόνα 4-3 Διάγραμμα περιπτώσεων χρήσης	37
Εικόνα 5-1 - Πακέτα της εφαρμογής	38
Εικόνα 5-2 - Σχέδιο διεπαφής	39
Εικόνα 5-3 - Κλάση GamePlayer	40
Εικόνα 5-4 - Κλάσεις Card, QCard και StatusCard	41
Εικόνα 5-5 - Κύριες κλάσεις γραφικών	42
Εικόνα 5-6 - Διάγραμμα βασικών κλάσεων	43
Εικόνα 5-7 - Διάγραμμα συνεργασίας νέου παιχνιδιού	44
Εικόνα 5-8 - Διάγραμμα ακολουθίας νέου παιχνιδιού	44
Εικόνα 5-9 - Κλάση DataPack	45
Εικόνα 5-10 - Διάγραμμα ακολουθίας αρχικοποίησης παιχνιδιού	45
Εικόνα 5-11 - Διάγραμμα δραστηριοτήτων αποθήκευσης παιχνιδιού	46
Εικόνα 5-12 - Διάγραμμα ακολουθίας αποθήκευσης παιχνιδιού	47
Εικόνα 5-13 - Διάγραμμα δραστηριοτήτων φόρτωσης παιχνιδιού	48
Εικόνα 5-14 - Διάγραμμα ακολουθίας φόρτωσης παιχνιδιού	48
Εικόνα 5-15 - Διάγραμμα συνεργασίας τέλους παιχνιδιού	49
Εικόνα 5-16 - Κλάση GameEngine	50
Εικόνα 5-17 - Διάγραμμα συνεργασίας επιλογής παίχτη	50
Εικόνα 5-18 - Διάγραμμα δραστηριοτήτων τραβήγματος κάρτας	51
Εικόνα 5-19 - Διάγραμμα συνεργασίας τράβηγμα κάρτας	52
Εικόνα 5-20 - Διάγραμμα ακολουθίας για το τράβηγμα μη ασφαλούς κάρτας	52
Εικόνα 5-21 - Διάγραμμα δραστηριοτήτων κάρτας ερώτησης	53
Εικόνα 5-22 - Διάγραμμα ακολουθίας κάρτας ερώτησης	54
Εικόνα 5-23 - Διάγραμμα συνεργασίας ερώτησης βιβλίου	54
Εικόνα 5-24 - Διάγραμμα δραστηριοτήτων βρόχου GameEngine	55
Εικόνα 5-25 - Διάγραμμα κλάσεων συνολικό	56
Εικόνα 5-26 - Διάγραμμα κλάσεων SafeGameEditor	56
Εικόνα 6-1 - Κλάση GamePlayer	59
Εικόνα 6-2 - Κλάσεις Card, StatusCard και QCard	61
Εικόνα 6-3 - Κλάση DataPack	66
Εικόνα 6-4 - Κλάση Main	69
Εικόνα 6-5 - Παραθυρικές κλάσεις	73
Εικόνα 6-6 - Μενού παιχνιδιού	74
Εικόνα 6-7 - Κλάση SafeFrame	75
Εικόνα 6-8 - Κλάση NewGameDialog	82
Εικόνα 6-9 - Συστατικά της NewGameDialog	85
Εικόνα 6-10 - Κλάση HelpDialog	86

Εικόνα 6-11 - Κλάση AboutDialog	86
Εικόνα 6-12 - Κλάση GlassQuestionShow	87
Εικόνα 6-13 - Κλάση DangerCardsPanel	89
Εικόνα 6-14 - Συστατικά του PlayerPanel	89
Εικόνα 6-15 - Κλάση PlayerPanel	90
Εικόνα 6-16 - Διάγραμμα κλάσεων για το animation	94
Εικόνα 6-17 - Διεπαφή GeneralSprite και κλάση Sprite	95
Εικόνα 6-18 - Κλάση TextAnimator	98
Εικόνα 6-19 - Κλάση CursorAnimator	101
Εικόνα 6-20 - Κλάση ReadingAnimator	103
Εικόνα 6-21 - Κλάση CardAnimator	106
Εικόνα 6-22- Κλάση StatusCardAnimator	110
Εικόνα 6-23 - QCardAnimator	112
Εικόνα 6-24 - Κλάση TotalQCardAnimator	114
Εικόνα 6-25 - Φάσεις του κέρματος	117
Εικόνα 6-26 - Κλάση CoinAnimator	118
Εικόνα 6-27 - Κλάση VideoDialog	120
Εικόνα 6-28 - Κλάσεις ClipManager και ClipPlayer	122
Εικόνα 6-29 - Βρόχος επανάληψης της MainPanel	125
Εικόνα 6-30 - Κλάση MainPanel	126
Εικόνα 6-31 - Διάγραμμα κατάστασης βρόχου επανάληψης GameEngine	131
Εικόνα 6-32 - Διάγραμμα ακολουθίας νέου παιχνιδιού	134
Εικόνα 6-33 - Διάγραμμα καταστάσεων νέου παιχνιδιού	135
Εικόνα 6-34 - Διάγραμμα ακολουθίας αποθήκευσης παιχνιδιού	136
Εικόνα 6-35 - Διάγραμμα ακολουθίας φόρτωσης παιχνιδιού	137
Εικόνα 6-36 - Διάγραμμα καταστάσεων φόρτωσης παιχνιδιού	137
Εικόνα 6-37- Διάγραμμα δραστηριοτήτων τέλους παιχνιδιού	138
Εικόνα 6-38 - Διάγραμμα ακολουθίας περίπτωσης χρήσης μη ασφαλούς κάρτας	140
Εικόνα 6-39- Διάγραμμα ακολουθίας κάρτας ερώτησης	140
Εικόνα 6-40 - Διάγραμμα ακολουθίας ερώτησης βιβλίου	141
Εικόνα 6-41 - Διάγραμμα κλάσεων SafeGameEditor	142
Εικόνα 6-42 - Παράθυρο SafeGameEditor	143
Εικόνα 7-1 - Παράθυρο νέου παιχνιδιού	146
Εικόνα 7-2 - Διάγραμμα κλάσεων SafeGameEditor	150
Εικόνα A-1 - Καρτέλα Νεό παιχνίδι	155
Εικόνα A-2 - Ταμπλό του παιχνιδιού	155
Εικόνα A-3 - Όρια παιχνιδιού	157
Εικόνα A-4 - Παράθυρο του SafeGameEditor	159

## Ευρετήριο πινάκων

Πίνακας 4-1 Στοιχεία εισαγωγής στη φόρμα	21
Πίνακας 6-1 - Ιδιότητες GamePlayer	59
Πίνακας 6-2 - Μέθοδοι GamePlayer	60
Πίνακας 6-3 - Ιδιότητες κλάσης Card	62
Πίνακας 6-4 – Μέθοδοι κλάσης Card	62
Πίνακας 6-5 - Ιδιότητες κλάσης StatusCard	63
Πίνακας 6-6 - Μέθοδοι κλάσης StatusCard	63
Πίνακας 6-7 - Ιδιότητες κλάσης QCard	64
Πίνακας 6-8 - Μέθοδοι κλάσης QCard	64
Πίνακας 6-9 - Ιδιότητες της κλάσης DataPack	66
Πίνακας 6-10 - Μέθοδοι της κλάσης DataPack	67
Πίνακας 6-11 - Ιδιότητες της Main	69
Πίνακας 6-12 - Μέθοδοι της Main	70
Πίνακας 6-13 - Ιδιότητες της SafeFrame	76
Πίνακας 6-14 - Μέθοδοι της SafeFrame	77
Πίνακας 6-15 - Ιδιότητες κλάσης NewGameDialog	83
Πίνακας 6-16 - Ιδιότητες της κλάσης NewGameDialog	84
Πίνακας 6-17 - Ιδιότητες της GlassQuestionShow	87
Πίνακας 6-18 - Μέθοδοι της GlassQuestionShow	88
Πίνακας 6-19 - Ιδιότητες της PlayerPanel	90
Πίνακας 6-20 - Μέθοδοι της PlayerPanel	92
Πίνακας 6-21 - Μέθοδοι της GeneralSprite	95
Πίνακας 6-22 - Ιδιότητες της Sprite	96
Πίνακας 6-23 - Ιδιότητες της Sprite	97
Πίνακας 6-24 - Ιδιότητες της TextAnimator	98
Πίνακας 6-25 - Μέθοδοι της TextAnimator	99
Πίνακας 6-26 - Ιδιότητες της CursorAnimator	101
Πίνακας 6-27- Μέθοδοι της CursorAnimator	102
Πίνακας 6-28 - Ιδιότητες της ReadingAnimator	103
Πίνακας 6-29 – Μέθοδοι της ReadingAnimator	104
Πίνακας 6-30 - Ιδιότητες της CardAnimator	106
Πίνακας 6-31 - Μέθοδοι της CardAnimator	108
Πίνακας 6-32 - Ιδιότητες της StatusCardAnimator	110
Πίνακας 6-33 – Μέθοδοι StatusCardAnimator	110
Πίνακας 6-34 - Ιδιότητες της QCardAnimator	112
Πίνακας 6-35 - Μέθοδοι της QCardAnimator	113
Πίνακας 6-36 - Ιδιότητες της TotalQCardAnimator	115
Πίνακας 6-37 - Μέθοδοι της TotalQCardAnimator	115
Πίνακας 6-38 - Ιδιότητες της CoinAnimator	118

<b>Πίνακας 6-39 - Μέθοδοι της CoinAnimator</b>	<b>119</b>
<b>Πίνακας 6-40 - Ιδιότητες της VideoDialog</b>	<b>120</b>
<b>Πίνακας 6-41 - Μέθοδοι της VideoDialog</b>	<b>121</b>
<b>Πίνακας 6-42 - Ιδιότητες της ClipPlayer</b>	<b>123</b>
<b>Πίνακας 6-43 - Μέθοδοι της ClipPlayer</b>	<b>123</b>
<b>Πίνακας 6-44 - Ιδιότητες της ClipManager</b>	<b>123</b>
<b>Πίνακας 6-45 - Μέθοδοι της ClipManager</b>	<b>124</b>
<b>Πίνακας 6-46 - Ιδιότητες της MainPanel</b>	<b>126</b>
<b>Πίνακας 6-47 - Μέθοδοι της MainPanel</b>	<b>127</b>
<b>Πίνακας 6-48 - Ιδιότητες της GameEngine</b>	<b>131</b>
<b>Πίνακας 6-49 - Μέθοδοι της GameEngine</b>	<b>132</b>



## 1. Εισαγωγή

Αρχικός στόχος της εργασίας είναι η δημιουργία ενός ηλεκτρονικού εκπαιδευτικού παιχνιδιού βασισμένο στο αντίστοιχο επιτραπέζιο. Μέσα από την δημιουργία του παιχνιδιού παρουσιάζεται κάθε στάδιο της διαδικασίας ανάπτυξης λογισμικού. Αρχικά προσδιορίζονται οι απαιτήσεις του λογισμικού που πρόκειται να παραχθεί, μέσα από προσεκτική μελέτη των κανόνων του παιχνιδιού εντοπίζονται οι περιπτώσεις χρήσης, οι λειτουργικές και μη λειτουργικές απαιτήσεις. Έπειτα συγκεντρώνονται και παρουσιάζονται σε ένα κεφάλαιο που τηρεί τη δομή του εγγράφου όπως ορίζει το πρότυπο IEEE Std 830.

Ακολουθεί η ανάλυση και η σχεδίαση του λογισμικού βασισμένες στις απαιτήσεις που ορίστηκαν. Πρόκειται για δύο στάδια που περιγράφονται με τη γλώσσα προτυποποίησης UML και απευθύνονται στους τεχνικούς του λογισμικού. Η ανάλυση συμβαδίζει πολλές φορές με την διαδικασία του εντοπισμού των απαιτήσεων με στόχο να αποκαλύψει απαιτήσεις που τυχόν έχουν παραλειφθεί. Συγκεντρώνεται κυρίως στο τι πρέπει να κάνει το λογισμικό, ενώ το πως παρουσιάζεται στο στάδιο της σχεδίασης. Τα όρια μεταξύ ανάλυσης και σχεδίασης είναι μερικές ασαφές. Η σχεδίαση παρουσιάζει με κάθε λεπτομέρεια το πως υλοποιείται το λογισμικό ενώ το παραγόμενο αποτέλεσμα είναι έτοιμο προς άμεση υλοποίηση.

Το στάδιο της υλοποίησης μας δίνει πλέον το πρώτο από αποτέλεσμα, στην παρούσα εργασία έχουν επιλεγεί να παρουσιαστούν συγκεκριμένα κομμάτια κώδικα που παρουσιάζουν ενδιαφέρον. Η υλοποίηση είναι άμεσα συνδεδεμένη με την γλώσσα προγραμματισμού που χρησιμοποιείται. Ο κύκλος ανάπτυξης λογισμικού κλείνει με την φάση του ελέγχου όπου το λογισμικό ελέγχεται για το κατά πόσο πληρεί τις απαιτήσεις ενώ παράλληλα εντοπίζονται λογικά λάθη στην κωδικοποίηση. Η φάση του ελέγχου δεν παρουσιάζεται στην εργασία.

Σκοπός του παιχνιδιού είναι η ενημέρωση των παιχτών σχετικά με τον ιό AIDS, τόσο για τον ίδιο τον ιό και την δράση του όσο και για τις προκαταλήψεις που υπάρχουν και τις κοινωνικές επιπτώσεις στους φορείς του. Το παιχνίδι είναι αρκετά απλό, αποτελείται κυρίως από κάρτες που μας περιγράφουν καταστάσεις της καθημερινής ζωής και από κάρτες με ερωτήσεις που καλούνται να απαντήσουν οι παίχτες, ενώ δεν δίνει ιδιαίτερη έμφαση στο νικητή. Η όψη του παιχνιδιού αποφασίστηκε να είναι όμοια με το επιτραπέζιο παιχνίδι, έτσι δίνεται η

αίσθηση ύπαρξης ενός ταμπλό παιχνιδιού στο οποίο οι παίχτες κάθονται κυκλικά. Επιπλέον για την αναπαράσταση του κειμένου που παρουσιάζουν οι κάρτες, επιλέχθηκε η προβολή μικρών βίντεο. Τέλος με στόχο την ευελιξία και την ανανέωση και επειδή πρόκειται για ένα παιχνίδι γνώσεων, αποφασίστηκε να δοθεί η δυνατότητα επεξεργασίας των δεδομένων αλλά και ορισμένων κανόνων του παιχνιδιού. Έτσι δημιουργήθηκε ένα δεύτερο λογισμικό, ως συντάκτης των δεδομένων και των ρυθμίσεων του παιχνιδιού, που επιτρέπει στον χρήστη την ενημέρωση των κειμένων των καρτών, χωρίς να είναι απαραίτητη η παρέμβαση στον κώδικα.

Για την γλώσσα Java χρησιμοποιήθηκε το JDK 1.6.0 και το Java Media Framework 2.1.1, ο κώδικας αναπτύχθηκε με τη βοήθεια του Netbeans IDE 6.0.1.

## **1.1 Περιγραφή των κεφαλαίων**

Στο 2ο κεφάλαιο γίνεται μία ιστορική αναδρομή στα ηλεκτρονικά παιχνίδια, στο 3ο κεφάλαιο υπάρχει μία περιγραφή του παιχνιδιού και των κανόνων του, ενώ στο 4ο κεφάλαιο παρατίθενται οι απαιτήσεις του λογισμικού όπως ορίζει το πρότυπο IEEE Std 830. Ακολουθεί η ανάλυση στο 5ο κεφάλαιο και η σχεδίαση στο 6ο κεφάλαιο χρησιμοποιώντας την γλώσσα UML, λεπτομέρειες της υλοποίησης αναλύονται στο 7ο κεφάλαιο. Το 8ο κεφάλαιο μας παρουσιάζει πιθανές βελτιώσεις και συμπεράσματα. Η εργασία τελειώνει με τον οδηγό λογισμικού στο παράρτημα.

Στο συνοδευτικό cd υπάρχουν τα εκτελέσιμα του παιχνιδιού όπως και οι πλήρεις κατάλογοι ανάπτυξης του Netbeans.

## 2. Ιστορική αναδρομή στα ηλεκτρονικά παιχνίδια.

Με την ανάπτυξη των υπολογιστών άρχισαν να εμφανίζονται και τα πρώτα παιχνίδια. Το 1948 ο Thomas T. Goldsmith Jr. και ο Estle Ray Mann δημιούργησαν μία συσκευή που χρησιμοποιούσε οχτώ ηλεκτρονικές λυχνίες για να προσομοιώσει την εκτόξευση πυραύλων προς στόχους, ενώ μπορούσε να ελέγξει την ταχύτητα και την κατεύθυνσή τους. Το 1952 δημιουργήθηκε από τον A.S. Douglas στα πλαίσια της εργασίας του για το πανεπιστήμιο του Cambridge το OXO , επρόκειτο για μία έκδοση του παιχνιδιού τρίλιζα (Tic-Tac-Toe) στο οποίο ο παίχτης χρησιμοποιούσε ένα αριθμητικό περιστροφικό πληκτρολόγιο, για να παίξει εναντίον του υπολογιστή. Το 1958 ο William Higginbotham δημιούργησε σε ένα παλμοσκόπιο (oscilloscope), ένα παιχνίδι με το όνομα "Tennis for two" με το οποίο έπαιζε κάποιος έναν αγώνα τένις, ο έλεγχος γινόταν με δύο χειριστήρια.

Το πρώτο πραγματικό παιχνίδι σε υπολογιστή με ενσωματωμένη οθόνη δημιουργήθηκε το 1962 από τον Steve Russel στο MIT και είχε την ονομασία "SpaceWar!". Ήταν ένα παιχνίδι που περιείχε δύο διαστημόπλοια και σκοπός του παίχτη ήταν να πυροβολήσει το αντίπαλο σκάφος μέχρι να καταστραφεί.

Η δεκαετία του '70 υπήρξε η χρυσή εποχή των arcades παιχνιδιών, δηλαδή μηχανές που λειτουργούν με κέρματα και βρίσκονται σε αίθουσες παιχνιδιών, εστιατόρια, εμπορικά κέντρα κ.λ.π. . Τότε εμφανίζεται το διάσημο 'Pong' της Atari το 1972, που εμφάνιζε ένα τραπέζι τένις. Το 'Space Invaders' της ιαπωνικής Taito ήταν το πρώτο παιχνίδι που με την έννοια της βαθμολογίας (score) και της μέγιστης βαθμολογίας (high score), παρακινώντας τους παίχτες ώστε να προσπαθούν να ξεπεράσουν την μέγιστη βαθμολογία. Το 1976 κυκλοφορεί μία κονσόλα παιχνιδιών με το όνομα Fairchild 'Video Entertainment System (VES)'. Η κονσόλα αυτή χρησιμοποιούσε μνήμες ROM (ROM cartridges) στις οποίες ήταν αποθηκευμένα τα παιχνίδια, μέχρι τώρα κάθε μηχανή μπορούσε να παίξει ένα μόνο παιχνίδι. Έπειτα ακολούθησαν και άλλες κονσόλες όπως η 'Video Computer System (VCS)' της Atari το 1977 με τεράστια εμπορική επιτυχία και η Intellivision της Mattel το 1980.

Το Μάιο του 1980 κυκλοφορεί στην Ιαπωνία ένα παιχνίδι με το όνομα Pac-Man από την Namco, ίσως από τα διασημότερα όλων των εποχών. Η αρχική ονομασία του παιχνιδιού ήταν Puck-Man, όμως λόγω της ενδεχόμενης αλλαγής του p με το f και του υπονοούμενου που δημιουργείται αποφασίστηκε το όνομα Pac-Man. Το

παιχνίδι το αποτελούσαν τελείες ανάμεσα από εμπόδια-τοίχους, 4 φαντασματάκια και ο ήρωας που είχε σχήμα πίτσας χωρίς ένα κομμάτι, σκοπός του παιχνιδιού ήταν ο ήρωας να φάει όλες τις τελείες χωρίς να τον πιάσουν τα φαντασματάκια. Ο δημιουργός του θέλησε φτιάξει ένα παιχνίδι χωρίς βία που να ταιριάζει και στα δύο φύλα, το κοινό αμέσως αγάπησε το Pac-Man.

Τη χρονιά του 1982 κυκλοφορεί το 'Commodore 64', ένας οικιακός προσωπικός υπολογιστής που γνώρισε πολύ μεγάλη επιτυχία λόγω των ανεπτυγμένων δυνατοτήτων του σε γραφικά και ήχο. Ο αντίστοιχος στην Ευρώπη και κυρίως στη Βρετανία ήταν ο 'ZX Spectrum'. Σχεδόν όλοι πλέον οι τίτλοι παιχνιδιών υπήρχαν και σ' αυτούς τους υπολογιστές. Το 1984 κυκλοφορεί από την IBM το PC/AT, ένας υπολογιστής ανταγωνιστικός στα γραφικά με τον 'Commodore 64' αλλά με φτωχό ήχο, ένα πρόβλημα που θα λυθεί στα τέλη της δεκαετίας του '80 με τις κάρτες ήχου.

Η τρίτη διάσταση αρχίζει να μπαίνει σταδιακά στα γραφικά των παιχνιδιών ξεκινώντας από το 'King's Quest', ένα παιχνίδι με ψευδο-τρισεπίσταντα γραφικά. Παράλληλα αναπτύσσεται μία νέα κατηγορία παιχνιδιών με βάση τη δικτύωση, τα παιχνίδια αυτά καλούνται MUD (Multi User Dungeons), πρόκειται για παιχνίδια ρόλων σε φανταστικούς κόσμους. Το περιβάλλον του παιχνιδιού βασιζόταν στο κείμενο, ο χρήστης διάβαζε τη περιγραφή των αντικειμένων, των συμβάντων και απαντούσε με κείμενο. Πρόκειται για τους προγόνους των σημερινών MMORPG.

Στη δεκαετία του '90, καθώς η υπολογιστική δύναμη αυξάνεται, αναπτύσσονται περαιτέρω τα γραφικά τριών διαστάσεων και εφαρμόζεται η τεχνολογία των οπτικών δίσκων. Οι κάρτες γραφικών μπορούν να επιταχύνουν τα τρισεπίσταντα γραφικά κάνοντας μόνες τους βασικούς υπολογισμούς για την εμφάνιση αντικειμένων. Το 'Quake', ένα από τα παιχνίδια πρώτου προσώπου είναι το πρώτο που κάνει εκτεταμένη χρήση αυτής της τεχνολογίας, το ίδιο παιχνίδι έχει τη δυνατότητα να παιχτεί σε δίκτυο. Τα παιχνίδια πλέον χρησιμοποιούν εκτεταμένα την έννοια της 'μηχανής παιχνιδιού' (game engine). Οι συνήθεις λειτουργίες μιας μηχανής παιχνιδιού είναι η προεπεξεργασία και η εμφάνιση των γραφικών του παιχνιδιού είτε είναι δισδιάστατα είτε τρισεπίσταντα, η εφαρμογή των φυσικών νόμων (αναπήδηση μιας μπάλας που χτυπάει σε έναν τοίχο), ο ήχος, η τεχνητή νοημοσύνη, η δικτύωση κ.α. . Συνήθως αναπτύσσονται σε συνδυασμό με προγράμματα που θα επιτρέψουν την γρήγορη ανάπτυξη κομματιών του παιχνιδιού που έχουν πολλά κοινά, όπως για παράδειγμα τις πίστες ενός

παιχνιδιού που μπορεί να έχουν διαφορετική διάταξη αλλά περιέχουν τα ίδια αντικείμενα που συμπεριφέρονται με τον ίδιο τρόπο.

Τα δικτυακά παιχνίδια πλέον γίνονται κανόνας, ενώ τεχνολογίες όπως η Java και το Macromedia Flash επιτρέπουν να παίζονται παιχνίδια μόνο μέσα από ένα πλοηγό του Internet, χωρίς καμία επιπλέον εγκατάσταση.

Στα επόμενα κεφάλαια θα δούμε την ανάπτυξη του παιχνιδιού «Κατά του AIDS», ξεκινώντας από την περιγραφή του.

### 3. Περιγραφή του προβλήματος

Το πρόβλημα που καλούμαστε να λύσουμε είναι η δημιουργία ενός εκπαιδευτικού παιχνιδιού σχετικό με τον ιό AIDS/HIV, το όνομα που δόθηκε στο παιχνίδι είναι SafeGame. Στόχος του παιχνιδιού είναι η ενημέρωση των παιχτών για τους κινδύνους του AIDS αλλά και τα μέτρα πρόληψης, βοηθάει στην εξάλειψη των προκαταλήψεων έναντι των φορέων του AIDS και ενθαρρύνει τη συζήτηση σχετικά με το ασφαλές σεξ. Οι παίχτες παρακολουθούν διάφορες καταστάσεις της πραγματικής ζωής και ενίοτε καλούνται να πάρουν αποφάσεις. Η νίκη δεν είναι αυτοσκοπός του παιχνιδιού καθώς η επιλογή του νικητή βασίζεται σε μεγάλο ποσοστό στον παράγοντα τύχη.

#### 3.1 Συστατικά του παιχνιδιού.

Το παιχνίδι αποτελείται από ένα ταμπλό στο οποίο τοποθετούνται κυκλικά οι παίχτες. Υπάρχουν δύο κατηγορίες καρτών οι μπλε και οι πράσινες.

Οι μπλε κάρτες αποτελούνται από :

- Κάρτες καταστάσεων οι οποίες διαχωρίζονται σε
  - Μη ασφαλείς κάρτες.
  - Μη επικίνδυνες κάρτες.
  - Ασφαλείς κάρτες.
- Κάρτες ερωτήσεων, οι οποίες περιέχουν ερωτήσεις του τύπου σωστό/λάθος και ο παίχτης καλείται να απαντήσει.
- Κάρτες με joker, οι οποίες μπορούν να χρησιμοποιηθούν για προστασία.

Οι πράσινες κάρτες είναι κάρτες με joker, η διαφορά τους από τους μπλε joker είναι στο τρόπο απόκτησης, ενώ η χρήση τους είναι ίδια.

Το πλήθος και το περιεχόμενο των καρτών είναι παράμετροι και δίνεται η δυνατότητα στην εκάστοτε διανομή του παιχνιδιού να αλλάξουν.

Υπάρχουν επίσης τα HIV token που δείχνουν το βαθμό μόλυνσης ενός παίχτη, τέλος υπάρχει ένα νόμισμα που αποφασίζει, με την τυχαία περιστροφή του, για το αν θα μολυνθεί ένας παίχτης.

### 3.2 Επεξήγηση καταστάσεων και εννοιών.

Το παιχνίδι παρουσιάζει στους παίκτες καταστάσεις και γεγονότα της αληθινής ζωής. Αυτό γίνεται μέσω των καρτών κατάστασης, θα μπορούσαμε να παρομοιάσουμε το τράβηγμα μιας κάρτας με ένα πραγματικό περιστατικό που θα μπορούσε να συμβεί στον καθένα μας. Κατηγοριοποιούμε αυτά τα περιστατικά σε τρεις μεγάλες ομάδες που αντιστοιχούν στις κάρτες κατάστασης, οι οποίες ερμηνεύονται ως εξής :

- Ασφαλείς κάρτες. Περιγράφουν μία κατάσταση στην οποία ο παίχτης έχει πάρει τις απαραίτητες προφυλάξεις, έτσι ώστε ακόμη κι αν εκτεθεί στον ιό, να μην κολλήσει.
- Μη επικίνδυνες κάρτες. Στη κατάσταση που περιγράφεται εδώ παίχτης εκτίθεται στον ιό, όμως δεν κινδυνεύει να τον κολλήσει.
- Μη ασφαλείς κάρτες. Ο παίχτης εκτίθεται στον ιό και λόγω της έλλειψης προφυλάξεων κολλάει τον ιό.

Οι παίκτες όταν εκτεθούν σε επικίνδυνες καταστάσεις μαζεύουν μη ασφαλή κάρτες αλλά και τα HIV Token, των οποίων το πλήθος δείχνει το βαθμό μόλυνσης ενός παίχτη.

### 3.3 Περιγραφή του παιχνιδιού και των κανόνων του.

Αρχικά οι μπλε κάρτες αφού ανακατευτούν τοποθετούνται σε μία στοίβα, ενώ μία δεύτερη στοίβα με τους πράσινους Joker τοποθετείται δίπλα. Έπειτα τα βήματα του παιχνιδιού είναι τα εξής :

- 1) Ο παίχτης που θα παίξει επιλέγεται με τυχαίο τρόπο.
- 2) Τραβάει μία μπλε κάρτα.
  - Αν είναι μία μη επικίνδυνη κάρτα ή ασφαλή κάρτα, ο παίχτης τη διαβάζει δυνατά και η κάρτα μπαίνει σε μία τρίτη στοίβα που καλείται 'στοίβα επιστροφής'.
  - Αν είναι μία μη ασφαλή κάρτα τότε ο παίχτης ρίχνει το HIV νόμισμα.

Αν το νόμισμα είναι θετικό στο ιό ( 😞 ), ο παίχτης για να μην κολλήσει τον ιό χρησιμοποιεί έναν joker, ενώ για να μην κρατήσει την μη ασφαλή κάρτα χρησιμοποιεί έναν δεύτερο joker. Στη περίπτωση που δεν έχει κανένα joker, κρατάει την μη ασφαλή κάρτα, η οποία

πηγαίνει στην στοίβας επιστροφής και λαμβάνει ένα HIV Token. Στην περίπτωση που το νόμισμα είναι αρνητικό στον ιό ( 😊 ), αν ο παίχτης διαθέτει joker τότε τοποθετεί την μη ασφαλή κάρτα στη στοίβα επιστροφής, αλλιώς την κρατά.

- Αν είναι μία κάρτα ερώτησης, ο παίχτης διαβάζει την ερώτηση και καλείται να την απαντήσει. Η απάντηση ελέγχεται, αν ο παίχτης έχει απαντήσει σωστά, επιβραβεύεται με έναν πράσινο joker στην αντίθετη περίπτωση δεν υπάρχει καμία επιβράβευση. Έπειτα η κάρτα της ερώτησης πηγαίνει στην στοίβα επιστροφής.
- Αν ο παίχτης τραβήξει μία κάρτα joker την κρατάει για μελλοντική προστασία.

4) Εκτελείται το 1<sup>ο</sup> βήμα μέχρι το τέλος του παιχνιδιού.

Υπάρχουν δύο καταστάσεις που μπορεί να μπει ένας παίχτης. Η πρώτη είναι όταν μαζέψει ένα συγκεκριμένο πλήθος μη ασφαλών καρτών (έστω πέντε), τότε θεωρούμε πως είναι εκτεθειμένος στον ιό και πρέπει να συμμετάσχει σε δημόσιο πρόγραμμα εκπαίδευσης με αποτέλεσμα κάθε φορά που έρχεται η σειρά του να παίξει, να παραθέτει μία ερώτηση με την απάντηση της και έπειτα να παίζει. Η δεύτερη κατάσταση είναι όταν ο παίχτης μαζέψει ένα πλήθος από HIV Token (έστω τρία), τότε θεωρούμε πως η υγεία του παίχτη έχει εξασθενήσει και ο παίχτης πρέπει να ξεκουραστεί. Για να βοηθήσει τους φίλους του παραθέτει μία ερώτηση και την απάντηση της κάθε φορά που πρέπει να παίζει. Το πλήθος των καρτών αλλά και των Token είναι μία παράμετρος που μπορεί να αλλάξει σε κάθε διανομή του παιχνιδιού.

Όταν τελειώσει η στοίβα με τις μπλε κάρτες ανακατεύουμε την στοίβα επιστροφής και τοποθετούμε εκ νέου τη στοίβα των μπλε καρτών. Το παιχνίδι τελειώνει στις εξής περιπτώσεις :

- Όλοι οι παίχτες έχουν εξασθενήσει, με αποτέλεσμα το μόνο που μπορούν να κάνουν είναι να παραθέτουν ερωτήσεις χωρίς όμως να μπορούν να τραβήξουν κάρτα.
- Όταν τελειώσει και για δεύτερη φορά η στοίβα με τις μπλε κάρτες.

Νικητής θεωρείται ο παίχτης με τις λιγότερες μη ασφαλείς κάρτες.



## 4. Προδιαγραφές Απαιτήσεων

Ακολουθεί το έγγραφο των προδιαγραφών απαιτήσεων με βάση το πρότυπο IEEE Std 830-1998.

### 4.1 Εισαγωγή

#### 4.1.1 Σκοπός

Σκοπός της ενότητας είναι να παρουσιάσει μία λεπτομερή περιγραφή του παιχνιδιού και των απαιτήσεων του. Απευθύνεται σε αυτούς που ενδιαφέρονται να μάθουν τις λεπτομέρειες του λογισμικού που καλύφθηκαν προγραμματιστικά.

#### 4.1.2 Γενική άποψη

Το λογισμικό που θα παραχθεί θα ονομάζεται "Safegame" και πρόκειται για ένα εκπαιδευτικό παιχνίδι. Σκοπός του είναι να ενημερώσει τους παίκτες σχετικά με τον ιό HIV/AIDS, τους κινδύνους που υπάρχουν αλλά και τα μέτρα πρόληψης. Το παιχνίδι περιέχει κάρτες που παρουσιάζουν καταστάσεις της αληθινής ζωής σχετικές με τον ιό και κάρτες με ερωτήσεις, στις οποίες οι χρήστες καλούνται να απαντήσουν. Η νίκη δεν είναι αυτοσκοπός του παιχνιδιού καθώς η επιλογή του νικητή βασίζεται σε μεγάλο ποσοστό στον παράγοντα τύχη.

Το παιχνίδι βοηθάει στην εξάλειψη των προκαταλήψεων έναντι των φορέων του AIDS και ενθαρρύνει τη συζήτηση σχετικά με το ασφαλές σεξ.

#### 4.1.3 Ορισμοί - Γλωσσάριο

Όρος	Επεξήγηση
Διαχειριστής	Φυσικό πρόσωπο που ορίζει τις παραμέτρους του παιχνιδιού, τις κάρτες και τις ερωτήσεις-απαντήσεις του βιβλίου.
Παίκτης	Φυσικό ή τεχνητό πρόσωπο που συμμετέχει στο παιχνίδι, το πλήθος τους σε κάθε παιχνίδι μπορεί να είναι από 2 έως 8. Σε κάθε παίκτη αντιστοιχούν πληροφορίες όπως οι κάρτες joker που έχει μαζέψει, οι μη ασφαλείς κάρτες και τα HIV Token.

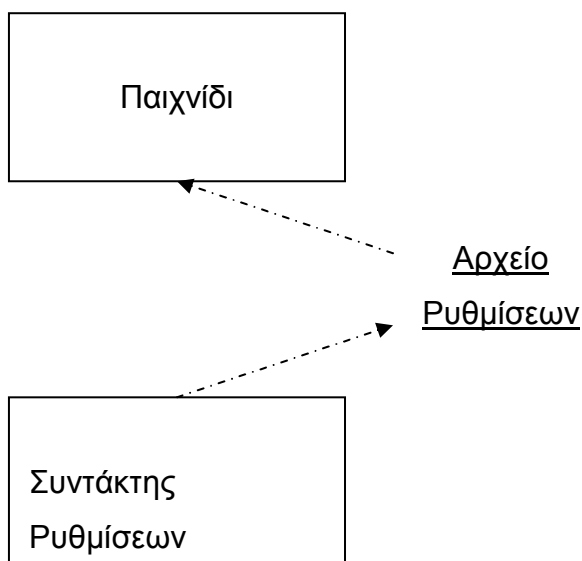
<b>Όρος</b>	<b>Επεξήγηση</b>
Κάρτες κατάστασης	Κάρτες χρώματος μπλε που περιγράφουν μία κατάσταση ή ένα γεγονός αληθινής ζωής. Διακρίνονται σε τρεις κατηγορίες ασφαλείς, μη ασφαλείς και μη επικίνδυνες. Συντάσσονται από τον διαχειριστή.
Ασφαλείς κάρτες	Περιγράφουν μία κατάσταση όπου ο παίχτης έχει πάρει τις απαραίτητες προφυλάξεις, έτσι ώστε ακόμη κι αν εκτεθεί στον ιό, να μην κολλήσει.
Μη ασφαλείς κάρτες	Περιγράφουν μία κατάσταση στην οποία ο παίχτης εκτίθεται στον ιό και λόγω της έλλειψης προφυλάξεων κολλάει τον ιό.
Μη επικίνδυνες κάρτες	Περιγράφουν μία κατάσταση στην οποία ο παίχτης εκτίθεται στον ιό, όμως δεν κινδυνεύει να τον κολλήσει.
Μπλε κάρτες joker	Κάρτες που χρησιμοποιούνται για προστασία και βρίσκονται στην στοίβα καρτών, το πλήθος τους ορίζεται από τον διαχειριστή.
Πράσινες κάρτες joker	Κάρτες που χρησιμοποιούνται για προστασία και βρίσκονται σε ξεχωριστή στοίβα, ο παίχτης της αποκτά ως επιβράβευση σε κάθε σωστή του απάντηση σε κάρτα ερώτησης. Έχουν συγκεκριμένο πλήθος που ρυθμίζεται από τον διαχειριστή.
Κάρτες ερωτήσεων	Κάρτες μπλε χρώματος, οι οποίες θέτουν στον παίχτη ένα ερώτημα με απάντηση σωστό ή λάθος ανήκουν και αυτές στην στοίβα καρτών.
Στοίβα καρτών	Αναφερόμαστε στην στοίβα με τις μπλε κάρτες, όπως αναφέραμε μπορεί να είναι κάρτες κατάστασης, κάρτες joker ή κάρτες ερωτήσεων.
Στοίβα επιστροφής	Όταν κάποιος παίχτης τραβήξει μία κάρτα από την αρχική στοίβα και δεν την κρατήσει, τότε αυτή μπαίνει σε μία δεύτερη στοίβα η οποία ονομάζεται στοίβα επιστροφής.

Όρος	Επεξήγηση
HIV νόμισμα	Ένα νόμισμα που έχει στην μία πλευρά του ένα χαρούμενο πρόσωπο ( 😊 ) και συμβολίζει ότι ο παίχτης δεν μολύνθηκε από τον ιό, ενώ από την άλλη πλευρά υπάρχει ένα θλιμμένο πρόσωπο ( ☹ ) και συμβολίζει την μόλυνση του παίχτη. Το νόμισμα το ρίχνει ο παίχτης όταν έχει βρεθεί σε μία μη ασφαλή κατάσταση, δηλαδή τράβηξε μία μη ασφαλή κάρτα κατάστασης.
HIV Token	Δείχνουν το βαθμό μόλυνσης ενός παίχτη από τον ιό.
Όριο HIV Token	Πρόκειται για ένα όριο στα HIV Token που όταν το φτάσει ένας παίχτης, τότε η υγεία του θεωρείται εξασθενημένη και πρέπει να ξεκουραστεί. Η τιμή του ορίζεται από τον διαχειριστή.
Όριο μη ασφαλών καρτών	Είναι το όριο των μη ασφαλών καρτών που όταν το μαζέψει ένας παίχτης, πρέπει να μπει σε δημόσιο πρόγραμμα εκπαίδευσης. Η τιμή του ορίζεται από τον διαχειριστή.
Εξασθενημένη υγεία	Η κατάσταση που μπαίνει ένας παίχτη όταν έχει φτάσει το όριο των HIV Token. Όταν επιδεινωθεί η υγεία του παίχτη, τότε ξεκουράζεται και κάθε φορά που έρχεται η σειρά για να παίξει, διαβάζει μία τυχαία ερώτηση-απάντηση του βιβλίου.
Δημόσιο πρόγραμμα εκπαίδευσης	Η κατάσταση που μπαίνει ένας παίχτης όταν μαζέψει το όριο των μη ασφαλών καρτών. Ο παίχτης που μπαίνει στο δημόσιο πρόγραμμα εκπαίδευσης κάθε φορά πριν παίξει διαβάζει μία τυχαία ερώτηση-απάντηση βιβλίου.
Ερωτήσεις – απαντήσεις βιβλίου	Πρόκειται για ένα σύνολο ερωτήσεων με τις απαντήσεις τους οι οποίες συμβάλλουν στην επιμόρφωση των παιχτών. Τις διαβάζουν δυνατά, οι παίχτες που έχουν μπει σε δημόσιο πρόγραμμα εκπαίδευσης ή έχει εξασθενήσει η υγεία τους. Τις συντάσσει ο διαχειριστής.

## 4.2 Γενική περιγραφή

### 4.2.1 Προοπτική του προϊόντος

Το προϊόν είναι αυτόνομο και δεν εντάσσεται σε κάποιο μεγαλύτερο σύστημα, αποτελείται δε από δύο ξεχωριστά υποπροϊόντα το βασικό και το δευτερεύον. Το βασικό που είναι το παιχνίδι και το δευτερεύον που είναι ένας συντάκτης των πληροφοριών (ερωτήσεις παιχνιδιού) και των κανόνων του παιχνιδιού, ο οποίος παράγει ένα αρχείο ρυθμίσεων που χρησιμοποιείται από το παιχνίδι. Η Εικόνα 4-1 μας παρουσιάζει τα βασικά συστατικά του προϊόντος.



Εικόνα 4-1 Βασική δομή προϊόντος

### 4.2.2 Λειτουργίες του προϊόντος

Θα μπορούσαμε να διακρίνουμε τις εξής λειτουργίες στο προϊόν:

- Ορισμός παραμέτρων, καρτών και ερωτήσεων-απαντήσεων του παιχνιδιού από τον χρήστη που ονομάζεται διαχειριστής, κάνοντας χρήση του συντάκτη, με δυνατότητα αποθήκευσης σε ξεχωριστό αρχείο.
- Ο χρήστης παίχτης μπορεί να παίξει το παιχνίδι.
- Ο χρήστης παίχτης μπορεί να αποθηκεύσει ένα παιχνίδι και στη συνέχεια να το φορτώσει.
- Ο χρήστης παίχτης έχει την δυνατότητα να ορίσει παύση του παιχνιδιού, όπως και να επιλέξει την ροή του παιχνιδιού χωρίς ήχο.

### **4.2.3 Χαρακτηριστικά χρηστών**

Όπως προαναφέρθηκε υπάρχουν δύο κατηγορίες χρηστών. Οι διαχειριστές, πρόκειται για χρήστες που ορίζουν τις παραμέτρους του παιχνιδιού και συντάσσουν τις ερωτήσεις και τις απαντήσεις του παιχνιδιού. Η δεύτερη κατηγορία χρηστών είναι οι παίκτες που ουσιαστικά είναι οι τελικοί χρήστες που θα παίξουν, το. Θεωρούμε πως μπορεί να είναι φυσικά αλλά και τεχνητά πρόσωπα (H/Y), το πλήθος του είναι από δύο έως οχτώ. Ορίζουμε ότι μπορεί να είναι πολλοί οι παίκτες, όμως μόνο ένας είναι ο ενεργός παίκτης κάθε φορά που χειρίζεται τις συσκευές εισόδου. Έτσι όταν ο παίκτης Α έχει επιλεχθεί να παίξει και έχει τραβήξει μία κάρτα ερώτησης, τότε αυτός θα κληθεί να απαντήσει. Το ίδιο θα συμβεί για τον παίκτη Β που ενδεχομένως να επιλεχθεί στον επόμενο γύρο.

## **4.3 Ειδικές απαιτήσεις**

### **4.3.1 Απαιτήσεις εξωτερικών διεπαφών**

#### **4.3.1.1 Διεπαφές χρηστών**

##### **4.3.1.1.1 Διεπαφές χρήστη παίκτη - παιχνιδιού**

Ο χρήστης παίκτης χρησιμοποιεί το βασικό προϊόν, το παιχνίδι. Με πλοήγηση σε μενού έχει την δυνατότητα να :

- i. Ξεκινήσει νέο παιχνίδι. Τότε εμφανίζεται μία φόρμα όπου συμπληρώνει το πλήθος των παιχτών, το όνομα του κάθε παίκτη και το αν ο εκάστοτε παίκτης είναι άνθρωπος ή όχι. Δίνεται η δυνατότητα στον χρήστη να ακυρώσει την διαδικασία με το ανάλογο κουμπί.
- ii. Να φορτώσει στην μνήμη ένα ήδη αποθηκευμένο παιχνίδι. Εμφανίζεται ένα παράθυρο διαλόγου για την επιλογή του αρχείου που περιέχει το αποθηκευμένο παιχνίδι. Δίνεται η δυνατότητα στον χρήστη να ακυρώσει την διαδικασία με το ανάλογο κουμπί.
- iii. Να αποθηκεύσει ένα παιχνίδι την στιγμή που παίζει. Εμφανίζεται ένα παράθυρο διαλόγου για την επιλογή του αρχείου στο οποίο θα αποθηκευτεί το παιχνίδι. Όταν επιλεχθεί ένα υπάρχον αρχείο ζητείται επιπλέον επιβεβαίωση εγγραφής. Δίνεται η δυνατότητα στον χρήστη να ακυρώσει την διαδικασία με το ανάλογο κουμπί.
- iv. Να προκαλέσει παύση του παιχνιδιού.

- v. Να σταματήσει την αναπαραγωγή ήχου στο παιχνίδι.
- vi. Να λάβει βοήθεια για το παιχνίδι.

Κατά την διάρκεια του παιχνιδιού κάθε στοιχείο – ιδιότητα των παιχτών θα πρέπει να είναι εμφανές και αυτά είναι :

- i. Το όνομα του παίχτη που δίνεται στην αρχικό παράθυρο δημιουργίας του παιχνιδιού.
- ii. Το τύπο του παίχτη αν δηλαδή είναι άνθρωπος ή Η/Υ.
- iii. Το πλήθος των μπλε joker που έχει μαζέψει ο παίχτης.
- iv. Το πλήθος των πράσινων joker που έχει μαζέψει ο παίχτης.
- v. Το πλήθος των μη ασφαλών καρτών που έχει μαζέψει ο παίχτης.
- vi. Το πλήθος των HIV Token που έχει μαζέψει ο παίχτης.
- vii. Αν βρίσκεται σε κάποια ιδιαίτερη κατάσταση και ποια είναι αυτή, δηλαδή αν παρακολουθεί κάποιο δημόσιο πρόγραμμα εκπαίδευσης ή αν η υγεία του έχει επιδεινωθεί.

Επίσης θεωρούμε πως υπάρχει ένα ταμπλό όπου οι παίχτες κάθονται κυκλικά γύρω από αυτό, εκεί εξελίσσεται το παιχνίδι. Έτσι ζητείται μία αναπαράσταση των παιχτών που κάθονται κυκλικά γύρω από ένα τραπέζι. Πάνω στο τραπέζι υπάρχουν οι στοίβες με τις κάρτες.

Κατά το τράβηγμα μιας κάρτας, ο χρήστης έχει όσο χρόνο χρειάζεται για να διαβάσει το κείμενο της όπως και μπορεί να δει κάποιο σχετικό βίντεο που ενδεχομένως να συνοδεύει την κάρτα, αυτό σημαίνει ότι υπάρχουν κουμπιά ελέγχου που δίνουν την δυνατότητα στο χρήστη να αποφασίσει το τέλος της παρουσίασης μιας κάρτας.

Τέλος ο χρήστης οποιαδήποτε στιγμή θελήσει μπορεί να λάβει βοήθεια σχετικά με τους κανόνες του παιχνιδιού μέσα από το ανάλογο μενού.

#### 4.3.1.1.2 Διεπαφές χρήστη διαχειριστή – συντάκτη ρυθμίσεων

Ο χρήστης διαχειριστής χρησιμοποιεί το δευτερεύον προϊόν, τον συντάκτη ρυθμίσεων. Πρόκειται για ανεξάρτητη εφαρμογή που παρέχει τις απαραίτητες φόρμες για την συμπλήρωση των πληροφοριών. Στον Πίνακα 4-1 αναφέρονται οι πληροφορίες που θα πρέπει να μπορεί να εισάγει ο χρήστης.

**Πίνακας 4-1 Στοιχεία εισαγωγής στη φόρμα**

Είδος πληροφορίας	Τύποι δεδομένων και όρια τιμών
-------------------	--------------------------------

<b>Είδος πληροφορίας</b>	<b>Τύποι δεδομένων και όρια τιμών</b>
Όριο HIV Token	Αριθμητικές τιμές με εύρος 1-5
Όριο μη ασφαλών καρτών	Αριθμητικές τιμές με εύρος 1-10
Αριθμός πράσινων joker	Θετικές αριθμητικές τιμές
Αριθμός μπλε joker	Θετικές αριθμητικές τιμές
Κάρτες κατάστασης	Περιέχει τα εξής δεδομένα : <ol style="list-style-type: none"> <li>1. Μοναδικός κωδικός αναγνώρισης της κάρτας που είναι θετικός αριθμός.</li> <li>2. Τύπος κάρτας που μπορεί να είναι ασφαλής, μη ασφαλής και μη επικίνδυνη.</li> <li>3. Κείμενο κάρτας που περιγράφει την κατάσταση.</li> <li>4. Αν η κάρτα συνοδεύεται από κάποιο βίντεο, η τιμή είναι αληθές ή ψευδές.</li> </ol>
Κάρτες ερωτήσεων	Περιέχει τα εξής δεδομένα : <ol style="list-style-type: none"> <li>1. Μοναδικός κωδικός αναγνώρισης της κάρτας που είναι θετικός αριθμός.</li> <li>2. Κείμενο κάρτας που περιέχει την ερώτηση.</li> <li>3. Απάντηση στην ερώτηση της κάρτας, οι τιμές είναι σωστό ή λάθος.</li> <li>4. Αν η κάρτα συνοδεύεται από κάποιο βίντεο, η τιμή είναι αληθές ή ψευδές.</li> </ol>
Ερωτήσεις – απαντήσεις βιβλίου	Κείμενο που περιέχει μία ερώτηση και την απάντησή της.

Επιπλέον απαιτείται η παροχή μενού για την αποθήκευση και την φόρτωση του αρχείου ρυθμίσεων με τα ανάλογα παράθυρα διαλόγου ώστε ο χρήστης να μπορεί να εντοπίσει με ευκολία το αρχείο.

Τέλος απαιτείται η παροχή βοήθειας με οδηγίες χρήσης του προγράμματος, όπως και συμβουλές κουμπιών για την κατανόηση της λειτουργίας κάθε κουμπιού της φόρμας.

#### 4.3.1.2 Διεπαφές υλικού

Συσκευές εισόδου : Πληκτρολόγιο και ποντίκι.

Συσκευές εξόδου : Οθόνη. Απαιτείται ανάλυση οθόνης μεγαλύτερη από 800x600 και τουλάχιστον 256 χρώματα.

#### 4.3.1.3 Διεπαφές λογισμικού

Απαιτείται χρήση της γλώσσας Java για την ανάπτυξη του κώδικα όπως και του ανάλογου JRE ( Java Runtime Environment) στο σύστημα λειτουργίας του προϊόντος.

#### 4.3.1.4 Διεπαφές επικοινωνιών

Το προϊόν δεν επικοινωνεί με άλλα συστήματα.

#### 4.3.2 Προδιαγραφές περιπτώσεων χρήσης βασικού προϊόντος

Στην Εικόνα 4-3 που βρίσκεται στο παράρτημα του κεφαλαίου διακρίνουμε το διάγραμμα περιπτώσεων χρήσης.

##### 4.3.2.1 Περίπτωση Χρήσης : Νέο παιχνίδι

Περίπτωση χρήσης :	ΠΧ01
Χειριστής :	Χρήστης <i>παίχτης</i>
Περιγραφή :	Δημιουργία νέου παιχνιδιού
Προϋποθέσεις :	Ο χειριστής να βρίσκεται στο κύριο παράθυρο του παιχνιδιού.
Κύρια ροή γεγονότων :	<ol style="list-style-type: none"> <li>1. Ο χειριστής εισάγει το πλήθος των παιχτών, τα ονόματά τους αλλά και την φύση τους (άνθρωπος-Η/Υ) και πατά το κουμπί OK.</li> <li>2. Οι παίχτες αρχικοποιούνται ανάλογα με τα στοιχεία που έχουν δοθεί.</li> </ol>
Μετέπειτα κατάσταση :	Το παιχνίδι ξεκινάει και έχουμε την τυχαία επιλογή παίχτη (ΠΧ05).
Εναλλακτικές ροές γεγονότων και εξαιρέσεις :	1α. Ο χειριστής πατά το κουμπί ΑΚΥΡΟ ή κλείνει το παράθυρο, τότε το σύστημα επανέρχεται στην προηγούμενη κατάσταση του.



#### 4.3.2.2 Περίπτωση Χρήσης : Αποθήκευση παιχνιδιού

Περίπτωση χρήσης :	ΠΧ02
Χειριστής :	Χρήστης <i>παίχτης</i>
Περιγραφή :	Ο χειριστής αποθηκεύει το τρέχων παιχνίδι.
Προϋποθέσεις :	Να έχει ξεκινήσει ένα παιχνίδι.
Κύρια ροή γεγονότων :	<ol style="list-style-type: none"> <li>1. Εμφανίζεται το παράθυρο διαλόγου ώστε να επιλέξει την διαδρομή στο δίσκο αλλά και το όνομα του αρχείου στο οποίο θέλει να αποθηκεύσει το παιχνίδι</li> <li>2. Ο χειριστής επιβεβαιώνει με ένα τελικό ΟΚ.</li> <li>3. Γίνεται εγγραφή των δεδομένων στο αρχείο που ζητήθηκε.</li> <li>4. Το παιχνίδι συνεχίζεται.</li> </ol>
Μετάπειτα κατάσταση :	Έχουν αποθηκευτεί στο δίσκο, στο αρχείο που επιλέχτηκε τα ακριβή στοιχεία του τρέχοντος παιχνιδιού, δηλαδή αριθμός και ονόματα παιχτών, οι κάρτες που έχουν τραβήξει οι παίχτες όπως και οι κάρτες που υπολείπονται στην υπάρχουσα στοίβα.
Εναλλακτικές ροές γεγονότων και εξαιρέσεις :	<p>2α. Ο χειριστής επιλέγει ένα υπάρχον αρχείο.</p> <ol style="list-style-type: none"> <li>1. Το σύστημα ζητά επιβεβαίωση για να γράψει επάνω του.</li> <li>2. Ο χειριστής επιβεβαιώνει θετικά και γίνεται η εγγραφή, σε αρνητική επιβεβαίωση δεν γίνεται καμία ενέργεια και το σύστημα επιστρέφει στο παιχνίδι.</li> </ol> <p>2β. Το σύστημα αρνείται να αποθηκεύσει στο αρχείο που ορίσαμε για διάφορες αιτίες (πχ. προστασία εγγραφής στο δίσκο).</p> <ol style="list-style-type: none"> <li>1. Ο χειριστής πληροφορείται το συμβάν.</li> <li>2. Το σύστημα επιστρέφει στο παιχνίδι.</li> </ol> <p>2γ. Ο χειριστής πατά το κουμπί ΑΚΥΡΟ ή κλείνει το παράθυρο, τότε παιχνίδι συνεχίζεται χωρίς να έχει γίνει καμία ενέργεια.</p>

#### 4.3.2.3 Περίπτωση Χρήσης : Φόρτωση παιχνιδιού

Περίπτωση χρήσης :	ΠΧ03
Χειριστής :	Χρήστης <i>παίχτης</i>
Περιγραφή :	Ο χειριστής ανακτά από αρχείο αποθηκευμένο παιχνίδι.
Προϋποθέσεις :	Ο χειριστής να βρίσκεται στο κύριο παράθυρο του παιχνιδιού.
Κύρια ροή γεγονότων :	<ol style="list-style-type: none"> <li>1. Εμφανίζεται το παράθυρο διαλόγου ώστε να επιλέξει την διαδρομή στο δίσκο αλλά και το όνομα του αρχείου στο οποίο βρίσκεται το αποθηκευμένο παιχνίδι.</li> <li>2. Ο χειριστής επιβεβαιώνει με ένα τελικό OK.</li> <li>3. Ανακτώνται τα δεδομένα από το επιλεγμένο αρχείο</li> <li>4. Το παιχνίδι συνεχίζεται με τα δεδομένα του αρχείου (παίχτες, πληροφορίες παιχτών, κάρτες παιχνιδιού).</li> </ol>
Μετάπειτα κατάσταση :	Το παιχνίδι που παίζεται περιέχει τα δεδομένα του αρχείου που επιλέχτηκε.
Εναλλακτικές ροές γεγονότων και εξαιρέσεις :	<ol style="list-style-type: none"> <li>2α. Ο χειριστής πατά το κουμπί ΑΚΥΡΟ ή κλείνει το παράθυρο, τότε το σύστημα επανέρχεται στην προηγούμενη κατάστασή του χωρίς να έχει γίνει καμία ενέργεια.</li> <li>2β. Πρόβλημα στην διαδικασία, το επιλεγμένο αρχείο είτε δεν περιέχει τα σωστά δεδομένα είτε είναι αδύνατη η ανάγνωσή του. <ol style="list-style-type: none"> <li>1. Ο χειριστής πληροφορείται για το συμβάν.</li> <li>2. Το σύστημα επανέρχεται στην προηγούμενη κατάσταση του χωρίς να έχει γίνει καμία ενέργεια.</li> </ol> </li> </ol>

#### 4.3.2.4 Περίπτωση Χρήσης : Τέλος παιχνιδιού

Περίπτωση χρήσης :	ΠΧ04
Χειριστής :	Χρήστης <i>παίχτης</i>
Περιγραφή :	Το παιχνίδι έχει φτάσει στο τέλος του.

Περίπτωση χρήσης :	ΠΧ04
Προϋποθέσεις :	<ol style="list-style-type: none"> <li>1. Να υπάρχει ένα παιχνίδι σε εξέλιξη.</li> <li>2. Η υγεία όλων των παιχτών έχει επιδεινωθεί (έφτασαν στο όριο των HIV Token) ή έχει τελειώσει για δεύτερη φορά η στοίβα με τις μπλε κάρτες.</li> </ol>
Κύρια ροή γεγονότων :	<ol style="list-style-type: none"> <li>1. Πληροφόρηση των παιχτών για την κατάταξη στην βαθμολογία. Η κατάταξη γίνεται ως εξής : <ol style="list-style-type: none"> <li>i. Οι παίκτες με τις λιγότερες μη ασφαλείς κάρτες και χωρίς κανένα HIV Token μπαίνουν πρώτα.</li> <li>ii. Ακολουθεί η κατάταξη των παιχτών ανάλογα με τα HIV Token που έχουν.</li> </ol> </li> </ol>
Μετάπειτα κατάσταση :	Το παιχνίδι έχει τελειώσει, οι επόμενες δυνατές καταστάσεις είναι η δημιουργία νέου παιχνιδιού (ΠΧ01) ή η φόρτωση κάποιου αποθηκευμένου (ΠΧ03).
Εναλλακτικές ροές γεγονότων και εξαιρέσεις :	Καμία

#### 4.3.2.5 Περίπτωση Χρήσης : Επιλογή παίχτη

Περίπτωση χρήσης :	ΠΧ05
Χειριστής :	Χρήστης <i>παίχτης</i>
Περιγραφή :	Διαδικασία τυχαίας επιλογής παίχτη.
Προϋποθέσεις :	<ol style="list-style-type: none"> <li>1. Να υπάρχει παιχνίδι σε εξέλιξη.</li> <li>2. Να ξεκινάει νέος γύρος του παιχνιδιού.</li> </ol>
Κύρια ροή γεγονότων :	<ol style="list-style-type: none"> <li>1. Οπτικοποιημένη εναλλαγή μεταξύ των παιχτών κατά προτίμηση κυκλικά έως την τυχαία κατάληξη σε κάποιον.</li> <li>2. Πληροφόρηση για τον παίχτη που επιλέχθηκε, όπως και για την επόμενη φάση ανάλογα με την κατάσταση του παίχτη.</li> </ol>
Μετάπειτα κατάσταση :	Ανάλογα με την κατάσταση του παίχτη οι επόμενες φάσεις μπορεί να είναι οι εξής :

Περίπτωση χρήσης :	ΠΧ05
	<ol style="list-style-type: none"> <li>1. Ο παίχτης έχει επιδεινωμένη υγεία (όριο HIV Token), οπότε διαβάζει μία ερώτηση-απάντηση βιβλίου (ΠΧ10).</li> <li>2. Ο παίχτης βρίσκεται σε δημόσιο πρόγραμμα εκπαίδευσης (όριο μη ασφαλών καρτών), οπότε διαβάζει μία ερώτηση-απάντηση βιβλίου (ΠΧ10).</li> <li>3. Ο παίχτης δεν βρίσκεται σε καμία από τις παραπάνω καταστάσεις επομένως μπορεί να τραβήξει μία μπλε κάρτα από την στοίβα (ΠΧ06).</li> </ol>
Εναλλακτικές ροές γεγονότων και εξαιρέσεις :	<p>1α. Ο χειριστής διακόπτει την εναλλαγή των παιχτών και επιλέγει <i>Νέο παιχνίδι</i> (ΠΧ01) ή <i>Φόρτωση παιχνιδιού</i> (ΠΧ03).</p> <ol style="list-style-type: none"> <li>1. Παύση του παιχνιδιού και ενημέρωση του χειριστή ότι ένα παιχνίδι είναι σε εξέλιξη και αν επιθυμεί την διακοπή του.</li> <li>2. Σε θετική απάντηση του χειριστή το σύστημα προχωράει αναλόγως στην ΠΧ01 ή ΠΧ03 , σε αρνητική απάντηση συνεχίζεται η εναλλαγή των παιχτών.</li> </ol> <p>1β. Ο χειριστής διακόπτει την εναλλαγή των παιχτών και επιλέγει <i>Αποθήκευση παιχνιδιού</i> (ΠΧ02), τότε το σύστημα προχωρά στην ΠΧ02.</p>

#### 4.3.2.6 Περίπτωση Χρήσης : Τράβηγμα κάρτας

Περίπτωση χρήσης :	ΠΧ06
Χειριστής :	Χρήστης <i>παίχτης</i>
Περιγραφή :	Ο παίχτης τραβάει μία κάρτα.
Προϋποθέσεις :	Μόλις έχει επιλεχθεί ένα παίχτης από την διαδικασία τυχαίας επιλογής παίχτη (ΠΧ05).
Κύρια ροή γεγονότων :	<ol style="list-style-type: none"> <li>1. Ο χειριστής αντιλαμβάνεται με οπτική αναπαράσταση ότι τραβάει μία μπλε κάρτα από την στοίβα με τις</li> </ol>

Περίπτωση χρήσης :	ΠΧ06
	<p>κάρτες.</p> <p>2. Το περιεχόμενο της κάρτας γίνεται πλέον ορατό στον χειριστή. Το περιεχόμενο είναι απαραίτητως κείμενο, ενώ σε μερικές κάρτες υπάρχει η δυνατότητα αναπαράστασης με την χρήση βίντεο.</p> <p>3. Στην περίπτωση ύπαρξης βίντεο εμφανίζεται ξεχωριστό παράθυρο με το βίντεο σε αναπαραγωγή, ενώ δίνεται η δυνατότητα στον χειριστή με κουμπί να επαναλάβει την αναπαραγωγή του βίντεο.</p>
Μετάπειτα κατάσταση :	<p>Ανάλογα με τον τύπο της κάρτας που έχει τραβηχτεί οι επόμενες φάσεις μπορεί να είναι οι εξής :</p> <p>1. Η κάρτα που έχει τραβηχτεί είναι κάρτα κατάστασης η οποία περιγράφει μία μη ασφαλή συμπεριφορά. Το σύστημα μεταπηδά στην περίπτωση χρήσης μη ασφαλούς κάρτας (ΠΧ07).</p> <p>2. Η κάρτα που έχει τραβηχτεί είναι είτε κάρτα joker είτε κάρτα κατάστασης η οποία περιγράφει μία ασφαλή συμπεριφορά ή μία μη επικίνδυνη συμπεριφορά. Το σύστημα μεταπηδά στην ανάλογη περίπτωση χρήσης (ΠΧ09).</p> <p>3. Η κάρτα που έχει τραβηχτεί ανήκει στην κατηγορία των καρτών ερωτήσεων. Το σύστημα μεταπηδά στην περίπτωση χρήσης κάρτα ερώτησης (ΠΧ08).</p>
Εναλλακτικές ροές γεγονότων και εξαιρέσεις :	<p>1-2α. Ο χειριστής διακόπτει και επιλέγει <i>Νέο παιχνίδι</i> (ΠΧ01) ή <i>Φόρτωση παιχνιδιού</i> (ΠΧ03).</p> <p>3. Παύση του παιχνιδιού Ενημέρωση του χειριστή ότι ένα παιχνίδι είναι σε εξέλιξη και αν επιθυμεί την διακοπή του.</p> <p>4. Σε θετική απάντηση του χειριστή το σύστημα προχωράει αναλόγως στην ΠΧ01 ή ΠΧ03 , σε αρνητική απάντηση συνεχίζεται η διαδικασία.</p> <p>1-2β. Ο χειριστής διακόπτει και επιλέγει <i>Αποθήκευση</i></p>

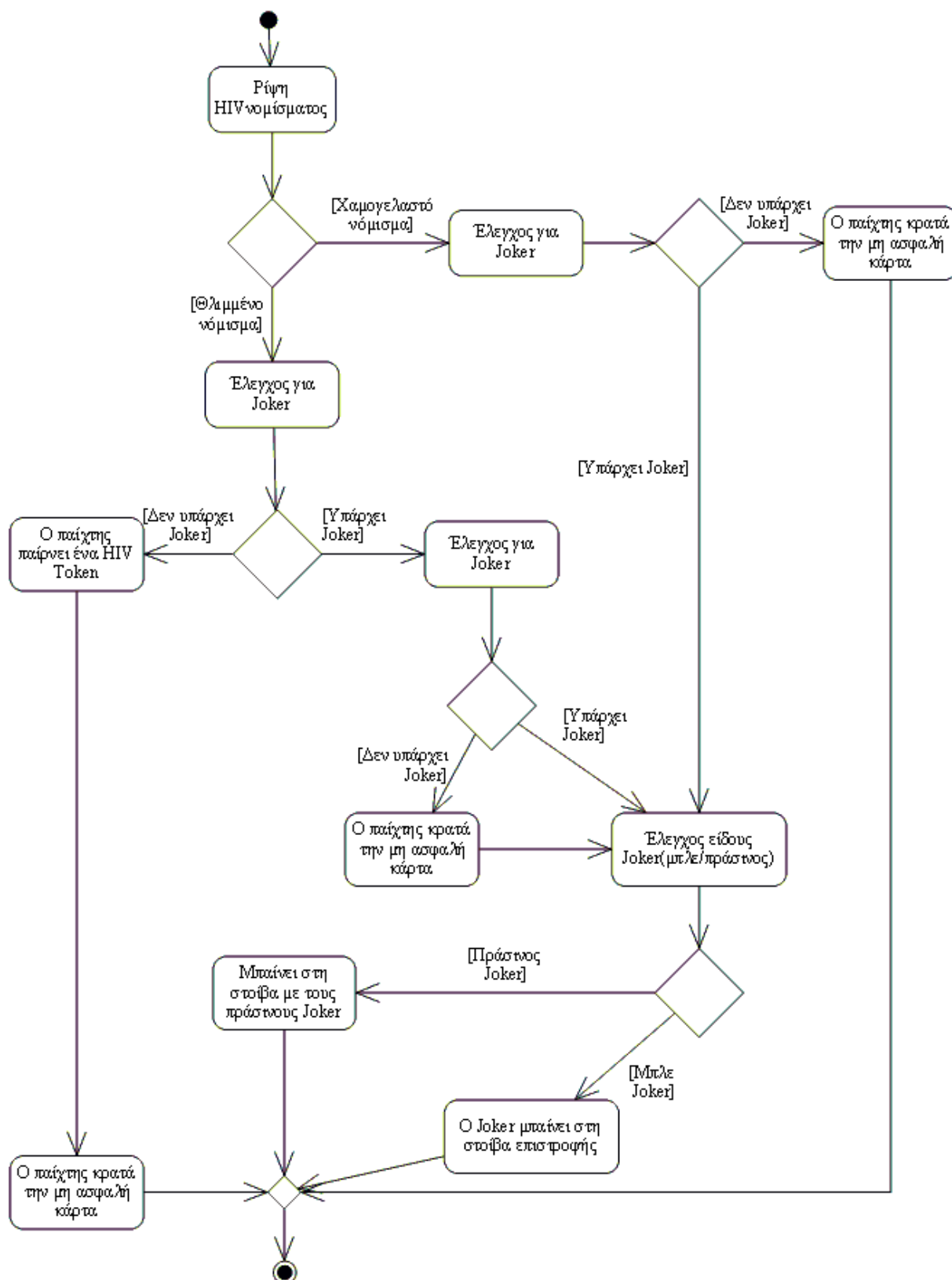
Περίπτωση χρήσης :	ΠΧ06
	<i>παιχνιδιού</i> (ΠΧ02), τότε το σύστημα προχωρά στην ΠΧ02.

#### 4.3.2.7 Περίπτωση Χρήσης : Μη ασφαλής κάρτα κατάστασης

Περίπτωση χρήσης :	ΠΧ07
Χειριστής :	Χρήστης <i>παίχτης</i>
Περιγραφή :	Ο παίχτης έχει τραβήξει μία κάρτα κατάστασης που περιγράφει μία μη ασφαλή συμπεριφορά.
Προϋποθέσεις :	Απαιτείται το τράβηγμα της κάρτας από την στοίβα με τις κάρτες (ΠΧ06).
Κύρια ροή γεγονότων :	<ol style="list-style-type: none"> <li>1. Σύμφωνα με τους κανόνες του παιχνιδιού ο παίχτης που έχει τραβήξει μη ασφαλή κάρτα πρέπει να ρίξει το HIV νόμισμα.</li> <li>2. Αν το νόμισμα είναι αρνητικό στον ιό (χαμογελαστό), τότε το σύστημα ελέγχει αν ο συγκεκριμένος παίχτης κατέχει κάποιο joker. Όταν δεν υπάρχει joker τότε ο παίχτης κρατά την μη ασφαλή κάρτα, ενώ αν υπάρχει η κάρτα πηγαίνει στην στοίβα επιστροφής. Το ίδιο συμβαίνει και με τον joker αν πρόκειται για μπλε πηγαίνει στην στοίβα επιστροφής, αν πρόκειται για πράσινο επιστρέφεται στην στοίβα με τους πράσινους joker.</li> <li>3. Αν το νόμισμα είναι θετικό στον ιό (θλιμμένο), τότε το σύστημα ελέγχει την ύπαρξη joker. Αν δεν υπάρχει κάποιος τότε ο παίχτης κρατά την μη ασφαλή κάρτα και ένα HIV Token προστίθεται στον παίχτη. Όταν υπάρχει ένας joker το σύστημα ελέγχει για την ύπαρξη και δεύτερου joker. Αν δεν υπάρχει ο δεύτερος τότε ο παίχτης κρατά την μη ασφαλή κάρτα και ο joker πηγαίνει στην ανάλογη στοίβα όπως στη περίπτωση 2. Όταν υπάρχει και ο δεύτερος joker, τότε ο παίχτης</li> </ol>

Περίπτωση χρήσης :	ΠΧ07
	επιστρέφει τους δύο jokers στην ανάλογη στοίβα.
Μετάπειτα κατάσταση :	Το σύστημα μεταπηδά στην περίπτωση χρήσης επιλογή παίχτη (ΠΧ05).
Εναλλακτικές ροές γεγονότων και εξαιρέσεις :	<p>1-3α. Ο χειριστής διακόπτει και επιλέγει <i>Νέο παιχνίδι</i> (ΠΧ01) ή <i>Φόρτωση παιχνιδιού</i> (ΠΧ03).</p> <p>5. Παύση του παιχνιδιού Ενημέρωση του χειριστή ότι ένα παιχνίδι είναι σε εξέλιξη και αν επιθυμεί την διακοπή του.</p> <p>6. Σε θετική απάντηση του χειριστή το σύστημα προχωράει αναλόγως στην ΠΧ01 ή ΠΧ03 , σε αρνητική απάντηση συνεχίζεται η διαδικασία.</p> <p>1-3β. Ο χειριστής διακόπτει και επιλέγει <i>Αποθήκευση παιχνιδιού</i> (ΠΧ02), τότε το σύστημα προχωρά στην ΠΧ02.</p>

Ακολουθεί το διάγραμμα της κύριας ροής γεγονότων της ΠΧ07 στην Εικόνα 4-2 .



Εικόνα 4-2 Κύρια ροή γεγονότων στην κατά το τράβηγμα μη ασφαλούς κάρτας



#### 4.3.2.8 Περίπτωση Χρήσης : Κάρτας ερώτησης

Περίπτωση χρήσης :	ΠΧ08
Χειριστής :	Χρήστης <i>παίχτης</i>
Περιγραφή :	Ο παίχτης έχει τραβήξει μία κάρτα ερώτησης.
Προϋποθέσεις :	Απαιτείται το τράβηγμα της κάρτας από την στοίβα με τις κάρτες (ΠΧ06).
Κύρια ροή γεγονότων :	<ol style="list-style-type: none"> <li>1. Ο χειριστής απαντά στην ερώτηση που έχει εμφανιστεί με δυνατές επιλογές σωστό ή λάθος.</li> <li>2. Γίνεται έλεγχος της απάντησης από το σύστημα και αν έχει απαντηθεί σωστά η ερώτηση ο παίχτης επιβραβεύεται με έναν πράσινο joker, όταν η απάντηση είναι λάθος τότε δεν υπάρχει επιβράβευση.</li> <li>3. Η κάρτα ερώτησης πηγαίνει στην στοίβα επιστροφής.</li> </ol>
Μετάπειτα κατάσταση :	Το σύστημα μεταπηδά στην περίπτωση χρήσης επιλογή παίχτη (ΠΧ05).
Εναλλακτικές ροές γεγονότων και εξαιρέσεις :	<p>1-3α. Ο χειριστής διακόπτει και επιλέγει <i>Νέο παιχνίδι</i> (ΠΧ01) ή <i>Φόρτωση παιχνιδιού</i> (ΠΧ03).</p> <p>7. Παύση του παιχνιδιού Ενημέρωση του χειριστή ότι ένα παιχνίδι είναι σε εξέλιξη και αν επιθυμεί την διακοπή του.</p> <p>8. Σε θετική απάντηση του χειριστή το σύστημα προχωράει αναλόγως στην ΠΧ01 ή ΠΧ03 , σε αρνητική απάντηση συνεχίζεται η διαδικασία.</p> <p>1-3β. Ο χειριστής διακόπτει και επιλέγει <i>Αποθήκευση παιχνιδιού</i> (ΠΧ02), τότε το σύστημα προχωρά στην ΠΧ02.</p>

#### 4.3.2.9 Περίπτωση Χρήσης : Λοιπές κάρτες κατάστασης και joker

Περίπτωση χρήσης :	ΠΧ09
Χειριστής :	Χρήστης <i>παίχτης</i>
Περιγραφή :	Ο παίχτης έχει τραβήξει από την στοίβα κάρτα joker ή μία κάρτα κατάστασης που περιγράφει ασφαλή ή μη

Περίπτωση χρήσης :	ΠΧ09
	επικίνδυνη κατάσταση.
Προϋποθέσεις :	Απαιτείται το τράβηγμα της κάρτας από την στοίβα με τις κάρτες (ΠΧ06).
Κύρια ροή γεγονότων :	1. Το σύστημα περιμένει από τον χειριστή επιβεβαίωση ότι διάβασε την κάρτα.
Μετάπειτα κατάσταση :	Το σύστημα μεταπηδά στην περίπτωση χρήσης επιλογή παίχτη (ΠΧ05).
Εναλλακτικές ροές γεγονότων και εξαιρέσεις :	<p>1α. Ο χειριστής διακόπτει και επιλέγει <i>Νέο παιχνίδι</i> (ΠΧ01) ή <i>Φόρτωση παιχνιδιού</i> (ΠΧ03).</p> <p>9. Παύση του παιχνιδιού Ενημέρωση του χειριστή ότι ένα παιχνίδι είναι σε εξέλιξη και αν επιθυμεί την διακοπή του.</p> <p>10. Σε θετική απάντηση του χειριστή το σύστημα προχωράει αναλόγως στην ΠΧ01 ή ΠΧ03 , σε αρνητική απάντηση συνεχίζεται η διαδικασία.</p> <p>1β. Ο χειριστής διακόπτει και επιλέγει <i>Αποθήκευση παιχνιδιού</i> (ΠΧ02), τότε το σύστημα προχωρά στην ΠΧ02.</p>

#### 4.3.2.10 Περίπτωση Χρήσης : Διάβασμα ερώτησης βιβλίου

Περίπτωση χρήσης :	ΠΧ10
Χειριστής :	Χρήστης <i>παίχτης</i>
Περιγραφή :	Ο παίχτης διαβάζει μία ερώτηση βιβλίου και την απάντηση της.
Προϋποθέσεις :	Το σύστημα μπαίνει σε αυτή τη περίπτωση χρήσης μετά την επιλογή παίχτη (ΠΧ05) και αν ο παίχτης που επιλέχθηκε έχει επιδεινωμένη υγεία (όριο HIV Token) ή βρίσκεται σε δημόσιο πρόγραμμα εκπαίδευσης (όριο μη ασφαλών καρτών).
Κύρια ροή γεγονότων :	1. Το σύστημα παρουσιάζει την ερώτηση βιβλίου και περιμένει επιβεβαίωση ότι διαβάστηκε.

Περίπτωση χρήσης :	ΠΧ10
Μετέπειτα κατάσταση :	Αν ο παίχτης έχει επιδεινωμένη υγεία τότε το σύστημα συνεχίζει στην επιλογή παίχτη (ΠΧ05), ενώ αν βρίσκεται σε δημόσιο πρόγραμμα εκπαίδευσης συνεχίζει με το τράβηγμα μίας κάρτας από την στοίβα (ΠΧ06).
Εναλλακτικές ροές γεγονότων και εξαιρέσεις :	Καμία

#### 4.3.3 Προδιαγραφές περιπτώσεων χρήσης δευτερεύοντος προϊόντος.

Στην Εικόνα 4-3 που βρίσκεται στο παράρτημα του κεφαλαίου διακρίνουμε το διάγραμμα περιπτώσεων χρήσης.

##### 4.3.3.1 Περίπτωση Χρήσης : Ρύθμιση παραμέτρων, καρτών και ερωτήσεων

Περίπτωση χρήσης :	ΠΧ11
Χειριστής :	Χρήστης διαχειριστής
Περιγραφή :	Ο χειριστής ορίζει τις παραμέτρους, τις κάρτες και τις ερωτήσεις του παιχνιδιού.
Προϋποθέσεις :	Καμία
Κύρια ροή γεγονότων :	Ο χειριστής ορίζει μέσω της φόρμας συμπλήρωσης στοιχείων το όριο των HIV Token για την κατάσταση επιδεινωμένης υγείας, το όριο των μη ασφαλών καρτών για να μπει κάποιος σε δημόσιο πρόγραμμα εκπαίδευσης, το πλήθος των μπλε και πράσινων joker, τις κάρτες του παιχνιδιού και τις ερωτήσεις-απαντήσεις βιβλίου.
Μετέπειτα κατάσταση :	Καμία
Εναλλακτικές ροές γεγονότων και εξαιρέσεις :	Καμία

##### 4.3.3.2 Περίπτωση Χρήσης : Αποθήκευση αρχείου ρυθμίσεων

Περίπτωση χρήσης :	ΠΧ12
Χειριστής :	Χρήστης διαχειριστής

Περίπτωση χρήσης :	ΠΧ12
Περιγραφή :	Ο χειριστής μπορεί να αποθηκεύσει τα δεδομένα σε αρχείο.
Προϋποθέσεις :	Καμία
Κύρια ροή γεγονότων :	<ol style="list-style-type: none"> <li>1. Ο χειριστής από το μενού επιλέγει <b>Αποθήκευση αρχείου</b>.</li> <li>2. Μέσω του παραθύρου που εμφανίζεται γίνεται η επιλογή της διαδρομής στο δίσκο και του ονόματος του αρχείου.</li> <li>3. Στο αρχείο που ορίστηκε αποθηκεύονται τα δεδομένα.</li> </ol>
Μετάπειτα κατάσταση :	Καμία
Εναλλακτικές ροές γεγονότων και εξαιρέσεις :	3α. Δεν μπορεί να γίνει αποθήκευση των δεδομένων λόγω προστασίας ανάγνωσης ή κάποιου άλλου σφάλματος.

#### 4.3.3.3 Περίπτωση Χρήσης : Επανάκτηση αρχείου ρυθμίσεων

Περίπτωση χρήσης :	ΠΧ13
Χειριστής :	Χρήστης διαχειριστής
Περιγραφή :	Ο χειριστής μπορεί να επανακτήσει τα δεδομένα από ήδη υπάρχον αρχείο.
Προϋποθέσεις :	Καμία
Κύρια ροή γεγονότων :	<ol style="list-style-type: none"> <li>1. Ο χειριστής από το μενού επιλέγει <b>Φόρτωση αρχείου</b>.</li> <li>2. Μέσω του παραθύρου που εμφανίζεται γίνεται η επιλογή της διαδρομής στο δίσκο και του ονόματος του αρχείου που περιέχει τα δεδομένα.</li> <li>3. Από το αρχείο που ορίστηκε ανακτώνται τα δεδομένα και μεταφέρονται στη φόρμα.</li> </ol>
Μετάπειτα κατάσταση :	Καμία

Περίπτωση χρήσης :	ΠΧ13
Εναλλακτικές ροές γεγονότων και εξαιρέσεις :	3α. Δεν μπορεί να γίνει επανάκτηση των δεδομένων επειδή δεν υπάρχουν στο αρχείο ή κάποιου άλλου σφάλματος ανάγνωσης.

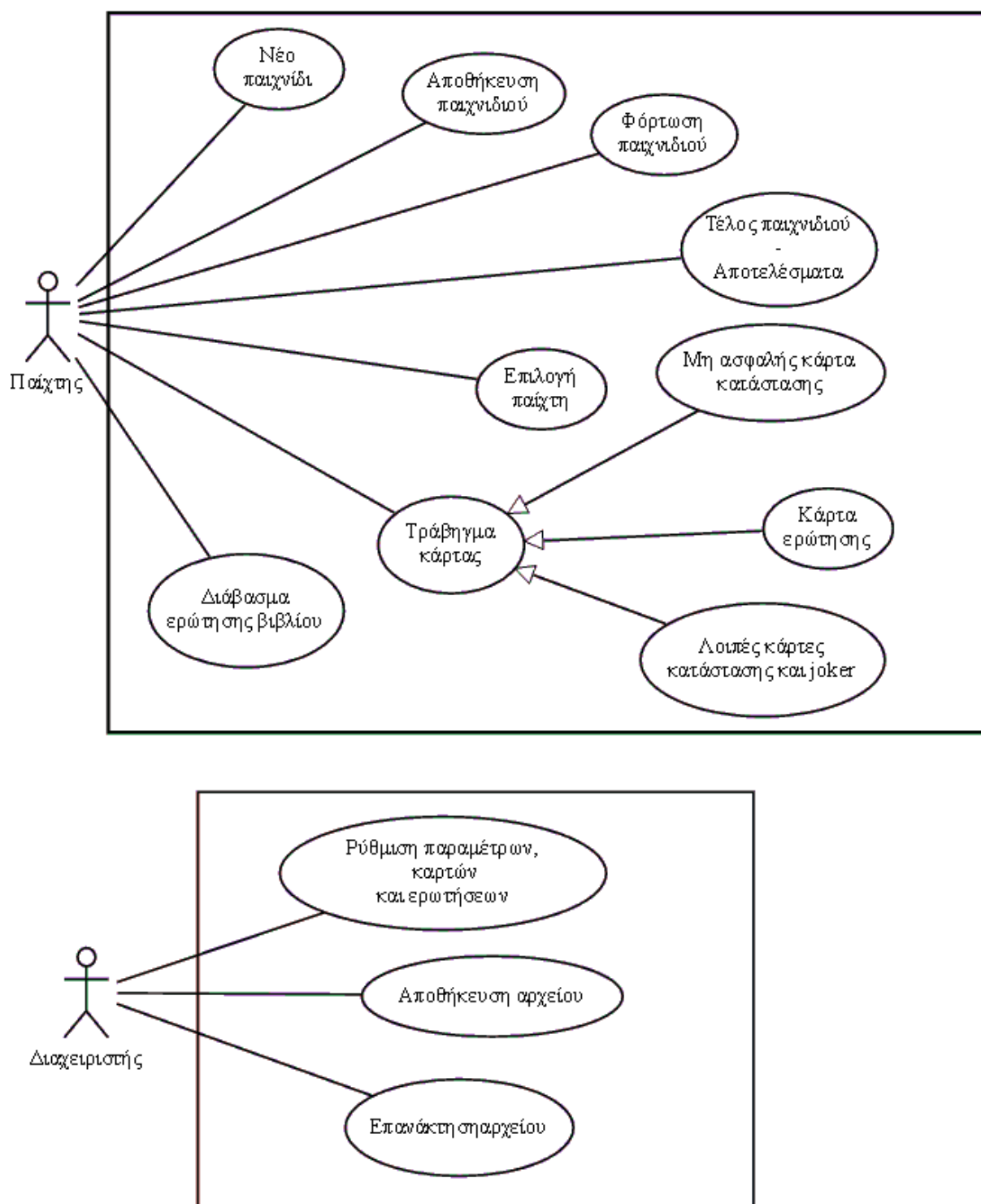
#### **4.3.4 Προδιαγραφές επιδόσεων**

Απαραίτητη προδιαγραφή είναι η δυνατότητα παύσης του παιχνιδιού οποιαδήποτε στιγμή το χρειαστεί ο χρήστης. Παύση του παιχνιδιού απαιτείται όταν κάποια άλλη εφαρμογή εμφανιστεί στο προσκήνιο ώστε να μην κυλάει το παιχνίδι χωρίς να το παρακολουθεί ο χρήστης.

Μία επιπλέον προδιαγραφή είναι η δυνατότητα παύσης του ήχου του παιχνιδιού όπως και η επαναφορά του οποιαδήποτε στιγμή επιθυμεί ο χρήστης.

### 4.3.5 Παράρτημα

#### 4.3.5.1 Διάγραμμα περιπτώσεων χρήσης



Εικόνα 4-3 Διάγραμμα περιπτώσεων χρήσης

## 5. Ανάλυση

Βασικός σκοπός της ανάλυσης είναι να προσδιορίσουμε τι ακριβώς το σύστημα πρέπει να κάνει χρησιμοποιώντας τη γλώσσα των κατασκευαστών. Πρόκειται για ένα ενδιάμεσο στάδιο μεταξύ του προσδιορισμού απαιτήσεων και της λεπτομερούς σχεδίασης.

Οι απαιτήσεις ορίζουν ξεκάθαρα το διαχωρισμό δύο ξεχωριστών προγραμμάτων, το παιχνίδι και τον συντάκτη ρυθμίσεων με μοναδικό σημείο επαφής το αρχείο ρυθμίσεων. Έτσι θα ορίσουμε δύο ξεχωριστά πακέτα (Εικόνα 5-1), το SafeGame που αντιστοιχεί στο παιχνίδι και το SafeGameEditor που αντιστοιχεί στον συντάκτη. Η ανάλυσή τους θα γίνει ξεχωριστά.



Εικόνα 5-1 - Πακέτα της εφαρμογής

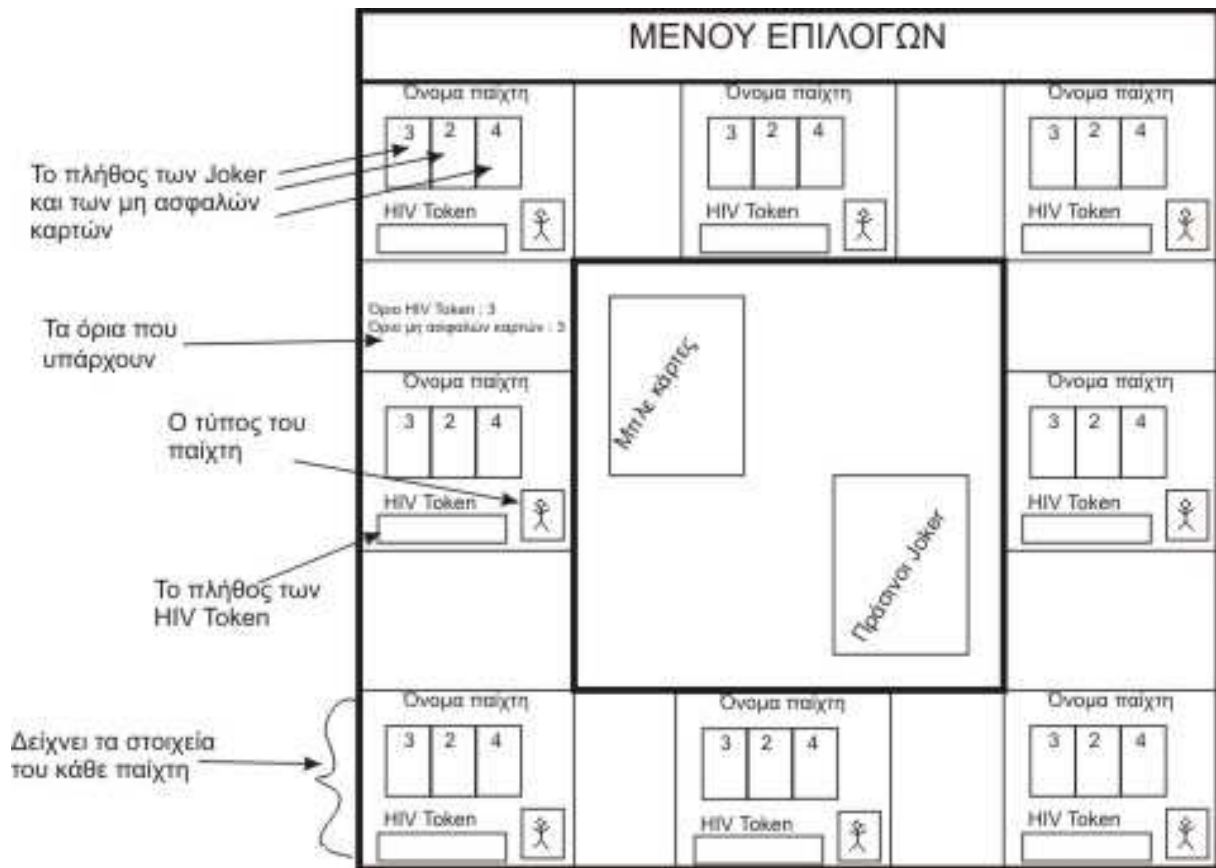
### 5.1 Ανάλυση του SafeGame

#### 5.1.1 Σχεδιασμός διεπαφής

Το SafeGame ως παιχνίδι βασίζεται σε μεγάλο βαθμό στα γραφικά, έτσι πρέπει αρχικά να σχεδιάσουμε την διεπαφή, ώστε στη συνέχεια να αναλύσουμε το ζητούμενο σύστημα. Από τις απαιτήσεις συμπεραίνουμε πως τα στοιχεία που θα πρέπει να αναπαραστήσουμε οπτικά είναι :

- i. Οι παίκτες με τα στοιχεία τους που είναι το όνομα, ο τύπος (άνθρωπος ή Η/Υ) και το πλήθος που πιθανόν να έχουν των HIV Token, των πράσινων και μπλε Joker και των μη ασφαλών καρτών.
- ii. Οπτική αναπαράσταση του τραβήγματος των καρτών από τον παίχτη εμφάνιση με του περιεχόμενου της κάρτας.

Μία πιθανή λύση φαίνεται στην Εικόνα 5-2.



Εικόνα 5-2 - Σχέδιο διεπαφής

Παρατηρούμε λοιπόν οχτώ ομάδες τετραγώνων που περιέχουν πληροφορίες για τον κάθε παίχτη. Οι ομάδες αυτές είναι διασκορπισμένες περιμετρικά ενός κύριου τετραγώνου, το οποίο αναπαριστά το ταμπλό του παιχνιδιού. Μέσα στο ταμπλό βλέπουμε δύο στοίβες καρτών, μία για τις μπλε κάρτες και μία για τους πράσινους Joker. Παρατηρούμε επίσης δύο παραμέτρους του παιχνιδιού που γίνονται ορατές στον παίχτη, τα όρια για τα HIV Token και για τις μη ασφαλείς κάρτες.

### 5.1.2 Αρχική διάκριση κλάσεων

Δύο βασικές κλάσεις οι οποίες ξεχωρίζουν πολύ εύκολα είναι αυτή που περιγράφει τους παίχτες του παιχνιδιού και αυτή που περιγράφει τις κάρτες του παιχνιδιού. Την κλάση που περιγράφει τους παίχτες του παιχνιδιού την ονομάζουμε `GamePlayer` και τα βασικά της στοιχεία τα βλέπουμε στην Εικόνα 5-3.





**Εικόνα 5-3 - Κλάση GamePlayer**

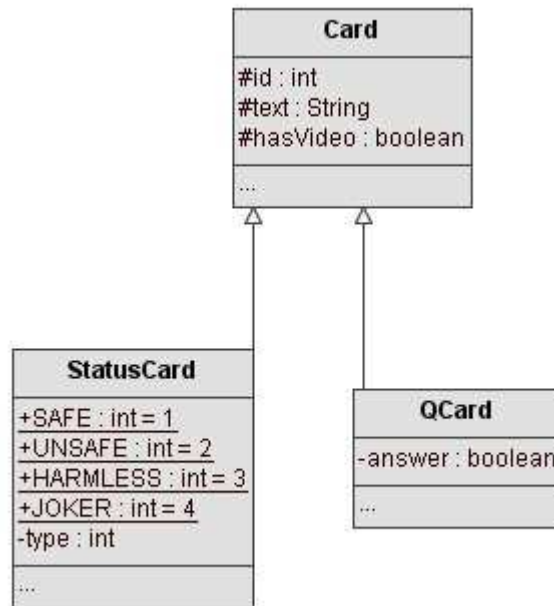
Ο παίχτης έχει όνομα, μπορεί να μαζέψει Joker, μη ασφαλείς κάρτες και HIV Token, επίσης μπορεί να είναι άνθρωπος ή όχι.

Μελετώντας τις μπλε κάρτες του παιχνιδιού όπως ορίζονται στις απαιτήσεις, οι πράσινες είναι μόνο Joker, μπορούμε να πούμε πως έχουν τα εξής χαρακτηριστικά :

- i. Κάποιο κείμενο που μπορεί να είναι είτε μία ερώτηση είτε κείμενο που περιγράφει κάποια κατάσταση. Μόνο οι μπλε κάρτες που είναι Joker δεν έχουν κείμενο.
- ii. Κάποιο βίντεο που αναπαριστά αυτό που γράφεται στο κείμενο. Επίσης οι κάρτες Joker δεν έχει νόημα να φέρουν βίντεο.
- iii. Αν πρόκειται για κάρτες ερωτήσεων, υπάρχει μία απάντηση στην ερώτηση του τύπου σωστό/λάθος.

Δεδομένου ότι οι κάρτες ανήκουν στις παραμέτρους του παιχνιδιού που ζητάμε να έχουν δυναμικό χαρακτήρα ώστε να τις τροποποιούμε με την χρήση του συντάκτη (SafeGameEditor), αποφασίζουμε να βάλουμε ακόμη ένα χαρακτηριστικό, το Id. Ένα πεδίο που θα προσδίδει στην κάρτα μοναδική ταυτότητα, ώστε να μπορούμε να την ξεχωρίζουμε από τις υπόλοιπες κατά την επεξεργασία αλλά και για λύσουμε το πρόβλημα της αντιστοιχίας μιας κάρτας με ένα αρχείο βίντεο. Έτσι μπορούμε χρησιμοποιήσουμε ως όνομα αρχείου βίντεο κάθε κάρτας το μοναδικό Id που υπάρχει στο αντικείμενο της κάρτας.

Χρησιμοποιώντας τα κοινά χαρακτηριστικά των καρτών και δεδομένου ότι δύο είναι οι μεγάλες ομάδες καρτών, ερωτήσεων και κατάστασης , μπορούμε να ορίσουμε τις κλάσεις που φαίνονται στην Εικόνα 5-4.



Εικόνα 5-4 - Κλάσεις Card, QCard και StatusCard

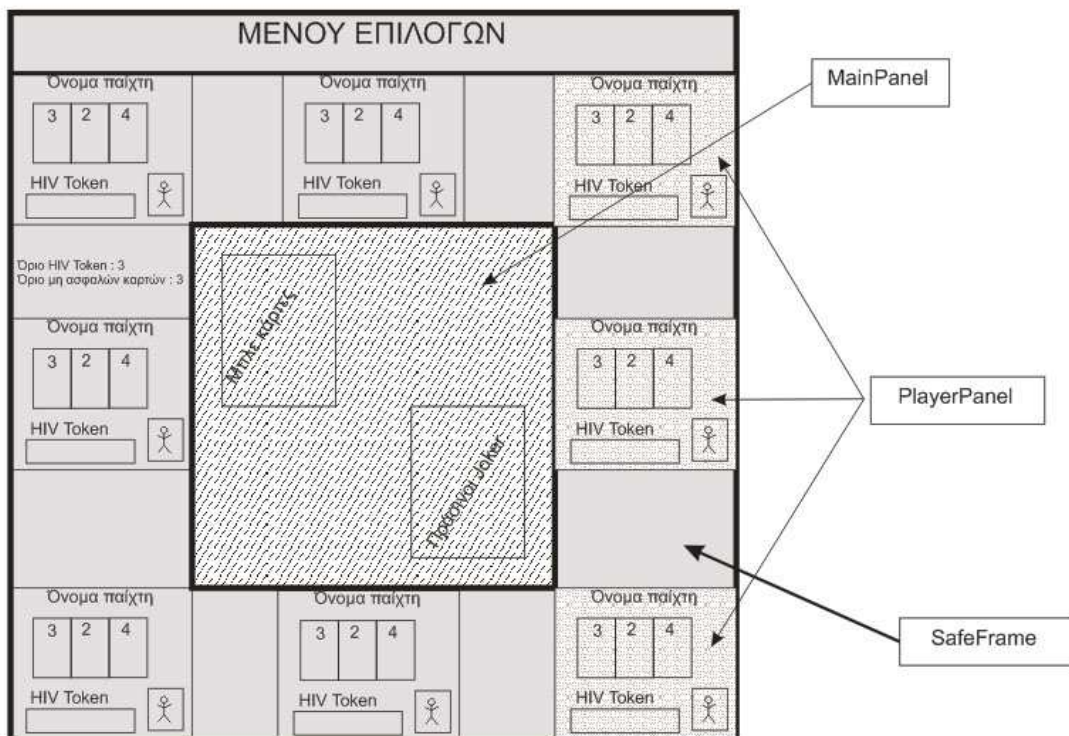
Κρατάμε ως κοινά χαρακτηριστικά στην κλάση Card το Id, το κείμενο και το αν έχει βίντεο η κάρτα και δημιουργούμε τις κλάσεις QCard για τις ερωτήσεις και StatusCard. Μοναδικό επιπλέον χαρακτηριστικό της QCard είναι η απάντηση (answer) που είναι τύπου boolean. Στην κλάση StatusCard διαχωρίσαμε τους τύπους των καρτών, συμπεριλαμβανομένων και των Joker, με μοναδικούς αριθμούς και ορίσαμε το αριθμητικό πεδίο type που περιέχει τον τύπο της κάρτας. Έτσι για να αναφερθούμε σε μία μη ασφαλή κάρτα δίνουμε στην type τη τιμή UNSAFE δηλαδή. Εδώ κάνουμε την παραδοχή πως οι Joker περιέχουν κενό κείμενο και δεν αντιστοιχίζονται με κάποιο βίντεο, ενώ δεν υπάρχει λόγος να έχουν μοναδική τιμή στο πεδίο Id.

### 5.1.3 Βασικές κλάσεις μηχανής παιχνιδιού και γραφικών

Πρωταρχικά ορίζουμε την κύρια κλάση που είναι υπεύθυνη για την λειτουργία του παιχνιδιού με το όνομα Main. Η Main είναι αυτή που θα επικοινωνήσει με τις κύριες παραθυρικές κλάσεις αλλά και με την κλάση που αντιπροσωπεύει την μηχανή του παιχνιδιού. Έπειτα ορίζουμε την κλάση η οποία θα είναι υπεύθυνη για την λειτουργία του παιχνιδιού και για τον χειρισμό των κλάσεων γραφικών, δηλαδή την μηχανή του παιχνιδιού, την ονομάζουμε GameEngine. Επειδή η GameEngine μπορεί να χειρίζεται κάποια δυναμική κίνηση γραφικών (animation), θεωρούμε πως θα πρέπει να εκτελείται σε ξεχωριστό νήμα από την υπόλοιπη εφαρμογή, έτσι

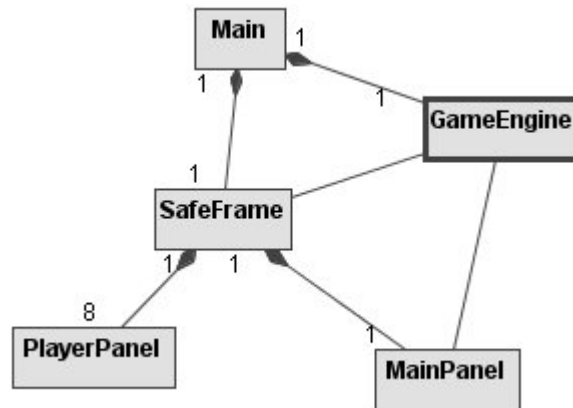
χρησιμοποιούμε τον όρο της UML ενεργή κλάση (active class) για να την χαρακτηρίσουμε.

Για τα γραφικά σε πρώτο επίπεδο θα θεωρήσουμε πως έχουμε μία κλάση η οποία θα χειριστεί το κύριο παράθυρο της εφαρμογής, την οποία ονομάζουμε SafeFrame και ακόμη δύο κλάσεις οι οποίες θα χειριστούν επιμέρους συστατικά του παραθύρου, την PlayerPanel που αντιστοιχεί στα στοιχεία του κάθε παίχτη και την MainPanel που αντιστοιχεί στο ταμπλό του παιχνιδιού. Η Εικόνα 5-5 δείχνει τον τομέα δράσης της κάθε κλάσης.



**Εικόνα 5-5 - Κύριες κλάσεις γραφικών**

Η SafeFrame είναι αυτή που χειρίζεται όλο το παράθυρο, όπως και το μενού του. Η PlayerPanel είναι υπεύθυνη για κάθε ένα τετράγωνο που αντιστοιχεί σε κάποιο παίχτη και μας παρουσιάζει τα στοιχεία του. Τέλος η MainPanel είναι αυτή που έχει τις κάρτες του παιχνιδιού και στον χώρο της θα εξελιχθεί όλο το παιχνίδι. Η Εικόνα 5-6 μας δείχνει ένα διάγραμμα των κλάσεων που μόλις ορίσαμε.



Εικόνα 5-6 - Διάγραμμα βασικών κλάσεων

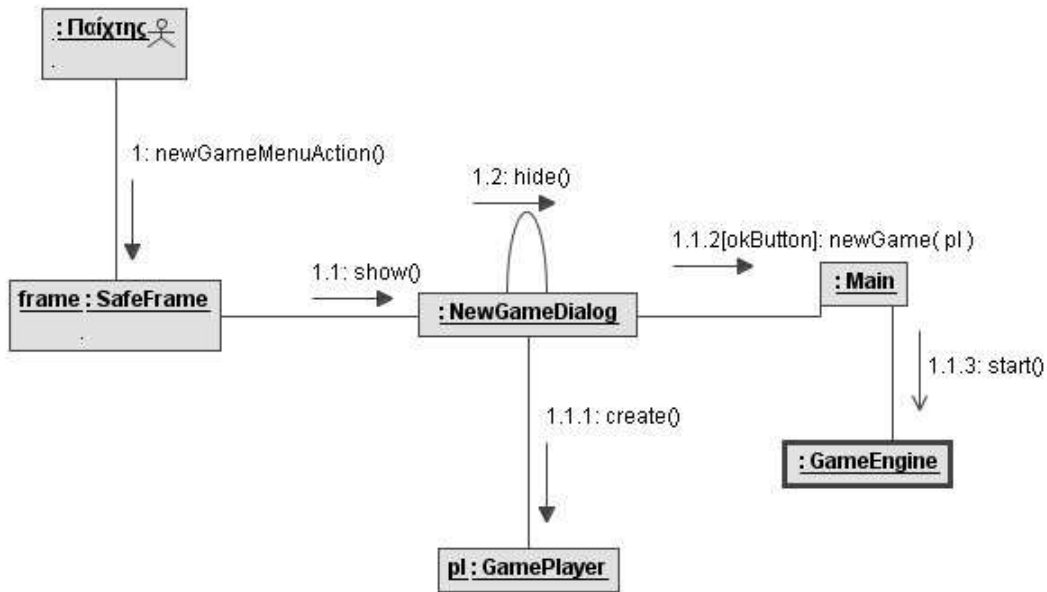
Στη φάση της σχεδίασης θα δούμε αναλυτικότερα περισσότερες κλάσεις για τα γραφικά με λεπτομέρεια.

#### 5.1.4 Ανάλυση περιπτώσεων χρήσης

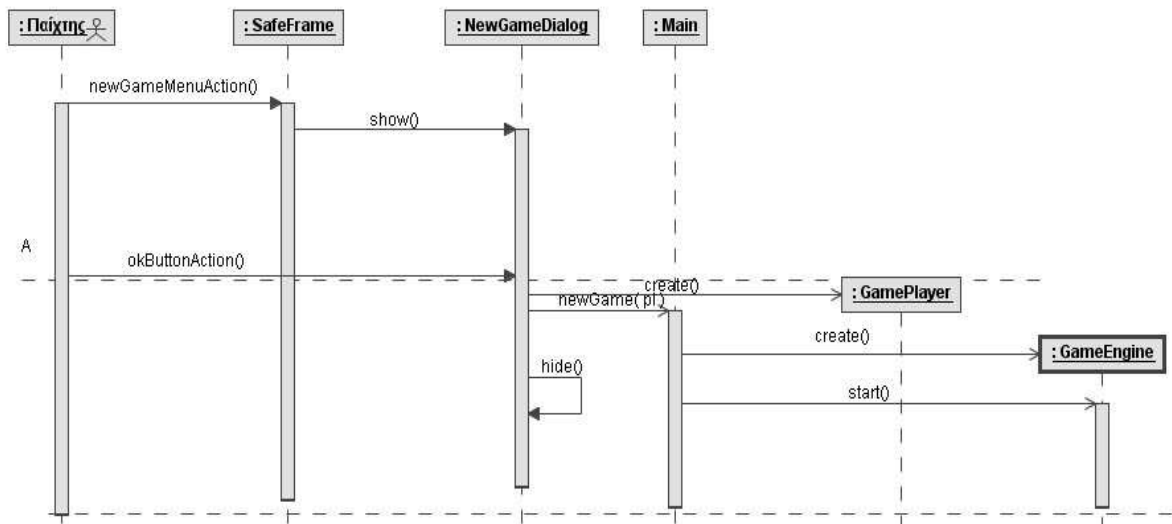
Στην συνέχεια θα προχωρήσουμε στην ανάλυση των περιπτώσεων χρήσης όπως δόθηκαν στο προηγούμενο κεφάλαιο. Η ανάλυση συνοδεύεται με τα απαραίτητα διαγράμματα συνεργασίας, ακολουθίας, κλάσεων και δραστηριοτήτων.

##### 5.1.4.1 Περίπτωση Χρήσης : Νέο παιχνίδι

Ο παίχτης επιλέγει από το μενού την δημιουργία νέου παιχνιδιού, ένα παράθυρο με επιλογές εμφανίζεται και ο παίχτης συμπληρώνει τις παραμέτρους που χρειάζονται για το νέο παιχνίδι. Δημιουργούμε μία νέα κλάση με το όνομα NewGameDialog, η οποία θα χειριστεί το παράθυρο διαλόγου. Η Εικόνα 5-7 δείχνει το διάγραμμα συνεργασίας μεταξύ των κλάσεων και η Εικόνα 5-7 το διάγραμμα ακολουθίας.



Εικόνα 5-7 - Διάγραμμα συνεργασίας νέου παιχνιδιού

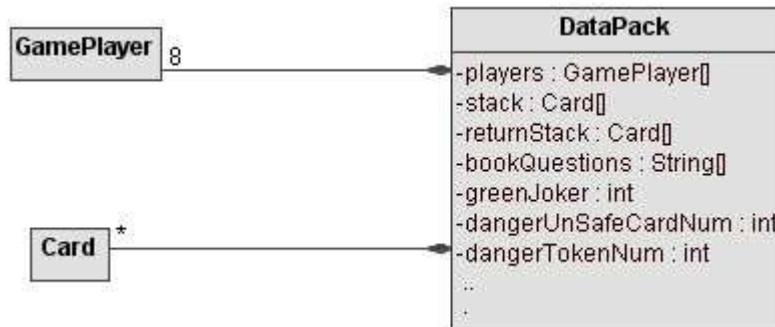


Εικόνα 5-8 - Διάγραμμα ακολουθίας νέου παιχνιδιού

Η SafeFrame στέλνει ένα μήνυμα στην NewGameDialog για να εμφανιστεί στην οθόνη και όταν ο παίχτης, αφού έχει συμπληρώσει τα στοιχεία, πατήσει το OK, η NewGameDialog στέλνει ένα μήνυμα στην Main για την δημιουργία νέου παιχνιδιού με έναν πίνακα από αντικείμενα παιχτών που έχει προηγουμένως δημιουργήσει. Η Main με την σειρά της στέλνει ένα ασύγχρονο μήνυμα στην GameEngine ώστε να ξεκινήσει σε νέο νήμα ένα παιχνίδι.

Για να ξεκινήσει ένα παιχνίδι, εκτός από τον προσδιορισμό των παιχτών η εφαρμογή χρειάζεται δεδομένα όπως τις κάρτες του παιχνιδιού και άλλες παραμέτρους. Οι πληροφορίες αυτές θα ληφθούν από το αρχείο ρυθμίσεων, το οποίο δημιουργείται από το δεύτερο πρόγραμμα, τον συντάκτη των παραμέτρων -

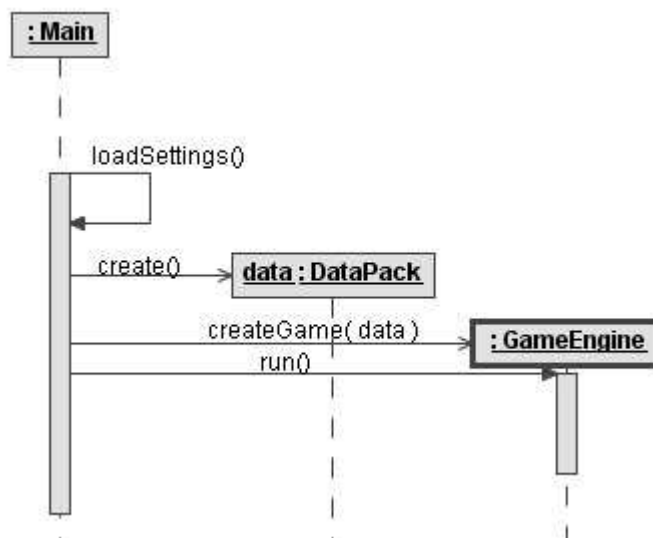
ρυθμίσεων. Δημιουργούμε μία κλάση με το όνομα DataPack η οποία θα περιέχει όλες τις απαραίτητες πληροφορίες για να ξεκινήσει ένα νέο παιχνίδι, η Εικόνα 5-9 δείχνει κάποια από τα στοιχεία που έχει.



Εικόνα 5-9 - Κλάση DataPack

Απαραίτητες πληροφορίες για την αρχικοποίηση ενός παιχνιδιού είναι μία λίστα με τις μπλε κάρτες και οι παράμετροι του παιχνιδιού, δηλαδή οι ερωτήσεις βιβλίου, οι πράσινοι joker και τα όρια στις μη ασφαλές κάρτες και τα HIV Token. Μπορούμε να επεκτείνουμε την εμβέλεια της κλάσης ώστε να χρησιμοποιηθεί και κατά την φόρτωση και αποθήκευση παιχνιδιών προσθέτοντας μία λίστα από παίχτες (GamePlayer) και μία δεύτερη λίστα από κάρτες η οποία θα χρησιμοποιηθεί ως στοίβα επιστροφής.

Όταν εκκινούμε ένα νέο παιχνίδι, διαβάζουμε το αρχείο των ρυθμίσεων σε ένα αντικείμενο τύπου DataPack και το τροφοδοτούμε στην GameEngine ώστε να αρχικοποιηθεί. Ακολουθεί το διάγραμμα ακολουθίας στην Εικόνα 5-10 για την αρχικοποίηση του παιχνιδιού.

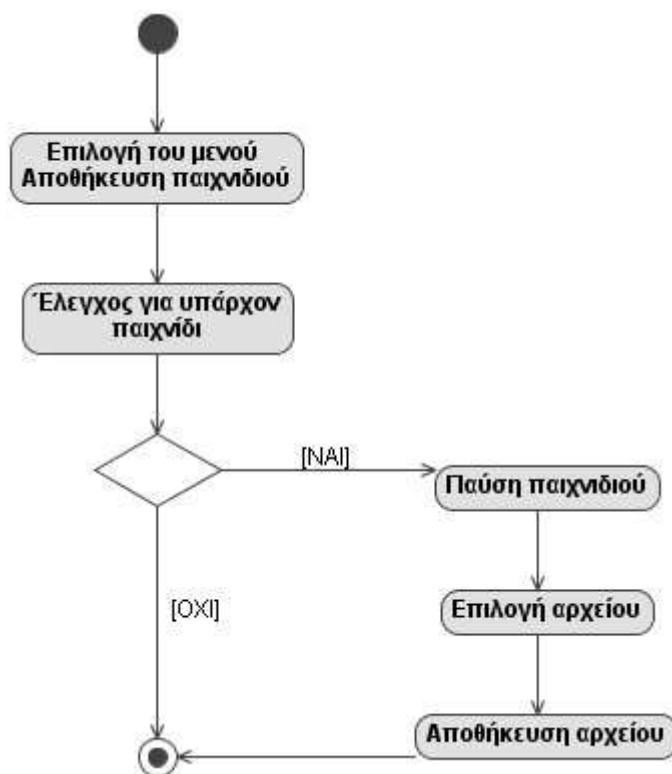


Εικόνα 5-10 - Διάγραμμα ακολουθίας αρχικοποίησης παιχνιδιού

#### 5.1.4.2 Περίπτωση Χρήσης : Αποθήκευση παιχνιδιού

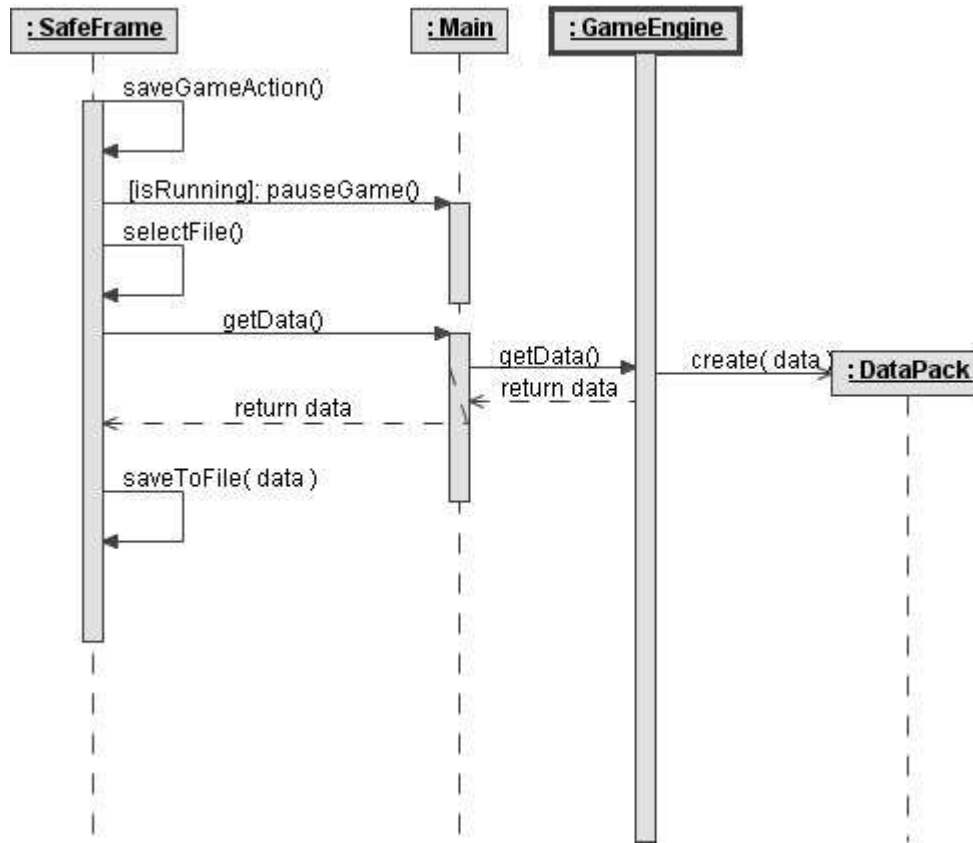
Ο χρήστης αφού έχει αρχίσει να παίζει κάποιο παιχνίδι, επιλέγει στο μενού αποθήκευση παιχνιδιού. Απαραίτητες πληροφορίες για να επαναφέρουμε ένα παιχνίδι είναι μία λίστα από τους παίκτες που έπαιζαν εκείνη τη στιγμή, μία λίστα με τις μπλε κάρτες που υπήρχαν στο παιχνίδι και οι παράμετροι του παιχνιδιού εκείνη τη στιγμή, όπως οι ερωτήσεις βιβλίου, οι πράσινοι joker και τα όρια στις μη ασφαλείς κάρτες και τα HIV Token.

Η Εικόνα 5-11 μας δείχνει το διάγραμμα δραστηριοτήτων της αποθήκευσης παιχνιδιού.



Εικόνα 5-11 - Διάγραμμα δραστηριοτήτων αποθήκευσης παιχνιδιού

Στην αποθήκευση παιχνιδιού χρησιμοποιείται η κλάση DataPack, η Εικόνα 5-12 μας δείχνει το ανάλογο διάγραμμα ακολουθίας.

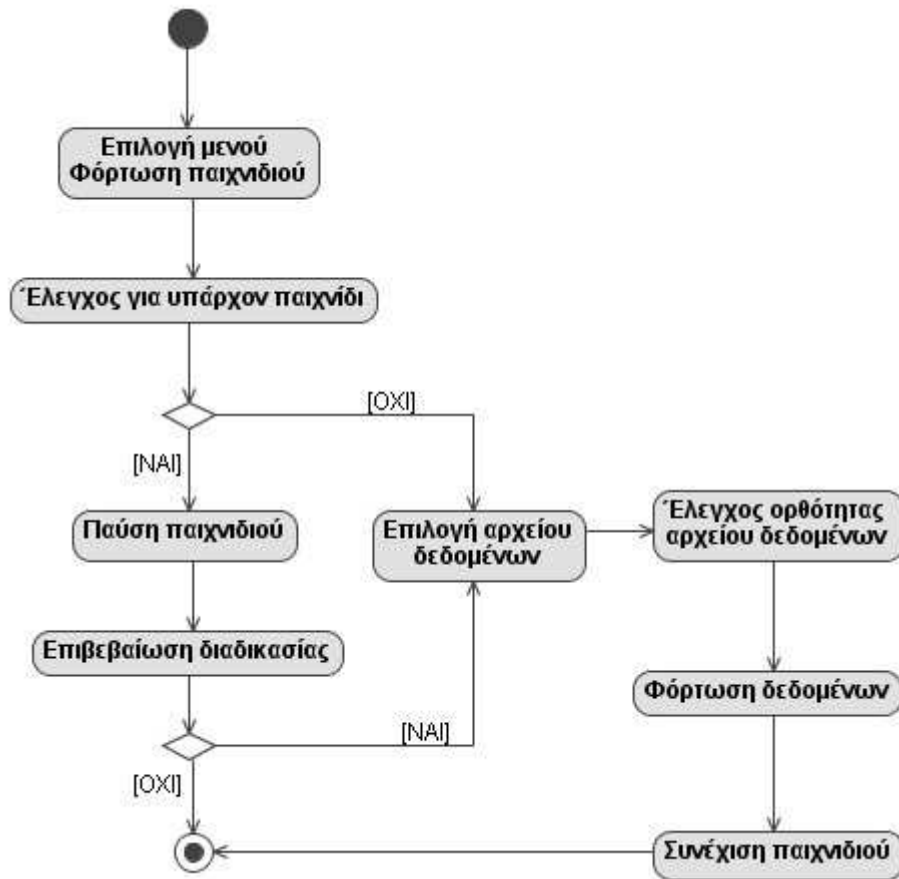


Εικόνα 5-12 - Διάγραμμα ακολουθίας αποθήκευσης παιχνιδιού

#### 5.1.4.3 Περίπτωση Χρήσης : Φόρτωση παιχνιδιού

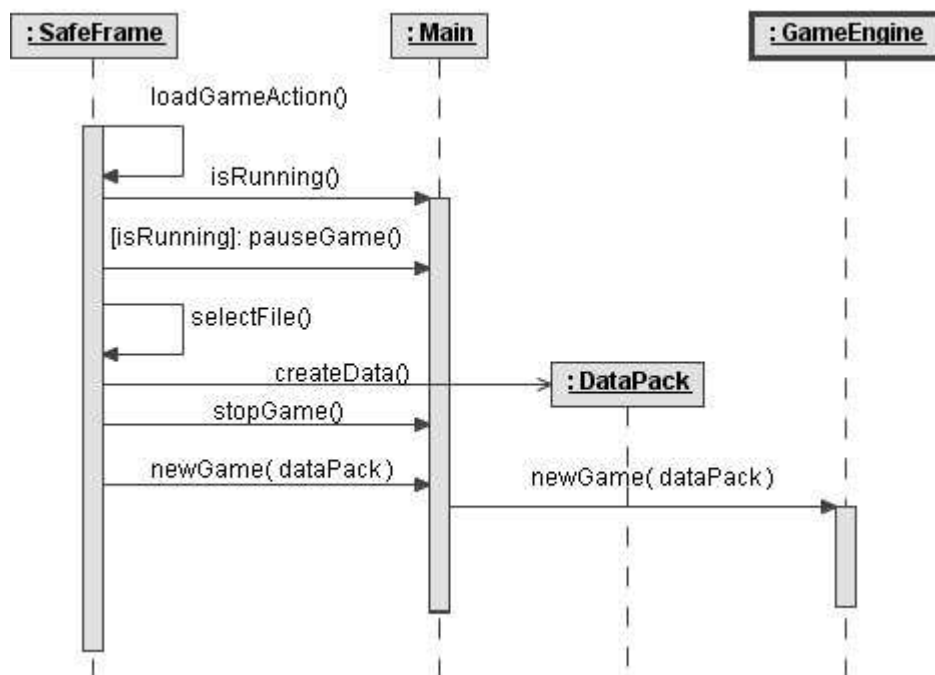
Όμοιο τρόπο λειτουργίας με την αποθήκευση παιχνιδιού χρησιμοποιεί η φόρτωση παιχνιδιού χρησιμοποιώντας την κλάση DataPack που περιέχει τις πληροφορίες του παιχνιδιού. Στην Εικόνα 5-13 βλέπουμε το διάγραμμα δραστηριοτήτων της φόρτωσης παιχνιδιού, όπου γίνονται οι απαραίτητοι έλεγχοι για την ομαλή λειτουργία του παιχνιδιού.





Εικόνα 5-13 - Διάγραμμα δραστηριοτήτων φόρτωσης παιχνιδιού

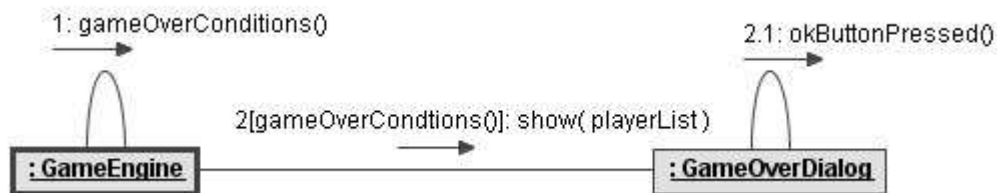
Η Εικόνα 5-14 με το διάγραμμα ακολουθίας μας δείχνει τον τρόπο αλληλεπίδρασης των κλάσεων.



Εικόνα 5-14 - Διάγραμμα ακολουθίας φόρτωσης παιχνιδιού

#### 5.1.4.4 Περίπτωση Χρήσης : Τέλος παιχνιδιού

Το τέλος του παιχνιδιού προκύπτει όταν όλοι οι παίχτες έχουν φτάσει στο όριο των HIV Token ή όταν η στοίβα με τις κάρτες έχει τελειώσει μία φορά, έχει ανακατευτεί και έχει τελειώσει και δεύτερη φορά. Όταν το παιχνίδι φτάσει στο τέλος του, εμφανίζεται μία καρτέλα που ανακοινώνει τα αποτελέσματα του παιχνιδιού. Για τον έλεγχο της καρτέλας αποτελεσμάτων ορίζουμε μία κλάση με το όνομα GameOverDialog. Η κλάση που χειρίζεται όλη την εξέλιξη του παιχνιδιού είναι η GameEngine, αυτή θα αποφασίσει πότε το παιχνίδι σταματά και θα στείλει μήνυμα με στην GameOverDialog να εμφανιστεί με τα αποτελέσματα. Στην Εικόνα 5-15 βλέπουμε το διάγραμμα συνεργασίας των δύο κλάσεων.

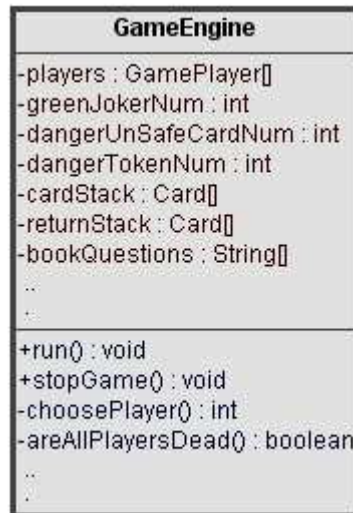


Εικόνα 5-15 - Διάγραμμα συνεργασίας τέλους παιχνιδιού

Η GameEngine στέλνει ένα μήνυμα στην GameOverDialog με την λίστα των παιχτών ως παράμετρο. Έτσι η GameOverDialog μπορεί να διακρίνει και να εμφανίσει τον νικητή.

#### 5.1.4.5 Περίπτωση Χρήσης : Επιλογή παίχτη

Η επιλογή παίχτη γίνεται κάθε φορά που ξεκινάει ένας γύρος παιχνιδιού. Όλο το παιχνίδι εξελίσσεται στην GameEngine και οι γύροι του παιχνιδιού αποτελούν μέρος ενός βρόχου που εκτελείται στην GameEngine. Στην Εικόνα 5-16 ορίζουμε κάποιες ιδιότητες και λειτουργίες της GameEngine.



**Εικόνα 5-16 - Κλάση GameEngine**

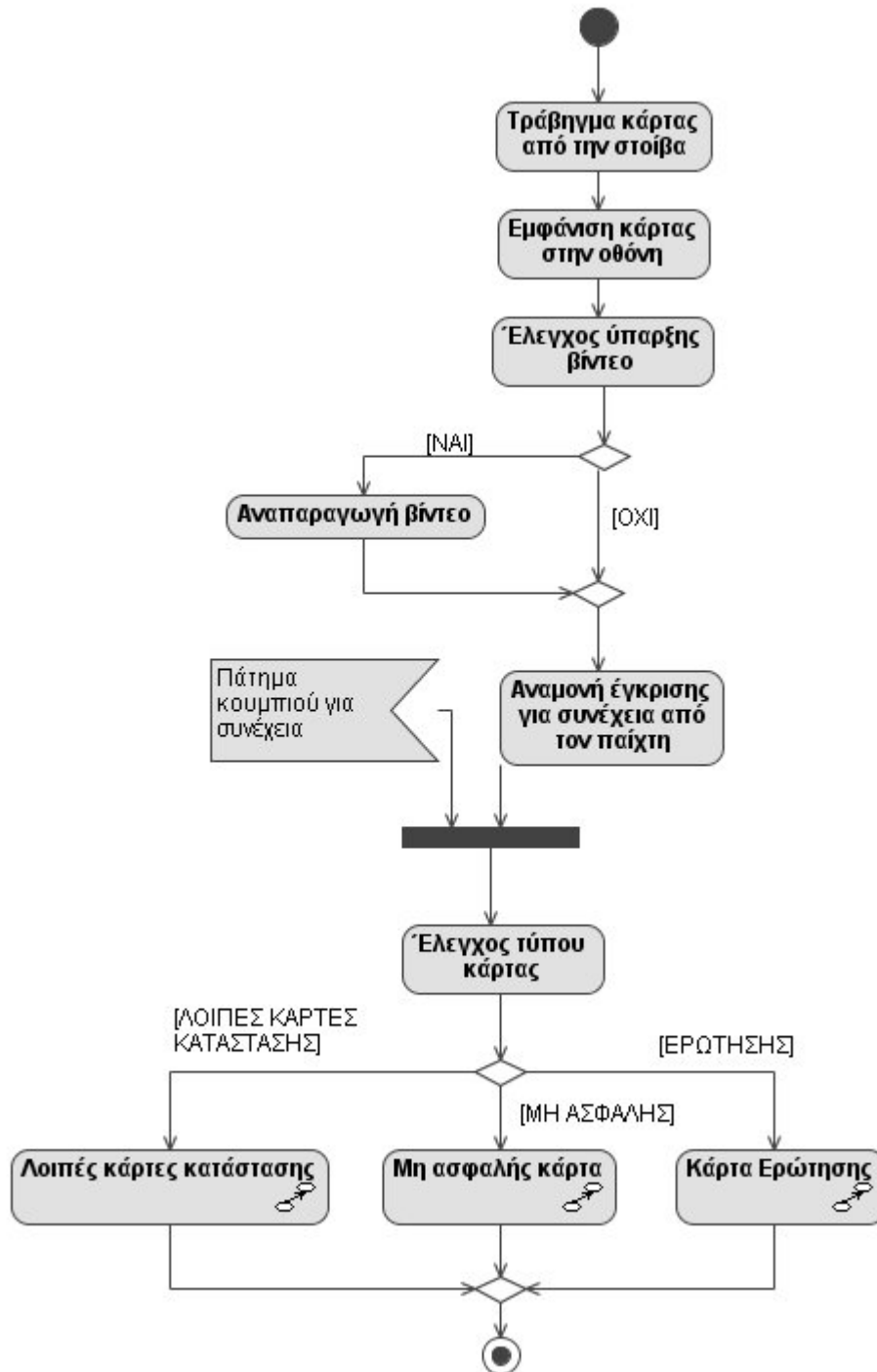
Στην λειτουργία run() τοποθετούμε την εκτέλεση του παιχνιδιού όπου υπάρχει ένας βρόχος που εκτελείται συνεχώς μέχρι το τέλος του παιχνιδιού. Κάθε επανάληψη του βρόχου αντιστοιχεί σε ένα γύρο του παιχνιδιού. Στην αρχή κάθε γύρου έχουμε την επιλογή του παίχτη που θα παίξει, η οποία εκτελείται από την λειτουργία choosePlayer(). Κατά την εκτέλεση της choosePlayer(), η GameEngine θα στείλει μήνυμα στην SafeFrame για την οπτική αναπαράσταση της επιλογής του παίχτη. Την ανταλλαγή των μηνυμάτων μεταξύ των κλάσεων βλέπουμε στην Εικόνα 5-17.



**Εικόνα 5-17 - Διάγραμμα συνεργασίας επιλογής παίχτη**

#### 5.1.4.6 Περίπτωση Χρήσης : Τράβηγμα κάρτας

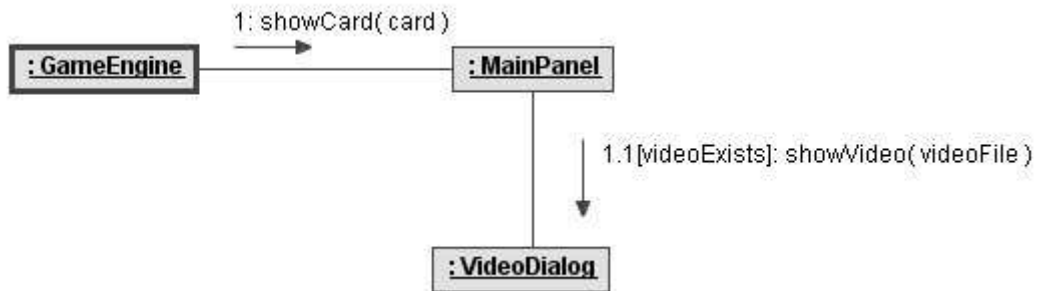
Το τράβηγμα κάρτας γίνεται αφού έχει επιλεγεί ο παίχτης και γίνεται αντιληπτό μέσα στο ταμπλό του παιχνιδιού. Εμφανίζουμε το κείμενο της κάρτας και αν υπάρχει κάποιο συσχετισμένο βίντεο. Στην Εικόνα 5-18 βλέπουμε το ανάλογο διάγραμμα δραστηριοτήτων.



**Εικόνα 5-18 - Διάγραμμα δραστηριοτήτων τραβήγματος κάρτας**

Η GameEngine αφού έχει επιλέξει τον παίχτη, ελέγχει αν ο παίχτης πρέπει να τραβήξει κάρτα και προχωρά στο τραβήγμα κάρτας στέλλοντας μήνυμα με την στην MainPanel, η οποία ελέγχει τον χώρο του ταμπλό. Η MainPanel εμφανίζει την κάρτα που της δόθηκε, ενώ κάποιο ενδεχόμενο βίντεο εμφανίζεται από μία ξεχωριστή κλάση που μπορεί να χειριστεί βίντεο, την VideoDialog. Έπειτα η

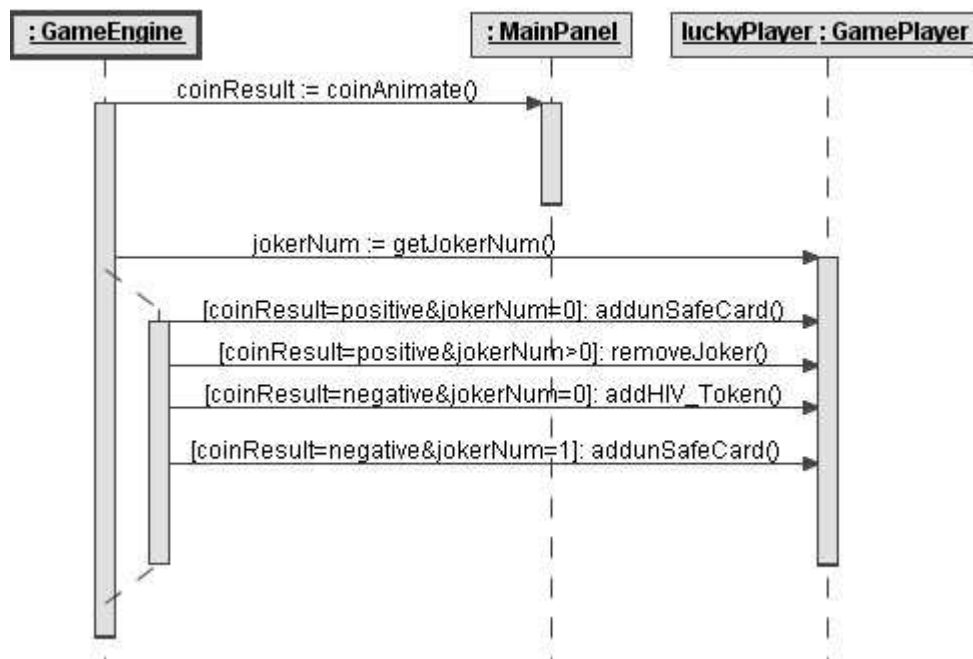
MainPanel περιμένει απόκριση από τον χρήστη. Η Εικόνα 5-19 μας δείχνει το διάγραμμα συνεργασίας για το τράβηγμα της κάρτας.



Εικόνα 5-19 - Διάγραμμα συνεργασίας τράβηγμα κάρτας

#### 5.1.4.7 Περίπτωση Χρήσης : Μη ασφαλής κάρτα κατάσταση

Για την περίπτωση χρήσης της επιλογής μη ασφαλούς κάρτας, έχει δοθεί διάγραμμα δραστηριοτήτων κατά τον ορισμό των απαιτήσεων και παρουσιάζεται στην Εικόνα 4-2. Επιπλέον η Εικόνα 5-20 μας παρουσιάζει το διάγραμμα ακολουθίας.

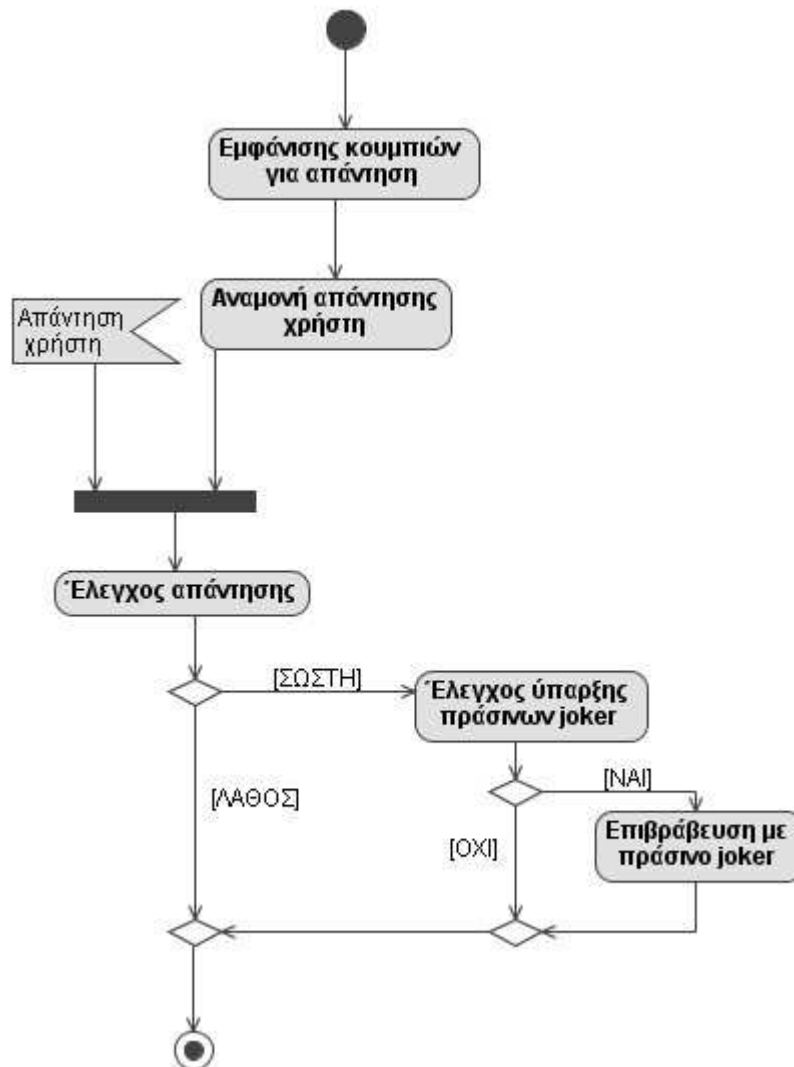


Εικόνα 5-20 - Διάγραμμα ακολουθίας για το τράβηγμα μη ασφαλούς κάρτας

Η κλάση GameEngine στέλνει μήνυμα στην MainPanel να παρουσιάσει την ρίψη του νομίσματος, στη συνέχεια στέλνει μήνυμα στο αντικείμενο luckyPlayer:GamePlayer για να ενημερωθεί ως προς τον αριθμό των joker που έχει αλλά και για να το ενημερώσει ανάλογα.

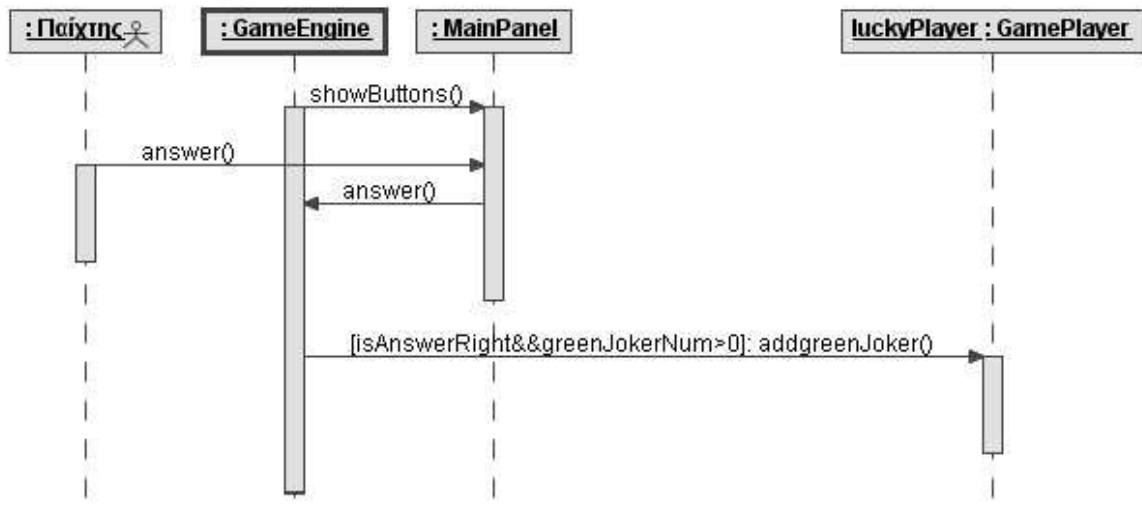
#### 5.1.4.8 Περίπτωση Χρήσης : Κάρτας ερώτησης

Όταν έχουμε κάρτα ερώτησης τότε ο χρήστης καλείται να απαντήσει. Σε ορθή απάντηση του επιβραβεύεται με πράσινο joker, όταν αυτός υπάρχει. Το διάγραμμα δραστηριοτήτων απεικονίζεται στην Εικόνα 5-21.



Εικόνα 5-21 - Διάγραμμα δραστηριοτήτων κάρτας ερώτησης

Ακολουθεί η Εικόνα 5-22 με το διάγραμμα ακολουθίας που δείχνει την ανταλλαγή μηνυμάτων μεταξύ των κλάσεων.



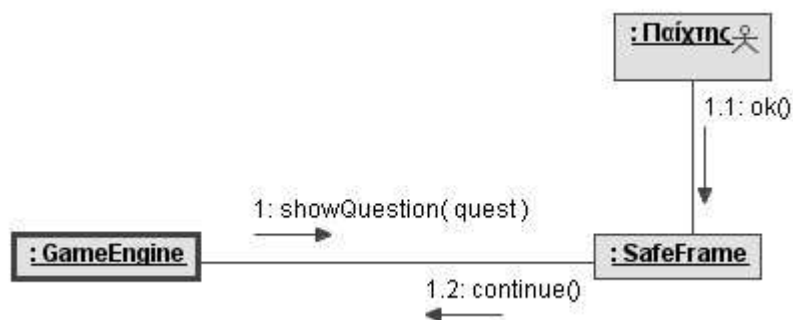
Εικόνα 5-22 - Διάγραμμα ακολουθίας κάρτας ερώτησης

#### 5.1.4.9 Περίπτωση Χρήσης : Λοιπές κάρτες κατάστασης και joker

Στις υπόλοιπες κάρτες κατάστασης δεν υπάρχει κάποια μεταβολή στις ιδιότητες του παίχτη, ενώ στην κάρτα joker ένας μπλε joker προστίθεται σε αυτούς που έχει ο παίχτης.

#### 5.1.4.10 Περίπτωση Χρήσης : Διάβασμα ερώτησης βιβλίου

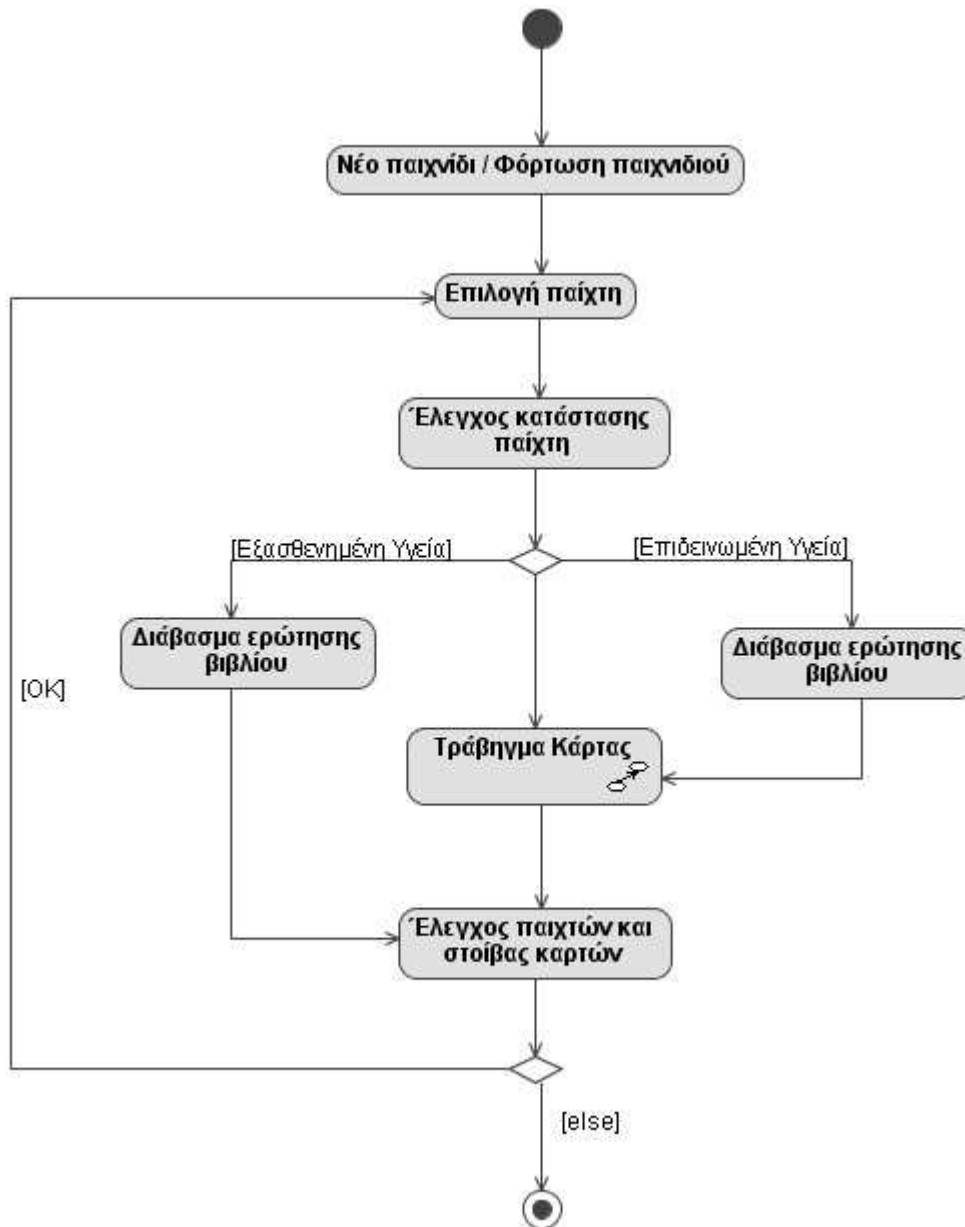
Το διάβασμα μιας ερώτησης βιβλίου συμβαίνει όταν ένας παίχτης έχει μπει σε δημόσιο πρόγραμμα εκπαίδευσης ( Αριθμός μη ασφαλών καρτών > Όριο μη ασφαλών καρτών) ή όταν η υγεία του έχει εξασθενήσει ( HIV Token > Όριο HIV Token). Κατά το διάβασμα μιας ερώτησης βιβλίου, το παιχνίδι διαλέγει τυχαία μία από την συλλογή που έχει και την εμφανίζει στην οθόνη. Η κλάση GameEngine κρατάει την συλλογή με τις ερωτήσεις βιβλίου και την τροφοδοτεί στην SafeFrame ώστε να την παρουσιάσει στην οθόνη. Στην Εικόνα 5-23 παρουσιάζεται το διάγραμμα συνεργασίας.



Εικόνα 5-23 - Διάγραμμα συνεργασίας ερώτησης βιβλίου

### 5.1.5 Συμπληρωματική ανάλυση στοιχείων

Όπως αναφέρθηκε η κλάση που ελέγχει την όλη ροή του παιχνιδιού, η GameEngine αποτελείται από έναν κύριο βρόχο στον οποίο εξελίσσεται το παιχνίδι. Η Εικόνα 5-24 μέσα από ένα διάγραμμα δραστηριότητας μας δείχνει τον τρόπο λειτουργίας του βρόχου.

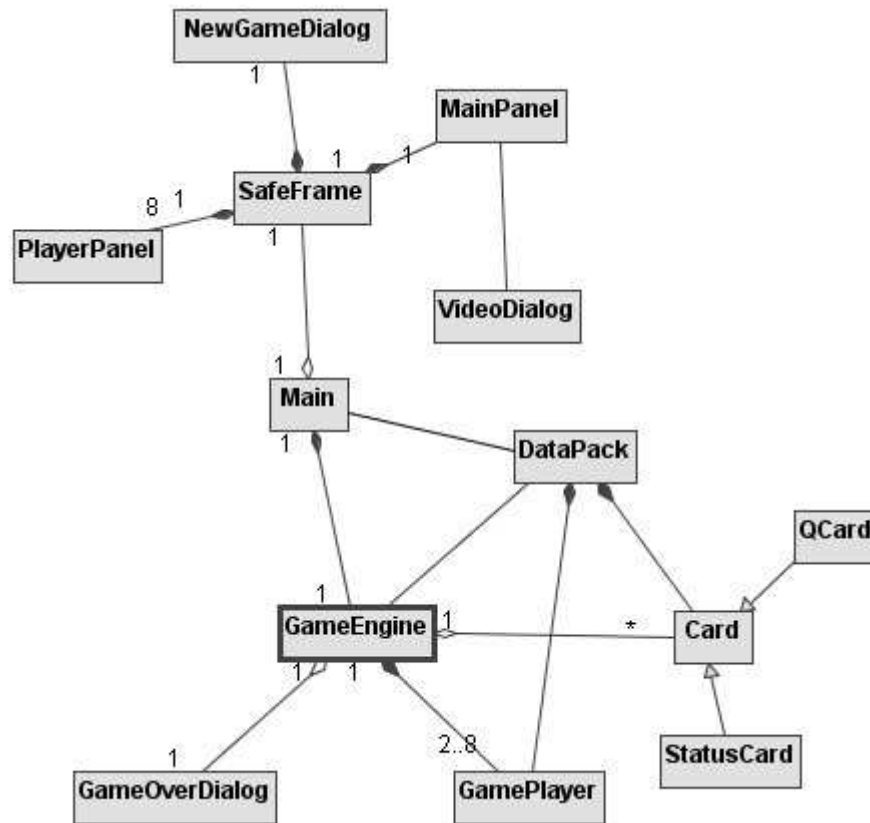


**Εικόνα 5-24 - Διάγραμμα δραστηριοτήτων βρόχου GameEngine**

Στο παραπάνω διάγραμμα η ομάδα δραστηριοτήτων 'Τράβηγμα Κάρτας' έχει παρουσιαστεί στην Εικόνα 5-18.

Τέλος η Εικόνα 5-25 μας παρουσιάζει ένα συνολικό διάγραμμα των κλάσεων που αναφέρθηκαν μέχρι τώρα.

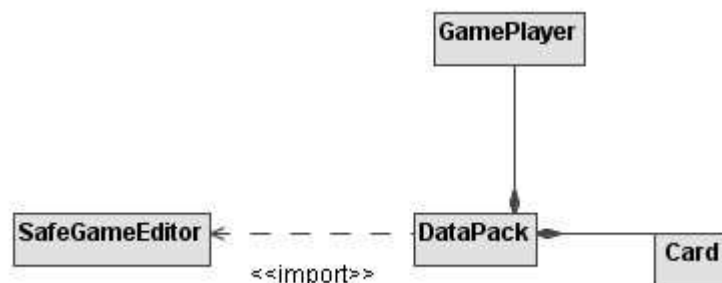




Εικόνα 5-25 - Διάγραμμα κλάσεων συνολικό

## 5.2 Ανάλυση του συντάκτη SafeGameEditor

Ο συντάκτης ρυθμίσεων σε σχέση με το παιχνίδι είναι πολύ πιο απλός στην λειτουργία του. Σκοπός του είναι να αποθηκεύσει όλες τις απαραίτητες παραμέτρους του παιχνιδιού σε ένα αρχείο ρυθμίσεων και όπως αναφέρθηκε και στην ανάλυση του SafeGame, οι ρυθμίσεις μπορούν να αποθηκευτούν υπό την μορφή αντικειμένου DataPack. Στην Εικόνα 5-26 βλέπουμε τις κλάσεις που απαρτίζουν τον συντάκτη.



Εικόνα 5-26 - Διάγραμμα κλάσεων SafeGameEditor

Η βασική κλάση για την λειτουργία του συντάκτη είναι η SafeGameEditor, ενώ γίνεται εισαγωγή της κλάσης DataPack από το πακέτο SafeGame. Οι κλάσεις

GamePlayer και Card χρησιμοποιούνται από την DataPack, όμως στην συγκεκριμένη εφαρμογή είναι χρήσιμη μόνο η Card για την αποθήκευση καρτών. Η GamePlayer θα πάρει την τιμή του κενού null.

Λόγω των απλών λειτουργιών του SafeGameEditor, δεν αναλύονται οι περιπτώσεις χρήσης του.

Στο επόμενο κεφάλαιο θα δούμε τον λεπτομερή σχεδιασμό του συστήματος.

## **6. Σχεδίαση**

Σκοπός της σχεδίασης είναι να προσδιορίσουμε με λεπτομέρεια πως θα υλοποιηθούν όσα αναφέρθηκαν στις απαιτήσεις και στην ανάλυση. Το μοντέλο που θα δημιουργηθεί θα είναι έτοιμο για να περάσει στη φάση της υλοποίησης με κώδικα. Αυτό σημαίνει πως θα ασχοληθούμε εκτενώς με τα γραφικά που είναι το κυρίαρχο στοιχείο, ενώ θα λάβουμε υπόψη την γλώσσα υλοποίησης, την Java και τις βιβλιοθήκες που την συνοδεύουν.

Όπως και στο προηγούμενο κεφάλαιο η σχεδίαση θα χωριστεί σε δύο κομμάτια, καθώς δύο είναι τα λογισμικά που παράγονται.

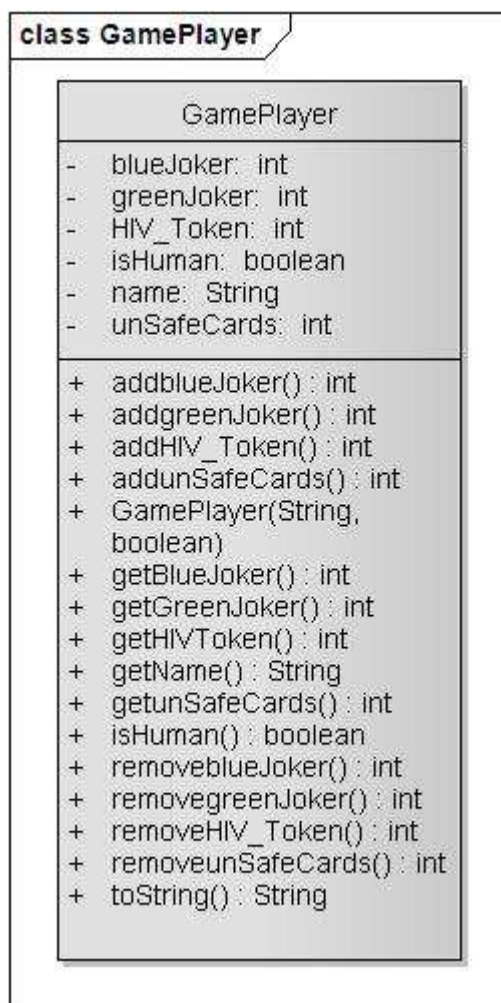
### **6.1 Σχεδίαση του παιχνιδιού**

Βασιζόμενοι στην ανάλυση που έγινε, θα δούμε τις κλάσεις του παιχνιδιού με λεπτομέρεια και έπειτα πως λειτουργούν στις περιπτώσεις χρήσης.

#### **6.1.1 Κλάσεις παιχτών, καρτών και δεδομένων**

##### **6.1.1.1 Κλάση `GamePlayer`**

Πρόκειται για την κλάση που αναπαριστά έναν παίχτη και τα χαρακτηριστικά του.



Εικόνα 6-1 - Κλάση GamePlayer

Πίνακας 6-1 - Ιδιότητες GamePlayer

Όνομα	Τύπος	Περιγραφή
<b>blueJoker</b>	int	Ο αριθμός των μπλε joker που έχει μαζέψει ο παίχτης.
<b>greenJoker</b>	int	Ο αριθμός των πράσινων joker που έχει μαζέψει ο παίχτης.
<b>HIV_Token</b>	int	Ο αριθμός των HIV Token που έχει μαζέψει ο παίχτης.
<b>isHuman</b>	boolean	Αν ο παίχτης είναι άνθρωπος ή Η/Υ, στην τιμή true θεωρούμε πως είναι άνθρωπος.
<b>name</b>	String	Το όνομα του παίχτη.
<b>unSafeCards</b>	int	Ο αριθμός των μη ασφαλών καρτών που έχει μαζέψει ο παίχτης.

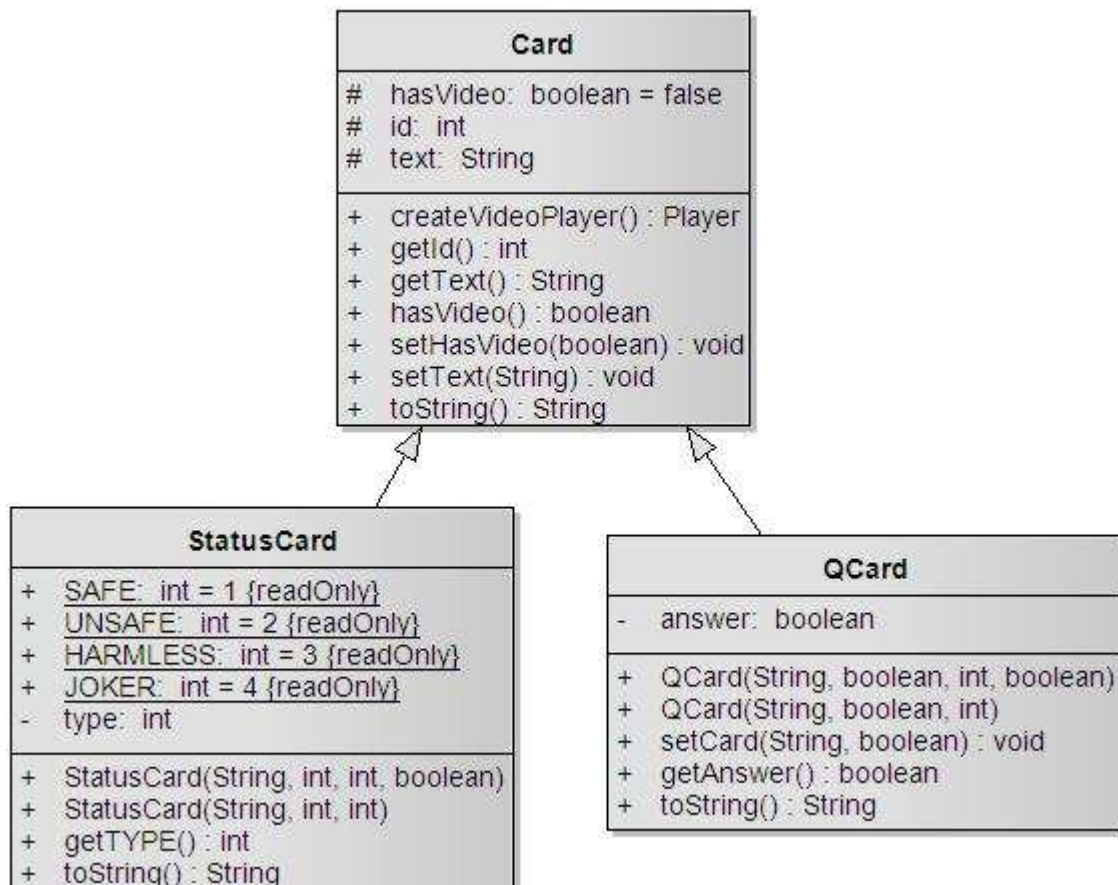
Πίνακας 6-2 - Μέθοδοι **GamePlayer**

<b>GamePlayer(String name, boolean isHuman)</b>	
Είσοδος	String name – το όνομα του παίχτη boolean isHuman – αν ο παίχτης είναι άνθρωπος ή όχι.
Έξοδος	-
Περιγραφή	Ο δομητής της κλάσης, αρχικοποιεί τον παίχτη με το όνομα και την φύση του , τα οποία δέχεται ως παραμέτρους, ενώ οι υπόλοιπες τιμές αρχικοποιούνται στην αριθμητική τιμή μηδέν.
<b>addblueJoker(), addgreenJoker(),addHIV_Token(), addunSafeCards()</b>	
Είσοδος	-
Έξοδος	int – η τιμή της κάθε ιδιότητας του παίχτη μετά την προσαύξηση.
Περιγραφή	Προσαυξάνει κατά ένα την ιδιότητα και επιστρέφει την τιμή της.
<b>getBlueJoker(), getGreenJoker(), getHIV_Token, getunSafeCards() ,</b>	
Είσοδος	-
Έξοδος	int
Περιγραφή	Επιστρέφει την τιμή της εκάστοτε ιδιότητας χωρίς καμία παρέμβαση.
<b>getName()</b>	
Είσοδος	-
Έξοδος	String
Περιγραφή	Επιστρέφει το όνομα του παίχτη.
<b>isHuman()</b>	
Είσοδος	-
Έξοδος	boolean
Περιγραφή	Επιστρέφει true όταν ο παίχτης είναι άνθρωπος, false όταν πρόκειται για Η/Υ.

<b>removeblueJoker(), removegreenJoker(), removeHIV_Token, removeunSafeCards()</b>	
Είσοδος	-
Έξοδος	int
Περιγραφή	Αφαιρεί μία μονάδα από την τιμή της κάθε ιδιότητας.
<b>toString()</b>	
Είσοδος	-
Έξοδος	String
Περιγραφή	Μέθοδος που χρησιμοποιείται για αποσφαλμάτωση

### 6.1.1.2 Κλάσεις Card, StatusCard και QCard

Η κλάση Card αναπαριστά μία κάρτα παιχνιδιού, ειδικότερα η StatusCard αντιστοιχεί στις κάρτες κατάστασης και τους μπλε joker ενώ η QCard στις κάρτες ερωτήσεων.



Εικόνα 6-2 - Κλάσεις Card, StatusCard και QCard

Πίνακας 6-3 - Ιδιότητες κλάσης Card

Όνομα	Τύπος	Περιγραφή
<b>hasVideo</b>	boolean	Μας λέει αν η κάρτα συνδέεται με κάποιο βίντεο. Αρχική του τιμή είναι false.
<b>id</b>	int	Χαρακτηρίζει μοναδικά την κάρτα, εκτός από τους joker.
<b>text</b>	String	Περιέχει το κείμενο της κάρτας, μπορεί να είναι είτε κάποια ερώτηση είτε το κείμενο μιας κάρτας κατάστασης.

Πίνακας 6-4 – Μέθοδοι κλάσης Card

<b>getId()</b>	
Είσοδος	-
Έξοδος	int
Περιγραφή	Επιστρέφει το Id της κάρτας.
<b>getText()</b>	
Είσοδος	-
Έξοδος	String
Περιγραφή	Επιστρέφει το κείμενο της κάρτας.
<b>hasVideo()</b>	
Είσοδος	-
Έξοδος	boolean
Περιγραφή	Επιστρέφει true, όταν υπάρχει βίντεο.
<b>setHasVideo(boolean hasVideo)</b>	
Είσοδος	boolean hasVideo – η τιμή του είναι ανάλογη με το αν υπάρχει βίντεο ή όχι.
Έξοδος	-
Περιγραφή	Ορίζει αν υπάρχει βίντεο.
<b>createVideoPlayer()</b>	
Είσοδος	-
Έξοδος	Player – Το αντικείμενο που επιστρέφεται ανήκει στην κλάση Player, που χρησιμοποιείται για αναπαραγωγή βίντεο. Την κλάση θα την δούμε παρακάτω.

Περιγραφή	Δημιουργεί ένα αντικείμενο για την αναπαραγωγή του βίντεο, σε περίπτωση που το αρχείο του βίντεο παρουσιάζει κάποιο πρόβλημα, τότε επιστρέφει την τιμή null.
<b>toString()</b>	
Είσοδος	-
Έξοδος	String
Περιγραφή	Μέθοδος για αποσφαλμάτωση.

Πίνακας 6-5 - Ιδιότητες κλάσης StatusCard

Όνομα	Τύπος	Περιγραφή
type	int	Αντιπροσωπεύει τον τύπο της κάρτας, οι τιμές του είναι από 1 έως 4.
SAFE	int	Έχει την τιμή 1 και αντιπροσωπεύει τις ασφαλείς κάρτες.
UNSAFE	int	Έχει την τιμή 2 και αντιπροσωπεύει τις μη ασφαλείς κάρτες.
HARMLESS	int	Έχει την τιμή 3 και αντιπροσωπεύει τις μη επικίνδυνες κάρτες.
JOKER	int	Έχει την τιμή 4 και αντιπροσωπεύει τους joker.

Πίνακας 6-6 - Μέθοδοι κλάσης StatusCard

<b>StatusCard(String text, int TYPE, int id)</b>	
Είσοδος	String text – το κείμενο της κάρτας int TYPE – ο τύπος της κάρτας με τιμές 1 - 4 int id – το Id της κάρτας
Έξοδος	-
Περιγραφή	Ο δομητής της κλάσης .



<b>StatusCard(String text, int TYPE, int id, boolean hasVideo)</b>	
Είσοδος	Όλες οι παραπάνω παράμετροι και επιπλέον boolean hasVideo – αν έχει βίντεο.
Έξοδος	-
Περιγραφή	Ο δομητής της κλάσης .
<b>getType()</b>	
Είσοδος	-
Έξοδος	int - Με τιμές από 1 – 4.
Περιγραφή	Επιστρέφει τον τύπο της κάρτας.
<b>toString()</b>	
Είσοδος	-
Έξοδος	String
Περιγραφή	Μέθοδος αποσφαλμάτωσης.

Πίνακας 6-7 - Ιδιότητες κλάσης QCard

Όνομα	Τύπος	Περιγραφή
answer	boolean	Περιέχει την απάντηση της ερώτησης στην οποία αντιστοιχεί η κάρτα, η τιμή true ανήκει στην σωστή απάντηση ενώ η false στην λανθασμένη.

Πίνακας 6-8 - Μέθοδοι κλάσης QCard

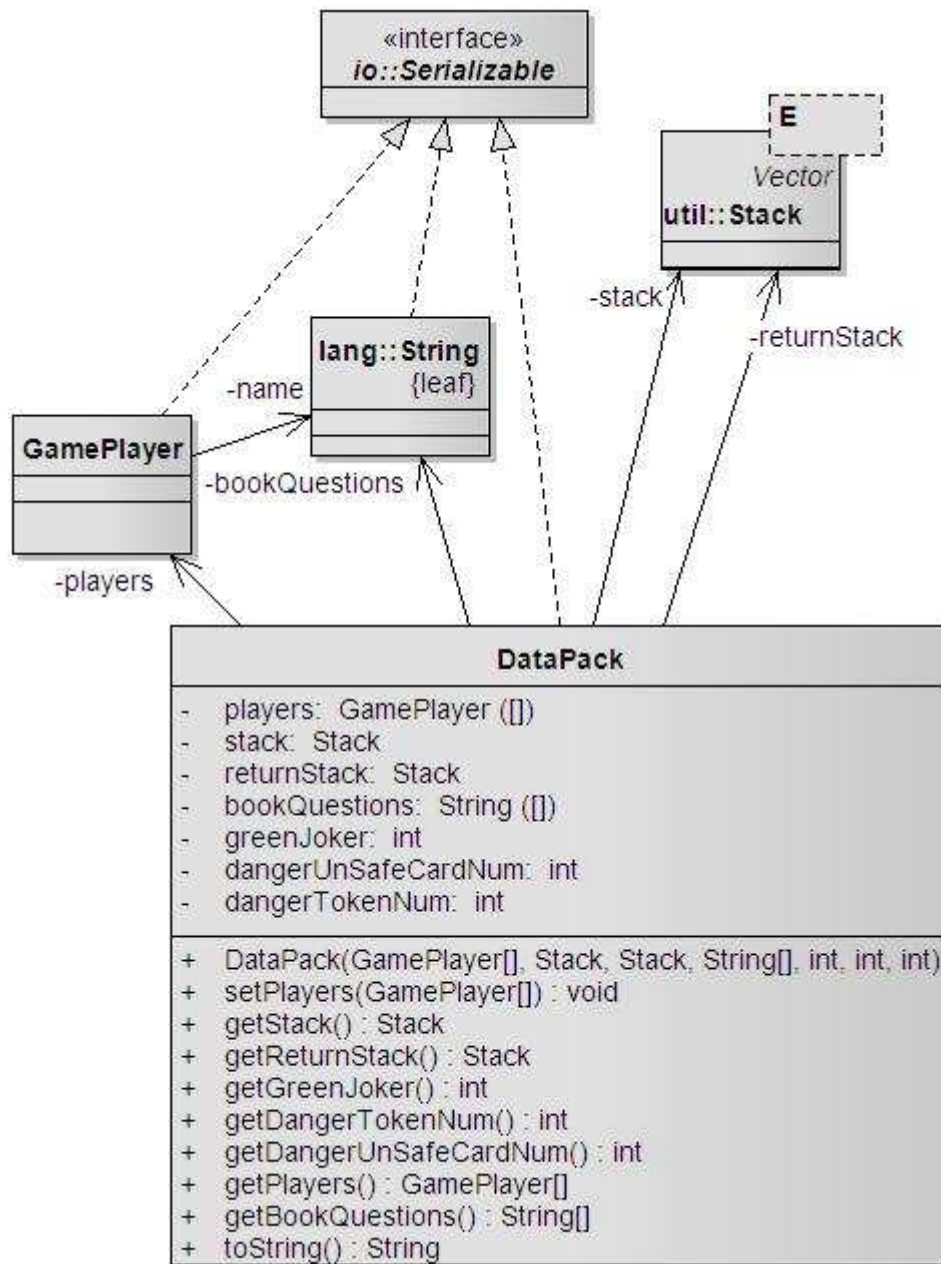
<b>QCard(String text, boolean answer, int id)</b>	
Είσοδος	String text – το κείμενο της ερώτησης boolean answer – την απάντηση στην ερώτηση int id – το Id της ερώτησης
Έξοδος	-
Περιγραφή	Ο δομητής της κλάσης.

<b>QCard(String text, boolean answer, int id, boolean hasVideo)</b>	
Είσοδος	Όλες οι παράμετροι του προηγούμενου δομητή και επιπλέον boolean hasVideo – το αν υπάρχει βίντεο ή όχι.
Έξοδος	-
Περιγραφή	Ο δομητής της κλάσης.
<b>setCard(String text, boolean answer)</b>	
Είσοδος	String text – κείμενο που αντιστοιχεί στην ερώτηση boolean answer – η απάντηση στην ερώτηση
Έξοδος	-
Περιγραφή	Ορίζει το περιεχόμενο της κάρτας ερώτησης.
<b>getAnswer()</b>	
Είσοδος	-
Έξοδος	boolean – η απάντηση στην ερώτηση
Περιγραφή	Επιστρέφει την απάντηση στην ερώτηση της κάρτας.
<b>toString()</b>	
Είσοδος	-
Έξοδος	String
Περιγραφή	Μέθοδος αποσφαλμάτωσης.

### 6.1.1.3 Κλάση DataPack

Όπως αναφέρθηκε στην ανάλυση η κλάση DataPack θα χρησιμοποιηθεί για να αποθηκεύσει τα δεδομένα του παιχνιδιού σε κάποιο αρχείο στον δίσκο. Για την αποθήκευση, θα επιλέξουμε να αποθηκευτεί το αντικείμενο ολόκληρο με τα δεδομένα σε δυαδική μορφή. Για να γίνει κάτι τέτοιο στην γλώσσα Java πρέπει η κλάση DataPack να υλοποιήσει την διεπαφή (interface) *java.io.Serializable*. Εκτός από την κλάση DataPack, πρέπει να υλοποιούν την διεπαφή και όλα τα συστατικά της κλάσης, Η Εικόνα 6-3 δείχνει την κλάση και τις σχέσεις της. Η DataPack περιέχει ένα πίνακα από αντικείμενα GamePlayer που αντιστοιχούν στους παίκτες και δύο αντικείμενα Stack, πρόκειται την κλάση της *java.util.Stack* που υλοποιεί την δομή στοίβα, στις δύο στοίβες αποθηκεύονται οι κάρτες του παιχνιδιού (κλάση Card). Όπως βλέπουμε η κλάση GamePlayer υλοποιεί την διεπαφή Serializable,

το ίδιο γίνεται και στην υπερκλάση της Stack την Vector, όπως και επίσης πρέπει να γίνει και στην Card που αποθηκεύεται στα αντικείμενα της Stack.



Εικόνα 6-3 - Κλάση DataPack

Πίνακας 6-9 - Ιδιότητες της κλάσης DataPack

Όνομα	Τύπος	Περιγραφή
players	GamePlayer []	Ένας πίνακας με τα αντικείμενα των παιχτών.
stack	Stack	Δομή στοίβας που θα περιέχει κάρτες.

Όνομα	Τύπος	Περιγραφή
returnStack	Stack	Δομή στοίβας που θα περιέχει κάρτες (στοίβα επιστροφής).
bookQuestions	String []	Πίνακας με ερωτήσεις βιβλίου.
greenJoker	int	Ο αριθμός των πράσινων joker
dangerUnSafeCardNum	int	Το όριο των μη ασφαλών καρτών
dangerTokenNum	int	Το όριο των HIV Token

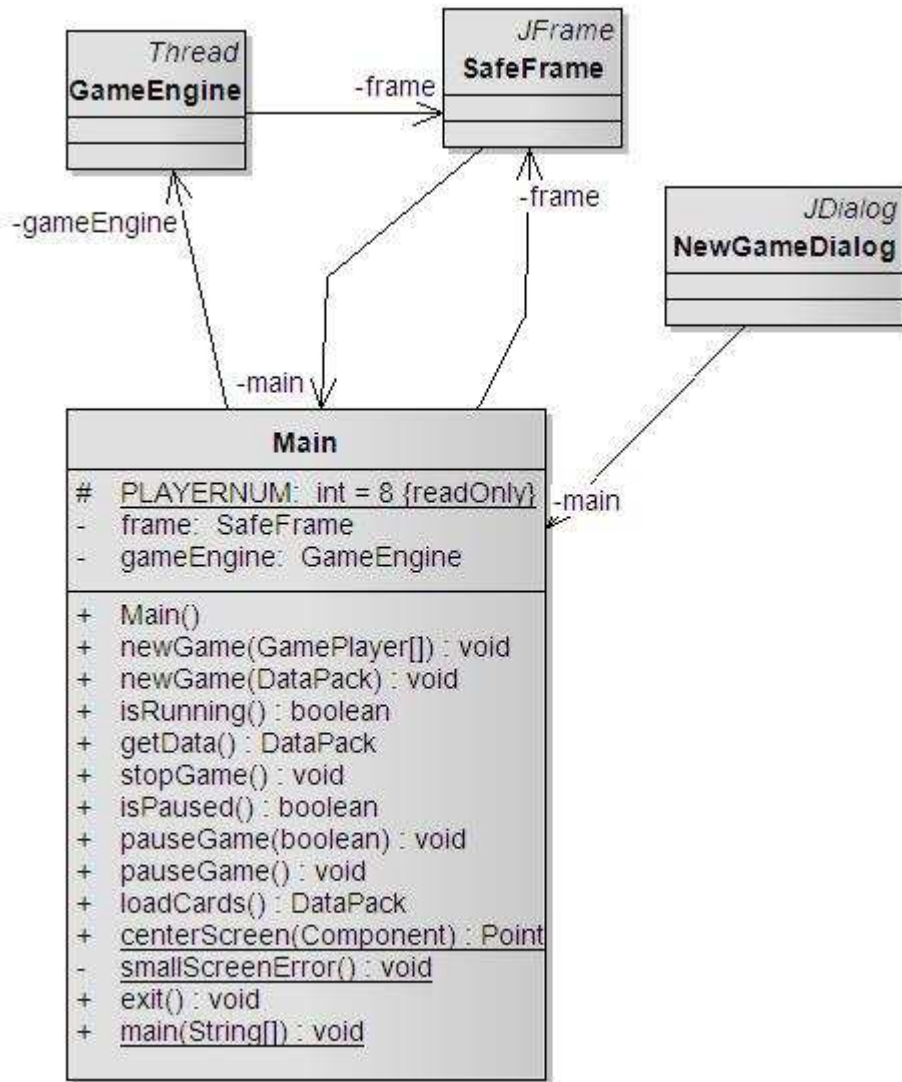
Πίνακας 6-10 - Μέθοδοι της κλάσης DataPack

<b>DataPack(GamePlayer[] players, Stack stack, Stack returnStack, String[] bookQuestions, int greenJoker, int dangerUnSafeCardNum, int dangerTokenNum)</b>	
Είσοδος	GamePlayer [] players, Stack stack , Stack returnStack ,String [] bookQuestions , int greenJoker, int dangerUnSafeCardNum, int dangerTokenNum Πρόκειται για ανάλογες παραμέτρους με τις ιδιότητες της κλάσης για την αρχικοποίηση του αντικειμένου.
Έξοδος	-
Περιγραφή	Δομητής της κλάσης που αρχικοποιεί τις ιδιότητές του ανάλογα με τις παραμέτρους που δέχεται.
<b>setPlayers(GamePlayer[] players)</b>	
Είσοδος	GamePlayer[] players – ο πίνακας των παιχτών του παιχνιδιού
Έξοδος	-
Περιγραφή	Θέτει στον πίνακα των παιχτών της κλάσης την παράμετρο που δέχεται.
<b>getStack()</b>	
Είσοδος	-
Έξοδος	Stack – κύρια στοίβα του παιχνιδιού
Περιγραφή	Επιστρέφει την κύρια στοίβα του παιχνιδιού.

<b>getReturnStack()</b>	
Είσοδος	
Έξοδος	Stack – στοίβα επιστροφής του παιχνιδιού.
Περιγραφή	Επιστρέφει την στοίβα επιστροφής του παιχνιδιού.
<b>getGreenJoker(), getDangerUnsafeCardNum(), getDangerTokenNum()</b>	
Είσοδος	-
Έξοδος	int
Περιγραφή	Επιστρέφει την τιμή της ιδιότητας στην οποία αναφέρεται.
<b>getPlayers()</b>	
Είσοδος	-
Έξοδος	GamePlayer [] – ο πίνακας των παιχτών
Περιγραφή	Επιστρέφει αναφορά στον πίνακα των παιχτών.
<b>getBookQuestions()</b>	
Είσοδος	-
Έξοδος	String [] – ο πίνακας με τις ερωτήσεις βιβλίου
Περιγραφή	Επιστρέφει αναφορά στον πίνακα με τις ερωτήσεις βιβλίου.

### 6.1.2 Κύρια κλάση του παιχνιδιού

Η κύρια κλάση του παιχνιδιού με το όνομα Main περιέχει τα βασικά συστατικά του, μία κλάση που αντιπροσωπεύει το βασικό παράθυρο την SafeFrame και την κλάση που παίζει το ρόλο της μηχανής του παιχνιδιού, την GameEngine.



Εικόνα 6-4 - Κλάση Main

Πίνακας 6-11 - Ιδιότητες της Main

Όνομα	Τύπος	Περιγραφή
PLAYERNUM	int	Ορίζει τον αριθμό των παιχτών του παιχνιδιού, πρόκειται για μία ιδιότητα που η τιμή της δεν αλλάζει.
frame	SafeFrame	Ανήκει στην κλάση SafeFrame και αντιστοιχεί στο κύριο παράθυρο της εφαρμογής.
gameEngine	GameEngine	Ανήκει στην κλάση GameEngine και αντιστοιχεί στην μηχανή του παιχνιδιού.

**Πίνακας 6-12 - Μέθοδοι της Main**

<b>Main()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Πρόκειται για τον δομητή της κλάσης, εδώ αρχικοποιείται το κεντρικό παράθυρο frame και εμφανίζεται.
<b>newGame(GamePlayer[] pl)</b>	
Είσοδος	GamePlayer[] pl – πίνακας παιχτών του παιχνιδιού απαραίτητος για την αρχικοποίηση νέου
Έξοδος	-
Περιγραφή	Δημιουργεί ένα νέο παιχνίδι σύμφωνα με τον πίνακα των παιχτών που δίνεται ως παράμετρος, εδώ δημιουργείται η μηχανή του παιχνιδιού gameEngine και φορτώνονται οι παράμετροι από το αρχείο των ρυθμίσεων.
<b>newGame(DataPack dataPack)</b>	
Είσοδος	DataPack dataPack – αντικείμενο που περιέχει όλες τις πληροφορίες του παιχνιδιού
Έξοδος	-
Περιγραφή	Χρησιμοποιείται για να συνεχίσουμε ένα παιχνίδι που φορτώσαμε, το dataPack παρέχει όχι μόνο τις ρυθμίσεις αλλά και τους παίχτες που έπαιζαν .
<b>isRunning()</b>	
Είσοδος	-
Έξοδος	boolean
Περιγραφή	Επιστρέφει αν υπάρχει παιχνίδι που παίζει εκείνη την στιγμή ή όχι. Ουσιαστικά ελέγχει αν υπάρχει κάποια gameEngine και την ρωτάει αν παίζει.

<b>getData()</b>	
Είσοδος	-
Έξοδος	DataPack
Περιγραφή	Επιστρέφει αντικείμενο με όλα τα περιεχόμενα του παιχνιδιού, αν δεν παίζει κάποιο παιχνίδι εκείνη τη στιγμή επιστρέφει την τιμή null.
<b>stopGame()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Αν εκτελείται κάποιο παιχνίδι εκείνη τη στιγμή το σταματάει.
<b>isPaused()</b>	
Είσοδος	-
Έξοδος	boolean
Περιγραφή	Μας λέει αν έχει γίνει παύση του παιχνιδιού, αν δεν εκτελείται παιχνίδι επιστρέφει false.
<b>pauseGame(boolean pause)</b>	
Είσοδος	boolean pause – λέει στο παιχνίδι να κάνει παύση ή όχι.
Έξοδος	-
Περιγραφή	Κάνει παύση ή συνέχιση του παιχνιδιού ανάλογα με την παράμετρο που δέχεται.
<b>resumeGame()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Αν εκτελείται κάποιο παιχνίδι εκείνη τη στιγμή κάνει παύση, αν όμως υπάρχει παιχνίδι που είναι ήδη σε κατάσταση παύσης, προκαλεί την συνέχισή του.
<b>loadCards()</b>	
Είσοδος	-
Έξοδος	DataPack
Περιγραφή	Φορτώνει το αρχείο ρυθμίσεων που έχει οριστεί.

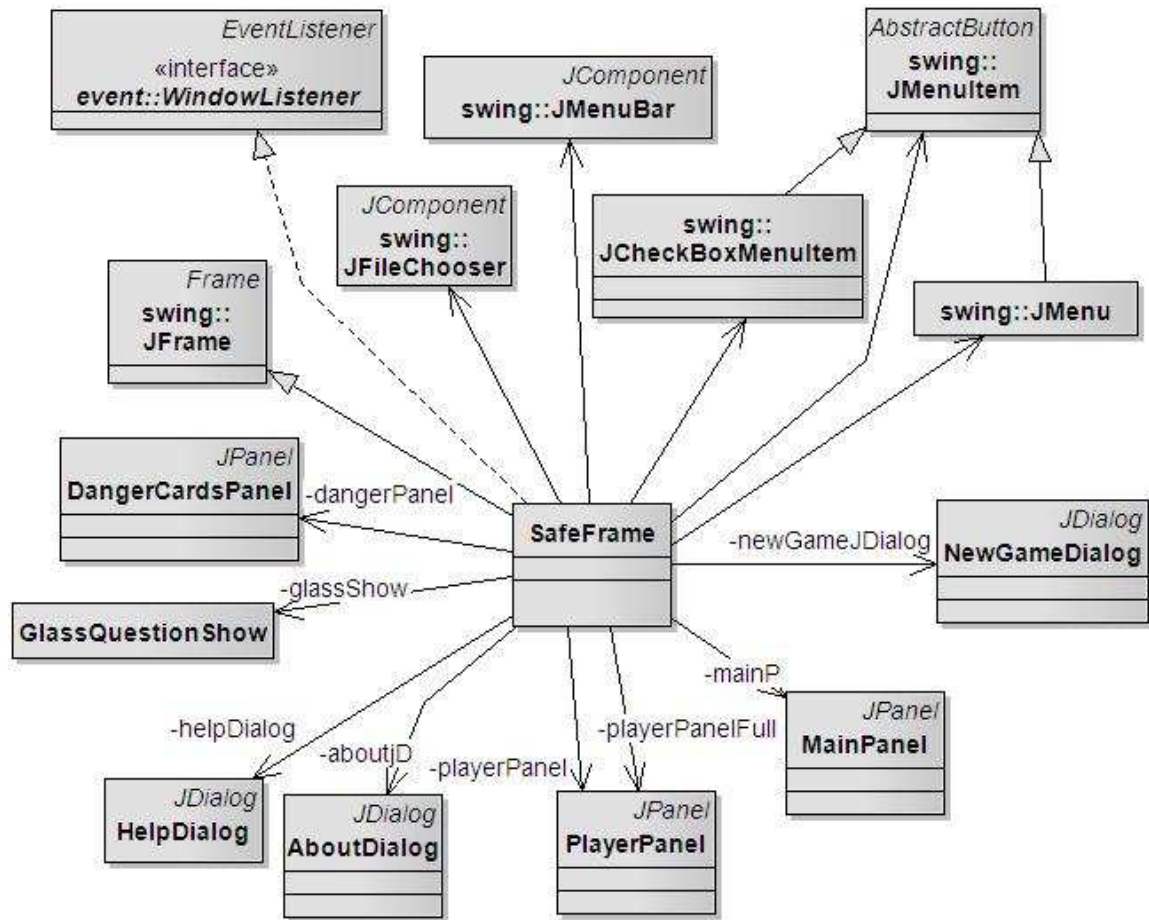


<b>centerScreen(Component comp)</b>	
Είσοδος	java.awt.Component – αντικείμενο που έχει γραφική αναπαράσταση στην οθόνη, ανήκει στις βιβλιοθήκες της java.
Έξοδος	java.awt.Point – Αντικείμενο που αναπαριστά ένα σημείο στην οθόνη.
Περιγραφή	Βοηθητική μέθοδος για το κεντράρισμα αντικειμένων στην οθόνη.
<b>smallScreenError()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Βοηθητική μέθοδος που προκαλεί έξοδο με το ανάλογο μήνυμα στην οθόνη, όταν καλεστεί. Την καλούμε όταν η ανάλυση της οθόνης είναι πολύ μικρή.
<b>exit()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Προκαλεί έξοδο από το πρόγραμμα.
<b>main(String[] args)</b>	
Είσοδος	String[] args
Έξοδος	-
Περιγραφή	Η μέθοδος που καλείται πρώτη για την εκτέλεση, κάνει έναν έλεγχο για την ανάλυση της οθόνης (θεωρούμε ελάχιστη ανάλυση 800x600) και έπειτα δημιουργεί το αντικείμενο Main.

### 6.1.3 Παραθυρικές κλάσεις

Στην Java μπορούμε να χειριστούμε παράθυρα και διαλόγους με τη βοήθεια του swing, μιας συλλογής κλάσεων που βοηθά στη δημιουργία γραφικών διεπαφών και εμπλουτίζει τον βαθμό αλληλεπίδρασης της εφαρμογής με τον χρήστη. Στην

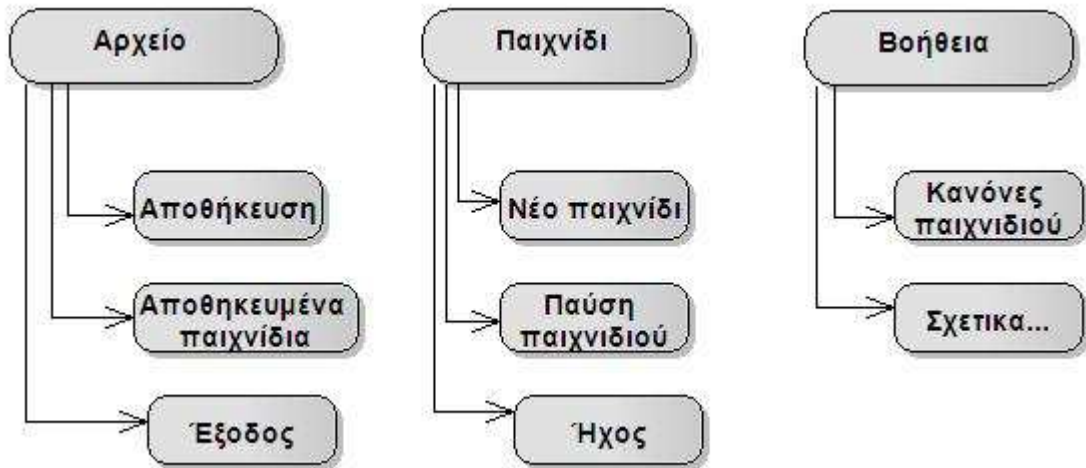
Εικόνα 6-5 βλέπουμε το διάγραμμα των κλάσεων που θα μελετήσουμε παρακάτω, το διάγραμμα βασίζεται στην κλάση του πρωτεύοντος παραθύρου την SafeFrame.



Εικόνα 6-5 - Παραθυρικές κλάσεις

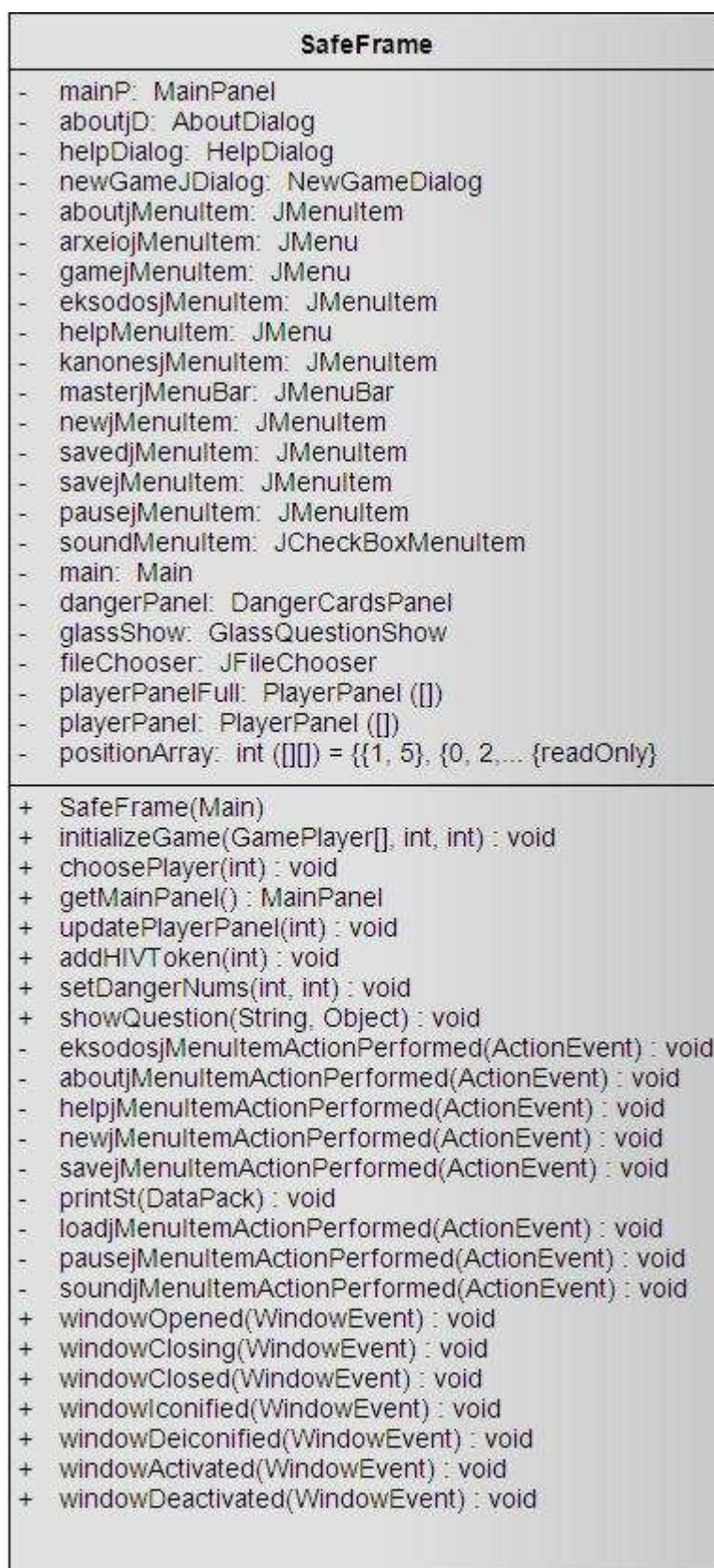
### 6.1.3.1 Κλάση SafeFrame

Το πρώτο και κύριο παράθυρο που αντικρίζουμε είναι από την SafeFrame, η οποία κληρονομεί την javax.swing.JFrame και υλοποιεί την java.awt.event.WindowListener. Η JFrame δημιουργεί ένα παράθυρο και η WindowListener είναι μία διεπαφή για το κάθε συμβάν που έχει να κάνει με το παράθυρο, όπως το κλείσιμο του παραθύρου. Η SafeFrame έχει το μενού της εφαρμογής το οποίο υλοποιείται από τις κλάσεις JMenu, JMenuBar, JMenuItem, JCheckBoxMenuItem του πακέτου javax.swing. Το μενού φαίνεται στην Εικόνα 6-6. Η επιλογή «Ήχος» ενεργοποιεί τον ήχο ή όχι, οπότε έχει και το ανάλογο σημάδι ενεργοποίησης.



Εικόνα 6-6 - Μενού παιχνιδιού

Οι κλάσεις MainPanel, PlayerPanel και DangerCardsPanel κληρονομούν την JPanel, κλάση της Java και αντιστοιχούν σε στοιχεία της κύριας οθόνης. Η MainPanel σχετίζεται με το ταμπλό του παιχνιδιού, η PlayerPanel σε κάθε παίχτη ξεχωριστά και η DangerCardsPanel στο σημείο της οθόνης που πληροφορεί για τις τιμές των ορίων των HIV Token και των μη ασφαλών καρτών. Οι κλάσεις NewGameDialog, HelpDialog, AboutDialog κληρονομούν την JDialog, κλάση της Java και εμφανίζονται με την ανάλογη επιλογή του μενού. Η κλάση GlassQuestionShow χρησιμοποιείται για την εμφάνιση των ερωτήσεων βιβλίου, καλύπτοντας όλη την επιφάνεια της SafeFrame.



**Εικόνα 6-7 - Κλάση SafeFrame**

Πίνακας 6-13 - Ιδιότητες της SafeFrame

Όνομα	Τύπος	Περιγραφή
masterJMenuBar	JMenuBar	Η κύρια μπάρα του μενού.
arxeioJMenuItem	JMenuItem	Το μενού «Αρχείο».
gameJMenuItem	JMenuItem	Το μενού «Παιχνίδι».
helpMenuItem	JMenuItem	Το μενού «Βοήθεια».
aboutJMenuItem	JMenuItem	Το στοιχείο του μενού «Σχετικά ...».
eksodosJMenuItem	JMenuItem	Το στοιχείο του μενού «Έξοδος».
kanonesJMenuItem	JMenuItem	Το στοιχείο του μενού «Κανόνες παιχνιδιού».
newJMenuItem	JMenuItem	Το στοιχείο του μενού «Νέο παιχνίδι».
savedJMenuItem	JMenuItem	Το στοιχείο του μενού «Αποθηκευμένα παιχνίδια».
saveJMenuItem	JMenuItem	Το στοιχείο του μενού «Αποθήκευση».
pauseJMenuItem	JMenuItem	Το στοιχείο του μενού «Παύση παιχνιδιού».
soundMenuItem	JCheckBoxMenuItem	Το στοιχείο του μενού «Ήχος».
fileChooser	JFileChooser	Βοηθητικό παράθυρο για την επιλογή αρχείων.
newGameJDialog	NewGameDialog	Παράθυρο διαλόγου για τη δημιουργία νέου παιχνιδιού.
aboutJD	AboutDialog	Παράθυρο διαλόγου για ενημέρωση σχετικά με το πρόγραμμα.
helpDialog	HelpDialog	Παράθυρο διαλόγου για την εμφάνιση της βοήθειας
mainP	MainPanel	Ο χώρος του ταμπλό του παιχνιδιού
dangerPanel	DangerCardsPanel	Ο χώρος που αναγράφονται οι παράμετροι του παιχνιδιού (όρια).
playerPanelFull	PlayerPanel[]	Ο χώρος που αντιστοιχεί στον κάθε παίκτη, ο πίνακας των αντικειμένων αντιστοιχεί σε όλους τους παίκτες.

Όνομα	Τύπος	Περιγραφή
playerPanel	PlayerPanel[]	Ο χώρος που αντιστοιχεί στον κάθε παίχτη, ο πίνακας των αντικειμένων αντιστοιχεί στους παίχτες που είναι ενεργοί, επομένως δημιουργείται αφού ξεκινήσει νέο παιχνίδι.
glassShow	GlassQuestionShow	Κλάση που εμφανίζει τις ερωτήσεις βιβλίου, σκοπός της είναι να καλύψει όλα τα υπόλοιπα αντικείμενα όταν προβάλει μία ερώτηση βιβλίου.
main	Main	Η κύρια κλάση του παιχνιδιού.
positionArray	int [] []	Πίνακας ακεραίων που παρουσιάζει μία αντιστοιχία αριθμού παιχτών και θέσεων γύρω από το ταμπλό.

Πίνακας 6-14 - Μέθοδοι της SafeFrame

<b>SafeFrame(Main main)</b>	
Είσοδος	Main main – η κύρια κλάση του παιχνιδιού
Έξοδος	-
Περιγραφή	Ο δομητής της κλάσης, αρχικοποιεί τις ιδιότητες της κλάσης, όπως τα μενού, τα πάνελ και τα παράθυρα διαλόγου και τα τοποθετεί, όσα είναι ορατά, στην ανάλογη θέση με την βοήθεια των layout managers.

<b>initializeGame(GamePlayer players[],int dangerUnsafeNum,int dangerTokenNum)</b>	
Είσοδος	GamePlayer players[] – ο πίνακας παιχτών που θα παίξουν. dangerUnsafeNum – το όριο των μη ασφαλών καρτών. dangerTokenNum - το όριο των HIV Token.
Έξοδος	-
Περιγραφή	Αρχικοποιεί κάθε απαραίτητο συστατικό ώστε να ξεκινήσει ένα νέο παιχνίδι ή να συνεχιστεί κάποιος που φορτώθηκε από τον δίσκο. Δίνει στον πίνακα playerPanel τους παίχτες που θα παίξουν και στο dangerPanel τα όρια που δίνονται ως παράμετροι.
<b>choosePlayer(int luckyPI)</b>	
Είσοδος	int luckyPI – ο τυχερός παίχτης που παίζει.
Έξοδος	-
Περιγραφή	Κάνει μία κυκλική οπτικά εναλλαγή μεταξύ των πάνελ των παιχτών, άρα και των ίδιων των παιχτών και καταλήγει στον τυχερό παίχτη που δίνεται ως παράμετρος.
<b>MainPanel getMainPanel()</b>	
Είσοδος	-
Έξοδος	MainPanel – το τρέχον βασικό πάνελ
Περιγραφή	Επιστρέφει το mainPanel
<b>updatePlayerPanel(int pl)</b>	
Είσοδος	int pl – αριθμός που ανήκει στο πάνελ του παίχτη που ενημερώνεται.
Έξοδος	-
Περιγραφή	Δίνει εντολή προς γραφική ενημέρωση του πάνελ που ανήκει στον παίχτη που δίνεται ως παράμετρος.

<b>addHIVToken(int pl)</b>	
Είσοδος	int pl – αριθμός που ανήκει στον παίχτη που παίζει.
Έξοδος	-
Περιγραφή	Προσθέτει ένα HIV Token στο πάνελ του παίχτη που παίζει και κατ' επέκταση και στον παίχτη.
<b>setDangerNums(int dangerTokenNum, int dangerUnsafeCardNum)</b>	
Είσοδος	int dangerTokenNum – το όριο των HIV Token int dangerUnsafeCardNum - το όριο των μη ασφαλών καρτών.
Έξοδος	-
Περιγραφή	Ενημερώνει κάθε πάνελ ενεργού παίχτη για τα όρια του παιχνιδιού, έτσι ώστε όταν ξεπεραστούν να μπορεί να φανεί στο πάνελ.
<b>showQuestion(String quest, Object locker)</b>	
Είσοδος	String quest – το κείμενο της ερώτησης βιβλίου που θα εμφανιστεί. Object locker – το αντικείμενο που θα χρησιμοποιηθεί ως κλειδαριά για να μην συνεχίσει το παιχνίδι, το οποίο είναι η GameEngine που παίζει εκείνη τη στιγμή.
Έξοδος	-
Περιγραφή	Προκαλεί την εμφάνιση της ερώτησης βιβλίου στο αντικείμενο glassShow της κλάσης GlassQuestionShow, δέχεται ως παράμετρο το κείμενο της ερώτησης βιβλίου και ένα αντικείμενο που θα χρησιμοποιηθεί ως κλειδαριά για παύση και συνέχιση.
<b>eksodosjMenuItemActionPerformed(ActionEvent evt)</b>	
Είσοδος	ActionEvent evt – συμβάν που προκαλείται όταν κάποιος επιλέξει το μενού με το οποίο συνδέεται.
Έξοδος	-
Περιγραφή	Προκαλεί έξοδο από το παιχνίδι.

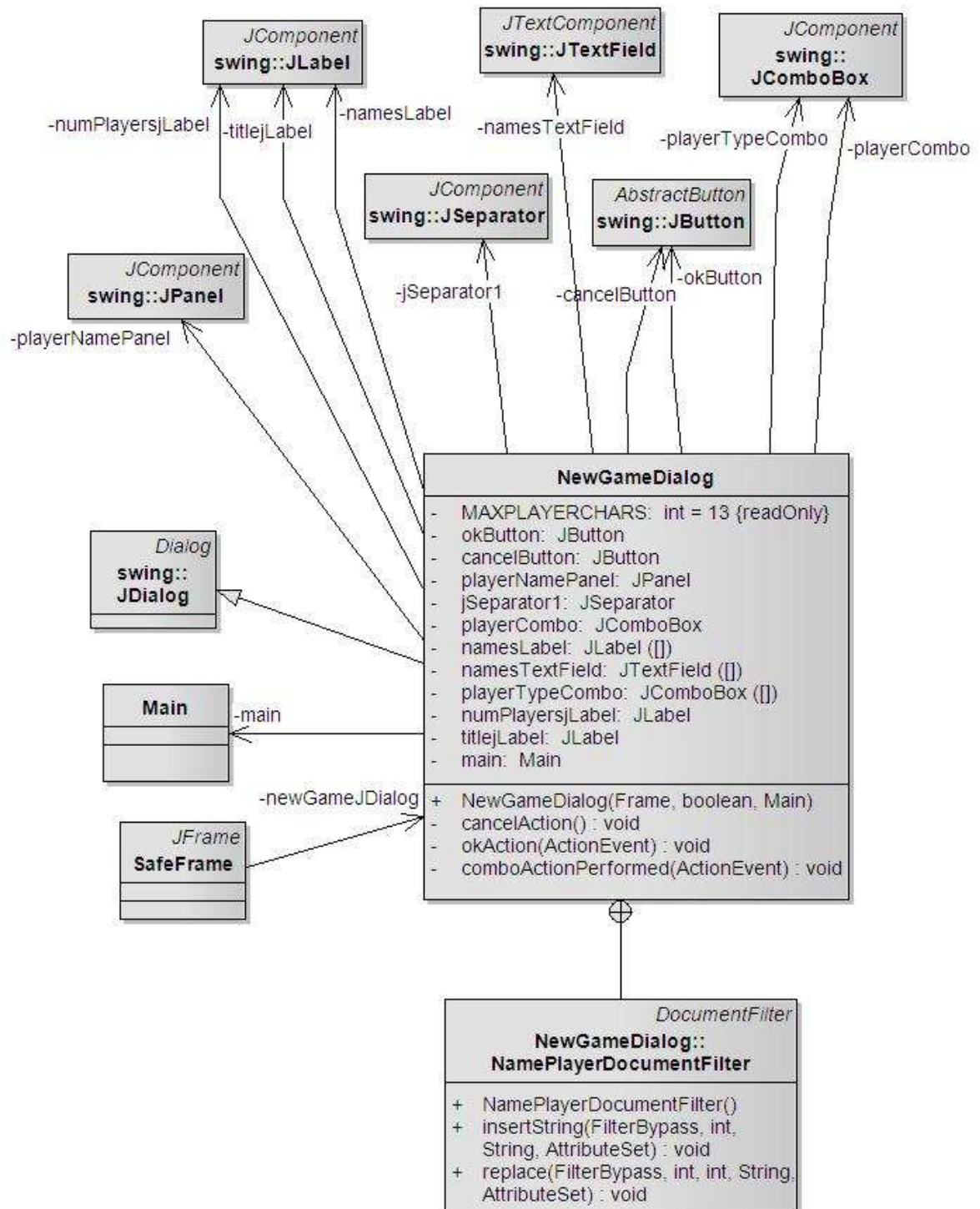


<b>aboutJMenuItemActionPerformed(ActionEvent evt)</b>	
Είσοδος	ActionEvent evt – συμβάν που προκαλείται όταν κάποιος επιλέξει το μενού με το οποίο συνδέεται.
Έξοδος	-
Περιγραφή	Εμφανίζει το παράθυρο διαλόγου «Σχετικά ...» που ανήκει στην κλάση AboutDialog.
<b>helpJMenuItemActionPerformed(ActionEvent evt)</b>	
Είσοδος	ActionEvent evt – συμβάν που προκαλείται όταν κάποιος επιλέξει το μενού με το οποίο συνδέεται.
Έξοδος	-
Περιγραφή	Εμφανίζει το παράθυρο διαλόγου που παρέχει την βοήθεια για το παιχνίδι και ανήκει στην κλάση HelpDialog.
<b>newJMenuItemActionPerformed(ActionEvent evt)</b>	
Είσοδος	ActionEvent evt – συμβάν που προκαλείται όταν κάποιος επιλέξει το μενού με το οποίο συνδέεται.
Έξοδος	-
Περιγραφή	Εμφανίζει το παράθυρο διαλόγου για νέο παιχνίδι κάνοντας έναν έλεγχο πρώτα αν εκτελείται κάποιο παιχνίδι.
<b>saveJMenuItemActionPerformed(ActionEvent evt)</b>	
Είσοδος	ActionEvent evt – συμβάν που προκαλείται όταν κάποιος επιλέξει το μενού με το οποίο συνδέεται.
Έξοδος	-
Περιγραφή	Εμφανίζει το παράθυρο διαλόγου για την αποθήκευση παιχνιδιού αφού πρώτα ελέγξει ότι εκτελείται κάποιο παιχνίδι και το βάλει σε κατάσταση παύσης. Έπειτα κάνοντας χρήση της κλάσης JFileChooser, εμφανίζει ένα παράθυρο διαλόγου για την επιλογή του αρχείου προς αποθήκευση.

<b>loadJMenuItemActionPerformed(ActionEvent evt)</b>	
Είσοδος	ActionEvent evt – συμβάν που προκαλείται όταν κάποιος επιλέξει το μενού με το οποίο συνδέεται.
Έξοδος	-
Περιγραφή	Εμφανίζει το παράθυρο διαλόγου για την φόρτωση παιχνιδιού από το δίσκο. Έπειτα κάνοντας χρήση της κλάσης JFileChooser, εμφανίζει ένα παράθυρο διαλόγου για την επιλογή του αρχείου προς φόρτωση.
<b>pauseJMenuItemActionPerformed(ActionEvent evt)</b>	
Είσοδος	ActionEvent evt – συμβάν που προκαλείται όταν κάποιος επιλέξει το μενού με το οποίο συνδέεται.
Έξοδος	-
Περιγραφή	Προκαλεί παύση ή συνέχιση του παιχνιδιού ανάλογα με την προηγούμενη κατάσταση.
<b>windowOpened(WindowEvent e), windowClosing(WindowEvent e), windowClosed(WindowEvent e), windowIconified(WindowEvent e), windowActivated(WindowEvent e), windowDeactivated(WindowEvent e),</b>	
Είσοδος	WindowEvent e – συμβάν σχετικό με το παράθυρο
Έξοδος	-
Περιγραφή	Πρόκειται για τις μεθόδους της διεπαφής WindowListener και είναι απαραίτητο να υλοποιηθούν όλες. Προστίθεται η λειτουργία της παύσης σε όσες έχουν να κάνουν με τον έλεγχο του παραθύρου ώστε όταν κάποιος αφήσει από τον έλεγχό του το παράθυρο να μην συνεχίζει να παίζει το παιχνίδι, αλλά να κάνει παύση.

### 6.1.3.2 Κλάση NewGameDialog

Η NewGameDialog κληρονομεί την JDialog και συλλέγει όλα τα απαραίτητα στοιχεία για να ξεκινήσει ένα νέο παιχνίδι. Πρόκειται για μία κλάση που απεικονίζει μία φόρμα εισαγωγής στοιχείων.



Εικόνα 6-8 - Κλάση `NewGameDialog`

Στην Εικόνα 6-8 βλέπουμε τις κλάσεις με τις οποίες αλληλεπιδρά η `NewGameDialog`, η εσωτερική κλάση `NamePlayerDocumentFilter` χρησιμοποιείται για φιλτράρισμα στα ονόματα των παιχτών και δεν τους επιτρέπει όνομα μεγαλύτερο από την τιμή της ιδιότητας `MAXPLAYERCHARS`.

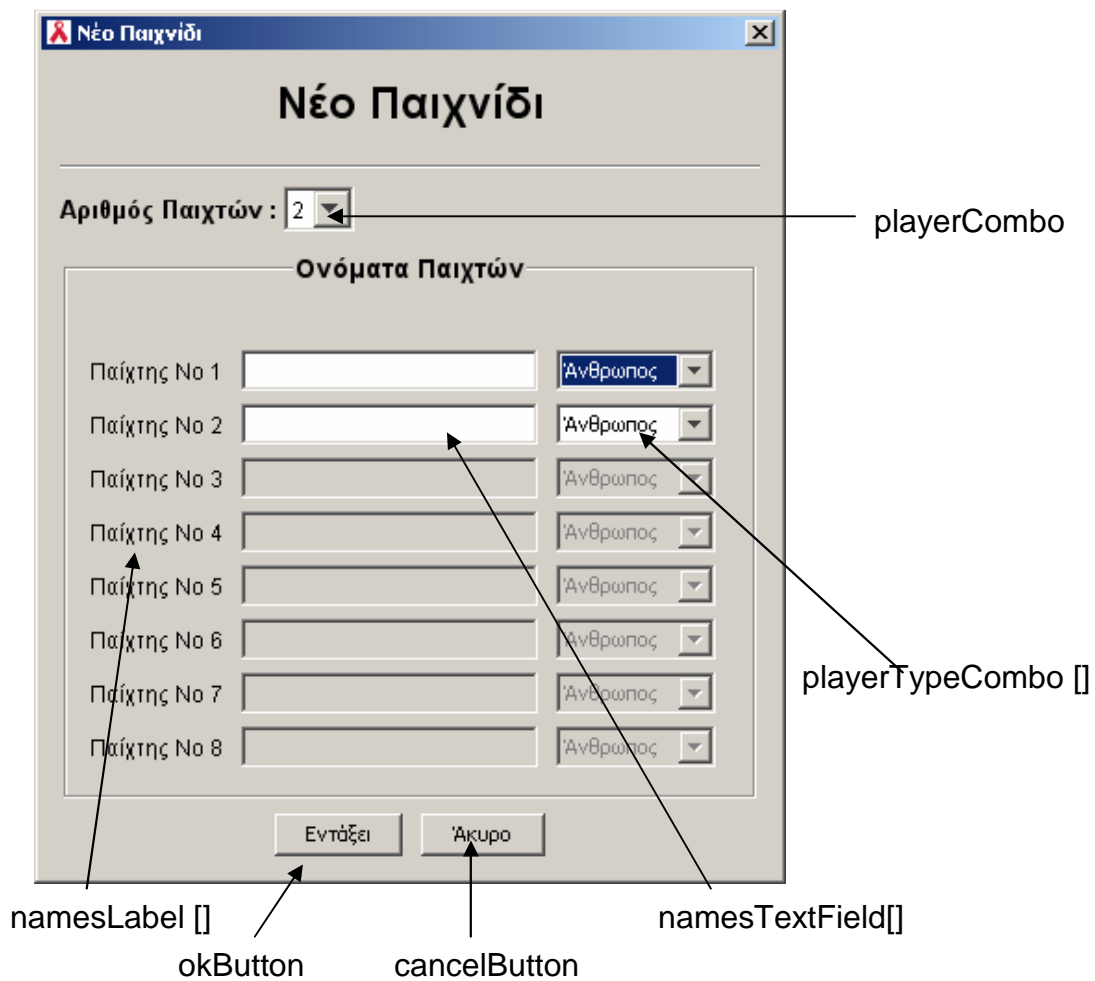
**Πίνακας 6-15 - Ιδιότητες κλάσης NewGameDialog**

Όνομα	Τύπος	Περιγραφή
MAXPLAYERCHARS	int	Αριθμητική τιμή που ορίζει το μέγιστο πλήθος χαρακτήρων του ονόματος ενός παίχτη.
okButton	JButton	Κουμπί για OK.
cancelButton	JButton	Κουμπί για ΑΚΥΡΟ.
playerNamePanel	JPanel	Πάνελ που περιέχει τα συστατικά στα οποία εισάγουν οι παίχτες τα στοιχεία τους.
jSeparator1	JSeparator	Διαχωριστική γραμμή.
playerCombo	JComboBox	Σύνθετο πλαίσιο (combo box) για την εισαγωγή του αριθμού των παιχτών.
namesLabel	JLabel[]	Πίνακας ετικετών για τους παίχτες
namesTextField	JTextField[]	Πίνακας πεδίων εισαγωγής κειμένου για το όνομα του κάθε παίχτη.
playerTypeCombo	JComboBox[]	Πίνακας από σύνθετα πλαίσια για την εισαγωγή του τύπου κάθε παίχτη.
numPlayersjLabel	JLabel	Ετικέτα της φόρμας.
titleLabel	JLabel	Ετικέτα της φόρμας.
main	Main	Η κύρια κλάση Main, χρησιμοποιείται για να σταλθεί μήνυμα προς αυτή όταν πρόκειται να ξεκινήσει νέο παιχνίδι.

**Πίνακας 6-16 - Ιδιότητες της κλάσης NewGameDialog**

<b>NewGameDialog(Frame parent, boolean modal, Main main)</b>	
Είσοδος	Frame parent – το γονικό αντικείμενο από το οποίο καλείται. boolean modal – αν θα είναι το παράθυρο ανεξάρτητο (modal) του γονικού παραθύρου. Main main – η κύρια κλάση.
Έξοδος	-
Περιγραφή	Ο δομητής δημιουργεί και αρχικοποιεί όλα τα στοιχεία της φόρμας.
<b>cancelAction()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Εσωτερική μέθοδος που προκαλεί ακύρωση, δηλαδή εξαφάνιση της φόρμας και συνέχιση του παιχνιδιού, αν αυτό υπάρχει.
<b>okAction(ActionEvent evt)</b>	
Είσοδος	ActionEvent evt – το συμβάν όταν πατηθεί το πλήκτρο με το οποίο συνδέεται.
Έξοδος	-
Περιγραφή	Κάνει έλεγχο αν έχει γίνει ορθή συμπλήρωση των στοιχείων και έπειτα στέλνει μήνυμα με τους παίχτες στην κύρια κλάση ώστε να ξεκινήσει νέο παιχνίδι. Τέλος εξαφανίζει την φόρμα.
<b>comboActionPerformed(ActionEvent evt)</b>	
Είσοδος	ActionEvent evt – το συμβάν όταν ενεργοποιηθεί το σύνθετο πλαίσιο με το οποίο συνδέεται.
Έξοδος	-
Περιγραφή	Χρησιμοποιείται από το σύνθετο πλαίσιο με τον αριθμό των παιχτών , όταν αυτό αλλάξει τιμή τότε ενεργοποιεί τα υπόλοιπα συστατικά, π.χ. όταν ο αριθμός των παιχτών γίνει 4, τότε σε 4 παίχτες θα μπορούμε να συμπληρώσουμε τα στοιχεία.

Η Εικόνα 6-9 μας δείχνει την αντιστοιχία των συστατικών της NewGameDialog.



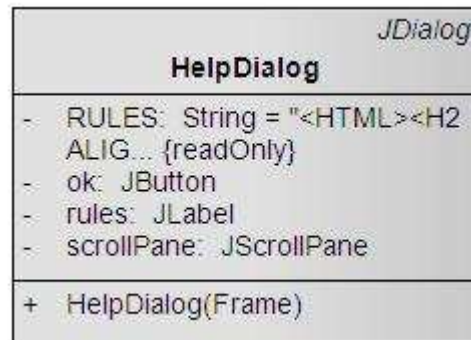
Εικόνα 6-9 - Συστατικά της NewGameDialog

Η εσωτερική κλάση NamePlayerDocumentFilter, που υπάρχει στην NewGameDialog, κληρονομεί την κλάση DocumentFilter (του πακέτου javax.swing.text) και έχει την ιδιότητα της προσάρτησης σε στοιχεία σχετικά με κείμενο, όπως το JTextField, με σκοπό το φιλτράρισμά του. Στην συγκεκριμένη περίπτωση δεν επιτρέπει την εισαγωγή περισσότερων από MAXPLAYERCHARS χαρακτήρες στα ονόματα των παιχτών.

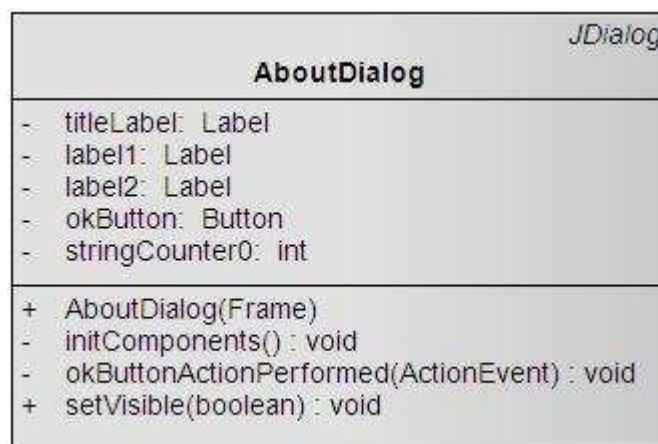
### 6.1.3.3 Κλάσεις HelpDialog και AboutDialog

Οι δύο κλάσεις έχουν ως μοναδικό τους σκοπό να παρουσιάζουν κείμενο για την πληροφόρηση των χρηστών, επομένως δεν απαιτείται κάποιο ιδιαίτερο χαρακτηριστικό εκτός των ετικετών και των κουμπιών τους.

Για την ευκολότερη εισαγωγή και μορφοποίηση κειμένου μπορεί να χρησιμοποιηθεί η γλώσσα HTML στις ετικέτες καθώς τα αντικείμενα της κλάσης JLabel μπορούν να παρουσιάσουν κείμενο με την μορφοποίηση της HTML.



**Εικόνα 6-10 - Κλάση HelpDialog**



**Εικόνα 6-11 - Κλάση AboutDialog**

Οι κλάσεις παρουσιάζονται στις Εικόνα 6-10 και Εικόνα 6-11, δεν γίνεται περαιτέρω ανάλυση καθώς δεν παρουσιάζουν ενδιαφέρον.

#### **6.1.3.4 Κλάση GlassQuestionShow**

Σκοπός της κλάσης είναι να παρουσιάσει μία ερώτηση βιβλίου όταν απαιτείται. Η ερώτηση παρουσιάζεται καλύπτοντας όλη την επιφάνεια του παραθύρου και τα συστατικά. Όταν πρόκειται να παρουσιαστεί προκαλούμε παύση του παιχνιδιού, αν στο ενδιάμεσο διάστημα δώσουμε τον έλεγχο σε ένα άλλο παράθυρο αυτόματα η ενέργεια αυτή δίνει εντολή για παύση. Το πρόβλημα δημιουργείται όταν επαναφέρουμε το παράθυρο του παιχνιδιού και δίνεται η εντολή για συνέχιση του παιχνιδιού, ενώ δεν το επιθυμούμε. Για τον διαχωρισμό των δύο καταστάσεων κλειδώνουμε την παύση από την GlassQuestionShow και το παιχνίδι συνεχίζεται

όταν αυτή το αποφασίσει. Αυτό επιτυγχάνεται με το μήνυμα wait() στο ξεχωριστό νήμα που τρέχει το παιχνίδι (GameEngine).

Η GlassQuestionShow για μπορέσει να καλύψει όλα τα υπόλοιπα συστατικά του βασικού παραθύρου (SafeFrame) πρέπει να ενεργήσει σε ένα επίπεδο της κλάσης JFrame που ονομάζεται επίπεδο γυαλιού (glass pane). Σε αυτό το επίπεδο ότι ζωγραφίζουμε είναι κρυμμένο και εμφανίζεται όταν εμείς το επιλέξουμε καλύπτοντας όλα τα υπόλοιπα συστατικά. Η GlassQuestionShow συνδέεται με το glass pane της SafeFrame και απλά εμφανίζει ένα κείμενο και ένα κουμπί τέλους.

<b>GlassQuestionShow</b>	
-	panel: JPanel
-	textLabel: JLabel
-	ok: JButton
-	locker: Object
+	GlassQuestionShow(JPanel, Dimension)
-	hide() : void
+	showText(String, Object) : void

**Εικόνα 6-12 - Κλάση GlassQuestionShow**

**Πίνακας 6-17 - Ιδιότητες της GlassQuestionShow**

Όνομα	Τύπος	Περιγραφή
panel	JPanel	Το πάνελ στο οποίο θα τοποθετηθούν τα υπόλοιπα συστατικά.
textLabel	JLabel	Η επικέτα που το κείμενο της περιέχει την ερώτηση βιβλίου.
ok	JButton	Κουμπί για να κλείσει η ερώτηση βιβλίου.
locker	Object	Το αντικείμενο που θα κλειδωθεί ώστε να περιμένει.



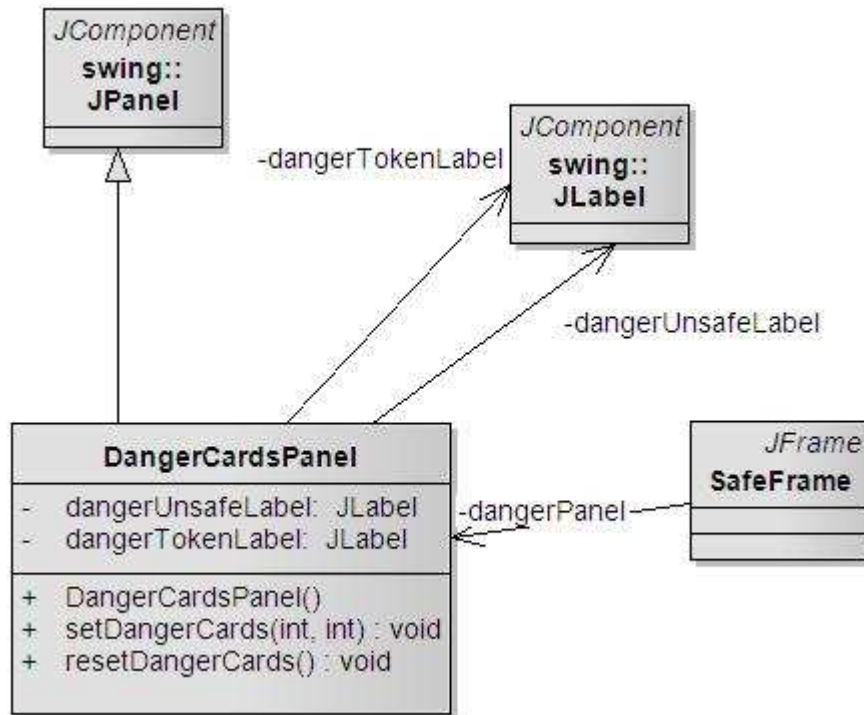
Πίνακας 6-18 - Μέθοδοι της GlassQuestionShow

<b>GlassQuestionShow(JPanel panel, Dimension size)</b>	
Είσοδος	JPanel panel – το πάνελ που θα ζωγραφίσουμε. Dimension size – κλάση της Java που περιέχει πληροφορίες διαστάσεων, στη συγκεκριμένη περίπτωση πρόκειται για τις επιθυμητές διαστάσεις του παραθύρου.
Έξοδος	-
Περιγραφή	Ο δομητής της κλάσης, αρχικοποιεί τα υπόλοιπα αντικείμενα και τα τοποθετεί στο παράθυρο.
<b>hide()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Δίνει την εντολή στο αντικείμενο locker να συνεχίσει την εκτέλεση του και κρύβει την ερώτηση βιβλίου.
<b>showText(String str, Object locker)</b>	
Είσοδος	String str – το κείμενο της ερώτησης βιβλίου Object locker – το αντικείμενο του οποίου η λειτουργία σταματά μέχρι νεωτέρας.
Έξοδος	-
Περιγραφή	Εμφανίζει στην οθόνη της ερώτηση βιβλίου.

#### 6.1.3.5 Κλάσεις DangerCardsPanel και PlayerPanel

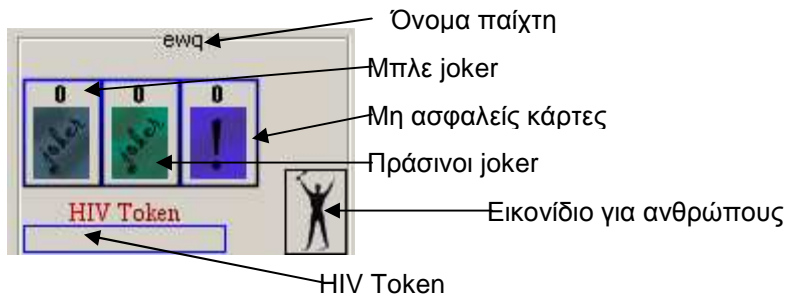
Δύο κλάσεις που αποτελούν συστατικά του βασικού παραθύρου και παρέχουν πληροφορίες για το παιχνίδι είναι η DangerCardsPanel και η PlayerPanel.

Η DangerCardsPanel είναι μία πολύ απλή κλάση που έχει δύο ετικέτες με τα τις τιμές των ορίων HIV Token και μη ασφαλών καρτών. Η δομή της φαίνεται στην Εικόνα 6-13 .



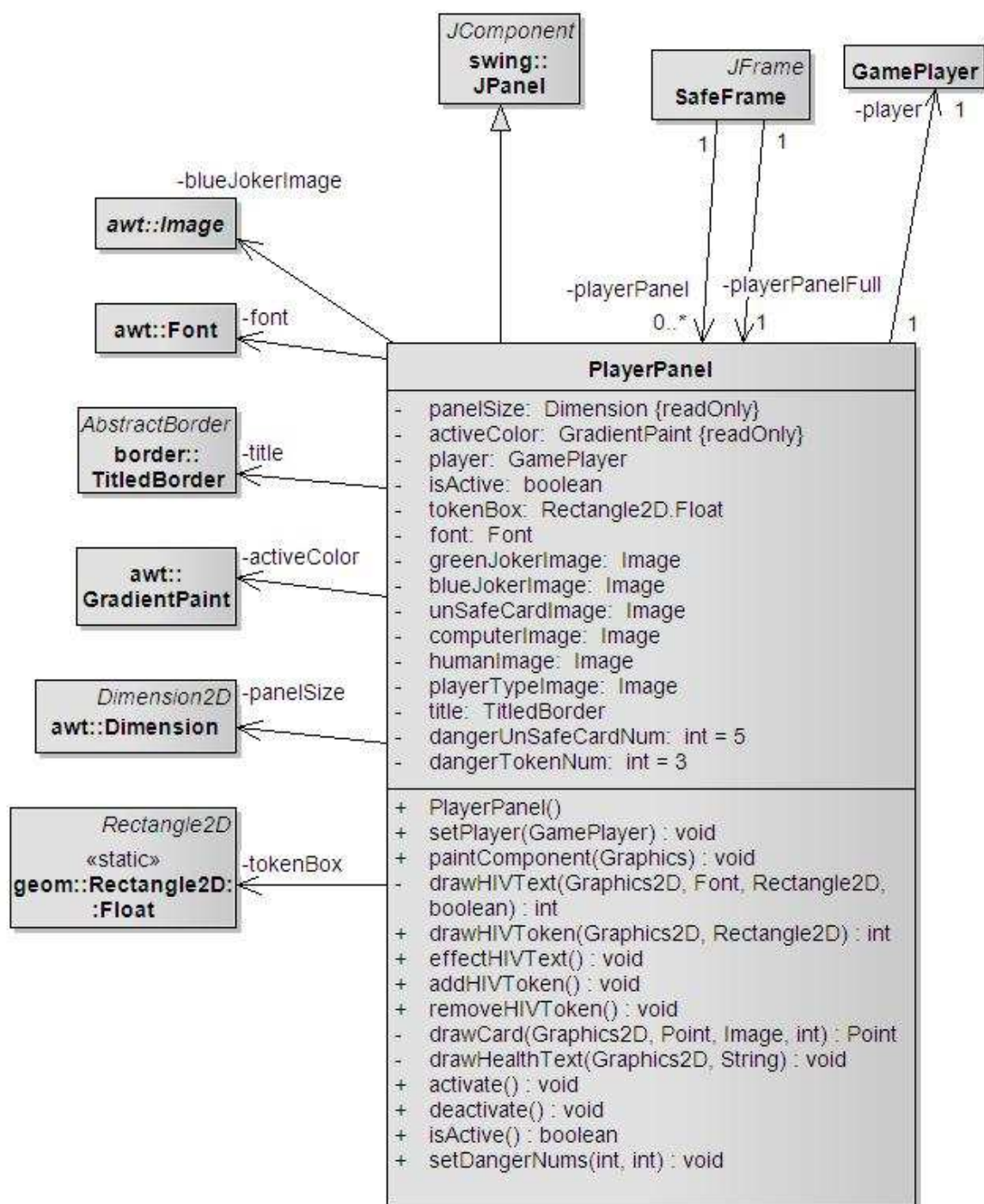
Εικόνα 6-13 - Κλάση DangerCardsPanel

Από την άλλη η PlayerPanel είναι πιο σύνθετη κλάση με περισσότερα συστατικά η οποία παρουσιάζει τα στοιχεία και την κατάσταση του παίχτη που παίζει κάθε στιγμή. Τα συστατικά της και πως είναι τοποθετημένα φαίνονται στην Εικόνα 6-14.



Εικόνα 6-14 - Συστατικά του PlayerPanel

Η κλάση συνδέεται με ένα αντικείμενο της GamePlayer ώστε να παρουσιάζει τα στοιχεία του κάθε στιγμή. Η Εικόνα 6-15 δείχνει την κλάση PlayerPanel και πως συνδέεται με τις άλλες κλάσεις.



Εικόνα 6-15 - Κλάση PlayerPanel

Πίνακας 6-19 - Ιδιότητες της PlayerPanel

Όνομα	Τύπος	Περιγραφή
panelSize	Dimension	Αντικείμενο της κλάσης Dimension που ορίζει διαστάσεις. Αναφέρεται στις διαστάσεις του πάνελ.

Όνομα	Τύπος	Περιγραφή
activeColor	GradientPaint	Αντικείμενο της κλάσης GradientPaint και αναφέρεται στο χρώμα που δείχνει ενεργό το πάνελ.
player	GamePlayer	Ο παίχτης με τον οποίο συνδέεται η κλάση κάθε στιγμή.
isActive	boolean	Ορίζει την κατάσταση του πάνελ, αν είναι ενεργό ή όχι.
tokenBox	Rectangle2D.Float	Ορίζει ένα ορθογώνιο σχήμα στο μέσα στο οποίο ζωγραφίζονται τα token που παίρνει ο παίχτης.
font	Font	Αναφέρεται στην γραμματοσειρά που χρησιμοποιείται.
greenJokerImage, blueJokerImage, unSafeCardImage, computerImage, humanImage, playerTypeImage	Image	Αντικείμενα της κλάσης Image, πρόκειται για εικόνες που χρησιμοποιούνται στο πάνελ.
title	TitledBorder	Το πλαίσιο γύρω από το πάνελ, εκεί αναγράφεται και το όνομα του παίχτη.
dangerUnSafeCardNum, dangerTokenNum	int	Τα όρια των μη ασφαλών καρτών, χρησιμοποιούνται από την κλάση ώστε όταν ξεπεραστούν να γίνει ορατό στον παίχτη με κάποιο μήνυμα.

Πίνακας 6-20 - Μέθοδοι της **PlayerPanel**

<b>PlayerPanel()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Ο δομητής της κλάσης, αρχικοποιεί την κλάση και κάθε οπτικό στοιχείο εκτός αυτών που ταυτοποιούν κάποιο παίχτη.
<b>setPlayer(GamePlayer player)</b>	
Είσοδος	GamePlayer player – ο παίχτης με τον οποίο συνδέεται το πάνελ
Έξοδος	-
Περιγραφή	Συνδέει τον παίχτη της παραμέτρου με την κλάση και βάζει το όνομά του ως τίτλο στο περίγραμμα του περιθωρίου.
<b>paintComponent(Graphics g)</b>	
Είσοδος	Graphics g – Αντικείμενο γραφικών που χρησιμοποιείται για ζωγραφική στην κλάση.
Έξοδος	-
Περιγραφή	Πρόκειται για μέθοδο υπερκλάσης που χρησιμοποιείται όταν θέλουμε να ζωγραφίσουμε.
<b>drawHIVText(Graphics2D g2, Font font, Rectangle2D tokenBox), effectHIVText(), addHIVToken(), removeHIVToken(), drawHealthText(Graphics2D g2, String str)</b>	
Είσοδος	
Έξοδος	
Περιγραφή	Μέθοδοι που χρησιμοποιούνται για οπτική αναπαράσταση των αντικειμένων
<b>activate(), deactivate()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Χρησιμοποιούνται για να ορίσουν αν το πάνελ είναι ενεργό ή όχι. Το ενεργό πάνελ έχει διαφορετικό φόντο.

<b>isActive()</b>	
Είσοδος	-
Έξοδος	boolean
Περιγραφή	Μας λέει αν το πάνελ είναι ενεργό.
<b>setDangerNums(int dangerTokenNum, int dangerUnSafeCardNum)</b>	
Είσοδος	int dangerTokenNum – το όριο των HIV Token int dangerUnSafeCardNum – το όριο των μη ασφαλών καρτών
Έξοδος	-
Περιγραφή	Πληροφορεί την κλάση για τα όρια των μη ασφαλών καρτών και HIV Token.

#### 6.1.4 Μηχανή του παιχνιδιού και κλάσεις animation

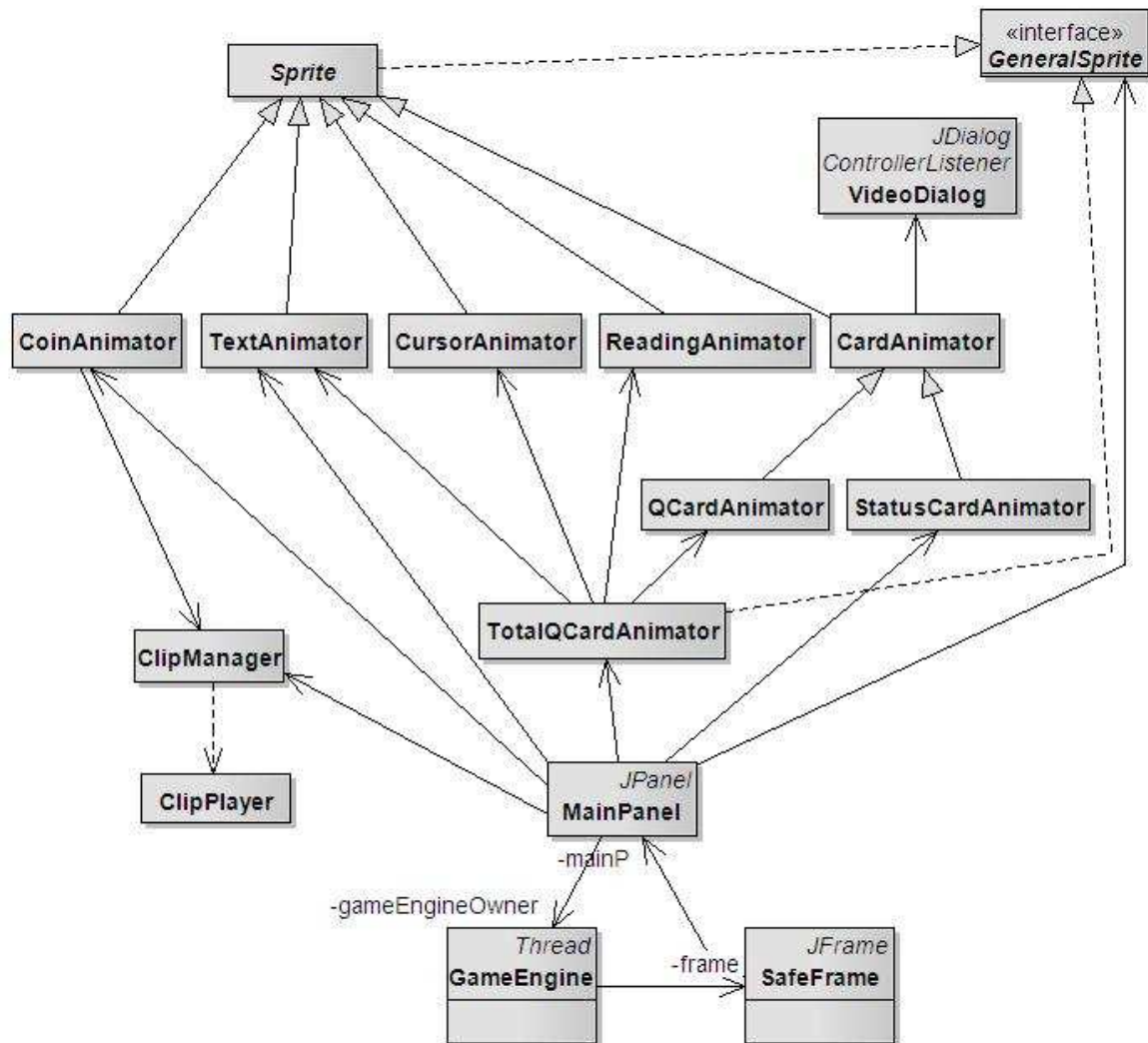
Το μεγαλύτερο μέρος του παιχνιδιού βασίζεται στο animation, για να επιτευχθεί θα χρησιμοποιήσουμε την MainPanel ως το χώρο που θα φιλοξενήσει τα σχέδια και η GameEngine θα συντονίσει το πότε και ποια θα εμφανιστούν. Στο συγκεκριμένο παιχνίδι η κίνηση των εικόνων-σχεδίων δεν είναι συνεχόμενη αλλά αποτελείται από διάφορες φάσεις όπως π.χ. την εμφάνιση μια κάρτας ή το ρίξιμο ενός κέρματος.

Η MainPanel αναλαμβάνει φέρει σε πέρας το κάθε animation με την χρήση ενός βρόχου που αποτελείται από τέσσερα στάδια :

- animUpdate() – ενημέρωση για την φάση που βρίσκεται το παιχνίδι και το αν πρέπει να συνεχίσει να εκτελείται το παιχνίδι.
- animRender() – δημιουργία των γραφικών σε έναν ενδιάμεσο buffer
- animPaint() - ζωγράφισμα των γραφικών που δημιουργήθηκαν
- μικρή παύση – παύση ανάλογη του χρόνου εκτέλεσης των προηγούμενων ενεργειών και της επιθυμητής ταχύτητας του παιχνιδιού.

Κάθε διαφορετικό αντικείμενο του animation εκπροσωπείται από μία ξεχωριστή κλάση, ενώ παράλληλα προστίθεται μία κλάση που δίνει την δυνατότητα αναπαραγωγής ήχου.

Το ποιο αντικείμενο θα εμφανιστεί, πότε και το χρονικό διάστημα που θα διαρκέσει η εμφάνισή του καθορίζεται από την GameEngine. Στην Εικόνα 6-16 βλέπουμε το διάγραμμα κλάσεων που είναι σχετικές με το animation.

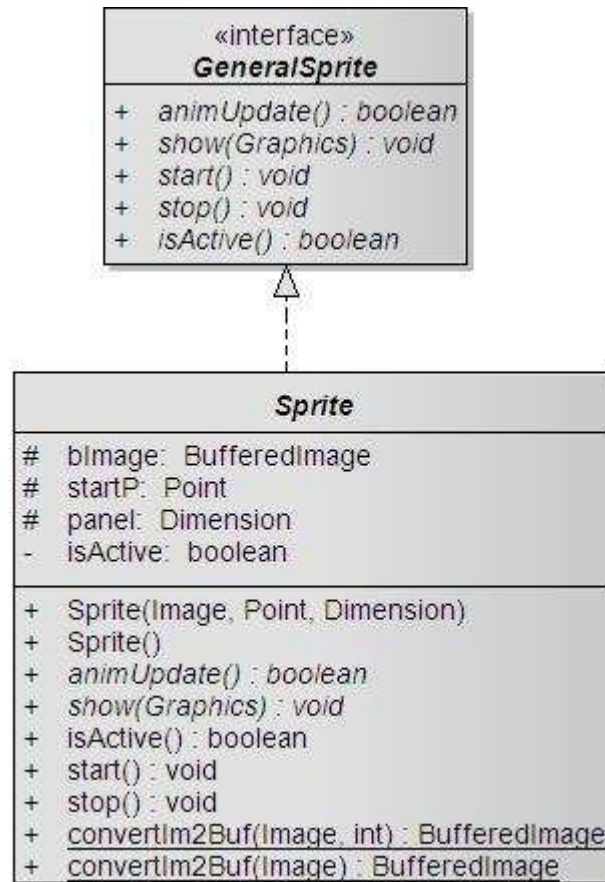


Εικόνα 6-16 - Διάγραμμα κλάσεων για το animation

Θεωρούμε ως μικρότερο συστατικό για κάποιο κινούμενο σχέδιο όλες τις κλάσεις που υλοποιούν την διεπαφή GeneralSprite. Η κλάση Sprite αναφέρεται αφηρημένα σε κάποιο αντικείμενο που κινείται και οι υποκλάσεις της σε συγκεκριμένα, όπως κάποιο κείμενο. Επιπλέον υπάρχουν κλάσεις που έχουν συνδυασμούς από Sprite αντικείμενα, υλοποιώντας την διεπαφή GeneralSprite. Οι κλάσεις ClipManager και ClipPlayer ασχολούνται με την αναπαραγωγή ήχου, ενώ η VideoDialog αναπαράγει βίντεο σε ένα ξεχωριστό παράθυρο.

### 6.1.4.1 Διεπαφή **GeneralSprite** και κλάση **Sprite**

Η διεπαφή **GeneralSprite** ορίζει τις βασικές μεθόδους που θα πρέπει να έχει ένα κινούμενο σχέδιο, χωρίς να τις υλοποιεί. Ένα επίπεδο πιο κάτω βρίσκεται η κλάση **Sprite** η οποία υλοποιεί την **GeneralSprite** ενώ αφήνει κάποιες μεθόδους ως αφηρημένες (abstract) για να υλοποιηθούν από τις υποκλάσεις της.



Εικόνα 6-17 - Διεπαφή **GeneralSprite** και κλάση **Sprite**

Πίνακας 6-21 - Μέθοδοι της **GeneralSprite**

<b>animUpdate()</b>	
Είσοδος	-
Έξοδος	boolean
Περιγραφή	Κάνει τις απαραίτητες αλλαγές στο σχέδιο και επιστρέφει true ή false ανάλογα με το αν το σχέδιο συνεχίζει να εμφανίζεται ή όχι.



<b>show(Graphics g)</b>	
Είσοδος	Graphics g – αντικείμενο της κλάσης Graphics (java.awt.Graphics)
Έξοδος	-
Περιγραφή	Η μέθοδος ζωγραφίζει το σχέδιο στο g της κλάσης Graphics που της δίνεται ως παράμετρος.
<b>start()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Ξεκινάει το animation.
<b>stop()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Σταματάει το animation.
<b>isActive()</b>	
Είσοδος	-
Έξοδος	boolean
Περιγραφή	Απαντά αν το animation είναι ενεργό εκείνη τη στιγμή.

Πίνακας 6-22 - Ιδιότητες της Sprite

Όνομα	Τύπος	Περιγραφή
bImage	BufferedImage	Αντικείμενο της κλάσης BufferedImage (java.awt.image.BufferedImage) που περιγράφει μία εικόνα με ενδιάμεση μνήμη(buffer), εκεί αποθηκεύεται και επεξεργάζεται η εικόνα του σχεδίου του animation.
startP	Point	Το σημείο εκκίνησης (πάνω αριστερά) του σχεδίου
panel	Dimension	Το πάνελ που θα κινηθεί το σχέδιο.
isActive	boolean	Μεταβλητή που δείχνει αν είναι ενεργό το animation.

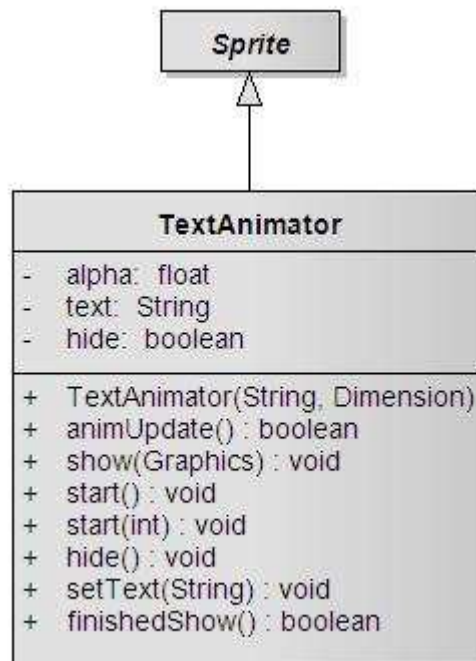
Πίνακας 6-23 - Ιδιότητες της Sprite

<b>Sprite(Image im, Point startP, Dimension panel)</b>	
Είσοδος	Image im – αντικείμενο της κλάσης Image (java.awt.Image) που αντιπροσωπεύει κάποια εικόνα, χρησιμοποιείται ως αρχικό σχέδιο για το animation Point startP – αντικείμενο της κλάσης Point (java.awt.Point) που αναπαριστά κάποιο σημείο στον χώρο, αναφέρεται πάνω αριστερό σημείο που θα τοποθετηθεί το σχέδιο. Dimension panel – οι διαστάσεις του πάνελ στο οποίο θα κινηθεί η εικόνα.
Έξοδος	-
Περιγραφή	Ο δομητής της κλάσης που αρχικοποιεί τις παραμέτρους.
<b>animUpdate(), show(Graphics g)</b>	
Περιγραφή	Μέθοδοι που παραμένουν αφηρημένες και υλοποιούνται από τις κλάσεις που βρίσκονται ένα επίπεδο πιο κάτω.
<b>isActive()</b>	
Είσοδος	-
Έξοδος	boolean
Περιγραφή	Επιστρέφει αν το animation είναι ενεργό ή όχι.
<b>start(), stop()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Υλοποιούνται και δίνουν εκκίνηση ή τέλος στο animation.

<b>convertIm2Buf(Image im,int imageType)</b>	
Είσοδος	Image im – η εικόνα που θα μετατραπεί int imageType – αριθμός που αντιπροσωπεύει την δημιουργία της εικόνας.
Έξοδος	BufferedImage – η εικόνα που μετατράπηκε.
Περιγραφή	Μέθοδος εργαλείο που χρησιμοποιείται για την μετατροπή ενός αντικειμένου της κλάσης Image σε αντικείμενο της κλάσης BufferedImage.

#### 6.1.4.2 Κλάση TextAnimator

Ζητάμε από την κλάση TextAnimator να εμφανίσει κάποιο μήνυμα προς τους παίκτες. Το μήνυμα θέλουμε να εμφανίζεται και να εξαφανίζεται σταδιακά, ο χρόνος παραμονής του είτε δίνεται από την αρχή , π.χ. είτε 30 δευτερόλεπτα και μετά εξαφανίζεται είτε γίνεται εμφάνιση του κειμένου και ο χρόνος παραμένει αόριστος και εξαφανίζεται όταν δοθεί η εντολή.



Εικόνα 6-18 - Κλάση TextAnimator

Πίνακας 6-24 - Ιδιότητες της TextAnimator

Όνομα	Τύπος	Περιγραφή
-------	-------	-----------

Όνομα	Τύπος	Περιγραφή
alpha	float	Μεταβλητή που χρησιμοποιείται για να ορίσει το ποσοστό διαφάνειας του σχεδίου κάθε στιγμή.
text	String	Το κείμενο προς εμφάνιση.
hide	boolean	Μεταβλητή που ορίζει αν το κείμενο είναι σε φάση εμφάνισης ή εξαφάνισης.

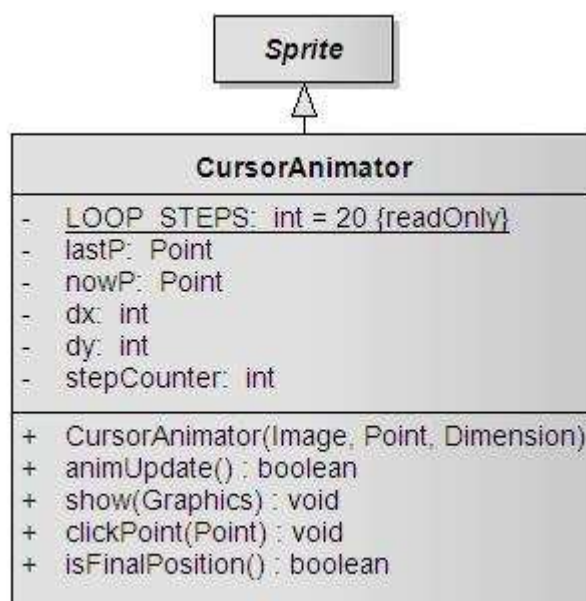
Πίνακας 6-25 - Μέθοδοι της TextAnimator

<b>TextAnimator(String text, Dimension panel)</b>	
Είσοδος	String text – το κείμενο προς εμφάνιση Dimension panel – οι διαστάσεις του πάνελ στο οποίο θα εμφανιστεί το κείμενο.
Έξοδος	-
Περιγραφή	Ο δομητής της κλάσης, αρχικοποιεί τα αντικείμενα και ζωγραφίζει τόσο το κείμενο αλλά και το πλαίσιο που το περιβάλλει. Σημαντικό είναι οι διαστάσεις του πλαισίου να μην είναι πιο μικρές από το χώρο που καταλαμβάνει το κείμενο.
<b>animUpdate()</b>	
Είσοδος	-
Έξοδος	boolean – αν θα γίνει εμφάνιση του γραφικού
Περιγραφή	Εδώ γίνεται ο έλεγχος της μορφής του σχεδίου, αυξάνουμε και μειώνουμε σταδιακά το alpha ανάλογα με την φάση του βρίσκεται το σχέδιο.
<b>show(Graphics g)</b>	
Είσοδος	Graphics g – το αντικείμενο πάνω στο οποίο θα ζωγραφίσουμε το σχέδιο.
Έξοδος	-
Περιγραφή	Ζωγραφίζουμε το σχέδιο που κρατάμε στην ενδιάμεση μνήμη στο αντικείμενο που μας έρχεται ως παράμετρος.

<b>start()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Δίνει εντολή για την εκκίνηση του animation.
<b>start(int hideMillisec)</b>	
Είσοδος	int hideMillisec – ο χρόνος που θα παραμείνει το κείμενο.
Έξοδος	-
Περιγραφή	Δίνει εντολή για την εκκίνηση του animation, να παραμείνει hideMillisec και έπειτα να αρχίσει την εξαφάνιση.
<b>hide()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Δίνει εντολή για την χειροκίνητη εξαφάνιση του σχεδίου.
<b>setText(String text)</b>	
Είσοδος	String text – το νέο κείμενο προς εμφάνιση
Έξοδος	-
Περιγραφή	Ορίζει ένα άλλο κείμενο προς εμφάνιση, όταν αλλαχθεί το κείμενο επιβάλλεται να αναπροσαρμοστεί το πλαίσιο που το περιβάλλει.
<b>finishedShow()</b>	
Είσοδος	-
Έξοδος	boolean – αν έχει τελειώσει η φάση της εμφάνισης
Περιγραφή	Μας πληροφορεί αν είναι το animation ενεργό και έχει τελειώσει η φάση της εμφάνισης του κειμένου.

#### 6.1.4.3 Κλάση CursorAnimator

Η κλάση CursorAnimator αναλαμβάνει να παρουσιάσει τον δείκτη του ποντικιού να διαγράφει μία πορεία. Το συγκεκριμένο animation χρησιμοποιείται όταν παίζει ο Η/Υ και θέλουμε να δείξουμε πως απαντά σε κάποια ερώτηση.



Εικόνα 6-19 - Κλάση CursorAnimator

Πίνακας 6-26 - Ιδιότητες της CursorAnimator

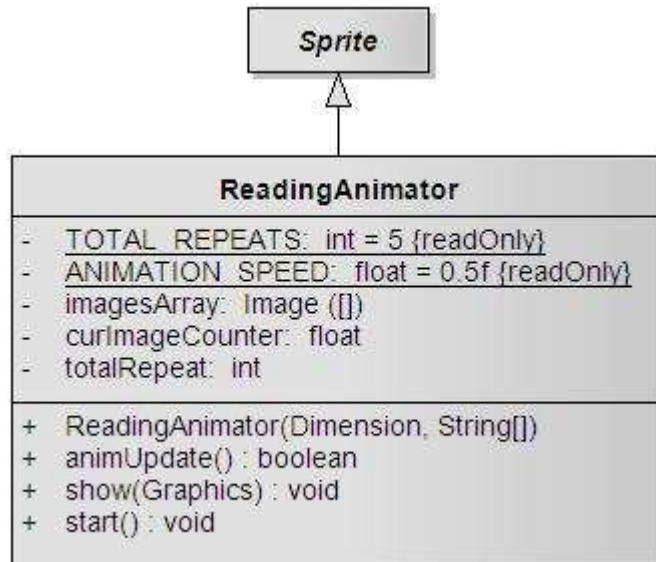
Όνομα	Τύπος	Περιγραφή
LOOP_STEPS	int	Ο αριθμός των βημάτων που θα γίνουν από το αρχικό μέχρι το τελικό σημείο, λιγότερα βήματα σημαίνει γρηγορότερη κίνηση του δείκτη.
lastP	Point	Τελικό σημείο όπου θα καταλήξει ο δείκτης.
nowP	Point	Το τρέχον σημείο που βρίσκεται ο δείκτης.
dx , dy	int	Το μήκος κατά το οποίο θα κινηθεί το σχέδιο στον άξονα x και y αντίστοιχα.
stepCounter	int	Μετρητής του τρέχοντος βήματος.

Πίνακας 6-27- Μέθοδοι της `CursorAnimator`

<b>CursorAnimator(Image im, Point startP, Dimension panel)</b>	
Είσοδος	Image im – η εικόνα που χρησιμοποιείται ως κέρσορας ποντικιού. Point startP – το σημείο εκκίνησης του κέρσορα. Dimension panel – οι διαστάσεις του πάνελ
Έξοδος	-
Περιγραφή	Δομητής της κλάσης
<b>animUpdate()</b>	
Είσοδος	-
Έξοδος	boolean - η τιμή που επιστρέφεται αν εκτελείται το animation
Περιγραφή	Μεταβάλλει κάθε φορά την θέση του κέρσορα και επιστρέφει την ανάλογη boolean τιμή αν είναι ενεργό το animation ή όχι.
<b>show(Graphics g)</b>	
Είσοδος	Graphics g – το αντικείμενο στο οποίο θα ζωγραφίσουμε το σχέδιο.
Έξοδος	-
Περιγραφή	Ζωγραφίζει το σχέδιο.
<b>clickPoint(Point p)</b>	
Είσοδος	Point p – το σημείο στο οποίο ζητάμε να καταλήξει το animation
Έξοδος	-
Περιγραφή	Δίνει εντολή να ξεκινήσει το animation και να καταλήξει στο σημείο p.
<b>isFinalPosition()</b>	
Είσοδος	-
Έξοδος	boolean
Περιγραφή	Επιστρέφει αν το animation έχει φτάσει στην τελική του θέση.

#### 6.1.4.4 Κλάση ReadingAnimator

Η ReadingAnimator εμφανίζει έναν κύριο να ξεφυλλίζει ένα βιβλίο, χρησιμοποιείται όταν θέλουμε να δείξουμε πως ελέγχουμε μία ερώτηση. Μία σειρά εικόνων στις διάφορες φάσεις του ξεφυλλίσματος αποτελούν το υλικό που θα παρουσιαστεί.



Εικόνα 6-20 - Κλάση ReadingAnimator

Πίνακας 6-28 - Ιδιότητες της ReadingAnimator

Όνομα	Τύπος	Περιγραφή
TOTAL_REPEATS	int	Δείκτης που ορίζει πόσες φορές θα εμφανιστεί η σειρά των εικόνων.
ANIMATION_SPEED	float	Η ταχύτητα με την οποία γίνεται η εναλλαγή των εικόνων.
imagesArray	Image[]	Πίνακας με τις εικόνες που θα εμφανιστούν, σημαντικό είναι οι εικόνες να είναι με την σωστή σειρά.
curlImageCounter	float	Δείκτης της τρέχουσας εικόνας προς εμφάνιση, είναι float για να συμβαδίζει με την ταχύτητα.
totalRepeat	int	Αριθμός που δείχνει σε ποια επανάληψη βρίσκεται το animation, μία επανάληψη θεωρείται ένα ολοκληρωμένο ξεφύλλισμα του βιβλίου.



Πίνακας 6-29 – Μέθοδοι της ReadingAnimator

<b>ReadingAnimator(Dimension panel, String[] imagesNames)</b>	
Είσοδος	Dimension panel – οι διαστάσεις του πάνελ όπου θα εμφανιστεί το animation String[] imagesNames – πίνακας ονομάτων αρχείων που αντιστοιχεί στα ονόματα των εικόνων που θα εμφανιστούν
Έξοδος	-
Περιγραφή	Ο δομητής της κλάσης φορτώνει από το δίσκο τις εικόνες και τις αρχικοποιεί στον ανάλογο πίνακα.
<b>animUpdate()</b>	
Είσοδος	-
Έξοδος	boolean
Περιγραφή	Αλλάζει τους δείκτες της εικόνας προς εμφάνιση και όταν τελειώσει το animation στο σύνολό του το κάνει ανενεργό.
<b>show(Graphics g)</b>	
Είσοδος	Graphics g – το αντικείμενο στο οποίο θα ζωγραφίσουμε το σχέδιο.
Έξοδος	-
Περιγραφή	Ζωγραφίζει την εικόνα του animation.
<b>start()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Δίνει την εκκίνηση του animation.

#### 6.1.4.5 Κλάση CardAnimator

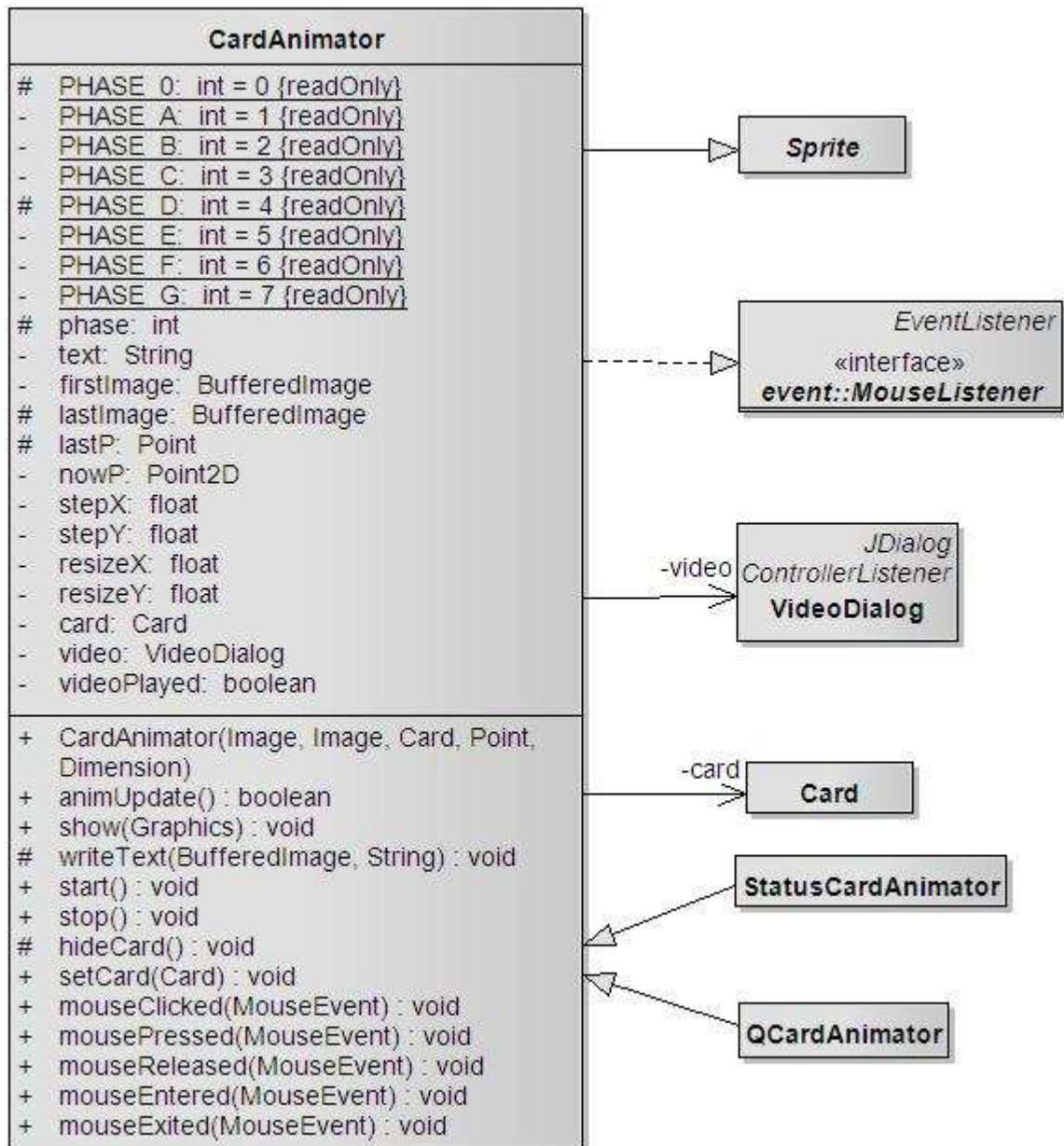
Η κλάση CardAnimator περιγράφει το τράβηγμα μιας κάρτας από το ταμπλό, την εμφάνισή της και την επανατοποθέτηση πίσω. Αντικείμενά της δεν εμφανίζονται παρά μόνο των κλάσεων που την κληρονομούν και αυτές είναι η

StatusCardAnimator που περιγράφει μία κάρτα κατάστασης και η QCardAnimator που περιγράφει μία κάρτα ερώτησης.

Για το χειρισμό της κάρτας χωρίζουμε την κίνηση στις εξής φάσεις:

- Φάση 0 : Αρχική φάση, δεν συμβαίνει τίποτα.
- Φάση A : Η κάρτα μεγαλώνει και κινείται προς το κέντρο.
- Φάση B : Η κάρτα έχει ήδη μεγαλώσει και ξεκινάει το γύρισμα της κάρτας.
- Φάση C : Η κάρτα έχει αλλάξει πλευρά και συνεχίζει να γυρίζει.
- Φάση D : Η κάρτα δεν κινείται, ο χρήστης μπορεί να διαβάσει το κείμενό της.
- Φάση E : Ξεκινάει η αντίστροφη διαδικασία, η κάρτα γυρνάει.
- Φάση F : Εμφανίζεται και πάλι η πίσω πλευρά της κάρτας και η κάρτα συνεχίζει να γυρνάει.
- Φάση G : Η κάρτα έχει πλέον γυρίσει και κινείται προς την αρχική της θέση.

Με βάση τις παραπάνω φάσεις γίνεται ο συντονισμός των σχεδίων προς εμφάνιση. Επίσης η κλάση υλοποιεί την διεπαφή `MouseListener` (`java.awt.event`) ώστε να χειρίζεται τα γεγονότα που συμβαίνουν με το ποντίκι, όπως ένα κλικ σε κάποιο σημείο. Κάτι τέτοιο είναι απαραίτητο για την αλληλεπίδραση με τον χρήστη και την κάρτα, πιο συγκεκριμένα στις κάρτες κατάστασης ο χρήστης κάνει κλικ σε ένα κουμπί της κάρτας ώστε να συνεχιστεί η εκτέλεση, στις κάρτες ερωτήσεων απαντάει στην ερώτηση στο ανάλογο κουμπί της κάρτας. Η `CardAnimator` υλοποιεί την διεπαφή, ορίζει όμως κενές τις μεθόδους της διεπαφής και αφήνει τις υποκλάσεις της να τις υπερβούν.



Εικόνα 6-21 - Κλάση CardAnimator

Πίνακας 6-30 - Ιδιότητες της CardAnimator

Όνομα	Τύπος	Περιγραφή
PHASE_0, PHASE_A, PHASE_C, PHASE_D, PHASE_E, PHASE_F,	int	Οι φάσεις του animation, σε κάθε φάση δίνεται ένας ακέραιος αριθμός.

Όνομα	Τύπος	Περιγραφή
PHASE_G		
phase	int	Αποθηκεύει την τρέχουσα φάση
text	String	Το κείμενο της κάρτας.
firstImage	BufferedImage	Η εικόνα της κάρτας που εμφανίζεται αρχικά, δηλαδή το πίσω μέρος της κάρτας.
lastImage	BufferedImage	Η εικόνα της κάρτας που εμφανίζεται κατά την παρουσίαση του κειμένου, δηλαδή το μέσα μέρος της κάρτας.
lastP	Point	Το σημείο που καταλήγει η κάρτα
nowP	Point2D	Το τρέχον σημείο που βρίσκεται η κάρτα.
stepX, stepY	float	Το μήκος κατά το οποίο μετατοπίζεται όταν μετακινείται η κάρτα.
resizeX, resizeY	float	Το μήκος κατά το οποίο μεγαλώνει ή μικραίνει η κάρτα(ανάλογα την φάση).
card	Card	Η τρέχουσα κάρτα στην οποία αντιστοιχεί το animation.
video	VideoDialog	Παράθυρο διαλόγου που αναπαράγει το βίντεο, αν η κάρτα που εμφανίζεται είναι συνδεδεμένη με κάποιο βίντεο.
videoPlayed	boolean	Αν έχει παιχτεί το βίντεο της κάρτας.

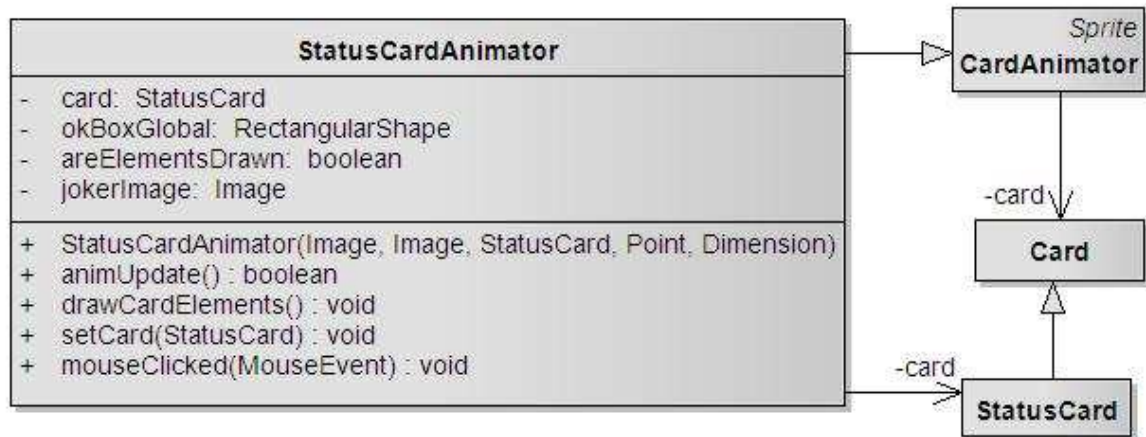
Πίνακας 6-31 - Μέθοδοι της CardAnimator

<b>CardAnimator(Image firstImage, Image lastImage, Card card, Point startPoint, Dimension panel)</b>	
Είσοδος	Image firstImage – η αρχική εικόνα της κάρτας. Image lastImage – η εικόνα της κάρτας που έχει όταν παρουσιάζεται το κείμενο. Card card – η κάρτα που αντιστοιχεί το animation Point startPoint – το σημείο εκκίνησης της κάρτας. Dimension panel – οι διαστάσεις του πάνελ στο οποίο θα κινηθεί η κάρτα.
Έξοδος	-
Περιγραφή	Ο δομητής της κλάσης αρχικοποιεί τις παραμέτρους που δέχεται.
<b>animUpdate()</b>	
Είσοδος	-
Έξοδος	boolean - η τιμή που επιστρέφεται λέει αν είναι ενεργό ή όχι το animation
Περιγραφή	Ανάλογα με την φάση που βρίσκεται η κάρτα, η μέθοδος την μετατοπίζει, της αλλάζει μέγεθος ή φάση.
<b>show(Graphics g)</b>	
Είσοδος	Graphics g – το αντικείμενο στο οποίο θα ζωγραφίσουμε το σχέδιο.
Έξοδος	-
Περιγραφή	Ζωγραφίζει την κάρτα στο πάνελ.
<b>writeText(BufferedImage im, String text)</b>	
Είσοδος	BufferedImage im – εικόνα στην οποία θα γράψουμε String text – το κείμενο που θα γράψουμε
Έξοδος	-
Περιγραφή	Βοηθητική μέθοδος για την εγγραφή κειμένου στην εικόνα που έρχεται ως παράμετρος.

<b>start()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Ορίζει την εκκίνηση του animation.
<b>stop()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Ορίζει τον τερματισμό του animation.
<b>hideCard()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Όταν βρισκόμαστε στη φάση όπου ο χρήστης διαβάζει την κάρτα (φάση D), δίνει εντολή να περάσουμε στην επόμενη φάση.
<b>setCard(Card card)</b>	
Είσοδος	Card card
Έξοδος	-
Περιγραφή	Θέτει την κάρτα που δέχεται ως παράμετρο ως την τρέχουσα κάρτα προς προβολή.
<b>mouseClicked(MouseEvent e), mousePressed(MouseEvent e), mouseReleased(MouseEvent e), mouseEntered(MouseEvent e), mouseExited(MouseEvent e)</b>	
Περιγραφή	Κενή υλοποίηση των μεθόδων της διεπαφής MouseListener

#### 6.1.4.6 Κλάση StatusCardAnimator

Η StatusCardAnimator κληρονομεί την CardAnimator και περιγράφει τις κάρτες κατάστασης. Εμφανίζει πληροφορίες για τον τύπο της κάρτας (ασφαλής, μη ασφαλής, κλπ) και διαθέτει ένα κουμπί με το κείμενο «ΕΝΤΑΞΕΙ» για το κλείσιμο της κάρτας.



Εικόνα 6-22- Κλάση StatusCardAnimator

Πίνακας 6-32 - Ιδιότητες της StatusCardAnimator

Όνομα	Τύπος	Περιγραφή
card	StatusCard	Η κάρτα που αντιστοιχεί στο animation.
okBoxGlobal	RectangularShape	Αντικείμενο που περιγράφει ένα ορθογώνιο κουτί,
areElementsDrawn	boolean	Μας λέει αν τα συστατικά έχουν ζωγραφιστεί.
jokerImage	Image	Εικόνα που αντιστοιχεί στις κάρτες joker.

Πίνακας 6-33 – Μέθοδοι StatusCardAnimator

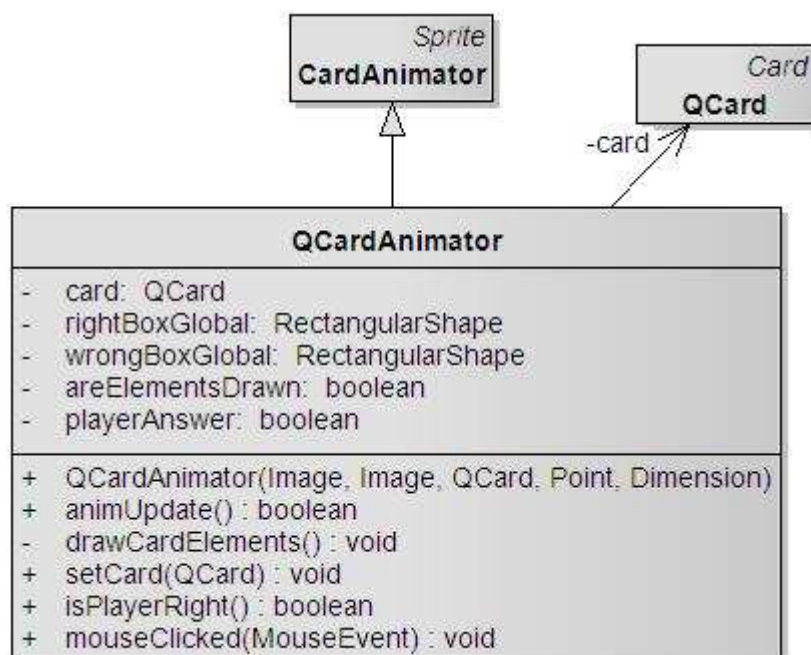
<b>StatusCardAnimator(Image firstImage, Image lastImage, StatusCard card, Point startPoint, Dimension panel)</b>	
Είσοδος	Image firstImage - η αρχική εικόνα της κάρτας. Image lastImage - η τελική εικόνα της κάρτας. StatusCard card - η κάρτα στην οποία αντιστοιχεί το animation. Point startPoint - το σημείο αρχικής τοποθέτησης της κάρτας. Dimension panel - οι διαστάσεις του πάνελ που χρησιμοποιείται.
Έξοδος	-
Περιγραφή	Αρχικοποιεί την κλάση με βάση τις παραμέτρους.

<b>animUpdate()</b>	
Είσοδος	-
Έξοδος	boolean - η τιμή που επιστρέφεται λέει αν είναι ενεργό ή όχι το animation
Περιγραφή	Καλεί την αντίστοιχη μέθοδο της υπερκλάσης που υπερβαίνει και επιπλέον γράφει το ανάλογο κείμενο με τον τύπο της κάρτας πάνω στην κάρτα χρησιμοποιώντας την drawCardElements().
<b>drawCardElements()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Γράφει το κείμενο της κάρτας και τον τύπο της π.χ. «ΜΗ ΑΣΦΑΛΗΣ ΚΑΡΤΑ» και ζωγραφίζει τα απαραίτητα πλαίσια.
<b>setCard(StatusCard card)</b>	
Είσοδος	StatusCard card - κάρτα κατάστασης
Έξοδος	-
Περιγραφή	Δίνει μία νέα κάρτα στην κλάση.
<b>mouseClicked(MouseEvent e)</b>	
Είσοδος	MouseEvent e - συμβάν ποντικιού
Έξοδος	-
Περιγραφή	Ελέγχει το συμβάν ποντικιού σχετικό με το αριστερό κλικ, όταν γίνεται στην περιοχή που έχει ζωγραφιστεί το κουμπί τότε δίνει εντολή να φύγει η κάρτα.

#### 6.1.4.7 Κλάση QCardAnimator

Αντίστοιχη κλάση με την StatusCardAnimator είναι και η QCardAnimator που κληρονομεί την CardAnimator. Πλέον το κείμενο που δίνεται είναι μία ερώτηση και ζωγραφίζονται δύο κουμπιά με το κείμενο «ΣΩΣΤΟ» και «ΛΑΘΟΣ», ο χρήστης απαντά στην ερώτηση και η κάρτα φεύγει.





Εικόνα 6-23 - QCardAnimator

Πίνακας 6-34 - Ιδιότητες της QCardAnimator

Όνομα	Τύπος	Περιγραφή
card	QCard	Η κάρτα ερώτησης που αντιστοιχεί στο animation
rightBoxGlobal, wrongBoxGlobal	RectangularShape	Αντικείμενα που περιγράφουν ένα ορθογώνιο σχήμα, το κάθε ένα αντιστοιχεί στην σωστή και λάθος απάντηση.
areElementsDrawn	boolean	Αν έχουν ζωγραφιστεί τα συσταικά της κάρτας.
playerAnswer	boolean	Την απάντηση που πάρθηκε από τον παίχτη.

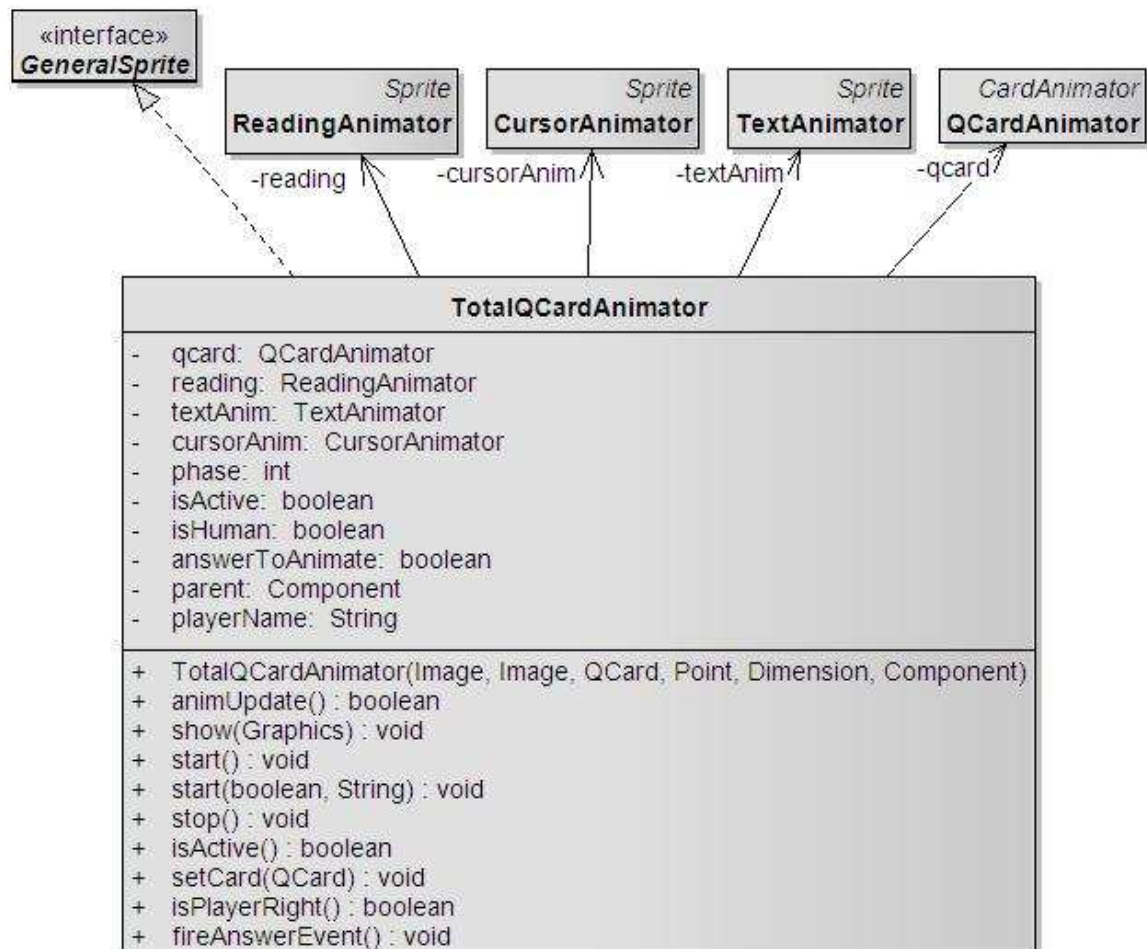
Πίνακας 6-35 - Μέθοδοι της QCardAnimator

<b>QCardAnimator(Image firstImage, Image lastImage, QCard card, Point startPoint, Dimension panel)</b>	
Είσοδος	Image firstImage - η αρχική εικόνα της κάρτας. Image lastImage - η τελική εικόνα της κάρτας. QCard card - η κάρτα ερώτησης. Point startPoint - το σημείο εκκίνησης της κάρτας. Dimension panel - οι διαστάσεις του πάνελ στο οποίο κινείται η κάρτα.
Έξοδος	-
Περιγραφή	Ο δομητής της κλάσης.
<b>animUpdate()</b>	
Είσοδος	-
Έξοδος	boolean - η τιμή που επιστρέφεται λέει αν είναι ενεργό ή όχι το animation
Περιγραφή	Κάνει υπέρβαση της αντίστοιχης μεθόδου της υπερκλάσης, την καλεί αρχικά και συμπληρώνει στοιχεία για την φάση D.
<b>drawCardElements()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Δημιουργεί τα κουμπιά «ΣΩΣΤΟ» και «ΛΑΘΟΣ»
<b>setCard(QCard card)</b>	
Είσοδος	QCard card - κάρτα ερώτησης.
Έξοδος	-
Περιγραφή	Αναθέτει την κάρτα στην κλάση.
<b>isPlayerRight()</b>	
Είσοδος	-
Έξοδος	boolean
Περιγραφή	Ελέγχει την απάντηση του παίχτη και επιστρέφει αν απάντησε σωστά στην ερώτηση ή λάθος.

<b>mouseClicked(MouseEvent e)</b>	
Είσοδος	MouseEvent e - συμβάν ποντικιού
Έξοδος	-
Περιγραφή	Ελέγχει κάθε συμβάν ποντικιού που έγινε από κλικ, όταν γίνεται στα κουμπιά απάντησης της ερώτησης , καταγράφει την ερώτηση και κρύβει την κάρτα.

#### 6.1.4.8 Κλάση TotalQCardAnimator

Συνδυασμό των προηγούμενων κλάσεων κάνει η TotalQCardAnimator προκειμένου να δείξει μια σειρά από animation για το τράβηγμα μιας κάρτας ερώτησης, την απάντηση της και τον έλεγχο της ορθότητας της απάντησης στην ερώτηση. Αυτή η κλάση δεν κληρονομεί την Sprite αλλά μόνο υλοποιεί την GeneralSprite.



Εικόνα 6-24 - Κλάση TotalQCardAnimator

Πίνακας 6-36 - Ιδιότητες της TotalQCardAnimator

Όνομα	Τύπος	Περιγραφή
qcard	QCardAnimator	Η κάρτα ερώτησης που θα σηκωθεί από το τραπέζι.
reading	ReadingAnimator	Ο έλεγχος της απάντησης.
textAnim	TextAnimator	Τα μηνύματα κειμένου προς τον χρήστη.
cursorAnim	CursorAnimator	Ο κινούμενος δείκτης ποντικιού.
phase	int	Ορίζει τις φάσεις του animation.
isActive	boolean	Αν είναι ενεργό το animation.
isHuman	boolean	Αν ο παίχτης είναι άνθρωπος, όταν είναι Η/Υ τότε δημιουργούμε μία εικονική απάντηση.
answerToAnimate	boolean	Όταν πρόκειται για Η/Υ , τότε αυτή είναι η απάντηση που θα δώσει.
parent	Component	Το συστατικό που θα φιλοξενεί την κλάση.
playerName	String	Το όνομα του παίχτη που θα παίξει με την κάρτα.

Πίνακας 6-37 - Μέθοδοι της TotalQCardAnimator

<b>TotalQCardAnimator(Image firstImage, Image lastImage, QCard card, Point startPoint, Dimension panel, Component parent)</b>	
Είσοδος	Image firstImage, Image lastImage, QCard card, Point startPoint - οι παράμετροι για την αρχικοποίηση της κάρτας ερώτησης. Dimension panel - οι διαστάσεις του πάνελ που θα γίνει το animation. Component parent - το συστατικό που θα φιλοξενεί την κλάση.
Έξοδος	-
Περιγραφή	Δομητής της κλάσης.

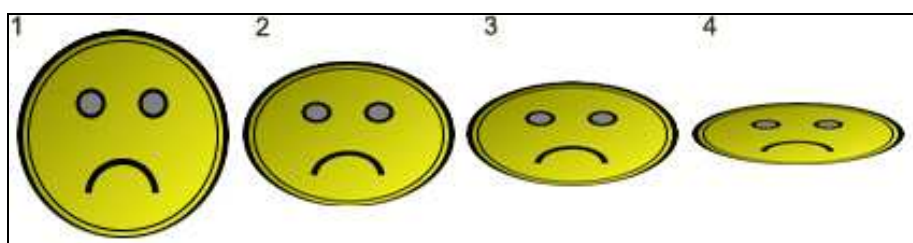
<b>animUpdate()</b>	
Είσοδος	-
Έξοδος	boolean
Περιγραφή	Χωρίζει το animation σε φάσεις και αν ο παίχτης είναι Η/Υ ή άνθρωπος. Κάθε συμβάν συνοδεύεται και με το απαραίτητο κείμενο για την πληροφόρηση του παίχτη.
<b>show(Graphics g)</b>	
Είσοδος	Graphics g - αντικείμενο στο οποίο ζωγραφίζεται το animation.
Έξοδος	-
Περιγραφή	Ζωγραφίζει το animation, ουσιαστικά καλεί τις show() των συστατικών.
<b>start()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Δίνει την εκκίνηση του animation, χρησιμοποιείται όταν ο παίχτης είναι άνθρωπος.
<b>start(boolean answ, String playerName)</b>	
Είσοδος	boolean answ - η απάντηση στην ερώτηση που θα γίνει String playerName - το όνομα του παίχτη που θα παίξει.
Έξοδος	-
Περιγραφή	Δίνει την εκκίνηση στο animation, χρησιμοποιείται όταν ο παίχτης είναι Η/Υ, γι' αυτό και μας δίνεται εξ' αρχής η απάντηση.
<b>stop()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Ορίζει το τέλος του animation.

<b>isActive()</b>	
Είσοδος	-
Έξοδος	Boolean
Περιγραφή	Επιστρέφει την ανάλογη τιμή όταν το animation είναι ενεργό ή όχι.
<b>setCard(QCard card)</b>	
Είσοδος	QCard card - κάρτα ερώτησης.
Έξοδος	-
Περιγραφή	Θέτει μια κάρτα ερώτησης.
<b>isPlayerRight()</b>	
Είσοδος	
Έξοδος	boolean
Περιγραφή	Επιστρέφει αληθές αν ο παίχτης απάντησε σωστά.
<b>fireAnswerEvent()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Τεχνητός τρόπος απάντησης για τον παίχτη που είναι Η/Υ.

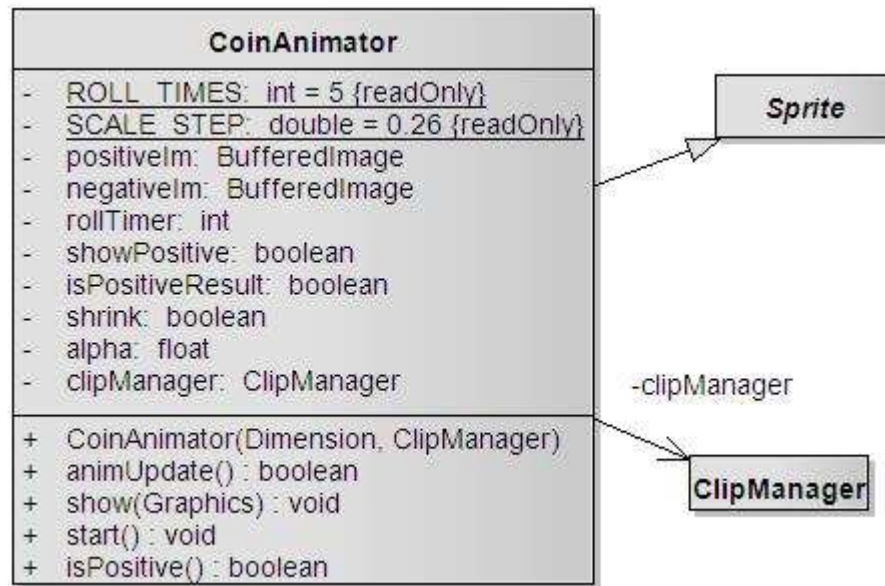
#### 6.1.4.9 Κλάση CoinAnimator

Το κέρμα που στριφογυρνά όταν ο παίχτης βρει μία μη ασφαλή κάρτα, παρουσιάζει η κλάση CoinAnimator. Το κέρμα στριφογυρνά και εμφανίζει τυχαία μία θλιμμένη ή μία χαρούμενη φάτσα. Στην συγκεκριμένη κλάση προστίθεται και ήχος στο τέλος, όταν εμφανίζεται το αποτέλεσμα ακούγεται και η ανάλογη επιφώνηση. Τον ήχο αναλαμβάνει η κλάση ClipManager.

Για να εμφανίσουμε ένα κέρμα να στριφογυρνάει, συμπιέζουμε το κέρμα κατά τον κατακόρυφο άξονα και έπειτα το μεγαλώνουμε ξανά(Εικόνα 6-25).



Εικόνα 6-25 - Φάσεις του κέρματος



Εικόνα 6-26 - Κλάση CoinAnimator

Πίνακας 6-38 - Ιδιότητες της CoinAnimator

Όνομα	Τύπος	Περιγραφή
ROLL_TIMES	int	Το πλήθος των στριφογυρισμάτων που θα κάνει το κέρμα.
SCALE_STEP	double	Το βήμα με το οποίο θα αλλάζει το μέγεθος του κέρματος.
positivelm	BufferedImage	Εικόνα θλιμμένου κέρματος.
negativelm	BufferedImage	Εικόνα χαρούμενου κέρματος.
rollTimer	int	Μετρητής στριφογυρισμάτων.
showPositive	boolean	Αν η πλευρά που δείχνει είναι η θετική.
isPositiveResult	boolean	Αν το αποτέλεσμα που θα προβληθεί στο τέλος είναι θετικό.
shrink	boolean	Αν το μέγεθος του κέρματος μικραίνει ή μεγαλώνει.
clipManager	ClipManager	Αντικείμενο που συνδέεται με την παραγωγή ήχου.

Πίνακας 6-39 - Μέθοδοι της CoinAnimator

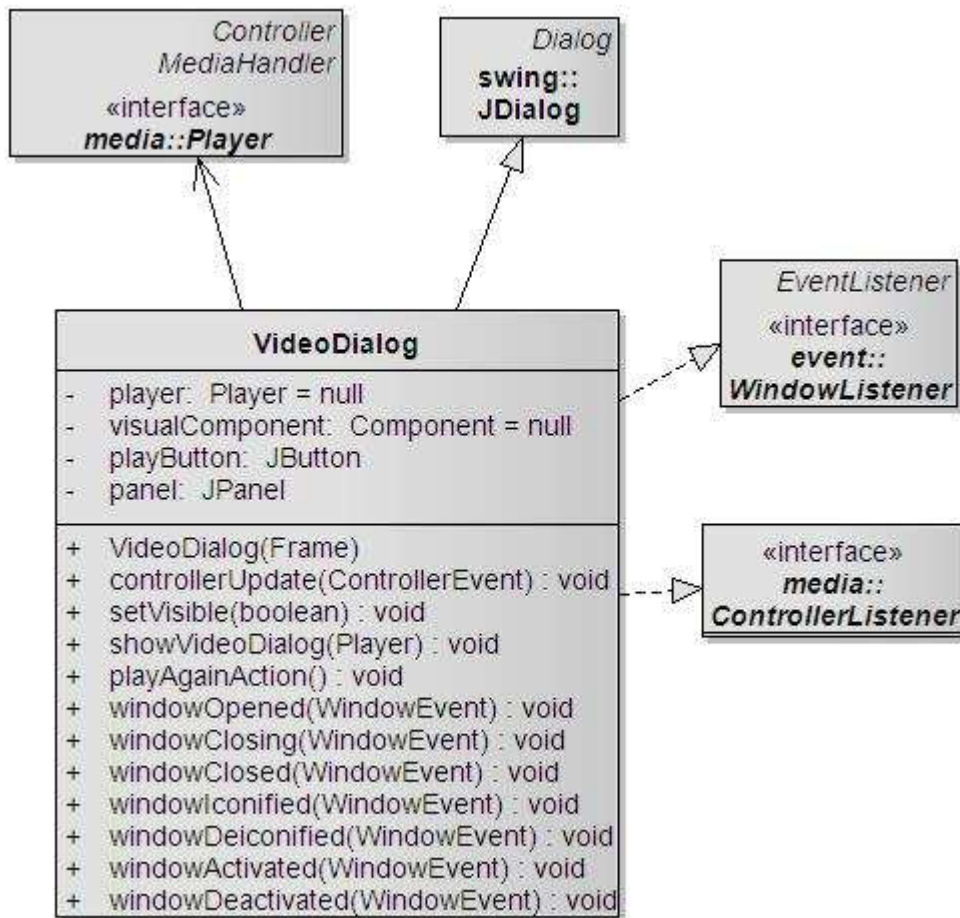
<b>CoinAnimator(Dimension panel, ClipManager clipManager)</b>	
Είσοδος	Dimension panel - οι διαστάσεις του πάνελ πάνω όπου θα εκτελεστεί το animation. ClipManager clipManager -
Έξοδος	-
Περιγραφή	Ο δομητής της κλάσης.
<b>animUpdate()</b>	
Είσοδος	-
Έξοδος	boolean
Περιγραφή	Προσδιορίζει το μέγεθος και την μορφή του κέρματος κάθε χρονική στιγμή.
<b>show(Graphics g)</b>	
Είσοδος	Graphics g - Αντικείμενο όπου ζωγραφίζεται το animation.
Έξοδος	-
Περιγραφή	Ζωγραφίζει το animation.
<b>start()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Ορίζει την εκκίνηση του animation και προαποφασίζει το τυχαίο τελικό αποτέλεσμα.
<b>isPositive()</b>	
Είσοδος	-
Έξοδος	boolean
Περιγραφή	Απαντά αν το αποτέλεσμα είναι θετικό ή όχι.

#### 6.1.4.10 Κλάση VideoDialog

Για την προβολή κάθε βίντεο σχετικού με κάποια κάρτα χρησιμοποιείται από την κλάση CardAnimator, η κλάση VideoDialog. Η κλάση εμφανίζει ένα παράθυρο με όπου αναπαράγεται κάποιο βίντεο σχετικό με το κείμενο της κάρτας, όταν τελειώσει δίνεται η δυνατότητα επανάληψης. Κάνοντας χρήση μιας κλάσης της



Java που ονομάζεται Player (javax.media.Player) και υλοποιώντας την διεπαφή ControllerListener (javax.media.ControllerListener) έχουμε την δυνατότητα να αναπαράγουμε και να ελέγξουμε την ροή αρχείων βίντεο.



Εικόνα 6-27 - Κλάση VideoDialog

Πίνακας 6-40 - Ιδιότητες της VideoDialog

Όνομα	Τύπος	Περιγραφή
player	Player	Το αντικείμενο που αναπαράγει το βίντεο.
visualComponent	Component	Γραφικό συστατικό που δείχνει το βίντεο.
playButton	JButton	Κουμπί για να παίξει ξανά το βίντεο.
panel	JPanel	Πάνελ όπου μπαίνουν τα γραφικά συστατικά.

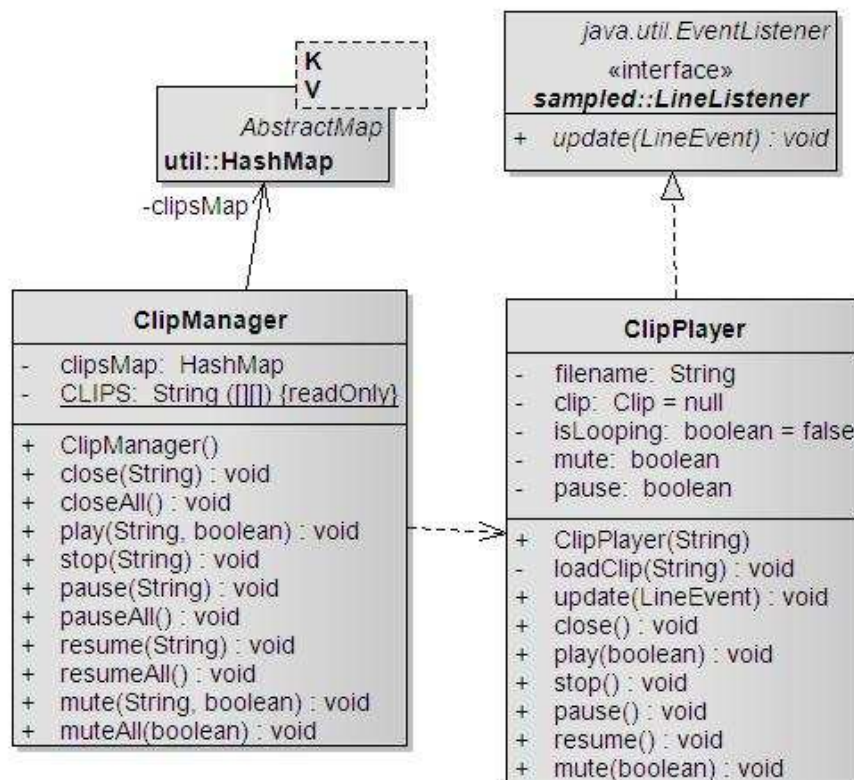
**Πίνακας 6-41 - Μέθοδοι της VideoDialog**

<b>VideoDialog(Frame owner)</b>	
Είσοδος	Frame owner - το παράθυρο γονέας της VideoDialog
Έξοδος	-
Περιγραφή	Ο δομητής της κλάσης, δημιουργεί και αρχικοποιεί τα γραφικά συστατικά χωρίς να τα εμφανίζει.
<b>controllerUpdate(ControllerEvent event)</b>	
Είσοδος	ControllerEvent event -
Έξοδος	-
Περιγραφή	Μέθοδος της διεπαφής ControllerListener, εδώ γίνεται χειρισμός κάθε συμβάντος σχετικό με την αναπαραγωγή του βίντεο.
<b>setVisible(boolean b)</b>	
Είσοδος	boolean b
Έξοδος	-
Περιγραφή	Υπέρβαση της κληρονομημένης κλάσης, δεν θέλουμε να εμφανίζεται το παράθυρο όταν δεν υπάρχει βίντεο να δείξουμε.
<b>showVideoDialog(Player pl)</b>	
Είσοδος	Player pl - αντιστοιχεί σε βίντεο προς αναπαραγωγή.
Έξοδος	-
Περιγραφή	Δίνει το βίντεο που πρόκειται να παιχτεί στο παράθυρο.
<b>playAgainAction()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Δίνει εντολή να παιχτεί πάλι το βίντεο.

<b>windowClosing(WindowEvent e) , windowOpened(WindowEvent e), windowClosed(WindowEvent e), windowIconified(WindowEvent e), windowDeiconified(WindowEvent e), windowActivated(WindowEvent e), windowDeactivated(WindowEvent e)</b>	
Είσοδος	WindowEvent e - συμβάν παραθύρου.
Έξοδος	-
Περιγραφή	Υλοποίηση των μεθόδων της διεπαφής WindowListener ως κενές, εκτός της windowClosing όπου σταματά τον Player και απελευθερώνει την μνήμη που δεσμεύτηκε.

### 6.1.4.11 Κλάσεις ClipManager και ClipPlayer

Για την αναπαραγωγή ήχου χρησιμοποιούνται δύο κλάσεις, η μία για να αναπαράγει ένα αρχείο ήχου (ClipPlayer) και η δεύτερη για να χειριστεί μία συλλογή από την πρώτη (ClipManager), δηλαδή μία συλλογή ήχων. Η ClipPlayer ανοίγει και αναπαράγει ένα αρχείου ήχου χρησιμοποιώντας βιβλιοθήκες τις Java, ενώ η ClipManager δημιουργεί μία συλλογή αντιστοιχίας ονομάτων με αντικείμενα της ClipPlayer, χρησιμοποιώντας μία βοηθητική κλάση την HashMap.



Εικόνα 6-28 - Κλάσεις ClipManager και ClipPlayer

Πίνακας 6-42 - Ιδιότητες της ClipPlayer

Όνομα	Τύπος	Περιγραφή
filename	String	Το όνομα του αρχείου ήχου.
clip	Clip	Κλάση που περιγράφει δεδομένα ήχου που μπορούν να αναπαραχθούν.
isLooping	boolean	Ορίζει την συνεχή αναπαραγωγή.
mute	boolean	Ορίζει το κλείσιμο του ήχου.
pause	boolean	Κατάσταση παύσης.

Πίνακας 6-43 - Μέθοδοι της ClipPlayer

<b>ClipPlayer(String fileName)</b>	
Είσοδος	String fileName - όνομα του αρχείου που περιέχει τον ήχο.
Έξοδος	-
Περιγραφή	Δομητής της κλάσης.
<b>update(LineEvent lineEvent)</b>	
Είσοδος	LineEvent lineEvent - συμβάν
Έξοδος	-
Περιγραφή	Υλοποίηση της μεθόδου της διεπαφής LineListener. Όταν ένα συμβάν τέλους εμφανιστεί, είτε κλείνει το κομμάτι ήχου είτε εκτελείται από την αρχή.
<b>play(boolean toLoop), stop(), pause(), resume(), mute(), close()</b>	
Περιγραφή	Απαραίτητες μέθοδοι για τον χειρισμό του ήχου.

Πίνακας 6-44 - Ιδιότητες της ClipManager

Όνομα	Τύπος	Περιγραφή
clipsMap	HashMap	Αντικείμενο που περιγράφει μία συλλογή ζευγαριών, στη συγκεκριμένη περίπτωση ζευγάρι αποτελεί ένα όνομα με ένα αντικείμενο ήχου ClipPlayer, π.χ. το όνομα "beep" αντιστοιχεί στο ClipPlayer αντικείμενο που φορτώνει τα δεδομένα του "beep.wav".

Όνομα	Τύπος	Περιγραφή
CLIPS	String[][]	Πίνακας με τα αντιστοιχίας ονομάτων με ονόματα αρχείων ήχου, πρόκειται για τα ονόματα που χρησιμοποιούνται στο clipsMap.

Πίνακας 6-45 - Μέθοδοι της ClipManager

<b>ClipManager()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Δομητής της κλάσης.
<b>close(String name), play(String name, boolean toLoop), stop(String name), pause(String name), resume(String name), mute(String name, boolean mute)</b>	
Περιγραφή	Μέθοδοι χειρισμού του για κάθε κομμάτι ήχου που το όνομά του δίνεται ως παράμετρος.
<b>closeAll(), pauseAll(),resumeAll(), muteAll(boolean mute)</b>	
Περιγραφή	Μέθοδοι χειρισμού για όλα τα κομμάτια ήχου.

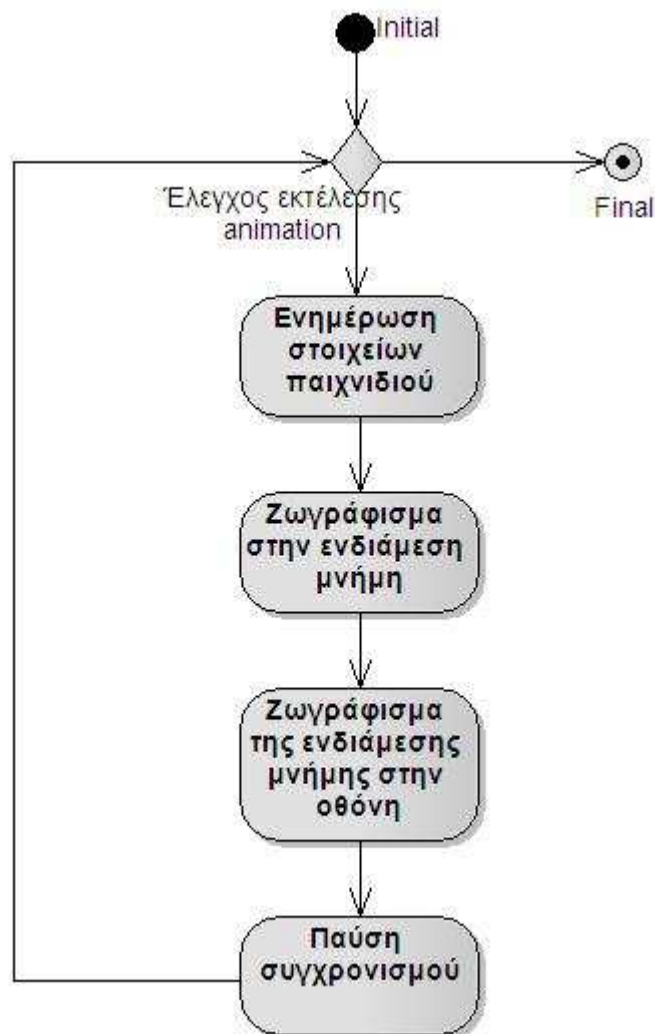
#### 6.1.4.12 Κλάση MainPanel

Το ταμπλό του παιχνιδιού το αναπαριστά η κλάση MainPanel. Η επιφάνειά της αποτελεί τον χώρο που εκτελούνται τα animation και συντονίζεται από την GameEngine, η οποία εκτελείται σε ξεχωριστό νήμα (thread). Η κλάση αποτελείται από στατικά στοιχεία όπως την εικόνα του φόντου και τις στοίβες των καρτών και δυναμικά στοιχεία όπως το κέρμα που γυρνάει ή την κάρτα που δείχνει το περιεχόμενό της. Για την αναπαράσταση των δυναμικών στοιχείων χρησιμοποιείται ένας βρόχος επανάληψης (Εικόνα 6-29) που αποτελείται από τα εξής στάδια :

- ενημέρωση στοιχείων του παιχνιδιού
- ζωγράφισμα στον ενδιάμεση μνήμη
- ζωγράφισμα της ενδιάμεσης μνήμης στην οθόνη
- παύση συγχρονισμού

Απαραίτητη τεχνική αποτελεί η χρήση ενδιάμεσης μνήμης ώστε να αποφευχθεί το τρεμόπαιγμα της οθόνης, στην τεχνική αυτή δεν ζωγραφίζουμε απευθείας στην οθόνη τα αντικείμενα αλλά γίνεται χρήση μιας δεύτερης εικόνας όπου ζωγραφίζουμε εκεί, μετά ζωγραφίζουμε την εικόνα στην οθόνη. Με την τεχνική αυτή αποφεύγουμε την καθυστέρηση που δημιουργείται κατά την ζωγραφική απευθείας στην οθόνη, διαδικασία αργή τόσο ώστε να γίνεται αντιληπτή από το ανθρώπινο μάτι. Η παύση συγχρονισμού είναι απαραίτητη ώστε να υπάρχει μία σταθερή επιθυμητή ταχύτητα στα κινούμενα σχέδια.

Τέλος η κλάση ClipManager χρησιμοποιείται για την αναπαραγωγή ήχων όπου είναι απαραίτητο.



Εικόνα 6-29 - Βρόχος επανάληψης της MainPanel

<b>MainPanel</b>		<i>JPanel</i>
<ul style="list-style-type: none"> <li>- <u>FRAMES_PER_SEC: int = 28 {readOnly}</u></li> <li>- period: long = (long) (1000 / ...</li> <li>- dbg: Graphics</li> <li>- dbImage: BufferedImage</li> <li>- imageBg: Image</li> <li>- blueCardImage: Image</li> <li>- questionCardImage: Image</li> <li>- statusCardImage: Image</li> <li>- greenJokerImage: BufferedImage</li> <li>- genSprite: GeneralSprite ([])</li> <li>- textAnim: TextAnimator</li> <li>- statusCardAnim: StatusCardAnimator</li> <li>- coinAnim: CoinAnimator</li> <li>- totalQCardAnim: TotalQCardAnimator</li> <li>- greenJokerNum: int</li> <li>- panelSize: Dimension</li> <li>- running: boolean</li> <li>- gameEngineOwner: GameEngine</li> <li>- clipManager: ClipManager</li> </ul>		
<ul style="list-style-type: none"> <li>+ MainPanel()</li> <li>+ init() : void</li> <li>+ paintComponent(Graphics) : void</li> <li>- run() : void</li> <li>- animUpdate() : boolean</li> <li>- animRender() : void</li> <li>- animPaint() : void</li> <li>- startShow() : void</li> <li>+ stop() : void</li> <li>+ textAnimate(String, int) : void</li> <li>+ statusCardAnimate(StatusCard) : void</li> <li>+ qCardAnimate(QCard, boolean, String) : void</li> <li>+ coinAnimate() : boolean</li> <li>- drawGreenJokerCard(Graphics) : void</li> <li>+ setGreenJokerNum(int) : void</li> <li>+ gameEngineParams(GameEngine) : void</li> </ul>		

**Εικόνα 6-30 - Κλάση MainPanel**

**Πίνακας 6-46 - Ιδιότητες της MainPanel**

Όνομα	Τύπος	Περιγραφή
FRAMES_PER_SEC	int	Ο αριθμός των εικόνων που εμφανίζονται σε ένα δευτερόλεπτο, ο αριθμός αυτός καθορίζει την ταχύτητα.
period	long	Ο χρόνος που διαρκεί ένας κύκλος, υπολογίζεται σε millisecond με βάση τα FRAMES_PER_SEC.

Όνομα	Τύπος	Περιγραφή
dbg, dblImage	Graphics, BufferedImage	Οι ενδιαμέσες μνήμες γραφικών
greenJokerImage	BufferedImage	Εικόνα των καρτών για τους πράσινους joker, εκεί γράφουμε και το πλήθος τους κάθε στιγμή.
imageBg, blueCardImage, questionCardImage, statusCardImage	Image	Εικόνες που φορτώνονται από αρχεία για το κάθε είδος κάρτας.
genSprite	GeneralSprite []	Πίνακας αντικειμένων GeneralSprite που αντιστοιχεί στα αντικείμενα προς εμφάνιση.
textAnim	TextAnimator	Το animation του κειμένου.
statusCardAnim	StatusCardAnimator	Το animation της κάρτας κατάστασης.
coinAnim	CoinAnimator	Το animation του νομίσματος.
totalQCardAnim	TotalQCardAnimator	Το animation της κάρτας ερώτησης.
greenJokerNum	int	Το πλήθος των πράσινων joker.
panelSize	Dimension	Οι διαστάσεις του πάνελ.
gameEngineOwner	GameEngine	Η μηχανή του παιχνιδιού που ελέγχει το MainPanel.
clipManager	ClipManager	Το αντικείμενο που αναπαράγει τους ήχους.

Πίνακας 6-47 - Μέθοδοι της MainPanel

<b>MainPanel()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Ο δομητής της κλάσης.



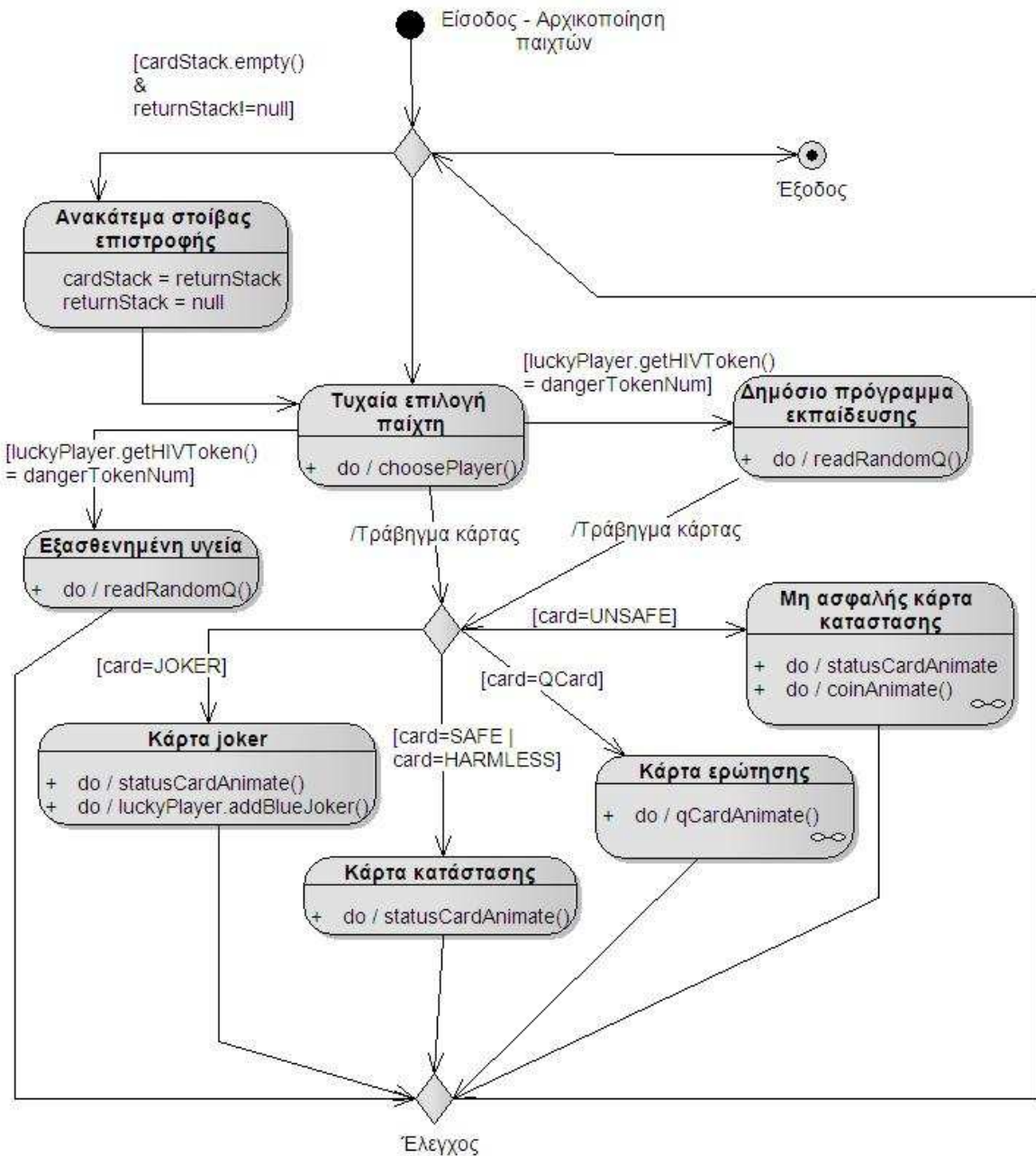
<b>init()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Αρχικοποιεί τα στοιχεία της κλάσης κάθε φορά που ξεκινάει ή φορτώνεται ένα παιχνίδι.
<b>paintComponent(Graphics g)</b>	
Είσοδος	Graphics g - το γραφικό της επιφάνειας του πάνελ
Έξοδος	-
Περιγραφή	Μέθοδος που κληρονομείται από την υπερκλάση JPanel και μας δίνει την δυνατότητα να ζωγραφίσουμε.
<b>run()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Περιέχει τον κύριο βρόχο επανάληψης.
<b>animUpdate()</b>	
Είσοδος	
Έξοδος	boolean - τιμή που επιστρέφεται για το αν θα εκτελεστεί το συνολικό animation
Περιγραφή	Δίνει εντολή στα υπόλοιπα animation να εκτελέσουν τον κώδικά τους και περιμένει απάντηση από αυτά ότι εκτελούνται. Αντιστοίχως αν έστω και ένα animation εκτελείται, απαντά και αυτή θετικά.
<b>animRender()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Ζωγράφισμα στην ενδιάμεση μνήμη των animation.
<b>animPaint()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Ζωγράφισμα της ενδιάμεσης μνήμης στην οθόνη.

<b>startShow(), stop()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Δίνουν αντιστοίχως εντολές έναρξης και τερματισμού.
<b>textAnimate(String text2Anim, int mils)</b>	
Είσοδος	String text2Anim - το κείμενο προς εμφάνιση. int mils - millisecond που παραμένει το κείμενο στην οθόνη.
Έξοδος	-
Περιγραφή	Ξεκινά το animation για την εμφάνιση του κειμένου.
<b>statusCardAnimate(StatusCard card)</b>	
Είσοδος	StatusCard card - κάρτα κατάστασης
Έξοδος	-
Περιγραφή	Ξεκινά το animation για το τράβηγμα μιας κάρτας κατάστασης.
<b>qCardAnimate(QCard card, boolean answer, String playerName)</b>	
Είσοδος	QCard card - κάρτα ερώτησης boolean answer - ορθή απάντηση στην ερώτηση String playerName - το όνομα του παίχτη
Έξοδος	-
Περιγραφή	Ξεκινά το animation για το τράβηγμα μιας κάρτα ερώτησης.
<b>coinAnimate()</b>	
Είσοδος	-
Έξοδος	boolean - το αποτέλεσμα του κέρματος.
Περιγραφή	Ξεκινά το animation με το κέρμα προσώπων να στριφογυρνά και επιστρέφει το αποτέλεσμα του κέρματος.

<b>drawGreenJokerCard(Graphics g)</b>	
Είσοδος	Graphics g - το γραφικό όπου ζωγραφίζονται οι πράσινοι joker
Έξοδος	-
Περιγραφή	Ζωγραφίζει την στοίβα με τους πράσινους joker και γράφει επάνω το τρέχον πλήθος τους.
<b>setGreenJokerNum(int greenJokerNum)</b>	
Είσοδος	int greenJokerNum
Έξοδος	-
Περιγραφή	Ορίζει το πλήθος των πράσινων joker.
<b>gameEngineParams(GameEngine gameEngine)</b>	
Είσοδος	GameEngine gameEngine
Έξοδος	-
Περιγραφή	Ορίζει την GameEngine που ελέγχει την MainPanel εκείνη τη στιγμή.

#### 6.1.4.13 Κλάση GameEngine

Η τελευταία και σημαντικότερη κλάση είναι η GameEngine και αυτό γιατί ελέγχει τη ροή του παιχνιδιού και δίνει τις απαραίτητες εντολές στα γραφικά για το τι θα παρουσιαστεί στο χρήστη. Βασική της μέθοδος είναι run(), η οποία ξεκινάει την εκτέλεσή της όταν ξεκινάει νέο ή φορτώνεται ένα παλιό παιχνίδι και τερματίζεται με τον τερματισμό του παιχνιδιού. Η run() αποτελείται από έναν κεντρικό βρόχο επανάληψης, κάθε επανάληψη του βρόχου αντιστοιχεί σε κάθε γύρο του παιχνιδιού. Το διάγραμμα καταστάσεων στην Εικόνα 6-31 μας δείχνει την εκτέλεσή της.



Εικόνα 6-31 - Διάγραμμα κατάστασης βρόχου επανάληψης GameEngine

Πίνακας 6-48 - Ιδιότητες της GameEngine

Όνομα	Τύπος	Περιγραφή
players	GamePlayer[]	Ο πίνακας με τους παίχτες του παιχνιδιού.
frame	SafeFrame	Το κεντρικό παράθυρο γραφικών.
greenJokerNum	int	Το πλήθος των πράσινων joker κάθε στιγμή.

Όνομα	Τύπος	Περιγραφή
dangerUnSafeCardNum, dangerTokenNum	int	Τα όρια των μη ασφαλών καρτών και HIV Token.
cardStack	Stack	Η κύρια στοίβα καρτών.
returnStack	Stack	Η στοίβα επιστροφής.
bookQuestions	String []	Πίνακας με τις ερωτήσεις βιβλίου.
running	boolean	Δείχνει αν εκτελείται το παιχνίδι.
pause	boolean	Δείχνει αν βρίσκεται σε κατάσταση παύσης το παιχνίδι.

Πίνακας 6-49 - Μέθοδοι της GameEngine

<b>GameEngine(DataPack data, SafeFrame frame)</b>	
Είσοδος	DataPack data - όλες οι παράμετροι του παιχνιδιού είτε πρόκειται για νέο είτε αποθηκευμένο. SafeFrame frame - το κεντρικό παράθυρο γραφικών.
Έξοδος	-
Περιγραφή	Ο δομητής της κλάσης.
<b>run()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Η μέθοδος που ελέγχει το παιχνίδι στο σύνολό του.
<b>stopGame()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Προκαλεί τερματισμό του παιχνιδιού. Γίνεται χρήση αυτής της μεθόδου όταν θέλουμε να σταματήσουμε το παιχνίδι γιατί θέλουμε να φορτώσουμε ένα αποθηκευμένο.
<b>pauseGame(boolean pause)</b>	
Είσοδος	boolean pause
Έξοδος	-
Περιγραφή	Προκαλείται παύση ή συνέχιση του παιχνιδιού, ανάλογα με την παράμετρο.

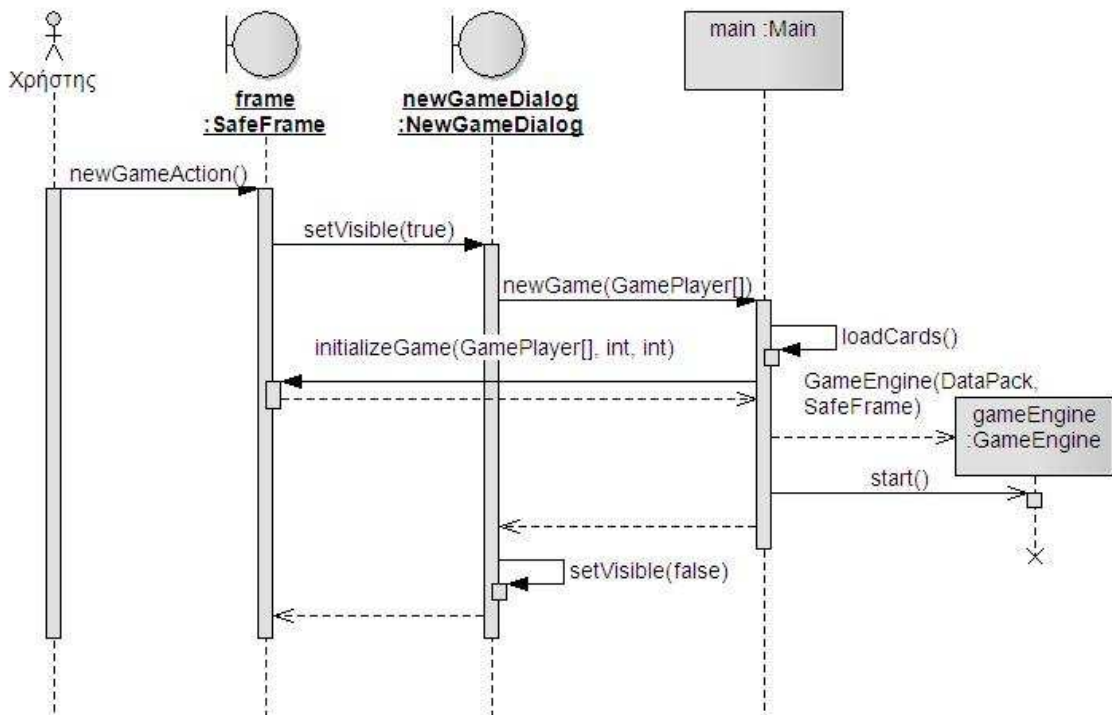
<b>pauseTime(int mills)</b>	
Είσοδος	int mills
Έξοδος	-
Περιγραφή	Βοηθητική μέθοδος για να προκαλέσει παύση του παιχνιδιού κατά mills millisecond.
<b>isRunning(), isPaused()</b>	
Είσοδος	-
Έξοδος	boolean
Περιγραφή	Επιστρέφουν η πρώτη αν το παιχνίδι εκτελείται, η δεύτερη αν βρίσκεται σε παύση.
<b>choosePlayer()</b>	
Είσοδος	-
Έξοδος	int
Περιγραφή	Κινεί την διαδικασία τυχαίας επιλογής παίχτη και επιστρέφει τον παίχτη που επιλέχθηκε.
<b>readRandomQ()</b>	
Είσοδος	-
Έξοδος	-
Περιγραφή	Ξεκινάει την διαδικασία διαβάσματος μιας τυχαίας ερώτησης βιβλίου.
<b>areAllPlayersDead()</b>	
Είσοδος	-
Έξοδος	boolean
Περιγραφή	Ελέγχει αν όλοι οι παίχτες έχουν εξασθενημένη υγεία.
<b>DataPack getData()</b>	
Είσοδος	-
Έξοδος	DataPack
Περιγραφή	Επιστρέφει όλες τις παραμέτρους του παιχνιδιού εκείνη τη στιγμή, χρησιμοποιείται για την αποθήκευση ενός παιχνιδιού.

### 6.1.5 Σχεδίαση περιπτώσεων χρήσης

Στη σχεδίαση περιπτώσεων χρήσης παρουσιάζεται ο τρόπος αντιμετώπισης κάθε περίπτωσης χρήσης κάνοντας χρήση των κλάσεων που παρουσιάστηκαν με την συνοδεία των απαραίτητων διαγραμμάτων.

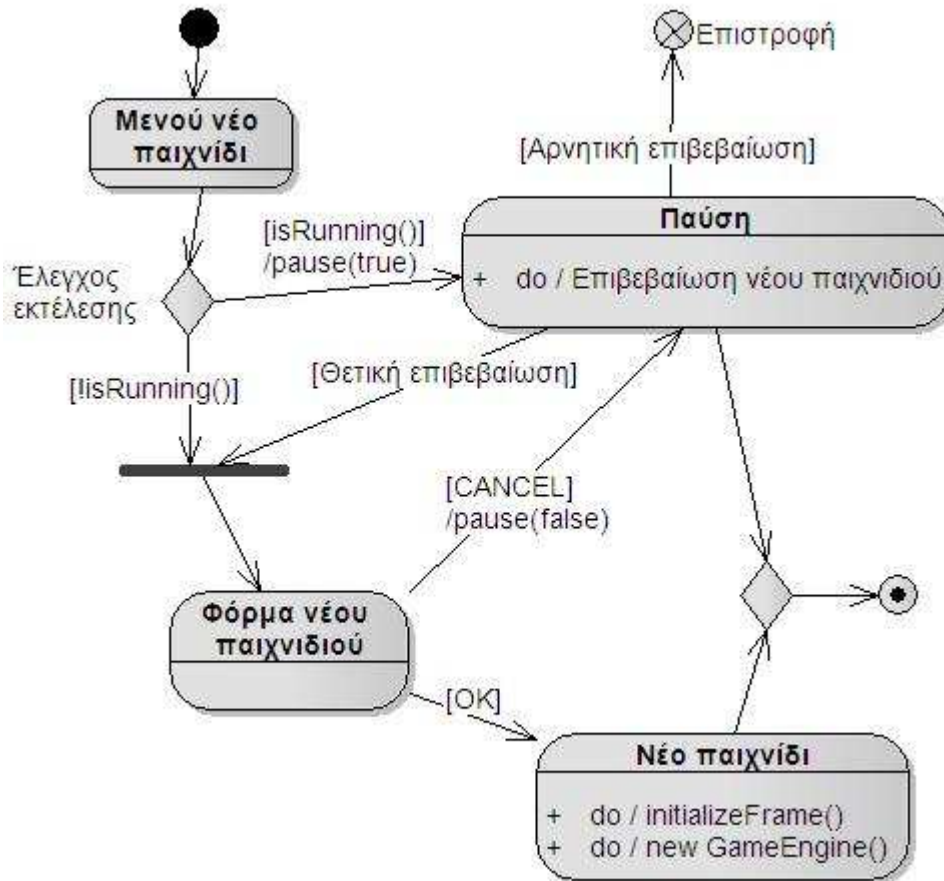
#### 6.1.5.1 Περίπτωση Χρήσης : Νέο παιχνίδι

Ο χρήστης από το μενού επιλέγει «Νέο παιχνίδι», το παράθυρο διαλόγου εμφανίζεται και συμπληρώνει τα στοιχεία των παιχτών που θα παίξουν.



Εικόνα 6-32 - Διάγραμμα ακολουθίας νέου παιχνιδιού

Μετά την επιλογή του μενού γίνεται έλεγχος αν εκτελείται κάποιο παιχνίδι, σε περίπτωση εκτέλεσης ζητείται επιβεβαίωση για το νέο παιχνίδι.

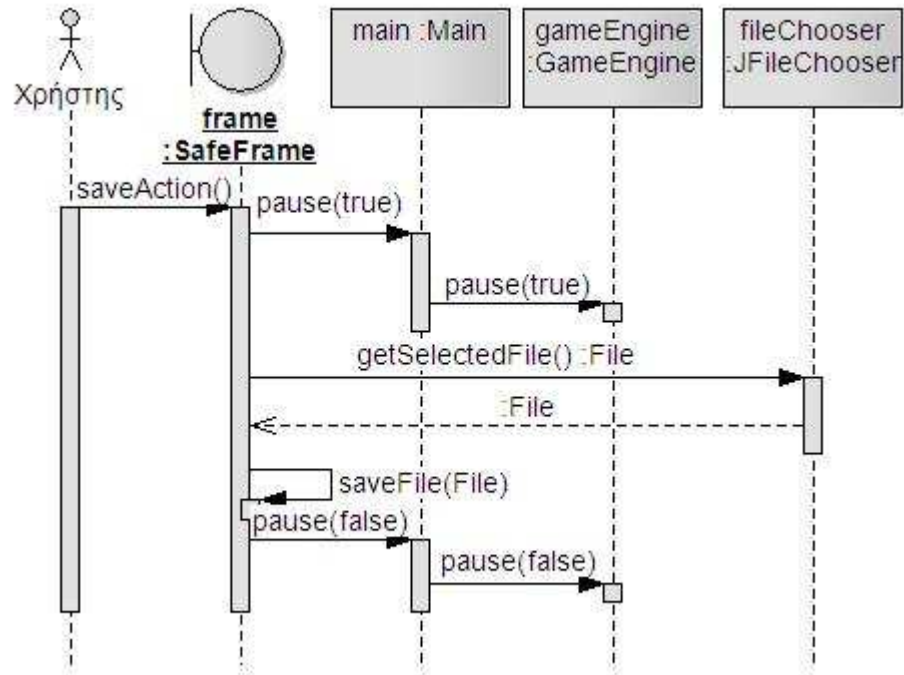


Εικόνα 6-33 - Διάγραμμα καταστάσεων νέου παιχνιδιού

### 6.1.5.2 Περίπτωση Χρήσης : Αποθήκευση παιχνιδιού

Από το μενού ο χρήστης επιλέγει «Αποθήκευση παιχνιδιού», σε περίπτωση που δεν παίζει κάποιο παιχνίδι το μενού παραμένει απενεργοποιημένο. Η επιλογή του ονόματος γίνεται με μία βοηθητική κλάση της java την JFileChooser, η οποία εμφανίζει ένα παράθυρο διαλόγου για την επιλογή αρχείου.

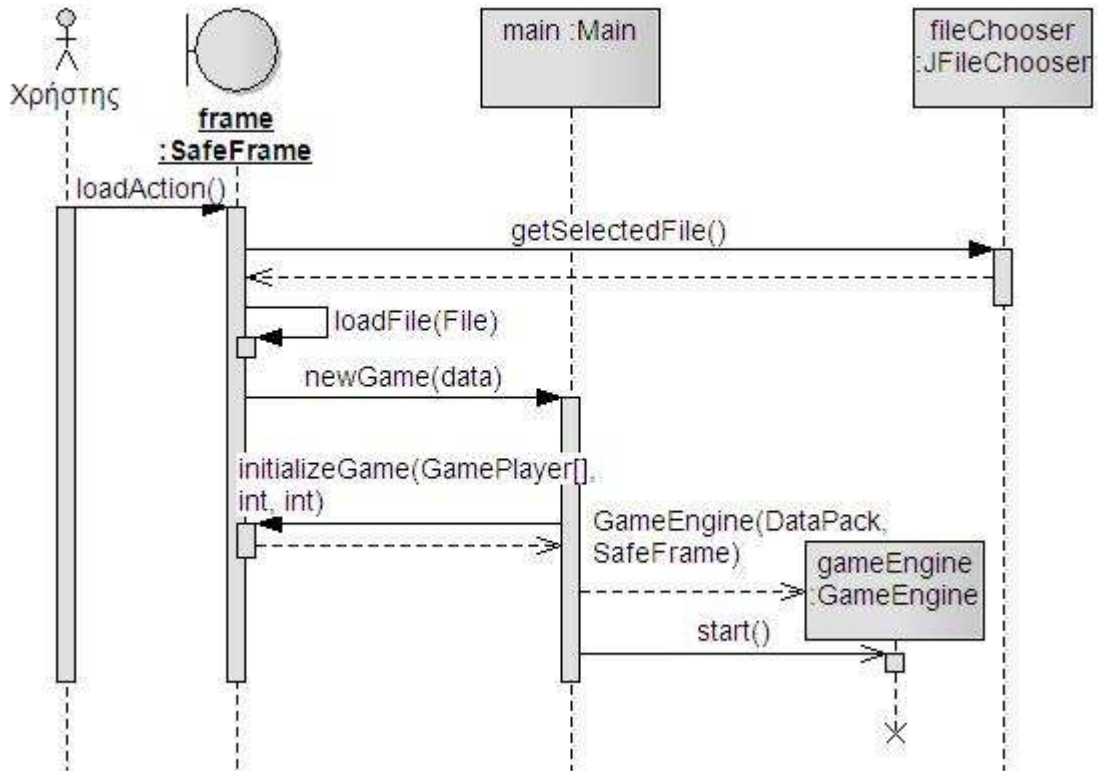




Εικόνα 6-34 - Διάγραμμα ακολουθίας αποθήκευσης παιχνιδιού

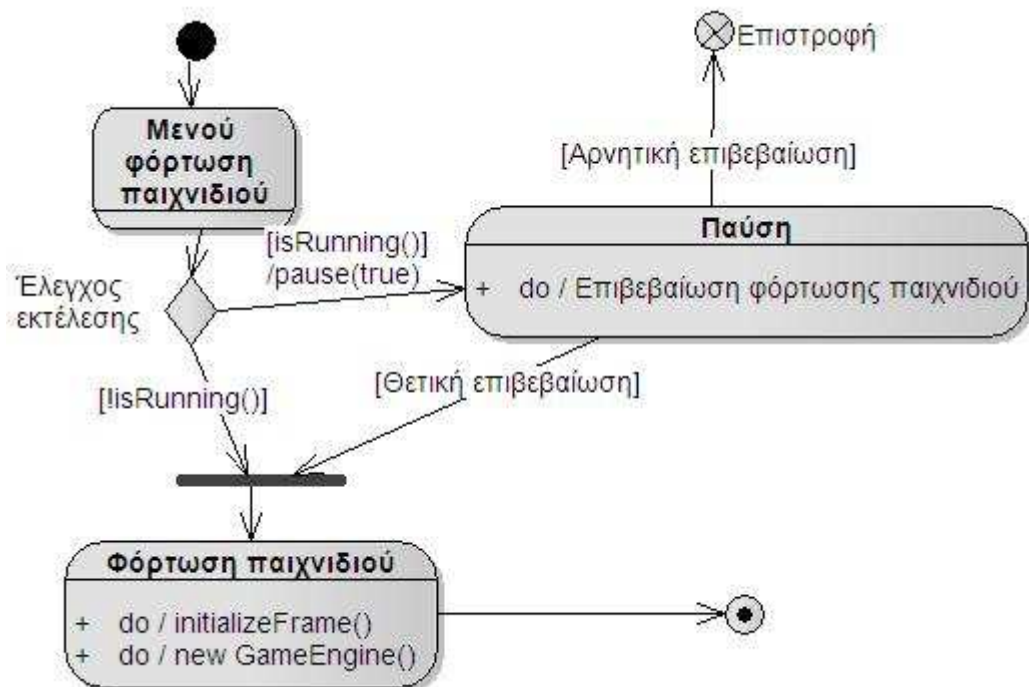
### 6.1.5.3 Περίπτωση Χρήσης : Φόρτωση παιχνιδιού

Ομοίως με τις προηγούμενες περιπτώσεις από το μενού «Αποθηκευμένα παιχνίδια», ο χρήστης επιλέγει να επιλέγει να επαναφέρει ένα αποθηκευμένο παιχνίδι. Η επιλογή του αρχείου γίνεται και πάλι κάνοντας χρήση της βοηθητικής κλάσης JFileChooser.



Εικόνα 6-35 - Διάγραμμα ακολουθίας φόρτωσης παιχνιδιού

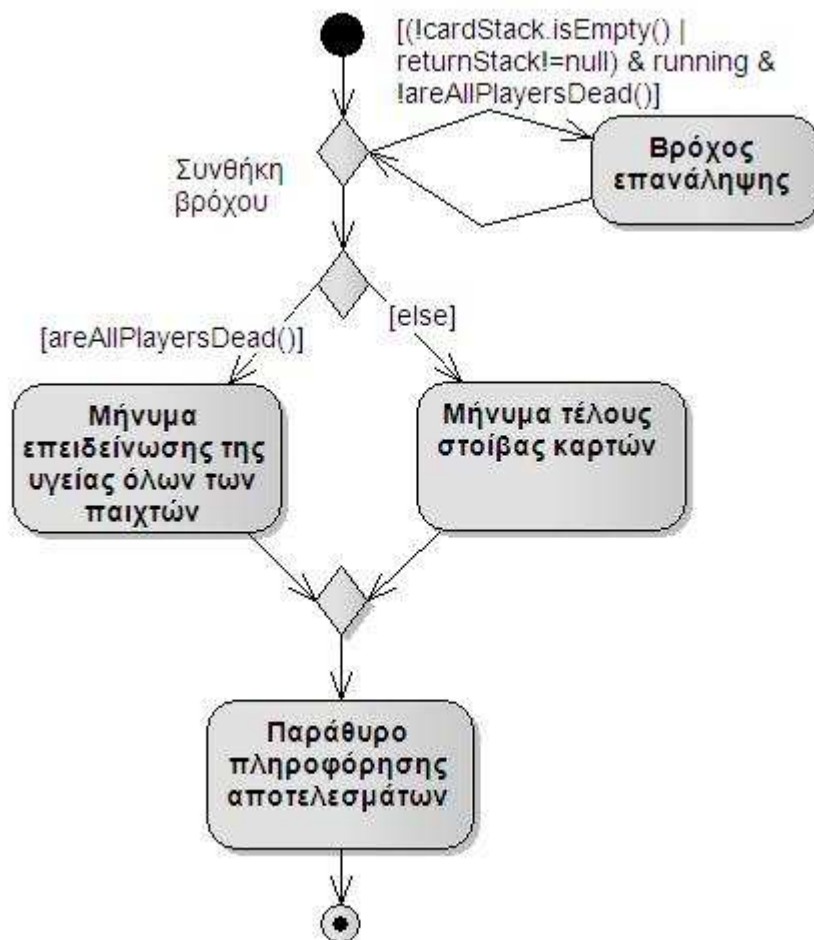
Και σε αυτή την περίπτωση γίνεται ο απαραίτητος έλεγχος για την εκτέλεση παιχνιδιού και την επιβεβαίωση της αντικατάστασης του υπάρχοντος παιχνιδιού.



Εικόνα 6-36 - Διάγραμμα καταστάσεων φόρτωσης παιχνιδιού

#### 6.1.5.4 Περίπτωση Χρήσης : Τέλος παιχνιδιού

Το τέλος του παιχνιδιού φτάνει όταν ικανοποιηθούν ορισμένες συνθήκες κατά τη διάρκεια του παιχνιδιού, αυτές είναι όταν έχει επιδεινωθεί η υγεία όλων των παιχτών ή έχει τελειώσει για δεύτερη φορά η στοίβα με τις μπλε κάρτες. Ο έλεγχος των συνθηκών γίνεται κατά το τέλος κάθε γύρου. Η κλάση GameEngine που ελέγχει όλη τη ροή του παιχνιδιού είναι αυτή που θα αποφασίσει το τέλος του παιχνιδιού. Συγκεκριμένα στην μέθοδο run() εκτελείται ένας βρόχος επανάληψης που αντιστοιχεί σε κάθε γύρο του παιχνιδιού. Η συνθήκη επανάληψης του βρόχου είναι αυτή που ορίζει και το τέλος του παιχνιδιού. Με το τέλος του παιχνιδιού ένα παράθυρο διαλόγου πληροφορεί τους παίκτες για τα αποτελέσματα.



Εικόνα 6-37- Διάγραμμα δραστηριοτήτων τέλους παιχνιδιού

Το παράθυρο διαλόγου περιγράφεται από την κλάση GameOverDialog, όπου και δίνεται ως παράμετρος ο πίνακας με των παιχτών ώστε να εμφανιστεί η τελική κατάταξη.

#### **6.1.5.5 Περίπτωση Χρήσης : Επιλογή παίχτη**

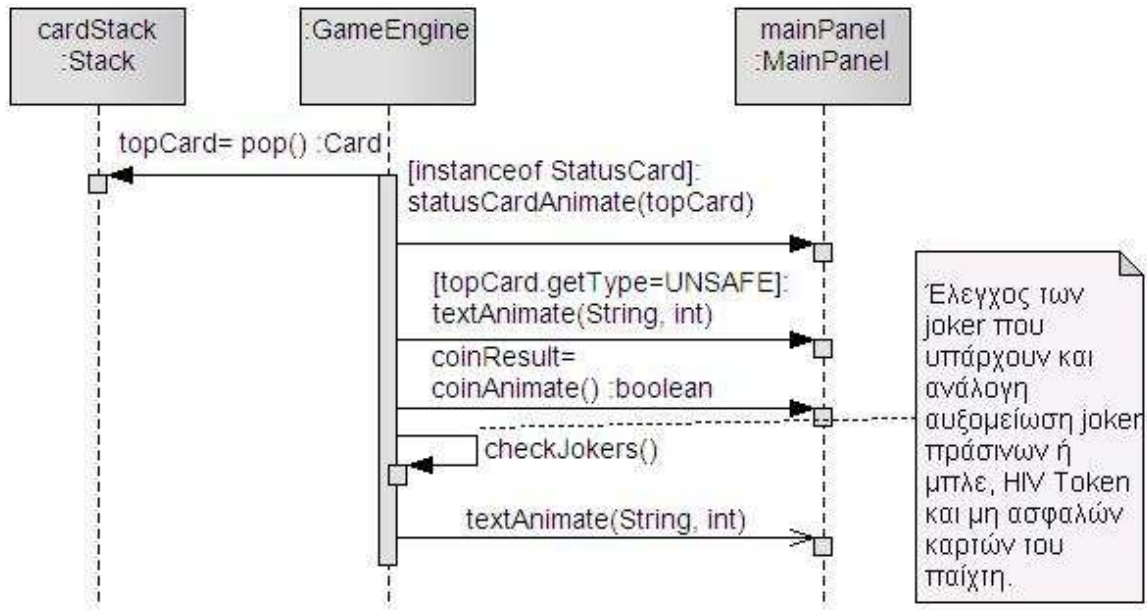
Η επιλογή παίχτη γίνεται στην αρχή κάθε γύρου και ξεκινάει όταν δώσει εντολή μέσα από την μέθοδο `run()`, η `GameEngine`. Η επιλογή του παίχτη γίνεται αντιληπτή στο χρήστη μέσω της αλλαγής χρώματος στο φόντο του πάνελ του κάθε παίχτη, έτσι με κυκλικό τρόπο διαδοχικά το φόντο του πάνελ σε κάθε παίχτη αλλάζει και επαναφέρεται στο προηγούμενο δημιουργώντας ένα εφέ. Το εφέ υλοποιείται στη μέθοδο `choosePlayer()` της `SafeFrame`.

#### **6.1.5.6 Περίπτωση Χρήσης : Τράβηγμα κάρτας**

Το τράβηγμα κάρτας από τη στοίβα με τις μπλε κάρτες απευθύνεται κυρίως στο κομμάτι των γραφικών. Η κλάση `GameEngine` κατά τη διάρκεια της εκτέλεσης της επιλέγει να πάρει το πρώτο αντικείμενο `Card` από την συλλογή που κρατά, την `Stack cardStack`. Έπειτα δίνει εντολή στην `MainPanel` να εμφανίσει την κάρτα που έλαβε. Η συγκεκριμένη περίπτωση χρήσης αποτελεί γενίκευση των περιπτώσεων χρήσης «Μη ασφαλής κάρτα κατάστασης», «Κάρτα ερώτησης» και «Λοιπές κάρτες κατάστασης και joker». Και στις τρεις περιπτώσεις χρήσης το `animation` για την εμφάνιση της κάρτας είναι ίδιο και υλοποιείται από την κλάση `CardAnimator`.

#### **6.1.5.7 Περίπτωση Χρήσης : Μη ασφαλής κάρτα κατάστασης**

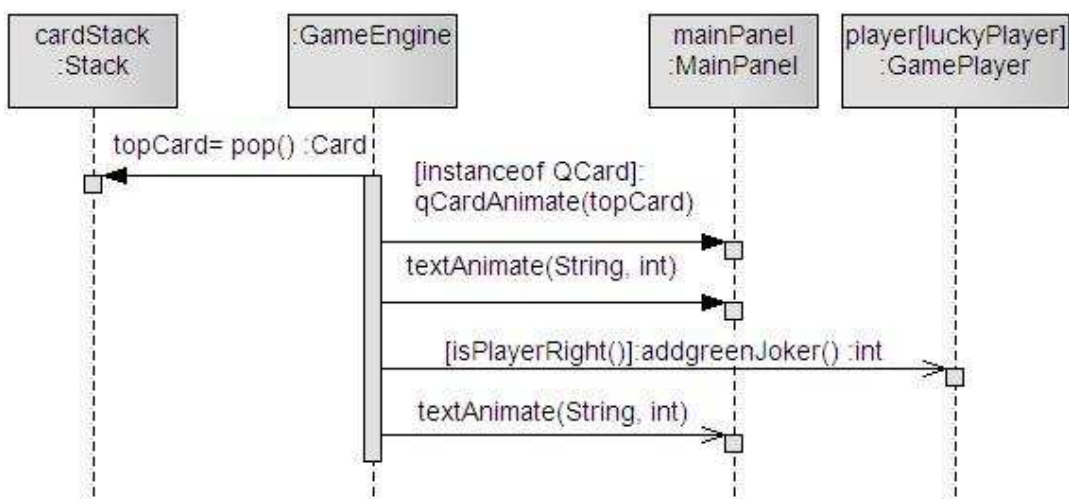
Όταν η κάρτα που έχει τραβηχτεί από την στοίβα περιέχει μία μη ασφαλής κάρτα κατάστασης, ενημερώνουμε τον χρήστη για το είδος της κάρτας που τραβήχτηκε, όπως για την ρίψη του νομίσματος που ακολουθεί. Η Εικόνα 6-38 μας δείχνει τις ενέργειες που εκτελούνται ενώ η Εικόνα 4-2 που παρουσιάστηκε στις απαιτήσεις μας δείχνει τον τρόπο ανταλλαγής `joker`, `HIV Token` και μη ασφαλών καρτών, ανάλογα με το αποτέλεσμα του νομίσματος.



Εικόνα 6-38 - Διάγραμμα ακολουθίας περίπτωσης χρήσης μη ασφαλούς κάρτας

### 6.1.5.8 Περίπτωση Χρήσης : Κάρτα ερώτησης

Η GameEngine όταν δει πως η κάρτα που επιλέχθηκε είναι κάρτα ερώτησης δίνει εντολή να εκτελεστεί το ανάλογο animation. Το animation μιας κάρτας ερώτησης εκτελείται από την κλάση QCardAnimator που περιέχει την σειρά γεγονότων του τραβήγματος της κάρτας και της παρουσίασης της ερώτησης, την απάντηση του χρήστη, την καταγραφή της απάντησης και τέλος τον έλεγχο της απάντησής του.



Εικόνα 6-39- Διάγραμμα ακολουθίας κάρτας ερώτησης

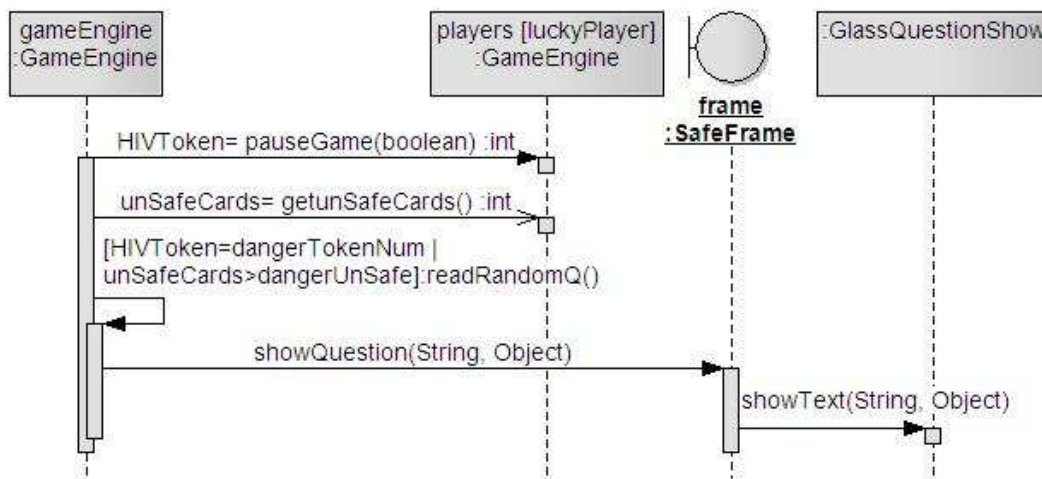
### 6.1.5.9 Περίπτωση Χρήσης : Λοιπές κάρτες κατάστασης και joker

Οι υπόλοιπες κάρτες κατάστασης δηλαδή οι μη επικίνδυνες και ασφαλείς κάρτες , όπως και οι κάρτες joker δεν έχουν κάποια ιδιαιτερότητα. Ο χρήστης όταν τραβήξει

μία τέτοια κάρτα απλά ενημερώνεται για το περιεχόμενό της. Η ενημέρωση γίνεται με την κλάση `StatusCardAnimator`, όμοια με την περίπτωση των μη ασφαλών καρτών.

#### 6.1.5.10 Περίπτωση Χρήσης : Διάβασμα ερώτησης βιβλίου

Ένας παίχτης διαβάζει μία ερώτηση βιβλίου όταν η υγεία του έχει εξασθενήσει (όριο HIV Token) ή όταν παρακολουθεί πρόγραμμα δημόσιας εκπαίδευσης (όριο μη ασφαλών καρτών). Στην αρχή του κάθε γύρου και αφού έχει επιλεγθεί κάποιος παίχτης, η `GameEngine` ελέγχει την κατάσταση του και αναλόγως στέλνει μήνυμα στην `SafeFrame` και κατ' επέκταση στην `GlassQuestionShow`. Η `GlassQuestionShow` καλύπτει όλη την επιφάνεια του παραθύρου με την ερώτηση, ενώ παράλληλα προκαλεί παύση του παιχνιδιού.

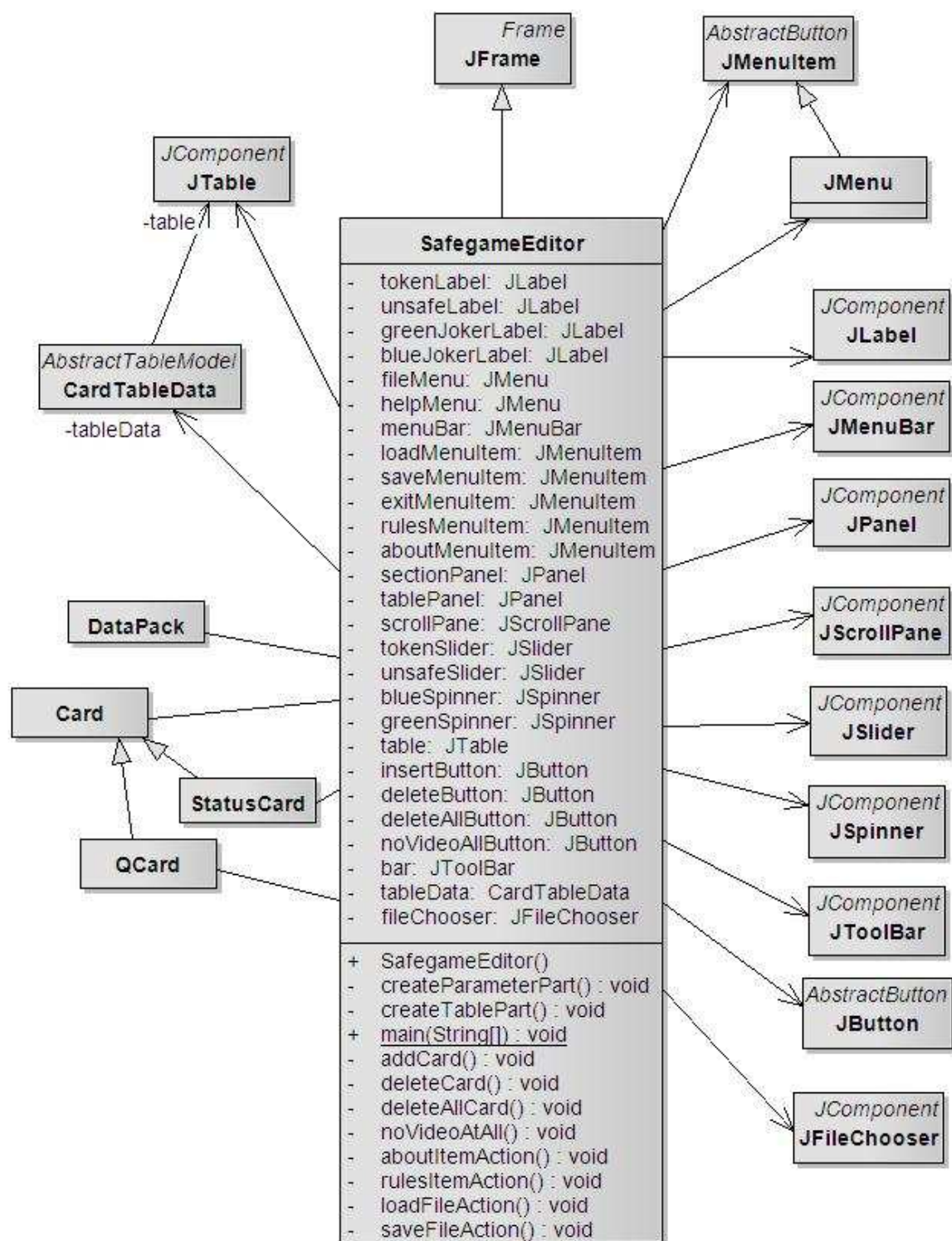


Εικόνα 6-40 - Διάγραμμα ακολουθίας ερώτησης βιβλίου

Για να προσθέσουμε μεγαλύτερη ευελιξία γίνεται εκμετάλλευση της δυνατότητας της `java` στο να διαβάζει κώδικα `HTML`. Συγκεκριμένα η κλάση `JLabel`, μπορεί να εμφανίσει την μορφοποίηση της `HTML` γλώσσας, έτσι ο συντάκτης της εκάστοτε ερώτησης βιβλίου, μπορεί να της δώσει την μορφή που επιθυμεί.

## 6.2 Σχεδίαση του συντάκτη `SafeGameEditor`

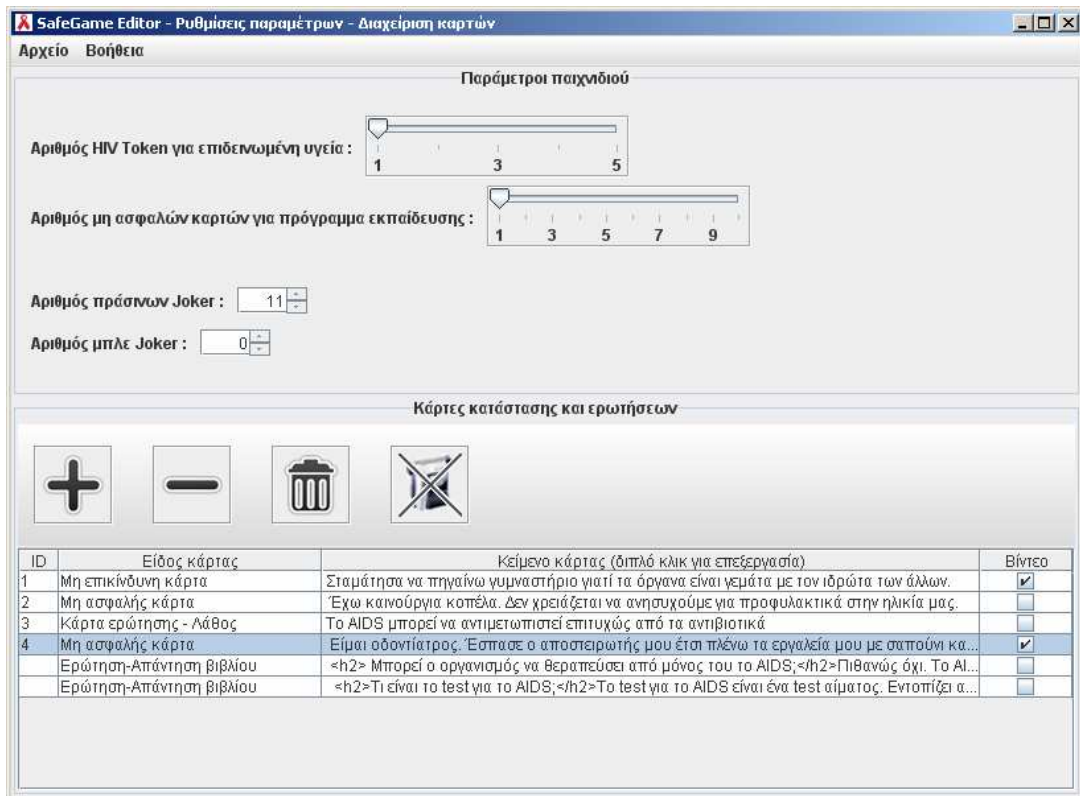
Ο συντάκτης δεν αποτελεί παρά μία φόρμα που αποθηκεύει τα στοιχεία που συλλέγει στο αρχείο ρυθμίσεων. Η δομή του είναι πολύ απλή, μία κύρια κλάση (`SafeGameEditor`) που φέρει τη φόρμα των στοιχείων και είναι το κυρίως εκτελέσιμο, ενώ ως βοηθητικές χρησιμοποιούνται οι κλάσεις του `SafeGame` για την αποθήκευση των στοιχείων.



Εικόνα 6-41 - Διάγραμμα κλάσεων SafeGameEditor

Όπως βλέπουμε και στην Εικόνα 6-41, οι κλάσεις Card, QCard, StatusCard και DataPack είναι κλάσεις του SafeGame και εισάγονται ως βιβλιοθήκες. Η κλάση SafeGameEditor κληρονομεί την JFrame καθώς η ίδια αποτελεί το παράθυρο που εμφανίζεται. Η κλάση CardTableData χρησιμεύει στην παραμετροποίηση των

λειτουργιών του πίνακα που με τις κάρτες. Η Εικόνα 6-42 μας δίνει ένα δείγμα της SafeGameEditor.



Εικόνα 6-42 - Παράθυρο SafeGameEditor

Το μενού Αρχείο έχει τρεις επιλογές :

- Φόρτωση αρχείου - φορτώνει από τον δίσκο το αρχείο που επιλέγεται.
- Αποθήκευση αρχείου - αποθηκεύει στο αρχείου που επιλέγεται τις πληροφορίες που βρίσκονται στην φόρμα.
- Έξοδος

Το μενού Βοήθεια παρέχει την απαραίτητη βοήθεια.

Όλο το ενδιαφέρον επικεντρώνεται στον πίνακα ο οποίος μπορεί να φιλοξενήσει κάρτες κατάστασης, κάρτες ερωτήσεων αλλά και ερωτήσεις-απαντήσεις βιβλίου. Οι κάρτες κατάστασης και ερωτήσεων έχουν ένα μοναδικό αριθμό που τους χαρακτηρίζει και συμπληρώνεται αυτόματα από το σύστημα, ενώ μπορούν να είναι συσχετισμένες με κάποιο βίντεο. Η συσχέτιση με το βίντεο γίνεται με το να βάζει κάποιος ή όχι σημάδι στο ανάλογο κουτάκι. Όταν ο χρήστης επιλέγει για το είδος της κάρτας τότε ένα πτυσσόμενο μενού εμφανίζεται και του δίνει την δυνατότητα να επιλέξει. Για να δημιουργηθεί το μοντέλο του πίνακα η κλάση CardTableData



κληρονομεί την αφηρημένη κλάση `AbstractTableModel` της `java` και υλοποιεί τις μεθόδους της.

## 7. Υλοποίηση

Στο παρόν κεφάλαιο θα αναφερθούν ενδεικτικά κομμάτια υλοποίησης που παρουσιάζουν ενδιαφέρον. Όπως και τα προηγούμενα δύο κεφάλαια θα χωριστεί δύο μέρη, για το SafeGame και για τον SafeGameEditor.

### 7.1 Υλοποίηση του SafeGame

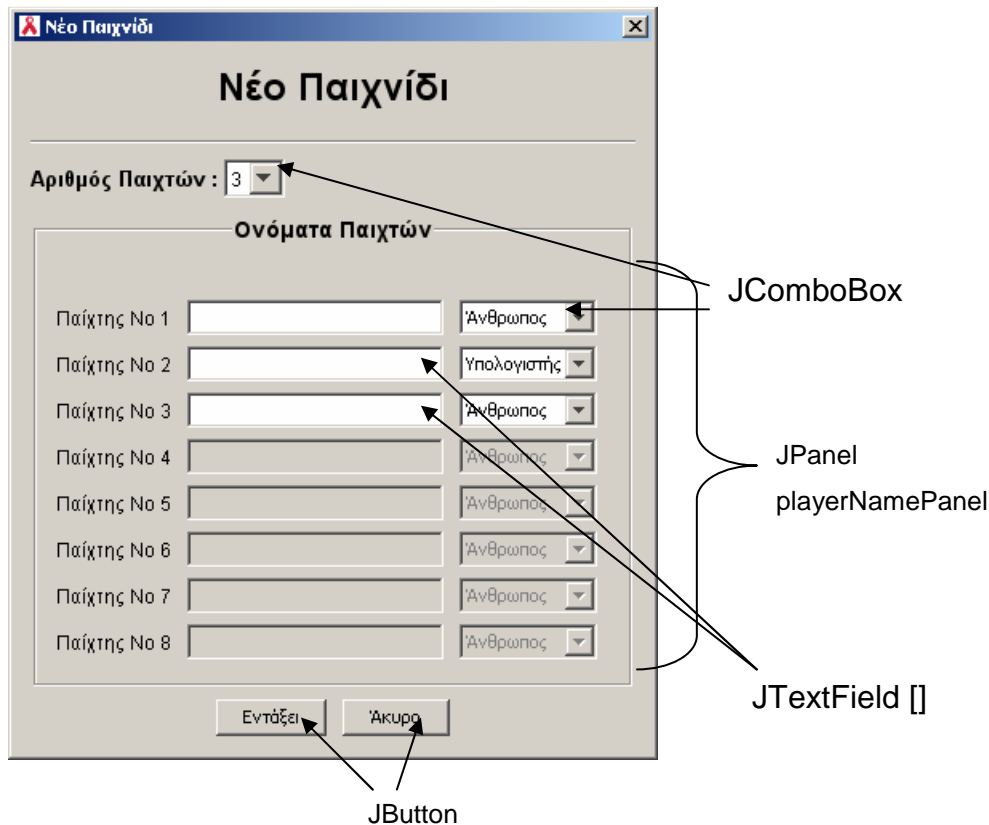
Όλες οι κλάσεις που αναφέρθηκαν στην σχεδίαση αντιπροσωπεύονται από ένα αρχείο java, όλα τα αρχεία java καθώς και οι απαραίτητες εικόνες πακετάρονται σε ένα αρχείο jar. Στον ίδιο κατάλογο που βρίσκεται το jar, υπάρχει το αρχείο των ρυθμίσεων με το όνομα *settings*, και δύο κατάλογοι. Ο πρώτος έχει το όνομα *lib* και περιέχει την βιβλιοθήκη *jmfl.jar*, απαραίτητη για την αναπαραγωγή βίντεο, ενώ ο δεύτερος κατάλογος φέρει το όνομα *video* και περιέχει τα βίντεο των καρτών που ενδεχομένως να υπάρχουν. Το όνομα του κάθε βίντεο που βρίσκεται στον κατάλογο πρέπει να είναι ίδιο με το id που δόθηκε στην αντίστοιχη κάρτα, π.χ. η κάρτα με id 100 αντιστοιχεί στο βίντεο με το όνομα *100.mov*. Παρακάτω παρουσιάζονται κλάσεις που παρουσιάζουν ενδιαφέρον.

#### 7.1.1 Κλάση NewGameDialog

Η κλάση *NewGameDialog* περιέχει την φόρμα που συμπληρώνει ο παίχτης για να ξεκινήσει ένα νέο παιχνίδι, αποτελεί παράδειγμα χρήσης στοιχείων του *swing*. Η όψη του παραθύρου φαίνεται στην Εικόνα 7-1.

Η κλάση κληρονομεί την *JDialog* που χρησιμοποιείται για παράθυρα διαλόγου. Στον δομητή της κλάσης δίνουμε ως παράμετρο την κύρια μέθοδο *Main*, έτσι ώστε όταν πατηθεί το κουμπί «Εντάξει», να σταλθεί μήνυμα έναρξης νέου παιχνιδιού στην *Main*.

Στο δομητή, αρχικοποιούνται τα απαραίτητα συστατικά του παραθύρου και διατάσσονται στον χώρο με βάση το *layout* που χρησιμοποιείται. Στην συγκεκριμένη κλάση χρησιμοποιείται το *SpringLayout* για το πάνελ με τα ονόματα των παιχτών και το *GroupLayout* για την διάταξη όλου του παραθύρου.



Εικόνα 7-1 - Παράθυρο νέου παιχνιδιού

Επιπλέον στο πεδίο εισαγωγής ονόματος του παίχτη θέλουμε να υπάρχει ένα όριο στους χαρακτήρες εισαγωγής, για αυτό το σκοπό γίνεται χρήση ενός φίλτρου στα πεδία των ονομάτων. Μία επιπλέον κλάση δημιουργείται που κληρονομεί την `DocumentFilter` και συνδέεται με το πεδίο του ονόματος.

### 7.1.2 Κλάση `MainPanel` και κλάσεις `animation`

Το χαρακτηριστικό της `MainPanel` είναι ότι αποτελεί το χώρο όπου συμβαίνουν τα `animation`. Η `MainPanel` κληρονομεί την `JPanel` και υπερβαίνει την μέθοδο `paintComponent`, ζωγραφίζοντας στο χώρο της τα αντικείμενα που επιθυμεί.

#### Κώδικας 7-1 - Κλάση `MainPanel`, μέθοδος `paintComponent()`

```
public void paintComponent(Graphics g) {
    super.paintComponent(g);
    if (imageBg != null) {
        g.drawImage(imageBg, 0, 0, this.getWidth(), this.getHeight(), this);
    }
    drawGreenJokerCard(g);
    for (int i = 0; i < genSprite.length; i++) {
        genSprite[i].show(g);
    }
}
```

Κάθε animation εκτελείται σε μία μέθοδο που περιέχει ένα βρόχο επανάληψης με το όνομα run(), η μέθοδος φροντίζει να ζωγραφίζεται στο πάνελ κάθε animation καλώντας τρεις μεθόδους και έπειτα πραγματοποιώντας μία παύση.

#### Κώδικας 7-2 - Κλάση MainPanel μέθοδος run()

```
private void run() {
    long beforeTime, afterTime, timeDiff, sleepTime;
    long overSleepTime = 0L;
    int noDelays = 0;
    long excess = 0L;
    //Παίρνουμε τον χρόνο πριν μπούμε στο loop
    beforeTime = System.nanoTime();
    running = true;
    while (running) {
        if (!animUpdate()) // Αν το update αποφασίσει αρνητικά τότε δε
        //χρειάζεται να πάμε παρακάτω
        {
            continue;
        }
        animRender();
        animPaint();
        afterTime = System.nanoTime(); //Αφού εκτελέστηκαν οι ενέργειες
        timeDiff = afterTime - beforeTime;
        sleepTime = (period - timeDiff) - overSleepTime; //Ο χρόνος
        //καθυστερήσης
        if (sleepTime > 0) { // υπολοιπόμενος χρόνος στον κύκλο
            try {
                Thread.sleep(sleepTime / 1000000L); // nano -> ms
            } catch (InterruptedException ex) {
            }

            overSleepTime = (System.nanoTime() - afterTime) - sleepTime;
        } else { // sleepTime <= 0; καθυστερήσαμε περισσότερο από όσο θα
        //έπρεπε
            excess -= sleepTime; // αποθηκεύουμε το χρόνο υπέρβασης
            overSleepTime = 0L;
            if (++noDelays >= NO_DELAYS_PER_YIELD) {
                Thread.yield(); //Δίνουμε χρόνο και σε άλλο thread να
                //τρέξει
                noDelays = 0;
            }
        }

        beforeTime = System.nanoTime();
        //Σε περίπτωση που καθυστερεί πολύ το animation, παραλείπουμε
        //κάποιους κύκλους με ζωγραφική κάνοντας μόνο animUpdate()
        int skips = 0;
        while ((excess > period) && (skips < MAX_FRAME_SKIPS)) {
            excess -= period;
            animUpdate(); // update αλλά όχι render
            skips++;
        }
        excess = 0;
    } //while

    repaint();
} //run
```

Όπως φαίνεται και στον Κώδικας 7-2 γίνεται κλήση στις μεθόδους animUpdate(), animRender() και animPaint(). Η animUpdate() αποφασίζει αν θα εκτελεστεί το animation και αναλόγως καλεί τις animUpdate των GeneralSprite, δηλαδή του κάθε animation (π.χ. κέρμα, κάρτα κλπ). Η animRender() φροντίζει να ζωγραφίζει

στην εικόνα που χρησιμοποιείται ως ενδιάμεση μνήμη, δίνει επίσης εντολή στα GeneralSprite να ζωγραφίσουν στην ίδια εικόνα κάθε ένα το κομμάτι του. Τέλος η animPaint() ζωγραφίζει την ενδιάμεση μνήμη στην οθόνη. Ο διαχωρισμός μεταξύ update και render γίνεται γιατί μερικές φορές όταν καθυστερεί το animation, απαιτείται να μην εμφανιστούν μερικοί κύκλοι με αποτέλεσμα να κάνουμε μόνο το update.

Οι κλάσεις για το animation υλοποιούν όλες την διεπαφή GeneralSprite που περιέχει όλες τις απαραίτητες μεθόδους για να εμφανιστεί ένα animation.

#### Κώδικας 7-3 - Διεπαφή GeneralSprite

```
public interface GeneralSprite {
    public abstract boolean animUpdate();
    public abstract void show(Graphics g);
    public void start();
    public void stop();
    public boolean isActive();
}
```

#### 7.1.3 Κλάση VideoDialog

Η κλάση VideoDialog εμφανίζει ένα παράθυρο στο οποίο αναπαράγεται ένα βίντεο με τη βοήθεια του Java Media Framework (JMF) . Ενώ συνήθως απαιτείται εγκατάσταση του JMF, λόγω των περιορισμένων απαιτήσεων μας έγινε ενσωμάτωση του jmf.jar που περιέχει βασικές κλάσεις για την αναπαραγωγή βίντεο, σε έναν ξεχωριστό κατάλογο με το όνομα lib. Έτσι πλέον δεν είναι απαραίτητο για τον τελικό χρήστη να εγκαταστήσει ξεχωριστά το JMF.

Η κλάση VideoDialog υλοποιεί δύο διεπαφές, μία τον χειρισμό των βίντεο με το όνομα ControllerListener (javax.media.ControllerListener) και μία για τον χειρισμό του παραθύρου με το όνομα WindowListener (java.awt.event.WindowListener). Το βίντεο αναπαράγεται με τη βοήθεια μιας κλάσης με το όνομα Player (javax.media.Player) και την υλοποίηση της μεθόδου controllerUpdate().

#### Κώδικας 7-4 - Κλάση VideoDialog, μέθοδος controllerUpdate()

```
public synchronized void controllerUpdate(ControllerEvent event) {
    if (player == null) {
        return;
    }
    //Όταν έχει τελειώσει το realize event μπορούμε πλέον να πάρουμε τα
    Components ώστε να δούμε το video
    if (event instanceof RealizeCompleteEvent) {
        if ((visualComponent =
            player.getVisualComponent()) != null) {
            setTitle("Video");
        }
    }
}
```

```

        panel.add(visualComponent);
        panel.add(Box.createRigidArea(new Dimension(0, 10)));
        panel.add(playButton);
        //To repaint()χρειάζεται όταν ανοίξουμε για δεύτερη φορά το
        παράθυρο
        panel.repaint();
    }
    player.prefetch();
} else if (event instanceof EndOfMediaEvent) {
    setTitle("Video - Τέλος");

} else if (event instanceof ControllerErrorEvent) {
    player = null;
    System.out.println("Cannot play video!! ");
    return;
} else if (event instanceof PrefetchCompleteEvent) {
    if (visualComponent != null) {
        pack();
    }
    player.start();
} else if (event instanceof SizeChangeEvent) {
    pack();
}
}

```

#### 7.1.4 Κλάση GlassQuestionShow

Χρησιμοποιείται για την εμφάνιση ερωτήσεων βιβλίου, ουσιαστικά πρόκειται για κλάση που ενεργοποιεί την «γυάλινη διατομή» του παραθύρου (glass pane), πρόκειται για ένα διαφανές επίπεδο που καλύπτει όλη την επιφάνεια του παραθύρου. Οτιδήποτε ζωγραφίσουμε σ' αυτό καλύπτει όλα τα συστατικά του παραθύρου είτε πρόκειται για κουμπιά είτε για μενού.

Η SafeFrame δίνει ως παράμετρο στο δομητή της GlassQuestionShow, το αντικείμενο που περιγράφει την «γυάλινη διατομή», με την μέθοδο `getGlassPane()`.

Επιπλέον η κλάση GlassQuestionShow, αφού εμφανίσει το κείμενο προκαλεί παύση στο παιχνίδι, η παύση δημιουργείται με το αμοιβαίο κλείδωμα του αντικειμένου GameEngine.

##### Κώδικας 7-5 - Παύση με κλείδωμα της GameEngine

```

questionReading = true;
try {
    wait();
    questionReading = false;
} catch (InterruptedException e) {
}

```

Και όταν πατηθεί το κουμπί «Εντάξει» στη GlassQuestionShow, απελευθερώνει την κλειδαριά εκτελώντας τον παρακάτω κώδικα :

##### Κώδικας 7-6 - Απελευθέρωση κλειδαριάς στη GlasQuestionShow

```

synchronized (locker) {
    locker.notify();
}

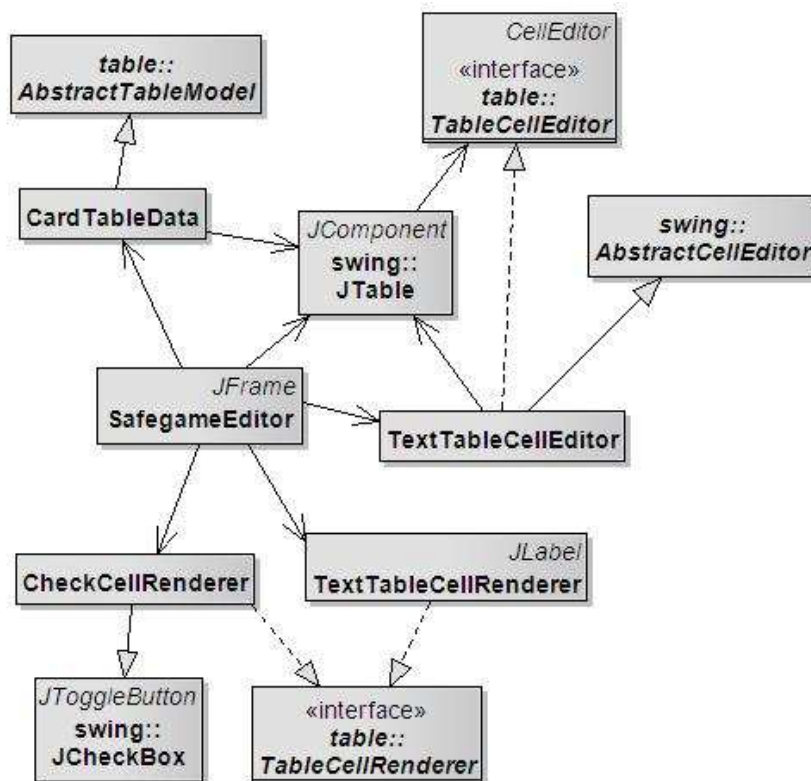
```

## 7.2 Υλοποίηση του SafeGameEditor

Ο συντάκτης ομοίως πακετάρεται σε ένα jar, ενώ στον κατάλογο που βρίσκεται το jar πρέπει να υπάρχει ένας κατάλογος με το όνομα *lib*. Στον κατάλογο αυτό πρέπει να βρίσκεται το jar του SafeGame, αυτό είναι απαραίτητο καθώς ο SafeGameEditor χρησιμοποιεί κλάσεις του SafeGame (π.χ. DataPack, Card κ.α.). Ο SafeGameEditor ουσιαστικά αποτελεί μία φόρμα που κάνει χρήση των στοιχείων του πακέτου swing. Από αυτό ενδιαφέρον κομμάτι αποτελεί η μορφοποίηση του πίνακα των καρτών και που φαίνεται στην Εικόνα 6-42.

### 7.2.1 Πίνακας καρτών και κλάση CardTableData

Στην κλάση SafeGameEditor υπάρχει ο πίνακας που περιγράφεται από την κλάση JTable, όταν θέλουμε να τροποποιήσουμε έναν πίνακα χρησιμοποιούμε ένα μοντέλο που περιγράφεται από την κλάση AbstractTableModel. Έτσι δημιουργούμε την CardTableData που κληρονομεί την AbstractTableModel και επιπλέον για το χειρισμό των κελιών δημιουργούμε τις CheckCellRender, TextTableCellRender, TextTableCellEditor().



Εικόνα 7-2 - Διάγραμμα κλάσεων SafeGameEditor

Στην κλάση SafeGameEditor δημιουργούμε και τον συσχετίζουμε με το μοντέλο, ομοίως για τις στήλες του πίνακα.

**Κώδικας 7-7 - Προσθήκη πίνακα στη SafeGameEditor**

```
TableCellRenderer renderer;  
TableCellEditor editor;  
TableColumn column;  
table = new JTable();  
tableData = new CardTableData(table);  
table.setModel(tableData);  
//1η στήλη με τον μοναδικό αριθμό της κάρτας  
renderer = new DefaultTableCellRenderer();  
//δεν βάζω editor editor = new DefaultCellEditor();  
column = new TableColumn(0, 30, renderer, null);  
table.addColumn(column);  
//2η στήλη με τους τύπους των καρτών  
JComboBox typeCombo = new JComboBox(CardTableData.CARD_TYPES);  
editor = new DefaultCellEditor(typeCombo);  
column = new TableColumn(1, 200, renderer, editor);  
table.addColumn(column);  
//3η στήλη με το κείμενο  
editor = new TextTableCellEditor(table);  
renderer = new TextTableCellRenderer();  
column = new TableColumn(2, 500, renderer, editor);  
table.addColumn(column);  
//4η στήλη με το checkBox  
JCheckBox videoCheckBox = new JCheckBox();  
videoCheckBox.setHorizontalAlignment(JCheckBox.CENTER);  
renderer = new CheckCellRenderer();  
editor = new DefaultCellEditor(videoCheckBox);  
column = new TableColumn(3, 70, renderer, editor);  
table.addColumn(column);
```



## 8. Προτάσεις - Συμπεράσματα

Το παιχνίδι μπορεί να βελτιωθεί, βελτιώνοντας τα γραφικά του. Ένα τρισδιάστατο περιβάλλον πολλές φορές εντυπωσιάζει τον χρήστη και κάνει το παιχνίδι ευχάριστο.

Μία δεύτερη βελτίωση είναι προσθήκη δικτυακής λειτουργίας έτσι ώστε πολλοί χρήστες να παίζουν ταυτόχρονα μέσω τοπικού ή παγκόσμιου δικτύου. Η δικτυακή λειτουργία του μπορεί να ακολουθήσει δύο πιθανά μοντέλα :

- 1) Κάθε χρήστης έχει ένα αντίγραφο εκτελέσιμου στον υπολογιστή του. Ένας από τους χρήστες αναλαμβάνει τον ρόλο διακομιστή (server) και οι υπόλοιποι συνδέονται σε αυτόν με ρόλους πελατών (client). Η επικοινωνία επιτυγχάνεται με βάση την διεύθυνση IP και ορίζοντας κάποια θύρα (port) επικοινωνίας. Το μοντέλο αυτό επικρατεί συνήθως σε τοπικά δίκτυα, ενώ η επικοινωνία μέσω του διαδικτύου πολλές φορές έχει ορισμένα προβλήματα όπως Η/Υ που έχουν ιδιωτική IP (private IP) ή βρίσκονται πίσω από τείχος προστασίας (firewall). Για να λυθούν τα προβλήματα, απαιτούν τεχνικές γνώσεις που πολλές φορές δεν έχουν οι χρήστες.
- 2) Ένα δεύτερο μοντέλο που μπορεί να χρησιμοποιηθεί είναι αυτό στο οποίο οι χρήστες παίζουν το παιχνίδι μέσω του πλοηγού ιστοσελίδων (web browser). Το παιχνίδι πλέον μπαίνει με τη μορφή applet σε μία ιστοσελίδα που φιλοξενείται σε έναν κεντρικό διακομιστή. Οι χρήστες μπορούν να συνδεθούν στην ιστοσελίδα ο καθένας με ένα προσωπικό όνομα χρήστη και συνθηματικό.

Τελειώνοντας την εργασία μπορούμε να πούμε πως η ανάπτυξη παιχνιδιών και γενικότερα η ανάπτυξη λογισμικού αποτελείται από κύκλους βελτιώσεων. Κάθε φορά που τελειώνει ένα έργο ελέγχεται και μπαίνει σε λειτουργία, έπειτα είναι δυνατόν να ανακαλυφθούν λάθη και προβλήματα που δεν ήταν άμεσα ορατά. Επιπλέον νέες ιδέες προστίθενται και το προϊόν μπαίνει πάλι στη φάση της ανάπτυξης.

## Βιβλιογραφία

- 1) Βεσκούκης, Β. (2001). Τεχνολογία Λογισμικού ΙΙ. Πάτρα: ΕΛΛΗΝΙΚΟ ΑΝΟΙΧΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ – Σχολή Θετικών Επιστημών και Τεχνολογίας.
- 2) Arlow, J. & Neustadt, I. (2002). UML and the unified process : practical object-oriented analysis and design. London: Addison-Wesley.
- 3) Booch, G. [et al.]. (2007). Object-oriented analysis and design with applications. (3η εκδ.). Boston, MA: Addison-Wesley.
- 4) Davinson, A. (2005). Killer Game Programming in Java. Sebastol, CA: O'Reilly Media.
- 5) Fowler, M. (2004). UML distilled : a brief guide to the Standard object modeling language. (3η εκδ.). Boston, MA: Addison-Wesley.
- 6) IEEE. (1998). IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society.
- 7) Kent, S. (2001). The Ultimate History of Video Games. Roseville, CA: Prima.
- 8) Knudsen, J. (1999). Java 2D Graphics. CA: O'Reilly Media.
- 9) Loy, M. [et. al.] (2002). Java Swing. (2η εκδ.). CA: O'Reilly Media.
- 10) Miles, R. & Hamilton, K. (2006). Learning UML 2.0. Sebastol, CA: O'Reilly Media.
- 11) Oaks, S. & Wong, H. (2004). Java Threads. (3η εκδ.). CA: O'Reilly Media.
- 12) Robinson, M. & Vorobiev, P. (2003). Swing. (2η εκδ.). Greenwich, CT: Manning.
- 13) Terrazas, A., Ostuni, J. & Barlow, M. (2002). Java™ Media APIs: Cross-Platform Imaging, Media, and Visualization. Indianapolis: Sams Publishing.
- 14) The Java Tutorials. (14 Μαρτίου 2008). Ανακτήθηκε 20 Απριλίου, 2008, από <http://java.sun.com/docs/books/tutorial/>

## ΠΑΡΑΡΤΗΜΑ Α. - Οδηγός χρήσης λογισμικού

Απαιτήσεις : Για την χρήση και των δύο κομματιών λογισμικού απαιτείται εγκατάσταση του Java RunTime Enviroment 1.4 ή νεώτερου.

### A.1 Οδηγός χρήσης του SafeGame

Εγκατάσταση: Είναι δυνατόν να έχουμε είτε ένα εκτελέσιμο .exe για το περιβάλλον των Windows είτε ένα αρχείο .jar για όλα τα λειτουργικά συστήματα. Αποσυμπίεζουμε τα περιεχόμενα του αρχείου *safegame.rar* στον ίδιο φάκελο, τα αρχεία που εμφανίζονται είναι :

- Τα εκτελέσιμα αρχεία είτε το *safegame.jar* είτε το *safegame.exe*.
- Το αρχείο των ρυθμίσεων settings.
- Ο φάκελος με τα βίντεο ./video/ .
- Ο φάκελος με την βιβλιοθήκη ./lib/ .

Εκτέλεση:

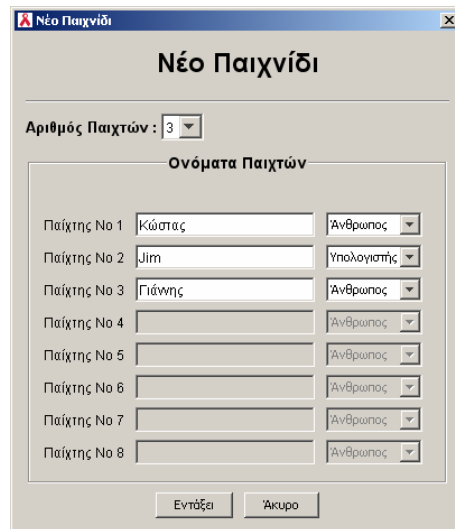
Το jar μπορούμε να το εκτελέσουμε με την εξής εντολή :

```
java -jar safegame.jar
```

Παιχνίδι:

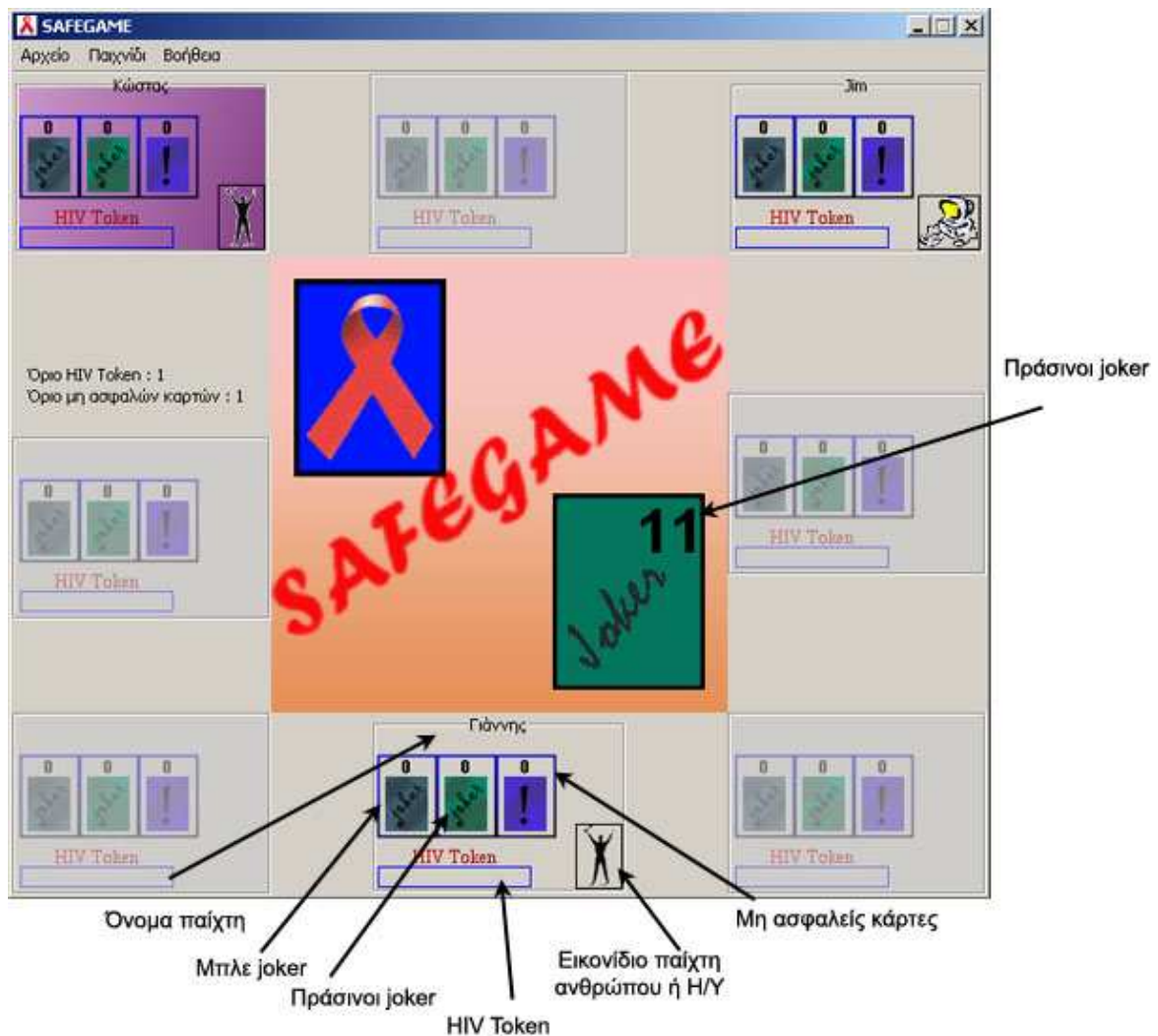
Οι κανόνες του παιχνιδιού περιγράφονται στο 3ο κεφάλαιο.

Έπειτα εμφανίζεται το κεντρικό παράθυρο του παιχνιδιού. Πλέον το παιχνίδι μπορεί να ξεκινήσει επιλέγοντας από το μενού «Παιχνίδι» και έπειτα «Νέο παιχνίδι» ή πατώντας τον συνδυασμό *Ctrl+N*. Μία φόρμα εμφανίζεται όπου επιλέγουμε το πλήθος των παιχτών, το όνομα τους και τη φύση τους (άνθρωπος-Η/Υ).



Εικόνα A-1 - Καρτέλα Νέο παιχνίδι

Έπειτα επιλέγουμε «Εντάξει» για να ξεκινήσει το νέο παιχνίδι. Οι παίχτες τοποθετούνται κυκλικά γύρω από το ταμπλό και το παιχνίδι ξεκινά. Κάθε παίχτης έχει το δικό του χώρο στο ταμπλό όπου περιέχει πληροφορίες για αυτόν.

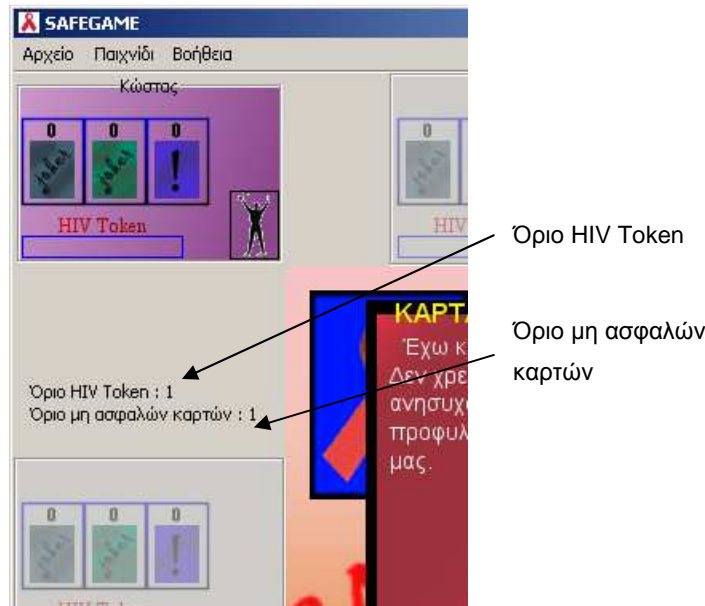


Εικόνα A-2 - Ταμπλό του παιχνιδιού

Με μία κυκλική εναλλαγή μεταξύ των παιχτών καταλήγει τυχαία στον παίχτη που θα παίξει. Έπειτα ο παίχτης τραβά μία κάρτα από τη στοίβα με τις μπλε κάρτες και την διαβάζει, οι εξής περιπτώσεις μπορούν να συμβούν :

- Ο παίχτης να τραβήξει μία μη ασφαλή κάρτα κατάστασης, δηλαδή μία κάρτα που περιγράφει μία μη ασφαλή κατάσταση. Η επόμενη κίνηση είναι να ρίξει ένα νόμισμα με ένα χαρούμενο και ένα θλιμμένο πρόσωπο. Αν έρθει ένα θλιμμένο πρόσωπο και ο παίχτης δεν έχει joker, τότε ένα HIV Token προστίθεται και κρατά και την μη ασφαλή κάρτα. Με έναν joker αποφεύγει το HIV Token, ενώ με δύο και την μη ασφαλή κάρτα. Στο χαρούμενο πρόσωπο, αν ο παίχτης δεν έχει joker κρατά την μη ασφαλή κάρτα, ενώ αν έχει όχι.
- Ο παίχτης να τραβήξει μία ασφαλή ή μία μη επικίνδυνη κάρτα κατάστασης. Ο παίχτης διαβάζει το κείμενο της κάρτας και αυτή μπαίνει στη στοίβα επιστροφής.
- Ο παίχτης να τραβήξει έναν μπλε joker, τον οποίο και κρατά.
- Ο παίχτης να τραβήξει μία κάρτα ερώτησης, όπου αν απαντήσει σωστά επιβραβεύεται με έναν πράσινο joker, εφόσον είναι διαθέσιμος.

Υπάρχουν δύο καταστάσεις που μπορεί να μπει ένας παίχτης, το πρόγραμμα εκπαίδευσης και η εξασθενημένη υγεία. Σε πρόγραμμα εκπαίδευσης μπαίνουν οι παίχτες που έχουν φτάσει ή ξεπεράσει το όριο των μη ασφαλών καρτών που αναγράφεται (Εικόνα A-3) και η υγεία τους εξασθενεί όταν τα HIV Token που έχουν μαζέψει έχουν φτάσει το όριο των HIV Token ().



**Εικόνα A-3 - Όρια παιχνιδιού**

Όταν ένας παίχτης μπει σε πρόγραμμα εκπαίδευσης, τότε διαβάσει μία ερώτηση βιβλίου, πράγμα που σημαίνει ότι μία ερώτηση και η απάντηση της εμφανίζονται στην οθόνη προς επιμόρφωση των παιχτών, έπειτα ο παίχτης παίζει. Ο παίχτης που η υγεία του έχει εξασθενήσει έχει πλέον σταματήσει και βοηθάει τους συμπαίχτες του διαβάζοντας μία ερώτηση βιβλίου κάθε φορά που είναι να παίζει. Όταν τελειώσει η στοίβα με τις μπλε κάρτες, ανακατεύεται και παίζει πάλι. Το παιχνίδι τελειώνει όταν η υγεία όλων των παιχτών εξασθενήσει ή όταν έχει τελειώσει και για δεύτερη φορά η στοίβα με τις κάρτες.

#### Λειτουργίες παιχνιδιού:

Ο χρήστης έχει την δυνατότητα να αποθηκεύσει ένα παιχνίδι επιλέγοντας από το μενού «Αρχείο» και έπειτα «Αποθήκευση» ή με τον συνδυασμό *Ctrl+S*, έπειτα μπορεί να συνεχίσει να παίζει το παιχνίδι.

Ομοίως μπορεί να φορτώσει ένα παιχνίδι από το μενού «Αρχείο» και «Αποθηκευμένα παιχνίδια» ή με τον συνδυασμό *Ctrl+O*.

Ο ήχος του παιχνιδιού μπορεί να απενεργοποιηθεί και να ενεργοποιηθεί από το μενού «Παιχνίδι» και έπειτα «Ήχος».

Ο χρήστης μπορεί να προκαλέσει παύση του παιχνιδιού από το μενού «Παιχνίδι» και έπειτα «Παύση παιχνιδιού» ή με τον συνδυασμό *Ctrl+P*. Παύση του παιχνιδιού επίσης προκαλείται όταν ενεργοποιήσουμε ένα άλλο παράθυρο ή κάνουμε minimize το παιχνίδι, έτσι ώστε να μην χάσουμε την ροή του παιχνιδιού.

### Ρυθμίσεις παιχνιδιού:

Μπορούμε να αλλάξουμε τις ρυθμίσεις του παιχνιδιού, τις κάρτες του και τις ερωτήσεις βιβλίου αλλάζοντας το αρχείο ρυθμίσεων. Το αρχείο με το όνομα *settings* που βρίσκεται στον ίδιο κατάλογο με το εκτελέσιμο του παιχνιδιού περιέχει όλες τις πληροφορίες του παιχνιδιού, ο χρήστης μπορεί να το αντικαταστήσει με ένα άλλο, απαραίτητο όμως είναι το όνομα του αρχείου να παραμείνει *settings*.

## **A.2 Οδηγός χρήσης του SafeGameEditor**

Εγκατάσταση: Ομοίως μπορούμε να έχουμε ένα απευθείας εκτελέσιμο για το λειτουργικό σύστημα των Windows ή ένα αρχείο jar.

Αποσυμπιέζουμε το *safegameeditor.rar* στον ίδιο φάκελο, τα αρχεία που εμφανίζονται είναι :

- Το εκτελέσιμο που μπορεί να είναι είτε το *safegameeditor.exe* είτε το *safegameeditor.jar*.
- Ένα φάκελος με το όνομα *lib* που περιέχει το αρχείο *safegame1.jar*.

### Εκτέλεση:

Τρέχουμε με διπλό κλικ το αρχείο *exe* ή από την γραμμή εντολών δίνουμε :

```
java -jar safegameeditor.jar
```

### Αποθήκευση - Ανάκτηση :

Στο συντάκτη μπορούμε να φορτώσουμε τις πληροφορίες ενός αρχείου ρυθμίσεων και να τις αποθηκεύσουμε κάνοντας από το μενού «Αρχείο» και έπειτα «Φόρτωση Αρχείου» ή «Αποθήκευση Αρχείου» αντίστοιχα. Το όνομα του αρχείου ή η επέκτασή του είναι στην επιλογή του χρήστη.

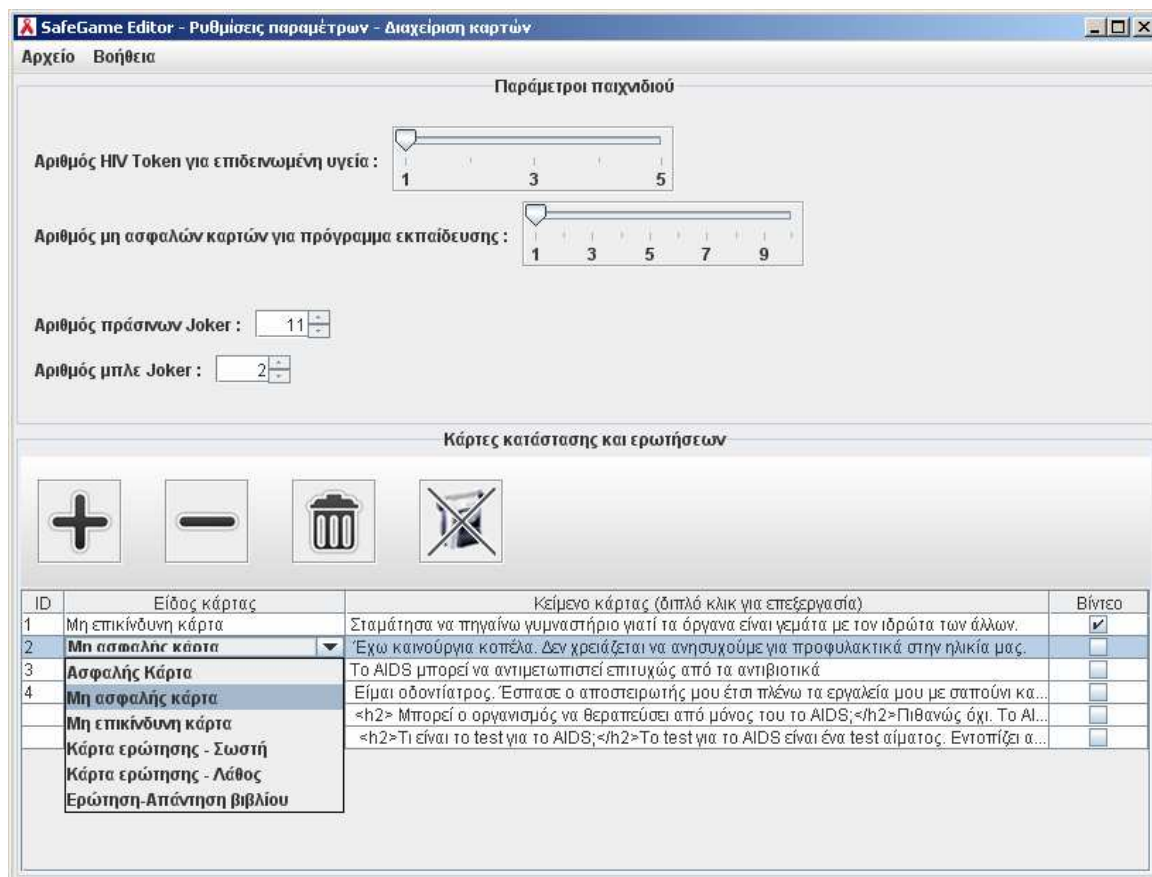
### Προσθήκη - επεξεργασία στοιχείων:

Για τον ορισμό των ορίων HIV Token και μη ασφαλών καρτών υπάρχουν δύο ροοστάτες, ομοίως δύο ρυθμιστές υπάρχουν για το πλήθος των πράσινων και μπλε joker.

Για την προσθήκη μιας κάρτας ή μια ερώτησης βιβλίου πατάμε το κουμπί με το σύμβολο «+», ενώ το σύμβολο «-» χρησιμεύει στο να αφαιρεθεί μία επιλεγμένη γραμμή του πίνακα. Το εικονίδιο με τον κάδο μπορεί να αφαιρέσει όλες τις γραμμές του πίνακα και τέλος το εικονίδιο με την κάμερα απενεργοποιεί το βίντεο από όλες τις κάρτες.

Κατά την προσθήκη μιας κάρτας ή μιας ερώτησης βιβλίου ο χρήστης κάνοντας κλικ στο πεδίο «Είδος κάρτας», ένα μενού αναπτύσσεται για να επιλέξει το είδος

της κάρτας που θέλει. Το κείμενο της κάρτας ή της ερώτησης μπορεί να το εισάγει κάνοντας διπλό κλικ στο πεδίο «Κείμενο κάρτας» και μόνο για τις κάρτες μπορεί να ενεργοποιήσει το βίντεο από το αντίστοιχο πεδίο.



Εικόνα A-4 - Παράθυρο του SafeGameEditor

Στις ερωτήσεις βιβλίου ο χρήστης μπορεί να μορφοποιήσει το κείμενο κάνοντας χρήση της γλώσσα HTML, παρακάτω δίνεται ένα παράδειγμα.

<h2>Τι είναι το test για το AIDS;</h2>

Το test για το AIDS είναι ένα test αίματος. Εντοπίζει αντισώματα που μάχονται το AIDS στο αίμα του φορέα. Ένα θετικό test σημαίνει ότι τα αντισώματα του AIDS βρέθηκαν, και το άτομο είναι μολυσμένο. Αρνητικό test σημαίνει ότι το άτομο δεν είναι μολυσμένο ή ότι το test έγινε κατά την διάρκεια του 'παραθύρου', το οποίο είναι τρεις μήνες ανάμεσα στην μόλυνση και τον εντοπισμό των αντισωμάτων.