ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ Η/Υ

Διπλωματική Εργασία

Θέμα:

" Ευφυή παιχνίδια με τεχνικές προφίλ χρήστη – μία εφαρμογή android για το wizard. "

"Intelligent gaming with user profiling – an android application for wizard. "

Παπαδάκης Γεώργιος

Επιβλέπουσα Καθηγήτρια :
Δασκαλοπούλου Ασπασία (Επίκουρη Καθηγήτρια Π.Θ.)

Συν επιβλέπων Καθηγητής:
Βασιλακόπουλος Μιχαήλ (Αναπληρωτής Καθηγητής Π.Θ).

Βόλος, Οκτώβριος 2017

ΕΥΧΑΡΙΣΤΙΕΣ

Με την ολοκλήρωση της Διπλωματικής μου Εργασίας φέρω εις πέρας τις Προπτυχιακές σπουδές μου στο τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας.

Αρχικά θα ήθελα να ευχαριστήσω την επίκουρη καθηγήτρια κυρία Δασκαλοπούλου, η οποία με βοήθησε αρκετά με τις συμβουλές της και την σωστή καθοδήγηση της, ώστε να ολοκληρώσω με επιτυχία την διπλωματική μου εργασία. Επίσης θα ήθελα να ευχαριστήσω και τον συν επιβλέπον αναπληρωτή καθηγητή Βασιλακόπουλο Μιχαήλ για την πολύτιμη βοήθεια του και τον χρόνο του.

Ένα ακόμη μεγάλο ευχαριστώ στους φίλους που μου συμπαραστάθηκαν και με βοήθησαν όλα αυτά τα χρόνια και ιδιαίτερα τον Πολυχρόνη Συμεωνίδη που ήταν πάντα δίπλα μου όταν τον χρειαζόμουν.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για την στήριξη και για τα απαραίτητα εφόδια που μου παρείχαν καθ' όλη την διάρκεια των ακαδημαϊκών σπουδών μου.

Αφιερωμένο στην οικογένεια μου

ΠΕΡΙΛΗΨΗ

Η εξέλιξη της τεχνολογίας και κατ' επέκταση των κινητών τηλεφώνων έχουν αλλάξει δραματικά τη ζωή μας , το πώς σκεφτόμαστε και το πώς λειτουργούμε. Συνεπώς, η καθημερινή χρήση των κινητών τηλεφώνων δεν περιορίζεται μόνο στις καθημερινές μας ανάγκες, αλλά και στην ανάγκη για διασκέδαση. Με τις εφαρμογές παιχνιδιών να αυξάνονται καθημερινά και μαζί με αυτές η τεχνητή νοημοσύνη που χρησιμοποιείτε για την λήψη αποφάσεων μέσα από τυχαία γεγονότα. Δεδομένου ότι τα on-line παιχνίδια καρτών είναι ένα είδος συστήματος προσομοίωσης εικονικής πραγματικότητας, το οποίο έχει την ισχυρή ικανότητα να δημιουργήσει ένα περιβάλλον παιχνιδιού με συναρπαστικές ρεαλιστικές σκηνές, τα παιχνίδια καρτών καθίστανται ολοένα και πιο δημοφιλή και έχουν γίνει ευρέως διαδεδομένα.

Η διπλωματική αυτή εργασία θα έχει σαν σκοπό την υλοποίηση ενός off-line παιχνιδιού με κάρτες το οποίο θα έχει σαφή χαρακτηριστικά τεχνητής νοημοσύνης, για την δημιουργία ενός εικονικού περιβάλλον με όσο το δυνατόν πιο ρεαλιστικές καταστάσεις, ώστε να το καθιστά ανταγωνιστικό με άλλα διαδικτυακά παιχνίδια.

Το θέμα της παρούσας διπλωματικής εργασίας ασχολείται με την υλοποίηση ενός επιτραπέζιου παιχνιδιού με κάρτες – Wizard - σε εφαρμογή android , συμπεριλαμβάνοντας κώδικα τεχνητής νοημοσύνης, ώστε να το καθιστά πιο ενδιαφέρον και ανταγωνιστικό.

Με την αναφορά μας σε παρόμοια παιχνίδια που χρησιμοποιούν τεχνητή νοημοσύνη θα ξεκινήσουμε την λεπτομερή ανάλυση της δημιουργίας αυτού του παιχνιδιού. Πριν αναφερθούμε στο πως παίζεται το επιτραπέζιο Wizard θα αναφερθούμε σε κάποιες τεχνικές που μας βοήθησαν για την ολοκλήρωση τις εργασίας. Στη συνέχεια, θα αναλύσουμε τον σχεδιασμό και την υλοποίηση της εφαρμογής μέσα από τους αλγόριθμους που αναπτύξαμε και τα διαγράμματα τους. Τέλος θα αναφέρουμε τα συμπεράσματα μας από αυτήν την εργασία και από πού πήραμε όλες αυτές τις πληροφορίες που μας βοήθησαν.

Λέξεις κλειδιά: « Τεχνητή νοημοσύνη, επιτραπέζιο Wizard, στρατηγική, επανάληψη παιχνιδιού, Μάγος, Γελωτοποιός, Ατού, πρόβλεψη, μπάζα »

ABSTRACT

The evolution of technology and, by extension, mobile phones have dramatically changed our lives, how we think and how we operate. Consequently, the daily use of mobile phones is not limited to our everyday needs, but also to the need for entertainment. Game applications grow daily and along with them the artificial intelligence, that use to make decisions through random events. Since on-line card games are a kind of virtual reality simulation system that has the strong ability to create a gaming environment with exciting realistic scenes, card games are becoming more and more popular and widespread.

This diploma thesis will aim to create an off-line card game with clear artificial intelligence to create a virtual environment with the most realistic possible situations to make it competitive with other online games.

The subject of this diploma thesis deals with the implementation of a board game - Wizard - in android application, including an artificial intelligence code to make it more interesting and competitive.

By referring to similar games using Artificial Intelligence, we will begin the detailed analysis of the creation of this game. Before we talk about how the board game Wizard is played, we'll refer to some techniques that helped us complete the work. Then we will analyze the design and implementation of the application through the algorithms we developed. Finally, we will report our findings from this paper and from where we got all this information that helped us.

Keywords: "Artificial intelligence, board game Wizard, strategy, game loop, wizard, joker, trump, prediction, stack"

Περιεχόμενα

1

Εισαγωγή

1.1 Εφαρμογές παιχνιδιών σε Android

Με την τεχνολογία της κινητής τηλεφωνίας να εξελίσσεται ολοένα και περισσότερο τα τελευταία χρόνια, τα έξυπνα κινητά τηλέφωνα ενσωματώνουν δυνατότητες που μέχρι σήμερα δεν υπήρχαν. Με αποτέλεσμα οι κινητές συσκευές να εισχωρούν όλο και περισσότερο στη ζωή μας. Είναι πλέον αναπόσπαστο κομμάτι στην καθημερινότητα πολλών ανθρώπων καθώς παρέχουν πολλές δυνατότητες στο χρήστη όπως να ψυχαγωγηθεί και να περάσει ευχάριστα την ώρα του μέσα από τα ψηφιακά παιχνίδια που μπορούν να ενσωματωθούν στις έξυπνες κινητές συσκευές.

Παλαιότερα, τα παιχνίδια που μπορούσε ο χρήστης να παίξει ήταν απλά και ενσωματωμένα στη συσκευή του τηλεφώνου. Τα γραφικά των παιχνιδιών δεν είχαν χρώματα και δεν ήταν ευχάριστα προς το χρήστη. Με την εξέλιξη όμως των κινητών τηλεφώνων και των υπηρεσιών τους εξελίσσονταν παράλληλα και τα παιχνίδια που μπορούσαν να ενσωματωθούν στις κινητές συσκευές. Μέσω αυτής της εξέλιξης τα παιχνίδια τείνουν να γίνονται πιο “έξυπνα” ώστε να μην χάνει το ενδιαφέρον του ο χρήστης.

1.2 Υλοποίηση του Wizard σε Android app

Η παρούσα διπλωματική εργασία έχει ως αντικείμενο την υλοποίηση του Wizard σε εφαρμογή android, χρησιμοποιώντας αλγορίθμους τεχνητής νοημοσύνης. Το Wizard είναι ένα επιτραπέζιο παιχνίδι με κάρτες, όπου κάθε παίκτης καλείται να κάνει μια πρόβλεψη για το πόσες φορές θα κερδίσει τους υπόλοιπους παίκτες και στη συνέχεια προσπαθεί να πέσει μέσα στην πρόβλεψη του. Η φύση του παιχνιδιού είναι τέτοια που προσφέρει την χρήση της τεχνητής νοημοσύνης σε μεγάλο βαθμό, τόσο στην πρόβλεψη του κάθε παίκτη , όσο και στον

τρόπο που ο παίκτης αυτός θα παίξει τα χαρτιά του ώστε να κερδίσει ακριβώς όσες φορές προέβλεψε.
Με αυτόν τον τρόπο το Wizard θα γίνει ένα ενδιαφέρον παιχνίδι το οποίο θα βάζει τον χρήστη σε μία διαδικασία να παρατηρεί την συμπεριφορά του αντίπαλου παίκτη (λόγω της τεχνητής νοημοσύνης), ώστε να μπορέσει να τον νικήσει. Τα περισσότερα παιχνίδια δεν ενδιαφέρονται για τις σκέψεις του παίκτη και αυτό τα κάνει λιγότερο ανταγωνιστικά σε αντίθεση με την εφαρμογή που θα υλοποιήσουμε.

1.2.1 Συνεισφορά

Παρακάτω παραθέτουμε τις ενέργειες και μεθοδολογίες που χρησιμοποιήσαμε για την υλοποίηση της εφαρμογής, καθώς και κάποια μοτίβα που παρατηρήσαμε μετά από κάποια τρεξίματα των αλγορίθμων.

1. Ερευνώντας για παρόμοιες εφαρμογές σε android κατανοήσαμε πλήρως πώς να μπορούμε να υλοποιήσουμε μία λειτουργική επανάληψη παιχνιδιού.
2. Μελετώντας παρόμοια παιχνίδια σχεδιάσαμε το interface την εφαρμογής έτσι ώστε ο χρήστης να έχει συνεχώς οπτική επαφή με όλα τα στοιχεία που χρειάζεται για να παίξει.
3. Υλοποιήσαμε 2 διαφορετικούς αλγορίθμους τεχνητής νοημοσύνης για την πρόβλεψη των νικών κάθε παίκτη.
4. Άλλοι 3 αλγόριθμοι χρησιμοποιήθηκαν με χαρακτηριστικά τεχνητής νοημοσύνης για τον τρόπου που κάθε παίκτης αποφασίζει πιο φύλο θα ρίξει αναλόγως την σειρά του , τα φύλα που έχουν ρίξει οι προηγούμενοι από αυτόν, την πρόβλεψη του και άλλους παράγοντες που θα εξηγήσουμε παρακάτω.
5. Η λογική που βασίστηκαν οι παραπάνω αλγόριθμοι είναι να προσπαθήσουμε να δώσουμε μία αξία στη κάθε κάρτα που κρατάει ο παίκτης, ώστε να δούμε πόσο πιθανόν είναι να κερδίσει με τη συγκεκριμένη κάρτα.
6. Τέλος, τρέχοντας αυτούς τους αλγορίθμους παρατηρήσαμε πως το αποτέλεσμα ήταν παραπάνω από ικανοποιητικό όσον αφορά τον νικητή του παιχνιδιού.

1.3 Διάρθρωση της Εργασίας

Στη συνέχεια στο δεύτερο κεφάλαιο θα αναφέρουμε κάποια παρόμοια παιχνίδια που χρησιμοποιούν την τεχνητή νοημοσύνη για την πρόβλεψη κινήσεων. Έπειτα στο τρίτο κεφάλαιο θα αναφερθούμε στην ήδη υπάρχουσα θεωρία και τεχνικές που μας βοήθησαν σε αυτή την εργασία. Συνεχίζοντας στο επόμενο κεφάλαιο θα εξηγήσουμε αναλυτικά πως παίζεται το Wizard. Στο πέμπτο κεφάλαιο θα δούμε πως δομήθηκε η εφαρμογή μας μέσα από μια σειρά αλγορίθμων και διαδικασιών. Τέλος, στο έκτο κεφάλαιο θα επισημάνουμε τα συμπεράσματα μας μαζί με μελλοντικές αναβαθμίσεις και στο έβδομο τις πηγές που χρησιμοποιήσαμε.

2

Σχετικές εργασίες

Όπως αναφέραμε ήδη η τεχνολογία έχει αλλάξει πολλά πράγματα γύρω μας και ο τρόπος που παίζουμε παιχνίδια είναι ένα από αυτά. Παρόμοια παιχνίδια σαν το wizard έχουν υλοποιηθεί παλιότερα σε εφαρμογές για κινητά τηλέφωνα, με την τεχνητή νοημοσύνη να παίζει μεγάλο ρόλο στον τρόπο παιχνιδιού. Σε αυτά τα παιχνίδια ένα είναι το κοινό χαρακτηριστικό που οδηγεί τον εκάστοτε παίκτη στην νίκη και αυτό είναι η στρατηγική. Τα πιο δημοφιλή παιχνίδια σε αυτόν τον τομέα είναι το σκάκι και το πόκερ.

2.1 Σκάκι

Το σκάκι, από παλιά ήταν πάντα ένα από τα πιο ενδιαφέροντα παιχνίδια. Απευθύνεται σε μια μεγάλη ποικιλία ανθρώπων που αγαπάνε αυτό το είδος παιχνιδιών. Επίσης, το σκάκι ήταν ένα από τα πρώτα παιχνίδια που έπαιζαν στον υπολογιστή με χρήση τεχνητής νοημοσύνης. Είναι ένα πολύ παλιό παιχνίδι το οποίο μπορεί να παιχτεί με ελάχιστες απαιτήσεις. Το σκάκι παίζεται με δύο παίκτες, μπορεί όμως ο ένας παίκτης να αντικατασταθεί με έναν υπολογιστή με την βοήθεια της τεχνητής νοημοσύνης.

Με την εξέλιξη της τεχνολογίας, το σκάκι έφτασε σε ένα επίπεδο όπου έκανε το παιχνίδι πιο ενδιαφέρον και προσβάσιμο από συσκευές όπως τα κινητά τηλέφωνα και τα tablet. Πλέον οποιοσδήποτε μπορεί να παίξει ανά πάσα στιγμή αυτό το παιχνίδι μέσα από τις ηλεκτρονικές συσκευές τους. Παρόλα αυτά οι περισσότεροι παίκτες θα επέλεγαν να παίξουν ένα πραγματικό παιχνίδι σκακιού από ένα εικονικό. Επίσης, εάν κάποιος παίκτης θέλει να κυριαρχήσει στο παιχνίδι αυτό, η κατανόηση των διαφορετικών μοτίβων των κινήσεων του σκακιού είναι σημαντική.

Με την πρόοδο της τεχνολογίας δημιουργήθηκαν πολλές εφαρμογές για παιχνίδια σκακιού. Όλες αυτές οι εφαρμογές ακολουθούν το ίδιο στυλ και δεν έχουν την ικανότητα να επιδεικνύουν αποτελεσματικά τις πιθανότητες για αποτελεσματικές και μη αποτελεσματικές κινήσεις. Οι πιο γνωστές όμως εφαρμογές είναι εκείνες που ένας παίκτης παίζει με έναν υπολογιστή ο οποίος χρησιμοποιεί αλγορίθμους τεχνητής νοημοσύνης. Τέτοιοι αλγόριθμοι είναι σχετικά απλοί αφού, ο υπολογιστής έχει την πλήρη επίγνωση για το πλήθος των καταστάσεων και κινήσεων του παιχνιδιού, σε αντίθεση με τα παιχνίδια με κάρτες όπου συμβαίνουν συνεχώς τυχαία γεγονότα.[1]

2.2 Πόκερ

Η τεχνητή νοημοσύνη που εφαρμόστηκε στα παιχνίδια τα τελευταία χρόνια, εστίασε περισσότερο στα παιχνίδια όπου οι παίκτες δεν έχουν πλήρη πληροφόρηση για το σύνολο των καταστάσεων του παιχνιδιού και όπου τυχαία γεγονότα εμφανίζονται καθ’ όλη την διάρκεια του παιχνιδιού. Μία κατηγορία τέτοιων παιχνιδιών είναι τα παιχνίδια με κάρτες.

Για παράδειγμα το Πόκερ είναι ένα τέλειο θέμα για τη μελέτη αυτής της κατηγορίας. Η πιο γνωστή παραλλαγή πόκερ είναι το Texas Hold'em που συνδυάζει απλούς κανόνες με ένα τεράστιο αριθμό πιθανών στρατηγικών.

Η ανάπτυξη ενός παιχνιδιού για το Πόκερ θα επικεντρώνεται στην ανάπτυξη αλγορίθμων για το "Opponent Modeling", επιλέγοντας την καλύτερη στρατηγική εναντίον κάθε δεδομένου αντιπάλου. Αυτές οι ικανότητες μοντελοποίησης αντιπάλων και η στρατηγική παιχνιδιού μπορούν να υλοποιηθούν με ένα Ευφυές Πράκτορα ικανό να παρατηρεί τους αντιπάλους και να προσαρμόζει τον τρόπο παιχνιδιού του με βάση τη συμπεριφορά των αντιπάλων.

Διάφοροι πράκτορες πρέπει να αναπτυχθούν για να προσομοιώσουν τη συμπεριφορά ενός τυπικού παίκτη πόκερ. Ένας άλλος πράκτορας πρέπει να αναπτυχθεί ώστε να είναι ικανός να χρησιμοποιεί τεχνικές μοντελοποίησης της συμπεριφοράς των αντιπάλων, για να επιλέγει την καλύτερη στρατηγική παιχνιδιού εναντίον κάθε αντιπάλου. Πρέπει όμως να γίνει πολλή δουλειά για να αποκτήσουμε έναν παράγοντα ικανό να παίζει σε ανθρώπινο επίπεδο. Παρόμοιες χρήσεις τεχνητής νοημοσύνης θα ήταν ένα θετικό έργο που θα μπορούσε να χρησιμοποιηθεί ως βάση για την ανάπτυξη μερικών πιο εξελιγμένων τεχνικών στο παιχνίδι.

Τα χαρακτηριστικά του πόκερ το κάνουν ενδιαφέρον για την μελέτη αυτού του είδους της τεχνητής νοημοσύνης. Τα χαρακτηριστικά αυτά είναι η ελλιπής γνώση, η διαχείριση κινδύνου,

η μοντελοποίηση της προσωπικότητας των αντιπάλων και η αντιμετώπιση αναξιόπιστων πληροφοριών. Σε αντίθεση με τα παιχνίδια τέλειας ενημέρωσης (όπως το σκάκι) στα οποία οι παίκτες είναι απόλυτα ενημερωμένοι για την κατάσταση του παιχνιδιού, στο πόκερ, οι παίκτες αντιμετωπίζουν κρυμμένες πληροφορίες από τις κάρτες αντιπάλων και μελλοντικές ενέργειες.

Σε ένα περιβάλλον με πολλούς πράκτορες, το αποτέλεσμα ενός πράκτορα θα επηρεάζεται από τις ενέργειες των άλλων πρακτόρων. Συνεπώς, ένας πράκτορας θα πρέπει να εξετάσει τις πιθανές ενέργειες των άλλων πρακτόρων. Η θεωρία των παιχνιδιών παρέχει το μαθηματικό υπόβαθρο για να εξηγεί πώς πρέπει να συμπεριφέρονται οι ορθολογικοί πράκτορες κάτω από τις παραπάνω ρυθμίσεις. Δυστυχώς, ακόμα και σε ρυθμίσεις όπου έχουμε συγκεκριμένη καθοδήγησή για τη βέλτιστη συμπεριφορά ενός πράκτορα, το υπολογιστικό πρόβλημα του προσδιορισμού αυτών των στρατηγικών παραμένει δύσκολο. Το "Opponent Modeling" μπορεί να βελτιώνεται συνεχώς με βάση τα συλλεχθέντα στατιστικά στοιχεία και το ιστορικό στοιχημάτων κάθε αντιπάλου. [2]

3

Θεωρία και Τεχνικές

Σε αυτό το κεφάλαιο θα αναλύσουμε τις θεωρίες και τις τεχνικές μου χρησιμοποιήσαμε για την υλοποίηση αυτής της εφαρμογής. Επίσης θα αναφερθούμε και στην γλώσσα προγραμματισμού καθώς και το εργαλείο που μας βοήθησε να αναπτύξουμε το παιχνίδι.

3.1 Java

Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού και έχει σχεδιαστεί ειδικά για να έχει όσο το δυνατόν λιγότερες εξαρτήσεις εκτελέσεων. Σκοπός της είναι να επιτρέπει στους προγραμματιστές εφαρμογών να "γράψουν μία φορά, να τρέξουν οπουδήποτε", που σημαίνει ότι ο κώδικας Java μπορεί να εκτελεστεί σε όλες τις πλατφόρμες που υποστηρίζουν την Java. Οι εφαρμογές Java μπορούν να τρέξουν σε οποιαδήποτε εικονική μηχανή Java ανεξάρτητα από την αρχιτεκτονική του υπολογιστή. Η Java είναι μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού που χρησιμοποιούνται, ειδικά για Web εφαρμογές. Η γλώσσα java αποδίδει μεγάλο μέρος της σύνταξής της από τη C και τη C ++, αλλά έχει λιγότερες εγκαταστάσεις χαμηλού επιπέδου από ότι και οι δύο.[3][9]
«*Η τελευταία έκδοση είναι η Java 9, η οποία κυκλοφόρησε στις 21 Σεπτεμβρίου 2017 και είναι μία από τις δύο εκδόσεις που υποστηρίζονται δωρεάν από την Oracle. Οι εκδόσεις νωρίτερα από την Java 8 υποστηρίζονται τόσο από την Oracle όσο και από άλλες εταιρείες σε εμπορική βάση*».[9]

3.2 Android Studio

Το Android Studio είναι το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης για το λειτουργικό σύστημα Android της Google, το οποίο βασίζεται στο λογισμικό IntelliJ IDEA της JetBrains

και έχει σχεδιαστεί ειδικά για ανάπτυξη Android. Είναι διαθέσιμο για λήψη σε λειτουργικά συστήματα που βασίζονται σε Windows, macOS και Linux. Αποτελεί αντικατάσταση του Eclipse Android Development Tools για την ανάπτυξη εφαρμογών Android.

«*Το Android Studio ανακοινώθηκε στις 16 Μαΐου 2013. Ήταν σε στάδιο early access ξεκινώντας από την έκδοση 0.1 τον Μάιο του 2013 και στη συνέχεια εισήλθε σε beta στάδιο ξεκινώντας από την έκδοση 0.8 που κυκλοφόρησε τον Ιούνιο του 2014. Η πρώτη σταθερή έκδοση κυκλοφόρησε τον Δεκέμβριο του 2014, ξεκινώντας από την έκδοση 1.0. Η τρέχουσα σταθερή έκδοση είναι 2.3.3, που κυκλοφόρησε τον Ιούνιο του 2017. Η επόμενη μεγάλη ενημέρωση, έκδοση 3.0, βρίσκεται σε στάδιο προεπισκόπησης από τον Σεπτέμβριο του 2017*.»[9]

Ακόμα το Android Studio είναι γραμμένο σε γλώσσα προγραμματισμού java και ήταν το εργαλείο που χρησιμοποιήθηκε για την υλοποίηση του Wizard σε Android εφαρμογή.[10]

3.3 Τεχνική ανάπτυξης Παιχνιδιού

Υπάρχουν διάφορες τεχνικές ανάπτυξης και υλοποίησης ενός παιχνιδιού και σε αυτό το κεφάλαιο θα μιλήσουμε για τις τεχνικές εκείνες που μας βοήθησαν για αυτήν την εργασία.

3.3.1 Game Loop

Μια τεχνική που μας βοήθησε και ενσωματώθηκε καλυτέρα στο παιχνίδι μας ήταν η δημιουργία μιας Game Loop.

```
boolean running = true;
while (!running)
{
   updateGameState();
   displayGameState();
}
```

Όπου η **updateGameState()** χρησιμοποιείτε για να ανανεώνει την κατάσταση του κάθε αντικειμένου στο παιχνίδι και η **displayGameState()** δημιουργεί τα αντικείμενα σε μία εικόνα η οποία εμφανίζεται στην οθόνη. Δύο πράγματα που πρέπει να κατανοήσουμε για τα γραφικά της οθόνης μας είναι τα **FPS** και **UPS**.

FPS – Frames Per Second είναι η συχνότητα με την οποία η updateGameState() καλείται και UPS – Update Per Second είναι η συχνότητα με την οποία η displayGameState() καλείται.
Λαμβάνοντας υπόψιν την είσοδο με τέτοιο τρόπο, ώστε να ενημερωθεί η εσωτερική κατάσταση του παιχνιδιού και στη συνέχεια να την εμφανίζουμε στην οθόνη.
Αφού διαχειριστούμε την είσοδο, ενημερώνουμε την κατάσταση των εσωτερικών αντικειμένων μας και ανανεώνουμε την τρέχουσα κατάσταση. Η Δραστηριότητα(activity) και η προβολή(view) είναι συνδεδεμένες μεταξύ τους και εκτελούνται η μία μετά την άλλη.
Οτιδήποτε στο Android συμβαίνει μέσα σε μια Δραστηριότητα. Η Δραστηριότητα θα δημιουργήσει μια Προβολή. Η προβολή είναι όπου συμβαίνουν τα πάντα. Εκεί λαμβάνει χώρα η αφή και εμφανίζεται η εικόνα που θέλουμε. Σκεφτείτε τη Δραστηριότητα ως ένα τραπέζι που περιέχει ένα φύλλο χαρτιού (την Προβολή) που μας επιτρέπει να σχεδιάσουμε κάτι. Θα χρησιμοποιήσουμε το μολύβι μας για να σχεδιάσουμε κάτι πάνω στο χαρτί. Αυτή θα είναι η αφή μας που αποτυπώνει κάτι στο χαρτί, έτσι ώστε το αποτέλεσμα αυτής της αλληλεπίδρασής μας με την Προβολή να παράγει μια εικόνα. Το ίδιο ισχύει και με τη Δραστηριότητα και την Προβολή.
Μια προβολή είναι μια απλή κλάση που μας παρέχει χειρισμό γεγονότων (όπως το onTouch) και ένα ορατό ορθογώνιο διαμορφωμένο χώρο για να αξιοποιήσουμε. Ο πιο απλός τρόπος είναι να επεκτείνετε το SurfaceView της Android. Θα εφαρμόσουμε επίσης το SurfaceHolder.Callback για να αποκτήσουμε πρόσβαση σε αλλαγές επιφάνειας, για παράδειγμα όταν καταστρέφεται(onDestroy).[4][3]
Στη συνέχεια θα δώσουμε εάν παράδειγμα τέτοιου κώδικα και θα τελειώσουμε την ενότητα αυτήν.

```
package net.obviam.droidz;
02
03 import android.content.Context;
04 import android.graphics.Canvas;
05 import android.view.MotionEvent;
06 import android.view.SurfaceHolder;
07 import android.view.SurfaceView;
08
09 public class MainGamePanel extends SurfaceView implements
10   SurfaceHolder.Callback {
11
12  public MainGamePanel(Context context) {
13   super(context);
```

```
14    // adding the callback (this) to the surface holder to intercept
      events
15    getHolder().addCallback(this);
16    // make the GamePanel focusable so it can handle events
17    setFocusable(true);
18   }
19
20   @Override
21   public void surfaceChanged(SurfaceHolder
      holder, int format, int width, int height) {
22   }
23
24   @Override
25   public void surfaceCreated(SurfaceHolder holder) {
26   }
27
28   @Override
29   public void surfaceDestroyed(SurfaceHolder holder) {
30   }
31
32   @Override
33   public boolean onTouchEvent(MotionEvent event) {
34    return super.onTouchEvent(event);
35   }
36
37   @Override
38   protected void onDraw(Canvas canvas) {
39   }
40 }
```

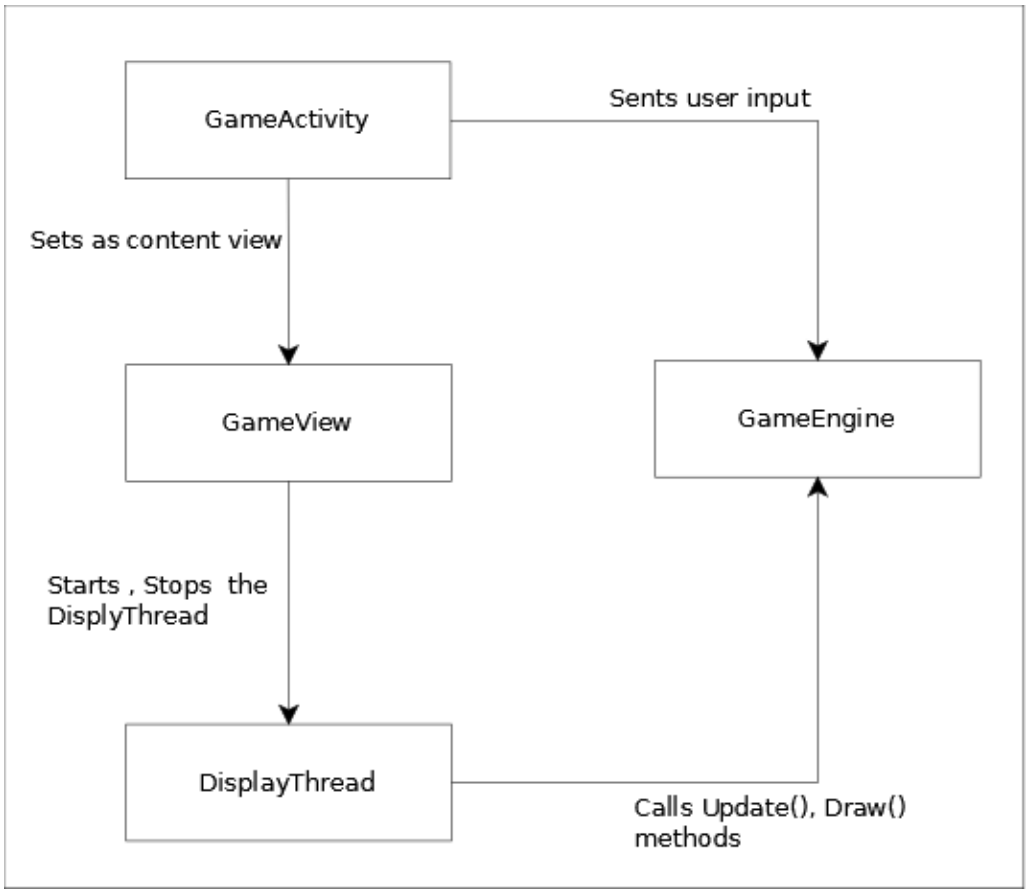
3.3.2 Διαγράμματα

Όπως αναφέραμε στη προηγούμενη ενότητα η βασική διαδικασία για την δημιουργία ενός παιχνιδιού είναι να προσδιορίσουμε κάποιες ενότητες, όπως:

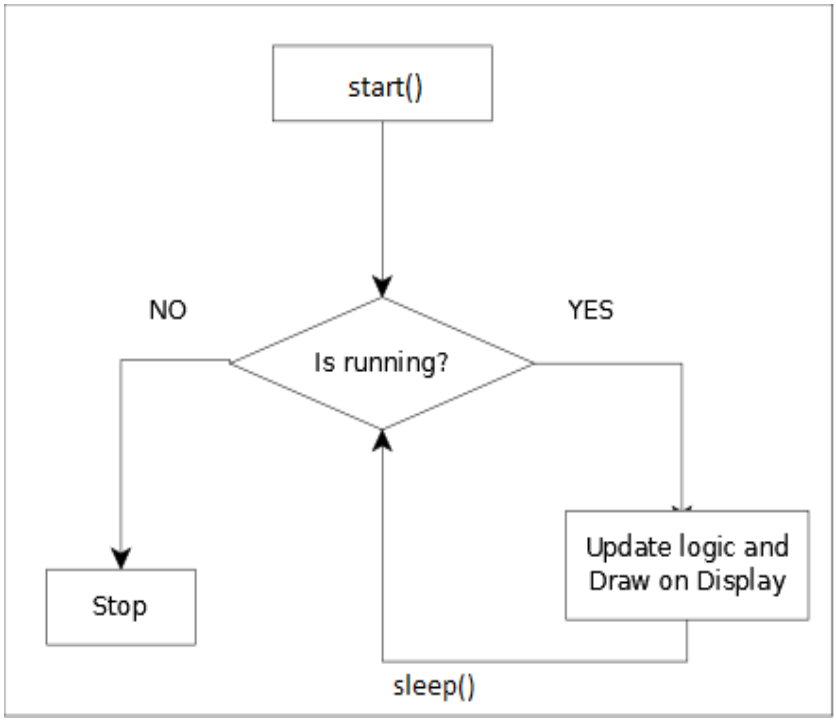
1. Να παίρνει την είσοδο τον δεδομένων από τον χρήστη.
2. Να αποθηκεύει αυτά τα δεδομένα κάπου στο πρόγραμμα.
3. Να εμφανίζει τα αποτελέσματα στην οθόνη.

Παρακάτω παραθέτουμε κάποια διαγράμματα για την καλύτερη κατανόηση αυτών των ενοτήτων.[5]

Διάγραμμα επικοινωνίας ενοτήτων



Διάγραμμα επανάληψης DisplayThread



3.4 Προσέγγιση στην Τεχνητή Νοημοσύνη

Είναι γνωστό πως έχουμε δύο ειδών προσεγγίσεις για την τεχνητή νοημοσύνη. Το ένα είδος εστιάζει στον άνθρωπο και άλλο είδος εστιάζει στην ορθολογικότητα. Η προσέγγιση που εστιάζει στην ορθολογικότητα ονομάζεται ορθολογιστική. Μία τέτοια προσέγγιση περιλαμβάνει ένα συνδυασμό μαθηματικών και τεχνολογίας, ενώ μία ανθρωποκεντρική προσέγγιση (Εστιάζει στον άνθρωπο) είναι μία εμπειρική επιστήμη με υποθέσεις και πειραματική επιβεβαίωση. Στην εφαρμογή μας υλοποιήσαμε αλγόριθμους με βάση την ανθρωποκεντρική προσέγγιση.

Πιο συγκεκριμένα χρησιμοποιήσαμε την Ανθρώπινη σκέψη. Για να καταλάβουμε αν ένα πρόγραμμα σκέπτεται σαν άνθρωπος, θα πρέπει να προσδιοριστεί το πώς σκέφτονται οι άνθρωποι. Υπάρχουν δύο τρόποι για να καταλάβουμε πως σκέπτεται ένας άνθρωπος. Ο ένας τρόπος είναι με ψυχολογικά πειράματα και ο δεύτερος είναι η καταγραφή των σκέψεων μας καθώς πραγματοποιούνται.

Με βάση την παραπάνω λογική καταγράφαμε τις σκέψεις μας και τα συμπεράσματα τα μας καθώς παίζαμε το παιχνίδι. Αφού μαζέψαμε αρκετές πληροφορίες υλοποιήσαμε τους αλγορίθμους για τις προβλέψεις και τον τρόπο παιξίματος, αλλά για αυτούς τους αλγορίθμους θα μιλήσουμε πιο αναλυτικά σε ένα από τα επόμενα κεφάλαια. Μετά την ενσωμάτωση αυτών τον αλγορίθμων στην εφαρμογή μας συνεχίσαμε την καταγραφή τον σκέψεων μας και των αποτελεσμάτων ώστε να συνεχίσουμε την βελτίωση του κώδικα.[11]

4

Wizard board game

Το Wizard είναι ένα trick-tacking παιχνίδι με κάρτες για τρεις έως 6 παίκτες. Το trick-tacking παιχνίδι είναι καρτών ή πλακιδίων με βάση το παιχνίδι ενός "χεριού" όπου επικεντρώνεται σε μια σειρά πεπερασμένων γύρων, όπου κάθε γύρος ονομάζεται τέχνασμα, τα οποία αξιολογούνται για να προσδιοριστεί ένας νικητής. Το αντικείμενο τέτοιων παιχνιδιών μπορεί τότε να είναι στενά συνδεδεμένο με τον αριθμό των κόλπων που έχουν ληφθεί Η τράπουλα του Wizard αποτελείται από μία τράπουλα με 60 κάρτες:

- 52 συνηθισμένα τραπουλόχαρτα που αποτελούνται από κάρτες τεσσάρων χρωμάτων και αριθμούς από το ένα μέχρι το δεκατρία.
- 4 κάρτες μάγων με το σύμβολο Z οι οποίες έχουν την μεγαλύτερη αξία και
- 4 κάρτες γελωτοποιών με το σύμβολο N οι οποίες έχουν την μικρότερη αξία.

Για την συγκεκριμένη έκδοση που θα υλοποιήσουμε τα χρώματα αντιστοιχούν σε μπλε(άνθρωποι), πράσινο(ξωτικά), κόκκινο(νάνοι), κίτρινο(γίγαντες). Στη συνέχεια παρουσιάζουμε ένα αντιπροσωπευτικό δείγμα της τράπουλας.

4.1 Ο Σκοπός

Το Wizard έχει σαν σκοπό, κάθε παίκτης να κάνει μία ακριβή πρόβλεψη για το πόσες μπάζες θα κερδίσει σε κάθε γύρο. Οι σωστές προβλέψεις δίνουν πόντους. Όποιος έχει τους περισσότερους πόντους στο τέλος του παιχνιδιού είναι ο νικητής.

Μπάζα : κάθε παίκτης με την σειρά του κατεβάζει μία κάρτα ανοιχτή στο κέντρο του τραπεζιού. Όλες αυτές οι κάρτες είναι μία μπάζα. Η μπάζα αυτή κερδίζεται από τον παίκτη που έριξε την πιο ισχυρή κάρτα.

4.2 Μοίρασμα των καρτών

Στο Wizard οι παίκτες παίρνουν ένα διαφορετικό αριθμό καρτών σε κάθε γύρο. Στον πρώτο γύρο κάθε παίκτης παίρνει μόνο μία κάρτα. Σε αυτόν τον γύρο θα βαθμολογηθεί μια μπάζα. Στον δεύτερο γύρο κάθε παίκτης παίρνει δύο κάρτες και βαθμολογούνται δύο μπάζες. Στον τρίτο γύρο κάθε παίκτης παίρνει 3 κάρτες, στον τέταρτο τέσσερις κτλ. μέχρι τον τελευταίο γύρο όπου μοιράζονται όλες οι κάρτες στου παίκτες. Οι κάρτες που δεν μοιράστηκαν τοποθετούνται κλειστές στο κέντρο του τραπεζιού ως απόθεμα. Στο τέλος κάθε παιχνιδιού το μοίρασμα το αναλαμβάνει ο επόμενος δεξιόστροφα παίκτης.

Όταν μοιραστεί ο σωστός αριθμός καρτών, ανοίγεται η πρώτη κάρτα του αποθέματος και τοποθετείται ανοιχτή πάνω στο απόθεμα. Αυτή η κάρτα καθορίζει το χρώμα “Ατού” για αυτόν τον γύρο. Αν ανοιχτεί Γελωτοποιός, τότε δεν υπάρχει χρώμα Ατού σε αυτόν τον γύρο. Αν ανοιχτεί Μάγος, τότε ο παίκτης που μοιράζει μπορεί να επιλέξει ποιο θα είναι το χρώμα Ατού για αυτόν τον γύρο, αφού πρώτα δει τις κάρτες του. Στον τελευταίο γύρο δεν υπάρχει Ατού, αφού δεν υπάρχουν κάρτες στο απόθεμα.

Ατού : μία κάρτα του χρώματος ατού κερδίζει οποιαδήποτε κάρτα άλλου χρώματος.

4.3 Τρόπος παιχνιδιού

4.3.1 Πρόβλεψη

Κάθε παίκτης αφού δει τις κάρτες του, θα πρέπει να προβλέψει πόσες μπάζες θα πάρει σε αυτόν τον γύρο. Ξεκινώντας από τον παίκτη αριστερά αυτού που μοίρασε, κάθε παίκτης με την σειρά του ανακοινώνει τον αριθμό με τις μπάζες που έχει προβλέψει πως θα κερδίσει αυτόν τον γύρο. Αυτός ο αριθμός δεν είναι κρυφός και οποιοσδήποτε παίκτης μπορεί να τον δει ανά πάσα στιγμή. Οι παίκτες διαδέχονται ο ένας τον άλλο δεξιόστροφα.

4.3.2 Η μάχη για τις μπάζες

Ο παίκτης στα αριστερά αυτού που μοίρασε παίζει την πρώτη κάρτα για την πρώτη μπάζα. Οι Υπόλοιποι παίκτες ακολουθούν δεξιόστροφα. Κάθε παίκτης πρέπει να παίξει μία κάρτα του ίδιου χρώματος με την πρώτη. Αν κάποιος παίκτης δεν έχει κάρτα του ίδιου χρώματος, πρέπει να ρίξει μία οποιαδήποτε άλλη κάρτα.

Οι Μάγοι και οι Γελωτοποιοί μπορούν να παιχτούν ασχέτως με το αν ο παίκτης έχει κάρτα του ίδιου χρώματος ή όχι. Αν ένας παίκτης παίξει Μάγο ή Γελωτοποιό, δεν χρειάζεται ο επόμενος παίκτης να ακολουθήσει με κάρτα του ίδιου χρώματος με την πρώτη.

Η μεγαλύτερη κάρτα κερδίζει την μπάζα. Ένας Μάγος είναι πάντα μεγαλύτερος από οποιαδήποτε άλλη κάρτα, ακόμα και από το Ατού. Ο νικητής παίρνει τη μπάζα, την τοποθετεί σε μία στοίβα μπροστά του και ξεκινάει την επόμενη μπάζα τοποθετώντας μία κάρτα ανοιχτή στο τραπέζι. Με εξαίρεση τον πρώτο γύρο όπου παίζεται μόνο μία μπάζα.

Μία μπάζα κερδίζεται με την συγκεκριμένη σειρά :

- Από τον πρώτο Μάγο που θα παιχτεί ή
- Από το μεγαλύτερο ατού αν δεν έχει παιχτεί Μάγος ή
- Την μεγαλύτερη κάρτα του ίδιου χρώματος που παίχτηκε πρώτο, αν δεν έχει παιχτεί Μάγος ή Ατού.

4.3.3 Μάγοι και Γελωτοποιοί

Αν η πρώτη κάρτα μίας μπάζας είναι Μάγος, οι παίκτες μπορούν να ακολουθήσουν παίζοντας οποιαδήποτε κάρτα επιθυμούν. Την μπάζα την κερδίζει ο παίκτης που έπαιξε τον πρώτο Μάγο. Αν η πρώτη κάρτα μίας μπάζας είναι Γελωτοποιός, ο επόμενος παίκτης μπορεί να παίξει οποιαδήποτε κάρτα επιθυμεί. Αυτή η κάρτα καθορίζει το χρώμα της μπάζας. Αν σε μία μπάζα παιχτούν μόνο Γελωτοποιοί, τότε την μπάζα την κερδίζει ο παίκτης που έριξε τον πρώτο. Αυτό μπορεί να συμβεί μόνο όταν παίζουν τρείς ή τέσσερις παίκτες.

4.4 Βαθμολόγηση

Όταν η πρόβλεψη ενός παίκτη συμπέσει με τον αριθμό από μπάζες που κέρδισε, ο παίκτης κερδίζει 20 πόντους και επιπλέον 10 πόντους για κάθε μπάζα που πήρε. Όποιος κάνει λάθος πρόβλεψη χάνει 10 πόντους για κάθε μπάζα περισσότερη ή λιγότερη από την πρόβλεψη του.

Παράδειγμα :

- 1ος γύρος :
    - Η Βασιλεία προέβλεψε ότι δεν θα πάρει την μπάζα. Είχε δίκιο και κερδίζει 20 πόντους.
    - Ο Γιάννης προέβλεψε ότι θα πάρει την μπάζα, αλλά δεν τα κατάφερε, οπότε χάνει 10 πόντους.
    - Τέλος, ο Σωτήρης προέβλεψε ότι θα πάρει την μπάζα και όντως την πήρε, οπότε παίρνει 30 πόντους.

Προβλέψεις : Βασιλεία = 0 , Γιάννης = 1 , Σωτήρης = 1

Βαθμολογία : Βασιλεία = 20 , Γιάννης = -10 , Σωτήρης = 30

- 2ος γύρος :
    - Η Βασιλεία προέβλεψε ότι θα κερδίσει και τις δύο μπάζες. Αλλά κατάφερε να κερδίζει μόνο τη μία. Επειδή η πρόβλεψη της δεν ήταν σωστή χάνει 10 πόντους αφού κέρδισε μία μπάζα λιγότερη από τη πρόβλεψη της.
    - Ο Σωτήρης προέβλεψε πως δεν θα πάρει καμία μπάζα. Είχε δίκιο και κερδίζει 20 πόντους.

- Τέλος, ο Γιάννης προέβλεψε ότι δεν θα κερδίσει καμία μπάζα, κέρδισε όμως μία οπότε χάνει 10 πόντους.

Τα αποτελέσματα του τελευταίου γύρου προστίθενται ή αφαιρούνται στην βαθμολογία των προηγούμενων γύρων. Άρα στον 2ο γύρο η βαθμολογία διαμορφώνεται ως εξής:

Προβλέψεις : Βασιλεία = 2 , Γιάννης = 0 , Σωτήρης = 0

Μπάζες : Βασιλεία = 1 , Γιάννης = 1 , Σωτήρης = 0

Βαθμολογία : Βασιλεία = 20 , Γιάννης = -20 , Σωτήρης = 50

4.5 Τέλος

Υπάρχουν 60 κάρτες στην τράπουλα. Οι παίκτες συνεχίζουν να παίζουν μέχρι να μοιραστούν όλες οι κάρτες στον τελευταίο γύρο. Με 6 παίκτες αυτός είναι ο 10ος γύρος, με 5 παίκτες είναι ο 12ος , με 4 παίκτες είναι ο 15ος και με 3 παίκτες ο 20ος. Ο τελικός γύρος βαθμολογείτε και ο παίκτης με τους περισσότερους πόντους είναι ο νικητής.

4.6 Παραλλαγές

Συν/πλην 1 : Οι βασικοί κανόνες παραμένουν ίδιοι. Η μόνη διαφορά είναι στις ανακοινώσεις. Το άθροισμα των προβλέψεων των παικτών σε κάθε γύρο πρέπει να είναι διαφορετικό από τις μπάζες που θα παιχτούν. Για παράδειγμα στον 5ο γύρο ο το άθροισμα των προβλέψεων όλων των παικτών πρέπει να είναι διαφορετικό από 5.

Κρυφή πρόβλεψη : Οι παίκτες γράφουν τις προβλέψεις τους σε ένα χαρτί. Οι προβλέψεις εμφανίζονται όταν όλες οι προβλέψεις γραφτούν. Με αυτόν τον τρόπο οι παίκτες δεν επηρεάζονται από τις προβλέψεις τον υπολοίπων.

Μυστική πρόβλεψη : Οι παίκτες γράφουν τις προβλέψεις τους σε ένα χαρτί. Μόνο όταν τελειώσει ο γύρος αποκαλύπτονται οι προβλέψεις.

Μαντεία : Στον 1ο γύρο, κάθε παίκτης έχει μόνο μία κάρτα, δεν την βλέπει αλλά την ακουμπάει στο κούτελο του έτσι ώστε όλοι να μπορούν να την δουν εκτός από τον ίδιο. Όταν

όλοι οι παίκτες δουν τις κάρτες των υπολοίπων και όχι τις δικές τους, γίνονται οι προβλέψεις. Όλοι οι υπόλοιποι κανόνες ισχύουν κανονικά και όλοι οι υπόλοιποι γύροι παίζονται επίσης κανονικά.

Μονοχρωμία (για 3 ή 4 παίκτες)**:** Κάθε παίκτης παίρνει τις κάρτες ενός χρώματος, ένα Μάγο και ένα Γελωτοποιό. Όταν παίζουν μόνο 3 παίκτες ένα χρώμα δεν αχρησιμοποίητε και επιστέφει στο κουτί. Σε αυτή την παραλλαγή δεν υπάρχουν Ατού. Όλοι οι παίκτες ανακατεύουν τις κάρτες τους, τις τοποθετούν σε μία κλειστή στοίβα μπροστά τους και τραβάνε από αυτήν 4 κάρτες για τον $1^{ο}$ γύρο. Σε κάθε έναν από τους επόμενους γύρους τραβάνε ακόμα μία κάρτα. Κατόπιν οι παίκτες πρέπει να κάνουν τις προβλέψεις τους κρυφά ή φανερά αναλόγως τι έχει συμφωνηθεί. Ισχύουν οι εξής νέοι κανόνες :

- Για να κερδηθεί μία μπάζα, το χρώμα δεν παίζει ρόλο.
- Την μπάζα την κερδίζει η κάρτα που έχει τον μεγαλύτερο αριθμό.
- Σε περίπτωση που υπάρχουν 2 κάρτες του με τον ίδιο αριθμό, κερδίζει την μπάζα αυτή που παίχτηκε πρώτη.
- Μετά από κάθε γύρο, οι κάρτες που χρησιμοποιήθηκαν ξεχωρίζονται ανάλογα με το χρώμα τους και επιστρέφονται στους ιδιοκτήτες τους.
- Οι παίκτες ανακατεύουν όλες τις κάρτες τους πριν τραβήξουν την νέα τετράδα.
- Το παιχνίδι διαρκεί 12 γύρους.[8]

5

Σχεδιασμός και Υλοποίηση της εφαρμογής

Σε αυτό το κεφάλαιο γίνεται μία αναλυτική αναφορά στον τρόπου σχεδιασμού της εφαρμογής με διαγράμματα ροής για την καλύτερη κατανόηση. Επίσης θα γίνει μία εκτενής ανάπτυξη του τρόπου υλοποίησης της εφαρμογής με αναφορές και σχεδιαγράμματα των συναρτήσεων που χρησιμοποιήσαμε. Η εφαρμογή αυτή, όπως έχουμε προαναφέρει έχει σαν σκοπό την υλοποίηση του επιτραπέζιου παιχνιδιού Wizard, με χαρακτηριστικά τεχνητής νοημοσύνης, σε εφαρμογή Android.Το όνομα της θα είναι «Wizard app». Η υλοποίηση του Wizard app ελπίζουμε πως θα βοηθήσει την διάδοση της τεχνητής νοημοσύνης και σε άλλα παιχνίδια που ο υπολογιστής θα καλείται να κάνει επιλογές βάση τυχαίων δεδομένων. Σε αυτή την έκδοση της εφαρμογής υλοποιήσαμε το παιχνίδι έτσι ώστε να παίζεται από τρεις παίκτες. Όπως κάθε άλλη εφαρμογή έτσι και το Wizard app έχει αρκετά περιθώρια βελτίωσης. Όμως για αυτά θα μιλήσουμε στο επόμενο κεφάλαιο.

5.1 Δομή εφαρμογής

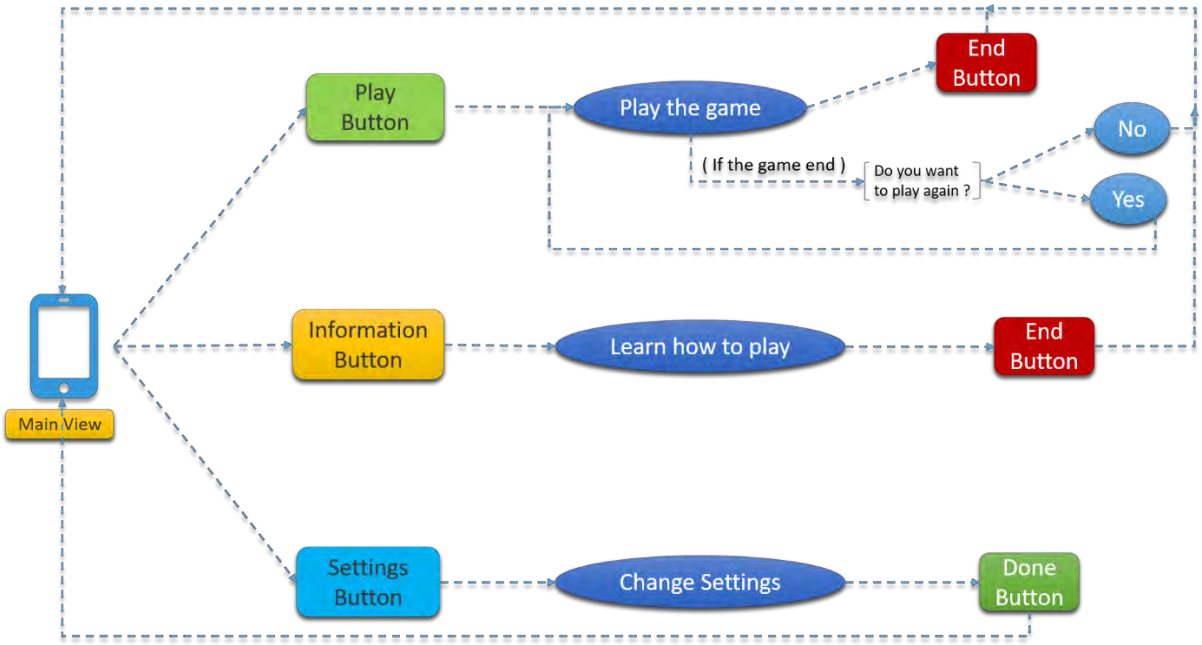
Αρχικά ο χρήστης, αφού κατεβάσει και εγκαταστήσει το Wizard app στο κινητό του τηλέφωνο ή το tablet του, ανοίγοντας την εφαρμογή θα βρεθεί αντιμέτωπος με την αρχική οθόνη, όπου θα του δίνονται τρείς επιλογές. Η μία από τις τρεις αυτές επιλογές θα είναι να διαβάσει και να κατανοήσει τους κανόνες του παιχνιδιού, η οποία του δίνεται από το πάτημα του κουμπιού **INSTRUCTIONS.**

Όταν ο χρήστης νοιώσει πως είναι έτοιμος να ξεκινήσει το παιχνίδι μπορεί με το πάτημα του κουμπιού **BACK** να επιστρέψει πίσω στην αρχική σελίδα, όπου από εκεί με την επιλογή του **PLAY** μπορεί να ξεκινήσει το παιχνίδι. Όπως προαναφέραμε στην αρχή του κεφαλαίου η

εφαρμογή υλοποιήθηκε για να παίζουν 3 παίκτες αυτό σημαίνει πως ένα παιχνίδι θα κρατήσει 20 γύρους. Αυτό μπορεί κάποιος να το θεωρήσει βαρετό ή απλά ο χρήστης να θέλει ένα πιο γρήγορο παιχνίδι.

Με το πάτημα του τρίτου κουμπιού, **SETTINGS,** δίνεται η επιλογή στον χρήστη να επιλέξει εκείνος πόσους γύρους θα έχει το παιχνίδι που θα παίξει, ακόμα θα έχει και την επιλογή να επιλέξει ένα ψευδώνυμο για το άβαταρ του. Με το πάτημα του κουμπιού **DONE** οι καινούριες επιλογές αποθηκεύονται και επιστρέφει στην αρχική σελίδα της εφαρμογής.

Παρακάτω θα δείξουμε ένα διάγραμμα ροής για την καλύτερη κατανόηση.



Όταν λοιπόν ο χρήστης αποφασίσει να παίξει θα πατήσει το play. Με αυτόν τον τρόπο θα ξεκινήσει το παιχνίδι. Όσο το παιχνίδι συνεχίζεται ο χρήστης μπορεί να επιστρέψει στην αρχική σελίδα ανά πάσα στιγμή πατώντας το κουμπί END. Όταν το παιχνίδι τελειώσει η εφαρμογή ρωτάει τον χρήστη αν θέλει να συνεχίσει να παίξει ξανά. Αν ο χρήστης επιλέξει «ναι» το παιχνίδι ξεκινάει από την αρχή. Αν ο χρήστης επιλέξει «όχι» το παιχνίδι τερματίζεται και η εφαρμογή επιστρέφει στην αρχική σελίδα. Στην επόμενη ενότητα θα σχολιάσουμε πως δουλεύει η εφαρμογή μας όταν πατηθεί το Play.[7]

5.2 Game_activity

Κάθε εφαρμογή έχει μία έως πολλές activity classes, όπως επίσης έχει και η δικιά μας. Η πιο σημαντική είναι η game_activity η οποία εκτελείται όταν ο χρήστης πατήσει το κουμπί Play.

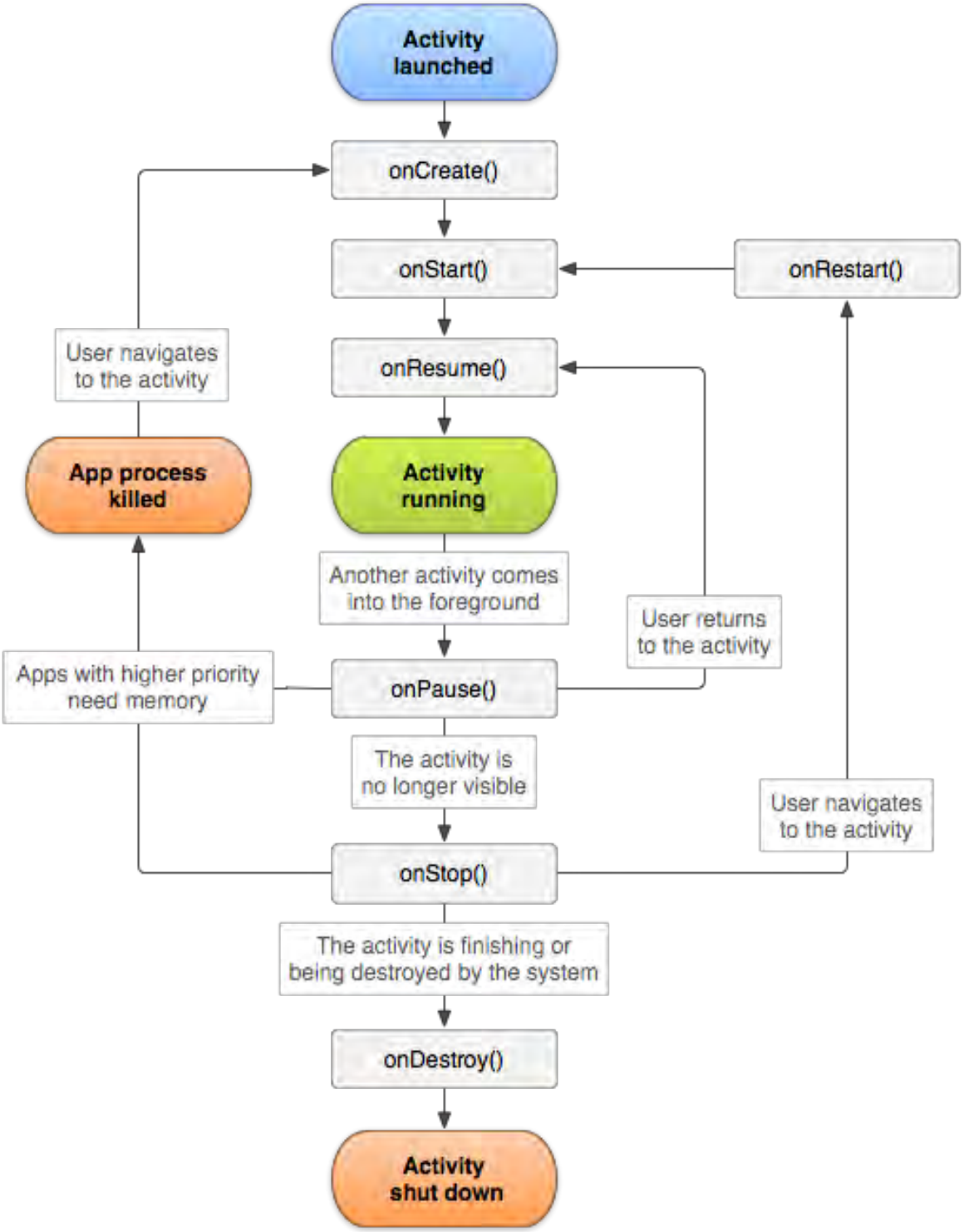
5.2.1 Τι είναι activity class

Καθώς ο χρήστης περιπλανιέται εκτός της εφαρμογής, η activity περνάει ανάμεσα από διάφορα στάδια, που ονομάζονται κύκλος ζωής της εφαρμογής. Η κλάση activity μας δίνει πρόσβαση σε αναφορές οι οποίες μας επιτρέπουν να ξέρουμε ποια κατάσταση άλλαξε. Τέτοιες αναφορές σημειώνονται όταν το σύστημα δημιουργεί, ξεκινάει, σταματάει , συνεχίζει , κάνει παύση ή καταστρέφει την διαδικασία την οποία βρίσκεται μέσα σε αυτή την δραστηριότητα.

Μέσα στις μεθόδους του κύκλου ζωής, μπορούμε να δηλώσουμε τον τρόπο με τον οποίο συμπεριφέρεται η δραστηριότητά μας όταν ο χρήστης εγκαταλείψει και επανεισέλθει στη δραστηριότητα. Η σωστή χρήση, στο σωστό χρόνο και η διαχείριση των μεταβάσεων κάνουν την εφαρμογή μας πιο ισχυρή και αποτελεσματική. Για παράδειγμα, η καλή υλοποίηση των επαναλήψεων κύκλου ζωής μπορεί να μας βοηθήσει να αποφύγουμε:

- Την διακοπή της εφαρμογής αν λάβουμε μια τηλεφωνική κλήση.
- Την κατανάλωση πολύτιμων πόρων του συστήματος όταν δεν χρησιμοποιούμε ενεργά την εφαρμογή.
- Την απώλεια της προόδου που έχουμε κάνει, εάν εγκαταλείψουμε την εφαρμογή μας και επιστρέψουμε σε αυτήν αργότερα.
- Την Διακοπή ή την απώλεια της προόδου μας όταν η οθόνη περιστρέφεται μεταξύ οριζόντιας και κατακόρυφου προσανατολισμού.

Ο κύκλος ζωής μίας δραστηριότητας φαίνεται από το παρακάτω διάγραμμα.[7]


Activity launched
onCreate()
onStart()
onRestart()
onResume()
User navigates to the activity
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

5.2.2 Δομή της game_activity

Στην game_activity προσθέσαμε κώδικα μόνο σε δύο μεθόδους επανάκλησης, στην OnCreate() και στην OnStart(). Στην OnCreate() , η δραστηριότητα μας δημιουργεί όλο το υπόβαθρο των γραφικών στοιχείων που θα χρησιμοποιήσουμε στο παιχνίδι. Στην συνέχεια δημιουργεί και καλεί το κουμπί BACK ώστε να είναι συνεχώς διαθέσιμο για τον χρήστη. Τέλος πριν τελειώσει η OnCreate() και δώσει την σειρά της στην OnStart(), δημιουργεί μία λίστα από αντικείμενα τύπου κάρτα. Αυτή η λίστα θα είναι και η τράπουλα του Wizard.

Στη συνέχεια του κύκλου ζωής της δραστηριότητας η OnStart() ξεκινάει και μαζί της ξεκινάει και η επανάληψη του παιχνιδιού. Η επανάληψη του παιχνιδιού θα έχει τα εξής χαρακτηριστικά :

- Θα ξεκινάει με τον χρήστη να παίζει πρώτος που σημαίνει πως ο παίκτης στα δεξιά του θα είναι εκείνος που μοιράζει πρώτος τις κάρτες (Dealer).
- Η βαθμολογία των παικτών πρέπει να είναι μηδέν όταν ξεκινήσει η επανάληψη.
- Όπως και μηδέν θα είναι και η μεταβλητή rounds, όπου θα αυξάνεται συν ένα κάθε φορά που τελειώνει ένας γύρος
- Κάθε γύρος θα τελειώνει όταν όλοι οι παίκτες θα ρίξουν την τελευταία κάρτα τους.
- Η επανάληψη θα τελειώσει όταν η μεταβλητή rounds περάσει το όριο που της έχουμε δώσει από της ρυθμίσεις της εφαρμογής.

Για να ξεκινήσει η επανάληψη που προαναφέραμε, καλείται η συνάρτηση dealer(). Μέσω αυτής της ξεκινάει το παιχνίδι. Αρχικά ανακατεύει την τράπουλα και μοιράζει της κάρτες σε κάθε παίκτη με την σειρά (στον 1^{ο} γύρο μία κάρτα, στον 2^{ο} δύο κτλ.). Στη συνέχεια, όταν σταματήσει το μοίρασμα ανοίγει την επόμενη κάρτα που καθορίζει το χρώμα Ατού και την εμφανίζει στο κέντρο της οθόνης, με τέτοιο τρόπο ώστε να φαίνεται πως είναι πάνω από μία στοίβα με κάρτες (οι υπόλοιπες κάρτες τις τράπουλας που δεν μοιράστηκαν). Με εξαίρεση τον τελευταίο γύρο που θα μοιραστούν όλες οι κάρτες και δεν θα υπάρχει απόθεμα.

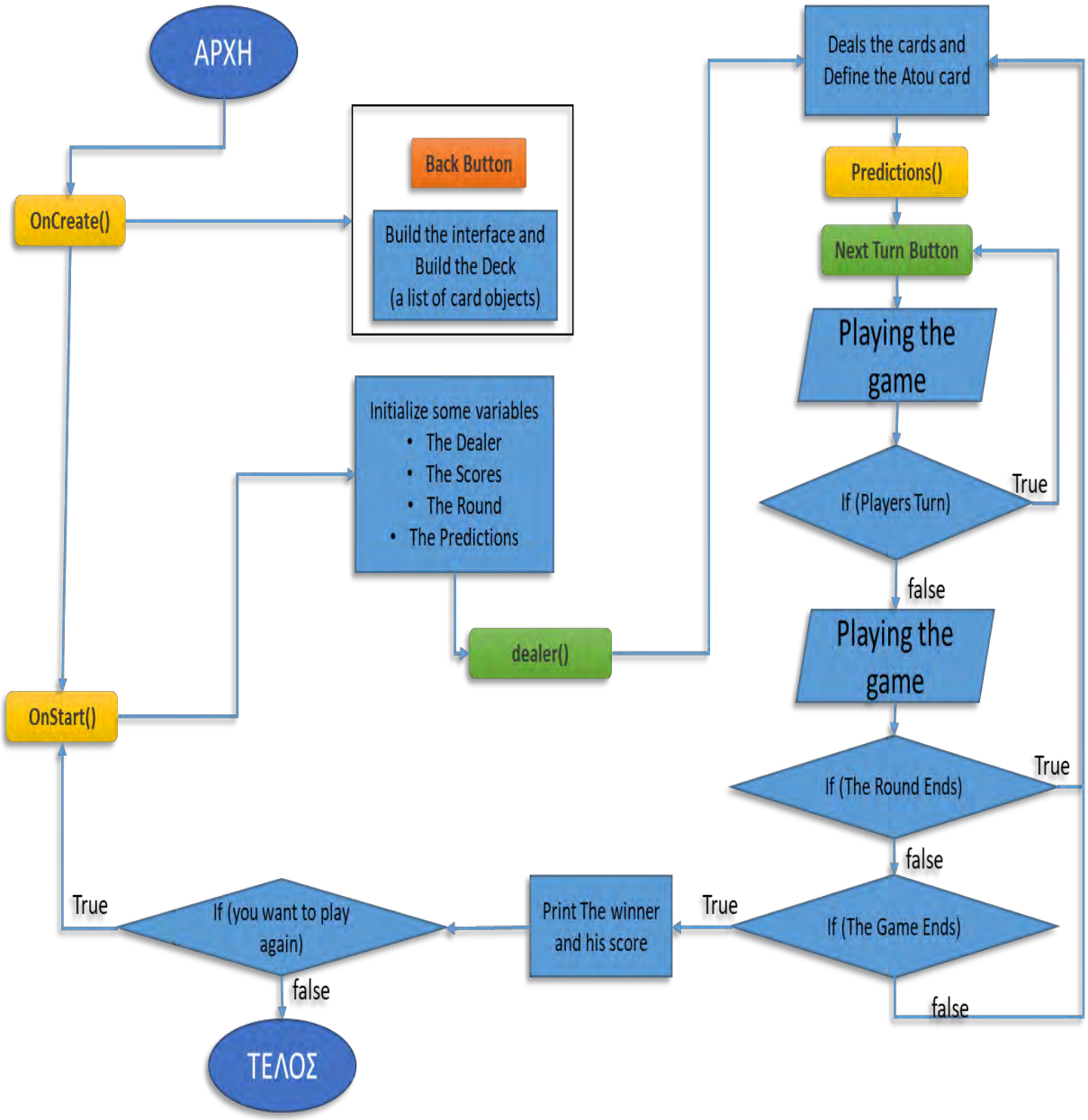
Για να ξεκινήσει η μάχη για τις μπάζες πρέπει οι παίκτες να ξεκινήσουν τις προβλέψεις. Αφού λοιπόν έχουν εμφανιστεί οι κάρτες του χρήστη στην οθόνη, εμφανίζεται ένα αναδυόμενο παράθυρο όπου ζητάει από τον χρήστη μα κάνει μία πρόβλεψη. Όταν ο χρήστης κάνει την επιλογή του και πατήσει «ok», ο υπολογιστής καλεί μία συνάρτηση για την πρόβλεψη του κάθε παίκτη. Για αυτή την όμως θα μιλήσουμε σε ένα από τα επόμενα κεφάλαια.

Αφού όλοι οι παίκτες κάνουν τις προβλέψεις τους τότε επιτρέπεται στον παίκτη να παίξει την 1η κάρτα για τον 1ο γύρο με το πάτημα του κουμπιού NEXT. Με το που ολοκληρώσει ο παίκτης την σειρά του ο επόμενος παίκτης που πρέπει να ρίξει μία κάρτα ώστε να συνεχιστεί και να τελειώσει ο 1ος γύρος. Με την σειρά του λοιπόν ο υπολογιστής αναλαμβάνει να παίξει τις απόμενες δύο κάρτες, μία για κάθε παίκτη χρησιμοποιώντας 2 διαφορετικές συναρτήσεις που θα αναφερθούμε και σε αυτές σε παρακάτω κεφάλαια.

Όταν παιχτούν τρεις κάρτες στην σειρά ολοκληρώνεται μία «μπάζα» (αφού έχουμε 3 παίκτες) και ελέγχουμε ποιος παίκτης κέρδισε αυτήν την μπάζα. Αυτός ο έλεγχος γίνεται με την συνάρτηση WinnersCard(), για την οποία θα μιλήσουμε αργότερα. Αφού προσδιοριστεί ο νικητής της μπάζας, αυξάνουμε έναν μετρητή κατά ένα ώστε να ξέρουμε πόσες μπάζες κέρδισε ο κάθε παίκτης στο τέλος του γύρου. Την καινούρια μπάζα την ξεκινάει ο παίκτης που κέρδισε την μπάζα και αυτό συνεχίζεται μέχρι να τελειώσει ο γύρος. Στο τέλος κάθε γύρου μέσω μίας συνάρτησης GetScore() υπολογίζουμε την βαθμολογία κάθε παίκτη ανάλογα της προβλέψεις τους και των αριθμό τον μπαζών που κέρδισαν. Το τέλος του γύρου καθορίζεται με το αν έχουν κάρτες οι παίκτες να παίξουν.

Αν δεν έχουν ο γύρος τελειώνει και η μεταβλητή round αυξάνεται κατά ένα. Αν η round είναι μικρότερη από την τιμή που έχουμε ορίσει σαν «πόσους γύρους θέλουμε να παίξουμε» η επανάληψη συνεχίζεται. Αν η επανάληψη τελειώσει εμφανίζεται ένα μήνυμα στην οθόνη με τον νικητή του παιχνιδιού, την βαθμολογία των υπόλοιπων παικτών καθώς και μία ερώτηση για τον χρήστη. Ο χρήστης καλείται να απαντήσει στην ερώτηση αν θα ήθελε να παίξει ξανά. Αν ο χρήστης δεν θέλει να παίξει άλλο, η εφαρμογή επιστρέφει στην αρχική σελίδα, αλλιώς αν επιθυμεί να συνεχίσει να παίζει τότε ξεκινάει μια καινούρια επανάληψη παιχνιδιού από την αρχή.

Παρακάτω θα παραθέσουμε ένα διάγραμμα για την game_activity και την βασική δομή της, χωρίς να αναφερθούμε στις συναρτήσεις που χρησιμοποιήσαμε. Για να γίνουν πιο κατανοητά αυτά που είπαμε στο διάγραμμα θα δούμε τις βασικές λειτουργείες της κλάσης μας και στα επόμενα κεφάλαια θα εξηγήσουμε καλύτερα τις σημαντικές συναρτήσεις που χρησιμοποιήσαμε. [10][7]



Στη συνέχεια θα αναλύσουμε τους αλγορίθμους που προ-είπαμε με την σειρά με την οποία καλούμε της συναρτήσεις στη ροή του παιχνιδιού.

5.3 Συνάρτηση AtouCardColor()

Ο σκοπός της συνάρτηση AtouCardColor() είναι να τσεκάρει την εγκυρότητα της κάρτας Ατού και να επιστρέφει το τελικό χρώμα Ατού. Στον κώδικα μας έχουμε δώσει ένα χαρακτηριστικό χρώμα για τα αντικείμενα κάρτες που έχουμε δημιουργήσει (Card.ColorId). Το Card.ColorId παίρνει τις τιμές κόκκινο, πράσινο, κίτρινο, μπλε και null αν αυτή η κάρτα είναι Z ή N. Ομοίως και η μεταβλητή που περιέχει το τελικό χρώμα ατού θα περιέχει μία από τις τιμές που προαναφερθήκαν, για αυτό και στον κώδικα μας τσεκάρουμε συνεχώς αν το χρώμα ατού είναι null.

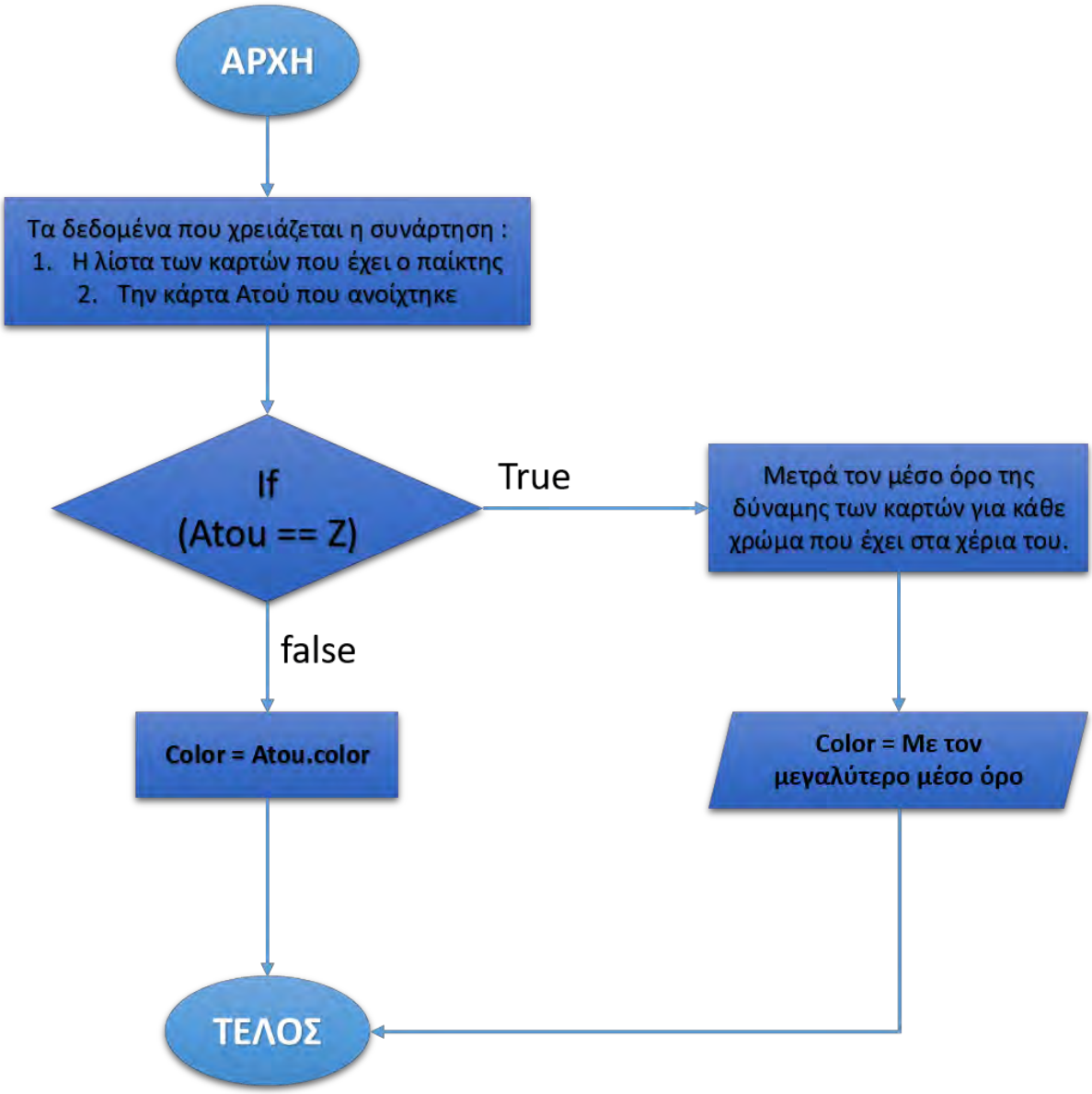
Η λογική της AtouCardColor() είναι απλή. Επιστρέφει το χρώμα ατού της κάρτας ατού αν αυτή η κάρτα δεν είναι Z. Στην περίπτωση που η κάρτα ατού είναι Z ο υπολογιστής προσπαθεί να αποφασίσει ποιο χρώμα θα επιλέξει για χρώμα ατού. Η στρατηγική που θα ακολουθήσει θα είναι όμοια με την κοινή λογική που έχουμε εμείς οι άνθρωποι, όπου στην προκειμένη περίπτωση θέλουμε να δούμε ποια ομάδα χρώματος έχει της πιο ισχυρές κάρτες, κοιτώντας πάντα τις κάρτες που έχει ο κάθε παίκτης στο χέρι του.

Αρχικά δέχεται σαν παράμετρο την λίστα με τις κάρτες του παίχτη που μοίρασε. Στη συνέχεια με μία if ελέγχει αν η κάρτα ατού είναι Z, αν ναι, τότε ακολουθεί μία διαδικασία για τον καθορισμό του χρώματος Ατού. Αν ο παίκτης που μοίρασε είναι ο χρήστης τότε εμφανίζεται ένα αναδυόμενο παράθυρο όπου ζητάει από τον χρήστη να ορίσει ένα χρώμα ατού.

Αν πάλι έχει μοιράσει ο υπολογιστής ακολουθείτε η εξής διαδικασία : ο παίκτης μετράει των μέσο όρο τις δύναμης των καρτών που έχει στην κατοχή του για κάθε ένα χρώμα χωριστά. Στη συνέχεια επιλέγει το χρώμα με τον μεγαλύτερο μέσο όρο.

Αν από την άλλη το Ατού δεν είναι Z και είναι N ή δεν υπάρχει ατού τότε εμφανίζεται ένα μήνυμα πως δεν υπάρχει χρώμα ατού σε αυτόν τον γύρο. Αλλιώς αν δεν οι ισχύουν καμία από τις παραπάνω περιπτώσεις τότε απλά θα επιστέφει το χρώμα ατού.

Παρακάτω φαίνεται το διάγραμμα της AtouCardColor().


ΑΡΧΗ
Τα δεδομένα που χρειάζεται η συνάρτηση :
1. Η λίστα των καρτών που έχει ο παίκτης
2. Την κάρτα Ατού που ανοίχτηκε
If
(Atou == Z)
True
Μετρά τον μέσο όρο της δύναμης των καρτών για κάθε χρώμα που έχει στα χέρια του.
false
Color = Atou.color
Color = Με τον μεγαλύτερο μέσο όρο
ΤΕΛΟΣ

5.4 Συνάρτηση Prediction()

Καλύπτοντας το ενδεχόμενο, για την κάρτα ατού που θα ανοίξει, να είναι Z , όπου ο παίκτης που μοίρασε καλείται να αποφασίσει το χρώμα Ατού, προχωράμε στην πρόβλεψη του κάθε παίχτη. Έχοντας επίγνωση του χρώματος Ατού και των καρτών που του έχουν μοιραστεί ο κάθε παίκτης καλείται να κάνει μία πρόβλεψη για το πόσες μπάζες θα κερδίσει σε αυτόν τον γύρο. Ο χρήστης μπορεί να επιλέξει ή να αλλάξει τον αριθμό τις πρόβλεψης του μέσω μίας ειδοποίησης που ενεργοποιείται με την έναρξη ενός καινούριου γύρου ή με το πάτημα του κουμπιού «change the prevision» αντίστοιχα.

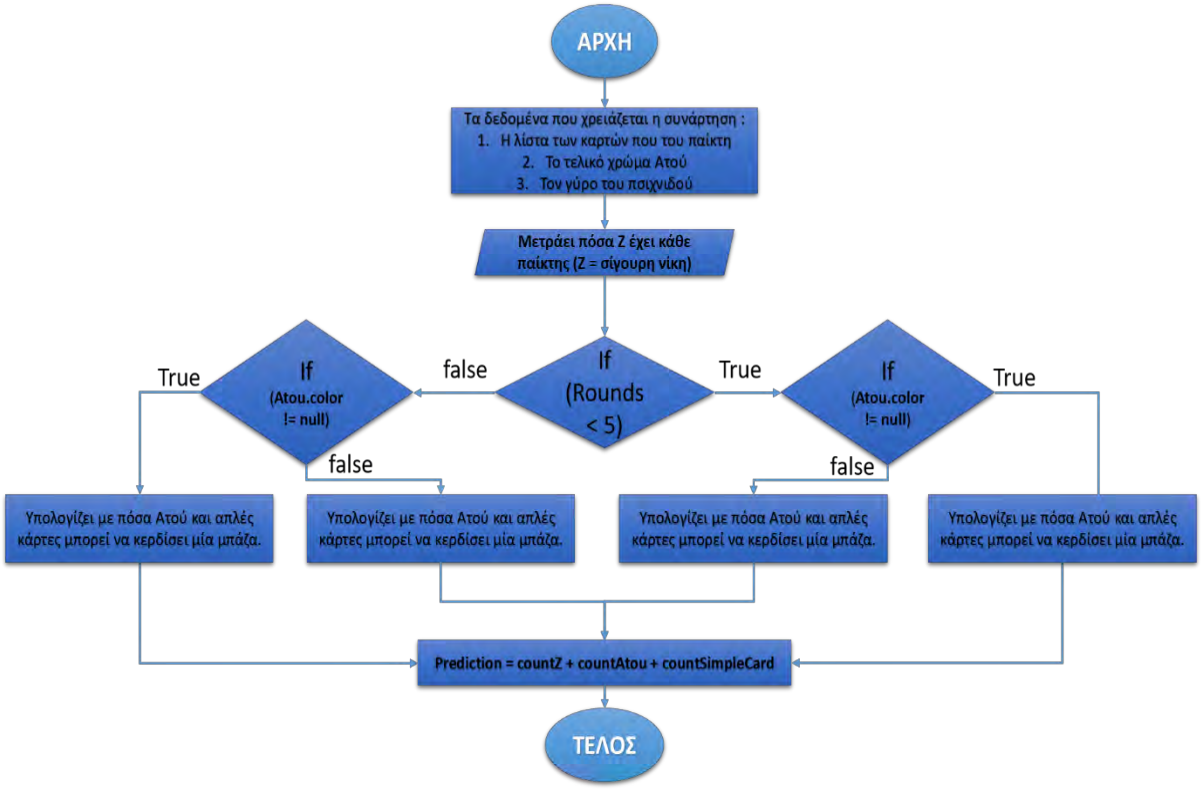
Όταν έρθει η σειρά του υπολογιστή να κάνει την πρόβλεψη πρέπει να σκεφτεί σαν έμπειρος παίχτης και να αποφασίσει βάση των δεδομένων που προαναφέραμε. Η λογική πίσω από αυτόν τον αλγόριθμο είναι να προσπαθήσουμε να προσδιορίσουμε την δύναμη της κάθε κάρτας που ο παίκτης έχει στα χέρια του. Η δύναμη της κάρτας θα καθορίζεται με το πόσο πιθανόν είναι να κερδίσει ένας παίκτης την μπάζα με την συγκεκριμένη κάρτα. Για παράδειγμα η δύναμη του Z είναι πολύ υψηλή, ενώ, η δύναμη του N είναι σχεδόν μηδαμινή. Αυτό όμως δεν σημαίνει ότι θα κερδίζει πάντα ο παίκτης που θα παίξει Z, ή ότι, η ήτα του παίκτη που θα παίξει N είναι σίγουρη.

Οι κάρτες που έχουν την μεγαλύτερη αξία είναι αρχικά οι Μάγοι (Z) , στην συνέχεια ακολουθούν οι κάρτες με το χρώμα Ατού, έπειτα ακολουθούν οι κάρτες του ίδιου χρώματος με την κάρτα που παίχτηκε πρώτη, μετά ακολουθούν οι υπόλοιπες κάρτες και τέλος οι Γελωτοποιοί (N). Επίσης σημαντική μεταβλητή για τη συνάρτηση της δύναμης μιας κάρτας είναι ο γύρος στον οποίο βρισκόμαστε. Αυτό συμβαίνει διότι όσο πιο λίγες κάρτες έχει ο αντίπαλος τόσο μικρότερες είναι οι πιθανότητες να έχει πιο ισχυρή κάρτα.

Η συνάρτηση που υλοποιήσαμε με αυτήν την λογική παίρνει σαν παραμέτρους τρία δεδομένα, τον γύρο τον οποίο βρίσκεται το παιχνίδι, την λίστα των καρτών που ο εκάστοτε παίκτης έχει στα χέρια του και το τελικό χρώμα ατού που έχουμε βρει. Αρχικά υπολογίζει πόσους Μάγους(Z) έχει ο παίκτης στα χέρια του, διότι θεωρούνται πολύ δυνατές κάρτες. Στη συνέχεια τσεκάρει σε ποιόν γύρο βρίσκεται το παιχνίδι. Αν δηλαδή το παιχνίδι βρίσκεται στους $1^{ους}$ γύρους ή όχι. Και στα δύο αποτελέσματα αυτής της if η διαδικασία υπολογισμού είναι ίδια απλά αλλάζουν κάποιες τιμές μεταβλητών. Για παράδειγμα, την περίπτωση που το Ατού είναι το χρώμα κόκκινο και ένας παίκτης έχει την κάρτα 2 κόκκινο. Έχει πολύ καλές πιθανότητες να κερδίσει την μπάζα εάν παίζουμε τον $1^{ο}$ γύρο του παιχνιδιού, όπου κάθε παίκτης έχει μία

κάρτα, σε αντίθεση με την περίπτωση που είμαστε στον 6ο γύρο, όπου κάθε παίκτης έχει 6 κάρτες στο χέρι του. Στην συνέχεια της συνάρτησης, ανεξάρτητα από το αν ήμαστε στους 1ους γύρους ή όχι , ελέγχεται αν υπάρχει χρώμα ατού για τον συγκεκριμένο γύρο και αναλόγως αλλάζουν οι μεταβλητές μας. Όμως ο κώδικας που χρησιμοποιήσαμε είναι παρόμοιος για την κάθε περίπτωση .

Θα αναλύσουμε περισσότερο αυτό το κομμάτι του κώδικα μετά από το διάγραμμα που δείχνει την ροή της συνάρτησης Prediction().



Όπως βλέπουμε από το διάγραμμα ροής της Prediction() έχουμε τέσσερα παρόμοια κομμάτια κώδικα. Οι μεταβλητές που αλλάζουν όπως προαναφέραμε είναι εκείνες που καθορίζουν την δύναμη της κάθε κάρτας. Άρα σε κάθε μία περίπτωση δεχόμαστε μία διαφορετική επιτρεπτή δύναμη για κάθε κατηγορία κάρτας (Ατού ή απλή κάρτα). Πριν το τέλος της συνάρτησης περνάμε τα αποτελέσματα στην μεταβλητή prediction την οποία περιέχει την τελική πρόβλεψη του παίκτη.

Αξίζει επίσης να αναφέρουμε πως έχουμε δώσει μία «προσωπικότητα» στον κάθε παίκτη που αντιπροσωπεύεται από τον υπολογιστή. Ο παίκτης αριστερά του χρήστη έχει το χαρακτηριστικό του πιο επιθετικού παίκτη κάνοντας πιο τολμηρές προβλέψεις. Σε αντίθεση με τον παίκτη στα δεξιά του χρήστη ο οποίος έχει τα χαρακτηριστικά του πιο έμπυρου παίκτη που κάνει προβλέψεις χωρίς να παίρνει ρίσκα.[6]

5.5 Playing a Card

Όταν οι προβλέψεις για τον κάθε παίκτη τελειώσουν, τότε ο επόμενος παίκτης αριστερά αυτού που μοίρασε πρέπει να παίξει 1ος. Όταν ο πρώτος παίκτης που παίζει είναι ο χρήστης αρκεί να επιλέξει μία επιτρεπτή κάρτα και να πατήσει το κουμπί Next. Αργά ή γρήγορα όμως, θα φτάσουμε στο σημείο όπου θα πρέπει να παίξει ο υπολογιστής. Πρέπει να σημειωθεί πως δεν έχουμε επιλέξει κάποιο μοτίβο συμπεριφοράς για τον τρόπο παιξίματος των παικτών που χειρίζεται ο υπολογιστής.

Ήρθε επιτέλους η στιγμή όπου ο υπολογιστής θα πρέπει να κάνει μία κίνηση. Πριν όμως από αυτήν την κίνηση θα αναρωτηθεί αν παίζει 1ος ή όχι. Αν όντως παίζει πρώτος τότε θα τρέξει η συνάρτηση PlayingFirst() , αλλιώς θα τρέξει η PlayingMiddle() η οποία έχει εντελώς διαφορετική λογική από την πρώτη.

5.5.1 Συνάρτηση PlayingFirst()

Παίζοντας πρώτος ένας παίκτης συνήθως σημαίνει πως δεν βρίσκεται σε πλεονεκτική θέση. Ο λόγος είναι ότι οι παίκτες που θα παίξουν μετά από αυτόν θα έχουν περισσότερα δεδομένα για να παίξουν μια κάρα (λόγω της κάρτας που ήδη έριξε ο 1ος). Η ιδανική θέση συνήθως είναι να παίζει τελευταίος , έτσι θα έχεις όσο πιο πολλά δεδομένα γίνεται. Αλλά παίζοντας ένας παίκτης πρώτος έχει την δυνατότητα, ρίχνοντας μία μικρή κάρτα ατού να αναγκάσει τους επόμενους παίκτες να ρίξουν ατού και να τους χαλάσει τα σχέδια.

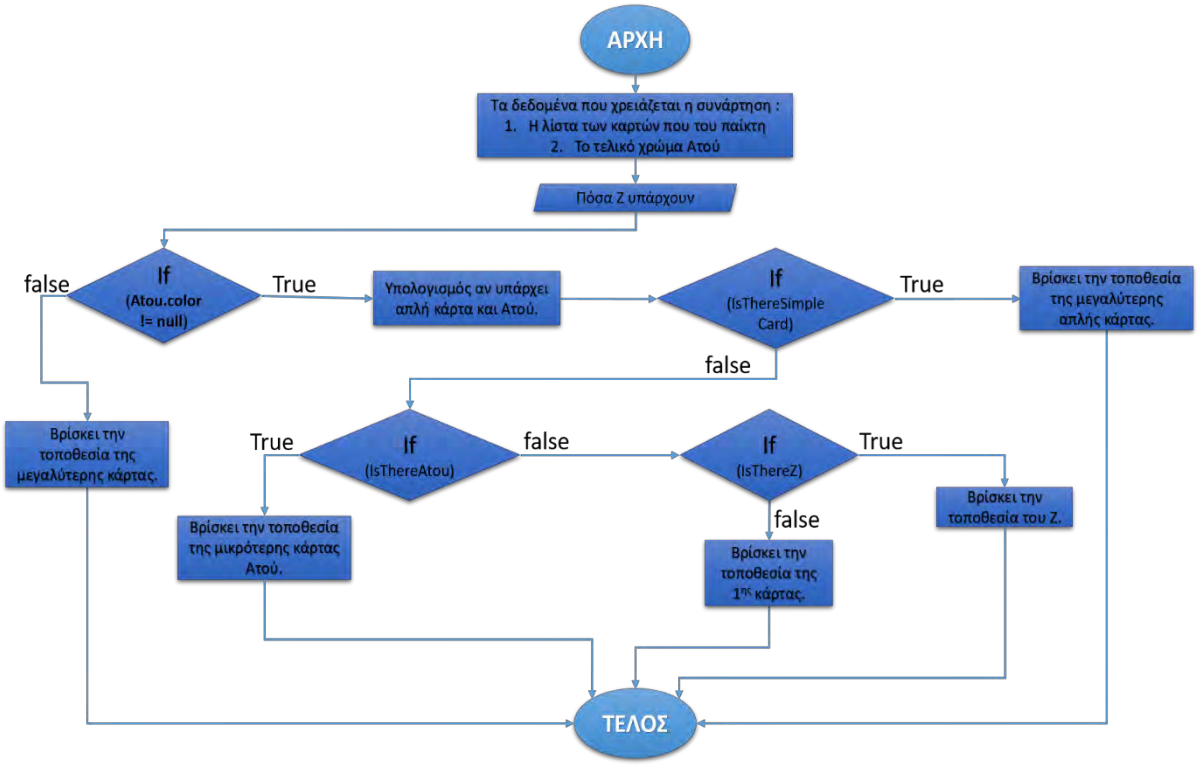
Ο γενικός κανόνας που πρέπει να ακολουθήσει ένας παίκτης παίζοντας πρώτος είναι να ρίχνει κάρτες μικρής αξίας ώστε να αποφύγει την απρόσμενη νίκη μπαζών. Το πώς ένας παίκτης θα ηγηθεί της μπάζας εξαρτάτε από το αν προσπαθεί να κερδίσει την μπάζα ή απλά να ξεφορτωθεί

μία κάρτα. Αυτή η επιλογή συνήθως εξαρτάτε από το αν ο παίκτης έχει φτάσει τον αριθμό των επιθυμητών μπαζών που έχει προβλέψει ή όχι.

Παίζοντας πρώτος είναι προτιμότερο να κρατήσεις τις επιλογές σου ανοιχτές και να αφήσεις τους αντιπάλους σου να αναρωτιούνται. Για αυτό τον λόγο οι Μάγοι και οι Γελωτοποιοί που σου προσφέρουν τεράστια ευελιξία, θα ήταν καλό να μην παίζονται σαν πρώτες κάρτες. Ο καλύτερος τρόπος να διαχειριστούμε καταστάσεις κρίσεις που προκύπτουν κατά την διάρκεια του παιχνιδιού, είναι να χρησιμοποιούμαι αυτές τις κάρτες.

Η συνάρτηση PlayingFirst() παίρνει σαν παραμέτρους την λίστα των καρτών που ο παίκτης έχει στην κατοχή του, το τελικό χρώμα Ατού και επιστρέφει την θέση στην λίστα όπου βρίσκεται το αντικείμενο κάρτα που θα παίξει ο παίκτης. Στην συνέχεια υπολογίζει πόσους Μάγους έχει ο παίκτης στις κάρτες του. Αν χρώμα Ατού είναι null επιστρέφει την τοποθεσία μίας μεγάλης σε αξία κάρτας. Αλλιώς βρίσκει αν υπάρχει Ατού και αν επίσης υπάρχει απλή κάρτα. Αν υπάρχει απλή κάρτα επιστρέφει την τοποθεσία της μεγαλύτερης σε αξία κάρτας, αλλιώς αν υπάρχει ατού επιστρέφω την μικρότερη δυνατή κάρτα ατού. Αν δεν υπάρχει ούτε ατού τότε παίζω Μάγο αν υπάρχει, αλλιώς όποια κάρτα έχει απομείνει.

Παρακάτω παραθέτουμε ένα διάγραμμα ροής για αυτή την PlayingFirst().[6]



5.5.2 Συνάρτηση PlayingMiddle

Μέχρι τώρα είδαμε ποια στρατηγική είναι η βέλτιστη όταν ο παίκτης παίζει πρώτος. Τώρα μένει να δούμε ποια στρατηγική θα ακολουθήσει ο παίκτης όταν δεν παίζει πρώτος. Η γενική στρατηγική σε αυτή την περίπτωση είναι πιο ξεκάθαρη αφού ο επόμενος παίκτης πρέπει να παίξει κάρτα ίδιου χρώματος με την πρώτη κάρτα που παίχτηκε.

Αν ο παίκτης δεν παίζει τελευταίος η στρατηγική που ακολουθεί είναι να ρίχνει την αμέσως επόμενη σε δύναμη κάρτα που έχει στην διάθεση του. Αν δεν έχει κάποια κάρτα που μεγαλύτερης αξίας τότε προσπαθεί να ξεφορτωθεί την χαμηλότερη σε αξία κάρτα. Φυσικά θα κρατήσει τους Μάγους και τους Γελωτοποιούς ώστε να μην βρεθεί σε μία αμήχανη κατάσταση όπως αναφέραμε στην προηγούμενη ενότητα.

Όταν ο παίκτης παίζει τελευταίος ακολουθεί περίπου την ίδια στρατηγική με αυτήν που αναφέραμε στην προηγούμενη παράγραφο. Δύο πράγματα όμως αλλάζουν. Το πρώτο είναι πως αν δεν έχει μεγαλύτερης αξίας κάρτα, αντί να ξεφορτωθεί μια μικρή αξίας κάρτα, θα παίξει Μάγο ώστε να εξασφαλίσει από νωρίς την πρόβλεψη του και στην συνέχεια να παίζει παθητικά. Το δεύτερο θα είναι ότι ο παίκτης θα προσπαθεί να νικήσει την ισχυρότερη κάρτα που έχει παιχτεί μέχρι εκείνη την στιγμή και όχι μόνο την πρώτη.

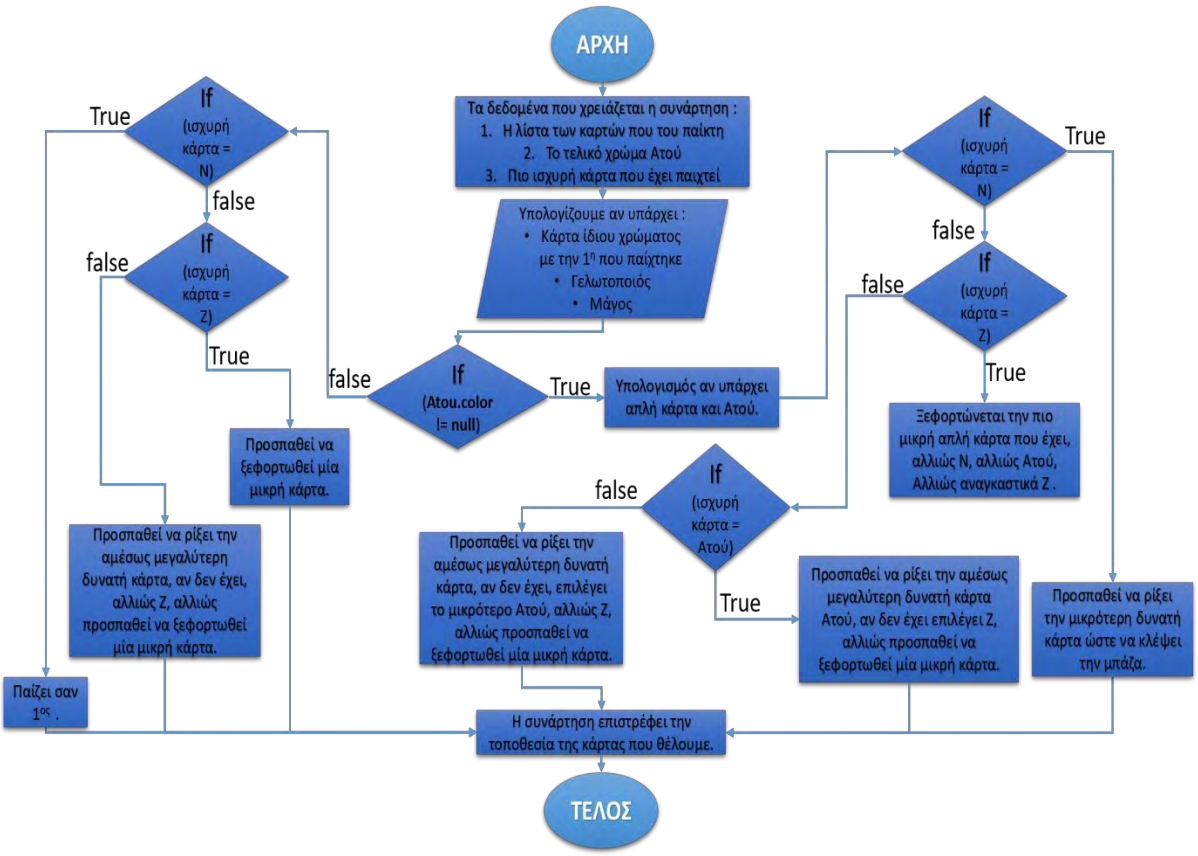
Η συνάρτηση ξεκινάει με 3 δεδομένα. Πρώτον την λίστα των καρτών που έχει ο παίκτης στην κατοχή του , δεύτερον το τελικό χρώμα Ατού και τρίτον την πιο ισχυρή κάρτα που έχει παιχτεί μέχρι στιγμής. Αρχικά υπολογίζει αν υπάρχει κάρτα του ίδιου χρώματος με αυτήν που παίχτηκε πρώτη και στη συνέχεια υπολογίζει αν ο παίκτης έχει στην κατοχή του έστω μία κάρτα Z ή N.

- Εάν δεν υπάρχει ατού και :
    - Η ισχυρή κάρτα που έχει παιχτεί είναι N τότε παίζει σαν να παίζει πρώτος. Βλέπε κεφάλαιο 5.5.1.
    - Εάν η ισχυρή κάρτα που έχει παιχτεί είναι Z τότε προσπαθεί να ξεφορτωθεί μια μικρής αξίας κάρτα, αλλιώς επιλέγει να ρίξει :
        - Την αμέσως μεγαλύτερη επιτρεπόμενη κάρτα, αν δεν έχει
        - Μάγο, αλλιώς

  Εάν υπάρχει ατού και :
    - Η ισχυρή κάρτα που έχει παιχτεί είναι N τότε παίζει την μικρότερη δυνατή κάρτα ώστε να κλέψει την μπάζα.

- Εάν η ισχυρή κάρτα που έχει παιχτεί είναι Z τότε προσπαθεί να ξεφορτωθεί μια μικρής αξίας κάρτα, αλλιώς επιλέγει να ρίξει :
  - Γελωτοποιό, αν δεν έχει
  - Κάρτα ατού, αλλιώς
  - Αναγκαστικά Μάγο.
- Εάν κανένα από τα προηγούμενα δύο δεν ισχύουν και η ισχυρή κάρτα που έχει παιχτεί είναι χρώματος ατού , προσπαθεί να ρίξει:
  - την αμέσως μεγαλύτερη σε δύναμη κάρτα ατού, αλλιώς
  - Μάγο , αλλιώς
  - προσπαθεί να ξεφορτωθεί μια μικρής αξίας κάρτα.
- Εάν κανένα από τα παραπάνω δεν ισχύει, ρίχνει :
  - Την αμέσως μεγαλύτερη επιτρεπόμενη κάρτα, αν δεν έχει
  - Το μικρότερο δυνατό ατού, αλλιώς
  - Μάγο , αλλιώς
  - προσπαθεί να ξεφορτωθεί μια μικρής αξίας κάρτα.

Παρακάτω θα δείξουμε με ένα διάγραμμα ροής για την PlayingMiddle() όσα προαναφέραμε, ώστε να γίνουν πιο κατανοητά. [6]



5.6 *Συνάρτηση WinnerCard()*

Μία σημαντική συνάρτηση που υλοποιήσαμε είναι ο καθορισμός της πιο δυνατής κάρτα που έχει παιχτεί μέχρι τώρα και συνάμα τον παίχτη που κέρδισε την μπάζα. Αυτή είναι η WinnerCard().

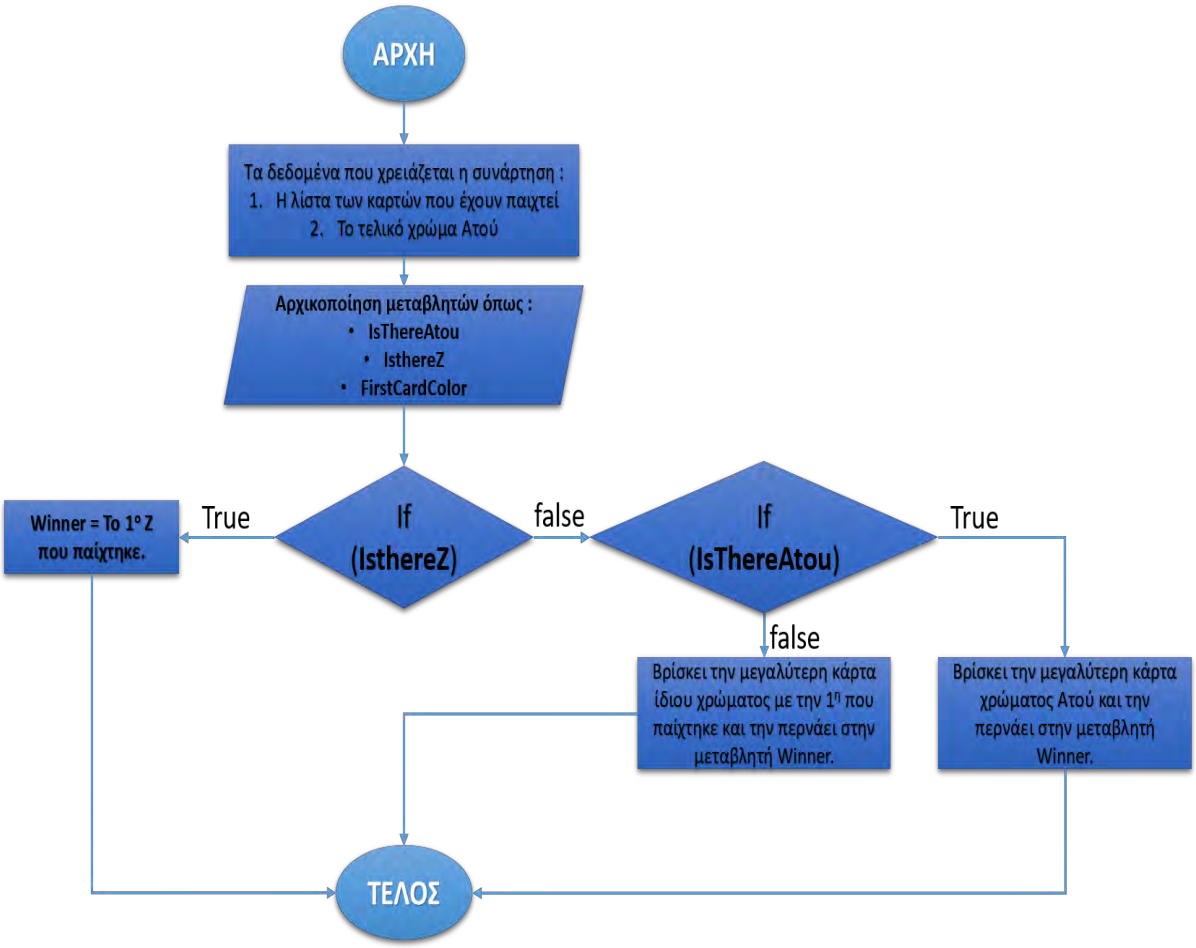
Η WinnerCard() παίρνει σαν παραμέτρους την λίστα των καρτών που έχουν παιχτεί και το τελικό χρώμα ατού. Στην συνέχεια η συνάρτηση αυτή εκτελεί μία επανάληψη και ακολουθεί την εξής διαδικασία :

2 Όταν βρει το πρώτο Μάγο σταματάει την επανάληψη και επιστρέφει την κάρτα αυτή.

3 Αν δεν υπάρχει Μάγος, και υπάρχει κάρτα χρώματος ατού, βρίσκει και επιστρέφει την μεγαλύτερη σε αξία κάρτα ατού.

4 Αν δεν υπάρχει μάγος ή ατού, βρίσκει και επιστρέφει την μεγαλύτερη σε αξία κάρτα ίδιου χρώματος με της πρώτης κάρτα που παίχτηκε.

Για την καλύτερη κατανόηση παραθέτουμε το παρακάτω διάγραμμα.



6

Συμπεράσματα και Μελλοντικές Επεκτάσεις

6.1 Συμπεράσματα

Ολοκληρώνοντας το Wizard app έπρεπε να επιβεβαιώσουμε πως όλα αυτά που αναλύσαμε στα προηγούμενα κεφάλαια λειτουργούν έτσι όπως θέλαμε. Μετά από πολλά τρεξίματα και βελτιώσεις του κώδικα της εφαρμογής προέκυψαν μερικά ενδιαφέροντα συμπεράσματα τα οποία αφορούν τους αλγόριθμους τεχνητής νοημοσύνης .

- Για τον αλγόριθμο όπου υπολογίζουμε την πρόβλεψη του κάθε παίκτη, παρατηρήσαμε μια μεγάλη βελτίωση. Αυτή η βελτίωση αφορά την αξιοπιστία των αποτελεσμάτων του αλγορίθμου που στην περίπτωση μας είναι μεγαλύτερη ακρίβεια στις προβλέψεις. Το συμπέρασμα αυτό το πήραμε μετά από πολλά τρεξίματα της εφαρμογής και παρατήρηση της συμπεριφοράς του αλγορίθμου κάτω από ιδικές περιπτώσεις του παιχνιδιού.
- Για τον αλγόριθμο που χρησιμοποιούμαι για να παίξει ο υπολογιστής τις σωστές κάρτες , παρατηρήσαμε με ικανοποίηση πως πληρεί όλες τις προδιαγραφές που θέλαμε. Όσο περισσότερο παίζαμε το παιχνίδι, τόσο περισσότερο βελτιώναμε και το συγκεκριμένο κομμάτι του κώδικα μας , το οποίο έφτασε σε ένα σημείο όπου πολλές φόρες ανάγκασε τον χρήστη να βρεθεί σε δύσκολη θέση ή να του κλέψει την μπάζα. Τέλος ένα ακόμα συμπέρασμα που βγάλαμε για αυτόν τον αλγόριθμο είναι πως έκανε το παιχνίδι αρκετά πιο ανταγωνιστικό. Αυτό το παρατηρήσαμε μετά από αρκετά παιχνίδια όπου ο νικητής δεν ήταν πάντα ο ίδιος.
- Τέλος είδαμε μεγάλη διαφορά μετά την εισαγωγή της συνάρτησης όπου καθορίζει το τελικό χρώμα ατού. Στην περίπτωση που ο παίκτης που αποφασίζει το χρώμα ατού

παρατηρήσαμε πως κερδίζει περισσότερες μπάζες από ότι πριν την χρήση αυτής της συνάρτησης.

Είναι σημαντικό να σημειωθεί πως η ενσωμάτωση τεχνικών τεχνητής νοημοσύνης επιτεύχθηκε σε υποτυπώδη βαθμό.

Όλα αυτά ήταν αρκετά θετικά αποτελέσματα, όμως πάντα υπάρχει χώρος για αναβαθμίσεις και βελτιώσεις τις οποίες θα αναλύσουμε στην επόμενη ενότητα.

6.2 Μελλοντικές επεκτάσεις

Κάθε εφαρμογή έχει περιθώρια βελτίωσης όσο καλή και αν είναι, ακόμα και αν δεν χρειάζεται πρέπει να κρατήσει το ενδιαφέρον του χρήστη σε υψηλό επίπεδο. Σε αυτή την ενότητα θα αναπτύξουμε κάποιες ιδέες για το πώς μπορεί να βελτιωθεί η δικιά μας εφαρμογή στο μέλλον.

- **Προβλέψεις** – Μία σημαντική προσθήκη στον κώδικα των προβλέψεων θα μπορούσε να είναι άλλη μία παράμετρος που θα επηρεάζει το αποτέλεσμα. Αυτή η παράμετρος θα είναι η προβλέψεις των προηγούμενων παικτών. Αφού ολοκληρώσουμε αυτή την αναβάθμιση του κώδικα μπορεί να δημιουργηθεί χώρος και για άλλη μία μεταβλητή στην εξίσωση. Θα μπορεί ο υπολογιστής να δημιουργήσει ένα προφίλ συμπεριφοράς προβλέψεων για τον κάθε χρήστη. Δηλαδή κάτω από ποιες συνθήκες ο παίκτης αποτυγχάνει να κερδίσει την πρόβλεψη που έκανε.
- **Παίξιμο καρτών** – Η προσθήκη ενός ελέγχου για τη συνεχή παρακολούθηση των μπαζών που κερδίζει ο κάθε παίκτης. Αυτός ο έλεγχος θα γίνεται για να αλλάζει ο παίκτης την συμπεριφορά του στο παιχνίδι ανάλογα με το πόσες μπάζες έχει κερδίσει. Για παράδειγμα όταν ο παίκτης έχει προβλέψει πως θα κερδίσει 3 μπάζες και έχει ήδη κερδίσει 3, ενώ του μένουν ακόμα 2 κάρτες, πρέπει να αλλάξει την στρατηγική του και να προσπαθήσει να μην πάρει άλλη μπάζα μέχρι να τελειώσει ο γύρος. Ακόμα μία ιδέα είναι ο υπολογιστής να προσπαθήσει να φταίξει ένα προφίλ για τον αντίπαλο παίκτη ώστε να μπορέσει να καταλάβει αν υπάρχει κάποιο μοτίβο στις κινήσεις του. Για παράδειγμα με ποια σειρά ρίχνει τις κάρτες του.
- **Γραφικά** – Ένα περιβάλλον πιο φιλικό προς τον χρήστη. Αυτό θα επιτευχθεί με την χρήση animation , ήχου ακόμα και την χρήση 3D γραφικών.

- **Περισσότεροι παίκτες –** Εισαγωγή μιας επιλογής στις ρυθμίσεις τις εφαρμογής όπου θα επιτρέπει στον χρήστη να αλλάζει τον αριθμό των παικτών που ανταγωνίζεται.
- **On-line** – Δυνατότητα στον χρήστη να μπορεί να παίξει online με άλλα άτομα.

7

Βιβλιογραφία

[1] Srujan Gopu, “Experimental Studies of Android APP Development for Smart Chess Board System”, M.S. thesis, Dep. of Comp. Science, Western Kentucky University, Bowling Green, Kentucky , 2013

[2] Dinis Félix, “Artificial Intelligence Techniques in Games with Incomplete Information: Opponent Modelling in Texas Hold'em”, M.S. thesis, Dep. El. and Comp. Eng. , Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2008

[3] Heeres, G. (2017). *Intro To Game Development (Part 1): The Game Loop – HEERES Online*. [online] Blog.heeresonline.com. Available at: http://blog.heeresonline.com/2015/01/intro-to-game-development-part-1-the-game-loop/ [Accessed 10 Oct. 2017].

[4] Java Code Geeks. (2017). *Android Game Development – The Game Loop*. [online] Available at: https://www.javacodegeeks.com/2011/07/android-game-development-game-loop.html [Accessed 10 Oct. 2017].

[5] Durov, P. (2017). *Android Basic Game Loop - CodeProject*. [online] Codeproject.com. Available at: https://www.codeproject.com/Articles/827608/Android-Basic-Game-Loop [Accessed 10 Oct. 2017].

[6] Wizardcards.com. (2017). *‚52.56.141.155, - Wizard Card Game Community Forums - powered by XMB*. [online] Available at: http://www.wizardcards.com/portal.php [Accessed 10 Oct. 2017].

[7] Developer.android.com. (2017). *Android Developers*. [online] Available at: https://developer.android.com/index.html [Accessed 10 Oct. 2017].

[8] Fun-adult-party-games.com. (2017). *Wizard Card Game Rules, Reviews, Tips and Printables!*. [online] Available at: http://www.fun-adult-party-games.com/wizard-card-game.html [Accessed 10 Oct. 2017].

[9] Wikipedia. (2017). *Main Page*. [online] Available at: https://en.wikipedia.org/wiki/Main_Page [Accessed 10 Oct. 2017].

[10] Developer.android.com. (2017). *Download Android Studio and SDK Tools | Android Studio*. [online] Available at: https://developer.android.com/studio/index.html [Accessed 10 Oct. 2017].

[11] Anon, (2017). [ebook] Available at: http://ai.uom.gr/aima/Samples/AIMA_chapter01.pdf [Accessed 14 Oct. 2017].