

OBJECT-ORIENTED PROGRAMMING HY-252

PROJECT 2017-18

Τσαϊνης Κυριάκος

AM: 3170

Σχεδιασμός της Εργασίας

Η υλοποίηση της εργασίας θα βασιστεί πάνω στο μοντέλο MVC – model, view, controller ούτως ώστε να είναι χωρισμένη σε τμήματα η κάθε λειτουργία του προγράμματός μας. Ουσιαστικά στο model θα υπάρχουν οι κλάσεις που θα δημιουργούν τα αντικείμενα που θα χρειαστούμε για να υλοποιηθεί το παιχνίδι, στην view θα φτιάξουμε το περιβάλλον το οποίο θα εμφανίζεται όταν τρέχουμε την εφαρμογή και ο Controller θα συνδέει τα δύο προηγούμενα.

Package Model

Card Interface and Other Classes for Cards

Αρχικά φτιάχνοντας τη διεπαφή Card μας δίνεται η δυνατότητα να προσπελάσουμε τα δεδομένα χωρίς να πρέπει να ορίσουμε αν μία κάρτα είναι απλή η ειδική.

To interface αυτό μας παρέχει τις εξής μεθόδους:

1. `public int getValue();`
Returns the value of a card.
2. `public void setValue(int value);`
Sets the value of a card.
3. `public int getPoints();`
Returns the points of a card.
4. `public void setPoints(int points);`
Sets the points of a card.
5. `public String toString();`
Returns a String Representation of a Card.

Στη συνέχεια έχουμε την SimpleCard και την SpecialCard που υλοποιούν την Card.

Class SimpleCard

Εδώ θα αναφέρουμε τα attributes και τις υπόλοιπες μεθόδους που έχει η κλάση αυτή (εκτός από αυτές που υλοποιεί μέσω της διεπαφής Card).

Τα attributes:

- 1) `private CardType col; //The type of a Simple Card.`
- 2) `private int value; //The value of a Simple Card.`
- 3) `private int points; //The points of a Simple Card.`

Οι υπόλοιπες μέθοδοι :

1. `public CardType getType();`
Get the type of a simple Card.
2. `public void setType(CardType type);`
Sets the type of a simple Card.

Class CardType

Είναι μια enum class που περιέχει τους τύπους των καρτών (KNOSSOS, MALIA, FESTOS, ZAKROS).

Class SpecialCard

Εδώ θα αναφέρουμε τα attributes και τις υπόλοιπες μεθόδους που έχει η κλάση αυτή (εκτός από αυτές που υλοποιεί μέσω της διεπαφής Card).

Τα attributes:

- private int value; //The value of a Special Card.
- private int points; //The points of a Special Card.
- private CardType type; //The type of a Special Card
- private String name; //The name of the special card.

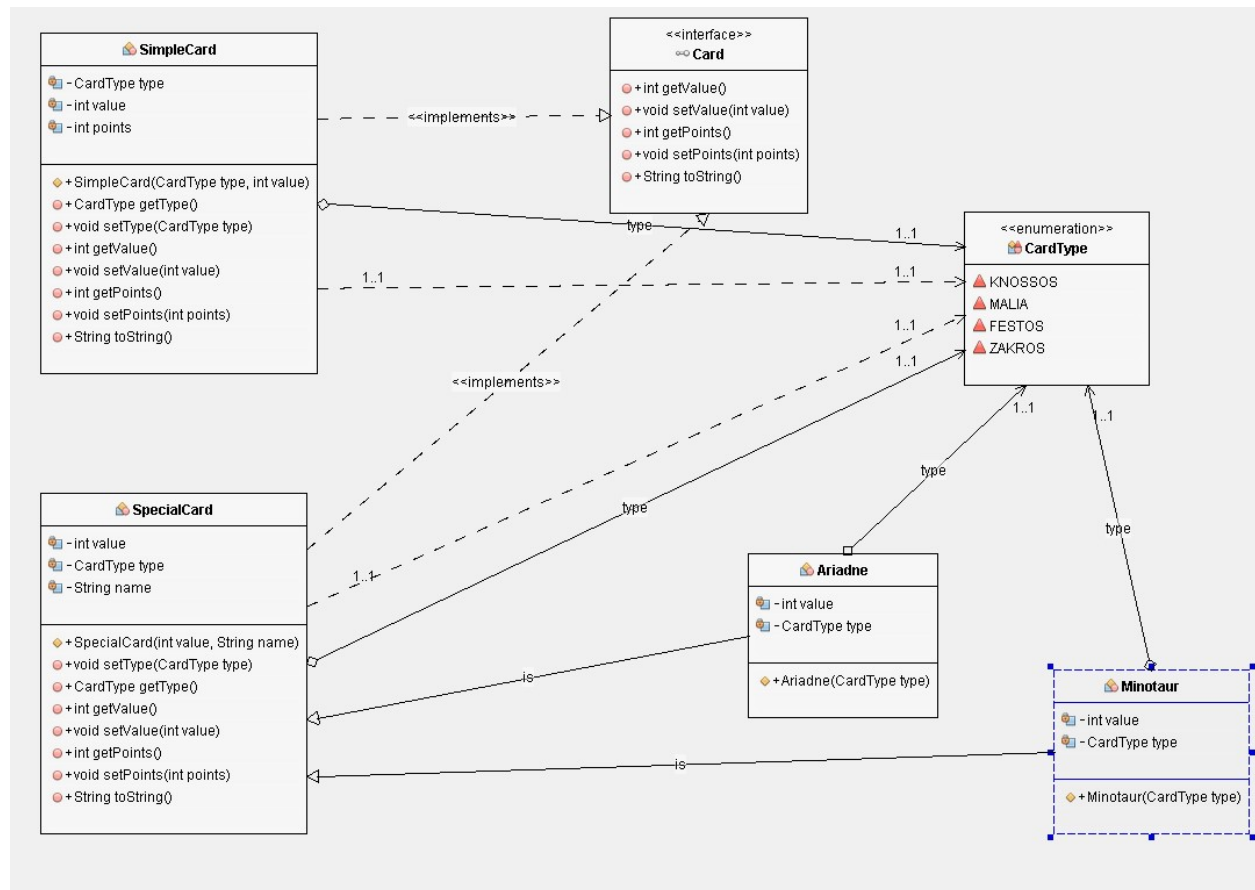
Οι υπόλοιπες μέθοδοι:

- public void setType(CardType type) ;
Sets the type of the Special Card.
- public CardType getType() ;
Gets the type of the Special Card.

Classes Aridane, Minotaur.

Αυτές οι κλάσεις κάνουν extend την Special Card και μέσω της εντολής super αποκτούν πρόσβαση στη κλάση Special Card και αρχικοποιούν τις τιμές value type και points.

- Τέλος, εδώ θα δείξουμε μια αναπαράσταση των κλάσεων που έχουν σχέση με τις κάρτες μέσω UML.



Class Monopati

Η κλάση αυτή δημιουργεί έναν πίνακα από Position όπου position είναι το καθε τετραγωνάκι που μπορεί να κινηθεί ένα πιόνι. Στην αρχικοποίηση του παιχνιδιού θα φτιάξουμε 4 μονοπάτια από 9 position το κάθε ένα.

To attribute:

- private CardType type; //The type of the path.

Class Position

Αποτελείται από δύο attributes τα οποία είναι :

- private CardType Anak_anoikey; // the type of the position
- private int score; //the score for each position square

Οι υπόλοιπες μέθοδοι:

- `public int getScore();` //returns the score of each position square
- `public void setScore(int scor);` //sets the score to ->int scor
- `public void setPositionType(CardType anak);` //sets the type of the position square
- `public CardType getPositionType();` //returns the position of each position square

Interface Finding

Το συγκεκριμένο interface μας προσφέρει τις παρακάτω μεθόδους ούτως ώστε να τις χρησιμοποιήσουμε παρακάτω στις κλάσεις **Fresco, RareFinding, SnakeGoddess** :

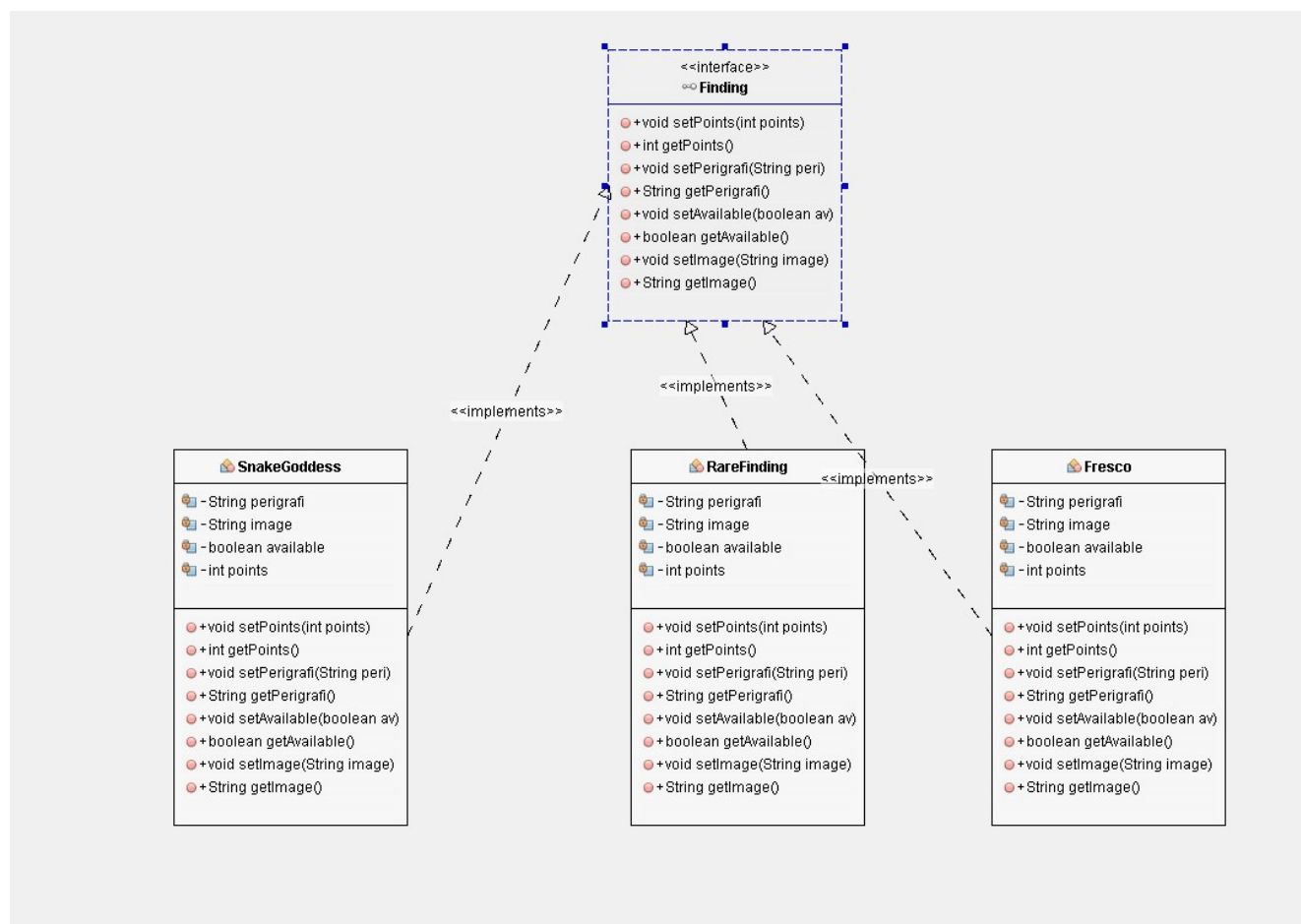
- `public void setPoints(int points);` // sets the points of the finding.
- `public int getPoints();` // returns the points of the finding
- `public void setPerigrafi(String peri);` // sets the description of the finding.
- `public String getPerigrafi();` // returns the description of the finding
- `public void setAvailable(boolean av);` // sets the availability of the finding.
- `public boolean getAvailable();` // returns the availability of the finding
- `public void setImage(String image);` // sets the image of the finding.
- `public String getImage();` // returns the image of the finding

Class RareFinding

Κατά την δημιουργία του προγράμματος με την κλάση rarefinding θα φτιάχνουμε τα σπάνια αρχαιολογικά ευρήματα θα τους περνάμε την εκάστοτε πληροφορία και την αξία τους σε πόντους.

Στη συνέχεια κατά το μοίρασμα θα ελέγχουμε κατάλληλα που θα μπουν ανάλογα με τους περιορισμούς τους.

Αντίστοιχα η κλάση Fresco για τις τοιχογραφίες και η κλάση SnakeGoddess για τα αγαλματάκια της θεάς των φιδιών.



Class Stoiva

Η κλάση στοίβα ουσιαστικά είναι η κλάση η οποία έχει όλο το πακάκι με τις κάρτες τις οποίες τραβάει κάθε παίκτης αφού έχει ξεφορτωθεί μία κάρτα από το χέρι του.

Το attribute της κλάσης αυτής είναι :

- **private ArrayList<Card> cards;** //creates a list of cards

το οποίο ίσως και να μην χρειάζεται να είναι σε ArrayList και στην υλοποίηση της Β φάσης να γίνει με πίνακα.

Οι μέθοδοι που περιέχονται στην κλάση αυτή είναι:

- `public void init_cards();` // Initializes and shuffles the cards.
- `public boolean isEmpty();` // Returns true if this list contains no elements.
- `public void get_card_from_stoiva(Card i);` // Removes a card from the list and a player now have it
- `public int size()` // Returns the size of a list.

Class CardType

Η συγκεκριμένη κλάση είναι ένα enumeration το οποίο περιέχει τα τέσσερα ανάκτορα που χρησιμοποιούμε στο παιχνίδι μας διότι θέλουμε ο χρήστης να περιορίζεται σ αυτά.

Class FindingPosition

Η κλάση FindingPosition έχει μία κατασκευάστρια μέθοδο η οποία αρχικοποιεί την μέθοδο της Position.

Το Attribute της είναι :

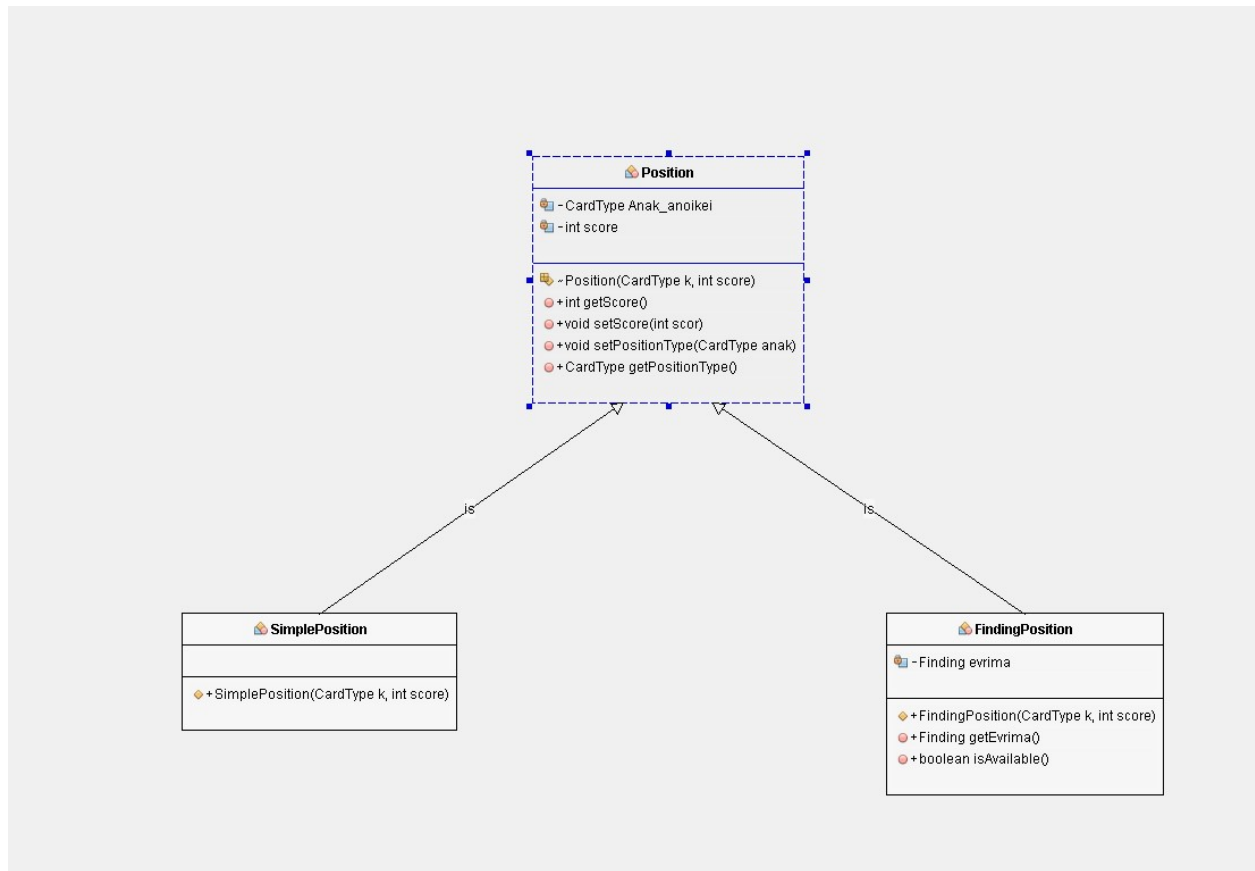
- `private Finding evrima;` // the finding

Και οι μέθοδοι που δημιουργούνται είναι :

- `public Finding getEvrima();` // returns the finding which is in this position
- `public boolean isAvailable();` // returns if the finding is available

Class SimplePosition

Η κλάση αυτή απλά κάνει extend την κλάση Position χωρίς προς το παρόν να έχω προσθέσει κάποια λειτουργία διαφορετική.



Class Player

Η κλάση αυτή αναπαριστά ένα παίκτη μέσα στο παιχνίδι.

Τα attributes:

- `private String name;` // the name of the player
- `private int score;` //the score of the player
- `private boolean hasPlayed,finished;` //hasPlayed is a variable who changes if a player has played and finished if the card of the player has finished.
- `private Card cards[];` //the cards the player have in his hand
- `private Card Card_to_play;` //the card which the player plays
- `private Position p1,p2,p3,p4;` //4 pawns for each player.

Μία κατασκευάστρια μέθοδο αρχικοποιώντας κάποιες από τις τιμές της.

public Player(String name)

και τέλος οι μέθοδοί της:

- public void setPioni(Position pos) // it sets the position of the pawn (pioni)
- public void setTurn(Player k) // it sets the turn of the players
- public void setCards(Card cards[]) // it sets the cards the player will have in his hand
- public String getName() // Returns the name of the player
- public void setName(String newName) // sets the name of the player to newName
- public void setCard_to_play(Card k) // set a card which a player wants to play to setCard_to_play
- public void Played() // sets the variable has_played to true
- public void has_finished() // Sets the variable has_finished to true
- public boolean Get_has_finished() // Returns if a player has finished the game(partida)

Package Controller

Class Controller

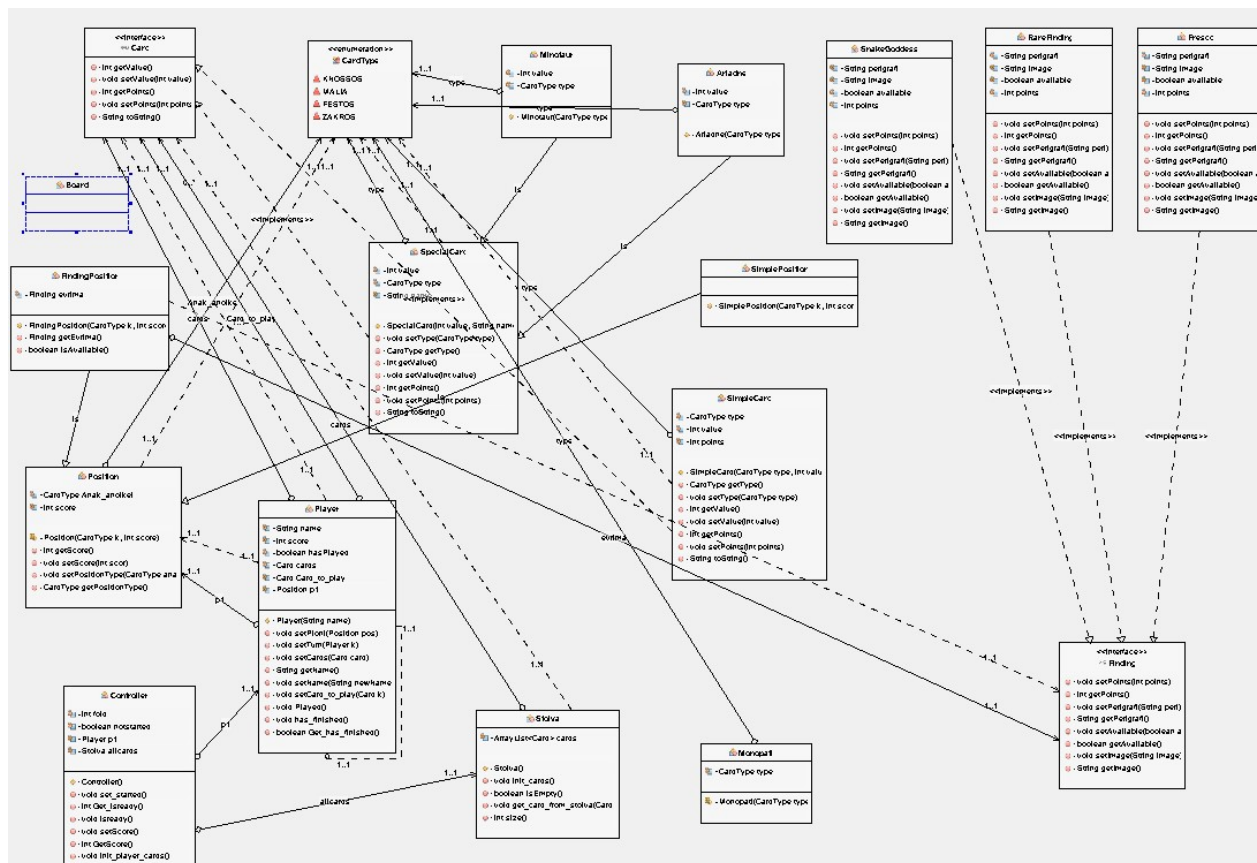
Αυτή η κλάση είναι ουσιαστικά το μυαλό του παιχνιδιού. Είναι υπεύθυνη για τη δημιουργία ενός νέου παιχνιδιού, μιας νέας παρτίδας, τη δημιουργία στιγμιοτύπων παικτών σειράς και φυσικά τη σύνδεση μεταξύ των γραφικών και του Model. Αυτό που κάνει η κλάση αυτή είναι να παίρνει τις επιλογές του χρήστη μέσω των γραφικών και να πραγματοποιεί οποιαδήποτε ενέργεια χρειάζεται έτσι ώστε το παιχνίδι να παίζεται σωστά. Φυσικά είναι υπεύθυνη αυτή η κλάση για να υπολογίζει το σκορ και να ενημερώνει τότε τελειώνει το παιχνίδι.

Τα attributes της κλάσης:

- private int fold, points, score;
- private boolean notstarted, empty_table, new_round, CardsHasPlayed;
- private Player p1, p2;
- private Stoiva allcards = new Stoiva();

Οι μέθοδοι:

- public void set_started() // sets the variable not_started to false
- public void isready() // increases the variable isready by 1(isready++)
- public void setScore() // sets the score of a game after the game has finished
- public int GetScore() // Returns the score of the game
- public void init_player_cards() // initializes players cards in the beginning
- public void init_table() // initializes some things (allcards,board,stoiva) for a new game
- game_has_finished() // Return true if a game has finished, false otherwise



Package View

Αυτό το πακέτο θα αποτελείται από μία κλάση που θα δημιουργεί ένα ένα frame και μέσα σε αυτό ένα panel. Μέσα σε αυτό το panel θα υπάρχουν 2 panels για κάθε παίχτη ,όπου στο πρώτο θα περιλαμβάνονται οι κάρτες του και στο δεύτερο η κάρτα που θέλει να παίξει. Επίσης θα υπάρχει ένα κεντρικό panel που θα είναι το ταμπλό του παιχνιδιού και κουμπιά τα οποία θα αντιστοιχούν στα μονοπάτια . Όταν πατιέται μια κάρτα αν είναι η σειρά του παίκτη θα μεταφέρεται στο χώρο όπου βρίσκεται η κάρτα που θέλει να παίξει και αν πατήσει το κουμπί παίξε και η κάρτα είναι έγκυρη θα μεταφέρεται η κάρτα στο κεντρικό panel (ταμπλό).Ακόμα θα υπάρχουν κουμπιά για το Quit και New Game και 2 χώροι μηνυμάτων , ένας για κάθε παίκτη που θα έχει ένα κουμπί για τις τοιχογραφίες που έχει βγάλει φωτογραφία , το σκόρ του και τα αγαλματάκια που έχει μαζέψει.

Ακόμα θα υπάρχει για κάθε παίκτη ένα label που εκεί θα πηγαίνουν τα σπάνια ευρήματα που βρίσκει για κάθε ανάκτορο.

Στο κεντρικό πάνελ του παιχνιδιού θα υπάρχει και η στοίβα με τις κάρτες που μπορεί να τραβήξει ο κάθε παίκτης.