



Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζιτζικας

Χειμερινό Εξάμηνο 2020-2021

[STRATEGGO]

*Εισαγωγή*

[ΣΠΥΡΙΔΩΝ ΔΟΥΚΕΡΗΣ]

[CSD4871]

[4/12/2022]

## Περιεχόμενα

1.Εισαγωγή.....	1
2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model .....	2
3.Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller .....	6
4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View .....	9
5.Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML.....	12

## 1. Εισαγωγή

Η υλοποίηση της εργασίας βασίζεται στο MVC (Model View Controller).

Αρχικά θα αναλύσουμε το πακέτο model και τις κλάσεις που περιέχει έπειτα το πακέτο Controller και τις κλάσεις του και τέλος το view και τις κλάσεις του. Επίσης θα αναφερθούμε στο πως συνδέονται και αν συνδέονται μεταξύ τους τα τρία παραπάνω. Διαγράμματα UML θα υπάρξουν για να μας βοηθήσουν την κατανόηση και την σύνδεση των κλάσεων και του κώδικα . Τέλος θα βγάλουμε συμπεράσματα για την υλοποίηση και την χρησιμότητα της αναφοράς και του κώδικα.

## 2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

1)player File : Αποτελείτε από τα packages Piece και Player i)Player class:Σκοπός αυτής της κλάσης είναι η δημιουργία ενός παίκτη και αρχικοποίηση τ και των δεδομένων του.Αποτελείτε Από:1 constructor και από 10 διαφορετικές μεθόδους που συμβάλουν στην αρχικοποίηση ενός παίκτη

**Constructor** : `public Player(boolean Player)`

Έχουμε 1 private Boolean μεταβλητή την player στην οποία στο constructor την αρχικοποιούμε ανάλογα με το τι θέλουμε να είναι ο παίκτης αν θέλουμε ο παίκτης να είναι της κόκκινης ομάδας τότε τον αρχικοποιούμε με λάθος αλλιώς την αρχικοποιούμε με αλήθεια . Έπειτα στον constructor αρχικοποιούμε τις λίστες με τα πιόνια του παίκτη τα ζωντανά (Πιόνια π έχει πάνω στο ταμπλό ο παίκτης και μπορεί να παίζει με αυτά) και τα πεθαμένα που χρειάζονται να ξέρουμε ποια είναι άμα χρειαστεί να τα αναστήσουμε να επιλέγουμε ένα από αυτά.

Έπειτα έχουμε 4 μεθόδους getters .Ποιο Συγκεκριμένα :

`public Boolean getPlayerteam()`

Επιστρέφει Boolean τύπο, μεταβλητή που θα επιστραφεί εξαρτάτε τι θα είναι αν ο παίκτης είναι της μπλε ομάδας τότε θα επιστραφεί ψέμα αλλιώς θα επιστραφεί αλήθεια αν ο παίκτης έχει τα πιόνια της κόκκινης ομάδας.

`public int getAlivearrSize()`

Επιστρέφει τύπο int.αυτή η μέθοδος επιστρέφει το πλήθος των ζωντανών πιονιών.

`public int getDeadarrSize()`

Επιστρέφει τύπο int,αυτή η μέθοδος επιστρέφει το πλήθος των ψόφρων πιονιών.

`public int getRescueCounter()`

Αυτή η μέθοδος επιστρέφει πόσες φορές ο παίκτης έχει κάνει διάσωση στο παιχνίδι επιτρέπονται μέχρι 2 διάσωση από κάθε παίκτη. (τύπος intεγγερ.

Έχουμε μια setter κλάση . Οπού:

`public void setPlayerteam(boolean Player)`

Σε αυτή την μέθοδο αρχικοποιούμε την ομάδα του παίκτη αλλάζοντας το πριβε πεδίο.

2 μεθόδους για να προσθέτουμε στην λιστα με τα πιόνια τα πιόνια στην μια προσθέτουμε τα ζωντανά πιόνια και στην άλλη τα σκοτωμένα.

```
public void addDead(Piece piece)

public void addAlive(Piece piece)
```

2 μεθόδους που θα βοηθήσουν στον να αναστήσουμε αργότερα κάποιο πιόνι:

```
public void removeFromDead(Piece piece)
σε αυτήν την μέθοδο π καλείτε όταν ανασταίνετε κάποιο πιόνι
αφαιρούμε από την λιστα τον σκοτωμένων πιονιών το πιόνι που θέλει ο
χρήστης να αναστήσει.

public Piece getDeadPiece(int i)
επιστρέφει την θέση ενός πιονιού από την λιστα των σκοτωμένων
πιονιών.
```

Τελος μια μεθοδος για αυξηση μιας μεταβλητης η:

```
public void IncreaseRescueCounter()
αυξάνει το rescue counter κατα 1
```

ii) Alivepieces class/iii) Deadpieces class: Αυτές οι δυο κλασεις αποτελουντε από τις ειδικες μεθόδους. Η κλασεις μας βοηθαν στην δημιουργια και στο περασμα των λιστων με τα πιονια τ παιχνιδιου ειτε ζωντανα ειτε σοτομενα. Οι Κλασεις αποτελουντε από έναν Constructor και 5 μεθόδους:

```
Constructors: public Alivepeaces()/public Deadpieces()
```

Τα κονστρακτορ δημιουργούν το καθένα την λιστα του για τα πιόνια π θα περιεχει. Η λιστα είναι τυπου piece

Έχουμε 2 getters για την επιστροφή τιμών τις:

```
public int getRank(int I)
```

η μέθοδος επιστρέφει το ρανκ του πιονιού . Παίρνει ως όρισμα έναν ιντιγгер ο οποίος αντιπροσωπεύει την θέση του πιονιού στην συγκεκριμένη θέση τις λίστας π θέλουμε να ξέρουμε το ρανκ

```
public Piece getPice(int I)
```

η μέθοδος επιστρέφει το πιονιού . Παίρνει ως όρισμα έναν ιντιγгер ο οποίος αντιπροσωπεύει την θέση του πιονιού στην συγκεκριμένη θέση τις λίστας που θέλουμε να πάρουμε το πιόνι.

Τέλος έχουμε 2 μεθόδους για την πρόσθεση και την αφαίρεση πιονιών από την λιστα:

```
public void removePiece(String name)
```

αφαιρεί πιόνια από την λιστα εχει ως όρισμα ένα πιόνι το οποίο θέλουμε να αφαιρέσουμε από την λιστα

```
public void addPiece(Piece piece)
```

προσθέτει πιόνια στην λιστα εχει ως όρισμα ένα πιόνι το οποίο θέλουμε να προσθέσουμε από την λιστα

2)Piece

### i)Interface Piece

Το Interface περιέχει 3 γκετερς και 3 σεττερς μεθόδους:

```
public String getName();
```

η μέθοδος μας επιστρέφει το όνομα του συγκεκριμένου πιονιού

```
public int getRank();
```

η μέθοδος μας επιστρέφει το αξίωμα του συγκεκριμένου πιονιού

```
public int getRescueCount();
```

η μέθοδος μας επιστρέφει τις φορές που συγκεκριμένου πιονιού εχει διασωθεί.

```
public void setRank(int rank) ;
```

εχει ως όρισμα το ρανκ στο οποίο σεταρει την τιμή του

```
public void setName(String name);
```

εχει ως όρισμα το όνομα το οποίο σεταρει την τιμή του

```
public void setRescueCount();
```

εχει ως όρισμα το ρανκ στο οποίο σεταρει την τιμή τ

ii)Class SpecialMovablePiece : Οι κλαση αυτή κάνει extend την class movablepiece.Μας έχουν δοθεί τρία πιόνια τα οποία έχουν ειδικές δυνάμεις .ΟΙ κλάσης αυτών των πιονιών κάνουν extend αυτήν την κλαση . Περιέχετε από 3 κλάσης.

Έναν κονστρακτορ:

```
public SpecialMovablePiece(int rank, String name)
```

Οι οποία δέχεται ως ορίσματα το αξίωμα και το όνομα και με super καλεί και θα τα αρχικοποιήσῃ μέσω της movablepiece κλάσης.

Μια μέθοδο σεττερ:

```
public void setSpecialPower(String specialpower)
```

Περνιέται ως όρισμα το String όνομα δυνάμεις και αυτή η μέθοδος θα το αρχικοποιήσει

Μια μέθοδο σεττερ:

```
public String getSpecialPower()
```

Επιστρέφει την δύναμη που έχει το δεδομένο πιόνι

iii) Class MovablePiece κάνει implements την κλαση Piece

Αρά η κλαση κάνει implements την κλαση Piece κάνει override όλες της μεθόδους της. Επίσης περιέχει έναν constructor:

```
public MovablePiece(int rank, String name)
```

Στον Constructor αυτόν δεχόμαστε ως ορίσματα το ρανκ και το όνομα του πιονιού και τα σιτάρει καλώντας τις μεθόδους που χιάζονται

iv) Class ImmovablePiece κανει implements την κλαση Piece

Αρά η κλαση κάνει implements την κλαση Piece κάνει override όλες της μεθόδους της. Επίσης περιέχει έναν constructor:

```
public ImmovablePiece(int rank, String name)
```

Στον Constructor αυτόν δεχόμαστε ως ορίσματα το ρανκ και το όνομα του πιονιού και τα σιτάρει καλώντας τις μεθόδους που χιάζονται

v) All possible pieces

Αυτές οι κλάσεις περιέχουν όλα τα δυνατά πιόνια που μπορεί να έχει ένας παίκτης σε αυτό το παιχνίδι . Από τους κανόνες ξέρουμε ότι κάθε πιόνι έχει διαφορετικές ιδιότητες και διαφορετικές υπερ δυνάμεις και τρόπος κινήσεως . Κάθε κλαση από τα πιόνια κάνει extend είτε την ImmovablePiece είτε την MovablePiece είτε την SpecialMovablePiece

Στις οποίες κάνουν super το όνομα και το αξίωμα του κάθε πιονιού . Κάθε πιόνι έχει διαφορετικό αξιώματά πιόνια που έχουν και ειδική δύναμη αρχικοποιείτε με κλήση μεθόδου.

### 3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

a) Controller

Στο Controller ελέγχουμε τις βασικές ενέργειες του παιχνιδιού και αποτελείτε από μοναδική κλαση στην οποία έχουμε έναν κονστροκτορ και 11 μεθόδους για την

υλοποίηση αρχικοποίηση και σωστό ενοχή ροής του παιχνιδιού και των κανόνων που αυτό αποτελείτε.

Constructor: `public Controller( )`

Στον Constructor θα δημιουργήσουμε τους δυο παίκτες θα αρχικοποιήσουμε Id του παίκτη οπότε θα παίξει στην ανερχόμενη σειρά Προτεραιότητας για να παίξει . Τέλος θα δημιουργήσει ένα νέο αντικείμενο (View Subclass)Movepiece το οποίο περιέχει το board (ο demo κώδικας που μας δόθηκε) οπότε μετρά από εκεί θα αρχίσει ο αγώνας.

Μέθοδος για αρχικοποίηση πονιού:

`public void initializeButton(Button button, String Name, String Color, int x, int y)`  
 Σε αυτήν την μέθοδο αρχικοποιήστε το κουμπί στο board.Εδώ με άλλες βοηθητικές μεθόδους από την κλάση ΠΑΙΚΤΗΣ δίνουμε στο πιόνι τις ιδιότητες του και την τοποθεσία του.  
 Επίσης περνάμε κάτι ορίσματα στη μέθοδο τα οποία επίσης με κλήση μεθόδων τα αρχικοποιούμε (όνομα το πιόνι το χρώμα του παίκτη και τις συντεταγμένες του).

Μέθοδος ελέγχου έγκυρης κίνησης:

`public boolean MoveValidity (Button start , Button end)`  
 Σε Αυτήν την μέθοδο δεχόμαστε ως ορίσματα δυο κουμπιά οποιαδήποτε στο ταμπλο και ελεγχουμε αν το αρχικο κουμπί μπορεί η το πως θα ανταπεξέλθει αν μπορεί να ανταπεξέλθει με το άλλο button πως θα ανταπεξέλθει . Υπάρχουν πολλές περιπτώσεις που θα χρειαστεί να προσέξουμε εδώ πχ αν το 1ο button θέλει να πάει στο 2° αλλά το 2° είναι απαγορευμένη περιοχή.

Μέθοδοι Ροής παιχνιδιού:

`public int EndGame()`

Σε αυτήν την μέθοδο μετρά από κάθε σειρά θα ελεγχουμε αν υπό διάφορες συνθήκες πχ αν δεν μπορεί ένας παίκτης να κουνήσει κανένα πιόνι του η άμα επιτεθεί η σημαία . Τότε να επιστρέφει ένα και το παιχνίδι να λήγει.

`public int Attack (Button start , Button end)`

Μέθοδος Επίθεσης . Σε αυτήν την μέθοδο θα κάνουμε τους ελέγχου κόκκινο πιόνι ενάντια μπλε πιόνι . Κάθε πιόνι έχει το δικό του αξίωμα αριθμημένα από το πιο δυνατό έως το πιο αδύναμο . Επίσης ένα πιόνι αν έχει την δικιά του ειδική δύναμη θα μπορεί να ανταπεξέλθει με συγκεκριμένα πιόνια καλύτερα.

```
public void Rescue(Button button)
```

Μέθοδος ανάστασης πιονιού . Σε αυτή την μέθοδο θα ελεούμε και αν τηρούνται οι κανόνες του παιχνιδιού τότε θα ανασταίνουμε το μπα τον που περνιέται σαν όρισμα στην μέθοδο και θα πραγματοποιούνται οι απαραίτητες αλλαγές στις λίστες πιονιών του παίκτη.

```
public void hideButtons()
```

Μέθοδος κρύψε τα κουμπιά αυτήν την μέθοδο με την αλλαγή της σειράς προτεραιότητας των παικτών είτε θα κρύβονται είτε θα εμφανίζονται τα πιόνια μιας συγκεκριμένης ομάδας ανάλογα με την σειρά του παίκτη.

```
public int checkIfBlocked(Button button)
```

Σε αυτήν την μέθοδο όπως λέει και το όνομα ενέχουμε αν ένα πιόνι μπορεί να κουνηθεί . Αν όλα τα πιόνια ενός παίκτη δεν μπορούν να κουνηθούν τότε ο παίκτης χάνει το παιχνίδι . Επίσης αν ξέρουμε ότι ένα πιόνι δεν μπορεί να κουνηθεί τότε δεν θα αφήνουμε το παίκτη να το επιλέξει για να κάνει κινήσει εφόσον είναι μπλοκαρισμένο.

```
public void afterButtonMovement(Button button)
```

Σε αυτήν την μέθοδο εφόσον έχουμε ελέγξει την εγκυρότητα της κίνησης και πάει να γίνει η κίνηση θα ανταλλάζουμε τα κουμπιά η θα τα σκοτώνουμε ανάλογα την περίπτωση.

Μια μέθοδο για να ελέγχουμε αν τέλειωσε η σειρά:

```
public int endturn()
```

κάνει καταλλήλους ελέγχους και επιστρέφει αν τέλειωσε η όχι την κίνηση τ ο παίκτης. (η κίνηση να είναι αποδεκτή και να έχει κουνηθεί το πιόνι) .

Τέλος θα αναπτύξουμε τις κλάσεις στις οποίες θα ελεούμε την σειρά του παίκτη και θα αλλάζουμε την σειρά προτεραιότητας.

```
public void nextTurn()
```

Έχοντας τσεκάρει ότι ο προηγούμενος παίκτης έκανε την κίνηση του με αυτήν την μέθοδο αλλάζουμε το ID του παίκτη έτσι αλλάζουμε και την σειρά.



Έχουμε ένα setter + ένα getter στα οποία βάζουμε και έχουμε πρόσβαση στην μεταβλητή με το player id.

```
public int getPlayerID()
Accessor Returns the ID Post-condition ID return @return current
player id(current player turn)

public void setPlayerID(int id)
Transformer Sets the ID ,Precondition id s 1 or 0, Post-condition
set id @param id ID of player(current player turn)
```

#### 4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View

a)MovePiece:

Η Κλάση αυτή μας δίνεται ως βοηθητικός κώδικας κλάση περιέχετε από έναν Constructor για την δημιουργία του ταμπού και μερικές βοηθητικές συναρτήσεις για να αλλάξουμε τις εικόνες στα κουμπιά που δημιουργούμε (Ως πιόνια).Επίσης σε αυτήν την κλάση μας δίνεται και η συνάρτηση CardListener που κάνει implement την MousetListener για την κίνηση και την ανίχνευση κουμπιών.

Constructor:

```
public MovePiece() throws IOException
```

Βοηθητικοί μέθοδοι για να σετταρουμε τις εικόνες του ταμπού:

```
private ImageIcon getImageBack()
```

```
private ImageIcon getImageBack2()
```

```
private ImageIcon getImageCard()
```

## b)Button

Η κλάση περιέχει 14 μεθόδους και 1 constructor.Σε αυτήν την κλάση δημιουργείτε ένα κουμπί το οποίο έχει διάφορες ιδιότητες , συντεταγμένες , παίκτη και διαφορετικό Image κάθε γύρω αν είναι πόνι παίκτη.

Έχουμε 7 μέθοδος που είναι setters και 7 που είναι getters:

1 σεττερ και 1 γκεττερ για να ξέρουμε σε ποιον ανήκει το κουμπί:

```
public void setPlayer(int player):
```

Αρχικοποιείτε πάντα με κάθε δημιουργία κουμπιά σε 0 οπού σημαίνει ότι δεν ανήκει σε κάποιον παίκτη . Έπειτα μπορεί να αλλάξει αν είναι πόνι οπού η μέθοδος θα κληθεί και θα αλλάξει τη τιμές σε 1 αν ο παίκτης είναι 1 στην κόκκινη ομάδα και σε 2 αν ο παίκτης είναι στην μπλε ομάδα

```
public int getPlayer()
```

Μας δίνει προσβάσεις και μας επιστρέφει έναν ιντεγγερ ο οποίος δηλώνει σε ποιον παίκτη ανήκει ένα κουμπί.

2 Μέθοδοι σεττερς 2 Μέθοδοι γκεττερς για τις συντεταγμένες του κουμπιού:

```
public void setXcoordinates (int x)
```

```
public void setYcoordinates (int y)
```

Στις μεθόδους αυτές αλλάζουν οι συντεταγμένες ανάλογα με την κίνηση και την στάνταρ αρχικοποίηση πχ πιονιών θα είναι σε συγκεκριμένες συντεταγμένες η κενά μπλοκ π δεν μπορούν να πατηθούν . Οι Συναρτήσεις δέχονται ως ορίσματα έναν ιντεγгер ο οποίος είναι οι καινούργιες συντεταγμένες του κουμπιού.

```
public int getXcoordinates()
```

```
public int getYcoordinates()
```

Μας δίνει προσβάσεις και μας επιστρέφει έναν ιντεγгер και στις 2 μεθόδους, ο οποίος δηλώνει σε ποιες συντεταγμένες είναι το κουμπί αυτήν την στιγμή.

2 Μέθοδοι σέτερ και 2 μέθοδοι γκέτες για τις ιδιότητες του κουμπιού αν μπορεί να επιτεθεί η αν μπορεί να πατηθεί:

```
public void setCanBePressed(boolean pressed)
```

```
public void setCanBeAttacked(boolean attacked)
```

Στις μεθόδους αυτές αναθέτουμε τιμές στις διάφορες ιδιότητες που μπορεί να έχει ένα κουμπί . Αν μπορεί να επιτεθεί η αν μπορεί ένα άλλο πιόνι να μετακινηθεί προς αυτό. Στις συναρτήσεις περνάμε μπουλιαν μεταβλητές για να ορίσουμε τα παραπάνω.

```
public Boolean getCanBePressed()
```

```
public Boolean getCanBeAttacked()
```

Μας δίνει προσβάσεις και μας επιστρέφει έναν μπόλια και στις 2 μεθόδους, ο οποίος δηλώνει αν έχουν αλλά πιόνια δικαιώματα πάνω σε αυτό το πιόνι.

2 Μέθοδοι 1 γκετερς 1 σέτερ για να βάζω τα κρυμμένα μπλοκ

```
public void setHiddenImage(Image img) :
```

Ανάλογα τος παίκτη κόκκινος η μπλε θα σοιτάρουμε το κρυμμένο μπλοκ ώστε να μπορούμε να το χρησιμοποιούμε σε άλλες μεθόδους για εναλλαγή σειράς και κρύψιμο πιονιών.(όρισμα μια εικόνα)

```
public Image getHiddenImage()
```

Μας δίνει πρόσβαση στην εικόνα που χιάζετε για να κρύβουμε τα πιόνια αναλόγως την ομάδα.

Τέλος 2 μέθοδοι για την ανάθεση και την πρόσβαση στο πιόνι.

```
public void setbuttonsPiece(Piece piecebut):
```

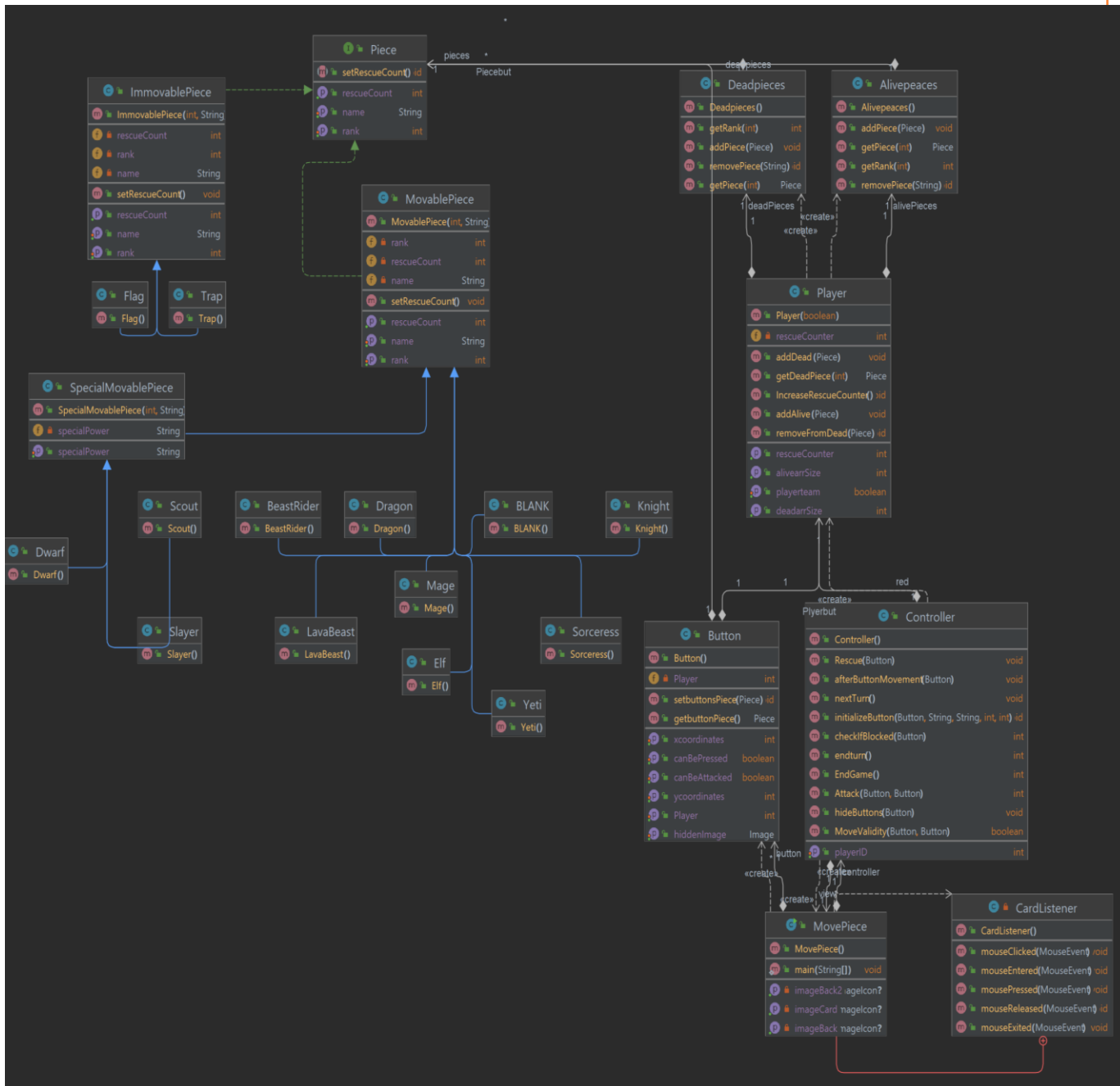
αναθέτουμε σε αυτήν την μέθοδο το πιόνι του συγκεκριμένου κουμπιού αν είναι κενό το κουμπί θα αναθέτω NULL.

```
public Piece getbuttonPiece()
```

Αυτή η μέθοδος μας δίνει πρόσβαση στο πιόνι του Συγκεκριμένου κουμπιού έτσι μπορεί να επιστρέψει είτε πιόνι είτε NULL αν το κουμπί δεν εμπεριέχει τίποτα.

(UML ΣΤΗΝ ΕΠΟΜΕΝΗ ΣΕΛΙΔΑ ΛΟΓΟ ΧΩΡΙΤΙΚΟΤΗΤΑΣ)

## 5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML



## 6. Λειτουργικότητα (B Φάση)

Σε αυτήν την ενότητα θα γράψετε στη B φάση ποια ερωτήματα καταφέρατε να υλοποιήσετε είτε επιτυχώς είτε εν μέρει (και ενδεχομένως ποια όχι).

Η Main Μου βρίσκετε στο  
View.BoardMovement.MovePiece

✓ σημαίνει ολοκληρώθηκε εντελώς

Ενώ ✗ σημαίνει δεν υλοποίησα ή έγινε μια προετοιμασία

Τήρηση σειράς και τοποθέτηση πιονιών (10%) ✓

• Σωστή Κίνηση πιονιών και ανιχνευτή (12%) ✓

• Επίθεση πιονιών (10%) ✓

• Επίθεση εξολοθρευτή-δράκου (7%) ✓

• Εξουδετέρωση παγίδας από νάνο (5%) ✓

• Διάσωση πιονιού που βγήκε από το παιχνίδι (10%) ✗

(είχα φτιάξει μερικές για την υλοποίηση του επίσης είχα τις λίστες με τα σκοτωμένα πιόνια και τα ζωντανά και counter για το πόσες φορές έχει το πιόνι διασωθεί αλλά δεν είχα χρόνο για την υλοποίηση του)

• Τέλος παιχνιδιού-κατάκτηση σημαίας (6%) ✓

• Μενού πληροφοριών στο πλάι (12%) ✗

- Game mode: Μειωμένος Στρατός (8%) ✓

- Game mode: Καμία Υποχώρηση (Bonus 5%) ✗

Αναφορά (Ανανεωμένη από την Φάση A) 8 ✓

Σχόλια Javadoc (Pre – Post Conditions) 4 ✓

JUnit Tests 4 ✗

UML Diagrams με βάση την τελική έκδοση και επεξήγηση τους 4 ✓

## 7. Αλλαγές

Controller:

Check if blocked method:

Δεν υλοποιήθηκε

Rescue:

δεν υλοποιήθηκε έχοντας μαζέψει δεδομένα ζωντανά πόνια πεθαμένα και φορές που έχουν διασωθεί.

Η Αλλαγμένη Attack:

Δηλαδή από το να επιστρέφω πιο παίκτης επιτέθηκε επέστρεφα κάτι σαν validity του attack αν θα πραγματοποιηθεί και αν είναι αρκετά τα δεδομένα που μας δίνονται για να πραγματοποιηθεί .Σε αντίθεση στην πρώτη φάση που θα επέστρεφα αν επιτέθηκε ποιος σε ποιον και αν χτυπήθηκε η flag.

After button movement method = MakeBlank(Ουσιαστικά κάνω κενό το κουμπί δηλαδή να μην ανήκει σε κανέναν και τις εικόνες του και τις ιδιότητες του)

View:

```
/**
 * Transformer Sets start img
 * Precondition passes type image
 * @param img Image will be used for start img(PROSUESA)
 */

public void setStartImg(Image img)
{
```

```

        this.sImg=img;
    }

    /**
     * Accessor Returns start img
     * Postcondition img returned
     * @return start image (PROSUESA)
     */

    public Image getStartImg()
    {
        return this.sImg;
    }

    /**
     * Transformer sets image of current turn
     * Precondition Argument is an image
     * Postcondition Image of button was set
     * @param img img used in next turn
     */

    public void TurnImg(Image img)
    {
        this.change=img;
        this.setIcon(new ImageIcon(change) );
    }

```

## 8. Συμπεράσματα

Σε αυτήν την ενότητα θα γράψετε τα συμπεράσματά σας για την εργασία, τυχόν προβλήματα που συναντήσατε και γενικά ότι άλλο κρίνετε απαραίτητο να αναφερθεί.

Δεν άλλαξα καθόλου το model ενώ υπήρξαν αρκετές αλλαγές στο View και στο controller

Αφαίρεσα μερικές μεθόδους και πρόσθεσα μερικές άλλες στο controller.

Επίσης την Attack method άλλαξα αρκετά την σκέψη υλοποίησης της και δεν ακολούθησα το σκεπτικό της Α φάσης.

Αφαίρεσα μερικές μεθόδους από την πρώτη φάση και πρόσθεσε μερικές αναλυτικά στις ΑΛΛΑΓΕΣ.



