

Τ.Ε.Ι. ΠΕΛΟΠΟΝΝΗΣΟΥ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

Πτυχιακή εργασία

ΥΛΟΠΟΙΗΣΗ ΔΙΑΔΙΚΤΥΑΚΟΥ  
MULTIPLAYER (ΠΟΛΛΩΝ ΧΡΗΣΤΩΝ)  
ΠΑΙΧΝΙΔΙΟΥ ΣΤΑΤΗΓΙΚΗΣ  
ΣΕ ΠΕΡΙΒΑΛΛΟΝ  
WEB BROWSER (ΦΥΛΛΟΜΕΤΡΗΤΗ)

Γερούλάκου Αικατερίνη  
Επιβλέπων Καθηγητής Πανάγος Νικόλαος

- 2014 -

TEI PELOPONNISOU  
FACULTY OF ENGINEERING  
DEPARTMENT OF COMPUTER ENGINEERING TE

IMPLEMENTING WEB  
MULTIPLAYER  
GAME STRATEGY  
IN ENVIRONMENT  
WEB BROWSER

**Geroulakou Aikaterini**  
Supervisor Panagos Nikolaos

## ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή Νικόλαο Πανάγο αλλά και όσους με βοήθησαν για τη υλοποίηση και λειτουργία του παιχνιδιού.

Επίσης θέλω να ευχαριστήσω όσους και όσες με βοήθησαν στη συγγραφή της αναφοράς αλλά και στις διορθώσεις που χρειάστηκε να γίνουν πάνω σε αυτήν.

Τέλος ένα ευχαριστώ στους χρήστες που έπαιζαν το παιχνίδι και με βοήθησαν να βρω επιπλέον αλλαγές που χρειάστηκε να κάνω αλλά και για διάφορες ιδέες που μου έδωσαν για την επέκταση του παιχνιδιού.

## ΣΥΝΟΨΗ

Η παρούσα διπλωματική εργασία ασχολείται με τη δημιουργία ενός διαδικτυακού πολλών χρηστών παιχνιδιού στρατηγικής, βασισμένο σε αληθινό χρόνο. Ο αγγλικός όρος είναι γνωστός και ως MMORTS (massively multiplayer online real-time strategy).

Τα πρώτα διαδικτυακά παιχνίδια (MMOGs massively multiplayer online games) εμφανίστηκαν στα τέλη του 80' με αρχές του 90'. Αυτού του είδους τα παιχνίδια μπορούν να δεχτούν πολλούς χρήστες και από όλο τον κόσμο. Αρχικά ξεκίνησαν από τους υπολογιστές και τις κονσόλες και με τα χρόνια πέρασαν και στα κινητά.

Υπάρχουν πολλοί τύποι διαδικτυακών παιχνιδιών. Τα πιο γνωστά είναι τα RPGs (role playing games) και ακολουθούν τα first-person shooter, real-time strategy, turn-based strategy, simulations, sports, racing, casual, music/rhythm και social.

Βασικό χαρακτηριστικό αυτών των παιχνιδιών είναι η ύπαρξη ενός χάρτη (map) ή και παραπάνω όπου εμπεριέχει όλους τους χρήστες.

Ενός server δηλαδή που δέχεται όλους τους χρήστες εκτός και αν είναι τόσοι πολλοί όπου ένας server δεν είναι αρκετός να φιλοξενήσει όλους τους χρήστες (Ένας server μπορεί να φιλοξενήσει από 10.000-12.000 χρήστες).

Σε μεγάλο ποσοστό αυτών το παιχνιδιών υπάρχει και το εικονικό χρήμα (virtual currency), όπου οι χρήστες μπορούν να το αποκτήσουν και να αγοράσουν με αυτό προϊόντα. Επίσης υπάρχει και η ανταλλαγή χρημάτων με τα εικονικά χρήματα.

Η διαφορά των παιχνιδιών με πολλαπλούς χρήστες είναι πως λειτουργούν συνέχεια ασχέτως αν ο δικός μας χρήστης είναι μέσα στο παιχνίδι και σε πολλά από αυτά το παιχνίδι δεν τερματίζεται ποτέ.

Επίσης από την στιγμή που το παιχνίδι περιέχει πολλούς χρήστες και όχι μόνο ένα, μπορούν να δημιουργηθούν και ομάδες μεταξύ παικτών και αυτό είναι που τα κάνει και τόσο ενδιαφέρον.

## ABSTRACT

This thesis deals with the creation of an online multiuser strategy game, based on real time. The English term is aka MMORTS (massively multiplayer online real-time strategy).

The first online games (MMOGs massively multiplayer online games) appeared in the late 80's to early 90's. Such games can accommodate multiple users from around the world. Originally started by computers and consoles and in years passed in mobiles too.

There are many types of online games. The most famous are the RPGs (role playing games), followed by first-person shooter, real-time strategy, turn-based strategy, simulations, sports, racing, casual, music / rhythm and social.

A key characteristic of these games is the existence of a map or above which contains all users.

A server that receives all users unless they are so many where one server is not enough to accommodate all users (A server can accommodate 10,000-12,000 users).

A large proportion of these game there and the virtual money (virtual currency), where users can obtain and buy with this some products. There is also the money exchange with virtual money.

The difference in the game with multiple users is how they work then regardless of whether our user is in on the game, and in many of them the game does not ever quit.

Also by the time the game contains many users, not just one, can create groups and between players and that's what makes them so interesting.

## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

<b>1 ΕΙΣΑΓΩΓΗ .....</b>	<b>7</b>
1.1 Περίληψη.....	7
1.2 Κίνητρο για την διεξαγωγή της εργασίας.....	7
1.3 Σκοπός και στόχοι εργασίας.....	7
1.4 Δομή εργασία.....	8
<b>2 ΣΧΕΔΙΟ ΔΡΑΣΗΣ ΓΙΑ ΕΚΠΟΝΗΣΗ ΤΗΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ .....</b>	<b>9</b>
2.1 Μέθοδος ανάλυσης και ανάπτυξης πτυχιακής.....	9
2.2 Λειτουργία παιχνιδιού.....	9
<b>3 ΜΕΘΟΔΟΛΟΓΙΑ ΥΛΟΠΟΙΗΣΗΣ .....</b>	<b>17</b>
3.1 Δημιουργία βάσης δεδομένων.....	17
3.2 Σχήμα βάσης δεδομένων.....	17
3.3 Υλοποίηση κώδικα.....	20
3.4 Έλεγχος λαθών.....	28
3.5 Προβλήματα που αντιμετώπισα.....	28
3.6 Τεχνολογίες που χρησιμοποίησα.....	29
3.7 Χρόνος για την ολοκλήρωση της πτυχιακής εργασίας.....	29
<b>4 ΣΥΜΠΕΡΑΣΜΑΤΑ.....</b>	<b>31</b>
4.1 Κριτική.....	31
4.2 Μελέτη.....	31
4.3 Επέκταση.....	32
<b>5 ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>33</b>
<b>6 ΠΑΡΑΡΤΗΜΑΤΑ.....</b>	<b>34</b>
6.1 Πίνακας εικόνων.....	34
6.2 Παραπομπές κώδικα.....	38

# 1

## ΕΙΣΑΓΩΓΗ

### 1.1 Περίληψη

Σε αυτή τη διπλωματική θα αναφερθεί ο τρόπος με τον οποίο γίνεται η δημιουργία ενός ηλεκτρονικού παιχνιδιού πολλαπλών χρηστών βασισμένο σε αληθινό χρόνο.

Θα αναφερθούν τα εργαλεία που χρειάζονται για την υλοποίηση του και τα βήματα που θα πρέπει να ακολουθήσει κάποιος για το καλύτερο αποτέλεσμα.

Γίνεται μια αναλυτική περιγραφή της λειτουργίας του παιχνιδιού και τις δυνατότητες των χρηστών σε αυτό. Επίσης αναφέρονται τα προβλήματα που αντιμετωπίσα κατά τη διάρκεια προγραμματισμού και της εφαρμογής του παιχνιδιού, ο χρόνος διεξαγωγής της πτυχιακής και περιέχονται αναλυτικά κάποια μέρη του προγραμματιστικού περιβάλλοντος και της βάσεων δεδομένων.

### 1.2 Κίνητρο για την διεξαγωγή της εργασίας

Πάντα είχα αγάπη για τα παιχνίδια και θυμάμαι πως από μικρή ήθελε να δημιουργήσω ένα, έτσι η ιδέα για την επιλογή της συγκεκριμένης εργασίας προήλθε από ένα παιχνίδι που ξεκίνησα να παίζω πριν από κάποια χρόνια. το shogunwars και θέλησα σαν πρώτο βήμα να φτιάξω ένα παρόμοιο παιχνίδι αλλά με διαφορετικά χαρακτηριστικά, έτσι και το δοκίμασα.

### 1.3 Σκοπός και στόχοι εργασίας

Ο κυριότερος στόχος είναι η δημιουργία ενός ηλεκτρονικού παιχνιδιού στρατηγικής με πολλούς χρήστες όπου θα δημιουργούν υπερφυσικά όντα, θα κάνουν αναβαθμίσεις και θα αγοράζουν όπλα με στόχο την καλύτερη κατάταξη τους στο παιχνίδι.

Άλλος ένας στόχος είναι η εκμάθηση κάποιων βασικών στοιχείων πάνω σε php, mysql, html, css η κατανόηση του πως λειτουργεί ένα παιχνίδι και τα προβλήματα που μπορεί να αντιμετωπίσουμε αλλά και η κατασκευή του παιχνιδιού με όσο πιο απλό τρόπο γίνεται έτσι ώστε να είναι κατανοητό για όσους δεν γνωρίζουν αλλά και εύκολο για όσους θέλουν να δοκιμάσουν να φτιάξουν το δικό τους παιχνίδι.

### 1.4 Δομή εργασίας.



Σε αυτό το σημείο θα κάνω περιληπτική αναφορά στο περιεχόμενο των κεφαλαίων που θα ακολουθήσουν .

### **Κεφάλαιο 2: Σχέδιο δράσης για εκπόνηση της πτυχιακής εργασίας.**

Στο κεφάλαιο αυτό γίνεται αναφορά της μεθοδολογίας που ακολούθησα για την ανάπτυξη της πτυχιακής και γίνεται ανάλυση της λειτουργίας του παιχνιδιού.

### **Κεφάλαιο 3: Μεθοδολογία υλοποίησης.**

Στο κεφάλαιο αυτό αναλύω το πως γίνεται η δημιουργία της βάσης δεδομένων και παρατίθεται το σχήμα της, επίσης αναγράφεται ο κώδικας υλοποίησης και αναφέρονται τα προβλήματα που αντιμετώπισα και πως τα ξεπέρασα αλλά και κάποια τρικ που χρησιμοποίησα. Τέλος γίνεται μια μικρή αναφορά στις τεχνολογίες που χρησιμοποίησα καθώς και ο χρόνος που χρειάστηκε για την ολοκλήρωση της εργασίας.

### **Κεφάλαιο 4: Συμπεράσματα.**

Σε τέταρτο κεφάλαιο αναφέρω τα συμπεράσματα μου πάνω στην όλη διαδικασία ολοκλήρωσης του παιχνιδιού. Επίσης γίνεται μια περιληπτική αναφορά πάνω σε προγράμματα, γλώσσες και άλλα αντικείμενα που χρησιμοποιήθηκαν για να ολοκληρώσω την πτυχιακή εργασία. Τέλος γράφω κάποια επιπλέον στοιχεία που έχω σκεφτεί που μπορούν να επεκτείνουν στο μέλλον το παιχνίδι.

### **Κεφάλαιο 5: Βιβλιογραφία.**

Σε αυτό το κεφάλαιο αναγράφω της πηγές από τις οποίες προμηθεύτηκα αρκετές πληροφορίες. Κυρίως θα βρείτε αρκετά site τα οποία με βοήθησαν στην υλοποίηση της πτυχιακής εργασίας, κυρίως πάνω στο προγραμματιστικό κομμάτι σαν πρώτο βήμα και αργότερα και για κάποιες εικαστικές διορθώσεις που έκανα. Επίσης υπάρχουν και μερικά βιβλία που αναφέρω τα οποία τα βρήκα και αυτά μέσω του διαδικτύου και ήταν εξίσου βοηθητικά στην όλη διαδικασία και αυτά κυρίως πάνω στο κομμάτι του προγραμματισμού.

### **Κεφάλαιο 6: Παραρτήματα.**

Στο τελευταίο κεφάλαιο υπάρχουν οι εικόνες από το σχήμα τις βάσης δεδομένος και οι παραπομπές που καταγράφουν σημεία του κώδικα του παιχνιδιού.



## ΣΧΕΔΙΟ ΔΡΑΣΗΣ ΓΙΑ ΕΚΠΟΝΗΣΗ ΤΗΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

### 2.1 Μέθοδος ανάλυσης και ανάπτυξης πτυχιακής

Το πρώτο πράμα που χρειάστηκε να γίνει πριν προχωρήσω στα πιο πρακτικά και την υλοποίηση του παιχνιδιού, ήταν να σχεδιάσω γραπτά πως θα είναι το παιχνίδι. Δηλαδή τι θα περιλαμβάνει, τι ρόλο θα έχει ο χρήστης στο παιχνίδι και τι ενέργειες θα μπορεί να πραγματοποιήσει.

Εφόσον είχε πάρει μια μορφή το πως ακριβώς θα είναι, προχώρησα με τη δημιουργία των σελίδων του παιχνιδιού. Μετά από αυτό ακολούθησε και η δημιουργία της βάσης δεδομένων και η σύνδεση της με το παιχνίδι. Αργότερα ασχολήθηκα με το εικαστικό κομμάτι και την διόρθωση των λαθών που υπήρχαν και τέλος έγινε η διαδικασία ολοκλήρωσης του παιχνιδιού, που ήταν να ανέβει στο ίντερνετ και να είναι πλέον προσβάσιμο σε χρήστες.

### 2.2 Λειτουργία παιχνιδιού

Σε αυτή την ενότητα θα γίνει μια αναλυτική περιγραφή της λειτουργίας του παιχνιδιού και τις δυνατότητες που έχει ένας χρήστης σε αυτό, όπου θα βοηθήσει αργότερα και στην κατανόηση των υπόλοιπων κεφαλαίων.

Στο παιχνίδι κάθε παίκτης βρίσκεται σε ένα χωριό και έχει τις ίδιες δυνατότητες και επιλογές με τους υπόλοιπους. Βασικό χαρακτηριστικό είναι οι ύλες που παράγει όπως χρήματα, ξύλο, αίμα αλλά και οι χωριάτες που έχει.

Με τις ύλες που παράγει μπορεί να κάνει αναβαθμίσεις και να αγοράσει όπλα. Επίσης οι ύλες μπορούν να χρησιμοποιηθούν για να δημιουργήσει υπερφυσικά όντα, στρατιώτες και ζώα όπως είναι vampires, werewolves, hybrids, witches, hunters και crows. Για την δημιουργία μερικών χρειάζεται όμως να θυσιάσει και κάποιους χωριάτες. Σε ορισμένα ίσως χρειάζεται να θυσιάσει και κάποιους άλλους χαρακτήρες αλλά και να χρησιμοποιήσει κάποιο ποσό από τις ύλες που παράγει.

Ο τρόπος με τον οποίο ένας παίκτης παράγει ύλες έχει σχέση με τον αριθμό των χαρακτήρων που έχει. Για να παράγει χρήματα χρειάζεται τους χωριάτες, για να παράγει ξύλο χρειάζεται τους κυνηγούς και για να παράγει αίμα χρειάζεται τους βρικόλακες.

Με αυτή τη διαδικασία κάθε τι που έχει ο παίκτης, από τον κάθε ένα χαρακτήρα μέχρι την κάθε ύλη που παράγει είναι χρήσιμο για την λειτουργία του παιχνιδιού.

Παρακάτω θα εξηγήσω τις δυνατότητες του κάθε χαρακτήρα, για να κατανοήσετε πως λειτουργεί το παιχνίδι.

**Villagers:** είναι οι βασικότεροι χαρακτήρες του παιχνιδιού μιας και είναι αυτοί που παράγουν τα χρήματα

και θυσιάζονται για να δημιουργηθούν άλλοι χαρακτήρες. Με τα χρήματα που παράγουν οι χωριάτες ο παίκτης μπορεί να κάνει τις αναβαθμίσεις του, να αγοράσει τα όπλα του και να χρησιμοποιήσει κάποια από αυτά για να δημιουργήσει άλλους χαρακτήρες. Όσους περισσότερους χωριάτες έχει ένας παίκτης τόσο περισσότερο χρήμα παράγει. Όλοι οι παίκτες αποκτούν χωριάτες ανά μια ώρα. Ο κάθε παίκτης στο αρχικό του στάδιο έχει 5 χωριάτες και παράγει 25 coins ανά ώρα. Αν όμως ο παίκτης κάνει αναβαθμίσεις μπορεί ο κάθε χωριάτης του να παράγει από 6-10 coins την ώρα. Επίσης υπάρχει και η αναβάθμιση με τη οποία ο παίκτης μπορεί να αποκτήσει 50 επιπλέον χωριάτες.

**Hunter:** είναι υπεύθυνος για την παραγωγή του wood. Για να αποκτήσει ένα παίκτης κυνηγό χρειάζεται να θυσιάσει 1 villager, 500 coins και 150 wood και η δύναμη του είναι στο 40. Ο κάθε κυνηγός στο αρχικό του στάδιο παράγει 10 wood ανά ώρα. Με την κατάλληλη αναβάθμιση μπορεί ο κάθε ένας να παράγει από 13,16,19,22 και 25 wood την ώρα. Άλλη μια αναβάθμιση που αφορά του κυνηγούς είναι τα νέα όπλα που μπορούν να έχουν. Χρησιμοποιούνται για να σκοτώσουν βρικόλακες, υβρίδια και κοράκια.

**Vampire:** είναι υπεύθυνος για το blood που παράγεται. Για να αποκτήσει ένα παίκτης βρικόλακα χρειάζεται να θυσιάσει 1 villager και 30 blood και η δύναμη τους είναι στο 100. Ο κάθε βρικόλακας στο αρχικό του στάδιο παράγει 2 blood ανά ώρα. Με την κατάλληλη αναβάθμιση μπορεί ο κάθε ένας να παράγει από 4,6,8 και 10 blood την ώρα. Χρησιμοποιούνται για να κατασκολεύσουν να κλέψουν χρήματα, να κυνηγήσουν έναν βρικόλακα και να σκοτώσουν έναν κυνηγό, λυκάνθρωπο, υβρίδιο και μια μάγισσα.

**Werewolf:** Για να αποκτήσει ένα παίκτης λυκάνθρωπο χρειάζεται να θυσιάσει 2 villagers και η δύναμη του είναι στο 150. Χρησιμοποιούνται για να κλέψουν χρήματα, να κυνηγήσουν έναν λυκάνθρωπο και να σκοτώσουν έναν κυνηγό, βρικόλακα και μια μάγισσα.

**Hybrid:** Για να αποκτήσει ένα παίκτης υβρίδιο χρειάζεται να θυσιάσει 2 werewolves και 50 blood και η δύναμη του είναι στο 500. Χρησιμοποιούνται για να κλέψουν χρήματα, να κυνηγήσουν υβρίδια και να σκοτώσουν βρικόλακες, κυνηγούς, λυκάνθρωπο και μάγισσες.

**Witch:** Για να αποκτήσει ένα παίκτης μια μάγισσα χρειάζεται να θυσιάσει 1 villager, 1 vampire, 1 werewolf και 1000 coins και η δύναμη της είναι στο 800. Μια αναβάθμιση είναι τα νέα ξόρκια που είναι διαθέσιμα. Χρησιμοποιούνται για να προστατέψουν τον παίκτη από το να τον κατασκολεύσουν και να του κλέψουν χρήματα.

**Crow :** Για να αποκτήσει ένα παίκτης κοράκι χρειάζεται 50 coins και 30 wood και η δύναμη του είναι στο 10. Χρησιμοποιούνται για να κατασκολεύουν τους αντίπαλους παίκτες.

Εδώ θα γίνει περιγραφή των **Upgrades** (αναβαθμίσεων) που μπορεί να επιλέξει ένας παίκτης. Υπάρχουν 6 διαφορετικές αναβαθμίσεις και για να γίνουν, επέλεξα πως ο παίκτης θα πρέπει να δώσει κάποια χρήματα.

**Villager coin production:** Αφορά την παραγωγή χρημάτων από τους χωριάτες και έχει 5 επίπεδα (levels). Σε κάθε αναβάθμιση κάθε χωριάτης φέρνει επιπλέον χρήματα την ώρα. Το αρχικό στάδιο είναι η παραγωγή 5 coins από κάθε χωριάτη την ώρα και στα επόμενα επίπεδα αυξάνεται κατά ένα. Δηλαδή 6,7,8,9 και 10 coins την ώρα. Το κόστος σε κάθε επίπεδο διαφέρει. Ξεκινάει με 3000 coins και κάθε φορά διπλασιάζεται και συνεχίζει με 6.000, 12.000, 24.000 και 48.000.

**Villager population:** Με αυτή την αναβάθμιση ο παίκτης αποκτάει επιπλέον 50 χωριάτες. Κοστίζει 5000 coins για όλα τα επίπεδα και έχει 6 επίπεδα.

**Hunter wood production:** Αφορά την παραγωγή ξύλου από τους κυνηγούς και έχει 5 επίπεδα. Σε κάθε αναβάθμιση κάθε κυνηγός φέρνει επιπλέον ξύλο την ώρα. Το αρχικό στάδιο είναι η παραγωγή 10 wood από κάθε κυνηγό την ώρα και στα επόμενα επίπεδα αυξάνεται κατά τρία. Δηλαδή 13,16,19,22 και 25 wood την ώρα. Το κόστος είναι ίδιο με την παραγωγή χρημάτων.

**Hunter weapon:** Με κάθε νέα αναβάθμιση εμφανίζονται νέα όπλα που μπορεί να αγοράσει ένα παίκτης για τους κυνηγούς του και έχει 6 επίπεδα. Σε πρώτο στάδιο δεν υπάρχουν διαθέσιμα όπλα για αγορά. το κόστος ξεκινάει από 1.000 coins και φτάνει στα 6.000. Τα όπλα που εμφανίζονται πάνε ως εξής. knife, crossbow, grenade, pistol, rifle και flamethrower. Αναλυτικές πληροφορίες για τα όπλα θα βρείτε παρακάτω.

**Vampire blood production:** Αφορά την παραγωγή αίματος από τους βρικόλακες και έχει 4 επίπεδα. Σε κάθε αναβάθμιση κάθε βρικόλακας φέρνει επιπλέον αίμα την ώρα. Το αρχικό στάδιο είναι η παραγωγή 2 blood από τον κάθε ένα την ώρα και στα επόμενα επίπεδα αυξάνεται κατά 2. Δηλαδή 4,6,8 και 10 blood την ώρα. Το κόστος είναι ίδιο με την παραγωγή χρημάτων.

**Witch spell:** Με κάθε νέα αναβάθμιση εμφανίζονται νέα ξόρκια που μπορεί να κάνει ένα παίκτης και έχει 3 επίπεδα. Σε πρώτο στάδιο δεν υπάρχουν διαθέσιμα ξόρκια για χρήση. Το κόστος είναι στα 4.000 coins για όλα τα επίπεδα. Τα ξόρκια που εμφανίζονται πάνε ως εξής: full spy, undercover και coin protection. Αναλυτικές πληροφορίες θα βρείτε παρακάτω.

Στο σημείο αυτό θα περιγράψω τα **weapons** (όπλα) και τα **spells** (ξόρκια) που υπάρχουν στο παιχνίδι τις δυνατότητές του και πως μπορεί να τα αποκτήσει ένας παίκτης.

Τα όπλα που μπορεί να έχει ένας παίκτης είναι 6 και χρησιμοποιούνται μόνο από τους κυνηγούς.

Η δύναμη δηλαδή που έχει κάθε όπλο προστίθεται στην συνολική δύναμη των κυνηγών μόνο.

**Knife** έχει 10 power και χρειάζεται 1.000 coins και 500 wood.

**Crossbow** έχει 50 power και χρειάζεται 3.000 coins και 1.500 wood.

**Grenade** έχει 60 power και χρειάζεται 5.000 coins και 2.500 wood.



*Pistol* έχει 70 power και χρειάζεται 8.000 coins και 4.000 wood.

*Rifle* έχει 110 power και χρειάζεται 10.000 coins και 5.000 wood.

*Flamethrower* έχει 170 power και χρειάζεται 50.000 coins και 25.000 wood.

Επίσης υπάρχει και η επιλογή να καταστρέψει κάποιος τα όλα που δεν χρειάζεται. Με αυτή την επιλογή χάνει τη δύναμη που του έδινε ένα όπλο και κερδίζει κάποια χρήματα.

Όσον αναφορά τα ξόρκια, είναι 3 στο σύνολο.

*Full spy* κάνει πλήρη κατασκοπία των στοιχείων ενός αντιπάλου, η διάρκειά του είναι 24 ώρες και κοστίζει 100 blood.

*Undercover* προστατεύει το παίκτη από τον κίνδυνο να τον κατασκοπεύσουν οι αντίπαλοί του. Η διάρκειά του είναι 4 ώρες και κοστίζει 250 blood.

*Coin protection* προστατεύει το παίκτη από τον κίνδυνο να του κλέψουν λεφτά οι αντίπαλοί του. Η διάρκειά του είναι 2 ώρες και κοστίζει 500 blood.

Τέλος θα περιγράψω τις επιλογές σε μια μάχη (**battles**) που έχουν δύο παίκτες. Συνολικά είναι 12 διαφορετικές ενέργειες κάποιες από τις οποίες όμως δεν είναι τόσο επιθετικές. Οι πρώτες στη σειρά είναι και αυτές που είναι πιο απλές. Σε όλες τις ενέργειες ο επιτιθέμενος και ο αμυνόμενος χρησιμοποιεί ξεχωριστές δυνάμεις που μετρώνται με έναν αριθμό. Ο παίκτης που έχει τον μεγαλύτερο είναι και ο νικητής σε μια μάχη. Αναλυτικά θα δείτε παρακάτω.

**Spy:** με αυτή την ενέργεια ο παίκτης μπορεί να κατασκοπεύσει κάποια στοιχεία του αντιπάλου του, όπως είναι ο αριθμός των χαρακτήρων που έχει, τα όπλα και η συνολική δύναμη του. Η δύναμη που έχει ένας παίκτης ως επιτιθέμενος (*attacker*) και αμυνόμενος (*defender*) για την συγκεκριμένη ενέργεια πάει ως εξής:

Attacker: number of crows × vampire total power.

Defender: witch total power.

Δηλαδή ο επιτιθέμενος χρησιμοποιεί ως δύναμη τον αριθμό των κορακιών του επί την συνολική δύναμη των βρικόλάκων του και ο αμυνόμενος χρησιμοποιεί ως δύναμη την συνολική δύναμη των μαγισσών του.

Ένα παράδειγμα που θα βοηθήσει και στην κατανόηση των παρακάτω:

Αν ο επιτιθέμενος έχει 10 κοράκια και η συνολική δύναμη των βρικόλάκων του είναι 1300, τότε ο αριθμός για τον επιτιθέμενο είναι στο 13000. Αν ο αμυνόμενος έχει συνολική δύναμη μαγισσών 800 τότε ο επιτιθέμενος είναι και ο νικητής.

**Steal Coins:** ένας παίκτης μπορεί να κλέψει τα λεφτά από έναν άλλο παίκτη.

Attacker: vampires + werewolves + hybrids total power.

Χρησιμοποιείται το άθροισμα της συνολικής δύναμης των βρικόλάκων, λυκανθρώπων και υβριδίων.

Defender: witch total power.

Ο αμυνόμενος χρησιμοποιεί την συνολική δύναμη των μαγισσών του.

**Hunt Vampire:** ένας παίκτης μπορεί να κυνηγήσει έναν βρικόλακα από έναν άλλο και να τον πάρει με το μέρος του.

Attacker: number of vampires.

Defender: number of vampires.

Στη συγκεκριμένη ενέργεια ο παίκτης που έχει τον μεγαλύτερο αριθμό βρικόλακων είναι και ο νικητής.

**Hunt Werewolf:** ένας παίκτης μπορεί να κυνηγήσει έναν λυκάνθρωπο από έναν άλλο και να τον πάρει με το μέρος του.

Attacker: number of werewolves

Defender: number of werewolves

Το ίδιο ισχύει και σε αυτή την ενέργεια. Ο παίκτης που έχει τον μεγαλύτερο αριθμό λυκανθρώπων είναι και ο νικητής

**Hunt Hybrid:** ένας παίκτης μπορεί να κυνηγήσει ένα υβρίδιο από έναν άλλο και να τον πάρει με το μέρος του.

Attacker: number of hybrids

Defender: number of hybrids

Επίσης και εδώ ο παίκτης που έχει τον μεγαλύτερο αριθμό υβριδίων είναι και ο νικητής.

**Full Spy:** με αυτή την ενέργεια ο παίκτης χρησιμοποιεί ένα ξόρκι κατασκοπίας στον αντίπαλό του.

Οι πληροφορίες που δέχεται μετά από την ενέργεια αυτή είναι περισσότερες από ότι σε μια απλή κατασκοπία. Ο παίκτης βλέπει τις ύλες που παράγει ο αντίπαλός του, τους χαρακτήρες που έχει, τα όπλα, τη συνολική δύναμη και τις αναβαθμίσεις που έχει κάνει.

Attacker: cost 100 blood.

Αυτή η ενέργεια έχει αναλυθεί παραπάνω.

**Kill Hunter:** ένας παίκτης σκοτώνει έναν κυνηγό του αντιπάλου του.

Attacker: Vampires or Hybrids total power

Defender: hunters total power.

Ο επιτιθέμενος χρησιμοποιεί είτε την συνολική δύναμη των βρικόλακων που έχει είτε την συνολική δύναμη των υβριδίων. Ο αμυνόμενος χρησιμοποιεί την συνολική δύναμη των κυνηγών του.

**Kill Vampire:** ένας παίκτης σκοτώνει έναν βρικόλακα του αντιπάλου του.

Attacker: Hunters or Werewolves or Hybrids total power

Defender: vampires total power

Ο επιτιθέμενος χρησιμοποιεί ή την συνολική δύναμη των κυνηγών, ή την συνολική δύναμη των λυκανθρώπων ή την συνολική δύναμη των υβριδίων. Ο αμυνόμενος χρησιμοποιεί την συνολική δύναμη των

βρικόλακων του.

**Kill Werewolf:** ένας παίκτης σκοτώνει έναν λυκάνθρωπο του αντιπάλου του.

Attacker: Vampires or Hybrids total power

Defender: werewolves total power

Ο επιτιθέμενος χρησιμοποιεί ή την συνολική δύναμη των βρικόλακων ή την συνολική δύναμη των υβριδίων.

Ο αμυνόμενος χρησιμοποιεί την συνολική δύναμη των λυκανθρώπων.

**Kill Hybrid:** ένας παίκτης σκοτώνει ένα υβρίδιο του αντιπάλου του.

Attacker: Vampires+Werewolves or Hunters total power

Defender: hybrids total power  $\times$  2

Ο επιτιθέμενος χρησιμοποιεί το άθροισμα την συνολική δύναμη των βρικόλακων του και των λυκανθρώπων.

Ο αμυνόμενος χρησιμοποιεί την συνολική δύναμη των υβριδίων επί 2.

**Kill Witch:** ένας παίκτης σκοτώνει μια μάγισσα του αντιπάλου του.

Attacker: vampires + werewolves + hybrids total power

Defender: witches total power

Ο επιτιθέμενος χρησιμοποιεί το άθροισμα την συνολική δύναμη των βρικόλακων, των λυκανθρώπων και των υβριδίων. Ο αμυνόμενος χρησιμοποιεί την συνολική δύναμη των μαγισσών του.

**Kill Crow:** ένας παίκτης σκοτώνει ένα κοράκι του αντιπάλου του.

Attacker: hunters number + (crossbow + pistol + rifle) power.

Defender: crows total power

Ο επιτιθέμενος χρησιμοποιεί το άθροισμα από τον συνολικό αριθμό των κυνηγών που έχει και των δυνάμεων από τα παραπάνω όπλα . Ο αμυνόμενος χρησιμοποιεί την συνολική δύναμη των κορακιών του.

Σε αυτό το σημείο να αναφέρω πως όλα τα παραπάνω στοιχεία που έχω δώσει, τα περισσότερα από αυτά αναγράφονται και στο παιχνίδι, όπως θα δείτε παρακάτω, για να είναι κατανοητό και από τους παίκτες πώς λειτουργεί.

Hunt Vampire	Hunt Werewolf	Hunt Hybrid
<i>attacker:</i> number of vampires.	<i>attacker</i> number of werewolves.	<i>attacker:</i> number of hybrids.
<i>defender</i> number of vampires.	<i>defender:</i> number of werewolves.	<i>defender</i> number of hybrids.

Εφόσον έχει γίνει μια καλή ανάλυση των παραπάνω θα περάσω στον τρόπο με τον οποίο γίνεται ο υπολογισμός της κατάταξης ενός παίκτη, μια και σκοπός του παιχνιδιού είναι ένας παίκτης να φτάσει σε όσο το δυνατόν καλύτερη βαθμολογία, οπότε και επέλεξα να υπάρχουν αρκετά χαρακτηριστικά στα οποία ένας



παίκτης θα πρέπει να δώσει βάση για να έχει και καλύτερη κατάταξη.

Ο υπολογισμός γίνεται με βάση τον συνολικό αριθμό των χαρακτήρων που έχει ένας παίκτης, τη συνολική δύναμη, το επίπεδο κάθε αναβάθμισης και της μάχης που έχει κερδίσει. Όλα αυτά βέβαια υπολογίζονται με έναν πιο αναλυτικό τρόπο και για την καλύτερη κατανόηση έχω κάνει τον παρακάτω σχεδιασμό.

**Rank= 1.Total warriors + 2.Total power + 3.Upgrades Level + 4.Battles won**

#### *1.Total warriors:*

number of hunters × 4 +  
number of vampires × 5 +  
number of werewolves × 5 +  
number of hybrids × 6 +  
number of crows × 3 +  
number of witches × 7 +  
number of villagers × 8

Εδώ ο υπολογισμός των συνολικών χαρακτήρων γίνεται με το μέτρημα των αριθμών του κάθε χαρακτήρα χωριστά και πολλαπλασιάζεται το κάθε σύνολο με έναν βαθμό που έχω δώσει εγώ.

Οι αριθμοί που έχω επιλέξει να αντιστοιχούν στον κάθε χαρακτήρα έχει βάση με το πόσο δύσκολο είναι να τους αποκτήσεις. Έτσι για κάθε κοράκι έχω δώσει να αντιστοιχεί ο βαθμός 3 και για τους χωριάτες ο βαθμός 8.

#### *2.Total power:*

hunters total power +  
vampires total power +  
werewolves total power +  
hybrids total power +  
witches total power +  
crows total power

Σε αυτό το σημείο προστίθεται η συνολική δύναμη των παραπάνω χαρακτήρων. Ο κάθε χαρακτήρας έχει ξεχωριστή δύναμη (όπως αναγράφεται και στη σελίδα δημιουργίας τους στο παιχνίδι). Για τον υπολογισμό τους λοιπόν υπολογίζεται ο αριθμός του κάθε χαρακτήρα ξεχωριστά επί της δύναμης του. Δηλαδή αν ένας παίκτης έχει 5 κυνηγούς, τότε ο αριθμός αυτός πολλαπλασιάζεται με την δύναμη που έχει ένας κυνηγός που είναι 40. Άρα η συνολική δύναμη των κυνηγών του παίκτη είναι 200. Μια παραπάνω λεπτομέρεια είναι πως στους κυνηγούς υπολογίζεται και η κατοχή των όπλων που έχουν. Δηλαδή υπολογίζεται ο αριθμός των όπλων ξεχωριστά επί την δύναμή τους και προστίθεται με την συνολική δύναμη των κυνηγών.

### 3. Upgrades Level:

villagers coin pr.  $\times 5000 +$   
villagers population  $\times 1500 +$   
hunters weapon  $\times 2000 +$   
hunters wood pr.  $\times 5000 +$   
vampires blood pr.  $\times 5000 +$   
witch spells  $\times 2000$

Στις αναβαθμίσεις όπως και παραπάνω έχω βάλει έναν βαθμό δίπλα από την κάθε μία με βάση το επίπεδό της. Κάθε αναβάθμιση θεωρήσα ότι πρέπει να αξιολογηθεί διαφορετικά.

### 4. Battles won:

battles won  $\times 50$

Τέλος γίνεται και η πρόσθεση των βαθμών από τις μάχες που έχει κερδίσει ένας παίκτης.

Κάτι που πρέπει να προσέξει ένας παίκτης σε αυτό το σημείο είναι πως μπορεί να κερδίσει πόντους μάχης όταν ένας αντίπαλός του επιτεθεί και κερδίσει. Μπορεί δηλαδή να είναι σε άμυνα ο παίκτης αλλά σε συγκεκριμένες επιθέσεις αν ο αντίπαλος δεν κερδίσει τότε ο αμυνόμενος παίρνει μια νίκη.

Όλα τα παραπάνω δίνουν μια παραπάνω σημασία στον τρόπο που θα επιλέξει ένας παίκτης να κινηθεί στο παιχνίδι για να αποκτήσει μια καλή θέση στην κατάταξη.

Εκτός από την λειτουργία του παιχνιδιού θέλω να αναφέρω τις επιπλέον επιλογές του παιχνιδιού και ενέργειες που έχει ένας χρήστης σε αυτό.

Πολύ σημαντικό για έναν παίκτη είναι να διαβάσει πως λειτουργεί το παιχνίδι. Κυρίως αν δεν έχει παίξει κάτι παρόμοιο ίσως μπερδευτεί για το πως λειτουργεί. Έτσι έχω φτιάξει έναν οδηγό παιχνιδιού (**game guide**) που κάθε χρήστης έχει την δυνατότητα να τον κατεβάσει. Ακόμα έχω προσθέσει την πρώτη μέρα έναρξης και λήξης της σεζόν του παιχνιδιού, αυτό βοηθάει στο να ξέρει ο παίκτης πώς θα κινηθεί με βάση το χρόνο που απομένει μέχρι να τελειώσει το παιχνίδι.

Το θεωρήσα σημαντικό επίσης να υπάρχει επικοινωνία μεταξύ παικτών, οπότε στο προφίλ του κάθε παίκτης υπάρχει η επιλογή να του στείλουμε κάποιο μήνυμα και επίσης φαίνεται και πότε έπαιξε για πρώτη φορά το παιχνίδι.

Τέλος κάθε χρήστης έχει τη επιλογή να αλλάξει κωδικό στον λογαριασμό του αλλά και να διαγραφεί από το παιχνίδι. Μετά από αυτή την ενέργεια διαγράφονται όλα τα στοιχεία του παίκτη από τη βάση δεδομένων και βγαίνει έξω από το παιχνίδι.

## ΜΕΘΟΔΟΛΟΓΙΑ ΥΛΟΠΟΙΗΣΗΣ

Το κεφάλαιο αυτό αναφέρεται στα πιο πρακτικά κομμάτια της πτυχιακής εργασίας, όπως είναι η δημιουργία της βάσης δεδομένων και το σχήμα της, ο κώδικας υλοποίησης, τα προβλήματα που αντιμετώπισα και πως τα ξεπέρασα αλλά και κάποια τρικ που χρησιμοποίησα. Τέλος γίνεται μια μικρή αναφορά στις τεχνολογίες που χρησιμοποίησα καθώς και ο χρόνος που χρειάστηκε για την ολοκλήρωση της παρούσας εργασίας.

### 3.1 Δημιουργία βάσης δεδομένων

Μια βάση δεδομένων είναι απαραίτητη για κάθε παιχνίδι, καθώς μέσα στη βάση θα αποθηκεύονται όλοι οι χρήστες που κάνουν εγγραφή καθώς και τα κάποια από τα στοιχεία τους που θα δίνουν για την εγγραφή τους. Επίσης μέσα στη βάση θα αποθηκεύονται τα στοιχεία του κάθε παίκτη σχετικά με το παιχνίδι. Εκτός όμως από τα στοιχεία των χρηστών-παικτών μέσω της βάσης δεδομένων θα γίνει και η σύνδεση του πραγματικού χρόνου με το παιχνίδι και θα γίνονται οι απαραίτητες ενέργειες για την σωστή λειτουργία του. Για την καλύτερη χρήση και λειτουργία του παιχνιδιού καλό είναι η βάση μας να αποτελείται από όσο πιο πολλά tables έτσι ώστε το κάθε ένα να περιέχει όσο το δυνατόν λιγότερα columns.

Η βάση δεδομένων που έφτιαξα εγώ αποτελείται από 9 tables (εικόνα 1) και σε παρακάτω ενότητα θα εξηγήσω τα περιεχόμενα του κάθε table.

### 3.2 Σχήμα βάσης δεδομένων

Εδώ θα αναλύσω τη βάση δεδομένων (εικόνα 1) που έχω φτιάξει, δηλαδή τα tables που επέλεξα να δημιουργήσω και τα columns που περιέχει το κάθε ένα. Σε κάθε table υπάρχει για ευκολία και μια φωτογραφία που θα βρείτε στα παραρτήματα. Συνολικά έφτιαξα 12 tables και τα αναγράφω παρακάτω με την σειρά που θα ήταν ευκολότερο να φτιαχτούν με βάση τα στάδια που ακολούθησα για την δημιουργία του παιχνιδιού.

**1) players** (εικόνα 2): περιέχει 17 columns. Το *Id* που είναι ένας μοναδικός αριθμός για κάθε χρήστη και αυξάνεται από μόνο του, *username* το όνομα, *password* ο κωδικός, *email* το email του κάθε χρήστη.

*active* δέχεται τις τιμές 0 και 1 αν είναι ανενεργός η ενεργός ο χρήστης αντίστοιχα

*hash* εδώ εισάγετε ένας μοναδικός αριθμός που στέλνετε στο mail του κάθε χρήστη για την ενεργοποίηση του λογαριασμού του

*warriors* το σύνολο των στρατιωτών που έχει κάθε παίκτης

*villagers* οι χωριάτες, *vampires* οι βρικόλακες, *werewolves* οι λυκάνθρωποι, *hunters* οι κυνηγοί, *hybrid* τα υβρίδια, *witches* οι μάγισσες και *crow* τα κοράκια που έχει ο παίκτης.

**2) time** (εικόνα 3): περιέχει 8 columns.

*fdate* αφορά την πρώτη μέρα της σεζόν του παιχνιδιού η οποία ορίζεται από τον administrator,

*ldate* αφορά την τελευταία μέρα της σεζόν του παιχνιδιού η οποία ορίζεται από τον administrator,

*username*, *logindate* η μέρα εγγραφής του χρήστη και *loginhour* η ώρα εγγραφής τα οποία παραμένουν σταθερά.

*last\_day* αφορά την τελευταία μέρα της σεζόν του παιχνιδιού η οποία ορίζεται από τον administrator.

**3) resources** (εικόνα 4): περιέχει 8 columns.

Σε αυτό το table καταγράφεται η ποσότητα της κάθε ύλης που παράγει ο κάθε παίκτης (*coin\_p*, *wood\_p*, *blood\_p*) με βάση τις αναβαθμίσεις που έχει κάνει, αλλά και η ποσότητα παραγωγής της κάθε ύλης ανά μια ώρα (*coins\_perhour*, *wood\_perhour*, *blood\_perhour*). Έτσι όταν πρέπει να προστεθούν οι ύλες στους παίκτες κάθε μια ώρα γίνεται ο υπολογισμός των τιμών μέσω αυτού του πίνακα. Επίσης καταγράφεται η ποσότητα των χρημάτων (*oins*) που έχει, τα χρήματα που έχει προστατευμένα (*chest*) από τυχόν επιθέσεις άλλων παικτών αλλά και η συνολική ποσότητα από ξύλο (*wood*) και αίμα (*blood*).

**4) armory** (εικόνα 5): περιέχει 11 columns.

Σε αυτό το table καταγράφονται τα ξόρκια που χρησιμοποιεί ένας παίκτης (*spellFS*, *spellCP*, *spellUN*) που δέχονται τις τιμές 0 ή 1 και αν το ξόρκι είναι ενεργοποιημένο έχει την τιμή 1.

Επίσης καταγράφεται η ώρα που χρησιμοποιήθηκε για το κάθε ξόρκι χωριστά (*spellFS\_time*, *spellCP\_time*, *spellUN\_time*), η ώρα που το κάθε ξόρκι παύει να είναι ενεργό (*spellFS\_expires*, *spellCP\_expires*, *spellUN\_expires*) καθώς και η παγκόσμια ώρα (*time*).

Με αυτό τον τρόπο γίνεται σύγκριση της παγκόσμιας ώρας και τις ώρα που λήγει το ξόρκι και όταν αυτή η ώρα περάσει τότε το ξόρκι παίρνει πάλι την τιμή 0 και μπορεί να χρησιμοποιηθεί ξανά από τον παίκτη.

**5) weapons** (εικόνα 6): περιέχει 18 columns.

Το όνομα του χρήστη (*username*), την συνολική ποσότητα των όπλων που διαθέτει (*weapons*) και την ποσότητα του κάθε όπλου χωριστά (*knife*, *crossbow*, *pistol*, *grenade*, *rifle*, *flamethrower*)

**6) upgrades** (εικόνα 7): περιέχει 13 columns.

Το όνομα του χρήστη (*username*), το επίπεδο αναβάθμισης που βρίσκεται ένας παίκτης για την κάθε αναβάθμιση (*up\_v\_por*, *up\_v\_c*, *up\_wi\_s*, *up\_na\_b*, *up\_h\_w*, *up\_h\_wo*) αλλά και την τιμή που κοστίζει η επόμενη αναβάθμιση για το κάθε επίπεδο ξεχωριστά (*up1*, *up2*, *up3*, *up4*, *up5*, *up6*).

**7) battles\_stats** (εικόνα 8): περιέχει 4 columns.

Σε αυτό τον πίνακα αποθηκεύονται οι επιθέσεις που έχουν γίνει μεταξύ των παικτών. Το όνομα του παίκτη



που έκανε την επίθεση (*attacker*), το όνομα του παίκτη που δέχτηκε την επίθεση (*defender*), τι είδους επίθεση έγινε (*action*) και τέλος η ημέρα και η ώρα της επίθεσης (*date\_time*). Αυτά τα στοιχεία είναι χρήσιμα για κάθε παίκτη καθώς μπορούν να βλέπουν τις επιθέσεις που έχουν κάνει σε άλλους παίκτες αλλά κυρίως τις επιθέσεις που δέχτηκαν.

**8) battles** (εικόνα 9): περιέχει 27 columns.

Το όνομα του παίκτη που κάνει την επίθεση (*username*), το όνομα του παίκτη που δέχεται την επίθεση (*defender*), τον συνολικό αριθμό των μαχών που έχει δώσει ένας παίκτης (*battleA*), τον αριθμό των μαχών που έχει κερδίσει (*battleW*) και έχει χάσει ο παίκτης (*battleL*), τον αριθμό των μαχών που έχει δώσει εάνς παίκτης ως επιτιθέμενος (*battleAT*) και ως αμυνόμενος (*battleDE*), την ποσότητα χρημάτων που έχει χάσει (*lost\_coin*) και έχει κερδίσει (*win\_coin*) ένας παίκτης από μάχες και τον αριθμό των διαφορετικών χαρακτήρων που έχει χάσει και έχει κερδίσει ένας παίκτης από τις μάχες *.win\_vampires, lost\_vampires, win\_werewolves, lost\_werewolves, win\_hybrids, lost\_hybrids, hunter\_killes, hunter\_losses, vampire\_kills, werewolf\_losses, werewolf\_kills, hybrid\_kills, hybrid\_losses, crow\_kills, crow\_losses*

**9) total\_power** (εικόνα 10): περιέχει 9 columns.

Σε αυτό το table εκτός από το όνομα του χρήστη, καταγράφονται οι δυνάμεις που έχει ο παίκτης από κάθε χαρακτήρα δηλαδή του χωριάτες, τους βρυκόλακες, τους λυκάνθρωπους, τους κυνηγούς, τα υβρίδια, τις μάγισσες και τα κοράκια (*tp\_v, tp\_va, tp\_we, tp\_h, tp\_hub, tp\_wi, tp\_c*) και η συνολική δύναμη όλων αυτών (*totalpower*) που βοηθάει για την κατάταξη του παίκτη.

**10)messages** (εικόνα 11): περιέχει 6 columns.

Περιέχει το όνομα του χρήστη που στέλνει το μήνυμα (*username*), τον κωδικό του μηνύματος (*message\_id*) που είναι και μοναδικός για το κάθε μήνυμα, το όνομα του χρήστη που στέλνουμε το μήνυμα (*to\_user*), ο τίτλος του μηνύματος (*message\_title*), το περιεχόμενο του μηνύματος (*message\_contents*) και τέλος άλλη μια μεταβλητή (*message\_read*) που δέχεται την τιμή 0 και 1 σε περίπτωση που το μήνυμα έχει διαβαστεί από τον χρήστη ή όχι.

**11) end** (εικόνα 12): περιέχει 7 columns.

Στον πίνακα αυτό που είναι αρκετά σημαντικός για το τέλος του παιχνιδιού, καταγράφονται όλα τα στοιχεία των παικτών που είναι απαραίτητα για την κατάταξή τους. Δηλαδή έχουμε αρχικά το όνομα του κάθε χρήστη και τον αριθμό κατάταξής του (*rank*), τον συνολικό βαθμό που έχει συγκεντρώσει από όλα τα απαραίτητα στοιχεία (*R*) όπου με βάσει αυτόν όποιος παίκτης έχει και τον μεγαλύτερο βαθμό είναι και ο πρώτος στην κατάταξη. Και τα υπόλοιπα στοιχεία που είναι ο συνολικός βαθμός που έχει ο παίκτης από τους χαρακτήρες (*totalwarriorsrank*), ο συνολικός βαθμός που έχει ο παίκτης από τις δυνάμεις (*totalpowerrank*), ο συνολικός βαθμός που έχει ο παίκτης από τις αναβαθμίσεις (*totalupgraderank*) και ο συνολικός βαθμός που έχει ο παίκτης από τις μάχες που έχει κερδίσει (*battlesrank*).

12) seasons (εικόνα 13): περιέχει 6 columns.

Ο τελευταίος πίνακας της βάσης δεδομένων όπου είναι χρήσιμος για το τέλος τις κάθε σεζόν του παιχνιδιού. Σε αυτόν τον πίνακα θέλησα να κρατάω σαν αρχείο τους νικητές του παιχνιδιού από την κάθε σεζόν, έτσι σε κάθε τέλος του παιχνιδιού καταγράφεται ο αριθμός της σεζόν (*season*), τα ονόματα κάποιων παικτών (*username*), η κατάταξή τους (*rank*) καθώς και κάποια στοιχεία του παίκτη αυτού που τον καθόρισαν νικητή, όπως είναι το σύνολο των χαρακτήρων (*total\_warriors*) που είχε κατά τη διάρκεια της σεζόν, η συνολική του δύναμη (*total\_power*) και οι μάχες που κέρδισε (*battles\_won*).

### 3.3 Υλοποίηση κώδικα

Αυτή η υπο-ενότητα της παρούσας εργασίας, περιλαμβάνει κομμάτια του κώδικα που γράφτηκαν για την σχεδίαση – υλοποίηση του παιχνιδιού. Ο κώδικας αυτός είναι γραμμένος σε γλώσσα προγραμματισμού PHP και υπάρχουν και κομμάτια γραμμένα σε CSS και HTML. Για την εγγραφή του κώδικα χρησιμοποιήθηκε το πρόγραμμα eclipse.

Σαν πρώτο βήμα αφού έχει γίνει η δημιουργία της βάσης δεδομένων, φτιάχνω το αρχείο στο οποίο θα γίνεται η σύνδεση της βάσης με το παιχνίδι (**connect.php**).

```
$con = mysql_connect('host', 'username', 'password');  
mysql_select_db('database_name', $con);
```

Στην πρώτη σειρά βάζουμε τα αντίστοιχα στοιχεία μας και στην δεύτερη γίνεται η επιλογή της βάσης δεδομένων γράφοντας το όνομα που της έχουμε δώσει.

Για τον λόγο ότι σε όλα τα επόμενα php αρχεία που θα φτιαχτούν θα γίνει η σύνδεση με την βάση, θέλησα κυρίως για λόγους ευκολίας στην εύρεση αρχείων και σε άλλες περιπτώσεις για να μην υπάρχει μεγάλος κώδικας γραμμένος σε κάθε αρχείο, να συμπεριλάβω την σύνδεση της βάσης μέσα στα αρχεία. Καλώ δηλαδή ένα php αρχείο μέσα σε ένα άλλο ως εξής:

```
<?php  
include 'connect.php' ;  
?>
```

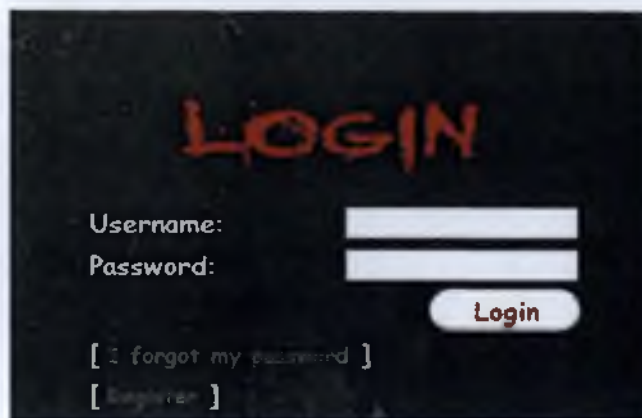
Σε επόμενο βήμα έφτιαξα τις βασικές σελίδες του παιχνιδιού. Δηλαδή την πρώτη σελίδα που θα βλέπει ο χρήστης όταν δει για πρώτη φορά το παιχνίδι καθώς και τις σελίδες που είναι απαραίτητες για την εγγραφή του και αυτές που θα είναι κύριες για την λειτουργία του παιχνιδιού.

Αρχικά, ξεκίνησα με τη φόρμα σύνδεσης των χρηστών στο παιχνίδι καθώς θα είναι η πρώτη σελίδα που θα βλέπει κάποιος.

Έχουμε λοιπόν τα εξής αρχεία:



1) **index.php**<sup>1</sup>: η πρώτη σελίδα του παιχνιδιού με την φόρμα σύνδεσης χρήστη, το κουμπί για την εγγραφή νέου χρήστη και το κουμπί για αποστολή νέου κωδικού.



A screenshot of a login form on a black background. The word "LOGIN" is written in large, red, stylized letters at the top. Below it, there are two input fields: "Username:" and "Password:". To the right of the "Password:" field is a "Login" button. At the bottom left, there are two links: "[ I forgot my password ]" and "[ Register ]".

2) **connection.php**<sup>2</sup>: Η σελίδα αυτή δεν είναι ορατή από τον χρήστη. Γίνεται η σύνδεση με την βάση δεδομένων και έλεγχος σε αυτήν για το αν τα στοιχεία που έδωσε ο χρήστη υπάρχουν μέσα σε αυτήν. Αν υπάρχουν τότε ο χρήστης οδηγείται στην αρχική σελίδα του παιχνιδιού (main page), αλλιώς πρέπει να κάνει εγγραφή.

3) **register.php**<sup>3</sup>: η σελίδα με τη φόρμα εγγραφής νέου χρήστη. Απαιτείται ένα όνομα χρήστη, κωδικός, επαλήθευση κωδικού και email, γίνεται έλεγχος των στοιχείων που εισάγει ο χρήστης, εισάγονται τα στοιχεία του στη βάση δεδομένων και στέλνεται ένα μήνυμα στο email του κάθε χρήστη για ενεργοποίηση του λογαριασμού του.



A screenshot of a register form on a black background. The word "REGISTER" is written in large, red, stylized letters at the top. Below it, there are four input fields: "Username:", "Password:", "Confirm Password:", and "E-mail:". To the right of the "E-mail:" field is a "Register" button. At the bottom left, there is a link: "[ Login ]".

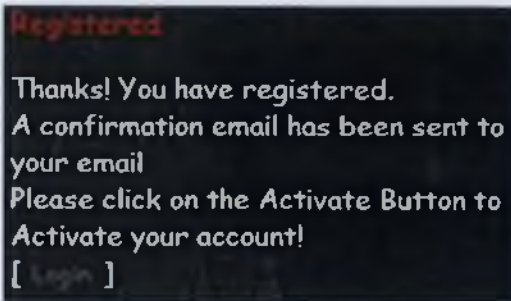
4) **verify.php**<sup>4</sup>: Όταν ο χρήστης πατήσει τον σύνδεσμο που του έχει σταλεί στο email, στέλνεται σε αυτή τη σελίδα και γίνεται έλεγχος του email

Ανακτήθηκε από:

<http://www.infotuts.com/email-verification-php/>

<http://stackoverflow.com/questions/14623923/verification-link-not-activating-account>

μήνυμα για την ενεργοποίηση του λογαριασμού ενός παίκτη από το προσωπικό του email:

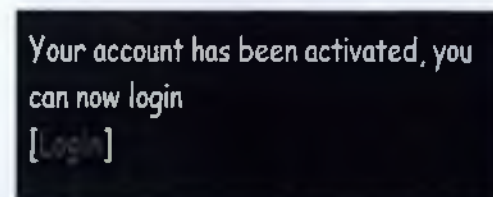


Thanks for signing up!  
Your account has been created, you can login with the following credentials after you have activated your account by pressing the url below

-----  
Username: katenna  
Password: kate  
-----

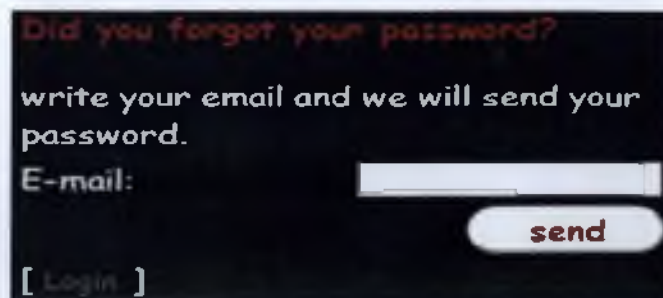
Please click this link to activate your account:  
<http://www.blood-valley.com/veniv.php?email=geroulakou.k@gmail.com&hash=50c3d7614917b21303ee6a220679dab3>

ηλεκτρονικό μήνυμα για την ενεργοποίηση του λογαριασμού:



μήνυμα μετά από την ενεργοποίηση του λογαριασμού μέσω του email:

5) **forgotpass.php** και **process.php**<sup>5</sup>: Στη πρώτη σελίδα υπάρχει μια φόρμα όπου ο χρήστης πληκτρολογεί το email σε περίπτωση που έχει ξεχάσει τον κωδικό του και με το πάτημα του κουμπιού αποδοχής, μεταφέρεται στη δεύτερη σελίδα όπου εκεί γίνεται η διαδικασία αποστολής του κωδικού στον email του χρήστη.



Το επόμενο μου βήμα είναι να φτιάξω τις βασικές σελίδες που θα χρησιμοποιεί ένας παίκτης στο παιχνίδι. Έχουμε λοιπόν τα εξής αρχεία.

1) **main.php**<sup>6</sup> αναγράφονται γενικές πληροφορίες του παίκτη για το παιχνίδι. Δηλαδή τι ύλες παράγει με βάση τις αναβαθμίσεις που έχει κάνει, η επιλογή να τοποθετήσει τα χρήματά του στο σεντούκι, τις μάχες

που έχει δώσει με άλλους παίκτες, τις νίκες και τις απώλειες που είχε από αυτές.

Σε αυτό το σημείο, αν ο παίκτης επιλέξει να βάλει κάποια χρήματα του στο σεντούκι οδηγείτε στο αρχείο **coins.php**<sup>7</sup> όπου και γίνεται η μεταφορά των χρημάτων και η επιστροφή στην αρχική σελίδα.



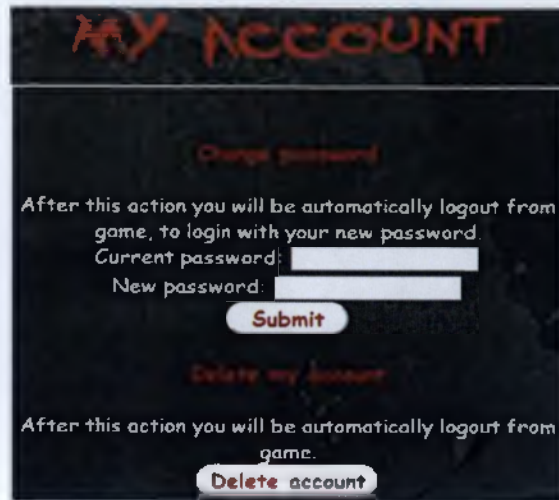
Επίσης υπάρχει και ένας σύνδεσμος που τον οδηγεί σε μια σελίδα προβολής αναλυτικών στοιχείων από κάθε μάχη, το **battlefieldstats.php**<sup>8</sup>

date time	attacker	action	defender
2014-12-25 11:20:55	marouska	steal coins	katerina
2014-12-19 08:16:52	marouska	spy mission	katerina
2014-12-19 08:15:42	marouska	spy mission	katerina

Ακόμη, στην πρώτη σελίδα υπάρχει η επιλογή ο παίκτης να δει τα εισερχόμενα και εξερχόμενα μηνύματα (**inbox.php**<sup>9</sup>, **send.php**<sup>10</sup>) και τέλος η επιλογή να δει το προφίλ του και τον λογαριασμό του (**myprofile.php**<sup>11</sup>, **myaccount.php**<sup>12</sup>).



Στο αρχείο **myaccount.php** ο χρήστης βλέπει τη φόρμα αλλαγής κωδικού και έχει την επιλογή να διαγράψει τον λογαριασμό του. Στη δεύτερη περίπτωση καλείται το αρχείο **account.php**<sup>13</sup>.



2) **village.php** ο παίκτης θα έχει τη δυνατότητα να δημιουργήσει τα υπερφυσικά όντα, τους κυνηγούς και τα κοράκια αλλά και να κάνει τις αναβαθμίσεις που θέλει.

Για την δημιουργία κάποιου χαρακτήρα καλείται το αρχείο **creation.php**<sup>14</sup> και για τις αναβαθμίσεις καλείται το αρχείο **upgrades.php**<sup>15</sup>

Hunter	Vampire	Werewolf	Hybrid	Witch	Crow
40 power	100 power	150 power	500 power	800 power	10 power
-1 villager -500 coins -150 wood	-1 villager -30 blood	-2 villagers	-2 werewolves -50 blood	-1 villager -1 vampire -1 werewolf -1000 coins	-50 coins -30 wood

Villager coin pr.	Villager population
+1coin per hour from each villager	+50 villagers
next level 3 -24000 coins	next level 1 -5000coins

3) **armory.php** περιέχονται τα όπλα που μπορεί να αγοράσει ένας παίκτης για τους κυνηγούς, αυτά που μπορεί να καταστρέψει και τα ξόρκια που μπορεί να χρησιμοποιήσει σε άλλους παίκτες ή και για δικό του όφελος.

Στη σελίδα αυτή όταν ο παίκτης συμπληρώσει τον αριθμό των όπλων που θέλει να αγοράσει καλείται το αρχείο **weapons.php**<sup>16</sup> και όταν ο παίκτης συμπληρώσει τον αριθμό των όπλων που θέλει να καταστρέψει καλείται το αρχείο **restore.php**<sup>17</sup>. Επίσης όταν πατηθεί το κουμπί για την χρήση ενός από τα ξόρκια καλείται



το αρχείο **spells.php**<sup>18</sup> όπου γίνεται έλεγχος αν ο παίκτης έχει ότι χρειάζεται για την χρησιμοποιήσει το ξόρκι και τότε αποθηκεύεται η ώρα που έγινε η χρήση του και υπολογίζεται η ώρα που το ξόρκι παύει να ισχύει.

Buy weapons:

Knife	Crossbow	Grenade	Pistol	Rifle	Flamethrower
10 power	50 power	60 power	70 power	110 power	170 power
-1.000 coins -500 wood	-3.000 coins -1.500 wood	-5.000 coins -2.500 wood	-8.000 coins -4.000 wood	-10.000 coins -5.000 wood	-50.000 coins -25.000 wood

Restore weapons:

Knife	Crossbow	Grenade	Pistol	Rifle	Flamethrower
+250 coins	+750 coins	+1.250 coins	+2.000 coins	+2.500 coins	+12.500 coins
restores knife -10 power	restores crossbow -50 power	restores grenade -60 power	restores pistol -70 power	restores rifle -110 power	restores flamethrower -170 power

4) **battles.php**<sup>19</sup> σε αυτή τη σελίδα υπάρχει η λίστα με όλους τους εγγεγραμμένους παίκτες και η μπάρα αναζήτησης παικτών (**search.php**<sup>20</sup>). Οι παίκτες εμφανίζονται με βάση την κατάταξη τους. Ο πρώτος παίκτης στη λίστα είναι και ο πρώτος στην κατάταξη και δίπλα από κάθε παίκτη αναγράφεται και ο αριθμός των χρημάτων που έχει εκτεθειμένα. Αν η δύναμεις μας είναι μεγαλύτερες από του αντιπάλου τότε έχουμε την δυνατότητα να δούμε τα χρήματα που έχει, αλλιώς δεν μας εμφανίζονται.

Ανακτήθηκε από:

<http://www.webreference.com/programming/php/search/2.html>

[http://php.about.com/od/phpwithmysql/ss/php\\_search.htm](http://php.about.com/od/phpwithmysql/ss/php_search.htm)

Rank	Player username	Coins
#2	katerina	0

5) **profile.php** εδώ μεταφέρεται ένας παίκτης όταν επιλέξει το όνομα ενός αντιπάλου του για να του επιτεθεί και εκεί βρίσκονται όλες οι ενέργειες αλληλεπίδρασης.

Ανακτήθηκε από:

<https://www.youtube.com/watch?v=IYmJeri6r0Y>

Στη συγκεκριμένη σελίδα ο κάθε παίκτης μπορεί να δει το όνομα του αντιπάλου του, τα χρήματα που έχει,

την ημερομηνία εγγραφής του στο παιχνίδι και την επιλογή να του στείλει μήνυμα (**message.php**<sup>21</sup>).

Αφού συμπληρωθεί η φόρμα καλείται το αρχείο **check.php**<sup>22</sup> όπου εδώ γίνεται έλεγχος αν έχει συμπληρωθεί η φόρμα του μηνύματος και στη συνέχεια στέλνεται το μήνυμα ή βγάζει κάποιο λάθος.

Επίσης περιέχονται όλες οι επιθετικές κινήσεις προς τον αντίπαλο παίκτη.

**spy.php:** γίνεται κατασκοπεία του αντιπάλου και επιλέγονται τα στοιχεία του από την βάση δεδομένων. Αν η ενέργεια είναι επιτυχής ο παίκτης βλέπει τα στοιχεία του αντιπάλου του.

**steal.php:** Στο αρχείο γίνεται έλεγχος των δυνάμεων 2 παικτών και αν ο επιτιθέμενος υπερτερεί, τότε κλέβει τα χρήματα από τον αντίπαλό του.

**huntHybrid.php:** γίνεται έλεγχος των δυνάμεων του επιτιθέμενου και του αμυνόμενου και αν η ενέργεια είναι επιτυχής ο επιτιθέμενος αποκτάει έναν κυνηγό από τον αντίπαλό του. Το ίδιο συμβαίνει και στα επόμενα αρχεία **huntVampire.php** και **huntWerewolf.php**.

**spellSpy.php:** εδώ επιλέγονται κάποια από τα στοιχεία του αντιπάλου που έχει επιλέξει ένας χρήστης και αν ο χρήστης έχει την δυνατότητα να χρησιμοποιήσει το ξόρκι τότε εμφανίζονται τα αποτελέσματα της κατασκοπείας.

Στα επόμενα αρχεία (**killCrow.php**, **killHybrid.php**, **killHunter.php**, **killVampire.php**, **killWerewolf.php**, **killWitch.php**) γίνεται πάλι ο ίδιος έλεγχος μόνο που αν η ενέργεια είναι επιτυχής ο αντίπαλος χάνει έναν χαρακτήρα. Επίσης μετά από κάθε επίθεση αλλάζουν και τα στατιστικά στοιχεία του κάθε παίκτη. Δηλαδή πόσες επιθέσεις έχει κάνει, πόσες έχει δεχτεί κτλ.

Όλες όμως οι σελίδες έχουν από κάποια κοινά στοιχεία όπου ο παίκτης τα βλέπει σε όποια σελίδα και να βρίσκεται. Αυτές είναι:

1) Η μπάρα που βρίσκεται κάτω από το μενού (**toptables.php**<sup>23</sup>) όπου εμφανίζεται η ημερομηνία και η ώρα, πότε τελειώνει η σεζόν του παιχνιδιού, το όνομα και η βαθμολογία του χρήστη, δύο σύνδεσμοι που αφορούν τον οδηγό του παιχνιδιού (**guide.php**<sup>24</sup>) και πληροφορίες σχετικά με παλαιότερες σεζόν του παιχνιδιού (**legends.php**) όπως είναι η κατάταξη των παικτών και τέλος του κουμπί αποσύνδεσης από το παιχνίδι (**logout.php**<sup>25</sup>).



2) Η δεξιά στήλη (**rightable.php**) όπου ο παίκτης βλέπει την ποσότητα των υλών που έχει, τον αριθμό από όλους τους χαρακτήρες του, τα όπλα που κατέχει και το επίπεδο που βρίσκεται για κάθε αναβάθμιση που υπάρχει.

Από τα τελευταία αρχεία που έφτιαξα ήταν:



- 1) **rank.php**<sup>26</sup> όπου επιλέγονται τα στοιχεία που χρειάζονται από την βάση δεδομένων για κάθε χρήστη χωριστά και υπολογίζεται η συνολική τους βαθμολογία (ΣR). Αφού γίνει αυτό καταχωρείτε ο συνολικός βαθμός του κάθε παίκτη στη βάση. Ύστερα ταξινομούνται οι παίκτες με βάση την υψηλότερη βαθμολογία και δίπλα τους ο αριθμός κατάταξής τους.
- 2) **season.php** γίνεται η επιλογή του ονόματος, των χαρακτήρων, της συνολικής δύναμης και των κερδισμένων μαχών ενός χρήστη και αποθηκεύονται στην βάση δεδομένων και κρατούνται ως αρχεία στο τέλος κάθε σεζόν του παιχνιδιού.
- 3) **resources.php**<sup>27</sup> σε αυτό το αρχείο γίνεται η αύξηση των υλών και των villagers για όλους τους χρήστες. Το αρχείο αυτό καλείτε και λειτουργεί κάθε μια ώρα.
- 4) **inactive.php**<sup>28</sup> στο αρχείο αυτό υπάρχει και σε όλα τα υπόλοιπα και λειτουργεί ως μετρητής σύνδεσης ορίου ενός χρήστη. Αν ο χρήστης είναι ανενεργός μετά από 20 λεπτά τότε αποσυνδέεται και επιστρέφει στην φόρμα σύνδεσης του παιχνιδιού.

Ανακτήθηκε από:

<http://stackoverflow.com/questions/520237/how-do-i-expire-a-php-session-after-30-minutes>

- 5) **seasontime.php**<sup>29</sup> εδώ υπάρχει ένα Script που μετράει αντίστροφα τον χρόνο που απομένει για να τελειώσει η σεζόν του παιχνιδιού και όταν ο χρόνος μηδενιστεί εμφανίζεται ένα μήνυμα σε ένα πλαίσιο που ενημερώνει τους παίκτες πως το παιχνίδι τελείωσε και τους οδηγεί πίσω στην σελίδα σύνδεσης του παιχνιδιού.

Ανακτήθηκε από:

[http://www.rmwebdesign.com/Countdown\\_Timers/Style\\_1\\_A.html](http://www.rmwebdesign.com/Countdown_Timers/Style_1_A.html)



αντίστροφη μέτρηση για το τέλος της σεζόν του παιχνιδιού:

### 3.4 Έλεγχος λαθών

Εφόσον έχουν ολοκληρωθεί τα παραπάνω βήματα, πρέπει να δούμε αν όλα λειτουργούν σωστά στο παιχνίδι. Η καλύτερη λύση για τον έλεγχο των λαθών είναι όταν το παιχνίδι ανέβει στο διαδίκτυο και μπορέσουν οι παίκτες να έχουν πρόσβαση. Κατά τη διάρκεια του παιχνιδιού είναι εύκολο κάποιος παίκτης να δει τα λάθη

που υπάρχουν και να μας ενημερώσει. Πριν όμως γίνει αυτό το δοκιμαστικό του παιχνιδιού (όπως γίνεται και σε όλα τα παιχνίδια που βγαίνουν) καλό είναι να έχουμε ελέγξει πρώτα εμείς όλο το παιχνίδι. Έτσι ξεκινάμε από την πρώτη σελίδα μέχρι την τελευταία ένα ένα ότι υπάρχει στο παιχνίδι. Από εντολές που πρέπει να γίνουν και αν εκτελούνται σωστά, από τους συνδέσμους αν μας μεταφέρουν εκεί που πρέπει καθώς και τα κουμπιά που υπάρχουν. Η διαδικασία αυτή είναι χρονοβόρα και μπορώ να πω αρκετά κουραστική αλλά και βαρετή μα είναι κάτι που πρέπει να γίνει και πρέπει να δώσουμε στους χρήστες το καλύτερο δυνατό αποτέλεσμα και όχι να δώσουμε κάτι μη λειτουργικό πράγμα που θα κουράσει και τους ίδιους τους παίκτες.

Θα ήθελα να αναφέρω εδώ πως ενώ το παιχνίδι μπορεί να λειτουργούσε χωρίς κανένα πρόβλημα όσο έτρεχε στον υπολογιστή μου και σε τοπικό δίκτυο, όταν το ανέβασα στο διαδίκτυο είχα να αντιμετωπίσω και άλλα λάθη που εμφανίστηκαν και σίγουρα μου πήρε επιπλέον χρόνο γιατί να μάθω πως λειτουργεί.

### 3.5 Προβλήματα που αντιμετώπισα

Ένα από τα μεγαλύτερα προβλήματα που αντιμετώπισα ήταν η σύνδεση του παιχνιδιού με τον πραγματικό χρόνο. Μια και το παιχνίδι όπως έχω αναφέρει είναι βασισμένο σε πραγματικό χρόνο, άρα και κάποιες από τις βασικές του λειτουργίες βασίζονται σε αυτόν. Για παράδειγμα οι ύλες που παράγει κάθε παίκτης αλλά και οι χωριάτες που αποκτάει έρχονται σε κάθε παίκτη με το πέρασμα του χρόνου. Για την ακρίβεια κάθε μία ώρα ο κάθε παίκτης πρέπει να αποκτάει συγκεκριμένη ποσότητα από κάθε ύλη που παράγει ύλες και να αποκτάει χωριάτες.

Αυτό ήταν ένα θέμα που με απασχόλησε από την αρχή και τελικά μέχρι και το τέλος μια και ήταν το τελευταίο πράγμα που ήθελα να κάνω για να ολοκληρωθεί το παιχνίδι.

Έκανα μια έρευνα πάνω στο συγκεκριμένο θέμα και βρήκα απαντήσεις που δεν κατάφεραν τελικά να επιλύσουν το πρόβλημά μου. Μια από αυτές ήταν η δημιουργία **event** στη βάση δεδομένων, όπου ορίζεις εσύ κάθε πότε πρέπει να τρέχει ένα event και τι εντολή ή εντολές πρέπει να εκτελούνται. Για το συγκεκριμένο δεν είχα τα δικαιώματα χρήσης μιας και δεν παραχωρούνται από την υπηρεσία σε ορισμένους πελάτες.

Άλλη μια ήταν η δημιουργία **phpjobscheduler** όπου μπορείς να επιλέξεις ένα αρχείο, στη δική μου περίπτωση το αρχείο `resources.php` (όπου περιέχει τις εντολές αύξησης υλών), και να ορίσεις κάθε πότε θα εκτελείτε. Μετά από προσπάθειες δεν λειτούργησε ούτε αυτό μιας και το αρχείο εκτελούσε τις εντολές ανάλογα με την επισκεψιμότητα στο παιχνίδι.

Εκτός από το παραπάνω υπήρξαν και άλλα προβλήματα που αντιμετώπισα όπως η διαδικασία επιλογής και τοποθέτησης των χρημάτων ενός παίκτη στο σεντούκι. Αρχικά όταν επέλεγα το ποσό των χρημάτων με την επιλογή του κουμπιού (`put coins`) και την εισαγωγή αυτών στο σεντούκι πάλι με ένα κουμπί (`into chest`), εμφανιζόταν στην οθόνη μου ένα μήνυμα λάθους. Όταν επέλεγα να πάω πίσω στην αρχική σελίδα, τα χρήματα που είχα επιλέξει υπήρχαν ακόμα στο πεδίο και με την δεύτερη εκτέλεση (πάτημα του κουμπιού `into chest`) τα χρήματα πήγαιναν κανονικά στο σεντούκι. Σε αυτή την περίπτωση εντόπισα πως το πρόβλημα

ήταν πως η διαδικασία επιλογής χρημάτων με την τοποθέτηση αυτών στο σεντούκι, “τρέχανε” στο ίδιο αρχείο, πράμα που μπλόκαρε την διαδικασία και έβγαζε λάθος. Έτσι έφτιαξα άλλο ένα αρχείο (**coins.php**) όπου εκεί γινόταν η τοποθέτηση των χρημάτων ενώ στην αρχική σελίδα γινόταν μόνο η επιλογή του ποσού των χρημάτων. Να προσθέσω πως και σε άλλα σημεία το αντιμετώπισα αυτό και χρειάστηκε να καλώ αρχεία μέσα σε άλλα αρχεία για να διαχωριστούν οι εντολές και να μην γίνονται όλες μαζί, που πολλές φορές εμφανιζόταν πρόβλημα.

Ένα πολύ σημαντικό πρόβλημα αλλά και εύκολο στην επίλυσή του είναι η επαναλαμβανόμενη εκτέλεση εντολών με την ανανέωση μιας σελίδας. Για παράδειγμα όταν έκανα μια αναβάθμιση και πάταγα το κουμπί ανανέωσης της σελίδας η αναβάθμιση συνέχιζε και σε επόμενα επίπεδα. Η αποτροπή αυτού για τις περισσότερες εντολές που υπάρχουν μέσα στο παιχνίδι γίνεται με την τοποθέτηση του παρακάτω κώδικα στο τέλος κάθε εντολής: `header ( 'Location: file_name.php' ) ;`

Έτσι στο τέλος εκτέλεσης κάθε εντολής ο παίκτης οδηγείτε στη σελίδα που ορίζουμε.

### 3.6 Τεχνολογίες που χρησιμοποίησα

Πρώτη και βασική προϋπόθεση είναι η ύπαρξη ενός web server. Κατόπιν ελέγχου που πραγματοποιήθηκε κατέληξα στο να χρησιμοποιήσω τον πιο διαδεδομένο και που χρησιμοποιούν οι περισσότεροι online servers, τον Apache server.

Εν συνεχεία, χρειαζόμαστε την υποστήριξη κάποιας γλώσσας προγραμματισμού, εγώ επέλεξα τη γλώσσα PHP μιας και αυτή είναι που έμαθα και από την σχολή και επίσης έγινε και χρήση της SQL για τα θέματα της βάσης δεδομένων. Το προγραμματιστικό κομμάτι της εργασίας έγινε με τη χρήση του eclipse, που επίσης η επιλογή του έγινε λόγω της γνώσης που είχα σε αυτό από την σχολή, ενώ για την βάση δεδομένων χρησιμοποίησα MySql καθώς είναι πιο διάσημο και πολυχρησιμοποιημένο για δυναμικές ιστοσελίδες.

### 3.7 Χρόνος για την ολοκλήρωση της πτυχιακής εργασίας

Ο χρόνος που χρειάστηκε για να ολοκληρωθεί αυτή η πτυχιακή εργασία ήταν αρκετός., μπορώ να πω γύρω στα 2 χρόνια συνολικά χωρίς όμως να είναι και η καθημερινή μου ασχολία. Αυτό έγινε κυρίως γιατί όταν ξεκίνησα για τη δημιουργία του παιχνιδιού υπήρχαν αρκετά πράγματα που δεν τα γνώριζα, οπότε μου πήρε κάποιο χρονικό διάστημα ακόμα και για τα πιο απλά και αναφέρομαι περισσότερο στο κομμάτι του προγραμματισμού.

Αρχικά χρειάστηκε χρόνος για τον απλό και γραπτό σχεδιασμό του παιχνιδιού όπου πολλές φορές άλλαξε. Μπορεί να είχα βρει πάνω κάτω τι χαρακτήρες θα έχει ο παίκτης και τι ενέργειες θα μπορεί να κάνει αλλά έπρεπε να υπολογίσω και τις τιμές για το κάθε ένα. Για παράδειγμα πόσα χρήματα θα παράγουν οι χωριάτες, πόσο θα κοστίζει ένα όπλο αλλά και οι αναβαθμίσεις. Πάνω σε αυτό το κομμάτι έπρεπε να δω αν όλες οι τιμές μεταξύ τους ταιριάζουν και κατά πόσο θα είναι εύκολο η δύσκολο ένας παίκτης να κάνει κάποιες

ενέργειες, από την δημιουργία των χαρακτήρων του, τα όπλα που θα αγοράζει και τις αναβαθμίσεις που θα κάνει. Αρκετός χρόνος επίσης χρειάστηκε και για την δημιουργία των επιθέσεων μεταξύ παικτών. Πόσες θα είναι, πώς θα γίνονται, τι δυνάμεις θα έχει αυτός που επιτίθεται και με τι δυνάμεις θα ανταποκρίνεται ο αμυνόμενος.

Πέρα από αυτό χρειάστηκε να ασχοληθώ και με το εικαστικό κομμάτι. Στην αρχή είχα κάνει μια απλή διάταξη για να με διευκολύνει αλλά μετά έπρεπε να αλλάξει καθώς σημαντικό ρόλο παίζει και το πως είναι στημένο ένα παιχνίδι και να είναι εύκολα κατανοητό για να προσελκύσει χρήστες. Αυτό όμως δεν ήταν και τόσο χρονοβόρο και σίγουρα ήταν από τα πιο απλά μέρη όλης αυτής της διαδικασίας.

Τον περισσότερο χρόνο όμως μου πήρε στο προγραμματιστικό κομμάτι αλλά και στη δημιουργία της βάσης δεδομένων καθώς πολλές φορές χρειάστηκε να εκτελώ εντολές μέσα στη βάση δεδομένων πράγμα που δεν το γνώριζα τόσο καλά. Μαζί με όλα αυτά και σαν τελευταίο μέρος ήταν η διόρθωση των λαθών που έβρισκα κατά την όλη διαδικασία αλλά και στη τελική διαδικασία αφού είχα ολοκληρώσει σχεδόν το παιχνίδι και έπρεπε να ανέβει στο διαδίκτυο, οπότε και εκεί ήρθαν νέα προβλήματα τα οποία δεν είχα να αντιμετωπίσω όταν το παιχνίδι έτρεχε στον υπολογιστή μου.



## ΣΥΜΠΕΡΑΣΜΑΤΑ

## 4.1 Κριτική

Η εργασία αυτή μπορεί να φανεί χρήσιμη σε όσους θέλουν να ασχοληθούν με τη δημιουργία ενός ηλεκτρονικού παιχνιδιού. Αποτελεί ένα βασικό οδηγό καθώς αναφέρει τα πρώτα βήματα της όλης διαδικασίας αλλά και τα προβλήματα που μπορεί να αντιμετωπίσει κανείς.

Τα διαδικτυακά παιχνίδια στην εποχή μας ποικίλουν και είναι ευρέως διαδεδομένα. Ακόμα και με τις βασικές γνώσεις στον προγραμματισμό μπορείς να δημιουργήσεις και εσύ ένα. Παρόλα αυτά, απαιτεί υπομονή και σίγουρα αγάπη γι αυτό που κάνεις.

Πλέον υπάρχουν έτοιμες φόρμες για τα διαδικτυακά παιχνίδια που μπορούν να μας διευκολύνουν για τα πρώτα βήματα μας αλλά και για την λειτουργικότητα του. Επίσης, έχει δημιουργηθεί μια μεγάλη και διαρκώς αναπτυσσόμενη κοινότητα γύρω από προγραμματιστικά προβλήματα που απαρτίζεται από απλούς χρήστες πρωτάρηδες αλλά και προγραμματιστές που μπορεί κανείς να συναντήσει και να βρει εκεί τις απαντήσεις που θέλει.

## 4.2 Μελέτη

Με την υλοποίηση αυτής της πτυχιακής εργασίας μου δόθηκε η ευκαιρία να αποκομίσω αρκετές γνώσεις κυρίως πάνω σε php αλλά και σε MySQL, CSS, HTML και λίγα πράγματα πάνω σε javascript. Σημαντική είναι και η εμπειρία που απέκτησα πάνω στο αντικείμενο της δημιουργίας ενός ηλεκτρονικού παιχνιδιού που θα με βοηθήσει στο μέλλον για την δημιουργία κάτι ακόμα καλύτερου και στην ευκολία προσέγγισης νέων κομματιών. Για την ολοκλήρωση όλης αυτής της διαδικασίας πραγματοποίησα μια έρευνα γύρω από το αντικείμενο της δημιουργίας ενός παιχνιδιού, παίζοντας άλλα παρόμοια παιχνίδια για να δω τον τρόπο λειτουργίας τους αλλά και να πάρω αρκετές ιδέες για το πως θα λειτουργεί το δικό μου παιχνίδι καθώς και πως μπορώ να το επεκτείνω.

Αφιέρωσα αρκετό χρόνο στην αναζήτηση της διόρθωσης λαθών πάνω σε αρκετά σημεία που συνάντησα και μέσω αυτού κατάφερα να μάθω ακόμα πιο πολλά που με βοήθησαν αργότερα για την λειτουργία του παιχνιδιού.

Επίσης με το ανέβασμα του παιχνιδιού στο διαδίκτυο βρήκα την ευκαιρία να ασχοληθώ με κάτι παραπάνω και να μάθω πάνω σε αυτό, καθώς υπάρχουν μεγάλες διαφορές που βρήκα και έπρεπε να αντιμετωπίσω σε σχέση με την λειτουργία του παιχνιδιού μου όταν το έτρεχε στον υπολογιστή μου.

### 4.3 Επέκταση

Όσον αφορά την επέκταση του παιχνιδιού θα μπορούσαν να προστεθούν επιπλέον χαρακτηριστικά και ενέργειες για τους παίκτες που να το κάνουν πιο ενδιαφέρον.

Για παράδειγμα να προστεθούν επιπλέον ύλες που θα παράγει ο παίκτης, περισσότερες επιλογές στα όπλα και στις αναβαθμίσεις που μπορεί να κάνει, αλλά και η ύπαρξη νέων αντικειμένων που θα προσθέτουν νέες δυνάμεις στον παίκτη.

Ακόμα μπορούν να προστεθούν και άλλες επιθετικές κινήσεις που μπορούν να γίνουν μεταξύ των παικτών, κάποιες πιο εύκολες και κάποιες πιο δύσκολες στην υλοποίησή τους.

Μια καλή ιδέα είναι και η επιλογή της δημιουργίας συμμαχιών. Πολλοί παίκτες θα μπορούν να είναι μαζί στην ίδια συμμαχία μέσα στην οποία δεν θα επιτρέπονται οι επιθετικές κινήσεις μεταξύ των μελών της και μαζί μπορούν να έχουν ένα κοινό στόχο και επιπλέον να υπάρχει και κατάταξη μεταξύ των συμμαχιών. Κάθε συμμαχία θα θέτει τους δικούς της όρους μέσα στο παιχνίδι και θα επιλέγει την δική της στρατηγική.

Όσον αναφορά επιπλέον λειτουργίες του παιχνιδιού, θα μπορούσε να προστεθεί και ένα chat room για καλύτερη και άμεση επικοινωνία μεταξύ των παικτών.



## ΒΙΒΛΙΟΓΡΑΦΙΑ

**Βιβλία:**

Jon Duckett (2008) *Beginning Web Programming with HTML, XHTML, and CSS* (second edition)  
εκδόσεις: Wiley Publishing, Inc

Timothy Boronczyk, Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec,  
Jeremy Stolz, Michael K. Glass (2009) *Beginning PHP 6, Apache, MySQL Web Development*  
εκδόσεις: Wiley Publishing, Inc

W. Jason Gilmore(2010) *Beginning PHP and MySQL from novice to professional* (4th.Edition)  
εκδόσεις: apress

**Πηγές στο διαδίκτυο:**

Stack Overflow - <http://stackoverflow.com>

Stack Overflow, How to expire a php session - <http://stackoverflow.com/questions/520237/how-do-i-expire-a-php-session-after-30-minutes>

Stack Overflow, Email verification - <http://stackoverflow.com/questions/14623923/verification-link-not-activating-account>

PHP - <http://www.php.net>

Mysql - <http://dev.mysql.com>

Tutorials point - <http://www.tutorialspoint.com/index.htm>

Tuts - <http://code.tutsplus.com/posts?page=2>

Email verification - <http://www.infotuts.com/email-verification-php/>

Youtube - <http://www.youtube.com>

Youtube, user profile page system - <https://www.youtube.com/watch?v=IYmJeri6r0Y>

About Tech - <http://www.about.com/compute>

How to create search engine - [http://php.about.com/od/phpwithmysql/ss/php\\_search.htm](http://php.about.com/od/phpwithmysql/ss/php_search.htm)

How to create search engine - <http://www.webreference.com/programming/php/search/2.html>

Countdown script creation - [http://www.rmkwebdesign.com/Countdown\\_Timers/Style\\_1\\_A.html](http://www.rmkwebdesign.com/Countdown_Timers/Style_1_A.html)

# 6

## ΠΑΡΑΡΤΗΜΑΤΑ

### 6.1 Πίνακας εικόνων

Table
<input type="checkbox"/> armory
<input type="checkbox"/> battles
<input type="checkbox"/> battle_stats
<input type="checkbox"/> end
<input type="checkbox"/> messages
<input type="checkbox"/> players
<input type="checkbox"/> resources
<input type="checkbox"/> seasons
<input type="checkbox"/> time
<input type="checkbox"/> total_power
<input type="checkbox"/> upgrades
<input type="checkbox"/> weapons
<b>12 tables</b>

Εικόνα 1 – database

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	<b>id</b>	int(11)			No	None	AUTO_INCREMENT
2	<b>username</b>	varchar(20)	utf8_bin		No	None	
3	<b>password</b>	varchar(20)	utf8_bin		No	None	
4	<b>email</b>	varchar(50)	utf8_bin		No	None	
5	<b>active</b>	int(11)			No	0	
6	<b>hash</b>	varchar(32)	utf8_bin		No	None	
7	<b>warriors</b>	int(11)			Yes	15	
8	<b>villagers</b>	int(5)			No	5	
9	<b>add_villagers</b>	time			No	None	
10	<b>act</b>	int(11)			No	0	
11	<b>time</b>	time			No	None	
12	<b>vampires</b>	int(5)			No	1	
13	<b>werewolves</b>	int(5)			No	1	
14	<b>hunters</b>	int(5)			No	1	
15	<b>hubrid</b>	int(3)			No	0	
16	<b>witches</b>	int(5)			No	1	
17	<b>crow</b>	int(5)			No	6	

Εικόνα 2 – players

Εικόνα 3 – time

#	Name	Type	Collation	Attributes	Null
1	<b>username</b>	varchar(20)	utf8_bin		No
2	<b>logindate</b>	date			No
3	<b>loginhour</b>	time			No

Εικόνα 4 – resources

#	Name	Type	Collation	Attributes	Null	Default
1	<b>username</b>	varchar(20)	utf8_bin		No	None
2	<b>chest</b>	int(11)			Yes	0
3	<b>coins</b>	int(11)			No	5000
4	<b>wood</b>	int(11)			No	1000
5	<b>blood</b>	int(11)			No	200
6	<b>coin_p</b>	int(11)			No	5
7	<b>wood_p</b>	int(11)			No	10
8	<b>blood_p</b>	int(11)			No	2
9	<b>coins_perhour</b>	int(11)			No	None
10	<b>wood_perhour</b>	int(11)			No	None
11	<b>blood_perhour</b>	int(11)			No	None
12	<b>last_autoplay</b>	datetime			No	None
13	<b>login</b>	datetime			No	None

Εικόνα 5 – armory

#	Name	Type	Collation	Attributes	Null	Default
1	username	varchar(20)	utf8_bin		No	None
2	time	time			No	None
3	spellFS	int(11)			No	0
4	spellFS_time	time			No	None
5	spellFS_expires	time			No	None
6	spellCP	int(11)			Yes	0
7	spellCP_time	time			No	None
8	spellCP_expires	time			No	None
9	spellUN	int(11)			Yes	0
10	spellUN_time	time			No	None
11	spellUN_expires	time			No	None

Εικόνα 6 – weapons

#	Name	Type	Collation	Attributes	Null	Default
1	username	varchar(20)	utf8_bin		No	None
2	weapons	int(11)			No	None
3	knife	int(11)			No	1
4	crossbow	int(11)			No	1
5	pistol	int(11)			No	None
6	grenade	int(11)			No	None
7	rifle	int(11)			No	None
8	flamethrower	int(11)			No	None

Εικόνα 7 – upgrades

#	Name	Type	Collation	Attributes	Null	Default
1	username	varchar(20)	utf8_bin		No	None
2	up_v_pop	int(11)			No	None
3	up_v_c	int(11)			No	None
4	up_wi_s	int(11)			No	None
5	up_va_b	int(11)			No	None
6	up_h_w	int(11)			No	None
7	up_h_wo	int(11)			No	None
8	up1	int(11)			No	3000
9	up2	int(11)			No	5000
10	up3	int(11)			No	1000
11	up4	int(11)			No	3000
12	up5	int(11)			No	3000
13	up6	int(11)			No	4000

Εικόνα 8 – battles\_stats

#	Name	Type	Collation	Attributes	Null	Default
1	id	int(11)			Yes	NULL
2	attacker	varchar(20)	utf8_bin		Yes	NULL
3	defender	varchar(20)	utf8_bin		Yes	NULL
4	action	varchar(20)	utf8_bin		Yes	NULL
5	date_time	datetime			No	None

Εικόνα 9 – battles

#	Name	Type	Collation
1	username	varchar(20)	utf8_bin
2	defender	varchar(20)	utf8_bin
3	battleA	int(11)	
4	battleW	int(11)	
5	battleL	int(11)	
6	battleAT	int(11)	
7	battleDE	int(11)	
8	win_coin	int(11)	
9	lost_coin	int(11)	
10	win_vampires	int(11)	
11	lost_vampires	int(11)	
12	win_werewolves	int(11)	
13	lost_werewolves	int(11)	
14	win_hubrids	int(11)	
15	lost_hubrids	int(11)	
16	hunter_kills	int(11)	
17	hunter_losses	int(11)	
18	vampire_kills	int(11)	
19	vampire_losses	int(11)	
20	werewolf_kills	int(11)	
21	werewolf_losses	int(11)	
22	hubrid_kills	int(11)	
23	hubrid_losses	int(11)	
24	witch_kills	int(11)	
25	witch_losses	int(11)	
26	crow_kills	int(11)	
27	crow_losses	int(11)	

Εικόνα 10 – total\_power

#	Name	Type	Collation	Attributes	Null	Default
1	username	varchar(20)	utf8_bin		No	None
2	tp_v	int(11)			No	50
3	tp_va	int(11)			No	100
4	tp_we	int(11)			No	150
5	tp_h	int(11)			No	40
6	tp_hub	int(11)			No	0
7	tp_wi	int(11)			No	800
8	tp_c	int(11)			No	20
9	totalpower	int(11)			No	None



Εικόνα 11 – messages

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	username	varchar(20)	latin1_swedish_ci		No	None	
2	message_id	int(11)			No	None	AUTO_INCREMENT
3	to_user	varchar(65)	latin1_general_ci		No	None	
4	message_title	varchar(65)	latin1_swedish_ci		No	None	
5	message_contents	longtext	latin1_swedish_ci		No	None	
6	message_read	int(11)			No	0	

Εικόνα 12 – end

#	Name	Type	Collation	Attributes	Null
1	username	varchar(20)	utf8_bin		No
2	rank	int(11)			Yes
3	R	int(11)			Yes
4	totalwarriorsrank	int(11)			Yes
5	totalpowerrank	int(11)			Yes
6	totalupgraderank	int(11)			Yes
7	battlesrank	int(11)			Yes

Εικόνα 13 – seasons

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	season	int(11)			No	None	AUTO_INCREMENT
2	rank	int(11)			No	1	
3	username	varchar(20)	utf8_bin		No	None	
4	total_warriors	int(11)			No	None	
5	total_power	int(11)			No	None	
6	battles_won	int(11)			No	None	

## 6.2 Παραπομπές κώδικα

### 1 index.php

```
<form action="http://blood-valley.com/connection.php" method="post">
<table class="login" align="center">
<tr><td>Username:</td><td>
<input style="float:right" type="text" name="user" maxlength="20" >
</td></tr>
<tr><td>Password:</td><td>
<input style="float:right" type="password" name="pass" maxlength="20" >
</td></tr>
<tr><td colspan="2" align="right">
<input type="submit" class="button" name="submit" value="Login">
</td></tr>
</table>
```

### 2 connection.php

```
<? php include 'connect.php' ;
session_start();
if (isset($_POST['submit']))
{
$username=$_POST['user'];
$password=$_POST['pass'];
if(!$username | !$password)
{
die('error message'); }
$check_password= mysql_query("SELECT password FROM players WHERE username
= '". $username. "' ")or die(mysql_error());
$check_password_r= mysql_num_rows($check_password);
$check_username = mysql_query("SELECT * FROM players WHERE username =
'". $username. "' ")or die(mysql_error());
$check_username_r = mysql_num_rows($check_username);
if ($check_username_r == 0)
{
die('error message') ; }
while($info = mysql_fetch_array( $check_password ))
{
$PASS = stripslashes($password);
$PASS2 = stripslashes($info['password']);
if ($PASS != $PASS2 )
{
die('error message') ;
}
}
else
{
$check_activation= mysql_query("SELECT active FROM players WHERE username
= '". $username. "' ")or die(mysql_error());
$check_activation_r= mysql_result($check_activation, 0);
if ($check_activation_r == 0)
{
die('error message'); }
if ($check_activation_r == 1)
{
$_POST['user'] = stripslashes($_POST['user']);
$hour = time() + 3600;
setcookie(ID_my_site, $_POST['user'], $hour);
setcookie(Key_my_site, $_POST['pass'], $hour);
$_SESSION['username']=$_POST['user'];
header("Location:http://blood-valley.com/main.php");
}}}?>
```

### 3 register.php

```
<?php if (isset($_POST['submit'])) {
if (!$_POST['user'] | !$_POST['pass'] | !$_POST['pass2'] |
!$_POST['email'])
{
die('error message'); }
$usercheck = $_POST['user'];
```

```

        $check = mysql_query("SELECT username FROM players WHERE username =
'$usercheck'")
        or die(mysql_error());
        $check2 = mysql_num_rows($check);
        if ($check2 != 0)
        {
            die('error message');
        }
        if ($_POST['pass'] != $_POST['pass2'])
        {
            die('error message');
        }
        if(!preg_match("/^[_\.0-9a-zA-Z-]+@[([0-9a-zA-Z][0-9a-zA-Z-]+\.)+[a-zA-
Z]{2,6}$/i", $_POST['email']))
        {
            die('error message');
        }
        $emailcheck = $_POST['email'];
        $check_email = mysql_query("SELECT email FROM players WHERE email =
'$emailcheck'")
        or die(mysql_error());
        $check_email_r = mysql_num_rows($check_email);
        if ($check_email_r != 0)
        {
            die('error message');
        }
        $hash=md5(rand(0,1000));
        $insert="INSERT INTO players (username, password, email, hash) VALUES(
        '". addslashes($_POST['user']) ."',
        '". addslashes($_POST['pass']) ."',
        '". addslashes($_POST['email']) ."',
        '". addslashes($hash) ."' ) " or die(mysql_error());
        $add_member = mysql_query($insert);
        $insert2="INSERT INTO battles (username) VALUES ('".$_POST['user']. "')";
        (συνεχίζει η εισαγωγή του username σε όλα τα tables)
        $email= $_POST['email'];
        $name=$_POST['user'];
        $password=$_POST['pass'];
        $insert="UPDATE time SET logindate= UTC_DATE() WHERE username='$name' ";
        $fdate=mysql_query($insert);
        $insert="UPDATE time SET loginhour=UTC_TIME() WHERE username='$name' ";
        $fhour=mysql_query($insert);
        $to
        = $email;
        $subject = 'mail subject';
        $message = ' Please click this link to activate your account:
http://www.blood-valley.com/verify.php?email='.\$email.'&hash='.\$hash.'';
        $headers = 'From: bloodvalley@outlook.com' . "\r\n";
        mail($to, $subject, $message, $headers);
        ?>

```

#### 4 verify.php

```

<?php if(isset($_GET['email']) && !empty($_GET['email']) AND
isset($_GET['hash']) && !empty($_GET['hash']))
{
    $email = mysql_escape_string($_GET['email']);
    $hash = mysql_escape_string($_GET['hash']);
    $search = mysql_query("SELECT email, hash, active FROM players WHERE
email='".$_email."' AND hash='".$_hash."' AND active='0'") or die(mysql_error());
    $match = mysql_num_rows($search);
    if($match > 0)
    {
        mysql_query("UPDATE players SET active='1' WHERE
email='".$_email."' AND hash='".$_hash."' AND active='0'") or die(mysql_error());
        echo 'error message';
    }
    else
    {
        echo 'error message'; } }
    else
    {
        echo 'error message'; }
    ?>

```

#### 5 process.php

```

<?php if (!empty($_POST['email'])) )
{
    $email= $_POST['email'];
    $email_check = mysql_query("SELECT * FROM players WHERE email = '".$_email."'
")or die(mysql_error());
}

```

```

$email_check_r = mysql_num_rows($email_check);
$query=mysql_query("SELECT password FROM players WHERE email='$email' " );
$password=mysql_result($query, 0);
if ($email_check_r ==0)
{
    die( 'error message'); }
else{
    $to      = $email;
    $subject = 'Blood Valley - your password';
    $message = ' Password: '.$password.' ';
    $headers = 'From:bloodvalley@outlook.com' . "\r\n";
    mail($to, $subject, $message, $headers); }}
else
{
    die( 'error message'); } ?>

```

## 6 main.php

```

<script type="text/javascript">
    function selectAll() {
        document.getElementsByName("selectAll")[0].value="<?php echo
$coins_result; ?>" ; } </script>
<a href="javascript:selectAll();">Put Coins</a>
<form action="coins.php" method="post">
    <input type="text" name="selectAll" size="11" value="0">
    <input type="submit" name="into chest" class="buttonCHEST" value="into
chest"/>
</form>

```

## 7 coins.php

```

<?php
$coins=mysql_query("SELECT coins FROM resources WHERE username='$name' ");
    $coins_r=mysql_result($coins, 0) ;
    if ($_POST['selectAll']>0 && $_POST['selectAll']<=$coins_r)
    { $coins_in_chest="UPDATE resources SET coins=$coins_r - $_POST[selectAll]
WHERE username = '$name'";
        $coins_in_chest_r=mysql_query($coins_in_chest);
    $chest=mysql_query("SELECT chest FROM resources WHERE username = '$name' " );
        $chest_r =mysql_result($chest, 0);
    $coins_in_chest="UPDATE resources SET chest= $chest_r + $_POST[selectAll] WHERE
username = '$name'";
        $coins_in_chest_r=mysql_query($coins_in_chest);
header('Location: main.php'); }
else
{
    header('Location: main.php'); } ?>

```

## 8 battlefieldstats.php

```

<?php
$query=mysql_query("SELECT defender, action, date_time FROM battle_stats WHERE
attacker='$name' ORDER BY date_time DESC");
$query2=mysql_query("SELECT attacker, action, date_time FROM battle_stats WHERE
defender='$name' ORDER BY date_time DESC");
echo '<tr><th>date time</th>';
echo '<th>attacker</th>';
echo '<th>action</th>';
echo '<th>defender</th></tr>';
echo '<tr><td colspan="4"><h5></h5> </td></tr>';
while($row=mysql_fetch_assoc($query) )
{
    echo '<tr><th>' . $row['date_time']. '</th>';
    echo '<th>' . $name. '</th>';
    echo '<th>' . $row['action']. '</th>';
    echo '<th>' . $row['defender']. '</th></tr>'; }
echo '<tr><td colspan="4"><h5></h5> </td></tr>';

```



```

while($row=mysql_fetch_assoc($query2) )
{
    echo '<tr><th>' . $row['date_time'] . '</th>';
echo '<th>' . $row['attacker'] . '</th>';
echo '<th>' . $row['action'] . '</th>';
echo '<th>' . $name . '</th></tr>'; } ?>

```

#### 9 indox.php

```

<?php $get_messages=mysql_query("SELECT message_id FROM messages WHERE
to_user='$name'
ORDER BY message_id DESC");
$get_messages2=mysql_query("SELECT * FROM messages WHERE to_user='$name'
ORDER BY message_id DESC");
$num_messages=mysql_num_rows($get_messages);
for($count=1; $count<=$num_messages; $count++)
{
    $row=mysql_fetch_array($get_messages2);
    if ($row['message_read']==0)
    {
        $_SESSION['message_id']=$row['message_id'];
        echo '<a href="mailto:mailnew.php?message_id=' . $row['message_id'] . '>' .
$row['message_title'] . '</a> <font color="red"> New </font>';
        echo $row['to_user'] . '<br>';
    }
    if ($row['message_read']==1)
    {
        $_SESSION['message_id']=$row['message_id'];
        echo '<a href="mailto:mailinbox.php?message_id=' . $row['message_id'] . '>' .
$row['message_title'] . '</a> ';
        echo $row['to_user'] . '<br>';
    } } ?>

```

#### 10 send.php

```

<?php $get_messages=mysql_query("SELECT message_id FROM messages WHERE
username='$name'
ORDER BY message_id DESC");
$get_messages2=mysql_query("SELECT * FROM messages WHERE username='$name'
ORDER BY message_id DESC");
$from_user=$name;
$num_messages=mysql_num_rows($get_messages);
for($count=1; $count<=$num_messages; $count++)
{ $row=mysql_fetch_array($get_messages2);
if($from_user==$name)
{
    $_SESSION['message_id']=$num_messages;
    echo '<a href="mailto:mailsend.php?message_id=' . $row['message_id'] . '>' .
$row['message_title'] . '</a> ';
    echo $row['to_user'] . '<br>'; } } ?>

```

#### 11 myprofile.php

ο χρήστης βλέπει το όνομα, την κατάταξη και την ημερομηνία εγγραφής του.

```

<?php $rank=mysql_query("SELECT rank FROM end WHERE username='$name'");
$rank_r=mysql_result($rank, 0);
echo '<font color="#800080"><big><b>' . $name . '</b></big></font>';
echo '<br><b>Rank: #' . $rank_r . '</b>';
$query=mysql_query("SELECT logindate FROM time WHERE username='$name' ");
$logindate=mysql_result($query, 0);
$query=mysql_query("SELECT loginhour FROM time WHERE username='$name' ");
$loginhour=mysql_result($query, 0);
echo 'Join at:<br>' . $logindate . '<br>' . $loginhour ; ?>

```

#### 12 myaccount.php

<h4>Change password</h4>

After this action you will be automatically logout from game, to login with your new password.<br>

<form action="" method="post">

```

Current password:


```

### 13 account.php

διαγράφονται όλα τα στοιχεία του παίκτη από τη βάση δεδομένων.

```

<?php session_start();
$name=$_SESSION['username'];
include'connect.php';
if (isset($_POST['changePASS']))
{
    $changePASS="UPDATE players SET password='$newpass' WHERE
username='$name' ";
    $changePASS_r=mysql_query($changePASS);
    header("Location:http://blood-valley.com/index.php");}
if (isset($_POST['delete']))){
$delete="DELETE FROM players WHERE username='$name'";
$delete_r=mysql_query($delete);
(συνεχίζεται η διαδικασία για όλα τα tables)
header("Location:http://blood-valley.com/index.php");}?>

```

### 14 creation.php

```

<?php if (isset($_POST['WEsubmit']) && ($_POST['WEbuy']==NULL))
{
    header('Location: village.php'); }
if (isset($_POST['WEsubmit']) && ($_POST['WEbuy']>0) ) {
    $villagers=mysql_query( "SELECT villagers FROM players WHERE username =
'$name' " );
    $villagers_r =mysql_result($villagers, 0);
    $webuy=$_POST['WEbuy'];
    if($villagers_r>='2'*$webuy)
    { $villagers_minus= " UPDATE players
        SET villagers= '$villagers_r' - '2'*$webuy
        WHERE username= '$name' ";

```

```

    $villagers_minus_r= mysql_query($villagers_minus);
    $n=mysql_query("SELECT werewolves FROM players WHERE username='$name'");
    $n_result=mysql_result($n,0);
    $insert = " UPDATE players
                SET werewolves=".$_POST['WEbuy']." + ".$n_result."
                WHERE username='$name' ";
    $add_werewolves = mysql_query($insert);
header('Location: village.php'); } ?>

```

### 15 upgrades.php

```

<?php if (isset($_POST['WSsubmit'])) {
$coins=mysql_query("SELECT coins FROM resources WHERE username='$name' ");
$coins_r=mysql_result($coins, 0);
$chest=mysql_query( "SELECT chest FROM resources WHERE username = '$name' " );
$chest_r =mysql_result($chest, 0);
$up_wi_s=mysql_query("SELECT up_wi_s FROM upgrades WHERE username='$name' ");
$up_wi_s_r=mysql_result($up_wi_s, 0);
$up6=mysql_query("SELECT up6 FROM upgrades WHERE username='$name' ");
$up6_r=mysql_result($up6, 0);
if( $coins_r>=$up6_r)
{
    for ($i=0; $i<=3; $i++)
        if ($up_wi_s_r==$i)
        { $coins_minus= " UPDATE resources
                    SET coins= '$coins_r' - '$up6_r'
                    WHERE username= '$name' ";
$coins_minus_r= mysql_query($coins_minus);
$level_up= " UPDATE upgrades
                SET up_wi_s= '$up_wi_s_r' + 1
                WHERE username= '$name' ";
$level_up_r= mysql_query($level_up); }
header('Location: village.php'); }
else if (isset($_POST['WSsubmit']) && $coins_r<$up6_r)
{
    for ($i=0; $i<=3; $i++)
        if ($up_wi_s_r==$i)
        { $minus=$up6_r-$coins_r;
$coins_minus= " UPDATE resources
                SET coins= '0'
                WHERE username= '$name' ";
$coins_minus_r= mysql_query($coins_minus);
$chest_minus= " UPDATE resources
                SET chest= '$chest_r'-'$minus'
                WHERE username= '$name' ";
$chest_minus_r= mysql_query($chest_minus);
$level_up= " UPDATE upgrades
                SET up_wi_s= '$up_wi_s_r' + 1
                WHERE username= '$name' ";
$level_up_r= mysql_query($level_up); }
header('Location: village.php'); }}?>

```

### 16 weapons.php

```

<?php if (isset($_POST['Ksubmit']) && ($_POST['Kbuy']==NULL))
{
    header('Location: armory.php');}
if (isset($_POST['Ksubmit']) && ($_POST['Kbuy']>0) ){
    $chest=mysql_query( "SELECT chest FROM resources WHERE username = '$name'
" );
    $chest_r =mysql_result($chest, 0);
    $coins = mysql_query( "SELECT coins FROM resources WHERE username =
'$name' " );
    $coins_r = mysql_result($coins , 0);
    $sum_coins_chest=mysql_query("SELECT SUM(coins+chest) FROM resources WHERE
username='$name'");
    $sum_coins_chest_r=mysql_result($sum_coins_chest, 0);
    $wood=mysql_query( "SELECT wood FROM resources WHERE username = '$name' " );
    $wood_r =mysql_result($wood, 0);

```

```

$kbuy=$_POST['Kbuy'];
if ($sum_coins_chest_r>='1000'*$kbuy && $wood_r>='500'*$kbuy)
{
    if($coins_r>='1000'*$kbuy)
    {
        $coins_minus= " UPDATE resources
        SET coins= '$coins_r' - '1000'*$kbuy
        WHERE username= '$name' ";
        $coins_minus_r= mysql_query($coins_minus);
        $wood_minus= " UPDATE resources
        SET wood= '$wood_r' - '500'*$kbuy
        WHERE username= '$name' ";
        $wood_minus_r= mysql_query($wood_minus);
        $n=mysql_query("SELECT knife FROM weapons WHERE
username='$name'");
        $n_result=mysql_result($n,0);
        $insert = " UPDATE weapons
        SET knife=".$_POST['Kbuy']." + ".$n_result."
        WHERE username='$name' ";
        $add_knife = mysql_query($insert); }
    else if($coins_r<'1000'*$kbuy)
    {
        $minus='1000'*$kbuy-$coins_r;
        $coins_minus= " UPDATE resources
        SET coins= '0'
        WHERE username= '$name' ";
        $coins_minus_r= mysql_query($coins_minus);
        $chest_minus= " UPDATE resources
        SET chest= '$chest_r'-'$minus'
        WHERE username= '$name' ";
        $chest_minus_r= mysql_query($chest_minus);
        $wood_minus= " UPDATE resources
        SET wood= '$wood_r' - '500'*$kbuy
        WHERE username= '$name' ";
        $wood_minus_r= mysql_query($wood_minus);
        $n=mysql_query("SELECT knife FROM weapons WHERE username='$name'");
        $n_result=mysql_result($n,0);
        $insert = " UPDATE weapons
        SET knife=".$_POST['Kbuy']." + ".$n_result."
        WHERE username='$name' ";
        $add_knife = mysql_query($insert); } }
header('Location: armory.php'); }?>

```

#### 17 restore.php

```

<?php if (isset($_POST['KNsubmit']) && ($_POST['Krestore']==NULL))
{
    header('Location: armory.php');}
$knife =mysql_query( "SELECT knife FROM weapons WHERE username = '$name' " );
$knife_n =mysql_result($knife, 0);
if (isset($_POST['KNsubmit']) && ($_POST['Krestore']>0) && ($knife_n>0) &&
($_POST['Krestore']<=$knife_n)){
    $krestore=$_POST['Krestore'];
    $coins = mysql_query( "SELECT coins FROM resources WHERE username =
'$name' " );
    $coins_r = mysql_result($coins , 0);
    $coins_plus= " UPDATE resources
        SET coins= '$coins_r' + '250'*$krestore
        WHERE username= '$name' ";
    $coins_plus_r= mysql_query($coins_plus);
    $weapon_minus= " UPDATE weapons
        SET knife=".$knife_n." - ".$_POST['Krestore']."
        WHERE username= '$name' ";
    $weapon_minus_r=mysql_query($weapon_minus);
header('Location: armory.php');} ?>

```



## 18 spells.php

```

<?php if (isset($_POST['UNsubmit']))
{
    $blood=mysql_query( "SELECT blood FROM resources WHERE username = '$name'
" );
    $blood_r =mysql_result($blood, 0);
    if($blood_r >= 250)
    {
        $query="UPDATE armory SET spellUN=1 WHERE username='$name'";
        $insert=mysql_query($query);
        $query="UPDATE armory SET spellUN_time=NOW() WHERE username='$name'";
        $insert=mysql_query($query);
        $query="UPDATE armory SET spellUN_expires=CURRENT_TIMESTAMP + INTERVAL '4'
        HOUR WHERE username='$name'";
        $insert=mysql_query($query);
        $blood_minus= " UPDATE resources
        SET blood= '$blood_r' - '250'
        WHERE username= '$name' ";
        $blood_minus_r= mysql_query($blood_minus);
    }
    header('Location: armory.php'); }?>

```

## 19 battles.php

```

<?php $total_records=mysql_query("SELECT * FROM players");
$total=mysql_num_rows($total_records);
$usernames = mysql_query("SELECT username,R FROM end ORDER BY R DESC");
$i=1;
while ($row = mysql_fetch_assoc($usernames))
{
    if ($row['username']==$name)
    {
        echo '<tr><th>';
        echo '#'. $i ;
        echo '</th>';
        echo '<th>';
        echo '<a
href="myprofile.php?username='.$row['username'].'">',$row['username'].'</a><br>'
;
        echo '</th></tr>';
    }
    else
    {
        echo '<tr><th>';
        echo '#'. $i ;
        echo '</th>';
        echo '<th>';
        echo '<a
href="profile.php?username='.$row['username'].'">',$row['username'].'</a>' ;
        echo '</th>';
        $coins=mysql_query( "SELECT coins FROM resources WHERE username =
'$row[username]' " );
        $coins_r =mysql_result($coins, 0);
        $crow = mysql_query( "SELECT crow FROM players WHERE username = '$name' "
);
        $crow_n = mysql_result($crow , 0);
        $tp_va= mysql_query( "SELECT tp_va FROM total_power WHERE username =
'$name' " );
        $tp_va_r =mysql_result($tp_va, 0);
        $view=$crow_n * $tp_va_r;
        $tp_wi_n_f =mysql_query( "SELECT tp_wi FROM total_power WHERE username =
'$row[username]' " );
        $tp_wi_f=mysql_result($tp_wi_n_f, 0);
        if ($view > $tp_wi_f)
        {
            echo '<th>';
            echo '<font color ="#FDD017">&nbsp;',$coins_r. '</font><br>';
            echo '</th></tr>';
        }
        else
        {
            echo '<th>';
            echo '<font color ="#FDD017">&nbsp;?</font><br>' ;
            echo '</th></tr>';
        }
    }
}

```

```

        $i++; }
mysql_free_result($usernames); ?>

```

## 20 search.php

Ο παίκτης συμπληρώνει ένα όνομα στην φόρμα αναζήτησης και γίνεται έλεγχος αυτού στη βάση δεδομένων. Αν το όνομα υπάρχει εμφανίζεται στη λίστα.

```

<?php if( isset($_POST['submit']) && $_POST['search']!=NULL )
{
    $total_records=mysql_query("SELECT * FROM players");
    $total=mysql_num_rows($total_records);
    $usernames = mysql_query("SELECT username,R,rank FROM end ORDER BY R
DESC");
    while ($row = mysql_fetch_assoc($usernames))
    {
        if($_POST['search']==$row['username'] && $row['username']!=$name )
        {
            $coins=mysql_query( "SELECT coins FROM resources WHERE
username = '$row[username]' " );
            $coins_r =mysql_result($coins, 0);
            echo '<tr><th>';
            echo '#'. $row['rank'] ;
            echo '</th>';
            echo '<th>';
            echo '<a
href="profile.php?username='.$row['username'].'">'. $row['username'].'</a>' ;
            echo '</th>';
            echo '<th>';
            echo '<font color="#FDD017">&nbsp;'. $coins_r. '</font><br>';
            echo '</th></tr>';
        }
        else if ($_POST['search']==$row['username'] &&
$row['username']==$name )
        {
            echo '<tr><th>';
            echo '#'. $row['rank'] ;
            echo '</th>';
            echo '<th>';
            echo '<a
href="myprofile.php?username='.$row['username'].'">'. $row['username'].'</a><br>'
;
        }
    }
    mysql_free_result($usernames); } ?>

```

## 21 message.php

εδώ υπάρχει η φόρμα όπου ο χρήστης συμπληρώνει τον τίτλο του μηνύματος και το περιεχόμενο.

```

<form name="message" action="check.php" method="post">
Title: <input type="text" name="message_title"> <br>
To:<input type="text" name="message_to" value="<?php
if (isset ($_POST['sendmessage'])) )
{
    echo $userprofile; }
?>"> <br> Message: <br>
<textarea rows="20" cols="50" name="message_content">
</textarea>
<?php echo '<input type="hidden" name="message_from" value="'. $name.'"><br>';
?>
<input type="submit" value="Send"></form>

```

## 22 check.php

```

<?php $select_all = mysql_query("SELECT * FROM messages");
$title=$_POST['message_title'];
$to=$_POST['message_to'];
$content=$_POST['message_content'];
$from=$_POST['message_from'];
$check_receiver=mysql_query("SELECT username FROM players WHERE
username='". $to."' ");
$check_receiver_r=mysql_num_rows($check_receiver);

```

```

if($check_receiver_r==0)
{   echo ("error message"); }
else if (strlen($content)<1)
{   echo ("error message"); }
else if(strlen($title)<1)
{   echo ("error message"); }
else {
    mysql_query("INSERT INTO messages (username, to_user, message_title,
message_contents)
                VALUES ('$from', '$to', '$title', '$content')")
    OR die ("Could not send the message: <br>".mysql_error());
    header('Location: send.php');    ?>
<form name="back" action="main.php" method="post">
<input type="submit" value="Back to main page">
</form><?php } ?>

```

### 23 toptables.php

εμφάνιση ημερομηνίας και ώρας.

```

<?php date_default_timezone_set('UTC');
echo date('l jS \of F Y');
echo date("H:i:s e"); ?>

```

### 24 guide.php

εμφανίζεται το pdf αρχείο στην οθόνη και υπάρχει η επιλογή να το κατεβάσει ένας χρήστης.

```

<object data="gameguide.pdf" type="application/pdf" width="100%" height="100%">
<a href="gameguide.pdf"> click here to download the game guide [pdf file].</a>
</object>

```

### 25 logout.php

με το πάτημα του κουμπιού αποσύνδεσης σταματάει να ισχύει το session και ο χρήστης αποσυνδέεται και επιστρέφει στην αρχική σελίδα σύνδεσης.

```

<?php if (isset($_POST['logout']))
{ session_start();
unset($_SESSION);
session_destroy();
session_write_close();
header ('Location: index.php');
die; } ?>

```

### 26 rank.php

```

<?php

```

```

$R= $totalwarriorsrank + $totalpowerrank + $totalupgraderank + $battlesrank;
$query="UPDATE end SET R='.$R.' WHERE username='$name' ";
$insert=mysql_query($query);
$rank=mysql_query("SELECT username,R FROM end ORDER BY R DESC");
$r=1;
while ($R = mysql_fetch_assoc($rank))
{
    if ($name==$R['username'])
    {
        echo '#'. $r ;
        $query="UPDATE end SET rank='.$r.' WHERE username='$name' ";
        $query2=mysql_query($query); }
    $r++; } ?>

```

### 27 resources.php

```

<?php $query=mysql_query("SELECT * FROM resources");
$total_players=mysql_num_rows($query);
$player_resources=mysql_query("SELECT username, blood, blood_perhour, wood,
wood_perhour, coins, coins_perhour FROM resources");

```

```

$player_villagers=mysql_query("SELECT username, villagers FROM players");
for ($i=0; $i<$total_players; $i++)
{
    while ($row=mysql_fetch_assoc($player_resources))
    {
        $query="UPDATE resources SET coins='$row[coins]' + '$row[coins_perhour]'
WHERE username='$row[username]' ";
        $add_coins=mysql_query($query);
        $query="UPDATE resources SET blood='$row[blood]' + '$row[blood_perhour]' WHERE
username='$row[username]' ";
        $add_blood=mysql_query($query);
        $query="UPDATE resources SET wood='$row[wood]' + '$row[wood_perhour]' WHERE
username='$row[username]' ";
        $add_wood=mysql_query($query); }
    while ($row2=mysql_fetch_assoc($player_villagers))
    {
        $query="UPDATE players SET villagers='$row2[villagers]' + 1 WHERE
username='$row2[username]' ";
        $add_villagers=mysql_query($query);
    } } ?>

```

#### 28 inactive.php

```

<?php session_start();
$session_timeout = 1200;
if (!isset($_SESSION['last_visit']))
{ $_SESSION['last_visit'] = time(); }
if((time() - $_SESSION['last_visit']) > $session_timeout)
{
    session_destroy();
    header("Location: index.php");
    exit; }
$_SESSION['last_visit'] = time(); ?>

```

#### 29 seasontime.php

```

<script type="text/javascript">
    if(dday<=0&&dhour<=0&&dmin<=0&&dsec<=0)
    {
        alert('Season Has Ended! See you next season') ;
    }
window.location = 'index.php';
</script>

```