

- ◆ Θεοδόσης Παιδάκης A.M.:
1115201500118
- ◆ Δημήτρης Παπαχρήστου A.M.:
1115201500124

- ◆ Εντολές μεταγλώττισης και εκτέλεσης:

\$ g++ main.cpp

Implementation.cpp Interface.h

\$./a.out

- ◆ Σειρά υλοποίησης του κώδικά μας:

1) Υλοποίηση των κλάσεων που περιγράφει η εκφώνηση με τις εξής ιεραρχίες:

- Item :
 - Weapon

- Armor
- StrengthPotion
- DexterityPotion
- AgilityPotion
- Spell:
 - IceSpell
 - FireSpell
 - LightingSpell
- Living:
 - **Hero:**
 - Warrior
 - Sorcerer
 - Paladin
 - **Monster:**
 - Dragon
 - Exoskeleton
 - Spirit
- Grid:

- 2) Αρχικά, δημιουργήσαμε το μενού, και κάθε επιλογή που προσθέταμε σε αυτό, την υλοποιούσαμε στον κώδικα μας. Οι επιλογές όλων των μενού του παιχνιδιού, γίνονται επιλέγοντας ο παίχτης το κατάλληλο νούμερο.
- 3) Στην αρχή του παιχνιδιού, ο παίχτης δημιουργεί 1-3 ήρωες, χωρίς την δυνατότητα να δημιουργήσει παραπάνω ήρωες στην συνεχεια του παιχνιδιου. Ο κάθε τύπος ήρωα (Warrior,..κλπ), αρχικοποιείται με συγκεκριμένες προκαθορισμένες τιμές.

4) Οι θέσεις του χάρτη αρχικά δημιουργούνταν τυχαία, αλλά αυτό παρουσίαζε προβλήματα στο παιχνίδι, μιας και δεν εξασφαλιζόταν ότι δεν θα είχαμε κλειστό μονοπάτι. Οπότε, ο χάρτης πλέον είναι σταθερός και προκαθορισμένος από εμάς. Ο τύπος της κάθε θέσης (`enum square_type`) , αρχικοποιείται από το αρχείο `map.txt` . Οι ήρωες πάντα στην αρχή του παιχνιδιού, θα ξεκινούν από την θέση `[1 , 1]` .

5) Έπειτα, δημιουργήσαμε το `market`. Για κάθε `Weapon`, `Armor`, `Potions` και `Spells`, δημιουργήσαμε `structures` τα

οποία περιέχουν τα
χαρακτηριστικά που έχει το
κάθε Item ή Spell.

Έχουμε ακόμα μία λίστα για
κάθε ένα απο αυτά τα Items
και Spells, η οποία αποτελείται
απο Nodes του αντίστοιχου
structure.

Οι τιμές αυτών δίνονται από
τα αντίστοιχα αρχεία .txt .

6)