

# Προσχέδιο Υλοποίησης Λογισμικού

Δημήτρης Παύλου και Γιάννης Κούβελας

Χειμερινό Εξάμηνο 2022

## Στοιχεία Των Μελών Της Ομάδας

- Δημήτρης Παύλου, 1115202100151, 2o Έτος, sdi2100151@di.uoa.gr
- Γιάννης Κούβελας, 1115202100070, 2o Έτος, sdi2100070@di.uoa.gr

## Αναθέσεις Ομάδος

### Δημήτρης Παύλου)

Ο Δημήτρης Παύλου ασχολήθηκε με τις κλάσεις GameState και Map. Όρισε κατάλληλες μεθόδους για την διαχείρηση του παιχνιδιού και του χάρτη (Αρχεία Map.cpp, Map.hpp, GameState.hpp, GameState.cpp).

Ακόμη όρισε βοηθητικές κλάσεις όπως την Clock (Που αντιπροσωπεύει το ρολόϊ του παιχνιδιού), την Size (Που αντιπροσωπεύει το μέγεθος του χάρτη) και την Point2D (Η οποία αντιπροσωπεύει ένα σημείο στον δισδιάστατο χώρο). (Αρχεία Clock.hpp, Clock.cpp, Point2D.hpp, Point2D.cpp, Size.hpp, Size.cpp)

Τέλος όρισε την μέθοδο UpdateEntity σε κάθε μια από τις οντότητες (Avatar, Werewolf, Vampire), ανέλαβε το μοντάζ του βίντεο και το documentation.

## Γιάννης Κούβελας)

Ο Γιάννης Κούβελας ασχολήθηκε με τις κλάσεις Avatar, Potion, WereWolf, Vampire, GameObject, Tree, Ground, River. Δηλώσε τα μέλη των κλάσεων, όρισε Set / Get μεθόδους για την διαχείρηση των δεδομένων των κλάσεων αλλά και τις virtual μεθόδους GetGameObjectType, DrawGameObject. (Αρχεία Avatar.hpp, Avatar.cpp, Vampire.hpp, Vampire.cpp, WereWolf.hpp, WereWolf.cpp, Ground.hpp, Ground.cpp, River.hpp, River.cpp κτλ.)

Ακόμη όρισε την κλάση Input η οποία είναι υπεύθυνη για την επικοινωνία παίκτη και παιχνιδιού. (Αρχεία Input.cpp, Input.hpp)

## Περιγραφή Του Προγράμματος

Η GameState είναι η κλάση η οποία αντιπροσωπεύει την κατάσταση του παιχνιδιού. Περιέχει έναν χάρτη (Map), το μέγεθος του χάρτη (Size), το ρολόϊ του παιχνιδιού καθώς και μια λογική μεταβλητή που είναι αληθής όταν το παιχνίδι είναι σταματημένο.

Μέσα στην GameState υπάρχουν οι μέθοδοι UpdateGameState και η DrawGameState οι οποίες καλούνται επαναληπτικά στην main μέχρι το παιχνίδι να τελειώσει. Η UpdateGameState καλεί εσωτερικά την UpdateMap μέθοδο του χάρτη στην οποία εσωτερικά καλείται η μέθοδος UpdateEntity σε κάθε οντότητα του χάρτη. Η DrawGameState καλεί εσωτερικά της είται την DrawMap είτε την DrawGameStateInfo ανάλογα με το εαν το παιχνίδι είναι σταματημένο.

Για την υλοποίηση του χάρτη του παιχνιδιού γίνεται χρήση ενός vector που περιέχει δείκτες σε GameObjects. GameObject θεωρούνται τα πάντα (έδαφος, δέντρο, νερό, άβαταρ, βαμπίρ κτλ.). Στην αρχή του παιχνιδιού ο χαρτής γεμίζεται τυχαία από έδαφος, νερό και δέντρα. Ο avatar, οι vampires και οι WereWolves εμφανίζονται κάθε φορά στις ίδιες θέσεις ώστε το παιχνίδι να είναι πιο δίκαιο.

Σε κάθε ενημέρωση του χάρτη γίνεται έλεγχος του input από τον παίκτη και μόνο αν το input είναι ορθό ενημερώνεται ο avatar. Για τους vampires και werewolves γίνεται επίσης έλεγχος των νόμιμων κινήσεων ώστε να πάζουν μόνο νόμιμες κινήσεις οι οποίες επιλέγονται τυχεία μέσω της rand().

Αφότου ενημερωθούν όλα τα αντικείμενα του χάρτη τότε εκτυπώνεται ο χάρτης στην οθόνη αν η κατάσταση του παιχνιδιού είναι ενεργή. Διαφορετικά το παιχνίδι παραμένει ανενεργό μέχρι να δωθεί η εντολή pause από τον παίκτη.

## Παραδοχές

- Το μέγεθος του χάρτη πρέπει να είναι μεταξύ του 3 και 15 και δεν γίνεται να είναι άρτιος αριθμός. Ακόμη ο χάρτης είναι τετράγωνος.
- Στο παιχνίδι δεν υπάρχει σειρά. Οι οντότητες παίζουν ανάλογα με την σειρά εισαγωγής τους στο vector.
- Για να κάνει pick up ο avatar το potion αρχεί να βρίσκεται δίπλα του.

## IDE / Compiler / System

Για την συγγραφή των προγραμμάτων έγινε χρήση του vscode σε wsl (Ubuntu) και για το compile χρησιμοποιήσαμε τον g++.

## Δυσκολία

Η εργασία από πλευράς κώδικα ήταν σχετικά εύκολη, αλλά η εκφώνηση και η πολύ ελευθερία την έκαναν σχετικά δύσκολη από πλευρά υλοποίησης.

## Github

[https://github.com/DimitriosPavlouGR/oop\\_ergasia](https://github.com/DimitriosPavlouGR/oop_ergasia)

## Youtube

<https://youtu.be/4XYlfrJkp7Q>