Α. Τα μέλη της ομάδας μας:

Μέλος Α:

Όνομα: Χριστίνα

Επίθετο: Μήτσιου

Α.Μ.: 1115202000129

E-mail: sdi2000129@di.uoa.gr

Έτος φοίτησης: 3ο

Μέλος Β:

Όνομα: Ορφέας

Επίθετο: Ηλιάδης-Σιβρής

Α.Μ.: 1115202000057

E-mail: sdi2000057@di.uoa.gr

Έτος φοίτησης: 3ο

Β. Με ποιά τμήματα της εργασίας ασχολήθηκε το κάθε μέλος:

Τα πρωτότυπα της υλοποίησης των κλάσεων των:

- Map
- Point
- Obstacle (Δεν υπάρχει στο τελικό project -> ενσωματώθηκε στο object)

Ανέλαβε ο Ορφέας Ηλιάδης-Σιβρής.

Τα πρωτότυπα της υλοποίησης των κλάσεων των:

- Entity
- Player
- Werewolf
- Vampire

Ανέλαβε η Χριστίνα Μήτσιου.

Η περαιτέρω υλοποίηση, βελτίωση, τροποποίηση και η όποια προσθήκη κλάσεων (λ.χ. gamestate) η λοιπών λειτουργιών στον υπάρχοντα κώδικα, έγινε από κοινού με δια ζώσης συνεργασία των μελών, επομένως δεν αποδίδονται εύσημα σε συγκεκριμένο μέλος για συγκεκριμένο τμήμα κώδικα.

- Οργάνωση και σχολιασμός του κώδικα κατά κύριο λόγο από τη Χριστίνα Μήτσιου.

- Debugging και optimization του κώδικα κατά κύριο λόγο από τον Ορφέα Ηλιάδη.

C. Περιγραφή του κώδικα:

Οι κλάσεις που χρησιμοποιούνται είναι οι εξής : Point , Object, Entity(base class), Player(derived από την Entity)

Vampire (derived από την Entity), Werewolf(derived από την Entity), Map και GameState.

- Point: Χρησιμοποιείται για να περιγράψει την θέση μιας οντότητας η ενός αντικειμένου στο δισδιάστατο χώρο.

- Entity: Υπερκλάση χρησιμοποιείται για τα κοινά χαρακτηριστικά των κλάσεων Player, Werewolf και Vampire

δηλαδή την θέση τους καθώς και τον τύπο τους.

- Player: Έχει όλα τα χαρακτηριστικά και τις μεθόδους του παίκτη.

- Werewolf: Έχει όλα τα χαρακτηριστικά και τις μεθόδους των λυκανθρώπων.

- Vampire: Έχει όλα τα χαρακτηριστικά και τις μεθόδους των βαμπίρ.

- Map: Περιέχει όλα τα αντικείμενα και τις οντότητες. Πραγματοποιεί και συντονίζει τις αλληλεπιδράσεις

μεταξύ τους και εμφανίζει τον χάρτη του παιχνιδιού μαζί όλα τα χαρακτηριστικά του.

- GameState: Αναλαμβάνει την ενημέρωση του παιχνιδιού με άμεση αλληλεπίδραση με τον χάρτη

αλλά και με τον χρήστη λαμβάνοντας το input του από το πληκτρολόγιο και μεταφέροντας το στο map.

- Main function: Δέχεται τα απαραίτητα δεδομένα από τον χρήστη, αρχικοποιεί τον παίκτη, τον χάρτη και την κατάσταση του παιχνιδιού και ξεκινάει την βασική λούπα του παιχνιδιού που πραγματοποιείται μέχρι να τελειώσει το παιχνίδι.

D. Παραδοχές που πραγματοποιήσαμε για την υλοποίηση του παιχνιδιού:

I) Το πλήθος των οντοτήτων είναι μικρότερο από το ζητούμενο διότι στους περισσότερους συνδυασμούς διαστάσεων υπάρχει υπερβολικά μεγάλος αριθμός από werewolves και vampires στο χάρτη.

II) Στην μέθοδο attack_enemy των Vampires και των Werewolves στην ειδική περίπτωση που το επίπεδο άμυνας του αμυνόμενου είναι μεγαλύτερο από το επίπεδο επίθεσης του επιτιθέμενου, το επίπεδο υγείας του αμυνόμενου παραμένει αμετάβλητο (από την εκφώνηση δεν καθίσταται σαφές ότι αυτό θα έπρεπε να ισχύει).

III) Η κίνηση του παίκτη γίνεται με τα πλήκτρα W,A,S και D αντί για τα βελάκια. Επιπλέον για να χρησιμοποιήσει το πρόγραμμα το input πρέπει να πατηθεί το πλήκτρο "Enter" (Χρησιμοποιείται η scanf της cstdio για το input διότι οι συναρτήσεις της βιβλιοθήκης windows δεν μπορούσαν να χρησιμοποιηθούν στο WSL περιβάλλον στο οποίο εργαζόμασταν βλ. f.).

IV) Σε κάθε frame εκτυπώνεται κάτω από το map η μέση ζωή των δυο ομάδων ώστε ο παίκτης να γνωρίζει πότε είναι σκόπιμο να χρησιμοποιήσει τα healing potions.

V) Χρησιμοποιήθηκαν ορισμένες Εντολές για τον χρωματισμό του τερματικού με σκοπό να προσομοιώνει ένα γραφικό interface καθώς και ορισμένοι ειδικοί χαρακτήρες για να διαφοροποιούν τις οντότητες μέσα στον χάρτη. Οι πήγες από τις οποίες πήραμε τις εντολές και τους χαρακτήρες αυτούς υπάρχουν στους παρακάτω υπερσυνδέσμους:

Χρωματισμός τερματικού: https://www.codeproject.com/Articles/5329247/How-to-change-text-color-in-a-Linux-terminal

Ειδικοί χαρακτήρες: https://www.altcodes.pro/

E. IDE, compiler, λειτουργικό σύστημα:

Και τα δύο μέλη της ομάδας δουλεύαμε σε Windows 11 υπολογιστές αλλά σε περιβάλλον linux (Ubuntu 20.04 το ένα και Ubuntu 22.04) μέσω WSL. Ο IDE που χρησιμοποιήθηκε είναι το Visual Studio Code στην τελευταία του έκδοση. Τέλος ο compiler που χρησιμοποιήθηκε είναι ο g++.

F. Προβλήματα που αντιμετωπίσαμε:

- Πέρα από μικρά σχεδιαστικά σφάλματα και ελαφρώς προβληματικές υλοποιήσεις, το βασικό πρόβλημα που αντιμετωπίσαμε ήταν η δυσκολία να βρούμε κάποια ενναλακτική μέθοδο για να διαβάζουμε το input από το πληκτρολόγιο κατά την ώρα που ο χρήστης παίζει το παιχνίδι.

Αυτό οφείλεται στο ότι επιλέξαμε να δουλέψουμε σε WSL περιβάλλον καθώς είμαστε περισσότερο εξοικειωμένοι με αυτό από παρελθοντικές εργασίες και επιπλέον αντιμετωπίζαμε δυσκολίες με τον g++ στο περιβάλλον των Windows.

Ωστόσο αυτό δημιούργησε το πρόβλημα ότι δεν μπορούσαμε να χρησιμοποιήσουμε τις συναρτήσεις GetKeyState() και GetAsyncKeyState() της βιβλιοθήκης windows.h.

Έπειτα από μεγάλη προσπάθεια να βρούμε κάποια εξίσου καλή ενναλακτική βιβλιοθήκη ή συνάρτηση για να λαμβάνουμε το input από το πληκτρολόγιο, αποφασίσαμε να χρησιμοποιήσουμε μια υβριδική εκτέλεση η οποία συνδυάζει την χρήση της scanf από την βιβλιοθήκη cstdio της C και την συνάρτηση fcntl της βιβλιοθήκης fcntl.h για να κάνουμε την scanf non-blocking. Η πηγή από την οποία εμπνευστήκαμε για να πετύχουμε το παραπάνω είναι ο παρακάτω υπερσύνδεσμος:

https://stackoverflow.com/questions/58110168/read-input-from-console-without-waiting-unconditionallynon-waiting-scanf

Το τελικό αποτέλεσμα λειτουργεί με τον εξής τρόπο:

- Όταν δεν πατιέται κανένα πλήκτρο το παιχνίδι κάνει update από μόνο του

- Όταν πατηθεί κάποιο πλήκτρο πρέπει έπειτα να πατηθεί το "Enter" έτσι ώστε το input να ληφθεί από το πρόγραμμα και

να γίνουν οι απαραίτητες διεργασίες.

G. Η εργασία είναι πλήρης, υλοποιήθηκαν όλα τα ζητούμενα.

Η. Βαθμός δυσκολίας της εργασίας

Τα ζητούμενα τις εργασίας αυτά κάθε αυτά είναι κάποιας μέτριας δυσκολίας, ωστόσο η διαδικασία συγγραφής των παραδοτέων, η ασάφεια ορισμένων ζητουμένων και η έλλειψη βοηθητικού υλικού για την εργασία (έτοιμα tests, βιβλιοθήκες που θα μπορούσαν να χρησιμοποιηθούν, κάποιο template αρχείο ή κάποια ενδεικτική υλοποίηση του εκτελέσιμου προγράμματος) μας οδηγήσαν στο να αφιερώσουμε παραπάνω χρόνο σε επίλυση τεχνικών προβλημάτων (όπως τα προαναφερθέντα βλ. F) και λιγότερο σε ζητήματα αντικειμενοστραφούς προγραμματισμού. Αυτό φαίνεται και από το γεγονός ότι ξεφύγαμε κατά σημαντικό χρονικό διάστημα από τους αρχικούς μας υπολογισμούς (βλ. προσχέδιο υλοποίησης λογισμικού).

Github Repository: https://github.com/christinamitsiou/WnV_the_game

(Ο φάκελος όπου περιέχεται ο ολοκληρωμένος κώδικας είναι ο Final)

J: Youtube URL: https://www.youtube.com/watch?v=4Dnzq6mGy24

K: Επιπρόσθετες παρατηρήσεις:

- Τα αντικείμενα αρχεία παράγονται με την εντολή:

```
g++ -c main.cpp point.cpp entities.cpp object.cpp map.cpp gamestate.cpp
```

- Το εκτελέσιμο αρχείο παράγεται με την εντολή:

```
g++ -o game main.o point.o entities.o object.o map.o gamestate.o
```

- Το παιχνίδι τρέχει με την εντολή

```
.//game
```

Όλες οι παραπάνω εντολές πρέπει να εκτελεστούν στον τερματικό του directory Solution