Τελική αναφορά εργασίας

α) Μέλη ομάδας :
Μάριος Κινδύνης Α.Μ 1115202000223 έτος : 3ο Email sdi2000223@di.uoa.gr
Κωνσταντίνος Ματθαίου Α.Μ 1115202000225 έτος : 3ο Email: sdi2000225@di.uoa.gr
git hub link https://github.com/M-Kindynis25/Ergasia_AntikeimenostrafisProgrammatismos.git
youtube link https://youtu.be/i0H7gFfZZSw
β) Στην εργασία όπως αναφέραμε και στο 2ο παραδοταίο αρχείο συμφωνήθηκε οτι και τα δύο μέλη της ομάδας θα ασχοληθόυν με όλη την εργασία ταυτόχρονα για να μειώσουμε οσο το δυνατο πιο πολύ τα λάθη και να παρέχονται αάμεσες ιδέες σε περίπτωση που ένας από τους δύο κολλήσει κάπου. Κατά τη διάρκεια του προγραμματισμού και οι δύο παρείχαμε κώδικα ο ένας στον αλλο, τον βελτιστοποιούσαμε και τον προσθέταμε στο κύριο πρότζεκτ αφού είχαμε βεβαιωθεί οτι είναι σωστά δομημένος και τρέχει χωρίς λάθη. Σε όλη τη διάρκεια του πρότζεκτ είμασταν μαζί σε βιντεοκλήση παρακολουθόντας προσεκτηκά το τι έκανε ο καθένας μέσω shared screen .
γ) Αρχικά δημιουργήσαμε το header file entities.h όπου τοποθετήσαμε όλες τις κλάσεις μας :
class entitties :

Θέσαμε public όλες τις συναρτήσεις που καταχωρίσαμε σε αυτή τη κλάση η οποία είναι και η μητρική γιατί θα χησιμοποιούνται από τις υποκλάσεις και θέσαμε και όλες τις συναρτήσεις σαν virtual για να μπορούν να χρισημοποιηθούν από όλες τις υποκλάσεις. Επίσης βάλαμε σαν protected τις συντεταγμένες coordinates για να έχουν πρόσβαση σε αυτές οι υποκλάσεις.

Στη συνέχεια ορίσαμε την κλάση avatar με κληρονομηκότητα την μητρική κλάση entities ορίζοντας μέσα στην avatar public την entities. Στη συνέχεια ορίσαμε τον constructor του avatar με ορίσματα την ομάδα του , την στνεταγμένη χ και την συντεταγμένη ψ.
Μετά δημιουργήσαμε την συνάρτηση get_coordinates και την set_coordinates όπου λειτουργούν σαν accesor του coordinates.
Μετά γράψαμε την συνάρτηση move με 8 μεταβλητές που αντηστηχούν στις 8 πιυανές κινήσεις που μπορούν να γίνουν στο παιχνίδι. Ο παίκτης όμως μπορεί να κινηθεί μόνο σε 4 κατευθήνσης όμως υποιήσαμε τη συνάρτηση move και για τις 3 οντώτητεες και τα ορίσματα της είναι σε μορφή bool γιατί χρισημοποιούμε ελέγχουε με if από την είσοδο του χρήστη για να ορίσουμε την επόμενη κατέυθηνση του. Για το avatar ελέγχει τις διαθέσημες κινήσεις που μπορέι να κάνει και αν δεν υπάρχουν επιστρέφει b , για την κάθε κίνηση τυπώνει το κατάλληλο μύνημα για να γνωρίζει και ο χρίστης τις διαθέσημες κινήσεις. Έπειτα τυπώνει και τα κατάλληλα μυνήματα για την επόυλωση (heal) την παύση και το τέλος του παιχνιδιού και ζητά από τον χρίστη να του δώσει το κατάλληλο όρισμα και το επιστρέφει στην συνάρτηση next_cycle και αν λάβει λάθος όρισμα από τον χρίστη του ξαναζητά ορθά ορίσματα τυπόνοντας τα απαραιτητα μυνήματα.
Στη συνέχεια κάναμε τους accesors για την health_potions και μια συνάρτηση set για την ομάδα του χρίστη. Σαν private ορίσαμε τα health_potions καιτ το team του χρίστη για να είναι προσβάσημα μόνο από την κλάση του avatar.

Ακολούθος δημιουργήσαμε μια νέα μητρική κλάση που είναι απόγωνος της entities εν ονόματη NPC και ως παιδία τις κλάσεις werewolves και vampires γιατί είναι μη ελεγχώμενοι χαρακτήρες εξού και το όνομα τις κλάσης.
Στην npc θέσαμε σαν protected τα health, medical , power και defense για να έχουν πρόσβαση μόνο οι 2 υποκλάσεις της.
Στην κλάση werewolves δημιουργήσαμε τον constructor του όπου παίρνει σαν ορίσματα τις συντεταγμένες χ και ψ και τα τοποθετεί στις συντεταγμένες του συγκεκριμένου werewolf και τέθει το health με 5 (δική μας παραδωχή γιατί δεν προσδιοριζόταν στις σημειώσεις) , μετά αρχικοποιήσαμε το medical , power και defense με τυχαίες τιμές σύμφωνα με τις οδηγίες που λάβαμε από την εκφώνηση.

Επίσης διμιουργήσαμε accesor των coordinates, health , medical , power και defence. Η move στην werewolves λειτουργεί ως εξής : επιλέγει τυχαία αν θα κινηθεί ή αν θα παραμήνει στάσημη.
Βλέπει αν έχει κίνηση να κάνει και μετά επιλέγει τυχαία τι κίνηση θα κάνει ανάλογα με τις διαθέσημες κινήσεις με μέγιστες επαναλύψεις εύρεσης κίνησης το 20 (variable step) για να αποφύγουμε ένα endless loop.
Για την κλάση vampires κάναμε ακριβώς το ίδιο πράγμα με την κλάση werewolves με μόνη διαφορά στην συνάρτηση move όπου προσθέσαμε την ικανότητα κίνησης σε διαγώνιους άξωνες και τις 8 μεταβλητές .
Για όλες τις συναρτήσεις move των οντοτήτων επιστρέφει ένα χαρακτήρα όπου προσδιορίζει προς τα που θα κινηθούν και αν θα κινηθούν. Οι χαρακτήρες αυτοί είναι οι το b δεν θα κάνει κίνηση το entity το w πάνω κίνηση , s κάτω , a αριστερά , d δεξιά , q πάνω αριστερά , e πάνω δεξιά , x κάτω δεξιά και το z κάτω αριστερά.

Επίσης οι συντεταγμένες που χρισιμοποιούν όλα τα classes ορίστηκαν σε ένα άλλο header file εν ονόματι coordinates όπου διμιουργήσαμε μια κλάση που απλά έχει 2 ακέραιες μεταβλητές οι οποίες αντοιστηχούν στις θέσεις χ και ψ.

Για το MAP:
Δημιουργήσαμε την κλάση map σε ένα αλλο header file όπου θέσαμε private το day_night όπου με τιμή 1 είναι μέρα και με τιμή 2 είναι νύκτα , το max_day_night είναι ο μετριτής που όταν φτάσει δέκα εναλλάζεται η μέρα με την νύχτα , number_werewolves , number_vampires , number_entities όπου αποθηκεύεται ο αριθμός των αντοίστηχων οντωτήτων που υπάρχει στον χάρτη. Αρχικά τοποποθετήσαμε το μέγεθος του πίνακα με τη ματαβλητή size η οποία αρχικοποιήθηκε σε coordinates ,δημιουργήσαμε ένα array με διπλό δείκτη με χαρακτήρες όπου είναι ο χάρτης τον οπίο βλέπουμε στην οθώνη μας . Δημιουργήσαμε ένα πίνακα οντωτήτων με 1350 θέσεις (το 1350 δεν υπερβένεται γιατί οι διαστάσεις του array είναι maximum 100χ100 και γνωρίζουμε οτι δεν είναι σωστη διαχείρηση μνύμης αλλα δεν καταφέραμε να το κάνουμε να δουλέψει με δυναμική δέσμευση μνήμης) .
Στα public της map δημιουργήσαμε τον constructor να πέρνει σαν είσοδο το μήκος και το πλάτος που θα εισάγει ο χρήστης. Μέσα στον constructor τοποθετούμε το μήκος και το πλάτος στο size , χρισημοποιόντας την συνάρτηση τυχαίας τιμής (για την οποία μιλούμε πάρακατω) για να προσδιορήσουμε αν θα ξεκινήσει το πρόγραμμα όντας μέρα ή νύχτα . Έπειτα δεσμεύσαμε δυναμικά το κατάλληλο χώρο στη μνύμη για το array μας ,αρχικοποιήσαμε όλες τις θέσεις του πίνακα μας με τελίες (. για γή) ,τοποθετήσαμε σε τυχαία θέση τα δέντρα (D) όπου το πλήθος τους είναι ίσο με χ*ψ/30 , τέλος τοποθετίσαμε και το νερό σε τυχαία θέση (~) όπου το πλήθος του είναι ίσο με χ*ψ/30.
Ζητάμε από τη χρίστη να επιλέξει ομάδα συμβολίζοντας το L σαν λυκάνθρωπος και B τα βαμπίρ , έπειτα τοποθετούμε τον avatar σε τυχαία θέση στον πίνακα και διμηουργήσαμε τον avatar με τα κατάλληλα ορίσματα. Για να τοποθετήσουμε τους werewolves και vampires στον χάρτη μας κάναμε 2 for loops όπου επιλέγει τυχαία θέση στο array μας για να τοποθετηθέι το κάθε entity και επείτα διμηουργήσαμε τα werewolves και τα vampires με δυναμικό τρόπο αντοίστηχα. Για τον πίνακα ent[] στην θέση 0 είναι ο avatar μας , από τις θέσεις 1- number_werewolves ειίναι τα werewolves και από τις θέσεις (number_werewolves+1)- (number_entities-1) είναι τα vampires. Τοποθετήσαμε σε τυχαία θέση το μαγικό φίλτρο($) όπου εμφανίζεται μόνο μία φορά στο παιχνίδι.
Μετά δημιουργήσαμε τον destructor του map όπου αποδεσμέυσαμε τις θέσεις μνήμης για το array και το ent που δεσμεύσαμε.
Στο next_cycle ελέξαμε το array για να βρούμε αν 2 όντα είναι δίπλα το ενα στο άλλο στον άξονα χ και στον άξονα ψ και για τα βαμπίρ και στους 2 διαγώνιους άξωνες (χ*ψ και -χ*ψ) βάζωντας 2 συνθήκες: αν είναι ίδιας ομάδας κάνουμε ένα for το οποίο βρίσκει την θέση στον πίνακα ent[] των συγκεκριμένων οντωτήτων και τις αποθηκεύει στο e1 και e2 .Ελέγχουμε αν το e1 δεν έχει πλήρες ζωή και αν το e2 έχει διαθέσημο γιατρικό και επίσης επιλέγει τυχαία αν θα δώσει γιατρικό ή όχι και αν ισχύουν οι πάραπανω

προυποθέσεις αυξάνουμε τη ζωή του e1 κατά 1 και μειώνουμε το γιατρικό του e2 κατά 1. Αυτός ο έλεγχος γίνεται και ανάποδα δηλαδή το e2 γιατρεύει το e1 και γίνοται οι ανάλογες ρυθμήσεις. Αν είναι αντίπαλες οντώτητες κάνουμε 2 for loop τα οποία βρίσκουν την θέση στον πίνακα ent[] των συγκεκριμένων οντοτήτων όπου είναι αντίπαλες και τις αποθηκεύει στο e1 και στο e2. Ελέγχουμε αν η δύναμη του e1 είναι μεγαλήτερη ή ίση με τη δύναμη του e2 τότε αποθηκεύουμε στο attack τη διαφορά μεταξύ power του e1 και defense του e2 . Αν αυτή η διαφορά είναι θετική τότε μειώνετε η ζωη του e2 κατά αυτή τη διαφορά. Αν η ζωή του e2 φτάσει στο 0 και κάτω τότε διαγράφεται η οντότητα από το χάρτη και αντικαστήται απο γη(.) και βάζουμε στη διαγραμμένη οντώτητα ως συντεταγμένες το (-1,-1) για να μπορού ξέρουμε ότι διαγράφηκαν και vise versa.
Ελέγχουμε αν υπάρχει νικητής στο παιχνίδι και αν υπάρχει τυπώνει το κατάλληλο μύνημα, τελειώνει το παιχνίδι.
Για την μετακίνηση οντωτήτων χρισημοποιούμε ένα βρογχο για να προσπελάσουμε όλες τις οντότητες. Ελέγχουμε αν η οντότητα είναι διαγεγραμμένη και αν είναι δεν την πειράζουμε . Αρχικοποιούμε 8 boolean μεταβλητές με false (για να ξέρουμε σε ποιές θέσεις μπορεί να πάει η κάθε οντότητα για κάθε πιθανή κίνηση) για τους λυκάνθρωπους και το avatar ελέγχουμε 4 πιθανές μετακινήσεις ενώ για τα βαμπίρ 8. Καλούμε την συνάρτηση move η οποία μας επιστρέφει ένα χαρακτήρα . Ελέγχουμε άν είναι ο avatar και μας επιστρέφει P όπου σημαίνει παύση τότε το παιχνίδι τήθεται σε παύση και τυπώνει στο χρίστη τα μαγικά του φίλτρα και τον αριθμό των λυκάνθρωπων και των βαμπίρ που έχουν απομέινει . Με το πλήκτρο P συνεχίζει το παιχνίδι ενώ με το O τερματίζει το παιχνίδι (ο χρίστης έχει την επιλογή να τερματίσει το παιχνίδι όταν βρίσκεται σε παύση).
Ελέγχουμε άν είναι ο avatar και μας επιστρέφει O τότε τερματίζεται το παιχνίδι.
Αν επιστραφεί b τότε η οντώτητα δεν μπορεί να μετακινηθεί.
Μετά ελέγχουμε αν είναι avatar και μας επιστρέφει H τότε επουλώνει τα μέλη της ομάδας του και τυπώνει τα κατάλληλα μυνήματα.
Οτάν επιστρέψει W,S,A,D,Q,E,X,Z μετακινείται η κάθε οντότητα στην κατάλληλη θέση ανάλογα με το τι έχει επιστρέψει η συνάρτηση move. Ελέγχουμε αν είναι ο avatar και το η θέση που θέλει να πάει είναι το μαγικό φίλτρο και αν ισχύει αυτό αυξάνουμε το health potion κατά 1.
Επίσης εναλλάζεται και η μέρα με τη νύχτα κάθε δέκα επαναλύψεις της προαναφερόμενης διαδικασίας
Υπερφορτώσαμε το τελεστή << για να τυπώνει με τον επυθημητό τρόπο το πίνακα μας και επίσης τυπώνεται το αν είναι μέρα ή νύχτα κάθε επανάλυψη.
Η συνάρτηση number_of_entities μας επιστρέφει τον αριθμό των οντωτήτων που που εμείς θέλουμε να μάθουμε τον αριθμό τους.

Η συνάρτηση random_coordinates μας επιστρέφει μία τυχαία συντεταγμένη ανάμεσα στα χ και ψ που θα της δώσουμε.
Η συνάρτηση random_number μας επιστρέφει ένα τυχαίο αριθμό ανάμεσα στο min και max που τις δίνουμε .

Για την main:
Η main ζητά από το χρίστη πλάτος και ύψος , έπειτα διμηουργέι ένα Map με τα ορίσματα που τις έδωσε ο χρίστης και έπειτα χρισημοποιούμε ένα βρόγχο ο οποίος αρχικά μας εκτειπώνει το χάρτη και καλέι την συνάρτηση next_cycle και ανάλογα με το τι επιστρέφει συνεχίζει ή σταματά ο βρόγχος.

- Χρισημοποιήσαμε Microsoft Virtual studio
- Προβλήματα: Το μόνο πρόβλημα που αντιμετωπείσαμε ήταν οτι δεν βρήκαμε τρόπο για δυναμική δέσμευση μνήμης για τον πίνακα ent[] και ανναγκαστήκαμε να τον ορίσουμε από πριν.Δεν ξέρουμε πως να ανεβάσουμε όπως θελετε εσεις το βίντεο στο Youtube.
- Υλοποιήσαμε όλες τις απαιτίσης της εκφώνησης

- Βαθμός δυσκολίας ήταν σκετικά εντάξει δεν ήταν παρα πολύ δύσκολη εργασεια.