

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ ΕΡΓΑΣΙΑΣ

- **ΣΤΟΙΧΕΙΑ ΜΕΛΩΝ ΟΜΑΔΑΣ:**

- ο ΚΑΓΙΑΤΣΚΑ ΕΡΙΚ:
AM: 1115202100043
ΕΤΟΣ: 2^ο
EMAIL: sdi2100043@di.uoa.gr
- ο ΚΑΛΑΜΠΟΚΗΣ ΕΥΑΓΓΕΛΟΣ:
AM: 1115202100045
ΕΤΟΣ: 2^ο
EMAIL: sdi2100045@di.uoa.gr

- **ΑΣΧΟΛΙΑ ΤΟΥ ΚΑΘΕ ΜΕΛΟΥΣ:**

- ο ΚΑΓΙΑΤΣΚΑ ΕΡΙΚ: state.h, reload function, move function, create functions, mutators - accessors, help and attack functions, menu and board functions, deallocation, main function
- ο ΚΑΛΑΜΠΟΚΗΣ ΕΥΑΓΓΕΛΟΣ: state.h, check function, movement functions, create functions, add function, state_create function, state_update function, deallocation, main function

Ωστόσο κάθε συνάρτηση , struct, class και γενικά κάθε υλοποίηση πραγματοποιούνταν έπειτα από συνεννόηση και έγκριση και των δύο μελών, αφού σημαντική κρίθηκε η παρέμβαση και των δύο σε ορισμένα κομμάτια κώδικα.

- **ΠΕΡΙΓΡΑΦΗ ΠΡΟΓΡΑΜΜΑΤΟΣ/ΚΩΔΙΚΑ:**

Το πρόγραμμα ξεκινάει ζητώντας από τον χρήστη να δώσει τις διαστάσεις του χάρτη και να διαλέξει την ομάδα που θα υποστηρίξει μέσω της main. Ο χάρτης δεσμεύεται δυναμικά στη main και τα αντικείμενα(δέντρα, νερά , vampires, werewolves και ο avatar) δημιουργούνται μέσω των create functions και τοποθετούνται στον χάρτη μέσω της add η οποία καλείται μέσω της state_create που αρχικοποιεί την κατάσταση του παιχνιδιού. Παράλληλα τα αντικείμενα τοποθετούνται σε ξεχωριστά vectors που βρίσκονται στο struct state ώστε να είναι ευκολότερη η πρόσβαση σε αυτά. Έπειτα τοποθετούνται τυχαία σε ξεχωριστές θέσεις στον χάρτη και αμέσως ξεκινάνε να κινούνται τυχαία. Η ενημέρωση του state(κατάστασης του παιχνιδιού) γίνεται μέσω της state_update που καλείται όσο το παιχνίδι βρίσκεται σε εξέλιξη, η

πληροφορία αυτή παρέχεται από το `state_info` που περιέχει όλες τις βασικές πληροφορίες του παιχνιδιού. Μέσω της `state_update` καλούνται όλες οι απαραίτητες συναρτήσεις (κίνησης - επίθεσης και βοήθειας των αντικειμένων) οι οποίες με τη σειρά τους χρησιμοποιούν τα `accessors` και `mutators` που έχουμε ορίσει (επιλέξαμε να μη χρησιμοποιήσουμε `constructors` σε κλάσεις πέρα από τον ίδιο το χάρτη καθώς οι τιμές των μελών τους μεταβάλλονται συνεχώς και θεωρήσαμε ότι η διαχείριση αυτών των συνεχών μεταβολών θα ήταν ευκολότερη μέσω `accessors` και `mutators`). Η εκτύπωση του χάρτη και των αντικειμένων γίνεται μέσω της συνάρτησης `board` που καλείται έπειτα από κάθε κλήση της `state_update`. Μόλις η ζωή ενός αντικειμένου μηδενιστεί, η μνήμη που είχε δεσμευτεί για αυτό αποδεσμεύεται, και διαγράφεται από το `vector` στο οποίο βρισκόταν. Οι συντεταγμένες του αποδεσμεύονται και αυτές και διαγράφονται από το ξεχωριστό `vector Locations` που χρησιμοποιείται για να ρυθμίζεται σωστά η κίνηση και η τοποθέτηση των αντικειμένων στον χάρτη. Μόλις το παιχνίδι τερματιστεί (πράγμα που μπορεί να συμβεί νωρίτερα με το πάτημα του πλήκτρου `q` εάν το επιθυμεί ο χρήστης) η εναπομείνουσα δεσμευμένη μνήμη των αντικειμένων (`surroundings`, `points` και τα ίδια τα αντικείμενα) αποδεσμεύεται. Περαιτέρω πληροφορίες υπάρχουν στα σχόλια του κώδικα και στο βίντεο της παρουσίασης του παιχνιδιού.

- **ΠΑΡΑΔΟΧΕΣ ΠΑΙΧΝΙΔΙΟΥ:**

- Η επούλωση κάθε οντότητας από τον `avatar` σημαίνει ότι θα προστίθεται 1 επιπλέον μονάδα στην υγεία της κάθε οντότητας της ομάδας που θα έχει επιλέξει ο παίκτης. Αυτό θα γίνεται μόνο όταν η τιμή της ζωής της οντότητας δεν είναι η μέγιστη.
- Ο `avatar` κουνιέται με τα πλήκτρα `w,a,s,d`. Το `pause` γίνεται με το πλήκτρο `p` και `quit` με το `q`. Επιπλέον, το φίλτρο χρησιμοποιείται με το πλήκτρο `h`.
- Το `health` των `Creatures` έχει τυχαία τιμή από 1 έως 5.
- Ο `avatar` συμβολίζεται με `A`, οι `werewolves` με `w`, τα `vampires` με `v`, τα δέντρα με `T`, το νερό με `W` και το μαγικό `potion` με `P`.
- Οι διαστάσεις του χάρτη που δίνονται από τον χρήστη είναι, πλάτος από 20 έως 140 και ύψος από 15 έως 40, ώστε η εμπειρία του παιχνιδιού να είναι καλύτερη. Χωρίς αυτές τις προτεινόμενες διαστάσεις υπήρχε ο κίνδυνος ο χρήστης να

δώσει πολύ μεγάλες διαστάσεις, οι οποίες θα δυσκόλευαν τον ίδιο (θα χρειαζόταν scroll up-down / left- right στο terminal).

- Η εναλλαγή μεταξύ μέρας και νύχτας γίνεται κάθε 20 επαναληπτικές διαδικασίες της state_update.
- Ο αριθμός των δέντρων και των νερών επιλέχθηκε να είναι σταθερός και ίσος του 5.

- **IDE/COMPILER ΚΑΙ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ:**

Το παιχνίδι έχει δοκιμαστεί σε λειτουργικό περιβάλλον windows. Έχει χρησιμοποιηθεί το εργαλείο visual studio 2022 και ο local windows debugger που προσφέρεται μαζί με το εργαλείο αυτό.

- **ΠΡΟΒΛΗΜΑΤΑ:**

Ένα από τα προβλήματα που αντιμετωπίσαμε ήταν η τυχαία προσθήκη των αντικειμένων στο χάρτη, ώστε να μην τοποθετούνται στην ίδια θέση. Έπειτα από ομαδική δουλειά καταλήξαμε στο συμπέρασμα ότι ένα βέκτορ που θα είχε τις τοποθεσίες όλων των αντικειμένων θα έλυνε το πρόβλημα, όπως και έγινε.

Αυτό μας έλυσε ακόμη ένα πρόβλημα, αυτό της κίνησης των αντικειμένων στον χάρτη, καθώς με τους απαραίτητους ελέγχους, αποφύγαμε την περίπτωση να κινηθεί ένα αντικείμενο στην ίδια θέση με ένα άλλο.

- **ΜΗ ΥΛΟΠΟΙΗΜΕΝΕΣ ΑΠΑΙΤΗΣΕΙΣ:**

Δεν υπάρχουν απαιτήσεις που να μην έχουν υλοποιηθεί.

- **ΒΑΘΜΟΣ ΔΥΣΚΟΛΙΑΣ:**



- **ΑΠΟΘΕΤΗΡΙΟ GITHUB:**

<https://github.com/erikk03/WerewolvesVsVampires.git>

- **ΒΙΝΤΕΟ YOUTUBE:**

<https://youtu.be/9F4JwrVV3zk>