

Αντικειμενοστραφής Προγραμματισμός
Χειμερινό εξάμηνο 2022 - 2023
Ομαδική εργασία
WnV the Game

- Μέλη της ομάδας :

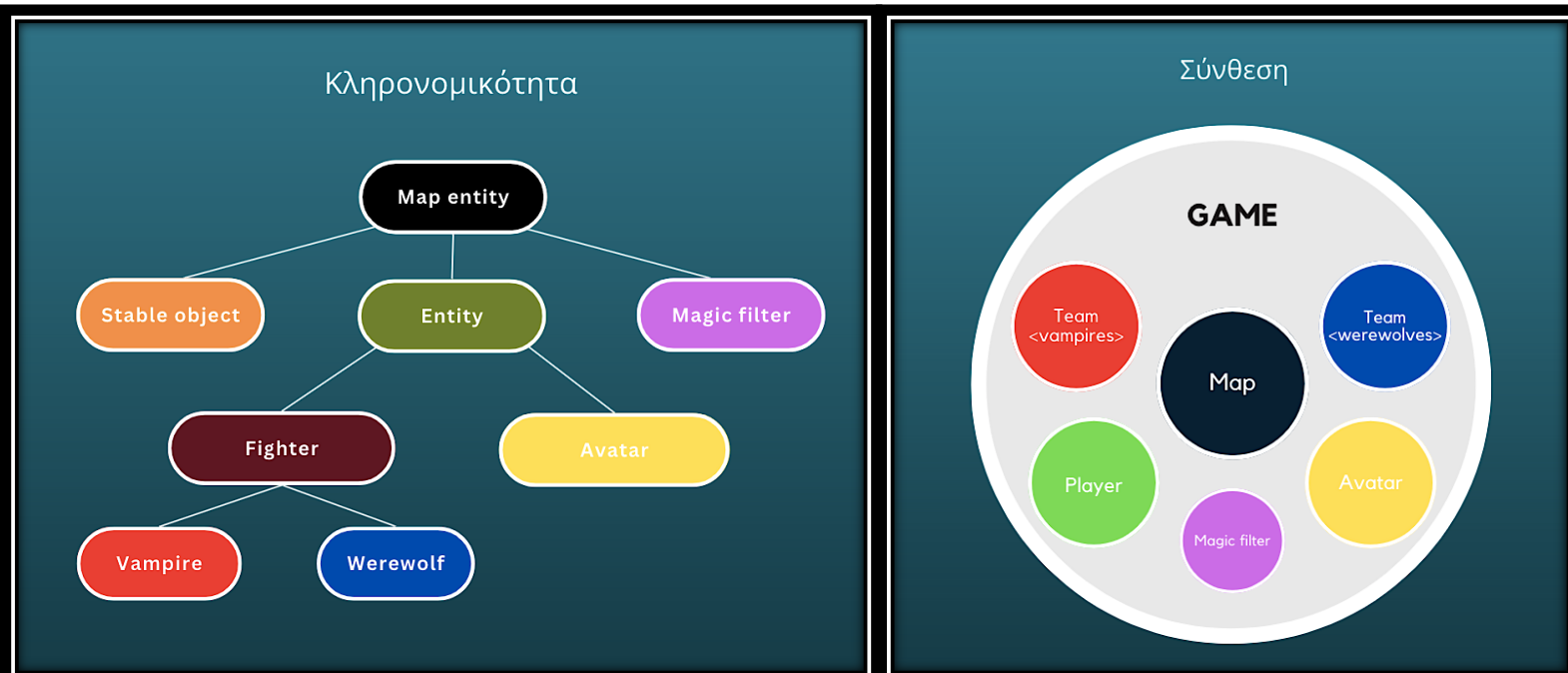
Ονοματεπώνυμο : Λαζανάκη Άρτεμις
Α.Μ. : 1115202100081
Έτος : 2^ο
email : art.lazanaki@gmail.com

Ονοματεπώνυμο : Παπαφίλη Ευθυμία
Α.Μ. : 1115202100144
Έτος : 2^ο
email : epapafili2003@gmail.com

- Διαμερισμός εργασίας

Η εργασία πραγματοποιήθηκε από κοινού.

- Περιεκτική περιγραφή του προγράμματος/κώδικα



Απεικόνιση των σχέσεων μεταξύ των κλάσεων του προγράμματος μας

Το πρόγραμμα ξεκινάει καλώντας τη συνάρτηση **set_player_preferences()**, η οποία παίρνει ως ορίσματα αναφορές των ήδη ορισμένων στη **main** μεταβλητών **dim_x(int)**, **dim_y(int)**, **team(char)** και είναι υπεύθυνη για την **εισαγωγή και τον έλεγχο της εγκυρότητας** των προτιμήσεων του παίκτη για το μέγεθος του map(διαστάσεις **dim_x**, **dim_y**) και την ομάδα που θα υποστηρίξει (**team**). Τα δεδομένα αυτά περνάνε στη **main** μέσω των αναφορών και έπειτα στον constructor του **game**. Μέσω του constructor, λόγω σύνθεσης, δημιουργούνται κατάλληλα και όλα τα αντικείμενα των κλάσεων που περιέχει.

Το παιχνίδι ξεκινά μέσω της κλήσης της μεθόδου του **Game run()**, η οποία αρχικά εκτυπώνει σημαντικές οδηγίες και πληροφορίες που αφορούν τον παίκτη. Με το πάτημα ενός οποιουδήποτε κουμπιού η μάχη ξεκινάει. Το παιχνίδι ουσιαστικά εξελίσσεται στο πλαίσιο ενός **while loop** που επαναλαμβάνεται όσο το παιχνίδι είναι **active**. Στην περίπτωση που ο παίκτης δεν πατήσει κάποιο κουμπί επαναλαμβάνεται ένα εμφωλευμένο **while** στο εξωτερικό με συνθήκη **!_kbhit()** (συνάρτηση που επιστρέφει **true** αν ανιχνεύεται πατημένο πλήκτρο) στο οποίο καλούνται διαδοχικά οι **update()**, **print()** και η **interactions()**, που επιστρέφει **true** μόνο αν έχει γίνει κάποιο **defense** και τότε μόνο καλείται ξανά η **print()**. Ελέγχεται και στα 2 loops αν έχει αφανιστεί κάποια από τις 2 ομάδες ώστε να τεθεί το **active false** και να κληθεί η συνάρτηση **end()** του **game** καθώς και αν έχουν γίνει 40 εκτυπώσεις ώστε να γίνει εναλλαγή από μέρα σε νύχτα και το αντίστροφο(παραδοχή). Στην περίπτωση που ο παίκτης πατήσει κάποιο κουμπί, εκτός των άλλων συναρτήσεων καλείται και η **set_input()** η οποία καλεί την **_getch()** (μας βόλεψε περισσότερο από την

GetKeyState() που μας έχετε δείξει) και περνάει τον χαρακτήρα που επιστρέφει στην ακέραια μεταβλητή input του object player. Ανάλογα με το key code που περιέχεται στην input μεταβλητή εκτελείται και η αντίστοιχη ενέργεια που αφορά την κατάσταση του παιχνιδιού ή τον avatar.

Ανάλυση των προαναφερόμενων συναρτήσεων

→ **update()** : μέθοδος της κλάσης game, υπεύθυνη για την τυχαία κίνηση των vampires και των werewolves. Αυτό το επιτυγχάνει προσπελάζοντας τα 2 vector που υπάρχουν στις template κλάσεις team_vampires και team_werewolves του game, τα οποία περιέχουν τα μέλη των ομάδων και αλλάζοντας στο κάθε μέλος τυχαία τις συντεταγμένες του αλλά και το που δείχνουν οι pointers που βρίσκονται στην παλιά και την νέα τους θέση μέσω της συνάρτησης swap(). Η κατεύθυνση επιλέγεται τυχαία μέσω της καθολικής συνάρτησης get_random(), η οποία επιστρέφει έναν τυχαίο ακεραίο στο διάστημα που ορίζουν τα 2 της ορίσματα. Τα ενδεχόμενα για τα werewolves είναι 5 και για τα vampires 9 συμπεριλαμβανομένης της ακινησίας.

→ **print()** : μέθοδος της κλάσης του map, υπεύθυνη για την εκτύπωση του grid με τα σύμβολα που αντιστοιχούν στο type που έχει ως μεταβλητή η κλάση map entity. Παίρνει ως όρισμα την bool μεταβλητή day ώστε να γίνει η κατάλληλη εκτύπωση.

→ **interactions()** : μέθοδος της κλάσης του game, υπεύθυνη για την αλληλεπίδραση των fighters. Προσπελάζει το grid και ελέγχει αρχικά αν η μεταβλητή checked του map_entity είναι true, μόνο τότε γίνεται continue. Η συγκεκριμένη μεταβλητή αρχικοποιείται ως true σε όλα τα stable objects και στον avatar, ενώ σε όλους τους fighters αρχικοποιείται ως false. Έτσι, μετά από αυτό τον έλεγχο ξέρουμε σίγουρα ότι στην εκάστοτε θέση του grid υπάρχει fighter. Στην συνέχεια, ελέγχονται όλες οι γειτονικές θέσεις του (reminder: γειτονικές θέσεις θεωρούνται οι πάνω, κάτω, δεξιά, αριστερά) για fighters και θέτονται οι αντίστοιχες λογικές μεταβλητές σε true ή false. Όταν μια λογική μεταβλητή που σχετίζεται με θέση γίνεται true τότε το γράμμα που αντιστοιχεί στη θέση προστίθεται σε ένα vector χαρακτήρων. Στο τέλος, προσθέτονται καθώς το άθροισμα τους αποτελεί και τον αριθμό των ενδεχόμενων αλληλεπιδράσεων(count). Αν υπάρχει τουλάχιστον ένα ενδεχόμενο, τότε καλείται η get_random() που επιστρέφει έναν αριθμό από 0 μέχρι count-1 και επιλέγεται έτσι η θέση του vector άρα και η σχετική θέση του fighter με τον οποίο θα αλληλεπιδράσει. Μέσω dynamic cast μετατρέπονται οι pointers σε map_entity σε pointers σε fighters, ώστε να κληθεί η μέθοδος της κλάσης fighter, interact(). Στο τέλος της συνάρτησης, ξαναγίνονται όλα τα checked των fighters false.

→ **end()** : μέθοδος της κλάσης του game, υπεύθυνη για την εκτύπωση της έκβασης του παιχνιδιού. Επίσης δίνει την δυνατότητα στον παίκτη να δει τις πληροφορίες των παικτών της νικητριάς ομάδας και του παιχνιδιού καλώντας την display_game_info().

→ **pause()** : μέθοδος της κλάσης του game, υπεύθυνη για την παύση του παιχνιδιού σε περίπτωση πατήματος του space από τον παίκτη, για την επανέναρξη του και για την εκτύπωση των πληροφοριών. Καλεί την display_info() η οποία καλεί την display_game_info() (εμφανίζει αριθμό ζωντανών werewolves, vampires και αριθμό magic filter που έχει ο avatar στην κατοχή του) και εκτυπώνει πληροφορίες για τα ζωντανά μέλη της ομάδας που υποστηρίζει ο παίκτης.

Ανάλυση των συναρτήσεων που προκύπτουν από την κλήση της interactions()

→ **interact()** : μέθοδος της κλάσης fighter, υπεύθυνη για την εκτέλεση της σωστής αλληλεπίδρασης των fighters. Επιστρέφει true αν έχει γίνει defense. Αν οι fighters ανήκουν στην ίδια ομάδα, τότε καλείται η συνάρτηση give_heal(). Διαφορετικά, καλείται πρώτα η defend() και αν αυτή αποτύχει τότε καλείται η attack().

→ **give_heal()** : μέθοδος της κλάσης fighter, υπεύθυνη για την προσφορά γιατρικού σε συμπαίκτη αν έχει κατά 1 στις 6 πιθανότητα(βλ. παραδοχές).

→ **defend()** : μέθοδος της κλάσης fighter, υπεύθυνη για την απομάκρυνση του fighter από αυτόν που του επιτίθεται. Δέχεται σαν ορίσματα την θέση που βρίσκεται ο αντίπαλος σε σχέση με τον ίδιο και το map προκειμένου να κινηθεί προς την αντίθετη κατεύθυνση από αυτή της επίθεσης.

→ **attack()** : μέθοδος της κλάσης fighter, υπεύθυνη για την επίθεση αλλά και την εξαφάνιση των fighters. Αρχικά μειώνεται η υγεία του θύματος ανάλογα και έπειτα ελέγχεται αν αυτή έχει μηδενιστεί. Αν ναι, τότε μέσω της find_if γίνεται η αναζήτηση αυτού στο κατάλληλο vector. Ακολουθεί η διαγραφή του από το vector και από το grid και μπαίνει στην θέση του γη.

• Παραδοχές για την υλοποίηση του προγράμματος

→ Στην αρχή του παιχνιδιού ο Avatar τοποθετείται περίπου στη μέση του grid (x/2+1 και y/2+1, όπου x, y οι διαστάσεις του map)

- Ο αρχικός αριθμός των vampires και werewolves είναι πάντα $x * y / 30$, , όπου x, y οι διαστάσεις του map
- Η υγεία των fighters είναι στην αρχή 3 και αυτή αποτελεί και τη μέγιστη τιμή κατά τη διάρκεια του παιχνιδιού
- Τα δέντρα και οι λίμνες καλύπτουν το 4% των θέσεων του grid.
- Όταν ξεκινάει το παιχνίδι είναι μέρα
- Η εναλλαγή από ημέρα σε νύχτα γίνεται κάθε 40 frames (εκτυπώσεις του grid)
- Αν ένας fighter έχει παραπάνω από έναν γειτονικούς, τότε επιλέγεται τυχαία με ποιον θα αλληλεπιδράσει
- Γειτονικές θέσεις για μια αλληλεπίδραση θεωρούνται η πάνω, η κάτω, η δεξιά και η αριστερά
- Η άμυνα πραγματοποιείται με την κίνηση του αμυνόμενου fighter προς την αντίθετη κατεύθυνση από εκείνη της επίθεσης
- Η επούλωση των μελών της ομάδας του από τον Avatar απαιτεί κατοχή και χρήση 10 magic filter.
- Όταν κατά την επίθεση η δύναμη του επιτιθέμενου είναι ίση με το επίπεδο άμυνας του αμυνόμενου τότε η υγεία του θα μειώνεται κατά ένα. Διαφορετικά, αν η δύναμη του επιτιθέμενου (power) είναι μεγαλύτερη από το επίπεδο άμυνας του αμυνόμενου (defense) τότε υπολογίζεται η διαφορά και αφαιρείται από την υγεία του.
- Αν δύο fighters έχουν ίσο επίπεδο δύναμης τότε δεν συμβαίνει defend και επιλέγεται τυχαία ποιος από τους δύο fighter θα επιτεθεί
- Οι διαστάσεις του grid πρέπει να κυμαίνονται μεταξύ $10 * 10$ και $30 * 30$ για να υπάρχει επαρκής αριθμός vampire και werewolf σε αυτό και για μία χρονικά λογική διάρκεια του παιχνιδιού, δίχως προβλήματα στην εκτύπωση του στο terminal.
- Το στίγμα του Avatar στο grid είναι 'A'. Τα δέντρα συμβολίζονται με '*' και οι λίμνες με '~'. Τα werewolves συμβολίζονται με 'w', τα vampires με 'v' και το magic filter με 'm'
- Για την κίνηση του Avatar χρησιμοποιούνται τα πλήκτρα βέλους. Η παύση του παιχνιδιού με την ταυτόχρονη προβολή των πληροφοριών γίνεται πατώντας space. Η χρήση του magic filter γίνεται πατώντας 'F'. Η έξοδος από το παιχνίδι γίνεται πατώντας 'X'.
- Υπάρχει πιθανότητα 1 στις 6 να δοθεί γιατρικό μεταξύ συμπαικτών

- **IDE/Compiler :** Visual Studio 2022 Microsoft (ISO C++ - 14 Standard)

- **Προβλήματα που αντιμετωπίσαμε**

- **Λειτουργικό σύστημα :** Ξεκινήσαμε την εργασία δουλεύοντας σε wsl στο vscode, αλλά αντιμετωπίσαμε πρόβλημα στην εύρεση συνάρτησης για τη λήψη του input-key του player. Έτσι, αποφασίσαμε να μεταφέρουμε τον κώδικα προσαρμόζοντάς τον στο Visual Studio, σε λειτουργικό windows.
- **Grid :** Στην αρχή το grid ήταν ένας δισδιάστατος πίνακας από χαρακτήρες. Στην προσπάθεια μας όμως να κάνουμε τους fighters να κινούνται , συνειδητοποιήσαμε ότι δεν υπήρχε σύνδεση του κάθε fighter(στιγμιότυπο) με την αντίστοιχη θέση του στο grid. Για το λόγο αυτό, μεταβάλλαμε το περιεχόμενο του πίνακα σε pointers σε Map_entity, που ήταν μια καινούργια κλάση που δημιουργήσαμε.
- **Μοντελοποίηση :** Στην αρχική μας μοντελοποίηση καθοριστική κλάση για το παιχνίδι ήταν η κλάση του map. Ωστόσο, συνειδητοποιήσαμε ότι αυτό δεν ήταν η βέλτιστη λύση καθώς με αυτό τον τρόπο το map αποκτούσε πρόσβαση σε δεδομένα που δεν συμβάδιζαν με τον ρόλο που είχε στο παιχνίδι. Έτσι, μεταφέραμε κάποιες μεθόδους και κλάσεις στην κλάση game.
- **Χρήση προτύπων :** Δεν γνωρίζαμε ότι για να γίνει compile πρέπει στο header file να περιέχονται η υλοποίηση και η δήλωση του προτύπου μαζί.

- **Βαθμός δυσκολίας εργασίας :** 8/10

- **Σύνδεσμος με το ιδιωτικό αποθετήριο στο github :** https://github.com/artemislz/WnV_the_game

- **Σύνδεσμος με βίντεο στο YouTube :** <https://www.youtube.com/watch?v=IEtQOYQZ9-4>