

Τελική Αναφορά Εργασίας

- Στοιχεία μελών την ομάδας:

1) Όνομα: Δημήτριος | Επίθετο: Χρυσός | A.M: 1115202100275 |
Έτος: 2ο | e-mail: sdi2100275@di.uoa.gr

2) Όνομα: Αναστάσιος | Επίθετο: Μουμουλίδης | A.M: 1115202100108 |
Έτος: 2ο | e-mail: sdi2100108@di.uoa.gr

(σημείωση: Λόγω λάθος git configuration που έγινε στο Visual Studio, σε όλα τα pushes εμφανίζεται ως username: Tassos_03 αντί για sdi2100108).

- Με τι ασχολήθηκε ο καθένας:

- Γενικά σε όλον τον κώδικα υπήρχε κοινή απασχόληση και των 2 ατόμων της εργασίας σε πρακτικά όλους τους τομείς. Παρόλα αυτά ο καθένας από εμάς ασχολήθηκε σε κάποια κομμάτια πιο πολύ από τον άλλο, πιο συγκεκριμένα:

1) Αναστάσιος Μουμουλίδης: κίνηση οντοτήτων (decide, genmove),
constructors οντοτήτων, constructor board, heal

2) Δημήτριος Χρυσός: Avatar movement, game_input, spawn οντοτήτων,
ελέγχους στην main

- Τα κομμάτια που δεν αναφέρονται παραπάνω έχουν γίνει από κοινού

- Περιεκτική περιγραφή του προγράμματος/κώδικα:

- Το πρόγραμμα μας αποτελείται από 5 Source Files, τα project_main.cpp, game_input.cpp, Decide.cpp, Entity.cpp, Board.cpp και 2 Header Files το game.h και το game_input.h

1) **project_main.cpp**: Είναι η main συνάρτηση του προγράμματος που δέχεται τα πρωταρχικά δεδομένα για τη δημιουργία του board κάνοντας τους απαραίτητους ελέγχους. Δημιουργεί το παιχνίδι και το αρχίζει μέσω της συνάρτησης game_input() και όταν τελειώσει, το διαγράφει από την μνήμη.

2) **game_input.cpp**: Αποτελείται από 2 συναρτήσεις την

a. check_player_really_stuck(), η οποία ελέγχει αν ο Avatar είναι κολλημένος και από τις 4 κατευθύνσεις από water & trees ώστε να δημιουργηθεί νέο παιχνίδι και να διαγραφεί το παλιό.

b. και την game_input() που δέχεται τα δεδομένα του χρήστη εκτελώντας την αντίστοιχη λειτουργία (κίνηση Avatar, heal, pause, end of game) και ελέγχει για το πότε αλλάζει η μέρα και τελειώνει το παιχνίδι.

3) **Decide.cpp**: Το αρχείο περιλαμβάνει τις συναρτήσεις decide() και heal_attack_avoid(), που περιέχουν τον αλγόριθμο τεχνίτης νοημοσύνης των οντοτήτων και πραγματοποιούν την αλληλεπίδρασή τους με γειτονικές οντότητες.

Τελική Αναφορά Εργασίας

- 4) **Entity.cpp:** Περιέχονται οι συναρτήσεις των κλάσεων που υλοποιούν τις οντότητες (Entity, Avatar, Npc, Werewolves, Vampires). Ανάμεσα σε αυτές είναι και οι contractors – destructors κάποιες συναρτήσεις getters, καθώς και οι συναρτήσεις make_avatar_movement(), gen_move() (για Vampires και Werewolves), move(), που σχετίζονται με την κίνηση των οντοτήτων και οι συναρτήσεις attack(), heal() που σχετίζονται με τις αλληλεπιδράσεις των οντοτήτων. Υπάρχουν επίσης κάποιες συμπληρωματικές συναρτήσεις που χρησιμοποιούνται στον υπόλοιπο κώδικα του προγράμματος.
- 5) **Board.cpp:** Περιέχει τις συναρτήσεις που σχετίζονται με τις κλάσεις Board και Block, ανάμεσα σε αυτές είναι ο constructor του Board που αρχικοποιεί το ταμπλό, η spawn_entities που τοποθετεί τις οντότητες στο ταμπλό και άλλες συμπληρωματικές συναρτήσεις, μαζί με συναρτήσεις setters και getters.
- 6) **game.h:** Περιέχονται οι ορισμοί των βασικών κλάσεων (Entity, Avatar, Npc, Werewolves, Vampires, Board, Block) με όλα τα members και τις συναρτήσεις τους.
- 7) **game_input.h:** Σε αυτό το αρχείο ορίζονται οι καθολικές συναρτήσεις check_player_really_stuck() & game_input().

• Παραδοχές που κάναμε:

- Οι οντότητες αλληλοεπιδρούν με όλες τις γειτονικές τους οντότητες σε μία κίνηση. Ειδικότερα, οι vampires μπορούν να κινηθούν διαγώνια ωστόσο δεν αλληλοεπιδρούν με τις οντότητες που βρίσκονται στα Blocks αυτά.
- Μετά από κάθε κίνηση του Avatar όλα τα Npc πραγματοποιούν μία κίνηση.
- Η κίνηση των Npc πραγματοποιείται ανεξάρτητα από την αλληλεπίδραση τους με τις άλλες οντότητες.
- Το αν το παιχνίδι ξεκινάει μέρα ή νύχτα είναι τυχαίο. Η μέρα εναλλάσσεται με την νύχτα κάθε 8 κινήσεις του Avatar.
- Όταν είναι μέρα μετά την κίνηση του Avatar κινούνται πρώτα όλοι οι Vampires και μετά όλοι οι Werewolves, ενώ την νύχτα το αντίθετο.
- Για εμάς πάντα ο αριθμός των οντοτήτων της κάθε ομάδας είναι: $x \cdot y / 15$.
- Το minimum size του Board είναι $4 \cdot 4$ και το maximum είναι $100 \cdot 100$.
- Στην οθόνη εμφανίζεται:
 - a. Η supporting team του Avatar -> «W» ή «V» για Werewolves & Vampires αντίστοιχα.
 - b. Werewolves -> «w».
 - c. Vampires -> «v».
 - d. Trees -> «*».
 - e. Water -> «~».
 - f. Magic Potion -> «+».

Τελική Αναφορά Εργασίας

- Έχουμε εξασφαλίσει την δυνατότητα επίθεσης και από τις 2 ομάδες, δίνοντας strength -> 3, σε τουλάχιστον 1 μέλος τους.
- Όλες οι οντότητες ξεκινάνε με ζωή 3 και όταν φτάσει η ζωή τους 0 πεθαίνουν.
- IDE, compiler και λειτουργικό σύστημα:
 - Χρησιμοποιήσαμε και οι 2, Windows και Visual Studio.
- Τυχόν προβλήματα που αντιμετωπίσαμε:
 - Αντιμετωπίσαμε κάποια μικρά προβλήματα αλλά ήταν κυρίως λογικά λάθη τα οποία με λίγο debugging αντιμετωπίσαμε.
- Απαιτήσεις που δεν υλοποιήθηκαν:
 - Καμία.
- Βαθμός Δυσκολίας:
 - Θεωρούμε ότι η εργασία ήταν μέτριας δυσκολίας.
- Σύνδεσμος με ιδιωτικό αποθετήριο στο github:
 - https://github.com/DimitrisChrysos/project_oop.git
- Σύνδεσμος με unlisted video στο YouTube:
 - <https://www.youtube.com/watch?v=cCJWJ2bw10s>