Μπακάλης Δημήτριος Α.Μ. 3033

Προγραμματιστική Άσκηση Γραφικών (OPENGL) 2019-2020

Το πρόγραμμα τρέχει αποκλειστικά και μόνο με την χρήση του CandyCrash.cpp, όπου γίνεται compile από το Makefile και δίνει ως έξοδο το output αρχείο, όπου είναι το executable.

Το πρόγραμμα αυτό υλοποιεί τα ερωτήματα i), ii), iii,) iv) και v) της εκφώνησης.

Υλοποιώ και χρησιμοποιώ τις κλάσεις **main(), startingMenu(), goMenu(), idle(), mouse(), destruction(), neighbor(), disOne(), disTwo**, **disThree(), init()** και την **display()**. Όλες οι κλάσεις παίρνουν κατάλληλα ορίσματα όπως φαίνεται και στον κώδικα.

Όλες οι λειτουργίες ξεκινούν με την **main().** Από εκεί καλούνται οι υπόλοιπες κλάσεις που αναφέρονται παραπάνω.

Για το ερώτημα i) και ii) χρησιμοποιώ τις: **startingMenu(), goMenu(), init()** και **display().**

Για το ερώτημα iii) χρησιμοποιώ τις: **mouse()**, **idle()** και **display().**

Για το ερώτημα iv) και v) χρησιμοποιώ τις: **destruction(), neighbor(), disOne(), disTwo**, **disThree()** και **display().**

- **startingMenu():** Υλοποιεί το menu και το «κουμπώνει» στο πάτημα του δεξιού κλικ του ποντικιού.
- **goMenu():** Ανάλογα την επιλογή του χρήστη (Start Game, Exit) κάνει την αντίστοιχη λειτουργία.
- **init():** Γεμίζω τον πίνακα των κύβων (board[16][16]) με συντεταγμένες και με έναν αριθμό κλειδί για το χρώμα που θα πάρει ο κάθε κύβος.
- **mouse():** Διαβάζω τις συντεταγμένες του αριστερού κλικ έτσι ώστε να καταλάβω σε ποιόν κύβο του πίνακα (board[16][16]) έκανε κλικ ο παίκτης.
- **idle():** Κάνει την αλλαγή των δυο κύβων, όπου επιλέγει ο χρήστης σε απόσταση αυστηρά 1 μεταξύ τους.
- **destruction():** Υλοποιεί την έκρηξη της τριάδας, με κάθε πιθανό συνδυασμό που μπορεί να υπάρξει τριάδα ίδιου χρώματος.
- **disOne:** Υλοποιεί την καταστροφή σε απόσταση +1 από την τριάδα. Καταστρέφει τα πάντα σε απόσταση 1, εκτός από αυτό που «καταστρέφει» τον τύπο της τριάδας.
- **disTwo:** Υλοποιεί την καταστροφή σε απόσταση +2 από την τριάδα. Καταστρέφει σε απόσταση 2, μόνο τους κύβους που «καταστρέφονται» από τον τύπο την τριάδας.
- **disThree:** Υλοποιεί την καταστροφή σε απόσταση +3 από την τριάδα. Καταστρέφει σε απόσταση 3, μόνο τους κύβους που «καταστρέφονται» από τον τύπο την τριάδας.

- **neighbor():** Υλοποιεί την καταστροφή όλων τον γειτονικών κύβων σε απόσταση +1 (εκτός των διαγώνιων κύβων), οι οποίοι «καταστρέφονται» όταν κάποιος κύβος μετακινηθεί σε κάποιο «κενό» (κενό = κύβος που καταστράφηκε και άφησε πίσω του κενό χώρο) **.**
- **display():** Υλοποιεί την εμφάνιση του Score(Πόντοι),Moves (Κινήσεις) και του μηνύματος «Game Over Final Score: …». Επίσης εμφανίζει όλους τους κύβους στο παράθυρο μας καθώς και κάθε μια αλλαγή που γίνεται από τις καταστροφές, τις αλλαγές των κύβων κτλπ.

  ΠΡΟΒΛΗΜΑ!!!
  Το πρόβλημα που αντιμετώπισα είναι ότι, δεν μπόρεσα να φορτώσω σωστά τα αρχεία των εικόνων (rock.pgm, paper.pgm και scissors.pgm) με αποτέλεσμα να έχω ένα πλέγμα από κύβους μόνο με χρώματα. Για την αποφυγή της σύγχυσης έχω βάλει τα εξής χρώματα για τους εξής τύπους.

  Πράσινο == **Ψαλίδι**
  Κίτρινο == **Χαρτί**
  Μωβ(magenta) == **Βράχος**

  Επομένως: ο πράσινος κύβος καταστρέφει τον κίτρινο κύβο
  , ο κίτρινος κύβος καταστρέφει τον μωβ κύβο
  και ο μωβ κύβος καταστρέφει τον πράσινο κύβο.

  Άρα με αυτόν τον τρόπο γίνονται και οι καταστροφές των κύβων στο παράθυρο, σεβόμενος πάντα την εκφώνηση της άσκησης για τις καταστροφές και τις αποστάσεις, καθώς και την γειτνίαση των κύβων.

Τέλος οι παράμετροι για το compile είναι: -lglut -lGL -lGLU -lstdc++