

# Υπολογιστική Γεωμετρία

## Εργασία 3: προγραμματιστική

### Άσκηση 1

Στο φάκελο “CH3D1” θα βρείτε το αρχείο με όνομα “CH3D\_1.cpp” το οποίο περιλαμβάνει τον κώδικα του πρώτου μέρους της εργασίας 3. Για τη χρήση της γεννήτριας τυχαίων αριθμών της CGAL αρκεί η χρήση του flag `-generate <n>`, όπου  $n > 0$ , κατά την εκτέλεση του προγράμματος. Εάν το πρόγραμμα κληθεί με  $n = 0$  ή χωρίς το συγκεκριμένο flag, τότε τα σημεία εισάγονται από το `standar input` από το πληκτρολόγιο ή με ανακατεύθυνση αρχείου.

Η ακτίνα της σφαίρας από την οποία εξάγονται τυχαία σημεία έχει τυχαία ακτίνα με εύρος  $(0,1000]$ . Για τον υπολογισμό του Convex Hull χρησιμοποιείται η συνάρτηση `“CGAL::convex_hull_d_to_polyhedron_3(CH,P)”` η οποία είναι βασισμένη στον αλγόριθμο που ζητείται.

Τα αποτελέσματα τυπώνονται στο terminal.

### Άσκηση 2

Στο αρχείο “BeneathBeyond.cpp” του φακέλου “CH3D2” θα βρείτε τον κώδικα για το δεύτερο μέρος της εργασίας. Η εισαγωγή σημείων στον τρισδιάστατο χώρο ακολουθεί τη λογική του πρώτου μέρους (Άσκηση 1). Έπειτα, δημιουργείται μια οντότητα κλάσης `“Build_polyhedron<HalfedgeDS>”` με την οποία γίνεται και ο υπολογισμός του Convex Hull.

Η κλάση περιλαμβάνει 5 συναρτήσεις και 5 μέλη.

Μέλη:

- Points

Ο vector όλων των σημείων που δόθηκαν ή παράχθηκαν από το πρόγραμμα.

- Index

Βοηθάει στην καταμέτρηση των παραπάνω σημείων ώστε να γίνει σειριακή προσπέλασή τους.

- AbsIndex

Δίνει το id των κορυφών του Convex Hull εφόσον χρησιμοποιείται `ABSOLUTE_INDEXING`.

- Scale

Η ακτίνα της σφαίρας στην οποία ανήκουν τα σημεία ώστε να γίνει σωστά η γραφική τους αναπαράσταση εφόσον χρειαστεί.

- iPoints

Αντιστοίχιση κάθε σημείου που αποτελεί κορυφή με τον ανάλογο index (id).

Συναρτήσεις:

- Constructor
- Draw

Μπορεί να κληθεί για να γίνει γραφική αναπαράσταση των σημείων της οντότητας hds στο χώρο.

- printHDS

Τυπώνει τις κορυφές της οντότητας hds ανά έδρα.

- operator()

Αρχικά κατασκευάζεται ένα τετράεδρο με χρήση incremental builder ως πρώτη μορφή του πολυέδρου Convex Hull που θέλουμε να φτιάξουμε. Με επανάληψη εξετάζουμε κάθε νέο σημείο του vector Points. Για κάθε έδρα βρίσκουμε αν αφήνει το νέο σημείο σε διαφορετικό επίπεδο σε σχέση με κορυφή του παρόντος Convex Hull εκτός της έδρας αυτής. Αν ισχύει, καταχωρείται στις κόκκινες έδρες (Red) μαζί με κάθε κορυφή της, αλλιώς καταχωρείται στις μπλε (Blue). Βρίσκουμε τις βυσσινί κορυφές (Purple\_V) ελέγχοντας αν υπάρχουν κοινές κορυφές στους vectors Red\_V, Blue\_V και βρίσκουμε τις κόκκινες κορυφές που πρέπει να διαγραφούν. Ύστερα, διαγράφονται κόκκινες έδρες και κορυφές και καλείται νέος incremental builder για να χτίσει στην ήδη υπάρχουσα επιφάνεια. Οι παλιές κορυφές εισάγονται στις νέες έδρες ακολουθώντας την ανάποδη σειρά από εκείνη που έχουν στις παλιές έδρες με τις οποίες γειτονεύουν και στο τέλος εισάγεται και το νέο σημείο, αν έχουμε βρει κόκκινες έδρες.

Στην υλοποίηση αυτή η CGAL εντοπίζει πρόβλημα κατά το χτίσιμο νέων εδρών.

Κατά τη δοκιμή διαφορετικής υλοποίησης στην οποία αποθηκεύονται halfedges αντί για κορυφές ώστε να εντοπιστούν οι μωβ ακμές και με προσεκτική διαγραφή κάθε καθαρά κόκκινης halfedge και κορυφής το αποτέλεσμα ήταν να διαγραφούν όλες οι κορυφές και να μην υπάρχει πολυέδρο για χτίσιμο.