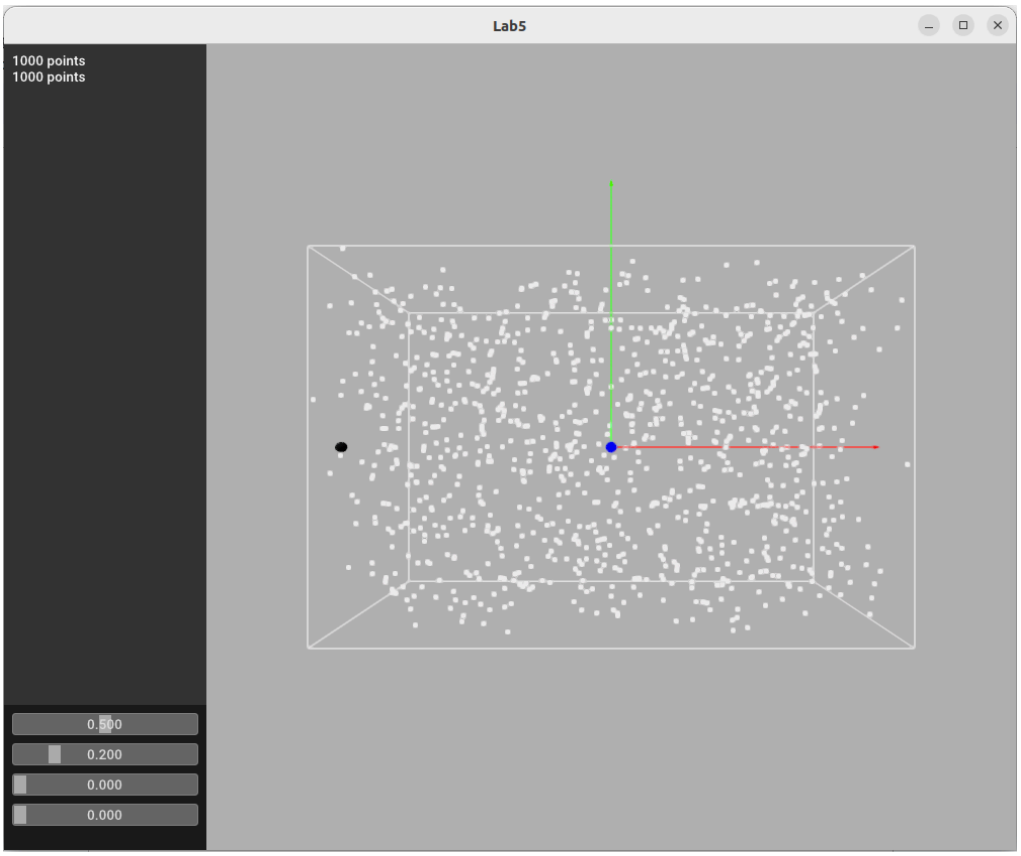
3Δ Υπολογιστική Γεωμετρία

Εργαστηριακή Άσκηση 5

Προετοιμασία Άσκησης:

Τρέξτε το αρχείο lab5.py.
Θα πρέπει στην οθόνη σας να δείτε το παρακάτω παράθυρο.



Η τρισδιάστατη σκηνή περιλαμβάνει ένα τρισδιάστατο κύβο, εσωτερικά του οποίου βρίσκεται ένα τυχαίο νέφος σημείων και μία σφαίρα στην αριστερή μεριά του κύβου.
Πατώντας το πλήκτρο ***enter*** η σφαίρα ξεκινάει να κινείται στο χώρο.
Στη γραφική διεπαφή θα παρατηρήσετε ότι υπάρχουν 4 sliders, οι sliders αυτοί είναι για να ρυθμίζουν από πάνω προς τα κάτω τις παραμέτρους:

1. Ταχύτητα της σφαίρας
2. Αριθμός σημείων
3. Μέγεθος περιβάλλουσας σφαίρας (χρήση σε επόμενο ερώτημα)
4. Αριθμός γειτόνων (χρήση σε επόμενο ερώτημα)

Εξοικειωθείτε με το περιβάλλον της άσκησης.

Άσκηση:

1. Δημιουργήστε ένα kd-tree από ένα νέφος σημείων, συμπληρώνοντας τον constructor της κλάσης KdNode, η οποία κατασκευάζει, με αναδρομικό τρόπο, κόμβους ενός kd-tree.

2. Απεικονίστε με διαφορετικό χρώμα τα σημεία του κάθε υπό-δέντρου του kd-tree δεδομένου του επιπέδου. Για να το κάνετε αυτό θα χρειαστεί να υλοποιήσετε τις παρακάτω συναρτήσεις:
    a. ***getMaxDepth:*** Η συνάρτηση αυτή υπολογίζει, με αναδρομικό τρόπο, το μέγιστο βάθος του kd-tree.
    b. ***getNodesBelow:*** Η συνάρτηση αυτή, με αναδρομικό τρόπο, εντοπίζει και επιστρέφει όσους κόμβους βρίσκονται σε βαθύτερο επίπεδο από τον κόμβο που λαμβάνει ως είσοδο.
    c. ***getNodesAtDepth***: Η συνάρτηση αυτή με αναδρομικό τρόπο, εντοπίζει και επιστρέφει τους κόμβους του kd-tree που βρίσκονται σε κάποιο ορισμένο βάθος.

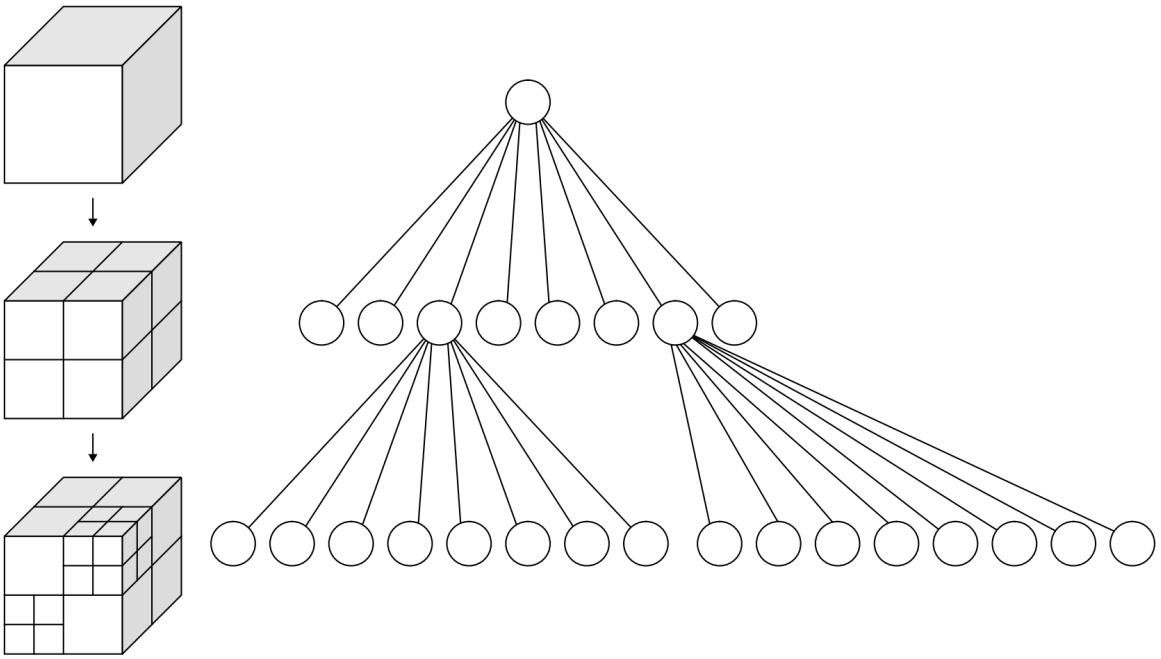
    Ο κώδικας που αναθέτει το χρώμα στα σημεία είναι ήδη υλοποιημένος. Με τη χρήση των ***up*** και ***down arrows*** μπορείτε να επιλέξετε διαφορετικά επίπεδα του kd-tree. Θα παρατηρήσετε επίσης ότι όλα τα pivot σημεία μέχρι και το επιλεγμένο βάθος απεικονίζονται με άσπρο χρώμα.

3. Συμπληρώστε τη συνάρτηση ***inSphere***, η οποία, με αναδρομικό τρόπο, εντοπίζει και επιστρέφει όλα τα σημεία του kd-tree που βρίσκονται εντός της σφαίρας που λαμβάνει ως είσοδο.
4. Συμπληρώστε τη συνάρτηση ***nearestNeighbor***, η οποία, με αναδρομικό τρόπο, εντοπίζει και επιστρέφει το σημείο του kd-tree που είναι κοντινότερο στο κέντρο της σφαίρας.
5. Συμπληρώστε τη συνάρτηση ***nearestK***, η οποία, με αναδρομικό τρόπο, εντοπίζει και επιστρέφει τα k σημεία του kd-tree που είναι κοντινότερα στο κέντρο της σφαίρας.

Homework:

6. Υλοποιήστε τα ερωτήματα 3-5 χωρίς να χρησιμοποιήσετε το kd-tree. Συγκρίνετε το χρόνο εκτέλεσης των συναρτήσεων και παρουσιάστε τα αποτελέσματά σας για διάφορα πλήθη σημείων. Εξηγείστε τι παρατηρείτε.
7. (Bonus/Προαιρετικό) Μία δομή δεδομένων που μοιάζει με τα kd-trees είναι τα octrees (ή quadtrees στην περίπτωση 2 διαστάσεων). Τα octrees απλοποιούν πολύ τη διαδικασία προσθήκης και αφαίρεσης κόμβων, όμως καθυστερούν σε ορισμένα ερωτήματα (π.χ. εύρεση κοντινότερου γείτονα) κάποιες φορές. Το κύριο χαρακτηριστικό τους είναι ότι ο άξονας με βάση τον οποίο τέμνουμε κάθε φορά το δέντρο δεν εξαρτάται από το πλήθος των σημείων, αλλά είναι σταθερός, έτσι ώστε να διαμερίζει στη μέση το χώρο και όχι τα σημεία. Συνήθως, δε, ομαδοποιούμε ανά τρεις τις διαμερίσεις έτσι ώστε κάθε κόμβος να έχει 8 παιδιά (εξ ου και το όνομα), καθένα από τα οποία αναπαριστά στο χώρο ένα ορθογώνιο παραλληλεπίπεδο όμοιο με αλλά μικρότερο από το αρχικό bounding box των σημείων. Εάν, κατά τη δημιουργία του

δέντρου, κάποιο παιδί αναπαριστά άδειο μέρος του χώρου (δεν περιέχει σημεία), αυτό κλαδεύεται και η διαμέριση σταματά για αυτό το υποδέντρο.



(πηγή: WhiteTimberwolf, Wikimedia Commons)

Ζητείται να υλοποιήσετε τη δομή δεδομένων octree και να δείξετε τα bounding boxes των επιπέδων, χρησιμοποιώντας ως είσοδο τις κορυφές ενός mesh της αρεσκείας σας (π.χ. Stanford Bunny).