

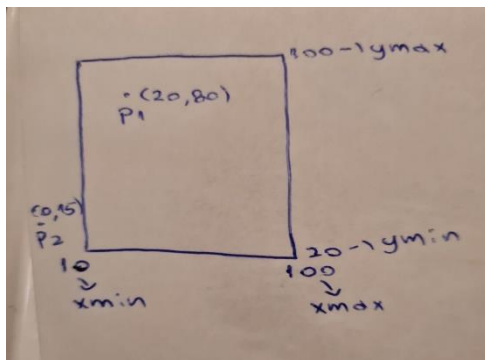
Μέρος Ε Αναφορά

Παντελίδης Ιπποκράτης ρ

Τανέρ Ιμάμ ρ3200057

a) Ανάλυση μεθόδων contains, intersects, distanceTo της κλάσης Rectangle.

- contains: Η μέθοδος αυτή δέχεται σαν όρισμα ένα Point και βλέπει αν αυτό βρίσκεται μέσα στο Rectangle. Πιο συγκεκριμένα ελέγχει αν η x συντεταγμένη του Point είναι ανάμεσα στις xmin και xmax συντεταγμένες (ή και ίσες) του Rectangle και αν ταυτόχρονα αν η y του Point είναι ανάμεσα στις ymin και ymax συντεταγμένες (ή και ίσες) του Rectangle. Αν ισχύουν αυτές οι συνθήκες τότε το Point βρίσκεται μέσα στο Rectangle ή στα όρια του, αλλιώς βρίσκεται εκτός αυτού. Ακολουθεί και μια σχηματική παρουσίαση της μεθόδου. Έστω ότι έχουμε τα σημεία $p1 \rightarrow (20,80)$ και $p2 \rightarrow (0,15)$ και το Rectangle $rec \rightarrow [10,100] \times [20,100]$. Παρατηρούμε και από την παρακάτω εικόνα ότι για να περιέχεται το point μέσα στο Rectangle πρέπει να ισχύουν οι συντεταγμένες που περιγράψαμε παραπάνω.



- intersects: Η μέθοδος αυτή δέχεται σαν όρισμα ένα Rectangle και ελέγχει αν αυτό έχει κοινό σημείο με το Rectangle της κλήσης. Για να το πέτυχει αυτό ορίζει τις κορυφές του ορθογωνίου κλήσης και αυτού που περνιέται σαν όρισμα και βλέπει αν κάποια από τις κορυφές του πρώτου περιέχονται (μέσω της contains) στο δεύτερο ορθογώνιο καθώς και αν κάποια από τις κορυφές του δεύτερου περιέχονται (μέσω της contains) στο πρώτο ορθογώνιο.

Ακολουθεί σχηματική αναπαράσταση για καλύτερη κατανόηση της μεθόδου.

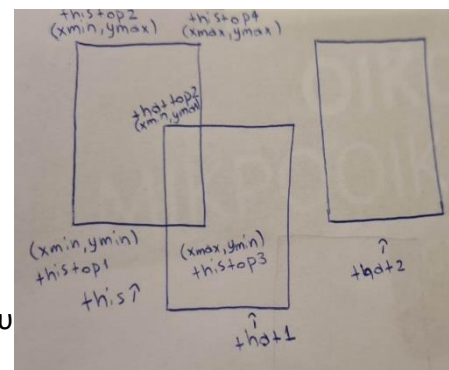
Παρατηρούμε λοιπόν ότι για να τέμνονται δυο ορθογώνια θα πρέπει κάποια από την κορυφή του ενός να εμπεριέχεται στο άλλο, όπως στο παράδειγμα συμβαίνει με το this και that Rectangle.

- distanceTo: Η μέθοδος αυτή δέχεται σαν όρισμα ένα Point

και επιστρέφει την απόσταση του από κοντινότερο σημείο του Rectangle.

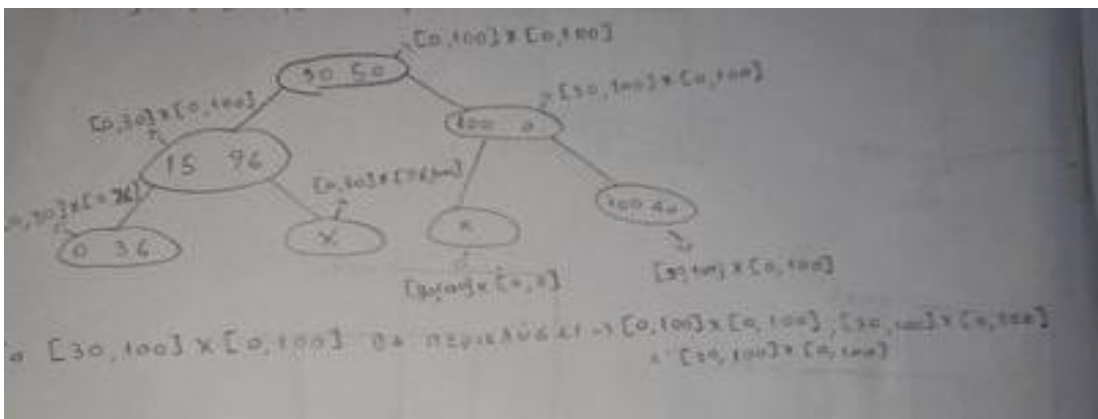
Αρχικά βλέπουμε αν το point περιέχεται στο Rectangle \rightarrow απόσταση 0

Αν το point βρίσκεται είτε στο άξονα x ή y τότε υπολογίζει την απόσταση με μια απλή αφαίρεση, ενώ αν το point είναι τελείως εκτός του ορθογωνίου υπολογίζει κάθε απόσταση (distanceTo) από κάθε κορυφή και βρίσκει την μικροτερη



b) Ανάλυση μεθόδων nearestNeighbour και rangeSearch της TreeNode:

- rangeSearch: Η μέθοδος αυτή δέχεται σαν όρισμα ένα Rectangle και ελέγχει ποια points περιέχονται σε αυτήν την επιφάνεια. Αρχικά δημιουργεί μια λίστα στην οποία θα μπουν αυτά τα points χρησιμοποιώντας την υλοποιημένη από την 1^η εργασία ουρά και καλεί την μέθοδο rangeSearchRecursive, η οποία θα καλείτε αναδρομικά μέχρι να δώσει τα απαιτούμενα αποτελέσματα. Η μέθοδος αυτή παίρνει σαν όρισμα ένα αντικείμενο Tree Node cur, το οποίο δείχνει σε ποιο κόμβο του δέντρου βρισκόμαστε, Rectangles το πρώτο είναι το παραλληλόγραμμο του cur και το δεύτερο το όρισμα της αρχικής rangeSearch. Επίσης η μέθοδος παίρνει την λίστα καθώς και το επίπεδο στο οποίο βρισκόμαστε πάνω στο δέντρο. Η πρώτη κλήση περνάει τις τιμές root, δηλαδή την ρίζα του δέντρου, το Rectangle $[0,100] \times [0,100]$ καθώς από τα hints ξέρουμε ότι η ρίζα έχει αυτό το παραλληλόγραμμο, το όρισμα του χρήστη, την άδεια ουρά που φτιάξαμε καθώς και το επίπεδο 0, δηλαδή αυτό της ρίζας. Στην αναδρομική συνάρτηση δημιουργούμε σε κάθε κλήση ένα Tree Node όπου θα είναι ο κόμβος στον οποίο βρισκόμαστε και δυο Rectangles όπου θα είναι τα νοητά παρ//μμο του δεξιού και αριστερού παιδιού. Αν το επίπεδο μας είναι 0 δηλαδή βρισκόμαστε στην ρίζα τότε φτιάχνουμε το δεξί και αριστερό παρ//μμο ως εξής: γ μένουν αμετάβλητα, αριστερό παρ//μμο παίρνει για xmax το πεδίο x του cur ενώ το δεξί το παίρνει ως xmin. Αν βρισκόμαστε σε περιττό επίπεδο τότε τα παρ//μμο διαμορφώνονται ως εξής: εδώ τα x μένουν αμετάβλητα (έχουμε προχωρήσει στην αναδρομή επομένως σαν δεύτερο όρισμα μπαίνει το αριστερό παιδί αν πάμε αριστερά αλλιώς το δεξί αν πάμε δεξιά), αριστερό παρ//μμο παίρνει για ymax το πεδίο y του cur ενώ το δεξί το παίρνει ως ymin. Τέλος στα άρτια επίπεδα τα y περνιούνται από τον γονέα και δεν αλλάζουν, ενώ στο πάλι στο αριστερό παρ//μμο σαν xmax μπαίνει το πεδίο x του current, και στο xmin στο δεξί. Ο λόγος όμως που δουλεύει η αναδρομή είναι το σωστό πέρασμα παραμέτρων στην αναδρομική κλήση χρήση του current rectangle μας κρατάει τις τιμές του πατέρα των νοητών παρ//μμων και έτσι τις περνάμε απλά στα σημεία όπου οι μεταβλητές μένουν αμετάβλητες. Πριν από κάθε αναδρομική κλήση ελέγχουμε αρχικά αν το point του current περιέχεται στο δοσμένο Rectangle καθώς και αν αυτό τέμνεται με κάποιο από τα δυο νοητά παρ//μμο του current, και αν ναι εισάγουμε το point στην ουρά. Έπειτα βλέπουμε αν ο current έχει αριστερό ή δεξιό παιδί και αφού βεβαιωθούμε ότι το υποδέντρο μπορεί να περιέχει σημεία που ανήκουν στο παραλληλόγραμμο συνεχίζουμε την αναδρομή στο δεξί και αριστερό παιδί αυξάνοντας το επίπεδο κατά ένα στην κλήση. Στο παράδειγμα μας η range Search δημιουργεί τα νοητά Rectangles και δουλεύει σύμφωνα



- nearestNeighbour: Η nearestNeighbour έχει όπως και η range Search μια αναδρομική μέθοδο στην οποία περνάει σαν ορίσματα στην πρώτη κλήση: τον κόμβο της ρίζας από όπου ξεκινάμε, το Point που δίνει ο χρήστης, μια μεταβλητή μικρότερης απόστασης του point από το Rectangle που έχει αρχικά σαν τιμή μια αρκετά μεγάλη ποσότητα εκτός ορίων, το Rectangle, το επίπεδο της ρίζας και το nearest Point που αρχικά είναι η ρίζα. Η αναδρομική μέθοδος αυτή βλέπει σε τι επίπεδο βρισκόμαστε και φτιάχνει όπως και η rangeSearch τα νοητά παραλληλόγραμμα των παιδιών του current κόμβου. Αν είμαστε σε άρτιο level βλέπει αν υπάρχει δεξί και αριστερό παιδί και σε περίπτωση που η απόσταση του παιδιού από το δοσμένο point είναι μικρότερη από την απόσταση του τωρινού κοντινότερου σημείου από το p καθώς και αν η απόσταση του p είναι από το αριστερό ή δεξί αντίστοιχα παρ//μμο είναι μικρότερη από την απόσταση του point από το προηγούμενο νοητό παρ//μμο τότε καλούμε αναδρομικά την μέθοδο αυξάνοντας το level κατά 1 και πηγαίνοντας πάλι αντίστοιχα αριστερά και δεξιά. Όταν είμαστε σε περιττό επίπεδο εργαζόμαστε αναλόγως απλά με Rectangles τα οποία έχουν οριστεί διαφορετικά, όπως έχουμε δει.