

Quadtree ile Mouse Tiklamalarının Modellenmesi ve Dairesel Aralık Sorgulama

Proje İlan Tarihi17 Mart 2017
Proje Teslim Tarihi:7 Nisan 2017

1 Giriş

İkili bir arama ağacı (binary search tree-BST), bir boyutta doğrusal aramaya (linear search) kıyasla daha hızlı arama yapmayı mümkün kıldığı için dörtlu ağac (quadtree), iki boyutta daha hızlı arama sağlar. Dörtlu ağac yapısal olarak BST'ye benzemektedir. Her nodun (node) dört adet çocuğu (child) vardır. ki boyutlu uzayda kullanılmaktadır. Quadtree nin farklı versiyonları vardır (region quadtree, point quadtree, point-region quadtree, edge quadtree vs.)

Dörtlu ağaçlar; bilgisayar grafikleri ve oyun içindeki nesnelerin birbirleriyle çarpışmalarının testinden tutun da görüntü işleme uygulamalarına, insan-bilgisayar etkileşiminden coğrafi bilgi sistemlerine kadar birçok alanda kullanılmaktadır.

2 Ödev

2.1 Problem

Bilgisayar ekranında mouse ile tiklanan yerlerin bir nokta topluluğu ile gösterildiğini varsayalım. Yeni bir mouse tiklaması ile dairesele bir sorgu yapıldığı zaman sorunun var olan noktalardan hangisi(leri) üzerine denk geldiğini/çakıştığını nasıl anlayabiliriz? Burada sorgu, bir bölgeye karşılık gelmektedir. Örneğin aşağıdaki gibi point-quadtree mantığında oluşturulmuş bir bölge verilmiş olsun. B civarında bir sorgu (siyah daire) olduğunu varsayalım. Ağacın root (A) dan başlanarak arama yapılır. Daire, A'nın dikdörtgeni (tüm çizim alanı) ile kesişmediği için A'nın çocuklarına (D, E, B, ve K) bakılır. Daire sadece B'nin dikdörtgeni

ile kesiliyor ve B noktası dairenin içinde olduğu için çakışma vardır. Rekursif olarak B'nin çocukları (yalnızca C var) ile aramaya devam edilir. Daire C dikdörtgenini kesmesine rağmen C noktası daire içine düşmediğinden C noktası ile çakışma yoktur. Böylece arama sonlanmış olur. Benzer şekilde G civarındaki bir sorguda (kırmızı daire) sadece G, A civarındaki bir sorguda (mavi daire) ise A ve L noktaları ile çakışma vardır (Fig 1'e bakınız).

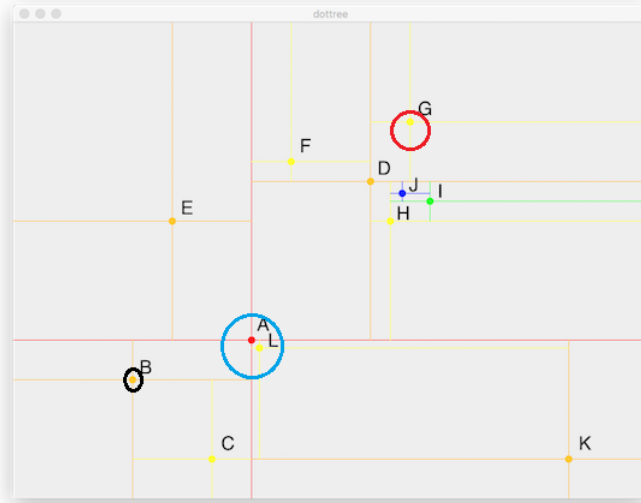


Fig. 1. Örnek Point Quadtree

2.2 Implementasyon Detayları

1. 512x512 lik grafiksel kullanıcı arayüzde (GUI) nokta toplulukları, hem kullanıcının mouse yardımıyla belirlemesi şeklinde (yukarıda A, B, C, K, L gösterildiği gibi) hem de random olarak arayüzde belirlenmelidir.
2. Nokta setlerine göre dörtlu ağac yapılarından biri point quadtree kullanılarak ağac yapısı oluşturulur, arayüzde quadrant'lar gösterilir.
3. Arayüzden sorgu dairesinin yarı çapına müdahale edilebilmelidir (büyütme-küçültme).
4. Arayüzdeki noktalar silinip tekrar random olarak veya kullanıcı vasıtasıyla ağac oluşturulabilmelidir.

5. Yapılan sorgu içine düşen nokta(ların) rengi değiştirilmelidir (eger çakışma varsa). Çakışmalar yukarıda anlatıldığı gibi oluşturulan dörtlü ağac üzerinde arama işlemine karşılık gelmektedir. Sorgu içine düşen noktaların sayısı ve noktaların kendisi önce x sonra y değerlerine göre sıralı olarak arayüzde listelenmelidir. Sorgular oluşturulan dörtlü ağac üzerinde yapılacaktır.

6. Yukarıdaki istekler nesneye dayalı programlama mantığıyla farklı sınıflar kullanılarak yerine getirilmelidir.

2.3 Önemli Kurallar

1. Ödevlerinizi bireysel olarak kendiniz yapmalısınız. Gönderilen tüm kodlar otomatik intihal algılama yazılımı sistemi (Moss, JPlag, vb.) kullanılarak karşılaştırılacaktır.

2. Nesneye yönelik programlama becerisinin geliştirilmesi için projeler; C++, C sharp veya Java kullanılarak gerçekleştirilmelidir.

3. Programınız tutarlı bir kodlama stiline (girintiler, yorum satırları, geçerli değişken isimlendirmelerine) sahip olmalıdır. Sınıf tanımlamaları başında sınıfın ne iş yaradığı açıklanmalı, metotlardan önce metodun ne iş yaptığı ve parametrelerin ne anlama geldiği aşağıdaki formatta verilmelidir.

```
/**
 * Daire ile dikdörtgen kesişiyor mu
 * @param dx daire merkez
 * @param dy daire merkez
 * @param r daire yarı çap
 * @param kose_x1 kose min x
 * @param kose_y1 kose min y
 * @param kose_x2 kose max x
 * @param kose_y2 kose max y
 */
public static boolean kesisim(double dx, double dy, double r,
    double kose_x1, double kose_y1, double kose_x2, double kose_y2)
{
    ....
}
```

2.4 Raporda olması gerekenler

1. Problemlere çözümler yalancı kod ve şekillerle açıklanmalıdır.

2. Sınıflara ait UML class diyagramları eklenmelidir. Kalitim, interface olması durumları diyagramlar ilişkilendirilerek belirtilmelidir.

3. Farklı çalışma senaryoları şekillerle açıklanmalıdır.

4. Yararlanılan kaynaklar belirtilmelidir.

2.5 Puanlama

-İsterlerin yerine getirilmesi 70p

Quadtree ye eleman ekleme (mouse random) 20p

Arayuzde cizgilerle quadrant lari gosterme 15p

Sorgu ile kesisen noktalarin adedinin bulunmasi, noktalarin siralanmasi 15p

Sorgu ile kesisen noktalarin renklendirilmesi 10p

Sorgu daire boyutunu degistirme, arayuzden noktalar temizleme 10p

-Kodlama stillerini yerine getirme 10p

-Raporlama 20p

Not: Proje ile ilgili sorularinizi piazza uzerinden sorabilir, paylasilacak dokumanlara ulasabilirsiniz. Derse ait sayfa (access code blm210) https://piazza.com/kocaeli_university/spring2017/blm210

Proje Sorumlulari: Ars.Gor. Suleyman EKEN ve Ars.Gor. Furkan GOZ