

1) Oyun programlama aşamaları nelerdir? Yazınız. (10-20 puan)

1) Konsept = İçerik = Senaryo oluşturmak

Örneğin; Metanlar olur, modelleri. Senaryo en az 5 sayfa olur.

Bilgisayarda modelimizi yapılar 3 boyutludur. (x, y ve z)

Senaryoda ortam, karakterler ve hareketler olmalıdır.

2) Prototip Oluşturma (Herhangi bir ortamda - Oyuna göre)

En az 2 ya da 3 hafta sürer.

Oyunla ilgili detay olmasının, oyunun oluşturulmasıdır.

3) İlk Üretim Aşaması

6 ay ya da 12 ay sürer.

Oyun geliştirilmesinin en önemli aşamasıdır. Kurulum araçları tespit edilir,

bu oyun için gerekli olan amaçlar ve süreçler tanımlanır. Oyunun

trial (ücretli) versiyonudur. Bu aşamada oyunun durum geçiş diyagramı çıkarılır.

4) Üretim Aşaması Oluşturma

En az 20 kişilik bir ekip olmalı

6/12 ay sürer

Bütün işlerin paralel olarak yürütüldüğü aşamadır. Ve bu aşamada, seviyelerin ortaya koyulduğu aşamadır.

5) Test / Ayarlama / Hata Ayıklama

En az 4 tane tester (test edici) kullanılır. Oyunla ilgili hata varsa bulurlar.

Oyunla ilgili bug/error ortaya konulur. Daha sonra oyunun Alfa, Beta, Full Version sürümleri çıkarılır.

6) Porting (Taşıma) : Bir ortama geliştirilen oyunun başka bir ortama taşınabilmesi işlemidir.

7) Localization : Bir dilde geliştirilen oyunun, o oyunun satıldığı bölgedeki kullanılan dillere otomatik olarak dönüştürülebilirliği demektir.

2)



Çemberlerin üçgenin içinde mi? dışında mı? Üzerinde mi? Olduğunu matematiksel olarak veriniz ya da algoritmasını yazınız.

Belirttiğimiz herhangi bir nokta üçgenin içinde mi? değil mi? nasıl buluruz?

1) Vektörel hesap yapma

2) Tarama algoritması kullanarak (scanning)

1. yöntem ile;



Bir nokta olsun.

Vektörlerimizi aldık. İki vektörün çarpımı yine bir vektöre eşittir.

$$\vec{A} \times \vec{B} = \vec{C}$$

$$\begin{matrix} i & j & k \\ A_x & A_y & A_z \\ B_x & B_y & B_z \end{matrix}$$

$$\begin{matrix} \vec{A} & + & \vec{A} & + & \vec{A} \\ A_x i & + & A_y j & + & A_z k \\ \vec{B} & + & \vec{B} & + & \vec{B} \\ B_x i & + & B_y j & + & B_z k \end{matrix}$$

}

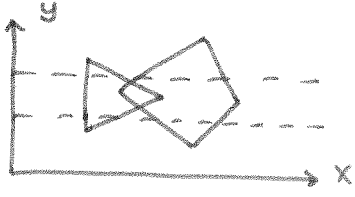
2 vektör alan içinde olmak şartı ile $\vec{A} \times \vec{B} = \vec{C}$ dışarı doğru ise o zaman nokta içindedir.
↓
vektör dışarı doğru demek

Neve, noktasal ise; vektörel hesap yeterlidir. Nerne sınırları eğriyel ise, yani bir çember içine oturmuşsa o zaman, yarıçap merkezi aynı zaman yeterlidir.

2. yöntem ile;



→ Bunun için yukarıdan aşağı, aşağıdan yukarı neye taramın.



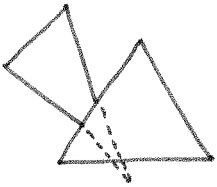
}

Yandaki şekilde görüldüğü üzere, iki doğru taraması yapılmış ve bu doğruların her iki yüzeyleri görülen nokta nokta olarak kodlanmıştır.

Algoritma aşağıdaki adımlardan oluşmaktadır:

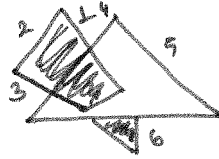
1. Yatay hatlarda tarama yapıldığı için x değeri artırılarak parçalanması gereken kenarları içeren bir liste hazırlanır. (verilen tarama doğrultusu için)
2. Tarama sırasında her yüzey için (surface) bir bayrak tanımlanarak, yüzeyin tarama hattından geçilip geçilmediği belirlenir.
3. Soldan sağa doğru imarek (pixel) konumları ilerlenerek girilür durumu yüzeyler belirlenir.

İki üçgenin kesişip, kesişmediğini matematiksel olarak yazınız ya da algoritmasını yazınız.



① Üçgenlerin izdüşümleri alınır. (yani 2 boyuta dönüştürülür)

2 üçgenin izdüşümünü alırsak;

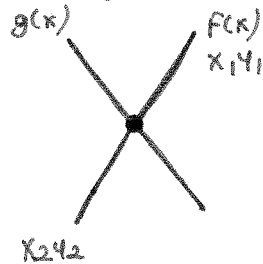


Artık bu iki boyutta olmuştuk oldu.

② Bu nesne bunu kesiyor mu? kesmiyor mu? (2 boyutta bakıyoruz.)

1. Üçgene ait 3 çizgi var. $3+3=6$
2. " " " " 4 } kombinezyon yaparız.

③ Bizim aşağıdaki gibi kesişen 2 doğrumuz olsun. $f(x)$ ve $g(x)$ doğruları.



$$\frac{x-x_1}{x_2-x_1} = \frac{y-y_1}{y_2-y_1}$$

İki kesişen doğru için denklem budur. Eğer bir denklem çıkmazsa kesişmez.

4) Veriyapıları oyunlarda nerede kullanılır? (30 puan)

(3)

1) Oyun içerisinde karakterin hareket ettirdiği ortamlara portal denir. Portallar arasındaki bağlantı, graf (ağ) 'lar ile gerçekleştirilir.

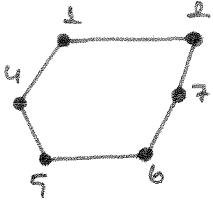
2) Veriyapılarındaki ağaç veri yapısı, bilgisayar bilimlerinde önemli yer tutar. Yazılım dünyasında birçok yerde programcının karsısına çıkar. Oyunların olası hamleleri için veriyapıları kullanılır.

Örneğin; bir oyunda oyuncuların aldığı en yüksek skorların tutulacağı bir veriyapısı düşünelim.

Oyuncu adı	Skor
Bilal	1453

Bunun için dizileri kullanırız. Kayıtlar, diziyeye sıralı olarak eklenir. Sıralamaya girme-ye hak kazanan oyuncuya ait kayıt doğru bulunarak yerleştirilir.

3)

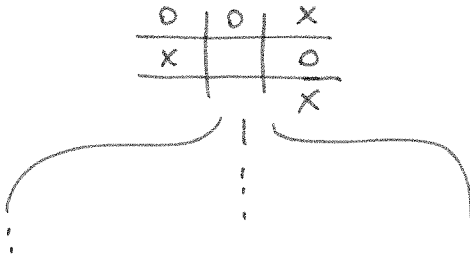


Örneğin; yandaki graf, bir oyundaki portallar arasındaki bağlantıyı gösterir. Eğer 2 nolu portaldaysam, hafızaya yüklenecek olan veriler 2 ya da 4'dedir. Ekleme - çıkarma işlemlerini ise veriyapılarındaki yığın (stack) kullanarak gerçekleştiririz.

4) Birisinin kaybının, başka birisinin kazancı olduğu (veya tam tersi) oyunlarda, veriyapılarındaki min/max tree yapılarını kullanmak oldukça kullanışlıdır. karar vermek için

Örneğin; satranç oyunu gibi. Bu oyunlarda, doğru karar verilmesini sağlayan min/max tree'ler ağacı minimize edip, kazancı maximize etmeye çalışır.

Örneğin; XOX oyununu ele alalım. Oyunun durumuna, göre mevcut durumdan yorulanlar olarak, bir karar ağacı çıkartmak mümkündür.



5) Sayı, karakter ve sözcük sınıflandırma yapan bir oyun düşünelim. Yine bu durumda veriyapılarındaki yığın (stack) kullanılabilir.

6) Aşağıdaki tanımları yapınız (20 puan)

Fizik Motoru: Video oyunlarında ve simülasyon yazılımlarında kullanılan, 4ü boyutlu nesnelerin geometrilerinin kesim testleri ve bu gibi fiziksel etkileşimlerini test eden bir yazılımdır. içinde birçok fizik formülü tanımlıdır.

Oyun Motoru: Kişilerin veya şirketlerin oyun yapmak amacıyla kullandığı ücretli veya ücretsiz olan programlara verilen isimdir. Bu program, esasında kütüphanelerden oluşan bir yapıdır.

Çarpışma Tespiti: Matematiksel olarak çarpışma tespiti, kesim bulma problemi adımlıdır.

CUDA: NVIDIA'nın GPU (grafik işlem birimi) gücünü kullanarak, hesaplama performansında büyük ölçüde artışlara olanak veren paralel hesaplama mimarisidir.

STREAM: Akış anlamına gelir. Örneğin; bir video kameradan bilgisayara gönderilen aktarılan video streaming yani görüntü aktarımı yapmış oluruz.