Karadeniz Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü
Öğr.Gör. Ömer ÇAKIR

BIL 409 Bilgisayar Grafikleri Lab.

Final Sınavı, 07.01.2013, 13:00, D1-D2
Süre : **100** Dakika

NUMARA : İMZA :	DEĞERLENDİRME				
AD SOYAD :	1 →	2 →	3 →	4 →	Toplam →

```
fCurrSampledHeight = NormalHeightMap.SampleGrad
      (samLinear, IN.texcoord+vCurrOffset, dx, dy ).a;

while ( fCurrSampledHeight < fCurrRayHeight )
{
  fCurrRayHeight -= fStepSize;
  vCurrOffset += fStepSize * vMaxOffset;

  fCurrSampledHeight = NormalHeightMap.SampleGrad
      (samLinear, IN.texcoord+vCurrOffset, dx, dy ).a;
}
```

1. **a)** Yukarıdaki kodu kısaca açıklayınız. **(15P)**

while döngüsünden önceki satırda **SampleGrad** fonksiyonu **IN.texcoord + vCurrOffset** parametresi ile çağrılmıştır. Burada **IN.texcoord** doku koordinatına eklenen **vCurrOffset** doku üzerinde ilerlemede kullanılan doğrultu vektörüdür. **SampleGrad** fonksiyonunun sonundaki **.a** ifadesi ile **NormalHeightMap** isimli dokudan **fCurrSampledHeight** a*lpha* değeri okunur. Bu değer ile, bakış noktasından yollanan **eye** ışınının, başlangıçta **1**'e setlenmiş yüksekliği **fCurrRayHeight** karşılaştırılır.
fCurrSampledHeight < fCurrRayHeight yani okunan alpha, ışının yüksekliğinden küçük olduğu müddetçe **fCurrRayHeight** değeri **fStepSize** kadar azaltılır. **fStepSize * vMaxOffset** ile **vCurrOffset** vektörü güncellenerek yeni **fCurrSampledHeight** değeri okunur. Dokudan okunan alpha değeri ışının yüksekliğinden **>=** olduğunda döngüden çıkılır.

b) Aşağıda verilen bump/parallax mapping modlarının nasıl gerçeklenebileceğini `N, vFinalNormal, fParallaxLimit` değişkenleri üzerinden açıklayınız. **(15P)**

"**00**": ne bump mapping ne de parallax mapping var,
"**01**": bump mapping yok ama parallax mapping var,
"**10**": bump mapping var ama parallax mapping yok,

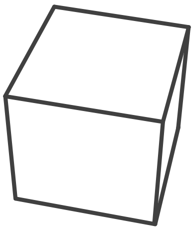
"**00**": `vFinalNormal = N, fParallaxLimit = 0`
"**01**": `vFinalNormal = N`
"**10**": `fParallaxLimit = 0`

2. **a)** Ters perspektif dönüşüm ile düzlemsel yüzey üzerine doku kaplanmak istenmektedir. Bu amaçla aşağıdaki bilgiler doğrultusunda konumu belirtilen ekrandaki noktanın dokudaki karşılığını hesaplayınız. **(15P)**

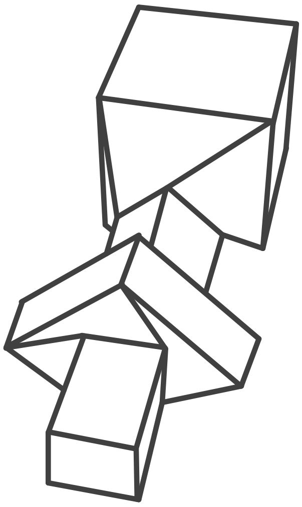
- Ekrandan görülen düzlem y=4x+2 düzlemi
- Gözlemci x ekseni boyunca bakmakta
- Ufuk noktası ekranın ortasından geçmekte
- Herhangi bir eksende rotasyon yapılmamış
- (Xa, Ya, Za)=(1,1,1)
- Dokunun boyutları (7x7)
- Noktanın ekrandaki konumu=(290, 290)
- Ekran çözünürlüğü=(400, 400)
- Gözlemcinin görüntü düzlemine uzaklığı=10

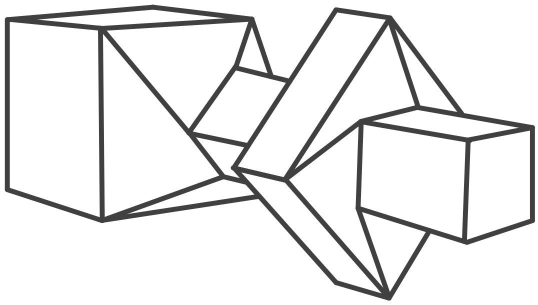
b) Perspektif izdüşüm ile oluşturulan görüntü aksonometrik izdüşüm ile oluşturulmuş görüntüye hangi durumlarda çok benzer, hangi durumlarda çok farklıdır? (Cismin konumunun sabit olduğu düşünülmelidir) **(5P)**

c) Işın izleme yönteminin adımlarını madde madde net olarak açıklayınız. **(10P)**



Küpün ilk hali





1. Move Tool'a tıklanır ve z ekseni boyunca çekilir
2. Rotate Tool'a tıklanır ve z ekseninde döndürülür
3. Scale Tool'a tıklanır ve merkeze doğru scale yapılır
4. Önyüz face olarak şeçili iken Extrude Tool'a tıklanır
5. Scale Tool'a tıklanır ve merkezden dışarı doğru scale yapılır

3. MAYA ortamında bir küpün Extrude, Move, Scale ve Rotate toolları ile yukarıda farklı iki açıdan verilen hale nasıl getirildiğini tanımlamak üzere **1..5** adımlarını sıralayarak yazınız. **(15P)**
İpucu→ Çözüm 15 adımlıdır.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
4	**3**	**2**	**1**	**4**	**1**	**4**	**5**	**4**	**1**	**4**	**3**	**2**	**4**	**1**

| (3,2,1) sıra değişebilir | | (2,3) olur |

4. Aşağıdaki yüzeyi, çekirdek doldurma algoritmasını kullanarak ve işaretli kutucuktan başlayıp, tüm yığına eleman ekleme, yığından eleman çekme, seçili pikseli boyama ve yığındaki eleman sayısını belirtme işlemlerini adım adım yazarak doldurunuz. Yığına eleman ekleme sırası sol, sağ, alt ve üst komşu şeklindedir.
Yığına eleman eklemek için ***yigin.Push(Piksel(x,y))***,
Yığından eleman çekmek için ***yigin.Pop(Piksel(x,y))***,
Seçili pikseli boyamak için ***Boya(Piksel(x,y))***,
Yığındaki eleman sayısı için ***yigin.Count=k***
kullanınız. **(25P)**