

**T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

BSM 421 BİLGİSAYAR GRAFİĞİ

PROJE RAPORU

3D SNAKE GAME

Abdurrahman DOĞAN – B181210394

Oğuzhan TOHUMCU – B181210397

Bölüm : BİLGİSAYAR MÜHENDİSLİĞİ

Ders Hocası : Dr. Gülizar ÇİT

2019-2020 Güz Dönemi

Açıklamalar

Kup.hpp:

- Gerekli glm kütüphaneleri ile vector ve string kütüphaneleri include edildi.
- 3 boyutlu pozisyon ve 2 boyutlu texture vektör koordinatlarını barındıran Vertex struct'ı tanımlandı.
- Kup sınıfında yapılanlar:
 - 4 farklı yön için enum tanımı
 - Kup constructor'ı
 - Yön tayini için setter-getter fonksiyonları
 - Dönüşüm işlemi için setter-getter fonksiyonları
 - Kaplama no fonksiyonu
 - Küpün hareketleri (sağ, sol, yukarı, aşağı) için fonksiyon
 - Küpü oluşturacak bir ana fonksiyon

Kup.cpp:

- 3 boyutlu dönüşüm için **matrix_transform** kütüphanesi eklendi.
- Kup constructor; küpün kenar uzunluğu, hareket yönü, çizileceği pozisyon, dönüş yönü, döndürme, öteleme ve kaplama numarası atandı ve kupOlustur fonksiyonu ile küp oluşturuldu.
- getYon fonksiyonu; küpe ait yön bilgisi alındı.
- setYon fonksiyonu; küpün yön bilgisi belirlendi.
- getPosition fonksiyonu; küpe ait koordinat bilgisi alındı.
- getTransform fonksiyonu; küpe ait dönüşüm bilgisi alındı.
- setTransform fonksiyonu; ilk olarak kamera pozisyonu ayarlanarak **kamera** matrisi, kameranın baktığı yer (orijin) tayini yapılarak **projeksiyon** matrisi ve kamera yönü (y eksenine paralel) belirlenerek **öteleme** matrisi oluşturulması işlemleri yapıldı. Ardından **döndürme** matrisi oluşturuldu. Son olarak bu 4 matrisin çarpılmasıyla **dönüşüm** matrisi elde edildi.
- getKaplamaNo fonksiyonu; küpe ait kaplama no bilgisi alındı.
- getRotation fonksiyonu; küpe ait dönüş bilgisi alındı.
- kupOlustur fonksiyonu; 3 boyutlu küp noktaları belirlenerek, kullanılan 24 farklı koordinatlı **Vertex** objesinde, 6 farklı yüzeydeki 4'er **pozisyon** ve **texture** değerlerinin her birine geçerli koordinat bilgileri atandı. Oluşturulan 24 nokta **vertices** vektörüne kopyalandı ve her yüzeye ait indeks bilgileri **indices** vektörüne eklendi.
- move fonksiyonu; 3 boyutlu yön tayini yapıldı.

ShaderProgram.hpp & ShaderProgram.cpp:

- **Shader** program oluşturma, dosyadan okuyarak elde etme, **link**, **attach**, matris ve vektör fonksiyonları tasarlandı ve gerekli işlemler yapıldı.
- Bu sınıfı hazırlarken hocanın yazdığı kodlardan faydalanılmıştır.

TextureManager.hpp:

- Map ve string kütüphaneleri eklendi.
- TextureManager sınıfında yapılanlar:
 - **Private** olarak bir no parameter constructor
 - Initilaze fonksiyonu,
 - Static bir TextureManger örneği (instance)
 - Kaplama ile kaplamaya ait id ilişkilendirilmesi için bir Map
 - **Public** olarak static örneğe erişim için bir fonksiyon
 - Kaplamayı aktif etme fonksiyonu
 - Kaplamayı yükleyecek fonksiyon
- Tek bir kaplama örneği oluşturulmasını kontrol edebilmek için constructor private scope'a alınarak, static fonksiyon yardımıyla ulaşılmıştır.

TextureManager.cpp:

- m Instance; **static** bir değişken olduğu için ilk değeri global scope'ta atandı.
- activateTexture fonksiyonu; kaplamayı aktive eden ve Map aracılığıyla erişilen dosya ile 2d kaplamayı ilişkilendiren bağlama işlemi yapıldı.
- loadTexture fonksiyonu; kaplama yüklendi ve kaplamaya ait id değerinin geri döndürülmesi sağlandı.
- getInstance fonksiyonu; **m Instance** üzerinden tek bir kaplama örneği oluşturulacak şekilde kontrol yapıldı.

Main.cpp:

- Ciz fonksiyonu; Shader programın açısına göre dönüşüm uygulayan ve kaplamayı aktif ederek çizimi yapan fonksiyon.
- moveSnake fonksiyonu; yılan hareketini uygulandığı genel hareket fonksiyonu.
- addToSnakeTail fonksiyonu; Daha önce hiç oluşturulmuş yılan küpü yoksa oluşturan; varsa yılanın kuyruğuna ekleyen fonksiyon. Bunu yaparken kaplamayı rastgele seçer ve yılanın oluşturulacağı pozisyon, yön ve dönüşüm değerlerini de hesaplar.
- key_callback fonksiyonu; Yön tuşları kombinasyonlarının algılanmasının sağlandığı fonksiyon.
- main fonksiyonu; Program ekranı ayarları, tuş kombinasyonları, kaplama yönetimi, yılan (küp) ekleme, shader bağlamaları, çeşitli gl fonksiyonlarıyla VAO ve VBO objelerinin buffer'a yüklenmesi ile Z-Tamponu ayarlamalarının yapıldığı ana fonksiyon.
- Space tuşu ile yılana yeni hücre eklenmesi, 'A' ve 'D' tuşları ile kameranın yaklaşma ve uzaklaşması sağlandı.

Not: Lütfen programı çalıştırmadan önce kendi GLFW kütüphane dosyanızı Dependencies dizinin altına ekleyiniz ve olası bir hata oluşması durumunda build klasörünü siliniz.