HAKAN DINDIŞ

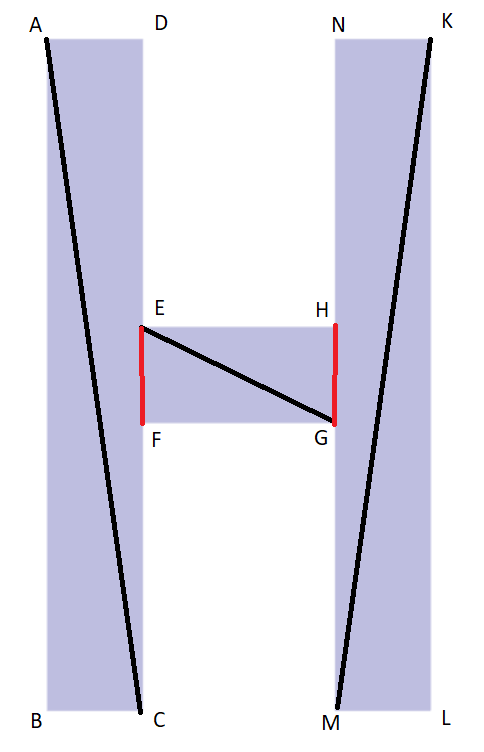
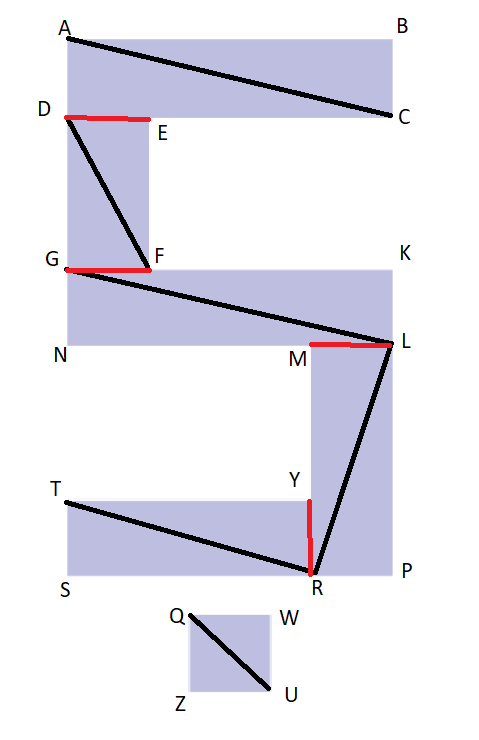
19290234

BLM-3037 BİRİNCİ ÖDEV RAPORU

NOT: <https://hakandindis.github.io/source/19290234.html> linki üzerinden uygulama aktif olarak çalışmaktadır.

HARF ÇİZİM AŞAMALARI

Çizmem gereken harfleri önce dikdörtgenlere, daha sonra da üçgenlere ayırdım.

Devamında bu üçgenleri çizebilmek için noktaların koordinat sistemi üzerindeki gerekli (x,y) noktalarını hesapladım.

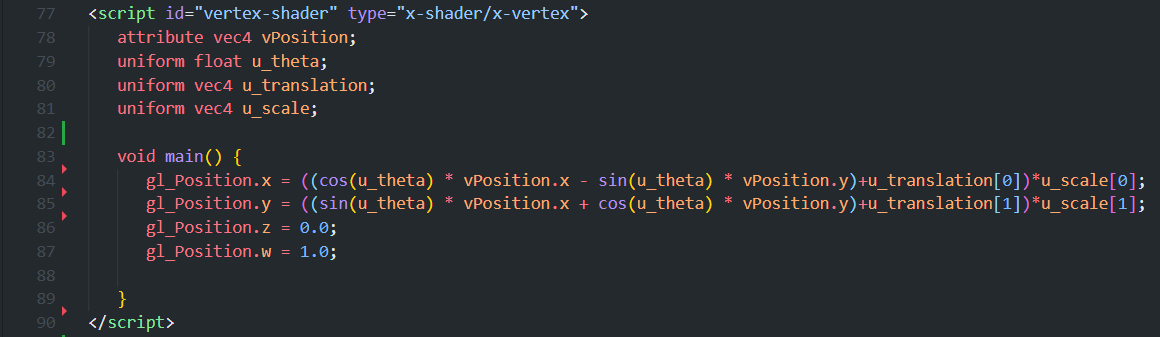
“ H ” HARFİ İÇİN ÜÇGEN NOKTA KOORDİNATLARI

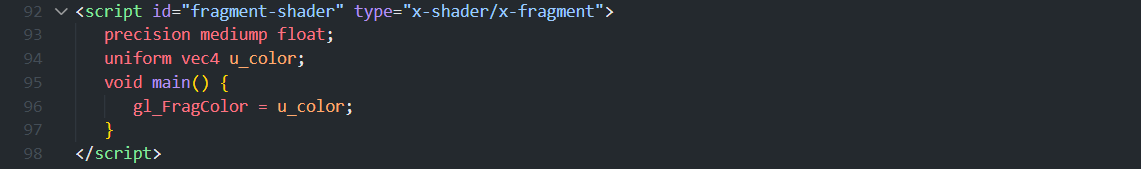
|  |  |  |  |
| --- | --- | --- | --- |
| ÜÇGENLER | P1 | P2 | P3 |
| ABC | vec2(-0.9, 0.7) | vec2(-0.9, -0.7) | vec2(-0.7, -0.7) |
| ADC | vec2(-0.9, 0.7) | vec2(-0.7, 0.7) | vec2(-0.7, -0.7) |
| EFG | vec2(-0.7, 0.1) | vec2(-0.7, -0.1) | vec2(-0.3, -0.1) |
| EHG | vec2(-0.7, 0.1) | vec2(-0.3, 0.1) | vec2(-0.3, -0.1) |
| KNM | vec2(-0.1, 0.7) | vec2(-0.3, 0.7) | vec2(-0.3, -0.7) |
| KLM | vec2(-0.1, 0.7 | vec2(-0.1, -0.7) | vec2(-0.3, -0.7) |

“ Ş ” HARFİ İÇİN ÜÇGEN NOKTA KOORDİNATLARI

|  |  |  |  |
| --- | --- | --- | --- |
| ÜÇGENLER | P1 | P2 | P3 |
| ABC | vec2(0.1, 0.7) | vec2(0.9, 0.7) | vec2(0.9, 0.5) |
| ADC | vec2(0.1, 0.7) | vec2(0.1, 0.5) | vec2(0.9, 0.5) |
| DEF | vec2(0.1, 0.5) | vec2(0.3, 0.5) | vec2(0.3, 0.1) |
| DGF | vec2(0.1, 0.5) | vec2(0.1, 0.1) | vec2(0.3, 0.1) |
| GNL | vec2(0.1, 0.1) | vec2(0.1, -0.1) | vec2(0.9, -0.1) |
| GKL | vec2(0.1, 0.1) | vec2(0.9, 0.1) | vec2(0.9, -0.1) |
| LPR | vec2(0.9, -0.1) | vec2(0.9, -0.7) | vec2(0.7, -0.7) |
| LMR | vec2(0.9, -0.1) | vec2(0.7, -0.1) | vec2(0.7, -0.7) |
| TYR | vec2(0.1, -0.5) | vec2(0.7, -0.5 | vec2(0.7, -0.7) |
| TSR | vec2(0.1, -0.5) | vec2(0.1, -0.7) | vec2(0.7, -0.7) |
| QZU | vec2(0.4, -0.8) | vec2(0.4, -1.0) | vec2(0.6, -1.0) |
| QWU | vec2(0.4, -0.8) | vec2(0.6, -0.8) | vec2(0.6, -1.0) |

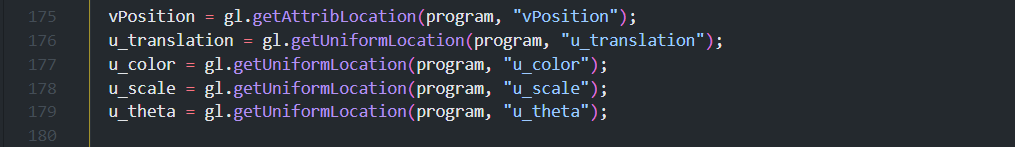
KAYNAK KOD MİMARİSİ



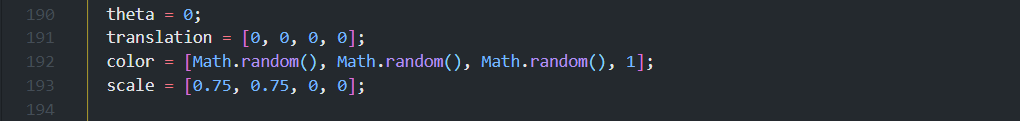


Canvas üzerinde öteleme, ölçeklendirme, döndürme ve renk değiştirme özelliklerini kontrol edebilmek için uniform değişkenler oluşturdum.

-----------------------------------------------------------------------------------------------------------------

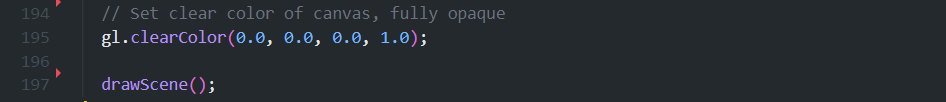


Javascript içerisinde bu değişkenlere erişmek ve manipüle edebilmek için getUniformLocation() ve getAttribLocation() fonksiyonlarını kullandım.



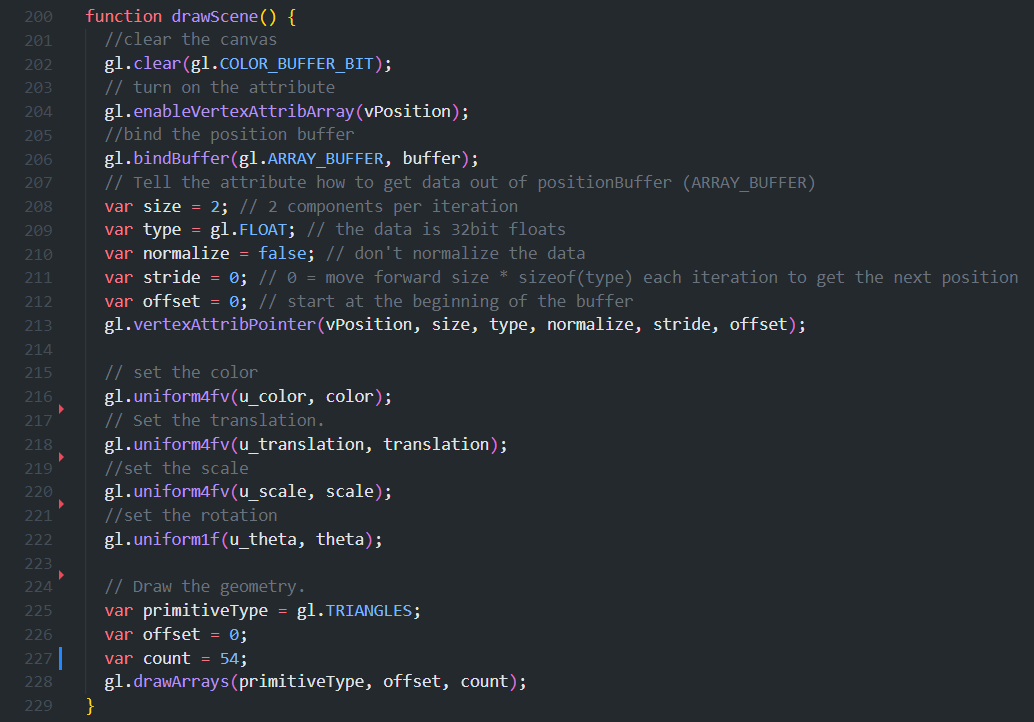
Canvas ilk kez ekrana çizilmeden önce kullanması gereken değerleri atadım.

-----------------------------------------------------------------------------------------------------------------



Canvas’ın rengini değiştirdim ve drawScene() fonksiyonunu çağırarak ilk çizdirme işlemini yaptım.

-----------------------------------------------------------------------------------------------------------------

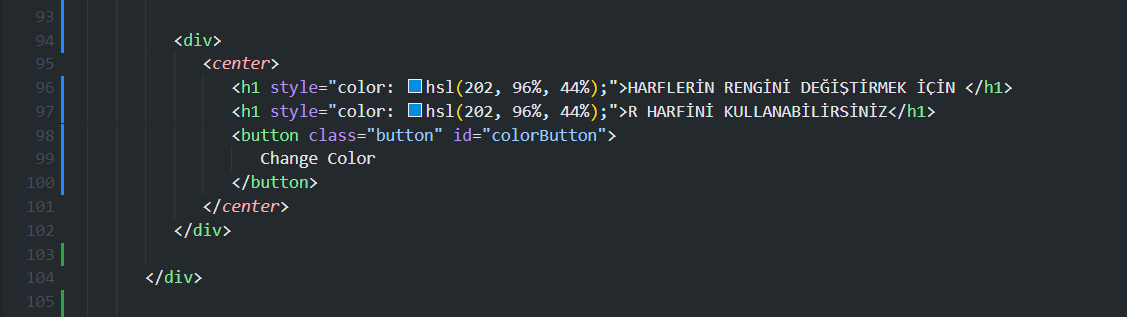


Bu fonksiyon vertex-shader ve fragment-shader için gerekli uniform ve attribute değişkenlerine atama işlemi yapıyor. Devamında ise canvas’ı ekrana çizdiriyor

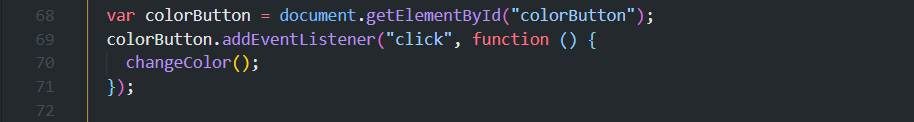
ÖZELLİKLER VE ETKİLEŞİM TÜRLERİ

|  |  |  |
| --- | --- | --- |
| ÖZELLİK | VAR / YOK | ETKİLEŞİM TÜRÜ |
| Harf Çizdirme | Var | - |
| Renk değiştirme | Var | Mouse & Keyboard & GUI |
| Ölçeklendirme | Var | Mouse & GUI |
| Öteleme | Var | Mouse & Keyboard & GUI |
| Döndürme | Var | Mouse & Keyboard & GUI |

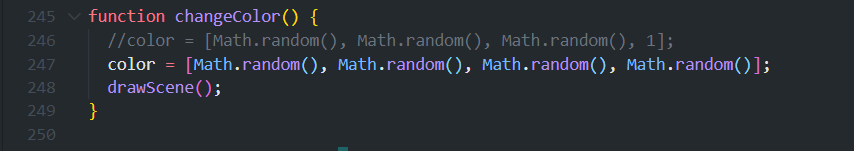
RENK DEĞİŞTİRME



HTML dosyası içerisinde bir buton oluşturdum.



Javascript dosyası içerisinde oluşturduğum butona id üzerinden eriştim. Butona tıklandığında harflerin rengini değiştirmek için changeColor() fonksiyonunu çağırdım.

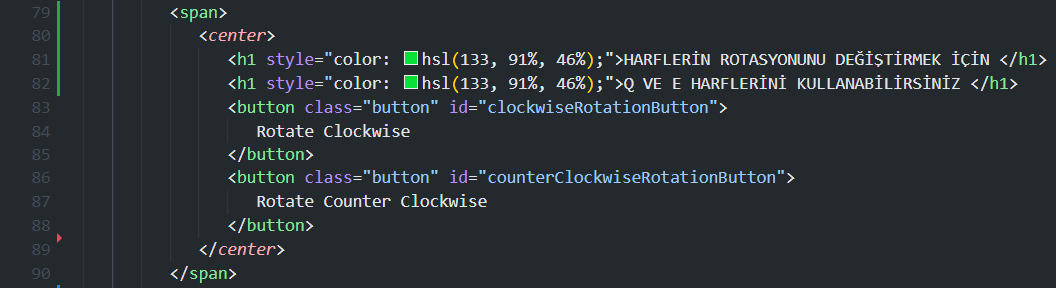


Bu fonksiyon her çağrıldığında harflerin rengi random olarak değişir.

BEFORE AFTER

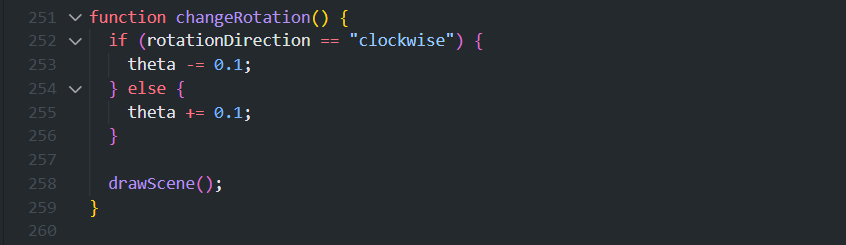
 

DÖNDÜRME



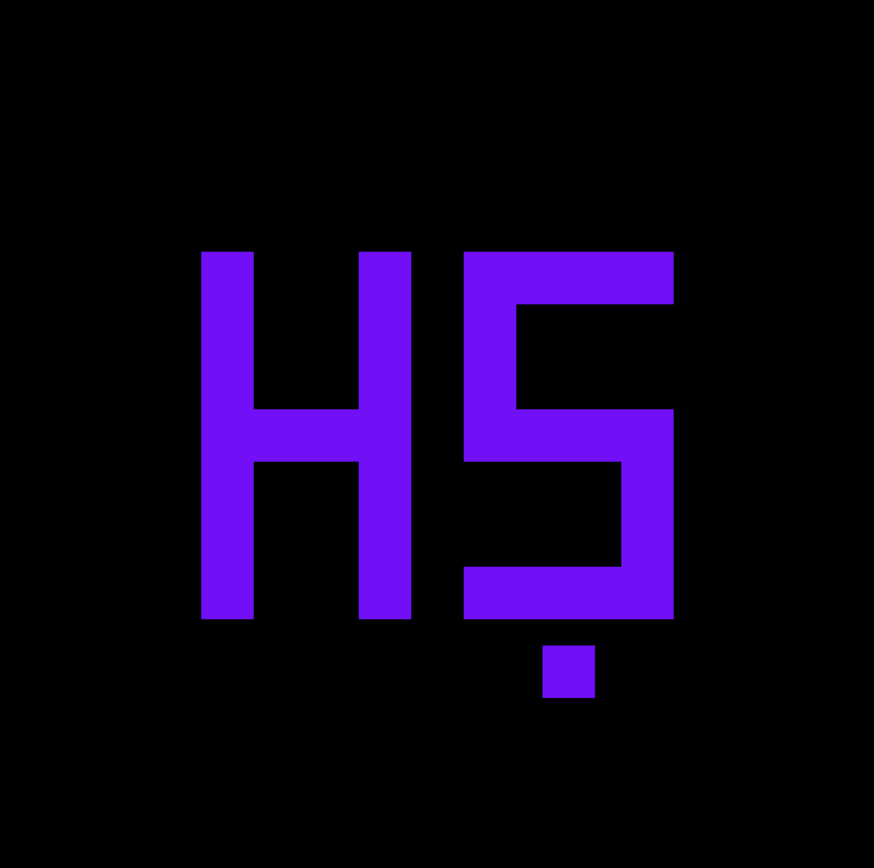
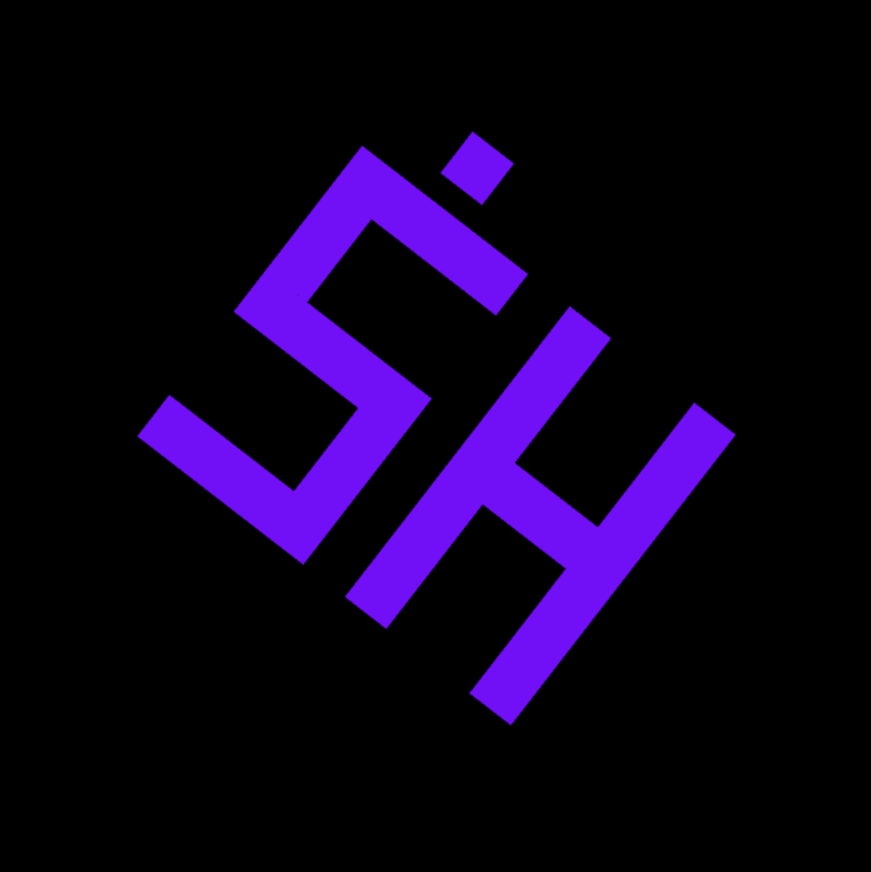
HTML dosyası içerisinde harfleri saat yönünde ve saat yönünün tersinde döndürebilmek için 2 tane buton tanımladım.

Javascript dosyası içerisinde oluşturduğum 2 butona id değerleri üzerinden eriştim. Butonlara tıklandığında harflerinin rotasyonunun değişmesi için changeRotation() fonksiyonunu oluşturdum.

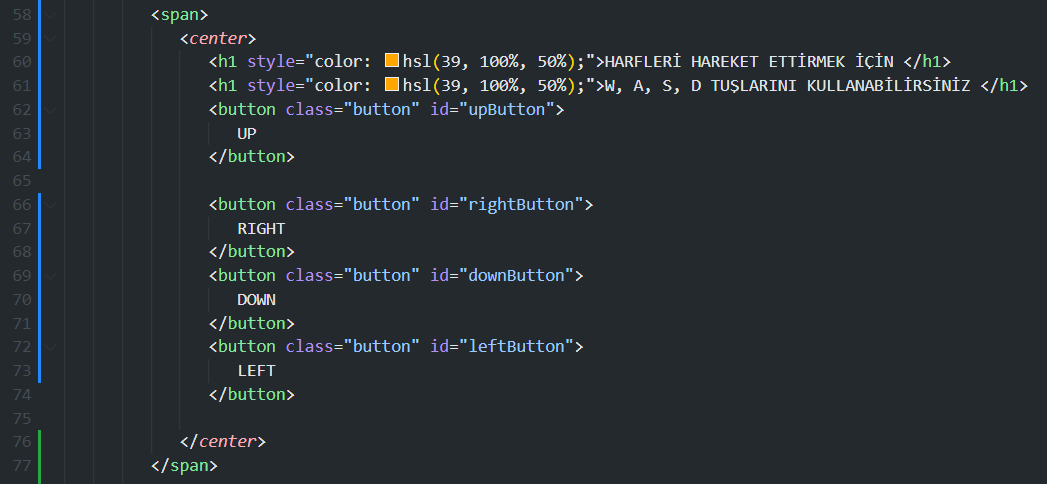


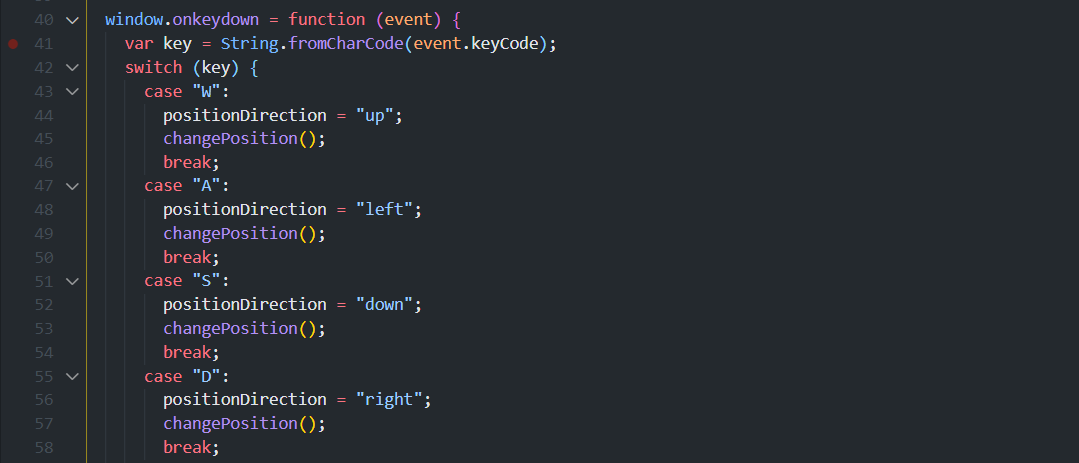
Bu fonksiyon theta değerini değiştirerek harflerin rotasyonları üzerinde değişiklik yapıyor.

BEFORE AFTER

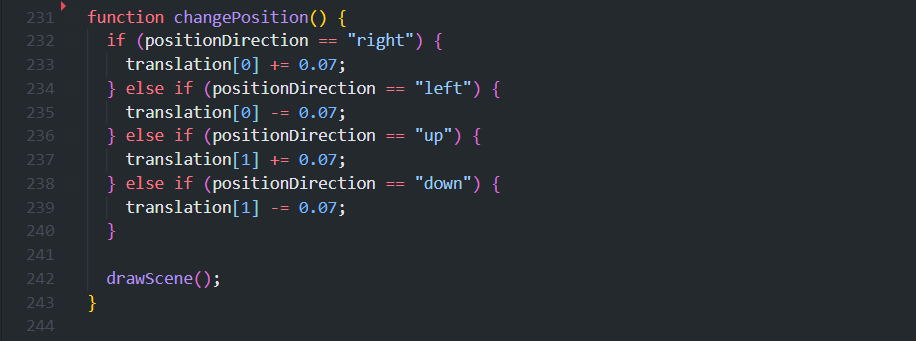
 

ÖTELEME

HTML dosyası içerisinde harfleri koordinat sisteminde öteleyebilmek için 4 adet buton oluşturdum.



Klavye üzerinden alınan inputlara göre harflerin gereken yöne ötelenmesini sağlayan bir switch-case bloğu oluşturdum.

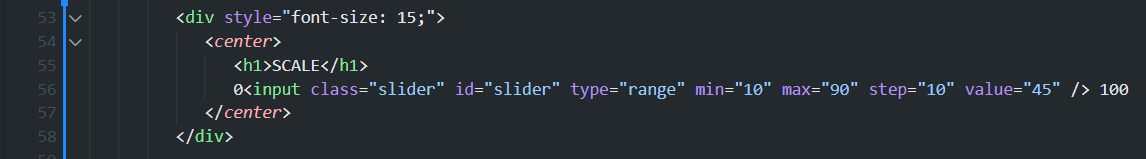


Bu fonksiyon klavyede tıklanan tuşlara göre harfleri seçilen yönde öteliyor.

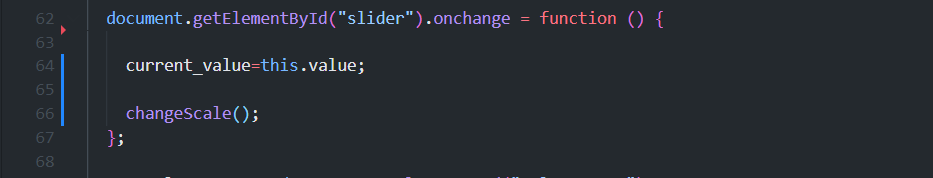
BEFORE AFTER

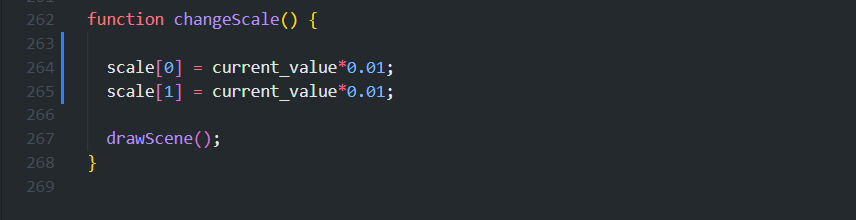
ÖLÇEKLENDİRME



HTML dosyası içerisinde bir range slider oluşturdum.

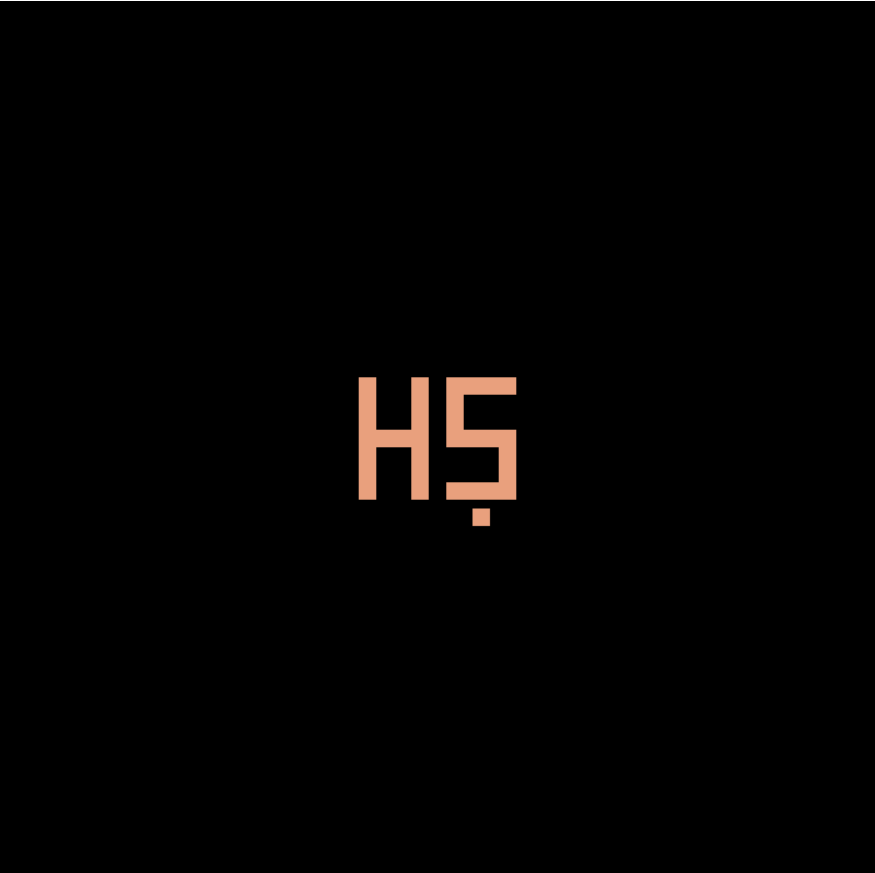


Javascript dosyası içerisinde oluşturduğum slider’a id üzerinden eriştim. Slider üzerindeki değeri current\_value üzerine atadım.

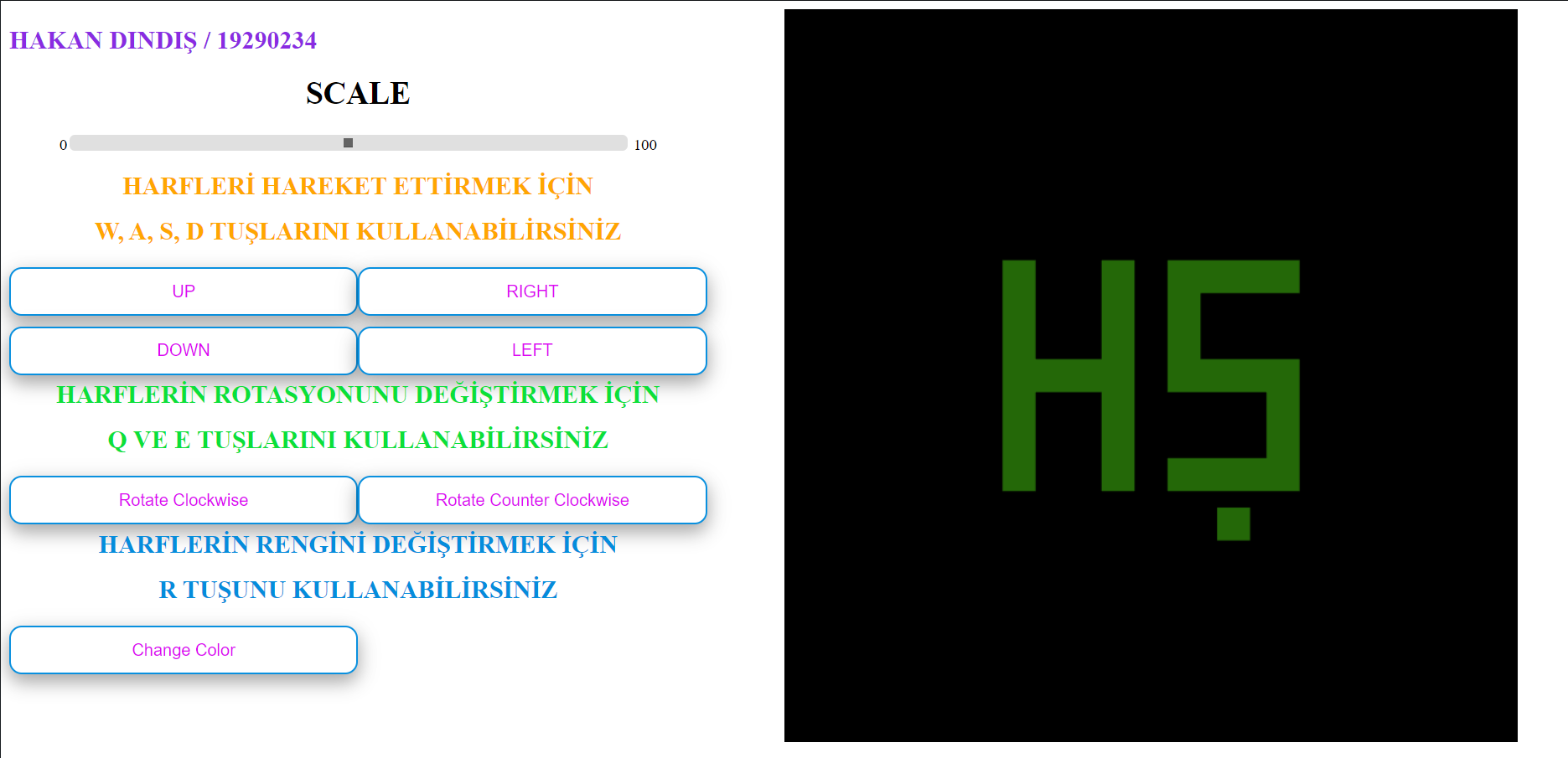


Bu fonksiyon ise x ve y değerlerini current\_value yani slider’ın o anki değeri ile çarparak harflerin ölçeklendirilmesini sağlıyor.

BEFORE AFTER

PROGRAMIN İLK KEZ ÇALIŞTIRILDIĞI HALİ



KULLANICININ YAĞTIĞI DEĞİŞİKLİKLERDEN SONRA



NOTLAR

Uygulamadaki etkileşimi arttırmak için öteleme, renk değiştirme, ölçeklendirme ve döndürme işlemlerini hem klavye hem de GUI ile değiştirebileceğimiz şekilde tasarladım. Böylece mobil platformlar için GUI aracılığı ile uygulamayı sorunsuz şekilde çalıştırabileceğiz. Ayrıca pc kullanıcıları ise klavye aracılığı ile her seferinde butonlara tıklamadan daha kolay bir şekilde uygulamayı kullanabilecek.

Uygulama github üzerinde aktif olarak çalışmaktadır. Aşağıdaki link üzerinden uygulamaya erişebilirsiniz.

<https://hakandindis.github.io/source/19290234.html>