



BLM2537- Ağ Tabanlı Programlama Proje Ödevi Ayça Örge - 24290759

Site: <https://aycaorge.github.io/projeoyun/>

(Mac bilgisayarda kodlandı windows bilgisayara optimize
edilmedi)

Video link: <https://www.youtube.com/watch?v=qmWiM8VJuyo>

Github repo:<https://github.com/aycaorge/projeoyun>

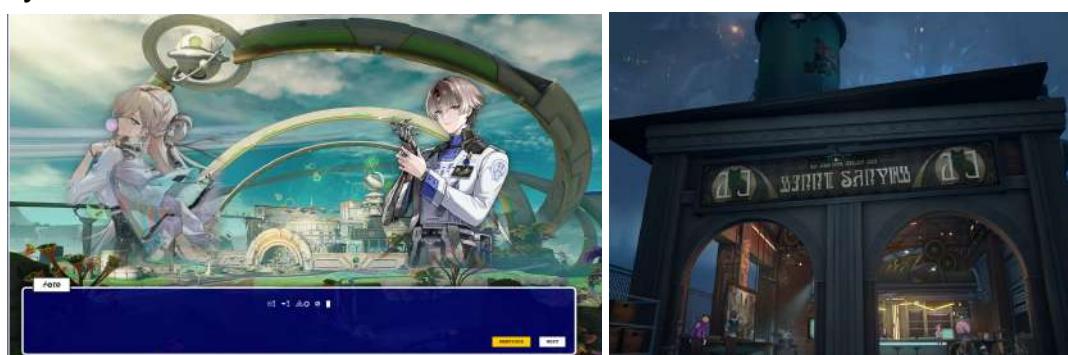
Bölüm 1: Websitenin Konsepti

Websiteyi yaparken ilk adımda bir mesaj veren resim galerisi gibi bir site yapmak istedim. Fakat gereken kalitede ve istenilen konsepte resim bulmak zordu. İki seçenekim vardı; Resimleri kendim çizmek ya da AI ile üretmek. AI üyeliklerine yetecek maddi kaynaklarım olmadığı ve öyle bir site yapmak istersem fazlasıyla görsel kaynak gerekeceği için bundan vazgeçtim. Sitemin yeni konsepti ise JRPGlerden yani hikaye bazlı japon "roleplaying" oyunlarından esinlenmiş mini bir hikayede ilerleme pokemon stilini boss ile savaşıma ve galeriden resim açma oyunu.

Websitesindeki elementleri hazırlarken zaten herkesin aşina olduğu rpg estetiğinden yararlanıldı. Örneğin karakterler arası diyalog kutularını hazırlarken aşağıdaki resimlerde görüldüğü gibi mavi üzerinde beyaz yazılı retro bir görüntü yakalamaya çalıştım.



Yaptığım sitede görüldüğü gibi resimler ne kadar "pixelated" olmasa bile mavi arka plan beyaz yazı ile rpg görüntüsüne sadık kaldım. Küçük sarı yazıları kullanmak yerine butonların üzerine geldiğinde sarı olmalarını sağlayarak konsept bütünlüğünü korudum. Aynı zamanda oyun şirketlerinin yaptığı gibi uydurma bir dil kullandım.



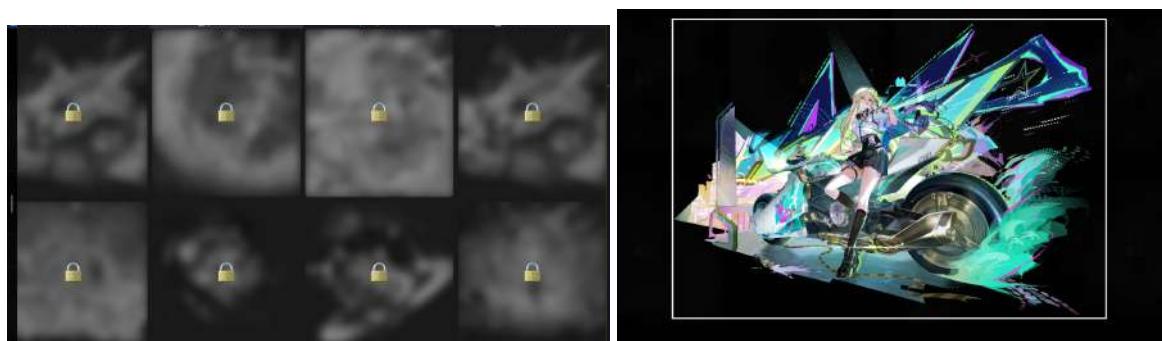
“Boss Fight” kısmında ise pokemon savaşlarını andıran bir görüntünden esinlendiğim. Aşağıdaki karşılaştırılmada görüldüğü gibi soldaki “player” karakteri ve “düşman” karakteri olacak şekilde yerleştirdim. Boss'a saldırduğımızda karşı saldırı yapıyor ve pokemonda olduğu gibi kendi karakterimizin canını azaltıp karakterimin resmini sarsıyor. Biraz daha moderleştirmek adına “What should MUDKIP do?” kısmındaki yazıları eklemeyip boss fightta minik chat bubblelar kullandım.



Ana Sayfada ise çok popüler mobil gacha oyunlarındaki banner sayfalarından esinlendim. Genelde karakterler yan yada düşey bloklar içinde çizimleri olacak şekilde dururlar ve üstüne mouse kaydırığınızda animasyonları oynamaya başlar bende bu havayı vermek için ana sayfamda bu stilı kullandım.



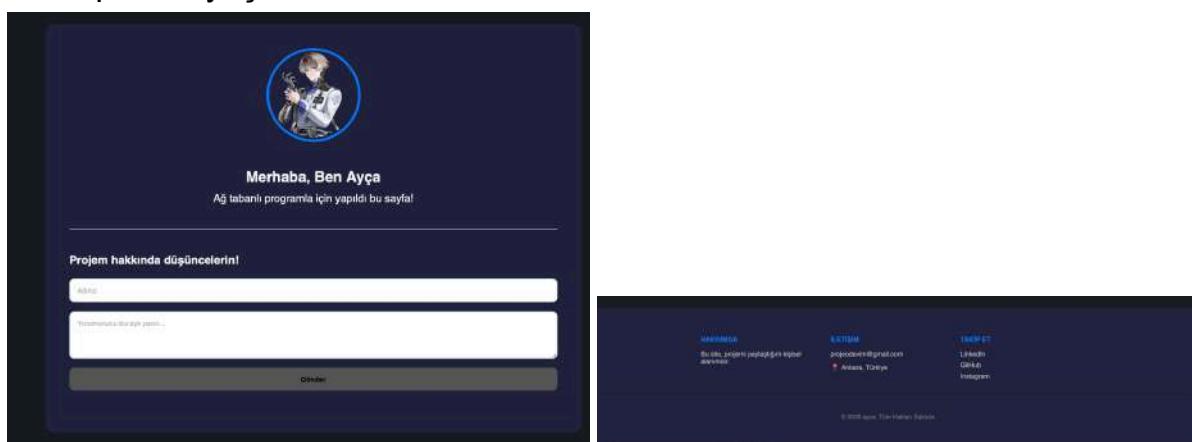
Galeri sayfasını sade yaptım, storyde ilerleyince galeriden resim açma mekanizması çoğu oyunda var ve ben de bunu eklemek istedim. Galerideki kapalı resimlere tıklarsanız hikayede ilerleme seviyenize göre resimleri açıyor eğer ilerlememişseniz hikayede ilerlemeniz için sizi uyarıyor.



Sitede yandan kaydırımalı menü kullanıldı. Hem sitede istenmeyen herhangi bir yeri kaplamıyor hem de görüldüğü gibi konsepte uyuyor.



About/hakkında sayfası projenin gereksinimi olduğu için eklendi. Normal sayfalarda footer olmamasına rağmen footer ve yorum alanı/mesaj alan bir yapı projenin gereksinimi olduğu için bu sayfaya eklendi. Ne kadar sitemin konsepti ile uyuşmasa bile.



Bölüm 2: Teknolojiler ve Kodlama.

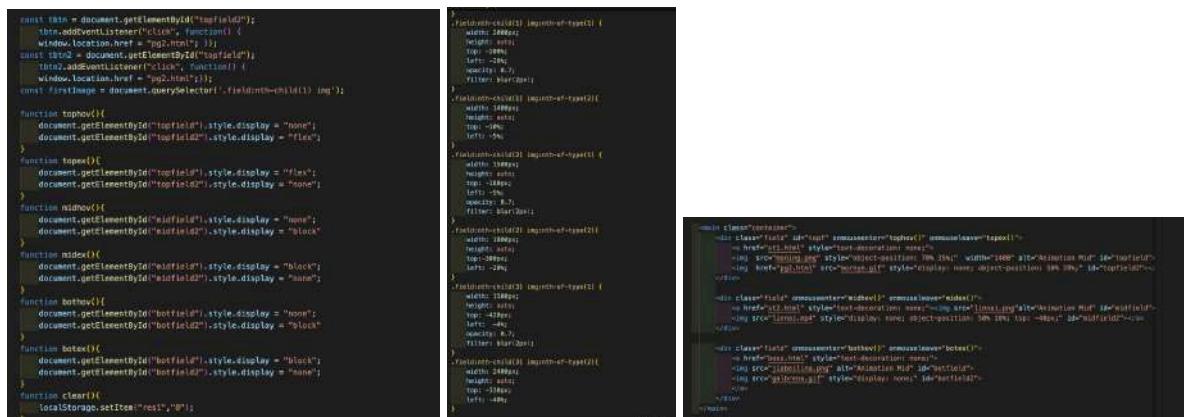
İlk olarak kullanılan teknolojiler sırayla : Github,HTML,CSS,Javascript,Gemini AI(Js, Html öğrenirken soru sorma amaçlı) olmak üzere 5 tane.

2.1:HTML

HTML genel olarak çok kullanmamaya yani daha temiz olsun diye CSS dosyasında objelerimi kodlamaya çalıştım fakat ne kadar HTML dosyamı temiz tutmak istesemde animasyonlar ve geçişleri yaparken bazen tekil objeler aynı class a ait olsada boyutlarında sorun olduğu için bazı özelliklerini HTML dosyamda kodlama becerilerimi zorladım ama sonradan aynı class'a ait olan objeleri CSS dosyasında `child` keywordu ile ayırabildiğimi öğrenmem ile kodumu yeniden düzenleyip bazı zorlama yazdığım kodları CSS ile temize çekmeyi başardım. Onun dışında script eklemek, objeleri belirlemek dışında kodumu CSS içinde yazmaya çalıştım.

2.2:CSS

Genel olarak htmlde yapabileceğim herşeyi CSS'de yapmaya çalıştım. Ama bazı özellikleri javascriptte değiştirirken id kullanarak element seçimi yapmam gerektiği için html dosyasında bazı özellikleri tanımladığım oldu.



Yukarıdaki örnek kodlardaki gibi ortadaki CSS kod bloğunda ana sayfamda üç resim bloğunu nth child notasyonuyla değiştirmiş olsamda yinele HTML kısmında javascript ile bahsi geçen elemanlara daha kolay erişebilmek için id ataması yapmak zorunda kaldım. Javascript kodunda görüldüğü gibi ana sayfamda kodları onmouseenter ve onmouseleave ile yaptım çünkü animasyonlu geçişlere en uygun bunun olduğuna karar verdim. Aynı zamanda animasyonların boyutları birbirinden farklı olduğu için position parametresinde flexible vs. gibi şeyle kullanıldığımda istediğim sonuçları alamadım. Position: absolute kullanarak top ve left parametreleriyle animasyonların lokasyonlarını kendim ayarladım. Sitenin en uğraştırıcı kısmı resimlerin ve animasyonların konumlarını ayarlamaktı. Uğraşlarımın sonunda genel olarak position absolute kullanarak yapmanın çoğu şeyi daha verimli yaptığına karar verdim.

```
<div id="chatBubble" class="chat-bubble">  
    Göldürme bemi!  
</div>  
<div id="chatBubble2" class="chat-bubble">  
    Güzel deneme ama yetmez!  
</div>
```

```
function showBubble() {  
    const bubble = document.getElementById("chatBubble");  
    bubble.style.display = "block";  
    setTimeout(() => {  
        bubble.style.display = "none";  
    }, 3000);  
}  
function showBubble2() {  
    const bubble = document.getElementById("chatBubble2");  
    bubble.style.display = "block";  
    setTimeout(() => {  
        bubble.style.display = "none";  
    }, 3000);  
}
```

Örneğin bazı yerlerde sürekli aynı olmayan şeyler yapmak istedim üstteki kod bloklarında olduğu gibi aslında düşmana vurduğunuzda size söylediğiniz repliği rastgele seçmek istedim ama random fonksiyonu ara sıra hata verdiği için 2 tane ayrı replik barındıran mesaj balonu yaratıp saldırısı tipine göre ikisinden birinin display şeklini 'none'dan 'block'a çeviren kod yazdım.

```
15 .lightbox {  
16     display: none;  
17     position: fixed;  
18     z-index: 200;  
19     top: 0;  
20     left: 0;  
21     width: 100%;  
22     height: 100%;  
23     background-color: #rgba(0,0,0,0.95);  
24     text-align: center;  
25 }  
26  
27 .lightbox img {  
28     max-width: 90%;  
29     max-height: 80%;  
30     margin-top: 5%;  
31     border: 3px solid #white;  
32 }  
33  
34 .close-lightbox {  
35     position: absolute;  
36     top: 20px;  
37     right: 40px;  
38     font-size: 40px;  
39     cursor: pointer;  
40 }
```

```
2 const CORRECT_PASSWORD = "1234";  
3  
4 function checkLock(id, fullImgUrl) {  
5     const lock = localStorage.getItem('res1');  
6  
7     if (lock === '1') {  
8         openLightbox(fullImgUrl);  
9     } else {  
10         alert("Hikayede ilerle");  
11     }  
12 }  
13  
14 function openLightbox(url) {  
15     const lightbox = document.getElementById("lightbox");  
16     const lightboxImg = document.getElementById("lightbox-img");  
17  
18     lightboxImg.src = url;  
19     lightbox.style.display = "block";  
20 }  
21  
22 function closeLightbox() {  
23     document.getElementById("lightbox").style.display = "none";  
24 }
```

Galeriyi kodlarken ise lightbox adında display=none olan objeler yarattım ve galerideki normal üzeri kilitli gözüken resimlerin üzerine tıklandığında hikayede ilerleme durumuna göre lightboxların display durumunu değiştiren kodu yazdım. Hikayede ilerleme durumunu ise localStorage.getItem kullanarak kontrol ettim.

2.3:Javascript

Beni en zorlamayan kısmı javascript oldu aşağıda javascripti sıkılıkla kullandığım yerleri açıkladım.

```
function gattack(moveName, attacker) {
    doWork();
    if (attacker === 'player') {
        enemyHP -= 10;
        updateBar('enemy-hp', enemyHP);
        animateHit('enemy-sprite');
        if (enemyHP <= 0) return endGame("Kazandın!");
        setTimeout(() => {
            enemyAttack2();
        }, 1000);
    }
}
function run() {
    // Do stuff ...
    let enemyHP = 200;
    let playerHP = 100;
    let x = localStorage.getItem("res1");
    let y = localStorage.getItem("res2");
    const listel = ["Yeteri kadar iyi değil","Bu kadar mı?","Goldürme beni!"]
    if(y == 1){
        enemyHP = 100;
    }
}
```

Hikayedede ilerlediğinizde local storageda “res1” ve “res2” değişkenlerini 1 olarak elde edeceksiniz. Bu değişkeni kullanarak Bossfightta kazanıp kazanmayacağınızı kontrol edecek mekanizma yukarıdaki kod bloklarında verilmiştir. Eğer 1. hikayedede ilerlemişseniz bossun gücü, 2. hikayedede ilerlemiş iseniz bossun canı azalmakta.

```
const listele1 = [
    "aa, ee, oo, ii",
    "ee -> eo, e -&gt; i",
    "ee -> EA, eo, AA, "
    "ei, ie -> E, I",
    "eo, oo -> O, U",
    "ea -> A, O",
    "oo, ee -> O, U",
    "ie, ei -> I, E",
    "oo, ee -> O, U",
    "eo, oo -> O, U",
    "ie, ei -> I, E",
    "ea -> A, O",
    "aa, ee, oo, ii"
];

const characterNames = [
    "aaor",
    "eeor",
    "oore",
    "iior",
    "eoir"
];

const listeright = ["sermey.png","lyneeg.png","xiangliyao.png"];
const listeleft = [0.7,1.8,7.1,8.7,2];
const listerights = [0.7,1.8,7.1,8.7,2];
const listelefts = [1,6,7,1,2,7,1,0.7];
const listdiagonals = ["lsermey.png","lyneeg.png","xiangliyao.png"];
const listdiagonals = [1,2,2,2,3,1,2,3,1,2,2,1,2,3,1,2,3,1,2];
```



```
function next(){
    if(dia<=listel.length-1){
        alert("Buyuk Son, Bossun can! acalid!");
        localStorage.setItem('res1', '1');
    }
    else{
        dia = dia +1;
        playtypewriter(listel[dia]);
        a = dialogmenu[dia];
        document.getElementById("leftc").src = listeright[a-1];
        document.getElementById("rightc").src = listeright[a];
        document.getElementById("name").innerHTML = characterNames[a];
        document.getElementById("leftc").style.opacity = listelefts[a];
        document.getElementById("rightc").style.opacity = listerights[a];
    }
}

function prev(){
    if(dia>0){
        alert("Ceriye Gidemezsin!")
    }
    else{
        dia = dia -1;
        a = dialogmenu[dia];
        playtypewriter(listel[dia]);
        document.getElementById("leftc").src = listeright[a-1];
        document.getElementById("rightc").src = listeright[a];
        document.getElementById("name").innerHTML = characterNames[a];
        document.getElementById("leftc").style.opacity = listelefts[a];
        document.getElementById("rightc").style.opacity = listerights[a];
    }
}

function playtypewriter(message) {
    let i = 0;
    let speed = 50;
    const displayElement = document.getElementById("dis");
    displayElement.innerHTML = "";
    function type() {
        if (i < message.length) {
            displayElement.innerHTML += message.charAt(i);
            i++;
            setTimeout(type, speed);
        }
    }
    type();
}
```

Hikayedeki diyalog kısmında değişen ifadeler böyle kodlandı. Yani siz aslında next tuşunda bastığınızda diyalog counterı 1 artırıyor ve html elementlerini listelerin o indexinde bulunan eleman ile değiştiriyor. Ve oyun havasını daha iyi yakalamak için sađdaki resimde görünen typewriter fonksiyonunu kullanıldı.

```
<script>
  document.getElementById('comment-form').addEventListener('submit', function(e) {
    e.preventDefault();
    const nameInput = document.getElementById('username');
    const textInput = document.getElementById('comment-text');
    const displayArea = document.getElementById('comments-display');
    if(nameInput.value.trim() === "" || textInput.value.trim() === "") return;
    const newComment = document.createElement('div');
    newComment.classList.add('comment-item');
    newComment.innerHTML = '<strong>' + nameInput.value + '</strong> <p>' + textInput.value + '</p>';
    displayArea.prepend(newComment);
    nameInput.value = "";
    textInput.value = "";
  });
</script>
```

Yukarıdaki kod bloklarında ise about sayfamdaki yorum ekleme mekanizması bulunmaktadır. Form doldurup submitlendiğinde formdaki bilgiler ile comment-item classına ait yeni bir obje oluşturuyor ve bu da yorumun gözükmekesini sağlıyor.

Projedeki kod miktarı bu kadarla sınırlı olmamakla birlikte. Bahsi geçen kod blokları en önemli site mekaniklerini içermektedir. Sitenin geri kalanındaki kod karmaşık olmadığı için rapora eklenmemiştir verilen github linkinden bu kısımlar kontrol edilebilir.

2.4:Github

Projeyi github extensionu indirip pushlayarak yapmak denenmiş olsada kendi local oturumumda kodlayıp sonrasında github sayesinde hostlamak daha verimli geldiği için hostlamak dışında kullanılmadı.

2.5:Gemini AI

Geminidan projeyi onun yapmamış olmasına dikkat ederek yararlanıldı. Bir şeyleri sıfırdan kodlamasını istemektense spesifik durumlarda hangi özelliklerin kullanılması gerektiğini ve başka programlama dillerinden aşina olunan konseptlerin JS özelinde nasıl çalıştığını sormak amaçlı kullanıldı. Spesifik yerleri gemini ile yapmak denendi fakat kodu debuglamak kodlamaktan daha uzun sürdüğü için onu daha çok mentor olarak kullanmak amaçlandı.

Bölüm 3: Diğer Şeyler

3.1 Asset Hazırlama

Sitede belirtildiği gibi çok sayıda animasyon ve resim var. Bunları elde etmek için bir oyunun promosyonel resimleri kullanıldı. Resimler fazla kaliteli olduğu için kalite düşürüldü. Bunun için internette bulunan GIF/PNG boyutlandırma webappleri kullanıldı.

3.2 Projemin başından itibaren değişen şeyler.



Projenin başlangıcındaki sitenin görsellerine bakıldığından arada büyük bir fark olduğu apaçık ortada. Sitenin ilk demo versiyonundan itibaren yapılan şeyleri listelemek gerekirse:

- 1- Resimler ve Animasyonlar bulundu.
- 2- Daha iyi animasyon kodu için uğraşındı.
- 3- Javascript öğreniminin ilerlemesiyle daha komplex animasyonlar yapıldı.

