

1. Değişkenler ve Veri Tipleri (Variables and Data Types)

Alıştırma: Oyuncu Bilgilerini Tanımla

Görev: Bir oyuncu karakterinin adı, yaşı, ve can değerini saklayacak değişkenler tanımlayın.

```
playerName (string)
playerAge (int)
playerHealth (float)
```

Örnek Kod:

```
public string playerName = "Hero";
public int playerAge = 20;
public float playerHealth = 100.0f;
```

2. Erişim Belirleyiciler ve [SerializeField] (Access Modifiers & [SerializeField])

Alıştırma: Unity Editor'de Görünen Değişken

Görev: private bir değişken tanımlayın ve [SerializeField] kullanarak bu değişkenin Unity Editor'de görünmesini sağlayın.

Örnek Kod:

```
[SerializeField] private int playerLives = 3;
```

Bonus: Oyuncunun can değerini Unity üzerinden değiştirip sahnede çalıştırın.

3. If-Else Yapısı (Conditional Statements)

Alıştırma: Kazanma Koşulu Kontrolü

Görev: Oyuncu skoru 100 veya daha yüksekse "Kazandınız!" mesajını yazdırın, değilse "Devam et!" mesajını yazdırın.

Örnek Kod:

```
public int playerScore = 50;

void CheckScore()
{
    if (playerScore >= 100)
    {
        Debug.Log("Kazandınız!");
    }
    else
    {
        Debug.Log("Devam et!");
    }
}
```

4. Döngüler (Loops)

Alıştırma: Düşman Spawn Etme

Görev: Bir for döngüsü kullanarak 5 kez bir düşman objesi spawn edin.

Düşman prefab'ini sahneye ekleyin.
Instantiate kullanarak döngü içinde spawn edin.

Örnek Kod:

```
public GameObject enemyPrefab;
public Transform spawnPoint;

void SpawnEnemies()
{
    for (int i = 0; i < 5; i++)
    {
        Instantiate(enemyPrefab, spawnPoint.position, Quaternion.identity);
    }
}
```

```
1  
    Instantiate(enemyPrefab, spawnPoint.position, Quaternion.identity);  
2  
}  
}
```

5. Metotlar (Methods)

Alıştırma: Oyuncu Hareketi

Görev: Bir MovePlayer metodu tanımlayın ve oyuncuyu ileri hareket ettirin.

speed parametresi alın.
transform.Translate kullanın.

Örnek Kod:

```
public void MovePlayer(float speed)  
{  
    transform.Translate(Vector3.forward * speed * Time.deltaTime);  
}
```

6. Parametrelili Metotlar (Methods with Parameters)

Alıştırma: Hasar Alma

Görev: Bir TakeDamage metodu yazın ve oyuncunun canını azaltın.

Parametre olarak hasar değeri (damage) alın.
playerHealth değişkeninden bu değeri çıkarın.

Örnek Kod:

```
public float playerHealth = 100.0f;  
  
void TakeDamage(float damage)  
{  
    playerHealth -= damage;  
    Debug.Log("Current Health: " + playerHealth);  
}
```

7. Dönüş Değeri Olan Metotlar (Return Methods)

Alıştırma: Sağlık Kontrolü

Görev: Oyuncunun sağlığını döndüren bir metot yazın.


Örnek Kod:

```
public int GetHealth()  
{  
    return playerHealth;  
}
```

Bonus: Sağlık değeri kritik seviyede olduğunda (örneğin, 20'nin altına düştüğünde) uyarı mesajı yazdırın.

Oturum-2

Ders-3 → Interacting with gameobjects via scripts.

- ↳ Transform component ✓
- ↳ Transform.Translate(); → Unity scripting API ✓
- ↳ Transform.Rotate(); ✓
- ↳ Transform.localScale; ✓
- ↳ Inputs  ✓
- ↳ Player movement ✓

↳ Inputs ✓

↳ Player Movement ✓

↳ Challenge: Player Bounds → HW

Ders 4 → Create and Manage multiple objects dynamically

↳ Prefabs? → Laser Prefab setup ✓

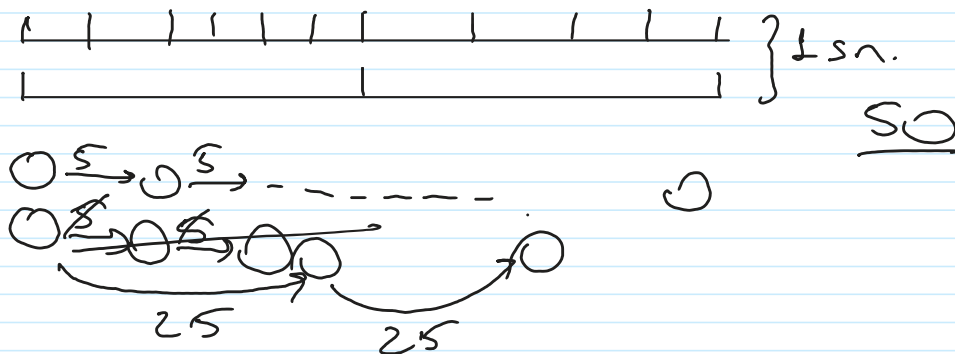
↳ Instantiating → Instantiate(); → Instantiate laser. ✓

↳ Destroy(); ✓

Challenge: Laser behaviour (movement) ✓
: Destroy laser. ✓

Bonus: Laser Position offset ~ 8

↳ Cooldown system.



$$\text{localScale} \rightarrow \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} x+1 \\ y+2 \\ z+3 \end{pmatrix}$$

time = 0, 1 - - - - - 9 - - - - - 15

cool... = 2

next = 0 → 9 + 2 = 11 → 15 + 2 = 17