

# Unity Workshop

**Hack'n Break**

**28.09.2019**

**Beyza Hilal Durak**

[www.beyzahilaldurak.com](http://www.beyzahilaldurak.com)  
[beyzahilaldurak@hotmail.com](mailto:beyzahilaldurak@hotmail.com)

# Install Unity 2019.1.6

(\*) Unity'i web sitesinden UnityHub yardımıyla indirebilirsiniz.  
Buradaki anlatım workshop anında USB'den offline olarak indirmek içindir.

- USB'deki *UnityOfflineInstaller* dosyasının içindeki işletim sisteminize ait olan dosyayı bilgisayarınıza kopyalayın.
- Mac OS
  - **Terminal.app** uygulamasını açın.
  - Bilgisayarınıza kopyaladığınız *Unity\_MacOS* klasörüne gidin. Bunun için Terminal de **cd dosya\_path** komutunu kullanmanız lazım.
  - Terminal de **sudo ./install.sh** komutunu çalıştırın. Bilgisayarınızın şifresini girmeniz istenecek. Şifreyi girip enter'e basın. Unity yüklenmeye başlayacak.
  - Yükleme tamamlandıktan sonra, uygulamalar klasörünün altında Unity klasörünü göreceksiniz.
  - Unity'i çalıştırmadan önce *Unity\_MacOS* klasörü içindeki Unity Hup.app uygulamasını **Uygulamalar** klasörünüzün altına taşıyın. UnityHub projelerinizi yönetmekte size yardımcı olacak bir uygulamadır.
  - Yükleme tamamlandı, Unity'i kullanmak için **UnityHub.app** uygulamasını çalıştırın.

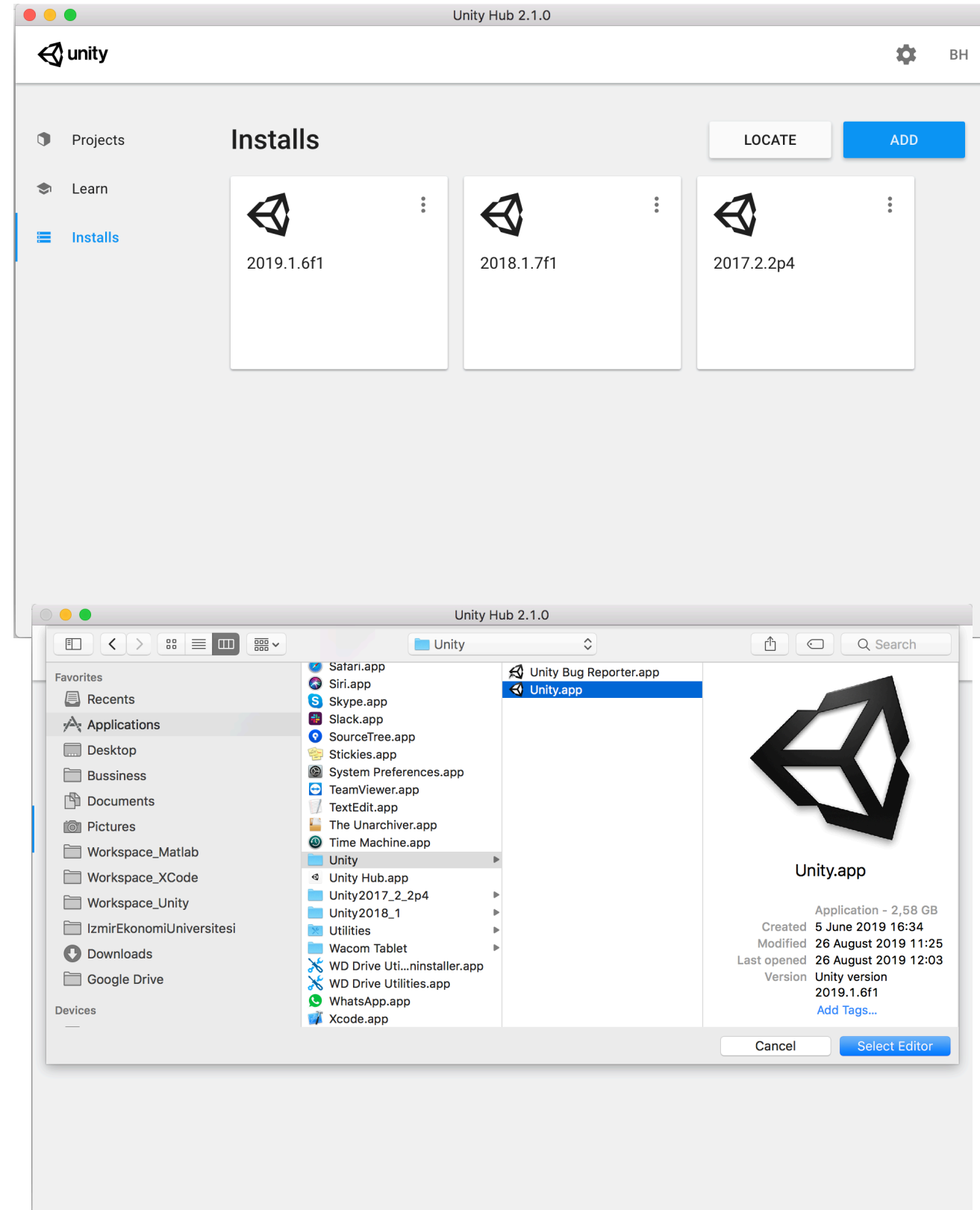
# Install Unity 2019.1.6

(\*) **Unity'i web sitesinden UnityHub yardımıyla indirebilirsiniz.**  
**Buradaki anlatım workshop anında USB'den offline olarak indirmek içindir.**

- USB'deki *UnityOfflineInstaller* dosyasının içindeki işletim sisteminize ait olan dosyayı bilgisayarınıza kopyalayın.
- Windows
  - Bilgisayarınıza kopyaladığınız *Unity\_Windows* klasörü içindeki **UnitySetup64.exe**'yi çalıştırın.
- Yada
  - Komut İstemini açın. Bunun için Start'da **cmd** yazıp arayın ve komut istemini **yönetici olarak çalıştırın**.
  - Bilgisayarınıza kopyaladığınız *Unity\_Windows* klasörüne gidin. Bunun için Terminal de **cd dosya\_path** komutunu kullanmanız lazım.
  - Terminal de **install.bat** komutunu çalıştırın. Unity yüklenmeye başlayacak.
  - Unity'i çalıştırmadan önce *Unity\_Windows* klasörü içindeki **UnityHubSetup.exe** uygulamasını çalıştırarak UnityHub'i bilgisayarınıza yükleyin. UnityHub projelerinizi yönetmekte size yardımcı olacak bir uygulamadır.
  - Yükleme tamamlandıktan sonra, bilgisayarı yeniden başlatın.
  - Unity'i kullanmak için **UnityHub** uygulamasını çalıştırın.

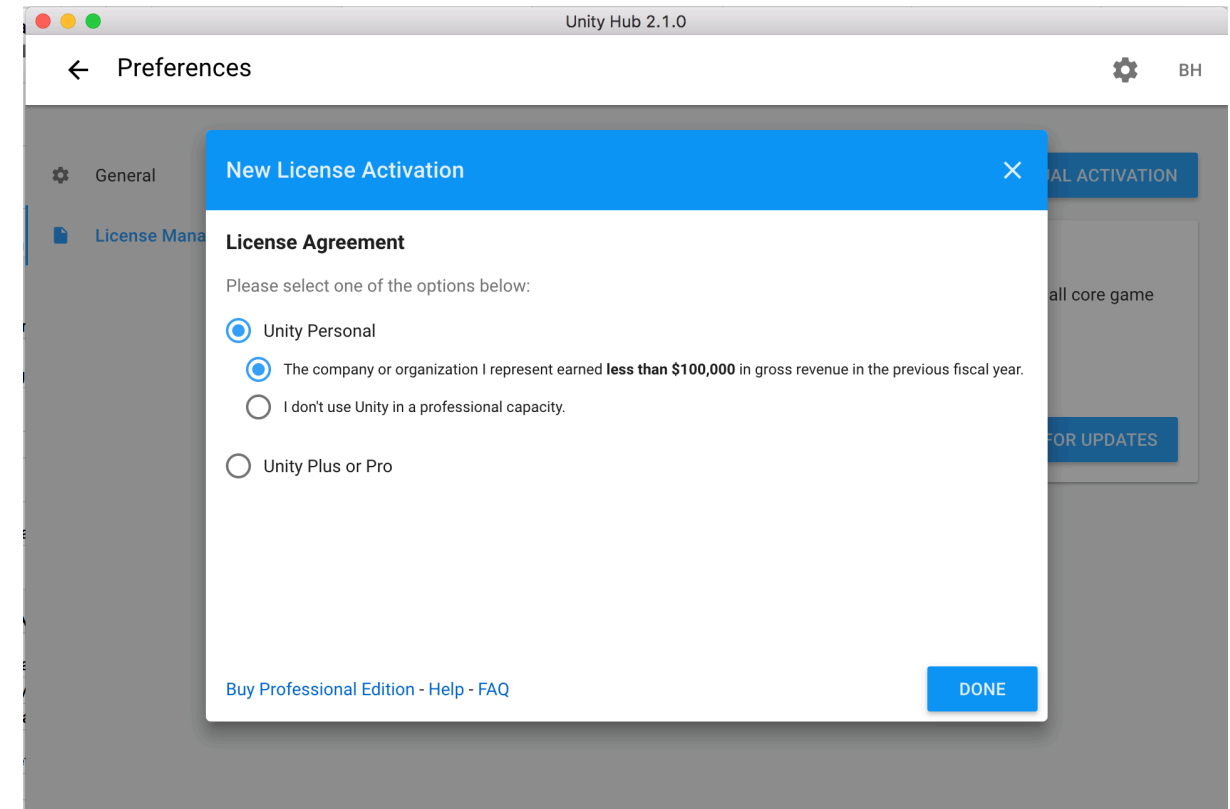
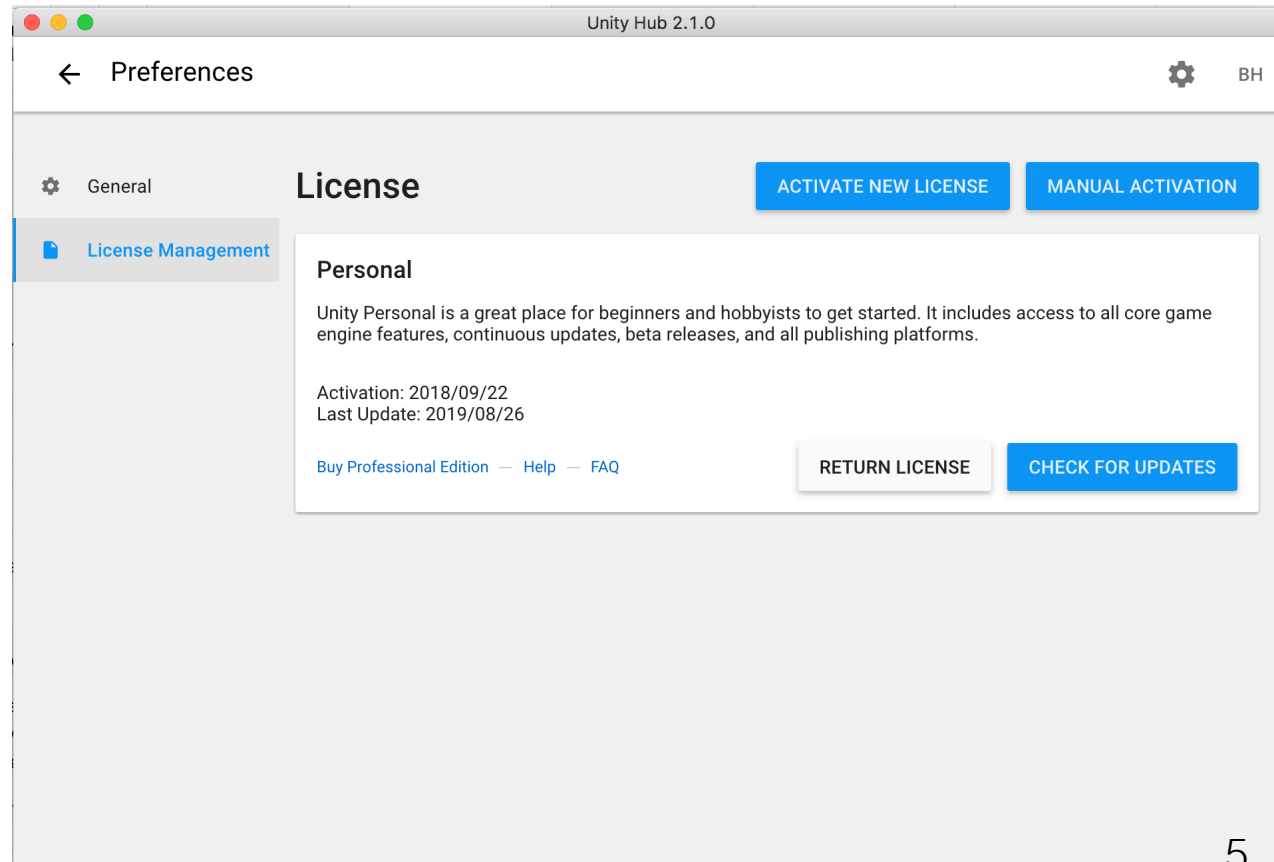
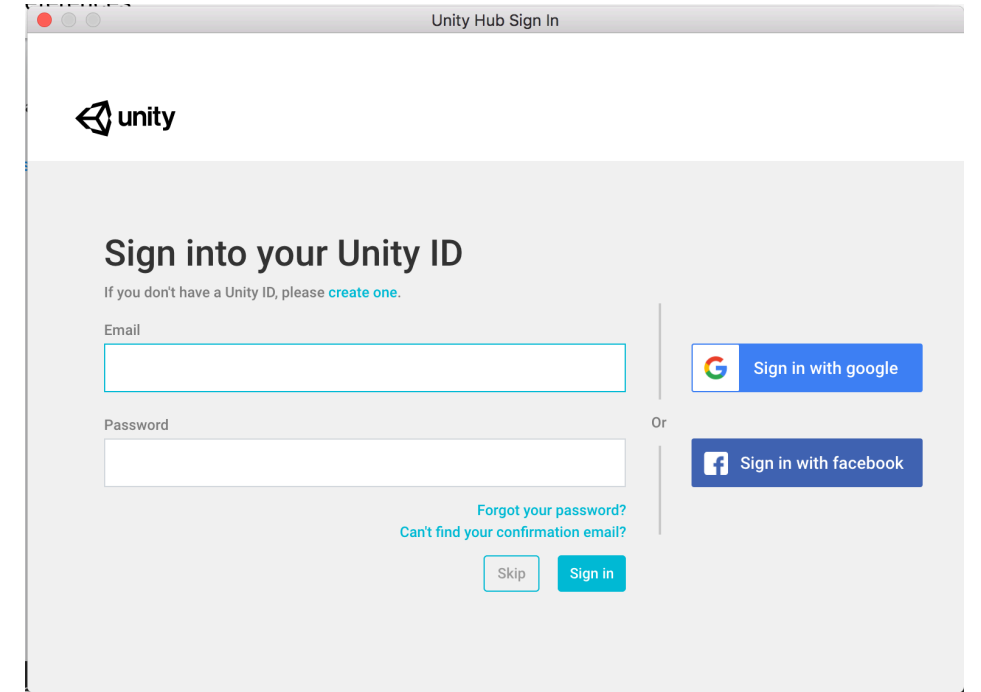
# Unity Hub

- Unity Hup uygulamasında Installs tab'ı altında 2019.1.6f1 Unity versiyonunu görmeniz lazım.
- Yoksa, Locate'e tıklayıp Unity.app'in konumunu seçmelisiniz.
- MacOS: **Uygulamalar/Unity/Unity.app**
- Windows: **C:\Program Files\Unity\Editor**



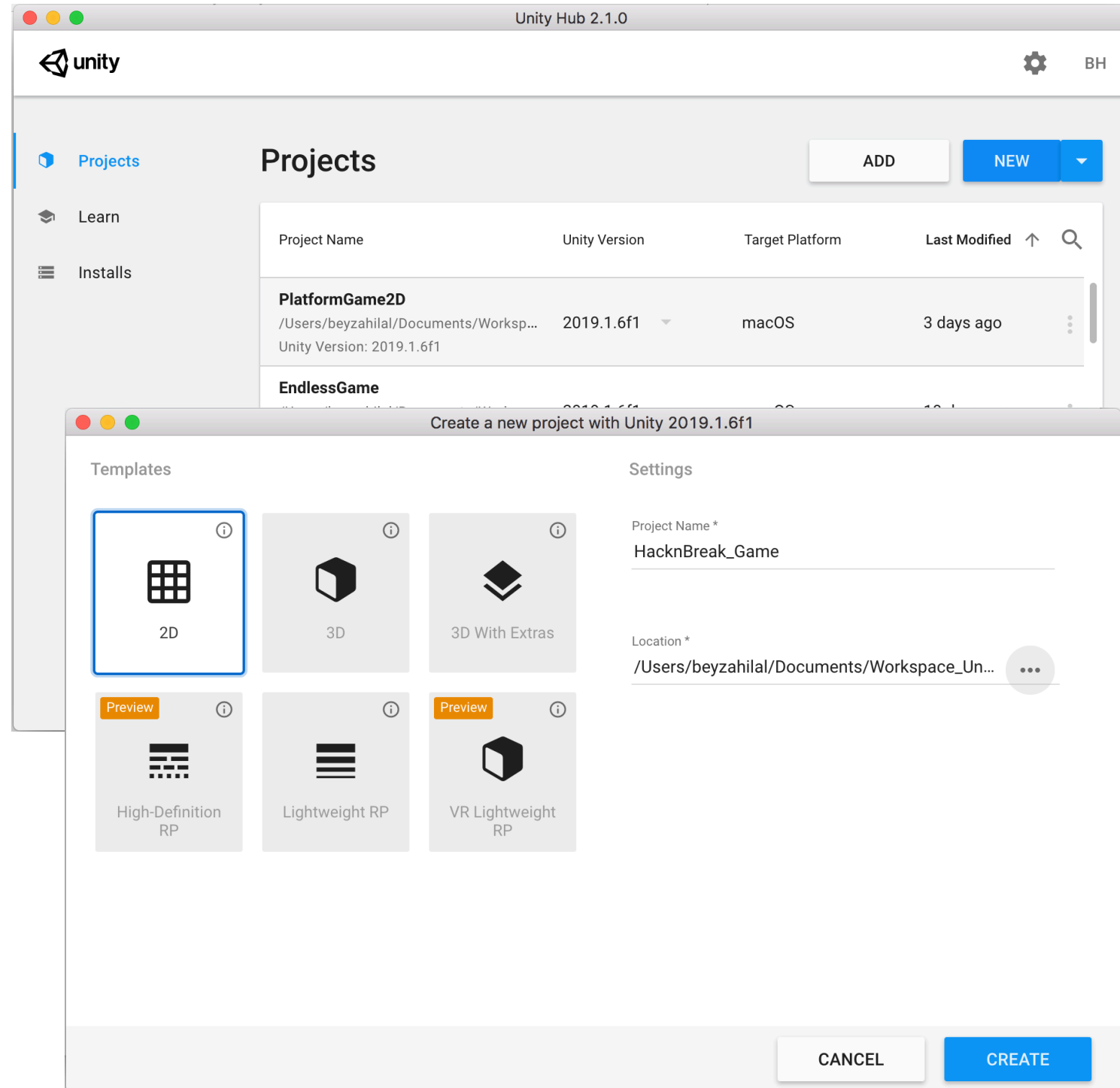
# Unity License

- Unity ID oluřturun.
- Unity Hub'a o ID ile giriř yapın.
- Unity Lisansı oluřturun.

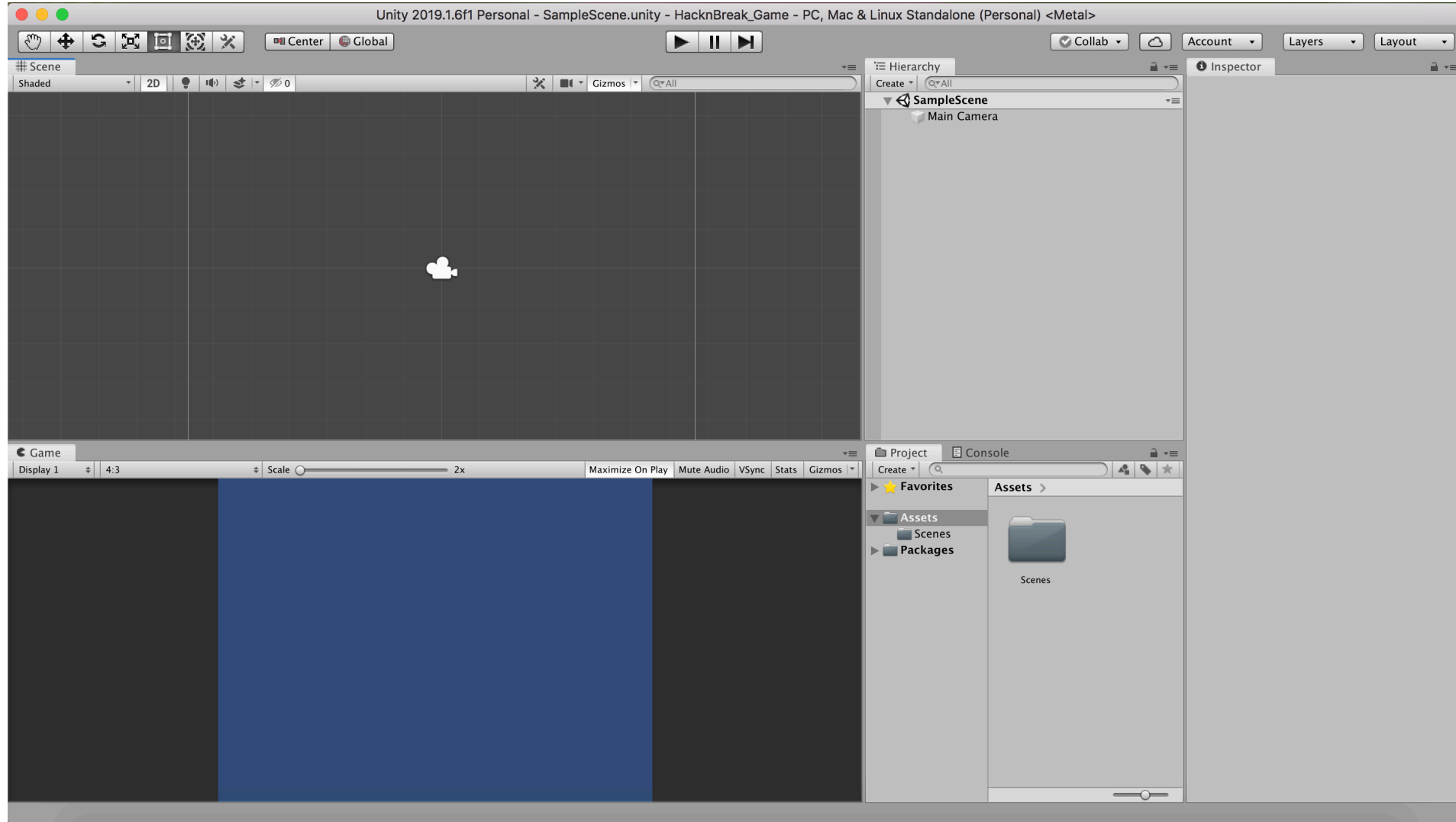


# Yeni Proje Oluřturma

- UnityHub'da Projects tab'ı altında projelerinizi görüp, yeni proje oluşturabilirsiniz.
- New'e tıklayıp proje ismi *HacknBreak\_Game* olan 2D yeni bir proje oluřturalım. Projenin konumunu istediğiniz şekilde ayarlayın.



# Yeni Proje Oluřturma



- Proje oluřtu artık beni bilgisayardan takip edebilirsiniz. İlerleyen zamanlarda kod paraları iin sunuma dnebiliriz.

1. Scene'i **Level1** adıyla kaydedelim. **SampleScene**'i silelim.
2. Component'leri tanıyalım; Scene, Game, Hierarchy, Project, Console, Inspector.
3. Game Object; Main Camera, New Sprite.
4. Transform Component. Vector3(x, y, z).
  1. x: Sağ-sol
  2. y: Yukarı-aşağı
  3. z: İleri-geri (3D: Derinlik, 2D: Sabit)



- **SpaceSampleAssets**'i bilgisayara indirip. Unity Projesinin içine kopyalayalım.

- **Player Game Object:** Empty Game Object yarat. Position'ı resetle. Sprite Renderer component ini ekle. Sprite için **Frigate**'i seç. Game objeyi **Player** olarak yeniden isimlendir. Player'a **BoxCollider2D** componentini ve **Rigidbody2D** component ini ekle. **Gravity**'i 0 yap. Ekranın altında olacak şekilde konumlandır.
- **Background Game Object:** Empty Game Object yarat. Position'ı resetle. Sprite Renderer component ini ekle. Sprite için **Background**'i seç. **Order in Layer**'i -10 yap. Game objeyi **Background** olarak yeniden isimlendir.
- **Astreoid Game Object:** Empty Game Object yarat. **Asteroid** olarak isimlendir. Position'ı resetle. Sprite Renderer componenti ekle, sprite için **Astreoid 5**'i seç. Ekranın üstünde olacak şekilde konumlandır. **BoxCollider2D** componentini ve **Rigidbody2D** component ini ekle. **Gravity**'i 0 yap.

# Player Script - Player Controller

- Project penceresinin içinde **Scripts** adında yeni bir klasör oluştur. PlayerController.cs adında C# dosyası oluştur. Player objesine PlayerController.cs componentini ekle.
- Hareket kontrolü ekleyelim. Speed'i 3 yapalım.

```
void Update()
{
    // Move
    if (Input.GetKey(KeyCode.RightArrow))
    {
        transform.Translate(Vector3.right * speed * Time.deltaTime);
    }

    if (Input.GetKey(KeyCode.LeftArrow))
    {
        transform.Translate(Vector3.left * speed * Time.deltaTime);
    }
}
```



# Asteroid Controller

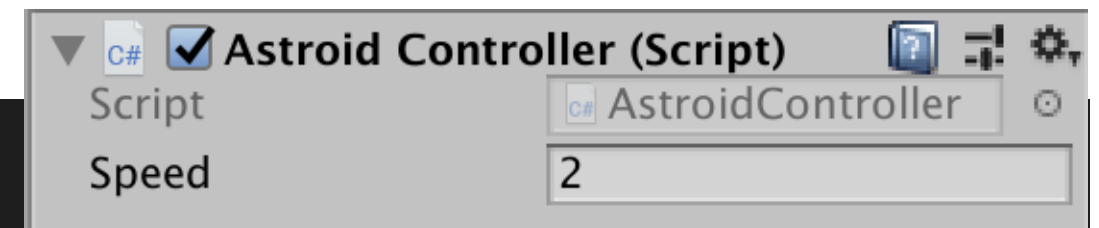
- Project penceresinde Script klasörünün içinde sağ tıklayıp ismi **AsteroidController** olan yeni c# dosyası oluşturalım. Bu script'i asteroid game objesine ekleyelim.
- Asteroid'e başlangıç yönü verip. Her frame de onu ilerletelim. Speed'i inspector den 2 yapalım.

```
public class AsteroidController : MonoBehaviour
{
    public float speed;

    private Vector2 direction;

    // Start is called before the first frame update
    void Start()
    {
        // direction = new Vector2(-1, -1);
        direction = new Vector2(Random.Range(-1.0f, 1.0f), -1); // Max 45 derecelik aci
    }

    // Update is called once per frame
    void Update()
    {
        transform.Translate(direction * speed * Time.deltaTime);
    }
}
```



# Prefabs

- Project penceresinde **Prefabs** isimli yeni bir klasör oluşturalım.
- Hierachry'deki Asteroid game objesini prefabs'ın içine atıp prefabs yapalım.

# GameController

- **GameController** isimli Empty Game Object yaratalım.
- **GameController.cs** isimli yeni bir script oluşturup GameController objesine ekleyelim.
- GameController objesi oyun yöneticisi olduğundan oyun boyunca değişmemeli ve tek bir tane olmalıdır.

```
public class GameController : MonoBehaviour
{
    public static GameController instance = null;

    void Awake()
    {
        if (instance == null)
        {
            instance = this;
        }
        else if(instance != this)
        {
            Destroy(gameObject);
        }
        DontDestroyOnLoad(gameObject);
    }
}
```

# AsteroidSpawner

- **AsteroidSpawner** isimli yeni bir script oluşturup GameController objesine component olarak ekleyelim.
- Bu script asteroidlerin belirli bir süre ara ile yaratılmasından sorumlu.

```
public class AsteroidSpawner : MonoBehaviour
{
    public GameObject AsteroidPrefab;
    public Vector2 SpawnPositionXRange;
    public float SpawnPositionY;
    public float WaitTimeForSpawn;

    // Start is called before the first frame update
    void Start()
    {
        StartCoroutine(SpawnAsteroid());
    }

    private IEnumerator SpawnAsteroid()
    {
        while (true)
        {
            if (!GameController.instance.isGameOver)
            {
                Vector3 position = new Vector3(Random.Range(SpawnPositionXRange.x, SpawnPositionXRange.y),
                    SpawnPositionY, 0);
                Instantiate(AsteroidPrefab, position, Quaternion.identity);
            }
            yield return new WaitForSeconds(WaitTimeForSpawn);
        }
    }
}
```

# Collision

- Player ve Asteroid game objelerine **Player** ve **Asteroid** olarak tag verelim.
- **Collision Detect:** Game objeler BoxCollider2D ve Rigidbody2D lere sahip oldukları için çarpışma durumlarını yakalayabileceğiz.
- Help document: <https://docs.unity3d.com/2019.1/Documentation/Manual/class-Rigidbody2D.html>

```
public class PlayerController : MonoBehaviour
{
    ...

    private void OnCollisionEnter2D(Collision2D other)
    {
        Debug.Log("Collision");
        if (other.gameObject.CompareTag("Asteroid"))
        {
            GameController.instance.GameOver();
            Destroy(other.gameObject);
            Destroy(gameObject);
        }
    }
}
```



# Game Over and Restart.

- Add GameOver functions and Restart function to GameController
- GameOver olunca time'i durduralım.

```
public class GameController : MonoBehaviour
{
    ...

    public void GameOver()
    {
        isGameOver = true;
        Time.timeScale = 0.0f;
    }

    public void Restart()
    {
        isGameOver = false;
        Time.timeScale = 1.0f;
    }
}
```

## Game Over and Restart.

- Game over yazısı için hierarchy'de **Text** UI elemanını oluşturalım. (UI > Text)
- Restart için hierarchy'de **Button** UI elemanını oluşturalım. (UI > Button)
- UI elemanları yaratınca gelen Canvas'a yeni bir script ekleyelim **UIController** isimli.

```
public class UIController : MonoBehaviour
{
    public GameObject StartButton;
    public GameObject GameOverText;

    void Start()
    {
        StartButton.gameObject.SetActive(false);
        GameOverText.gameObject.SetActive(false);
    }
    void Update()
    {
        if (GameController.instance.isGameOver)
        {
            StartButton.gameObject.SetActive(true);
            GameOverText.gameObject.SetActive(true);
        }
        else
        {
            StartButton.gameObject.SetActive(false);
            GameOverText.gameObject.SetActive(false);
        }
    }
    public void PressedStartButton()
    {
        SceneManager.LoadScene("Level1");
        GameController.instance.Restart();
    }
}
```

# Projectile

- Projectile isimli yeni bir game obje oluşturalım. Sprite Renderer ekleyelim, **Missile** sprite'ini seçelim. **BoxCollider2D** componenti ekleyelim. Positioni resetleyelim. **Projectile** diye tag verelim.
- Fire için **child** game object oluşturalım. **SpriteRenderer** ve **BoxCollider2D** ekleyelim. Sprite için **ProjectileSharp** seçelim. **Color**'ı kırmızı yapalım.
- Mermi hareketini sağlayacak **ProjectileController** isimli controller oluşturalım. Speed'i inspector den 5 olarak ayarlayalım.
- Projectile game objesini **Prefabs** klasörünün altına atalım.

```
public class ProjectileController : MonoBehaviour
{
    public float speed;

    // Update is called once per frame
    void Update()
    {
        transform.Translate(Vector3.up * speed * Time.deltaTime);
    }
}
```

# Fire

- PlayerController.cs ye fire kontrolünü ekleyelim.

```
public class PlayerController : MonoBehaviour
{
    public float speed;
    public GameObject BulletPrefab;

    // Update is called once per frame
    void Update()
    {
        // Move
        ...

        // Fire
        if (Input.GetKeyDown(KeyCode.Space) || Input.GetKeyDown(KeyCode.UpArrow))
        {
            Vector3 position = transform.GetChild(0).GetComponent<Transform>().position;
            Instantiate(BulletPrefab, position, Quaternion.identity);
        }
    }
    ...
}
```

# Destroy Asteroid

- Mermi asteroid'e çarparsa ikiside yok olacak.
- Bu çarpışmayı anlamak için **trigger** kullanacağız. Projectile'nin **BoxCollider2D**'sindeki **Is Trigger**'i true olarak seçmeliyiz. Rigidbody2D eklemeye gerek yok.

```
public class AsteroidController : MonoBehaviour
{
    ...
    private void OnTriggerEnter2D(Collider2D other)
    {
        if (other.gameObject.CompareTag("Projectile"))
        {
            Destroy(other.gameObject);
            Destroy(gameObject);
        }
    }
}
```

# Score

- Asteroidleri parçaladıkça score kazanalım.
- Score değişkenini GameController.sc scripti içinde oluşturalım.

```
public class GameController : MonoBehaviour
{
    public static GameController instance = null;
    public bool isGameOver;

    public int Score;
    private int asteroidPoint = 10;

    ...

    // Start is called before the first frame update
    void Start()
    {
        Score = 0;
        isGameOver = false;
    }
    public void IncreaseScore()
    {
        Score += asteroidPoint;
    }
    ...
}
```

# Score

- **AsteroidController.cs** de ilgili yerde **IncreaseScore** fonksiyonunu çağıralım.

```
public class AsteroidController : MonoBehaviour
{
    . . .
    private void OnTriggerEnter2D(Collider2D other)
    {
        if (other.gameObject.CompareTag("Projectile"))
        {
            GameController.instance.IncreaseScore();
            Destroy(other.gameObject);
            Destroy(gameObject);
        }
    }
}
```

# Score

- Score için Hierarchy'de **Text** UI objesi yaratalım.
- UIContoller.cs bağlayalım. Her update de ekrana mevcut score'u yazdıralım.

```
public class UIController : MonoBehaviour
{
    public GameObject StartButton;
    public GameObject GameOverText;
    public Text ScoreText;

    . . .

    // Update is called once per frame
    void Update()
    {
        ScoreText.text = "Score: " + GameController.instance.Score;
        . . .
    }

    . . .
}
```

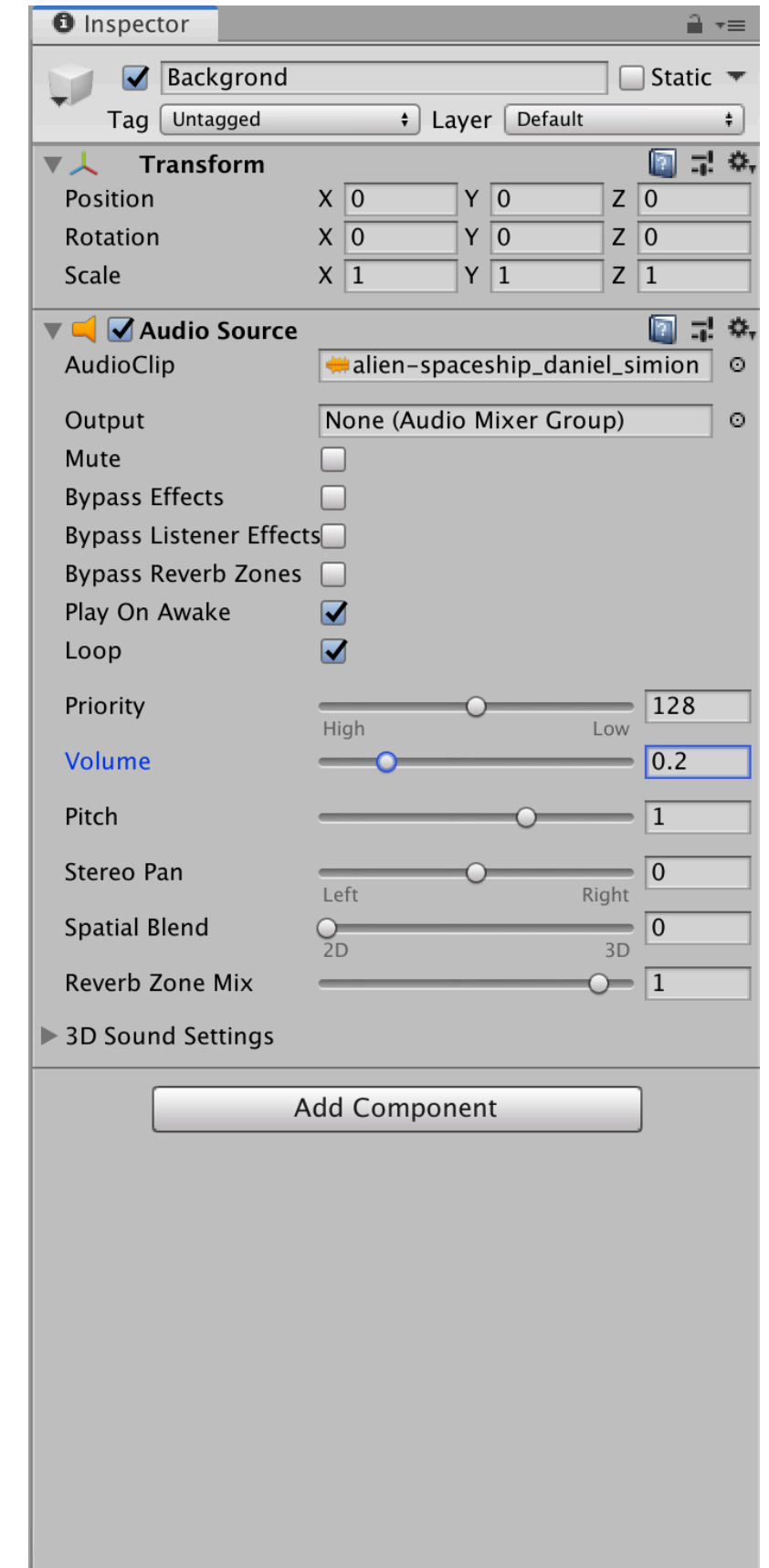


# Sounds

- **Sounds** isminde yeni Empty Object yaratalım.
- **Background** isimli child empty object yaratalım. **AudioSource** componentini ekleyelim. **AudioClip** olarak **alien-spac...** seçelim. **Loop**'u ve **PlayOnAwake**'i **true** yapalım.
- **Collision** isimli child empty object yaratalım. **AudioSource** componentini ekleyelim. **AudioClip** olarak **collision**'i seçelim. **PlayOnAwake**'i **false** yapalım.
- **Fire** isimli child empty object yaratalım. **AudioSource** componentini ekleyelim. **AudioClip** olarak **laser8**'i seçelim. **PlayOnAwake**'i **false** yapalım.
- **Explosion** isimli child empty object yaratalım. **AudioSource** componentini ekleyelim. **AudioClip** olarak **Explosion**'i seçelim. **PlayOnAwake**'i **false** yapalım.

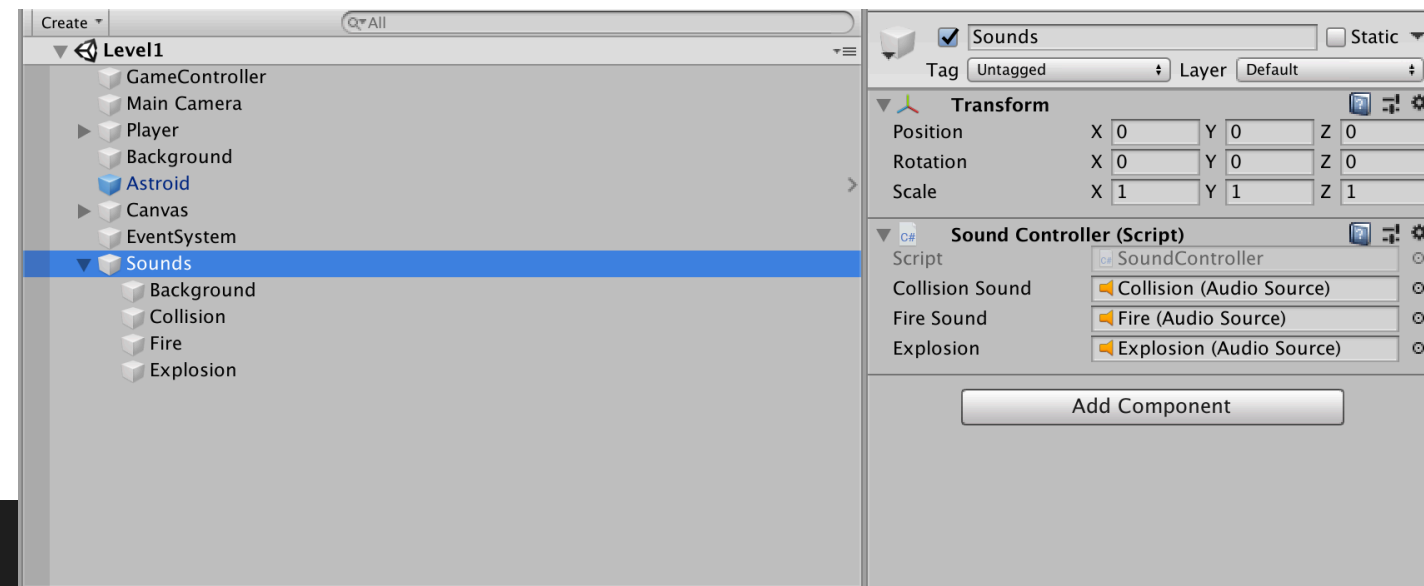
# Background sound

- <http://soundbible.com/2213-Alien-Spaceship-UFO.html>
- Background volume 0.2 yapalım.



# Create SoundController Script.

- Static instance
- Play fonksiyonu efektler için.



```
public class SoundController : MonoBehaviour
{
    public static SoundController instance = null;
    public AudioSource CollisionSound;
    public AudioSource FireSound;
    public AudioSource Explosion;
    private void Awake()
    {
        if (instance == null)
        {
            instance = this;
        }
        else if (instance != this)
        {
            Destroy(this);
        }
        DontDestroyOnLoad(gameObject);
    }
    public void PlayCollisionSound()
    {
        CollisionSound.Play();
    }
    public void PlayFireSound()
    {
        FireSound.Play();
    }
    public void PlayExplosionSound()
    {
        Explosion.Play();
    }
}
```

# Sounds

- İlgili yerlerde efektleri oynatalım; atış yapılırken, asteroid mermiye çarpınca, asteroid player a çarpınca.

```
public class AsteroidController : MonoBehaviour
{
    . . .
    private void OnTriggerEnter2D(Collider2D other)
    {
        if (other.gameObject.CompareTag("Projectile"))
        {
            SoundController.instance.PlayExplosionSound();
            GameController.instance.IncreaseScore();
            Destroy(other.gameObject);
            Destroy(gameObject);
        }
    }
}
```

```
public class PlayerController : MonoBehaviour
{
    . . .
    void Update()
    {
        . . .
        if (Input.GetKeyDown(KeyCode.Space) || Input.GetKeyDown(KeyCode.UpArrow)) // Fire
        {
            Vector3 position = transform.GetChild(0).GetComponent<Transform>().position;
            Instantiate(BulletPrefab, position, Quaternion.identity);
            SoundController.instance.PlayFireSound();
        }
    }
    private void OnCollisionEnter2D(Collision2D other)
    {
        if (other.gameObject.CompareTag("Asteroid")) // GameOver
        {
            SoundController.instance.PlayCollisionSound();
            GameController.instance.GameOver();
            Destroy(other.gameObject);
            Destroy(gameObject);
        }
    }
}
```

# Destroy Asteroid

- Ekranın dışına çıkan asteroidleri yok edelim.
- **AsteroidDestroyer** isimli empty game object oluşturup ekranın aşağısına yerleştirelim. **BoxCollider2D** ekleyelim. **IsTrigger**'i **true** yapalım. **Outside** diye tag verelim.
- AsteroidController.cs de trigger olunca destroy edelim.

```
public class AsteroidController : MonoBehaviour
{
    . . .

    private void OnTriggerEnter2D(Collider2D other)
    {
        . . .
        else if (other.gameObject.CompareTag("Outside"))
        {
            Destroy(gameObject);
        }
    }
}
```

Tamamlandı.

Build and enjoy.

:)

Beyza Hilal Durak

[www.beyzahilaldurak.com](http://www.beyzahilaldurak.com)  
[beyzahilaldurak@hotmail.com](mailto:beyzahilaldurak@hotmail.com)