**UNITY SHORTCUTS**

CTRL+SHIFT+F seçilen objeyi bulunduğumuz konuma getirme (Özellikle **Camera** için önemli)

SHIFT+SPACE maximize the selected scene

**Sahne**

SagMB + wasd = sahnede gezinme Alt+ sagmb = smooth zoom in&out

Hierarchyde secilen objeyi bulmak için f tuşuna bak.

**Obje**

Ctrl+ d = seçilen objeyinin kopyasini uretir.

Ctrl basarak hareket ettirirsek belirli oranda move rotate etc yapabiliriz. ( Bu oran Edit-> snap settings den ayarlanabilir ).

Obje seçiliyken v tuşuna basarak köşeler seçilir.

**Terrain**

Fırçadan büyükme yapılır shifte basarak küçültme yapılır.

Terrain->Set height, tan belirli yüksekliğe kadar boyama yapılır eğer kısa daha farklı boyda boyama yapılıcaksa shifte basılıp o alandan ref alınır.

Ilk eklenen layer tum terraini kaplar daha sonrakiler ile boyama yapılabilir. Tiling Settingsteki size ayari ile döşeme boyutu değiştirilir.

**Materials**

Material bir objeye sürükleyip bıraktığımızda o objenin nasıl görüneceğini anlatır bize.

Albedo = seçeneği renk olarak görünmesini istediğimiz.

Normal map = Detaylandırma için seçtiğimiz

Occlusion = isigin hangi bölgeleri aydinlatip hangilerini aydinlatmayacagi ile alakali

**Collider**

Basitce collider fiziksel olarak etkileşime girmez ama içine obje girdiğinde algılayabiliriz. Objelerin birbiriyle çarpması, etki alanına girmesi vs gibi konularda kullanılır.

**Rigid Body**

Objelerin fiziksel etkileşime girmesi & yer çekiminden etkilenmesi için kullanılır.

Is kinematic = fiziksel olarak etkilenmesini kapatip/açma. Koddan bunu başlat bitir yapılabilir.

**PreFabs**

Prefabin inspector sekmesinden Overrider kısmında apply dersek değişiklikler tüm prefablara revert dersek orj prefabin özelliklerini objelere uygulara.

Prefeb Variant = Bir objenin variantini üretmek orj prebain özelliklerini alır sadece ezdiği (yeni eklenen materyal gibi özellikler kendine has kalır.) Ezmediği özellikler ana prefabden alır.

**ProGrid**

Tools->ProGrid Window. Eksenleri kullanarak daha kolay objeleri konumlandırmaya yardım ediyor.

Edit için; Edit->Preferences->ProGrids.

**Light**

Işıkların modunu değiştirerek real time mixed ya da baked yapılabilir.

Real time ışıklar bir objenin yeri degişse dahi ışık ona göre şekillenir fakat performanstan kayıp olur. Işık her saniye başına hesaplanır.

Sürekli hareket etmeyen cisimlerde baked light kullanmak performans kazandırır.

Bir objenin baked lighttan etkilenmesini istiyorsak, inspector penceresinden objeyi static yada light map static yapmalıyız. Static olmayan objeler ışıktan hiç etkilenmez.

Indirect Multiplier: Işıgın duvardan yansıma sayısı.

Baked ışık sadece statik objelerin üzerine vurur. Işıkları baked yapmaya çalış hiç olmadi mixed yap.

**Particle**

Unity particle pack önemli! Indir

Particle objesi seçiliyken emissionun altindan ne kadar yoğunlukta olusacagini belirlemede bullanılır. Shape bildiğimiz particle yayilma sekli. Externel forces kullanarak rüzgar gibi seylerden etkilenmesini sağlayabiliriz…

Eklenen particklenin arka planini kaldırmak istiyorsak, Particle nin alpha sourcesini None->From Gray Scale yapmalıyız.

Material olarak partickle eklemek için: Material ekledikten sonra Shader kısmını Particles->Standard Unlit. Sectikten sonra Rendering Modeye Fade seçersek seçilen resmin arka planı silinik olarak particleye ekler.Color Mode dan particleler arası geçiş nasıl olsun istiyorsak onu seçiyoruz. Albedoya da istediğimiz particle resmi konur.

Particlenin Size over lifetimeden ilerleyişindeki boyut ayarını yapabiliriz. Sağ tık ile değişmesini istediğimiz nokta ekleyebiliyoruz grafik üzerinden.

**Navmesh (Bir nesnenin bir yerden diğer yere en kısa yolu kullanarak gitmesi) (AI)**

Window->AI->Navigation açtıktan sonra navigation sekmesinden bake ayarına geliyoruz. Burada karakterin boy genişlik tırmanma yük. Tırmanma eğimini tanımlayabiliyoruz.

Belirli bir alanı navmeshe kapamak istiyorsak, inspectordan navigation staticigini kapatırız. Yada Navigation sekmesinden objeyi seçtikten sonra walkableyi not walk. Yapabiliriz.

Navigation sekmesinden Areas kısmında belirli pathleri AI için daha maliyetli yol olarak atayabiliriz ve en kısa yolu seçerken bunları göze alır. (Water 5 gibi) Areastan ayarlandıktan sonra objeye o tag Navigation sekmesindeki Object içinden “Navigation Area” sını ayarladığımız (Water) seçmeliyiz.

Eğer düşmanın içinden geçilmesini istemiyorsak= Düşman olacak objeye component olarak “Nav Mesh Obstacle” eklemeliyiz ve Carveyi seçmeliyiz. Bu sayede düşman objesi hareket ettirildiğinde objemiz onun içinden geçmiyor olucak.

Nav Mesh Agent i playerimizin inspectoruna component olarak ekledikten sonra hareket scriptimizide ekliyoruz daha sonra nav mesh agent kısmından hız vs gibi ayar verebiliyoruz.

**Animation**

Animasyon için öncelikler animation klasörüne createden “animatör controller” oluşturmalıyız. Sonrasında Animator sekmesinden sağ tıklayıp yeni stateler ekleyebiliriz.

Daha sonrasında istediğimiz stateyi seçerek inspector sekmesinden motion kısmına istediğimiz animasyonu ekleyebiliriz.

En son ayarlandıktan sonra playerin inspectorundaki animatör->Controller kısmına ayarladığımız animatör controlleri eklemekliyiz.

Bir animasyondan diğerine geçmek için base stateye sağ tıklanıp make transition yapılır.

Condition ve Transition kısmında ise, Animator->Parameterstan float eklenir. Daha sonra animasyonlar üzerindeki oka basılıp inspector penceresindeki conditions kısmı ayarlanır.

Has Exit Time seçili ise transitiondan önceki animasyon tamamlanana kadar bekler o yüzden bu seçili olmamalı.

Bu düzenlemelerden sonra son olarak gerekli scripti eklemeliyiz.

Eğer kademeli bir animasyon üretilecekse, animatör kısmına Create State-> From New Blend Tree diyip oluşturulan stateye çift tıklanıp içine girilir. Daha sonrasında ise blend tree ye tıklayarak parameter->motion kısmından kullanılmak istenen motionlar eklenir.

Daha sonrasında automate threshold tiki kaldırılarak, istenilen threshold aralıklarında, istenilen animasyonun görünmesi sağlanabilir.

Eğer blend tree varsa animatördeki önceki stateler yerine bunun içindeki kullanılan stateler yeterli olur.

**Kendi Animasyonumu Olusturma;**

1- Öncelikle window->Animation penceresi açılır. Sonrasından Animasyonunu yapmak istediğimiz öğe seçiliyken create denilip yeni bir animasyon dosyası oluşturulur. (Assests/Animations içine)

2-Sonrasında “Add Property” diyip animasyonunu yapmak istediğimiz özelliği (mesela position) i seçiyoruz.

3- Time line da animasyonun gerçekleşmesi istenilen süreye tıklanır. [0:25 gibi]

4- Play animation kısmının sağ alt çaprazındaki “◊+” işarete (Add keyframe) basılır. (İstenilen süre seçiliyken) [ O sürenin altına ◊ işareti eklenecek ]

5- Preview kısmındaki Kırmızı yuvarlak kayıt tuşuna basılır. Ve objemizin seçilen saniyedeki bulunmasını istediğimiz konum/ hareket / özellik ayarlanır.

6-Kayıt tuşuna tekrar basılarak kayıt durdurulur ve bu sayede kaydedilmiş olunur. Animasyonu kontrol etmek için play tuşundan animasyonun süreye göre durumu izlenebilir.

7-Eğer ayni animasyonun yanında başka animasyonlarda gerçekleşmesini istiyorsak tekrardan add property diyip önceki özelliğin altına ekleme yapılır. (2. Adım ve sonrası tekrarlanır)

8-Eğer animasyon işlemleri bittiyse objemizin inspector penceresinden animatör kısmı düzenlenir.

**Sound**

Unityde sahnede sesi dinleyen bir audio listener olmaz zorunda. Bu sahneden gelen tüm sesleri dinliyip hoparlör /kulaklığa gönderen componenttir. Genelde kameranın üstünde olur.(Default olarak, yoksa kameraya ekle)

Empyt object oluşturulup Audio Source componenti eklenir ve AudioClipe istenen ses verilir ve özellikleri ayarlanır.

Eğer tüm oyunda değil de sadece bir objeye yaklaştığımızda belirli bir sesi duymak istiyorsak;

Söz konusu objeye audio source eklenir, ses ayarlanır ve “Spatial Blend” özelliği 1 yapılır(En sağa çekilir) . Bu sayede kamera objeye yaklaştığında atanan ses duyulacaktır.

Sesleri konumumuza göre bir kısmını kısıp bir kısmını arttırmak için, Assetlerin içine create ile “Audio mixer” oluşturulur ve açılır.

Groupsun sağındaki + işaretinden alt gruplar eklenir. Çıkmasını istediğimiz ses konuları için grup oluşturulur.

Daha sonra audioclip eklediğimiz objelerimizde seçtiğimiz seslere uygun olarak Inspector->Audio Source->output a gruplamalardaki uygun grup adı atanır. Gruplar atandıktan sonra Audio Mixerden tüm atanan ses seviyeleri ayarlanabilir.

Eklenen grubun altındaki add kullanılarak audio effect verilebilir.

Birkaç kanala birden ses efekti vermek istiyorsak : Yeni bir alt grup eklenir bu grubun Inspector penceresinden “Add effect->Receive” seçilir. **[ Bu grup sound effekleri burunduran grup olacak ].** Bu effekti kullanacak diğer alt gruplara ise “Add effect -> send” diyerek alacağı sound effekt grubu seçilir.

**Arayüz (UI)** Create->UI->Canvas

Eğer canvasın Render mode u “screen space-overlay” seçili ise, ekranı otomatik olarak kaplar. -camera seçili kameranın tüm ekranını kaplaması için kullanılır. “World Space” rect transform ayarlanmasına göre, canvası ekranın her hangi bir kısmına konumlandırabiliriz.

Canvas Scaler kısmındaki UI Scale Modedaki “Constant pixel size” canvasa konan belirli ölçüdeki image gibi şeylerin boyutu aynı kalır ekran boyutu değişse bile. “Scale with Screen Size” ise ekran çözünürlüğü ile bunu scale eder yani ekran boyutuna göre belli bir oranda büyür yada küçülür.

UI elementlerini buton gibi tıklanabilir/etkileşilebilir yapma ayarı ise “graphic raycaster”dan yapılır.

Canvas içerisindeki componentlerin rect transform kısmından ekrana göre hizalama ayarı yapılır. Ne kadar sağda ortada veya merkez duracağı gibi. Center – Middle croosu acık iken alt ve shift kullanarak farklı func. kullanılabilir.

Eğer sabitlenmesini istediğimiz yerin **pivot noktasını değiştirmek istiyorsak.** Qwer kısmının sağından centeri pivot yapıp, pivot noktanın(Mavi çember) sahne üzerinde hareket ettirilmesi gerek.

Eğer canvası çeşitlendirmek istiyorsak Hierarchy de canvasa sağ tıklıyıp UI dan istediğimiz özelliği ekleyebiliriz.

**Scripting**

Önemli kod parcalari;

a = GetComponent<x>( ); //Scriptin ekli olduğu objenin inspector icersindeki x componentine erişir ve onu a yi kullanarak ayarlamamıza olanak tanır.

**Bir scriptten baska bir objeyi/scripti nasıl bulurum onu nasıl değiştiririm;**

GameObject x = GameObject.Find(“asd”); //Sahnedeki asd adındaki objeyi bulup onu x variablesine atayıp üzerinde işlemler yapılabilir.

x.GetComponent<qwe>().mm = a; // Burada x e atanan objenin inspector penceresindeki qwe ögesindeki mm’ine eriştik ve onu a ile değiştirdik.

GameObject[] asdsss = GameObject.FindGameObjectsWithTag(“asd”); //asd tagine sahip tum objeleri bulup asdsss adli arrayin içine atar.

GetComponenet ve findlardan Update içindeyken kacinmaya calis yoksa fps sorunu yasanir.

if ( Input.GetKeyDown(KeyCode.E)) { } // E tuşuna basildiginda if in içine girer.

GetKey -> Basili tutuldugu surece.

GetKeyDown -> Basildigi anda demek.

GetKeyUp -> Elimizi cektigimiz zaman.

if ( Input.GetMouseButtonDown(0) { } //Sol MB clicke her basildiginda ife girer.

GetMouseButtonKey -> Basili tutuldugu surece.

GetMouseButtonDown -> Basildigi anda demek.

GetMouseButtonUp -> Elimizi cektigimiz zaman.

Instantiate(gameObject); //Sahnede ayni objeden bir tane daha üretir.

if ( Input.GetButton(“Run”)) { } //Run tuşuna basildiginda bu işlemi gerçekleştirir.

**Run tuşunu ya da herhangi bir x buttonu atamak için;**

**Edit->Project Settings->Input Manager->Axes->** diyip **Size** i 1 arttırılır. Daha sonrasinda atamak istediğimiz buttonun ismi (Run) yeni oluşan yerin **Name** bölümüne girilir ve kullanmak için atanacak tuş “**Positive** **Button**” kısmına girilir.

private void OnCollisionEnter(Collision XYZcollision) {

Debug.Log(“Main Obj:” this.gameObject.name + “ hitted by:” XYZcollision.gameObject.name);

} //Obje ile collision (Temas eden obje) arasindaki func. Debug satirinda obje ve colliosionun adini alip basıyor.

OnCollisionEnter -> Bir collision bu objeye değdiği/içine girdiği zaman aktif olur.

OnCollisionExit-> Bir collision bu objeye içinden/çarpısmadan çıktıgı zaman aktif olur.

OnCollisionStay -> Bir collision bu objenin içinde kalmayi/carpismayi sürdürdüğü sürece aktif olur.

**->İki objenin birbiri ile çarpışması için en az birinde rigidbody olmak zorunda.**

private void OnTriggerEnter (Collider other) { } //Belirli bir collider alanına girildiği an bu func calisir.

private void OnTriggerExit (Collider other) { } //Belirli bir collider alanından çıkıldıgı an bu func calisir.

Raycast: Scriptimizi attigimiz objeden bir lazer tutmamıza yarar ve bu carptigi objenin bilgisini alir.

Debug.DrawRay(transform.position, transform.forward \* range , Color.cyan); //Bu kod range boyutu uzunluğunda objenin forwardina doğru(z ekseni) ,cyan renkli ray çizer.

RaycastHit hit;

İf (Physics.Raycast(transform.position , transform.forward, out hit, range)) {}

//Eğer transformumun pozisyonundan objenin ilerisine doğru range boyutunda bir şeye çarparsa bunu hit e at ve ife gir.

**Obje Oluşturma ve yok etme**

**Spawn**

Vector3 randPosition = new Vector3 ( UnityEngine.Random.Range(-25,25) , Unity.Engine.Random.Range(2,10), UnityEngine.Random.Range(-25,25));

//Pozisyon oluşturma kodu, parantez içerisindeki x y z in rasgele belirlenen konumunda bir pozisyon ataması yaptık.

List<GameObject> boxes= new List<GameObject>();

…

GameObject box =Instantiate(box,Object,randPosition,Quaternion.identify);

boxes.Add(box);

//Objemizi gidip o random pozisyonda default rotation da oluşturup boxa attik ve ardindan Lisimize bu boxi ekledik.

**Destroy**

Destroy(boxes[boxes.Count -1]); //Listedeki son elemanı yok/imha eder.

boxes.RemoveAt(boxes.Count-1); //Listeki son elemanı silip, rangeni küçülttük bu sayede son eleman null görünmeyecek.

Count: Listedeki toplam eleman sayisi ( son index olması için yukarıdaki örnekte 1 cıkardık)

Bunlar yerine box.SetActive(true)/(false) yaparak object pulling yapılabilir. Objeleri active edip kapatarakta bu işlem yapılabilir. Performans bu sayede daha iyi olur.