



Spelmotor integration för byggandet av 3D-miljöer inom filmbranschen

Oliver Backman

Lärdomsprov

Informationsteknik

2024

Lärdomsprov

Oliver Backman

Spelmotor integration för byggandet av 3D-miljöer inom filmbranschen.

Yrkeshögskolan Arcada: Informationsteknik, 2024.

Identifikationsnummer:

8380

Sammandrag:

Detta arbete undersöker integration av spelmorteknik för byggandet av 3D-miljöer och VFX-effekter inom den moderna filmbranschen. Arbetet studerar de två kändaste spelmotorerna Unreal Engine 5 och Unity, och gör en analyserad jämförelse av bägge spelmotorer för att svara på frågan att hurudan filmproduktion tillämpar sig vilken spelmotor bättre för. För att uppnå dessa mål gjordes extensiv undersökning inom internetet i form av artiklar och intervjuer, samt även intervjuades en aktör inom den finska teaterbranschen gällande fysisk scenografi. Det skapades två stycken kortfilmer med Unreal Engine 5 och Unity och resultatet jämfördes först objektivt med kriterier relevanta för detta arbete och avslutades med subjektiv reflektion över arbetet och dess resultat.

Nyckelord: Unreal Engine 5, Unity, Spelmotor, VFX, 3D-miljö.

Degree Thesis

Oliver, Backman

Title. Game engine integration for building 3D environments in the movie industry.

Arcada University of Applied Sciences: Information Technology, 2023.

Identification number:

8380

Abstract:

This work examines the integration of game engine technology for the construction of 3D environments and VFX effects within the modern film industry. The study explores the two most well-known game engines, Unreal Engine 5 and Unity, and provides an analyzed comparison of both game engines to answer the question of what kind of film production is which game engine better suited for. To achieve these goals, extensive research was conducted on the internet in the form of articles and interviews, and a representative from the Finnish theater industry was interviewed regarding physical set design. Two short films were created using Unreal Engine 5 and Unity, the results were first objectively compared based on criteria relevant to this work and in the end subjectively reflected upon.

Keywords: Unreal Engine 5, Unity, Game engine, VFX, 3D environment.

Innehåll

1	Introduktion	6
1.2	Syfte.....	6
1.3	Metoder.....	7
1.4	Avgränsningar.....	7
1.5	Struktur.....	7
2	Teoretisk bakgrund.....	8
2.2	Spelmotor	8
2.2	Unity Bakgrund	8
2.3	Grundläggande tekniska detaljer Unity.....	9
2.4	Unity styrkor och svagheter	10
2.5	Unreal Engine bakgrund	11
2.6	Grundläggande tekniska detaljer Unreal Engine 5	11
2.7	Unreal Engine 5 styrkor och svagheter.....	13
3	Spelmotorintegration inom filmindustri	14
3.1	StageCraft	14
3.2	Fördelar med LED-rutor gentemot fysisk scenkonstruktion	15
3.3	VR teknologi och virtuell miljö i samband med traditionell filmning.....	16
3.4	Intervju med Peter Ahlqvist	16
4	Att göra en kortfilm med Unreal Engine 5	18
5	Att göra en kortfilm med Unity	36
6	Resultat	52
6.1	Jämförelse av filmresultaten - Resultat	52
7	Egna funderingar efter att ha använt Unity och UE	58
8.	Slutsatser	61
Källor	63

1 Introduktion

Under de senaste åren har det skett stora framsteg inom spelbranschens utveckling av spelmotorer. Stora aktörer inom branschen som till exempel Unity Technologies och Epic Games har i samband med tekniska utvecklingar skapat plattformar i vilka både företag av varierande storlekar och privatpersoner kan utnyttja företagets spelmotorer kostnadsfritt för diverse projekt. De tekniska möjligheterna med spelmotorerna har även noterats av dagens filmindustri. Aktörer i filmbranschen som till exempel Disney, har under de senaste tre åren arbetat flitigt med att integrera spelmotorteknik för sina egna produktioner. Kombinationen av tids- och kostnadseffektivitet samt strävan att skapa ökad realism i specialeffekter, bättre och billigare, vilket görs möjligt med de moderna spelmotorerna, väcker mycket stort intresse bland de olika aktörerna i den gigantiska filmindustrin.

I det här forskningsarbetet kommer fokuset att ligga vid analysen av de två kändaste spelmotorerna Unity och Unreal Engine 5. Vad är de två spelmotorernas styrkor och svagheter i områden gällande skapandet av 3D-miljöer, kostnads- och tidseffektivisering inom filmbranschen, tekniska funktioner relevanta för filmning och för hurudan typ av filmproduktion tillämpar sig de bägge spelmotorerna.

1.2 Syfte

Huvudsyftet med forskningsarbetet är att studera de två spelmotorernas, "Unity" och "Unreal Engine 5", tekniska egenskaper, kostnads- och tidseffektivisering inom filmproduktion i jämförelse med traditionella scenografi metoder som till exempel "Green Screen" och fysisk scenografi skapande, samt spelmotorernas skillnader gällande skapandet av 3D-miljöer för filmbranschens scenografi. Målsättningen är att i ljuset av denna forskning få ett svar på hurdana filmprojekt som spelmotorerna i fråga bäst lämpar sig för.

1.3 Metoder

Att skapa en god helhetsbild gällande spelmotorer och deras integrering inom filmbranschen i kontrast till tidigare metoder som fysisk scen byggande. En intervju om scenografi i praktiken inom teatervärlden. En jämförelse mellan Unity och Unreal Engine 5 som stöds med material från diverse forskningar, litteratur, intervjuer, artiklar och informella videor gällande ämnet. Ett fåtal experiment och undersökningar i vilka man kommer att jämföra kvalitetsskillnader och skillnader i användarupplevelsen i byggande av 3D-miljöer med Unreal Engine 5 och Unity.

1.4 Avgränsningar

Eftersom det finns flera olika spelmotorer av ett flertal utvecklare så kommer forskningsarbetet att avgränsa sig till analysen av de två mest kända spelmotorerna "Unity" och "Unreal Engine 5". Exempel på deras integrering inom den moderna filmbranschen kommer att hantera praktisk tillämpning framom det rent tekniska, d.v.s. samarbetet mellan filmregissören och mjukvaran, hur det tekniska sammansmälter ihop med det praktiska är i större fokus än det rena tekniska mjukvarans analys. I de avsnitten som går igenom tekniska egenskaper så avgränsas information som inte är relevant för detta arbete. För att förenkla och förkorta texten kommer jag att använda förkortningen UE5 när jag hänvisar till Unreal Engine 5.

1.5 Struktur

Kapitel 2 ger en grundlig teoretisk bakgrund om vad spelmotorer är och en grundlig bakgrund om Unity och Unreal Engine 5 spelmotorerna. Kapitel 3 handlar om hur spelmotorer har integrerats inom den moderna filmbranschen, vilka är det viktigaste påverkande faktorerna för denna integrering och kapitlet avslutar med en intervju gällande VFX och olika faktorer gällande fysisk scenkonstruktion. Kapitel 4 och 5 handlar om att skapa scener med 3D-miljöer och sedan inspela kortfilmer i både UE5 och Unity. Kapitel 6 är en jämförelseanalys av filmresultatet skapat i kapitel fyra och fem. Kapitel 7 är en jämförande analys av UE5 och Unity baserat på min användarupplevelse och de miljöer jag har skapat i bägge spelmotorerna. Kapitel 8 är slutsatser och reflektion av arbetet.

2 Teoretisk bakgrund

Det här kapitlet är en genomgång om vad en spelmotor är, bakgrundsinformation om de två spelmotorerna Unity och UE5, de tekniska egenskaperna av bägge spelmotorerna som är relevanta för detta arbete och spelmotorernas styrkor och svagheter.

2.2 Spelmotor

Spelmotorer är mjukvara i vilka bland annat spelutvecklare kan skapa och utveckla videospel inom en virtuell miljö. Inom den virtuella miljön har användaren tillgång till en mångfald av verktyg som stöder utvecklandet av spellogik, grafik, fysiklagar, filmning och överföring till olika spel plattformar med mera.

En spelmotor innehåller fem komponenter: Det huvudsakliga spelprogrammet som innehåller spel logiken; en framställnings motor för att skapa 3D-animerad grafik; en ljudmotor som består av ljud relaterade algoritmer; en fysikmotor för att implementera fysiklagar inom systemet; och en modul för artificiell intelligens (Interestingengineering, 2016). Spelmotorerna skiljer sig från varandra i både användarvänlighet samt tekniska detaljer och aspekter, beroende på hurudant projekt och slutmål som man vill åstadkomma så tillämpar sig olika spelmotorer för olika ändamål bättre. Till exempel en spelmotor kan lämpa sig bättre för att skapa 3D-spel medan en annan spelmotor lämpar sig bättre för 2D-spel, olika spelmotorer använder sig av olika programmeringsspråk vilket också påverkar på hela arbetsprocessen. Även det visuella, d.v.s. verktygen för att skapa visuella element som spelgrafik kan ha mycket stora skillnader som det till exempel är fallet med Unity och UE5.

2.2 Unity Bakgrund

Unity är en 2D- och 3D- spelmotor publicerad år 2005. Utvecklad av Unity Technologies, mjukvaran var skapad för att förse fler utvecklare tillgång åt spelutvecklings verktyg, vilket var ännu på den tiden något nytt och ovanligt. Unity spelmotorn har förändrats och expanderat drastiskt under åren, och klarat av att hållas i kapp med den senaste teknologiska utvecklingen. Även idag, så är spelmotorns huvudsakliga syfte att förse datorspelindustrin med starka och pålitliga verktyg, och att göra tröskeln för spelut-

veckling för utvecklare av olika grad så låg som möjlig. Unity har även utökat deras influens inom andra industrier med stora satsningar på realtid 3D-utveckling, vilket gör Unity till en av de kraftigaste spelmotorerna inom branschen. (zenva, 2023)

2.3 Grundläggande tekniska detaljer Unity

Unity är en långt utvecklad modern spelmotor som möjliggör utveckling av både spel och 3D-animationer för utvecklare. Dess funktioner inkluderar 3D- och 2D-grafiskt stöd, vilket ger användaren friheten att välja den konststil som passar deras projekt bäst. För varje grafiktyp medföljer specialiserade verktyg och API för att underlätta arbetsflödet. Arkitekturen är lättförståelig och organiserar spelens struktur i scener, där varje scen innehåller alla spelobjekt som krävs för den specifika spelomgången, inklusive bakgrunder och spelkaraktärer. Med verktyget "Inspector" får användarna snabb åtkomst till alla egenskaper hos spelobjekten, vilket gör det möjligt att ändra dem effektivt utan att manuellt behöva ändra koden. Unity Scripting API erbjuder att integrera egen kod och skapa anpassade funktioner. En av Unitys främsta fördelar är dess tvärplattformstöd, vilket gör det möjligt för utvecklare att exportera sina spel till olika plattformar såsom Android, IOS, Windows, MacOS, Linux, PS4, Xbox One med mera. Unity är även en viktig aktör inom VR- (virtuell verklighet) och AR- (förstärkt verklighet) teknologier och erbjuder stöd för utveckling inom dessa områden. Dess tillgångsbu- tik är omfattande och erbjuder både gratis och prissatta produkter som spelobjekt och grafik till utvecklarna. Unity erbjuder flera inbyggda möjligheter för renderingspipeli- nes, vilket ger utvecklare möjlighet att välja den renderingspipelinen som tillämpar sig bäst för deras projekt och grafiska behov. Användaren kan importera egna animationer från externa program som till exempel Blender, dock har Unity även eget inbyggt ani- meringsverktyg som är mycket sofistikerad för både 3D- och 2D-grafik och tillåter an- vändaren att skapa dynamiska och engagerande animationer för både spel och film. Trots att Unity är främst inriktat på spelutveckling har plattformen även anpassats och utökats för att stödja andra branscher, som till exempel filmproduktion och arkitektur, tack vare sina mångsidiga funktioner och verktyg (zenva, 2023). För att hålla sig kon- kurrenskraftig med UE5:s blueprint-system så skapade Unity deras egen blueprint- system Bolt, med vilken man kan skapa kodfunktionalitet utan att behöva skriva kod. (stream, 2022). Bolt kan importeras som ett tilläggsprogram i ens Unity projekt (Unity

Technologies, 2020). Unity är programmerat i språket C# och är kompatibelt med andra objektorienterade skriptspråk, vilket ger utvecklare flexibilitet och möjlighet att utnyttja sina programmeringskunskaper (Unity Technologies, 2024).

2.4 Unity styrkor och svagheter

Unitys främsta kännetecken är dess förmåga att möjliggöra utvecklingen av korsplattformprodukter. Genom att stödja utveckling för över 25 plattformar, inklusive mobila enheter, datorer, webb och spelkonsoler, erbjuder Unity en omfattande möjlighet för utvecklare att nå en bred publik. Spelmotorn innefattar en lättviktig renderingspipeline som är anpassad för mobila plattformar och kan skräddarsys genom skriptning. Utvecklingsmiljön i Unity anses vara optimal för nybörjare tack vare sitt enkla och intuitiva användargränssnitt, samt att kodningsspråket, C#, är relativt lätt att lära sig. Ytterligare underlättande för utvecklare är "Unity Asset Store", en marknadsplats som erbjuder ett brett utbud av färdiga 2D- och 3D-tillgångar och miljöer, vilket kan accelerera och förenkla spelutvecklingsprocessen. Programvaran är gratis för projekt som genererar mindre än 100 000 dollar per år. Unity erbjuder mycket bra verktyg för skapandet av mobilspel och indie-utveckling, och har en omfattande användarbas samt ett aktivt samhälle av utvecklare, vilket gör tröskeln för att söka rådgivning gällande olika problem via internet mycket låg.

Tillgång till läsbar källkod är endast möjligt genom betald prenumeration vilket gör det svårare att hitta och lösa prestandaproblem med motorn, och Unity har en lägre prestanda jämfört med mer högt specialiserade spelmotorer. Avancerade mångspelarfunktioner är endast tillgängliga genom att ladda ner ramverk (Kevuru Games, 2023). Även om Unitys basversion är gratis så låser Unity flera olika funktioner och egenskaper bakom olika nivåers kostnadsmissiga prenumerationer. Funktioner som till exempel molndiagnostik och implementering av spel till stängda plattformar som Playstation och Xbox ligger bakom betalningsväggar. Unity börjar också kräva betalningar efter att man gör 200 000\$ i årliga intäkter med ens Unity produkt, dock kan man öka denna årliga intäkts tröskel med att köpa de kostnadsmissiga versionerna av Unity (French, 2022). Tilläggsprogrammet Bolt är inte lika sofistikerad som Unreal Engine 5:s egna Blueprint

system och har mindre utvecklingsmöjligheter när det kommer till visuell skriptning (stream, 2022).

2.5 Unreal Engine bakgrund

Unreal Engine är en 3D-datorgrafikspelmotor utvecklad av Epic Games, som har uppnått betydande framgång sedan dess lansering. Den anses vara en av världens mest kraftfullaste spelmotor för 3D-datorgrafik och först introducerades mjukvaran år 1998 i förstapersonsskjutarens spel "Unreal". Trots att programvaran ursprungligen utvecklades för FPS-spel (förstapersonsskjutare), har den framgångsrikt expanderat till en mångfald olika genrer, inklusive men inte begränsat till MMORPGs (Massive Multiplayer Online Roleplaying Game), plattformsspel, stridsspel och mera. Unreal Engine är ett av de mest sofistikerade och öppna verktygen för skapande av realtids-3D-miljöer och används av utvecklare och CG-konstnärer över hela världen. Sedan dess lansering har Epic Games utökat användningen av deras spelmotor till andra branscher, såsom bilindustrin, arkitekturvisualisering, film och TV, live-evengemangsproduktion och andra realtidsapplikationer. Spelmotorn är känd för dess djupa funktionalitet och höga visuella kvalitet, och UE5 har fortsatt att vidare utveckla på dessa styrkor, vilket ger användarna ännu mera detaljerade 3D-miljöer, realistiska ansiktsuttryck och världs byggande i stor skala. (CGHERO, u.å)

2.6 Grundläggande tekniska detaljer Unreal Engine 5

Unreal Engine 5 är en spelmotor för utveckling av både spel och 3D-animationer för utvecklare. UE5 möjliggör för användare att ansluta till mediaproduktionspipelines genom att erbjuda stöd för olika standarder såsom FBX (Filmbox, FBX filer är en 3D-modell filtyp skapad med användning av AutoDesk FBX mjukvara. De kan bli designade och modifierade i ett flertal olika modellerings applikationer, som till exempel Maya, 3ds Max och Blender (Vection, 2022)), USD (Universal Scene Description, programvara för att robust och skalbart utbyta och förbättra 3D-scener som kan bestå av många elementära tillgångar bland olika applikationers rörledningar (openusd, u.å)) och Alembic (Alembic är ett utbytbar filformat för datorgrafik som används av animatörer för att destillera komplexa animerade scener till en icke-procedurberoende, applikat-

ionsoberoende uppsättning av bakade geometriska resultat (Muir, 2022)). Unreal Engine är tillgänglig som utvecklingsmiljö och innehållsskapande i Linux, macOS och Windows. Den inbyggda Landscape system verktyget gör det möjligt att bygga mycket stora öppna världar med olika slags terränger som till exempel berg, dalar och grottor. Unreal Engine 5 har olika verktyg, som till exempel "Automatic LOD generation", för att förbättra realtids prestandan genom att optimera komplexa modeller. Volumetric Cloud komponenten tillåter manipulering av omgivningens belysning, himmel, moln och atmosfärliga effekter. Med Animation Blueprint Editor kan användaren editera direkt skeltnätets animations rörelser eller uppställa logik som definierar den slutliga animations rörelsen. Med Control Rig verktyget kan användaren skapa och dela "riggar" för flera spelkaraktärer, och sedan posera samt animera dem med Sequencer. Sequencer verktyget är en icke-linjär, realtidsbaserad redigerings- och animationsmiljö. Med Sequencer kan man definiera och justera belysning, kameraövergångar, karaktärer och scenkläder för varje enskild scen. Verktyget tillåter aktörer att arbeta samtidigt på sekvenser. Virtualiserad mikropolygongeometrisystemet "Nanite" och Virtual Shadow Maps gör det möjligt att importera källkonst av filmkvalité som kan bestå upp till miljontals polygoner och placering av dessa tillgångar samtidigt som man bibehåller en realtidsbildhastighet och utan märkbar förlust av detaljer. Unreal Engine möjliggör stöd för mycket stora texturer genom att dela upp dem i små rutor och endast ladda de synliga rutorna. Streaming Virtual Texturing, som använder texeldata (texel data är pixelinformation i texturer (MDN, 2023)) från omvandlade texturer på disken minskar minnesanvändning för ljuskartor och detaljerade texturer skapade av UDIM UV (U-dimensionell mappning av UV-koordinater är ett system för att tilldela texturer på ytor enklare (Modo, u.å)) artister. Runtime Virtual Texturing i vilken texeldata generars av GPU:n (grafikprocessor) under körningstiden, förbättrar renderingsprestandan för procedur- och lagerbase-erade material. Lumen är en belysnings och reflektions lösning där indirekt ljuseffekter anpassar sig automatiskt till förändringar av direkta ljuseffekter och geometri. Användaren kan selektivt välja med Unreal Engines strålsökning att strålsöka reflektioner, skuggor, transparens, omgivningsocclusion, bildbaserad belysning och global belysning. UE5 tillåter användaren att välja mellan ett flertal av sofistikerade efterhandsprocesserings effekter för att justera den allmänna bilden och känslan för ens scen som till exempel HDR Bloom, tonmappning, linsfläck, skärpedjup, kromatisk aberration, vinjetter-

ring och automatisk exponering. Movie Render Queue är ett verktyg som tillåter slutprocessering av media för olika slags filmer. Verket renderar filmer och stillbilder med ackumulerad kantutjämning och rörelseoskärpa. nDisplay tillåter användaren att rendera innehåll i realtid i olika resolutioner och att uppvisa innehållet på flera fysiska eller projekterade rutor som till exempel Powerwalls (powerwall är en samling av ultrahögresolutions bildskärmar (wikipedia, u.å), domes (en dome är en 360 graders kupolformad bildskärm), CAVE (Cave automatic virtual environment är en virtuell miljö i vilken projektioner riktas mot mellan tre och sex av väggarna inom omgivningen (Wikipedia, u.å)) och LED-volymer (En LED-volym är ett system av länkade LED-paneler för att uppvisa video material eller 3D-innehåll som bakgrundsmaterial (Latvis, 2023)). Virtual Scouting verktyget tillåter filmskapare att utforska den skapade världen genom VR, för att till exempel komponera scener och ställa sig in i en realistisk virtuell representation av filminspelningsplatsen, vilket ger en bättre överblick av produktionen. När man skapar ett projekt i UE5 kan användaren välja bland olika färdiga mallar med specifika inställningar aktiverade. Mallarna varierar från spelproduktion till film, video, arkitektur och produktdesign mallar. (Epic Games, u.å).

2.7 Unreal Engine 5 styrkor och svagheter

Den huvudsakliga särdragen med Unreal Engine 5 är dess stöd för högdefinition grafik, näst generations fysik, belysning och visuella effekter. UE5 är kodat med C++ språket och erbjuder hög portabilitet och täcker nyckel spelplattformar som till exempel mobil, skrivbordsdator, konsoler och VR/AR. Fokuset av spelmotorn är AAA-kvalitets PC- och konsolprodukter, vilket är industristandarden för toppspel. UE5 är aktivt visat som utvecklaren för den senaste och bästa utvecklingen inom grafisk realism. UE5 är en mycket robust spelmotor som ger möjligheten att implementera toppmoderna visuella effekter till hög prestanda även vid hantering av tunga tillgångar och miljöer. Dess avancerade verktyg för mångspelarlägen, AI och de senaste VR-projekten ger en förstärkt användarupplevelse (Kevuru Games, 2023). Blueprint visual scripting systemet är ett användarvänligt verktyg möjliggör skapande av spelmekaniker utan kodning, vilket gynnar konstnärer och designers. Tillgång till spelmotorns C++ kod ger möjlighet till anpassning enligt användarens egna behov och krav. Gratis utveckling fram tills en viss inkomstnivå gör det till en attraktiv plattform för både erfarna utvecklare och nybörjare.

Den sofistikerade renderingspipelinen underlättar skalning av spel och produktioner över olika plattformar, vilket förenklar utvecklingsprocessen. Eftersom det finns så pass många företag och utvecklare som använder sig av UE5 så är det enkelt att rekrytera arbetare samt att göra samarbete med andra företag för olika projekt. Unreal Engines stora och aktiva samhälle av utvecklare och arbetare från Epic Games skapar ett brett nätverk av offentligt stöd som är tillgänglig för alla via internet (Dealessandri, 2020).

Svagheterna med programvaran är det komplexa användargränssnittet och branta inlärningskurvan för nybörjare och det kan kräva en stor arbetsinsats för att nå en tillräckligt stor förståelse om programmet innan man kan utnyttja varan till ens egna ändamål. Utvecklaren är även tvungen att betala 5% i royaltykostnader när produkten tjänar mera än 1 miljon dollar. De höga systemkraven för att använda spelmotorerna fulla potential skär bort potentiella utvecklare och användare som inte har tillgång till stora resurser (Kevuru Games, 2023). När det kommer till stöd för olika plattformar så erbjuder Unity mera alternativ än Unreal Engine (Geniuscrate u.å).

3 Spelmotorintegration inom filmindustri

Detta kapitel går in på hur spelmotorteknik har implementerats inom filmbranschen för film som Disneys The Mandalorian och Disneys nya Lion King film. Jag har också intervjuat Peter Ahqlvist som arbetar med scenografi inom olika teaterproduktioner för att få en insikt i fysisk scenkonstruktion.

3.1 StageCraft

Inom filmindustrin har det under de senaste åren blivit ett ökat intresse för att integrera spelmotorteknik i olika filmproduktioner. Det mest kända exemplet är Disney som kombinerade användningen av fysiska set och StageCraft (ett realtidssystem för virtuella scener utvecklat baserat på Unreal Engine, den omvandlar studion till en virtuell scen med dynamiskt växlande bakgrunder) under produktionen av Star Wars-serien The Mandalorian. När en scen spelas in projicerar teknikerna en renderad bakgrund på den virtuella LED-skärmen. Detta gör att filmprocessen i The Mandalorian kan ändra sce-

nen dynamiskt enligt behoven för scenen och sparar både tid och resurser eftersom man inte behöver bygga upp scenen på nytt eller använda flera olika studior. Skådespelarna agerar i virtuella scener som förberetts i förväg, och den slutliga effekten med både skådespelaren och den virtuella miljön visas i realtid på regissörens monitor. (UNIT LED, 2023)

3.2 Fördelar med LED-rutor gentemot fysisk scenkonstruktion

Att skapa en LED omgivning för ens produktion har varierande kostnader beroende på produktionens storlek och skalbarhet. För Disneys *Mandalorian* blev kostnaden för LED-väggen ca. 100 miljoner dollar, inkluderat all annan utrustning för filmningen. Trots den initiala höga investeringen blir det på lång sikt lönsamt både tekniskt sett och kvalitetsmässigt. Filmning blir effektivare i den virtuella studion, eftersom den inte påverkas av väderförhållanden. Dåligt väder kan annars försena inspelningsprocessen och öka de löpande kostnaderna. Skådespelarna kan snabbt fördjupa sig i scenen, vilket minskar antalet ogiltiga tagningar och sparar kostnader. Dessutom sammanfaller mycket av för- och efterbearbetning av produktionen med den faktiska inspelningen, vilket förenklar efterproduktionsprocessen. Den virtuella studion med LED-väggen eliminerar behovet av att transportera set, personal och utrustning till olika inspelningsplatser runt om i världen. Efter att LED-studion har tagits i bruk minskar kostnaderna avsevärt för framtida produktioner, eftersom tekniken kan återanvändas för en mängd olika TV- och filmproduktioner samt hyras ut under stillastående perioder. Kostnaden för att bygga en stor LED-vägg på dagens marknad är cirka 10 miljoner dollar. Faktorer som påverkar kostnaden för LED-väggen inkluderar pixelstorlek, där fler och mindre pixlar ger högre bildkvalitet, och ytan på LED-rutan. En virtuell studio kan bestå av LED-tak, vägg och golv, och ju större skala desto dyrare blir kostnaden för panelens yta. Andra faktorer inkluderar speciell teknik som hög uppdateringsfrekvens, låg skanning, frånvaro av flimmer och moiré-effekter (Moiré effekt hänvisar till ovanliga och repeterande linjer, prickar eller färger i högt detaljerade bilder (Painter, 2021)). (UNIT LED, 2023).

3.3 VR teknologi och virtuell miljö i samband med traditionell filmning

Disneys nya Lion King film som utsläpptes år 2019 sågs både som ett teknisk under inom fältet för fotorealistic animation och användning av virtuell verklighet. Filmen skapade en helt ny filminspelningsprocess som blev kallad för ”virtuell produktion” genom att bygga landskapet och karaktärerna inom en virtuell värld i Unity spelmotorn. Under produktionen så användes VR-teknologi för att bestämma hur man skulle filma de olika scenerna av filmen. Filmregissören Jon Favreau och hans team använde sig av VR-headsets som transporterade dem in i det virtuella filmsetet var de kunde gå igenom alla filminspelningsplatser och bestämma olika kameravinklar, belysning och även ta bort eller lägga till virtuella rekvisita. Med kameror insatta i den virtuella omgivningen kunde man rama in filmade scenen och filmen i realtid. (The University of Michigan Center for Academic Innovation, 2020).

3.4 Intervju med Peter Ahlqvist

I samband med detta arbete har jag intervjuat Peter Ahlqvist som arbetar med scenografi i finsk teater och musikal. Syftet med intervjun är att få en insikt i fysisk scenkonstruktion i jämförelse till virtuell scenkonstruktion.

Fråga 1: Vem är du och vad gör du?

Svar: Peter Ahlqvist, arbetar som scenograf främst inom teater och musikal på shownivå. Utbildad skådespelare. Sen tonåren har jag hållit på med scenbyggande. Jag arbetar för statligt finansierade produktioner med fast personal. Min uppgift som utomstående scenograf är att hålla mig till produktionens budget.

Fråga 2: Vad är det mest tidskrävande när man bygger en set inom TV och teater?

Svar: Den ihärdiga förhandsplaneringen är det mest tidskrävande. Det innefattar alla ritningar och planer för produktionen. Sedan kommer förverkligandet när man överlämnar materialet till teatern för att bygga upp allt beroende på storleken av produktionen. Färdigt materialet måste vara klart minst ett halvt år före premiären. Byggarbetet kan ta mellan 2-3 månader. Efter koronan så är det

mycket inhoppare ifall någon person är sjuk, så att man inte måste sätta stopp för allting.

Fråga 3: Av vad byggs upp scenografi budgeten för en teaterset i ett skådespel och hur stora delar av budgeten går det till dom olika delområden, d.v.s rekvisita, lönekostnader för personal och övrigt material?

Svar: Teatern fastställer en budget som man måste hålla sig till. Det är ett samarbete mellan mig och teatern gällande budgetfrågor. För stora musikaler är materialbudgeten max 50 000 euro. Material och lönekostnader är de största posterna. Inom tv-branschen finns det vissa privata firmor som bygger, medan teatern oftast har egna verkstäder för det mesta.

Fråga 4: Vad är det mest utmanande med scenografi?

Svar: Det mest utmanande är att lösa tekniska problem för att få allt att fungera i produktionen. Man jobbar med flera olika avdelningar och människor och ska få alla att förstå ens vision och samarbeta för att uppnå målen.

Fråga 5: Har du använt dig av någon form av digitala specialeffekter i ditt arbete, om du har så hurudana, om inte så kan du tänka dig att använda dig av det i framtiden?

Svar: I "Pieni Merenneito" på Helsingin Kaupungin Teatteri använde vi back projection och framifrån, samt hologramlycra/hologramtyll som är vävt med silvertrådar och tar emot ljus jättebra. Det ger en hologramkänsla. Tekniken fungerar så länge det inte tar bort magin av teatern utan istället stödjer den. Att sätta på VR/AR-glasögon kan bryta illusionen. Den största anledningen till att vi inte använder tekniken är budgeten, men som en förstärkare av något så varför inte?

Efter de förhandsbestämda frågorna hade vi med Peter en fri diskussion gällande hans åsikter om VFX, LED-teknik och VR inom teater.

Enligt Peter (personlig intervju, 7 februari, 2024), när det kommer till användning av LED-teknik inom finsk teater så är den största utmaningen att ha en tillräckligt stor budget för LED-tekniken och annat problem är att leasingfirmor för

tekniken inte vill hyra ut för långa perioder, vilket är ett problem eftersom en teaterproduktion brukar löpa en längre tidsperiod. När vi diskuterade om möjlig VR-tekniks implementering inom teater så anser han själv att ”teater på distans” leder till att skådespelarna tappar kontakten med publiken och den personliga upplevelsen av en live föreställningen försämras.

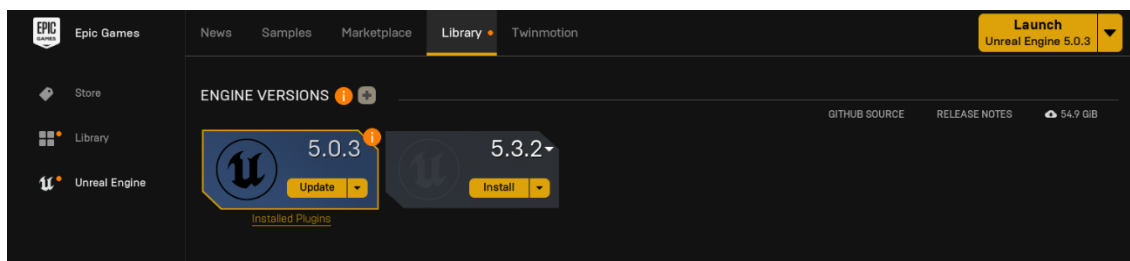
Intervjun med Peter tyder att det finns intresse och vilja att implementera LED-teknik för olika typer av teaterproduktioner, men så länge som budgeten inte är tillräcklig och tekniken är dyr så är det mycket utmanande att tillämpa tekniken inom branschen.

4 Att göra en kortfilm med Unreal Engine 5

För att uppnå målet att skapa en scen med 3D-omgivning så användes följande resurser: Blender, Unreal Engine 5 och DaVinci Resolver. Syftet är att få en mera praktisk bild på gräsrotsnivå om processen att skapa scener och arbetsflödet med UE5 och för att uppnå detta har jag byggt upp olika scener inom UE5.

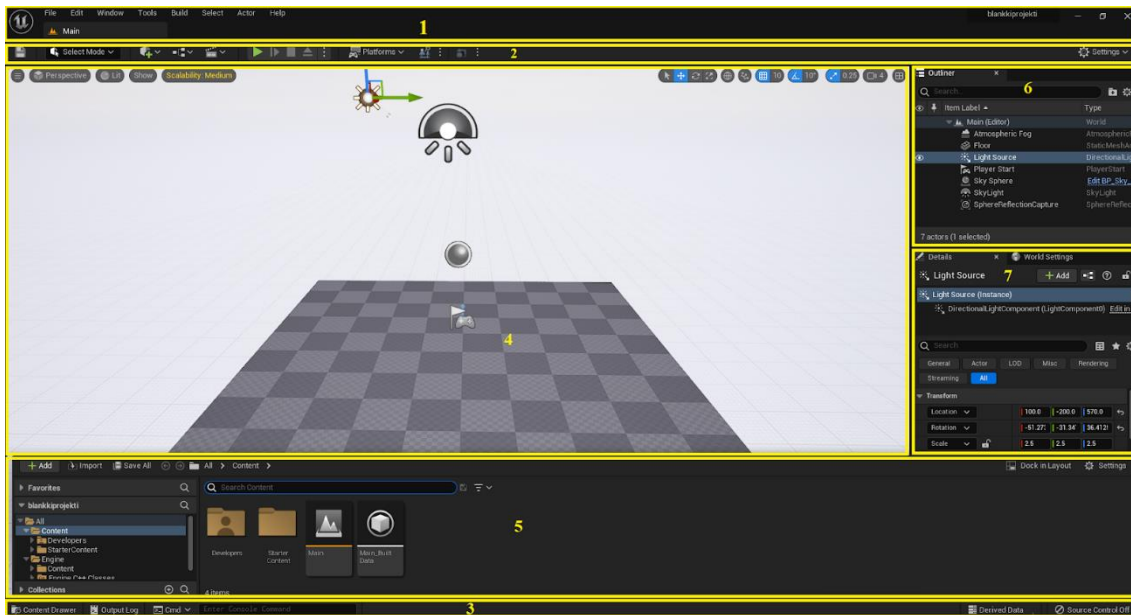
Målet är att skapa en scen vars syfte är att se så realistisk ut som möjligt med användning av olika 3D-modeller, fysik effekter och filmteknik.

Efter att Epic Games och Epic Games Launcher installerats kunde UE5 versionen 5.0.3, installeras på datorn. Därefter initierades UE5-programmet och ett nytt projekt skapades.



Figur 1. Epic Games programstartare.

Unreal Engine skapar en tom virtuell scen utan någorlunda tillgångar.



Figur 2. Standardutformning av användargränssnittet i UE5. De olika komponenterna är numrerade i ordning.

Bilden visar en tom scen inom UE5 med standardutformning av användargränssnittet.

De olika komponenterna enligt ordningen i bilden:

1: Flikfält och Menyraden.

Menyraden i redigeraren ger åtkomst till allmänna verktyg och kommandon som används när du arbetar med nivåer i redigeraren.

2: Verktygsfält.

Verktygsfältet visar en grupp av kommandon som ger snabb åtkomst till vanligt använda verktyg och operationer.

3: Nedre verktygsfältet.

Innehåller genvägar till kommandokonsolen, utmatningslogg och funktioner för härledda data. Visar också status för källkontroll

4. Vyport.

Denna panel innehåller en uppsättning vy-portar, var och en av dem kan maximeras för att fylla hela panelen och erbjuder möjlighet att visa världen från en av tre ortografiska vyer (Topp, Sida, Framifrån) eller en perspektiv vy som ger dig full kontroll över vad du ser och hur du ser det.

5. Innehållsfönstret.

Innehåller alla tillgångar som används i nivån.

6. Översikt.

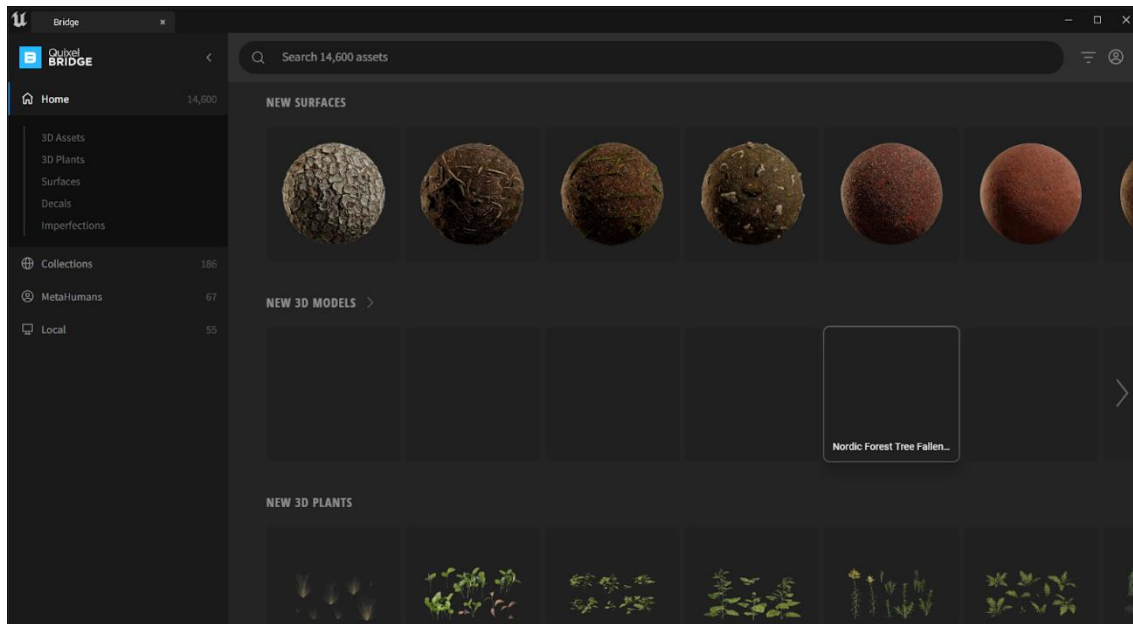
Översiktsfönstret visar alla skådespelare i scenen i en hierarkisk trädvvy. Du kan välja och ändra skådespelare direkt från översiktsfönstret.

7. Detaljer.

Detaljfönstret innehåller information, verktyg och funktioner för det aktuella urvalet i vy-porten. Det innehåller transformationsredigeringsrutor för att flytta, rotera och skala skådespelare, visar alla redigerbara egenskaper för de valda skådespelarna och ger snabb åtkomst till ytterligare redigeringsfunktionalitet beroende på vilken typ av skådespelare som är valda i vy-porten.

För den första scenen vill jag få en grundlig förståelse av hur programvaran fungerar. Jag visionerade en enkel scen, en bergsravin med grus, växter, stenar och belysning mm., och tog sedan de steg som krävdes för att förverkliga det. Det krävdes mycket experimentering av de olika verktygen och vad de faktiskt gör tills jag kunde påbörja byggandet av scenen.

Inom verktygsfältet finns det ett verktyg som heter Landscape Mode, vilket tillåter att forma landskapet i scenen till olika former. För att bygga ravinen krävdes det upphöjning och nedsänkning av delar i landskapet och denna mod hade flera olika verktyg för detta ändamål, efter olika experiment lyckades jag skapa formen för en ravin med olika höjder, djup och branthet. Efter att den råa formen för ravinen var färdig letar jag efter 3D-modeller för växtlighet och natur samt material för modellerna från Quixel Bridge, som erbjuder en mycket stor mängd av gratis modeller och material för Unreal Engine.



Figur 3. Quixel Bridge, ägd och hanterad av Epic Games. Innehåller en mångfald av gratis tillgångar som man kan implementera i ens Unreal Engine projekt.

Efter att ha valt modellen och dess grafiska kvalité från Quixel gick det mycket smidigt att ladda upp den i UE5 projektet för användning. Jag fyllde ravinen med diverse växter, stenar, bergvägg, grus m.m och för varje inlägg började ravinen få liv och ta sin form. Eftersom ljus har en så stor betydelse inom film så manipulerade jag scenens belysningsinställningar för att få uppnå önskad upplysning av landskapet. Man kan antingen skapa en ny ljuskälla eller använda sig av den ljuskällan som byggs upp i projektet av standard och manipulera på dess inställningar.



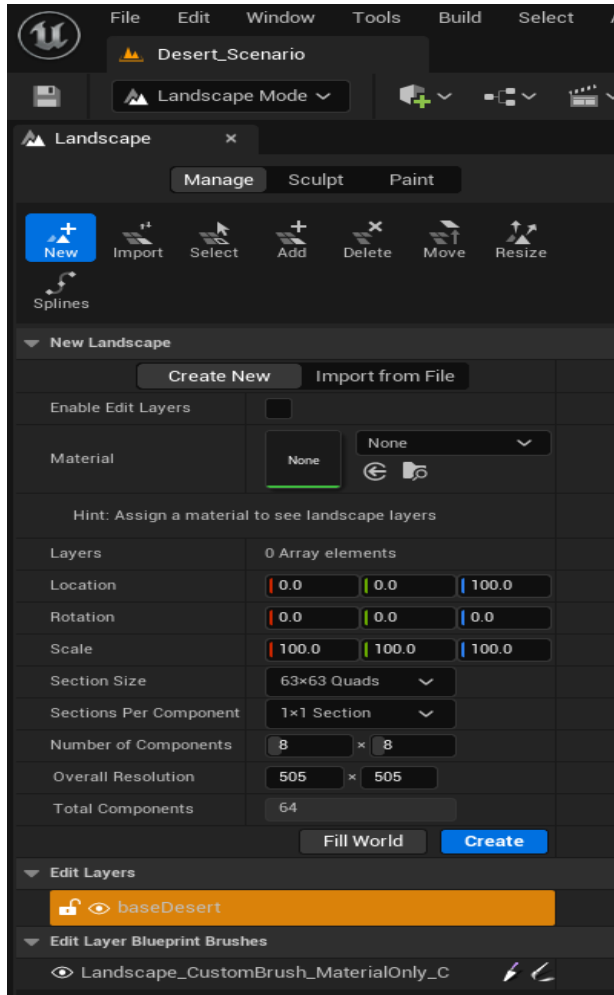
Figur 4. Ravinen skapad i UE5.

Efter att ravinen och belysningen var färdig påbörjades filmprocessen. UE5 har ett inbyggt verktyg som heter Level Sequence Actor. LSA (Level Sequence Actor) tillåter användaren att skapa filmer genom att kombinera filmsnutt tagna med programmets inbyggda Cine Camera Actor. CCA (Cine Camera Actor) replikerar en riktig kameras beteende, d.v.s. inställningar som fokusering, olika linser och man kan även välja färdiga insatta inställningar för olika filmkamera modeller. När allting var klart, d.v.s. miljön i scenen, belysningen och kamerainställningarna, ändrade jag spelmotorns grafik till den högsta nivån, "cinematic", och påbörjade inspelning. Själva filmningsprocessen gick smidigt och till slut renderade UE5 efter ca. 4 timmar resultatet i filer av jpeg- och exr-format. Det som UE5 dock inte gör är att den sammanslår dessa filer till en hel film i slutbehandlingen. För att sammanslå filerna till en sammanhängande film användes ett externt editorprogram "Da Vinci Resolve".

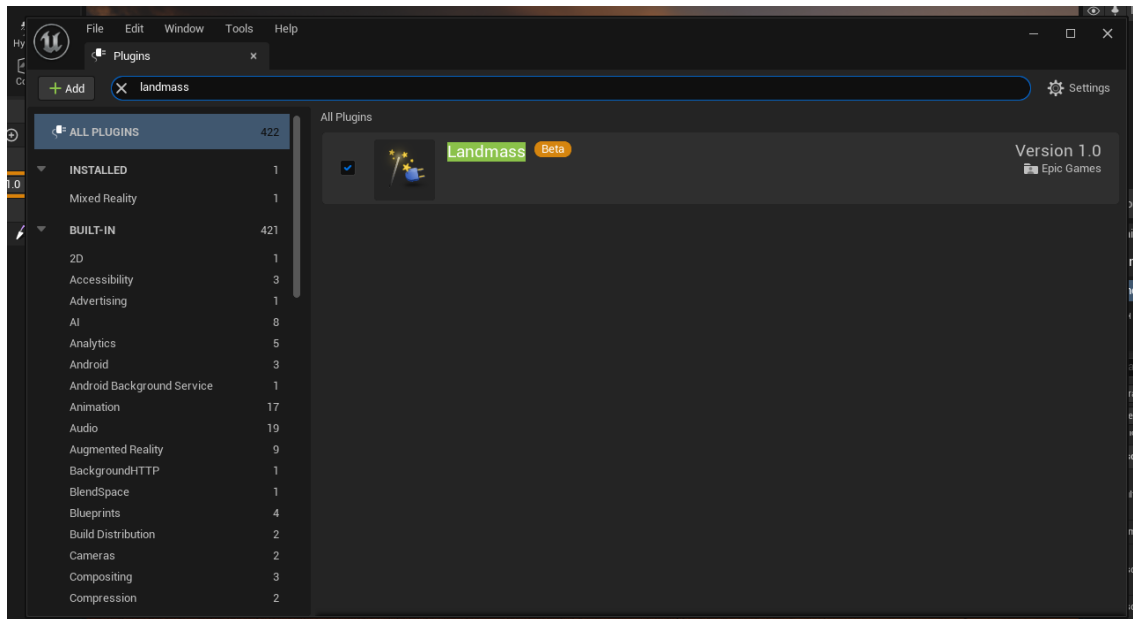
Efter att jag fått en grundlig förståelse om UE5 börjades arbetet för den huvudsakliga UE5 scenen som skulle användas som jämförelsepunkt mellan UE5 och Unity. Eftersom UE5 används inom filmbranschen för att bygga de omgivningar i vilket filmen i fråga utspelar sig så tillämpades det bäst för projektet att bygga ett stort virtuellt öken för att simulera detta. Tanken är att öknen med hjälp av belysning, material, effekter och kameravinklar ska se så realistiskt ut som möjligt.

Ett nytt tomt projekt skapades i UE5 och med Youtube-handledningsvideon (Unreality-

Bites, 2022) som beskriver steg för steg hur man proceduralt skapar en öken börjades processen. Med Unreal Engine:s landskaps-verktyg skapas ett nytt tomt landskap i projektet, "Landmass" -tilläggsprogrammets inställningar inom UEs tilläggsprogram aktiveras.



Figur 5. Landskaps verktyget.



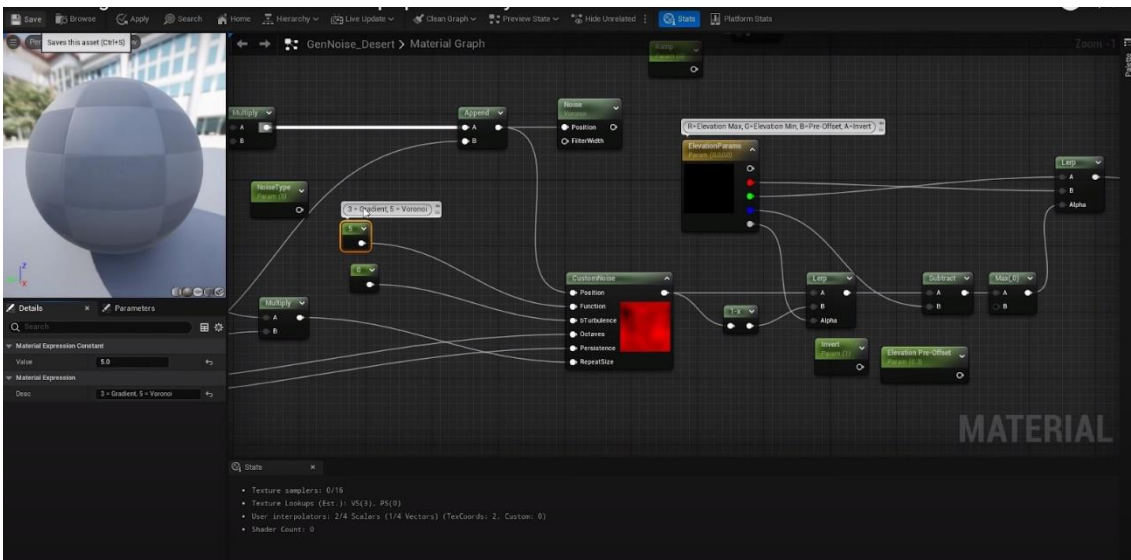
Figur 6. Landmass tillgången tillåter procedurrell generering av landskap.

Efter att basen på landskapet är skapad söktes det upp en sand liknande material från Quixel som används med terrängen. Med landskapet och materialet fixat har jag ett slätt öken skapat och börjar göra förändringar på landskapets elevation och tiles (rutor) för att uppnå strukturer som påminner om riktiga öken, d.v.s. dyner och sandformationer. Förändringarna ändrar på formen av släta ökenlandskapet till en mera realistisk form med kullar och backar, jag vill dock ha ännu vasshet och skarphet på slätterna för att få dem mera dyn lika i utseende.



Figur 7. Landskapet med sand textur och ändringar i elevation och rutor.

I UE har tillgångarna som används sina egna blåkopior med diverse noder, i vilka man kan göra skraddarsydda förändringar för att manipulera tillgångens beteende. Eftersom jag vill nå en ökad realism med dynerna så öppnar jag blåkopian för verktyget som styr landskap materialets form och inom blåkopian gör jag förändringar på noden “CustomNoise”.



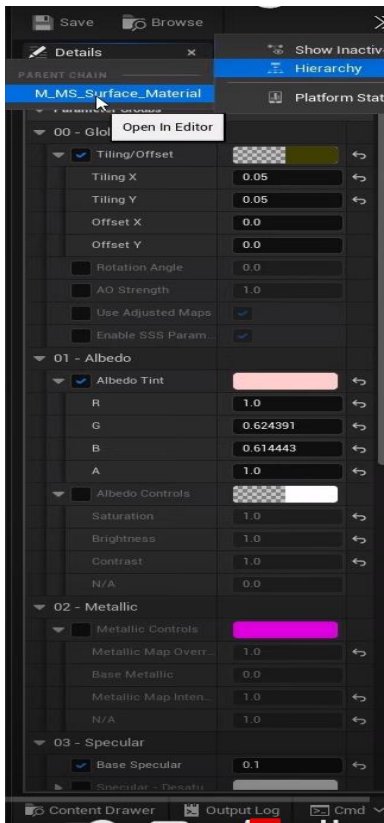
Figur 8. Blåkopian av landskapet. Ändringar i CustomNoise nodens funktion ändrar på Voronoi cellernas samverkan med det gradienta och som resultat skapas "vassare" dynformationer.

Genom att öka värdet på konstanten som styr materialets uttryck från 3 (Gradient) till 5 (voronoi) ändras landskapet till den önskade riktningen av mer realistiska dyner.



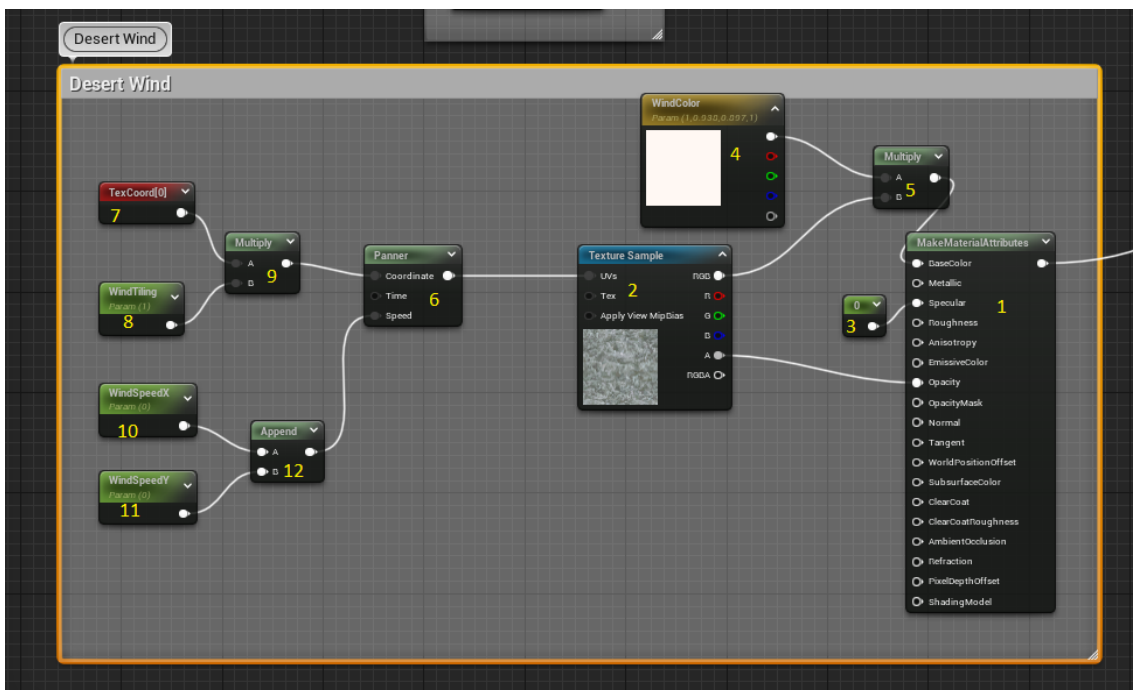
Figur 9. Landskapet efter förändringar i CustomNoise noden.

Processen för att forma landskapet är färdigt, jag vill dock ha även mera realism i miljön så jag söker upp en video i Youtube om hur man skapar en "vind" att blåsa över öknät. Med hjälp av youtube handledningsvideon (UnrealityBites, 2023) går jag steg för steg genom procedurer för att simulera en ökenvind. Jag börjar med att öppna min Landscape tillgång via detaljfönstret. Från detaljsfönstret öppnar jag materia instansen för landskaps ytans material, som används för att fylla landskapet. Inom ytans materialinstans kan jag öppna dess hierarki och hitta dess förälder material.



Figur 10. Hierarki fönstret med landskaps ytans materialinstans.

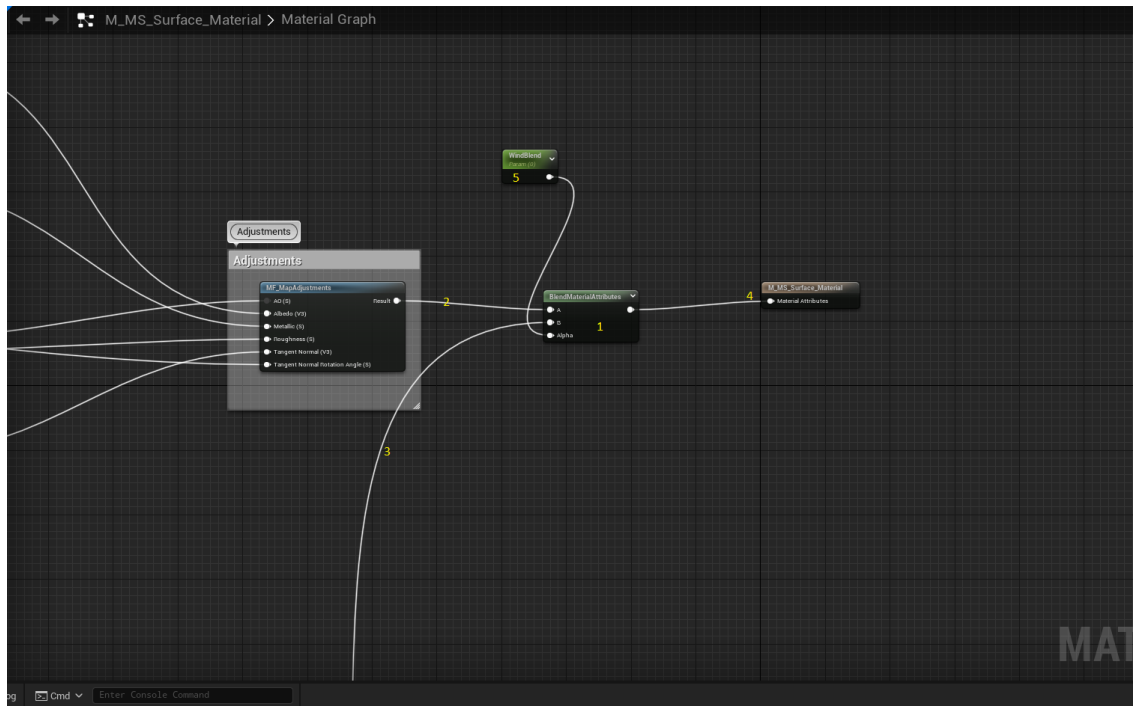
Jag öppnar blåkopian för förälder materialet och skapar hit ett nytt material som ska simulera ökenvinden. Jag använder mig av en vatten textur som huvud texturen eftersom jag vill skapa en våg liknande rörelse som flyter över öknen.



Figur 11. Ökenvindens noduppsättning inom materialets blåkopia.

- 1: För att skapa materialet behöver jag göra en hel del nya noder för att uppnå effekten. Jag lagar en ny nod inom förälder blåkopian som heter MakeMaterialAttributes vilken kontrollerar egenskaper för materialet.
 - 2: Jag lagar jag en nod som heter Texture Sample och inom dess detaljspalt söker jag upp och lägger i textur fältet "water_d" texturen. Jag länkar vatten textur "Texture Sample" nodens RGB parameter med MakeMaterialAttributes nodens BaseColor parameter och Alfa parametern med Opacity parametern.
 - 3: Eftersom Specular parametern är av standard 0.5 skapar jag en skalbar value nod med värde 0 och länkar det med Specular parametern inom MakeMaterialAttributes noden.
 - 4: Efter det vill jag kunna byta färgen på sandvågorna så att den sammansmälter med öknen. Jag skapar en vektor 3 nod, konverterar den till en parameternod och lägger som standardfärg vit.
 - 5: Jag skapar en Multiply nod som länkas så att den får input av både Vector 3 noden och Texture Sample noden, och outputn länkas sedan i BaseColor parametern av MakeMaterialAttributes.
 - 6: Nästa etapp är att få denna våg sand att röra sig längs öknet. Jag skapar en Panner nod som styr riktningen, tiden och hur snabbt våg-sandet rör sig längs landskapet.
 - 7: Jag skapar en TexCoord[0] nod som påverkar rutorna av Texture Samplet (tillåter mig att påverka sammansmältningen av texturen).
 - 8: Jag vill ha en parameter nod för TexCoord[0] noden så att jag kan bestämma parametrarna för sammansmältningen av texturen och skapar en ny nod för det som jag kallar för WindTiling.
 - 9: För att kombinera TexCoord[0] noden med WindTiling noden och få deras output in i Panner noden skapar jag en Multiply nod och länkar TexCoord[0] och Windtiles noderna fast i den så att deras gemensamma output styrs till Panner nodens Coordinate parameter.
 - 10&11: Till näst vill jag skapa de noder som styr "vindens" hastighet, för att göra detta skapar jag två nya noder som heter WindSpeedX och WindSpeedY.
 - 12: Jag skapar en Vector2 nod och länkar av WindSpeedX och WindSpeedY noderna in i Vector2 noden, till slut länkar jag den gemensamma inputen från Vector2 noden till Speed parametern i Panner noden.
- Efter alla dessa stegen är resultatet lik den i videon (UnrealityBites, 2023).

Med alla relevanta noder skapade på det sättet finns det nu ett material som gör rörelse till vatten texturen och tillåter mig att ändra på färgen också. Till skall det nya materialet sammansmälta med basmaterialet för landskapet.

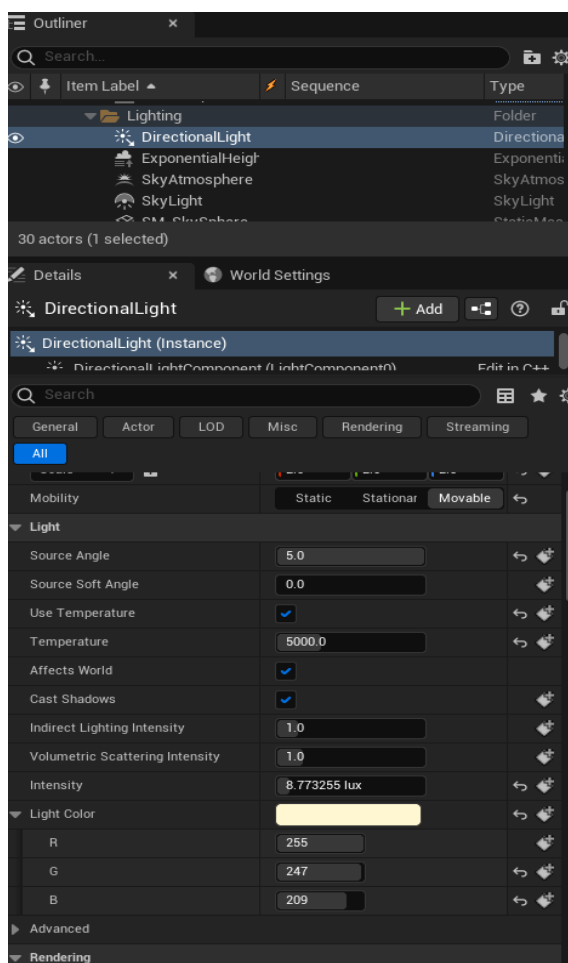


Figur 12. BlendMaterialAttributes noden som sammansmälter DesertWind nodernas funktioner med landskapsytans materialnods slut output.

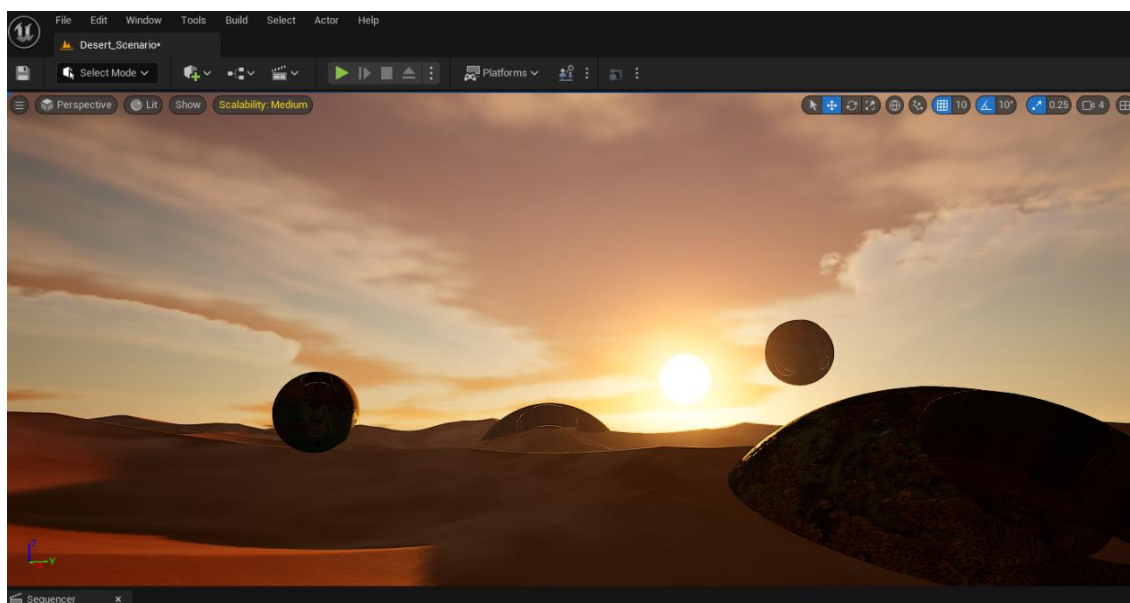
- 1: Jag skapar en nod som heter "BlendMaterialAttributes" som tar in 3 parametrar.
- 2: Jag länkar outputten av MF_MapAdjustments med A kanalen från BlendMaterialAttributes.
- 3: Jag länkar DesertWind materialet som jag just skapade in i B kanalen av BlendMaterialAttributes.
- 4: Jag länkar nodens slutliga output med M_MS_Surface_Material noden.
- 5: Som en sista process inom noderna skapar jag ännu en parameter för att kontrollera blandningen av gamla och nya materialet. Jag skapar en nod som jag kallar för WindBlend och lämnar parametrarna vid 0 och länkar det i Alpha kanalen i BlendMaterialAttributes.

Efter det ändrar jag på färgen av ökenvinden enligt vad jag önskar och så är en simulerad ökenvind färdig. Till näst börjar jag ändra på de olika funktioners parametrar som påverkar scenens belysning och atmosfärliga effekter. Genom att ändra på flera olika

funktionsparametrar som global temperatur, ljusankarets ljusfärg, blänk starkhet et cetera, så får jag scenen att se mera lik en varm ökenolnedgång vilket var tanken.



Figur x. Inspektörfönstret för olika funktionsparametrar att påverka på scenens volym.

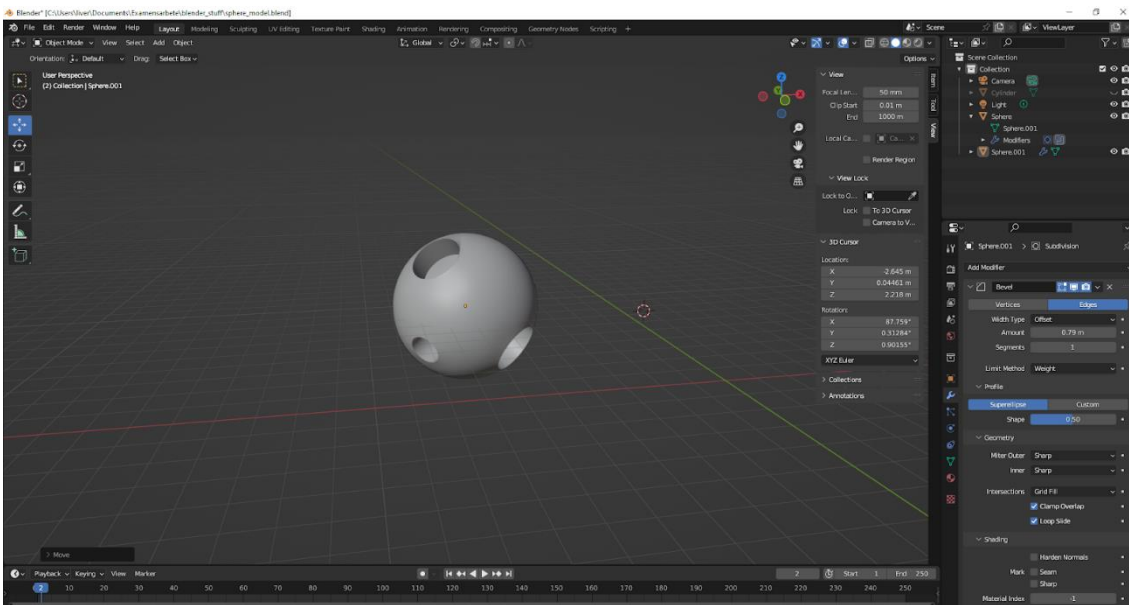


Figur x. Scenen innan förändringar i ljus och atmosfärliga parametrar.



Figur x. Scenen efter förändringar i ljus och atmosfärliga parametrar.

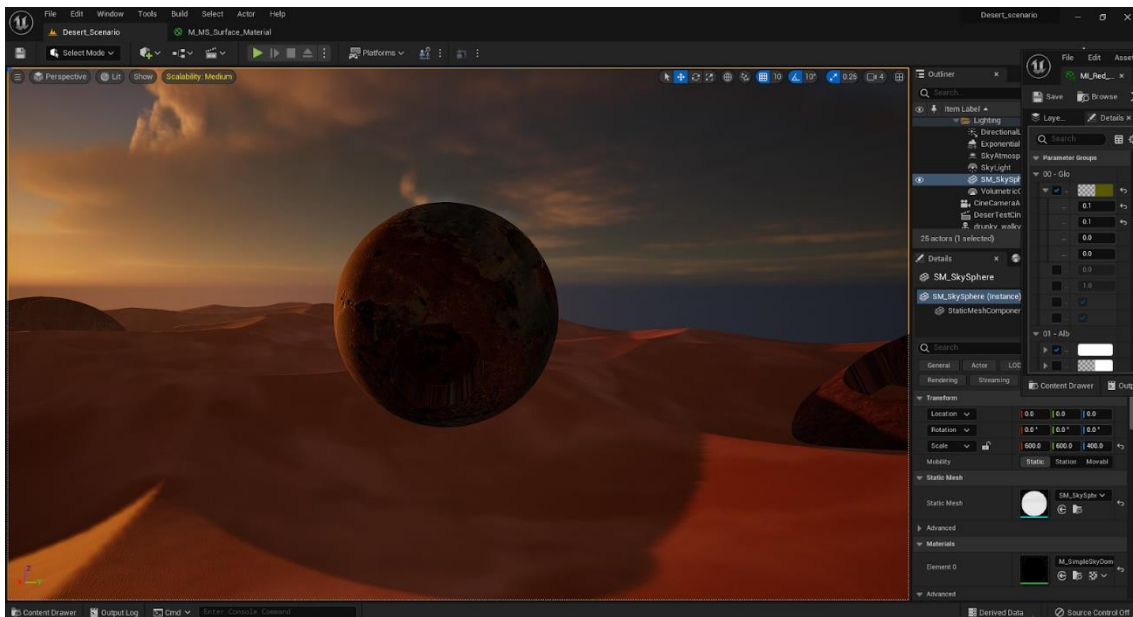
För att skapa mera liv in i UE filmen, samt för att få en basförståelse om hur man skapar och implementerar 3D-rekvisita och animations objekt in i film har jag laddat Blender. Blender är ett gratis öppet källprogram för bland annat skapandet av 3D-figurer och animationer. Jag skapade ett klotformat objekt för min UE scen.



Figur 13. Användargränssnittet i Blender med den färdiga klotmodellen.

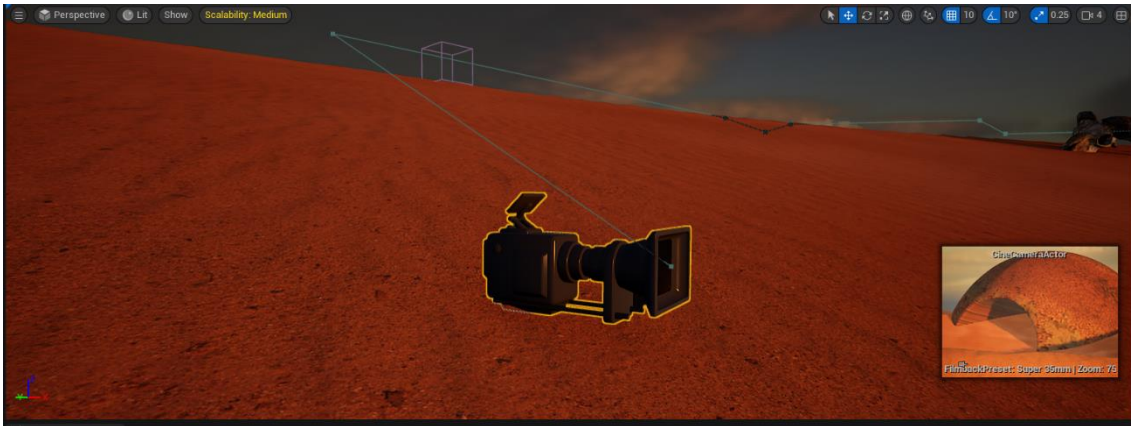
Det färdiga klotet skulle sedan importeras som en FBX-fil in till UE5, jag använde mig av en Youtube handledingsvideo (YR3Design, 2022) för att få alla inställningar för ex-

porten rätt. Efter att modellen var importerad i UE så laddade jag ännu en passande textur från Quixel och aktiverade texturen på modellen.



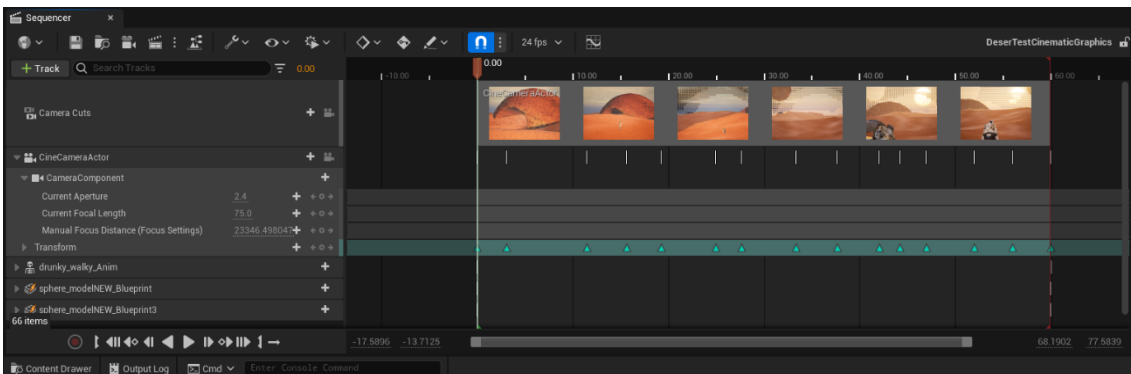
Figur 14. Blender modellen importerad in i UE5 med textur från Quixel.

Till näst ville jag ha en animerad karaktär som skulle fungera som huvudpersonen inom scenen. Jag sökte upp en nätsida som heter Mixamo, som har en mångfald av gratis karaktärer med färdiga skelett samt olika slags animationer. Efter att jag hittade en lämplig karaktär och valde en animation för det så importerade jag det till UE5-projektet, jag var tvungen att förlänga animationen eftersom Mixamo hade begränsad längd på dess animationer och jag behövde en animation som skulle räcka i 30 sekunder. Jag använde mig av Blender för att förlänga animationen. Efter en stund av sökande hittade jag en youtube video (Olav3D Tutorials), med hjälp av vilken jag lyckades förlänga animationen och så exporterade jag filen som en FBX-fil och importerade det in i UE5 scenen. Med min scen färdig uppbyggd nu med alla komponenter, karaktärer, objekt och miljö kunde jag påbörja filmningen av scenen. UE5 har färdigt inbyggda verktyg för att filma olika filmer och kortfilmer inom programmet, det kräver ingen importering av tillgångar utan man kan direkt lägga in tillgångarna i scenen för att påbörja processen. Jag skapar en Cine Camera Actor som replikerar en kamera i virtuell miljö och lägger den in i scenen.



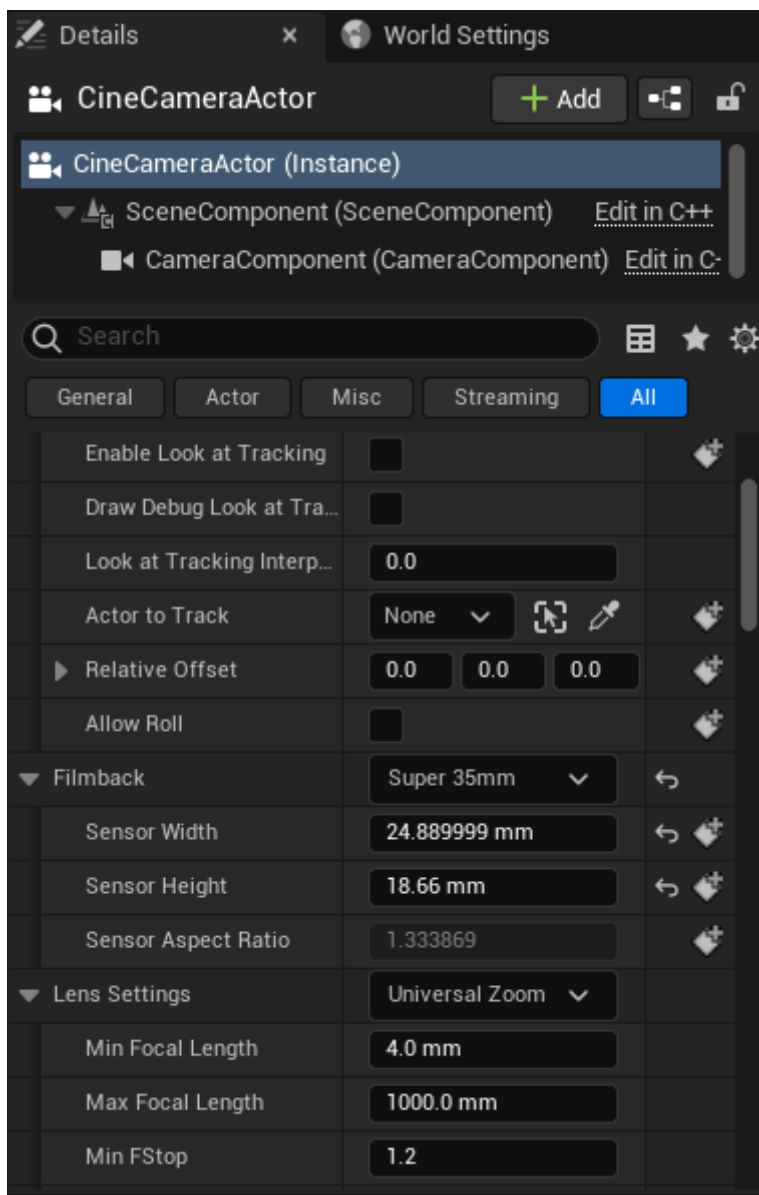
Figur 15. Virtuellt kamera Cine Camera Actor som simulerar funktionerna av en fysisk kamera.

Sequencer verktyget tillåter mig att hantera filmprocessen, d.v.s. styra nyckelbildrutor, animationer, filmklipp med mera.



Figur 14. Med Sequencer verktyget hanterar man filmprocessen inom UE5.

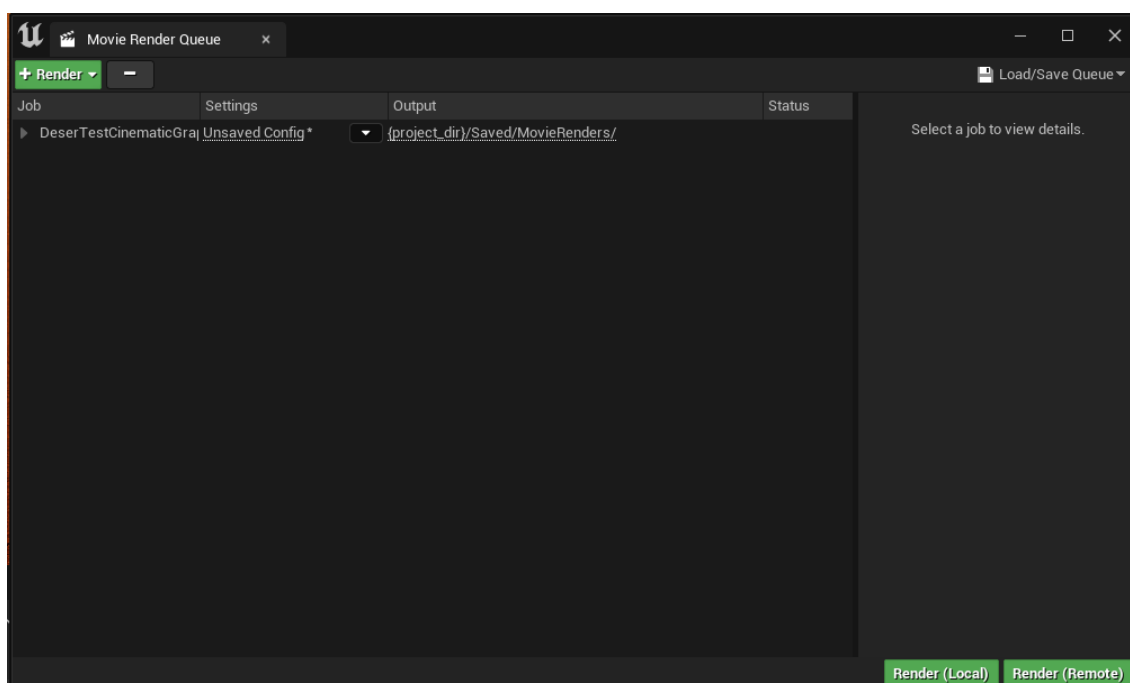
Efter att jag har film verktygen på plats så letar jag fram en Youtube handledningsvideo angående hur man inställer kameran inom UE för att motsvara en riktig filmkamera. Handledningsvideon (cinemyscope, 2023) går in på ett flertal olika kameramodeller som används inom film, jag bestämmer mig för att använda en Super 35mm kamera eftersom det är den vanligaste typen av kamera som används inom film (Boundless Entertainment, 2022). Jag ändrar på värdet i synfältet till 75 FoV (Field of View) som rekommenderas i youtube videon och sedan på bländaröppningen som kontrollerar kameran's ljusintagning (F-stop värdet) till f/1.2.



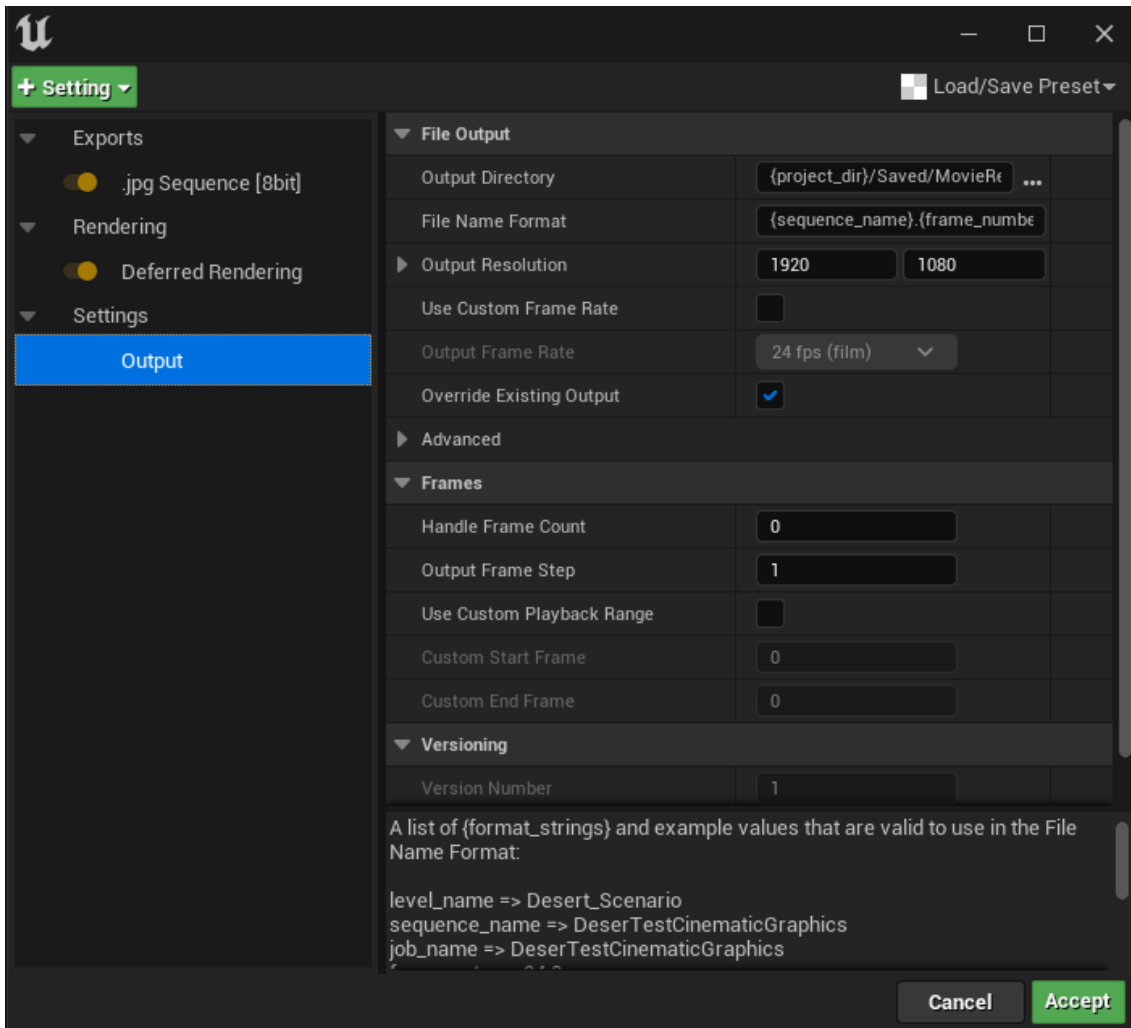
Figur 16. Användargränssnittet för Cine Camera Actor.

Jag bandar in ett antal filmer från olika vinklar för att få en överblick om hurdana kameravinklar och motioner skapar en bra film, tanken är att slutprodukten skall vara en smidig och kontinuerlig filmsnutt av miljön med karaktären i centrum. Jag sparar olika nyckelbildrutor inom filmen, d.v.s. inom den banan som min kamera följer har jag olika nyckelrutor sparade, när jag stannar kameran och byter riktning sparar jag en nyckelruta varifrån kamerabanan fortsätter ifrån, som i sig underlättar filmprocessen eftersom jag inte är tvungen att filma hela filmen i ett kör. Efteråt när jag är nöjd med resultatet påbörjar jag processen att ändra på belysning och färgbalans i miljön. Jag ändrar på kameran exponering för att påverka mängden ljus som bildsensorn utsätts för, jag minskar på linsljusets intensitet för att undvika att ljuset blossar i kameran, jag ökar färgtempera-

turen för att skapa illusionen av en varm miljö. Efter att kamerainställningarna är färdiga, sätter jag kameran med de nya inställningarna aktiverade, att banda in scenen enligt den färdiga bestämda inspelningsbanan som jag skapade tidigare. Inom Level Sequencer ändrar jag de tagna nyckelbildrutor från att vara automatiska nyckelrutor till linära nyckelrutor för att få inspelningen att se lineär ut. Efter det väljer jag att rendera filmen med UE5:s “Movie Render queue” och gör de sista förändringar inom renderingsinställningarna. Jag vill rendera ut .jpg- sekvenser samt .exr- format, jag vill att bildfrekvens för slutprodukten skall vara i 24 fps (bilder per sekund) som används inom film. Jag trycker på att rendera lokalt och efter en tid har jag en färdig film i .jpg- och .exr- filer, som jag sammansätter i Da Vinci Resolve programmet till en hel kontinuerlig film.



Figur 17. Inom Movie render queue verktyget kan man ändra på olika tekniska detaljer för inspelningen som till exempel output formatet, bildrutor per sekund och renderings inställningar.



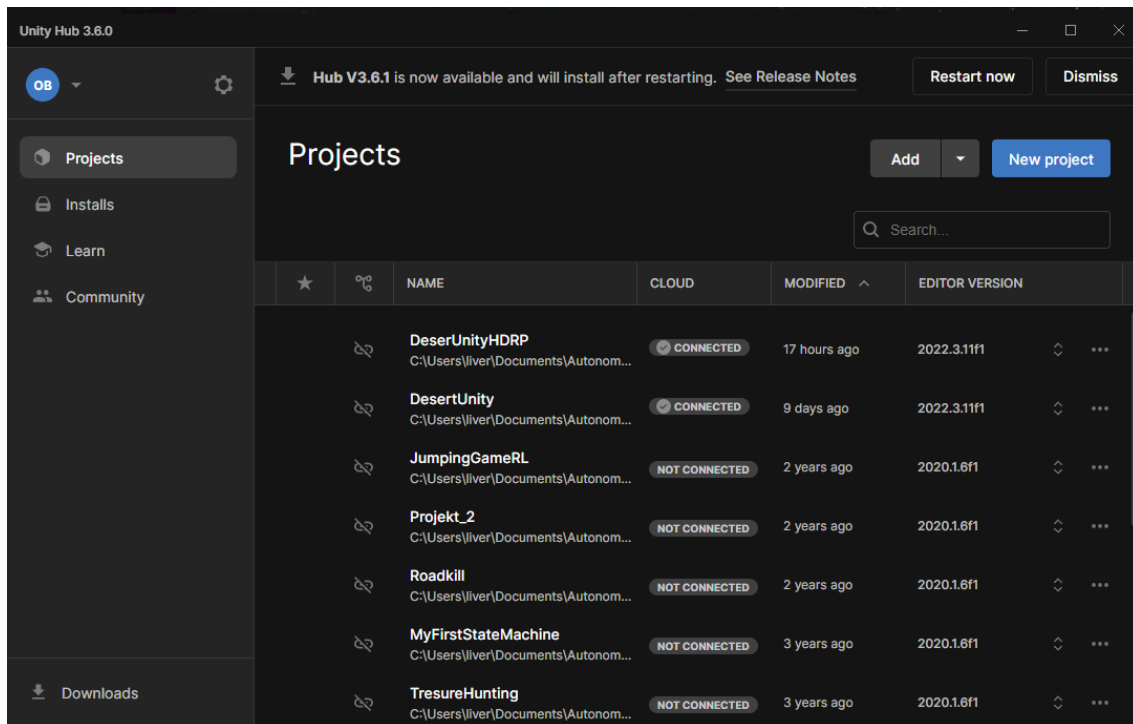
Figur 18. Jag ändrar till de outputinställningar som jag vill ha för filmen.

5 Att göra en kortfilm med Unity

Unity är den andra spelmotorn i vilken det skapas en kortfilm, för att sedan jämföra skillnaderna i processen att skapa en film mellan Unity och Unreal Engine 5. Även om processen och tekniken är till lik UE5 så finns det grundläggande skillnader mellan bägge spelmotorer. Det här avsnittet går igenom processen att skapa en kortfilm inom Unity för att sedan kunna jämföra resultaten mellan spelmotorerna i nästa kapitel. För projektet användes följande resurser: Blender och Unity.

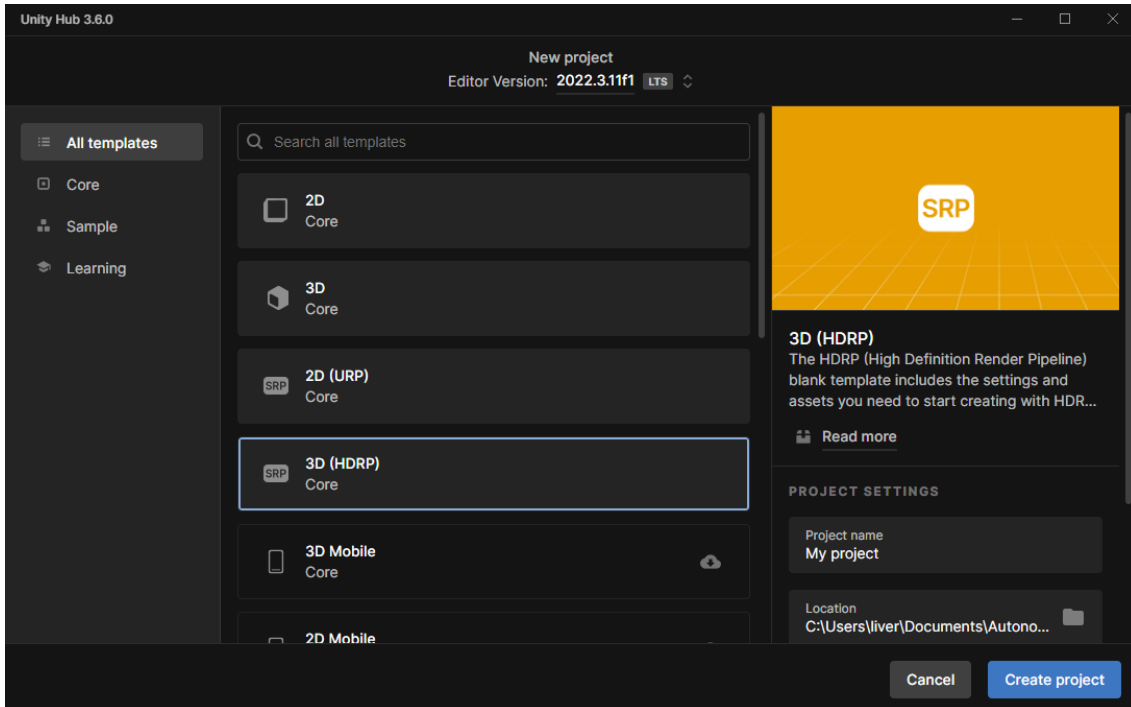
Arbetet börjar med installationen av Unity Hub. Unity Hub är ett program skapat av Unity Technologies för att underlätta hanteringen av själva Unity. Med hjälp av Unity

Hub gör det smidigt att byta mellan olika versioner av Unity, samt organiserar den användarens Unity projekt med diverse information om projektet i fråga.



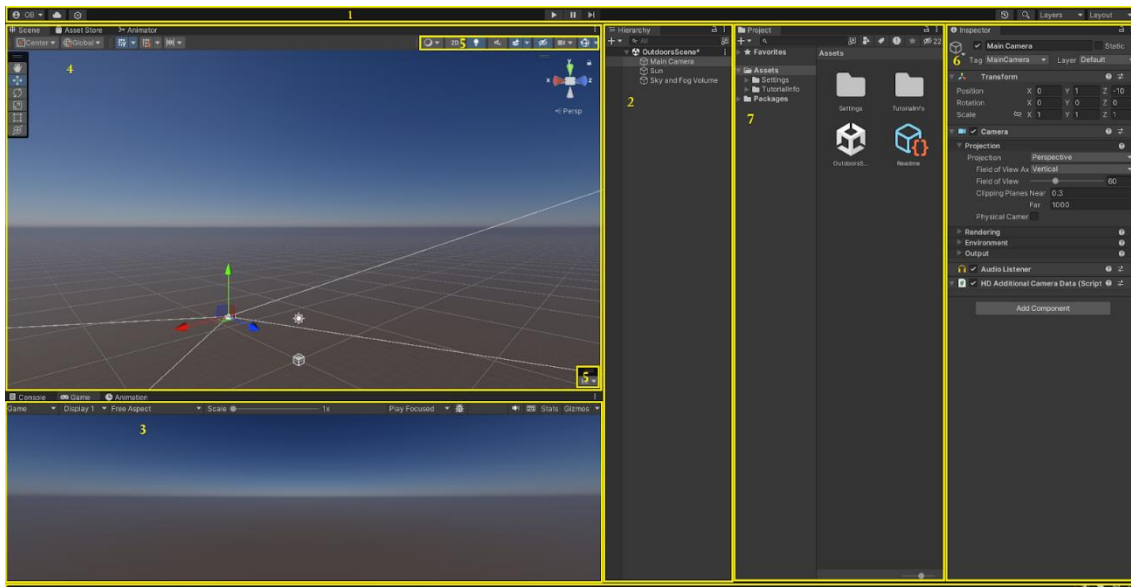
Figur 19. Unity hub användargränssnitt.

I Unity Hub kan du skapa projekt med olika tekniska egenskaper för att stöda ditt projekt. Du kan skapa projekt konfigurerade för till exempel 2D-applikationer, 3D-applikationer, AR, VR, blandad verklighet med mera



Figur 20. Jag använder 3D HDRP mallen som tillämpar sig bäst för mitt projekt.

För mitt projekt används Unity med versionen 2022.3.11f1 och för det nya projektet skapas ett tomt 3D HDRP (High-Definition-Render-Pipeline) med versionen 14.0.9. eftersom grafikmässigt stöder det arbetets ändamål bäst.



Figur 21. Standardutformning av användargränssnittet i Unity version x. De olika komponenterna är numrerade i ordning.

Bilden visar en tom scen inom Unity med standardutformning (layout). De olika komponenterna enligt ordningen i bilden.

1: Verktygsfältet.

Ger tillgång till ditt Unity-konto och Unity Cloud-tjänster. Det innehåller också en kontroller för Spel-läge, ångrad historik, unity-sökning, ett lager synlighetsmeny, och menyn för redigeringslayout.

2: Hierarkifönstret.

Är en hierarkisk textpresentation av varje spelobjekt i scenen. Varje objekt i scenen har en post i hierarkin, så de två fönstren är inbyggt kopplade. Hierarkin avslöjar strukturen för hur spelobjekt bifogas till varandra.

3: Spel-vyn.

Simulerar hur ditt slutgiltigt renderade spel kommer att se ut genom dina scenkameror. När du klickar på Spela-knappen börjar simuleringen.

4: Scen-vyn.

Tillåter dig att visuellt navigera och redigera din scen. Scen-vyn kan visa en 3D- eller 2D-perspektiv, beroende på vilken typ av projekt du arbetar med.

5: Överlagringar.

Innehåller grundläggande verktyg för att manipulera scen-vyn och spelobjekt inom den. Du kan också lägga till anpassade överlagringar för att förbättra ditt arbetsflöde.

6: Inspektörfönstret.

Tillåter dig att visa och redigera alla egenskaper hos det markerade spelobjektet. Eftersom olika typer av spelobjekt har olika uppsättningar egenskaper, ändras layouten och innehållet i inspektörfönstret varje gång du väljer ett annat spelobjekt.

7: Projektfönstret.

Visar ditt bibliotek med tillgängliga tillgångar som du kan använda i ditt projekt. När du importerar tillgångar till ditt projekt så visas de här.

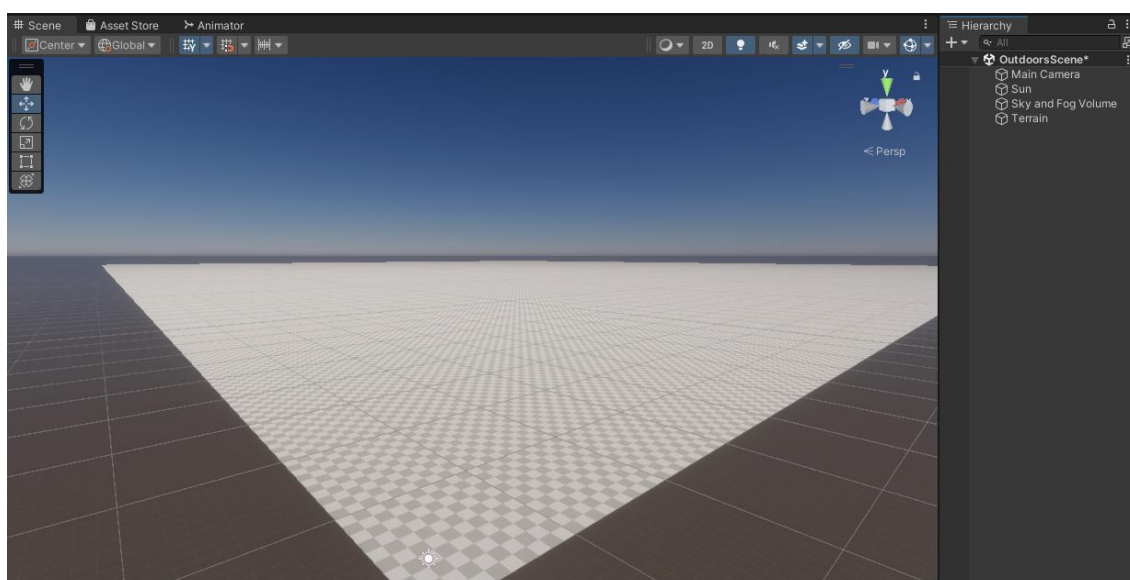
8: Statusfältet.

Ger meddelanden om olika Unity-processer och snabb åtkomst till relaterade verktyg och inställningar.

Jag påbörjar arbetet, vilket är att replikera min scen från UE5. Tanken är att skapa ett öken i Unity som så långt som möjligt skall påminna både grafiskt och miljömässigt

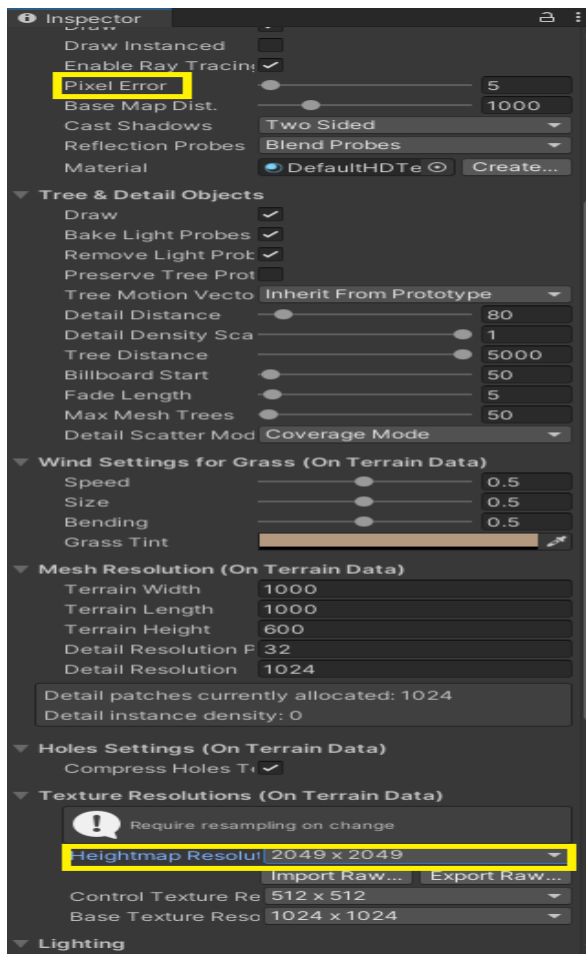
scenen i UE5. Jag söker upp en video i Youtube som går igenom skapandet av ett öken i Unity. Jag följer de steg som visas i Youtube videon (Roland C, 2022). Roland använder sig av olika slags tillgångar från Unity Asset Store. Jag laddar ner de tillgångarna som är gratis och kringgår de tillgångar som kostar pengar med mina egna lösningar. De gratis tillgångarna som jag laddar ner är: Substance 3D for Unity tillgången (tillgången importerar och tillåter anpassning av fysiskt baserade ämnes materialer skapade med Substance Designer med stöd för Unity Standard/Standard shader och HDRP), Stylized sand uniform material asseten (sand material för terrängen) och StampIT Collection - FREE examples (StampIt samlingen är en uppsamling av hög resolution 4K höjdmapper, stämplor och penslar som används för att stämpla egenskaper samt hela höjdmapper på Unity terrängen).

Jag skapar inom hierarkifönstret en ny tom terräng.



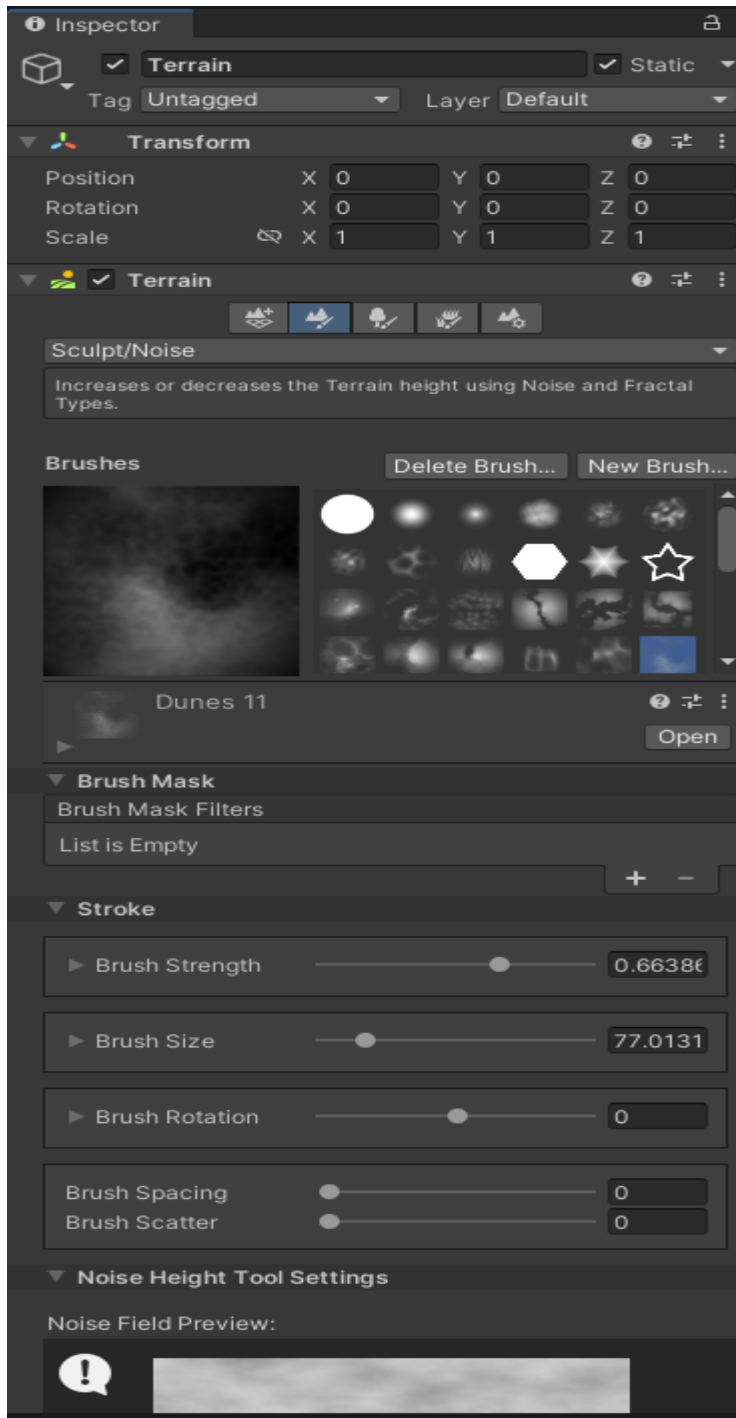
Figur 22. Blank terräng utan innehåll.

I inspektörfönstret minskar jag först Pixel Error inställningen till 1 för att få bättre noggrannhet mellan den genererande terrängen och terrängkartan Jag ändrar "Heightmap Resolutionen" till 2049 x 2049 eftersom det ändrar på pixelresolutionen på terrängens höjdmapper till mera lämplig för detta projekt.



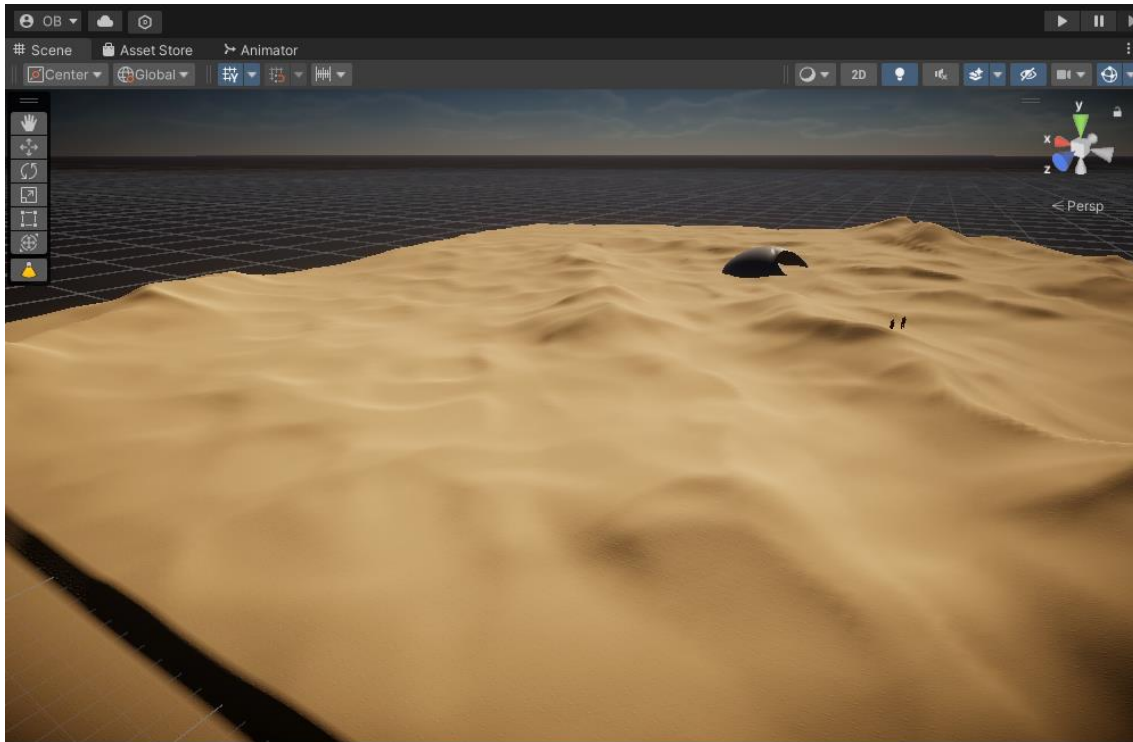
Figur 23. Inspektörfönstret med terrängens egenskaper.

Jag väljer terrängformnings verktyget vilket ger mig till användning olika slags terräng penslar att forma terrängen med.



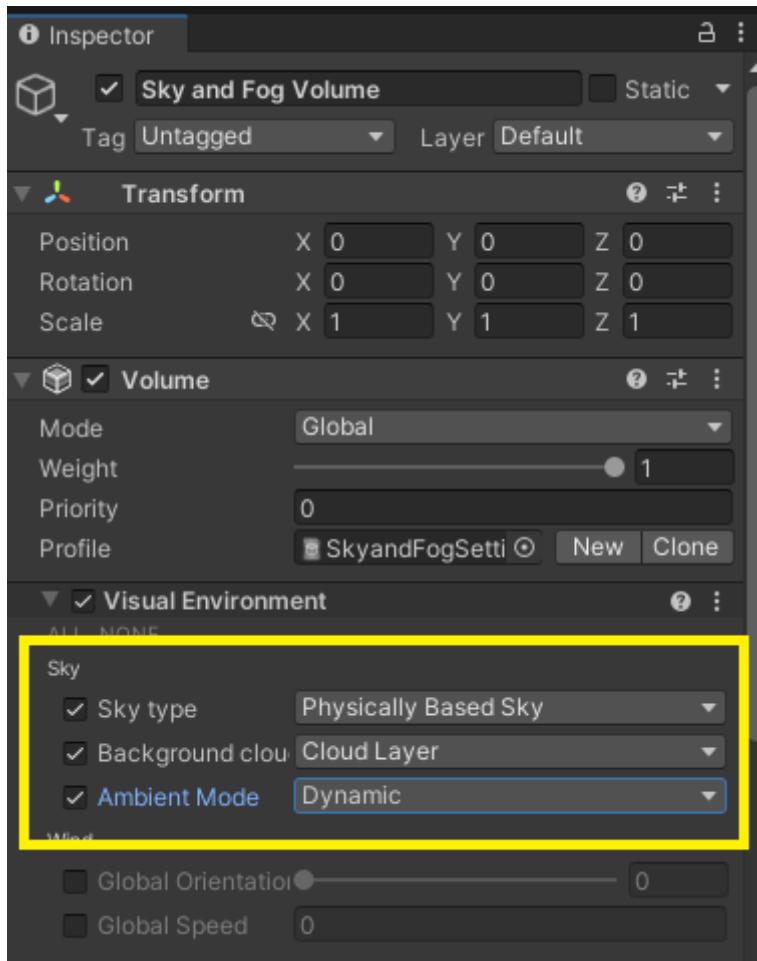
Figur 24. Skulpteringsverktyget inom terrängens inspektörfönster.

Jag använder en blandning av olika “Dune” penslar och påbörjar formningen av terrängen till ökenmiljö. Efter att terrängen ser passlig ut lägger jag på sand texturen på terrängen.



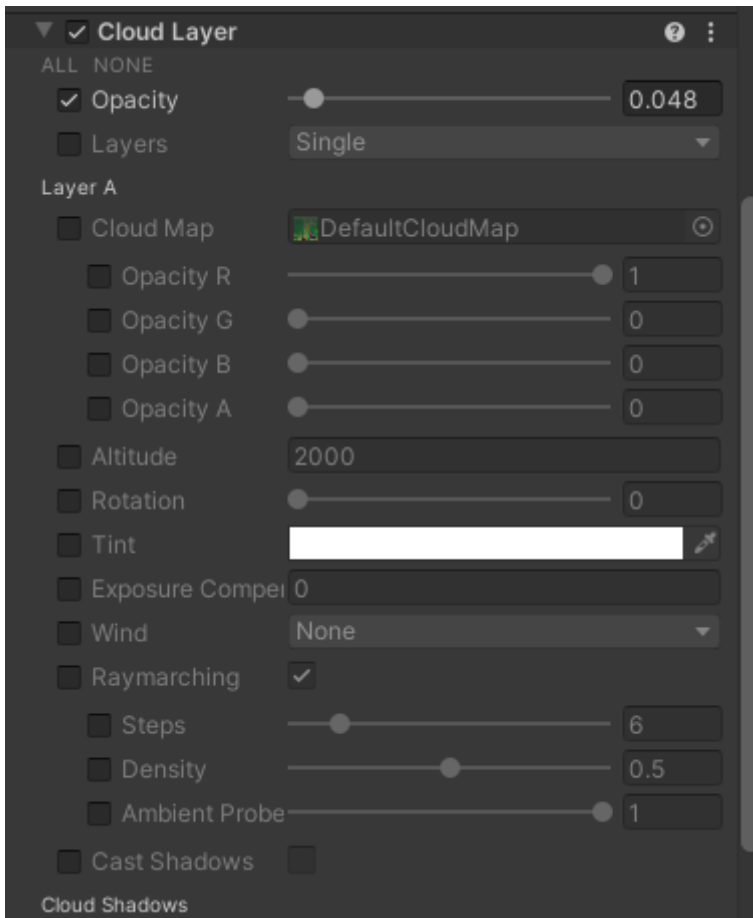
Figur 25. Terrängen efter att den är skulpterad.

I Sky and Fog Volume resursen aktiverar jag bakgrunds moln och sätter dess “ambient mode” inställning på dynamisk vilket styr den globala omgivning sonden så att den visar den nuvarande himmelen.

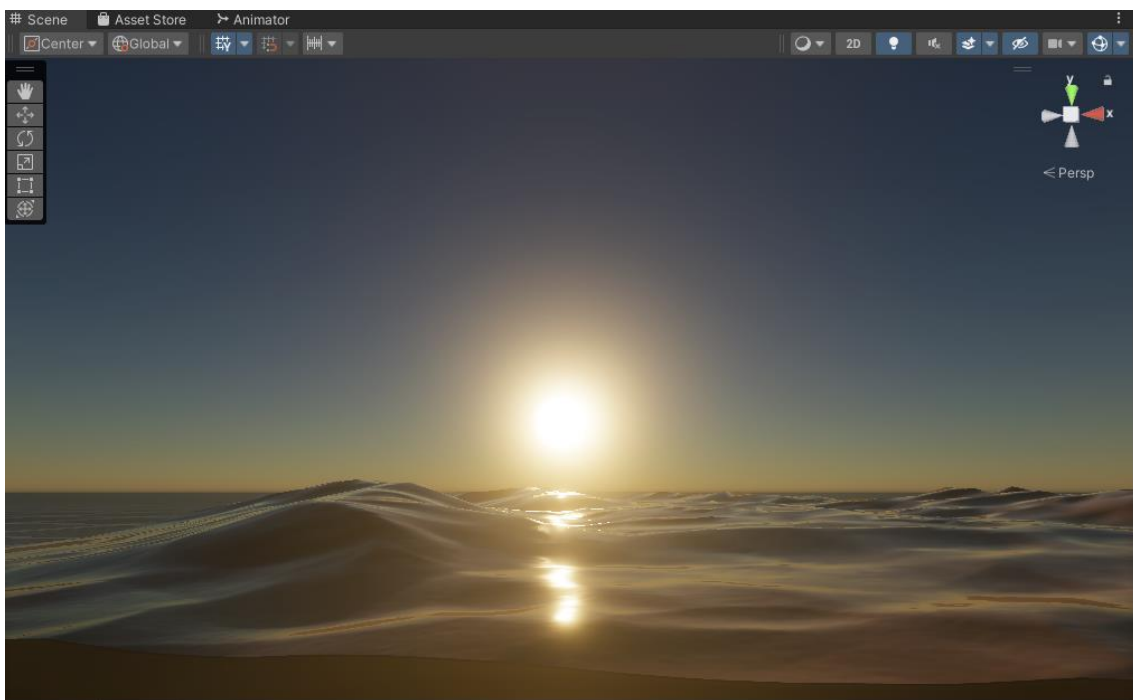


Figur 26. Inspektörfönstret med himmelens egenskaper.

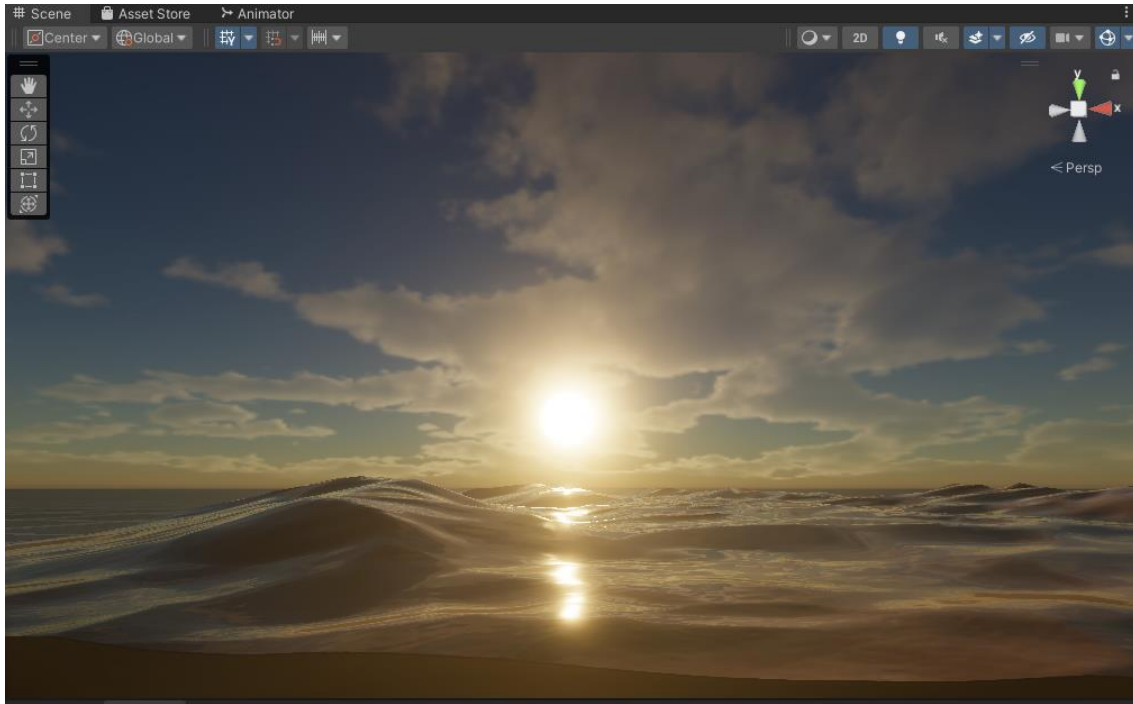
Jag lägger på Cloud Layer överskridelse på miljön och ändrar på opaciteten för att minska på styrkan av molnen.



Figur 27. Inställningar för miljöns molnsystem.

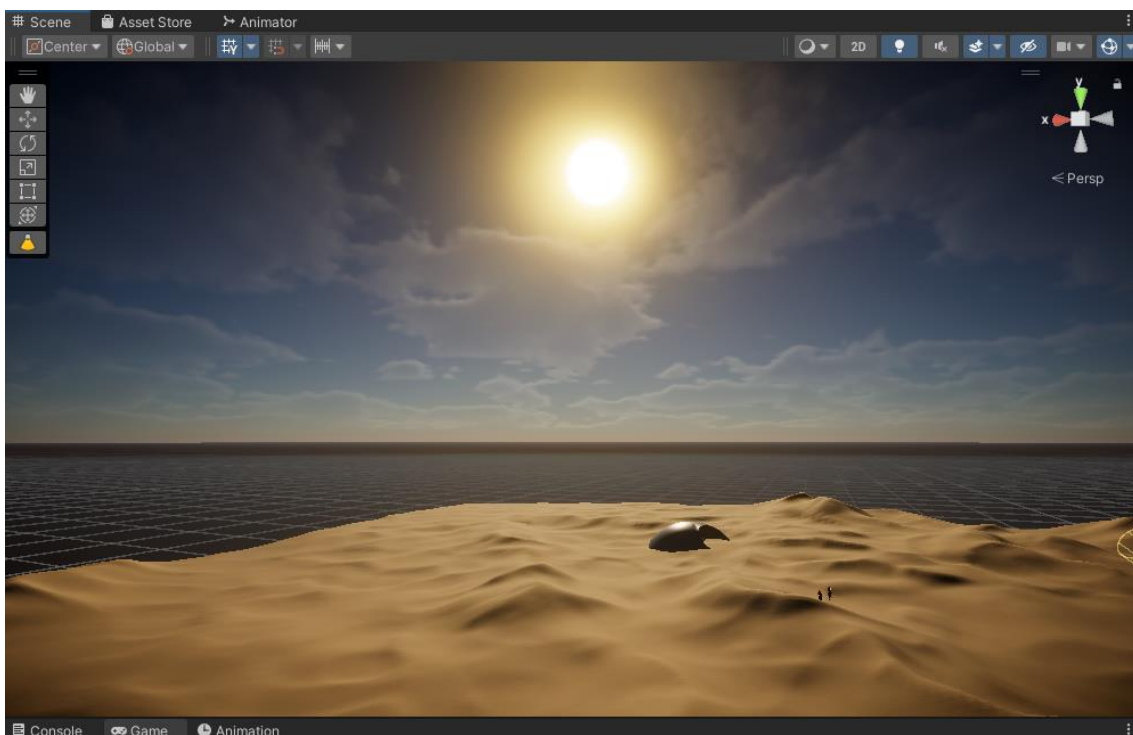


Figur 28. Projektet före jag aktiverar molninställningarna.

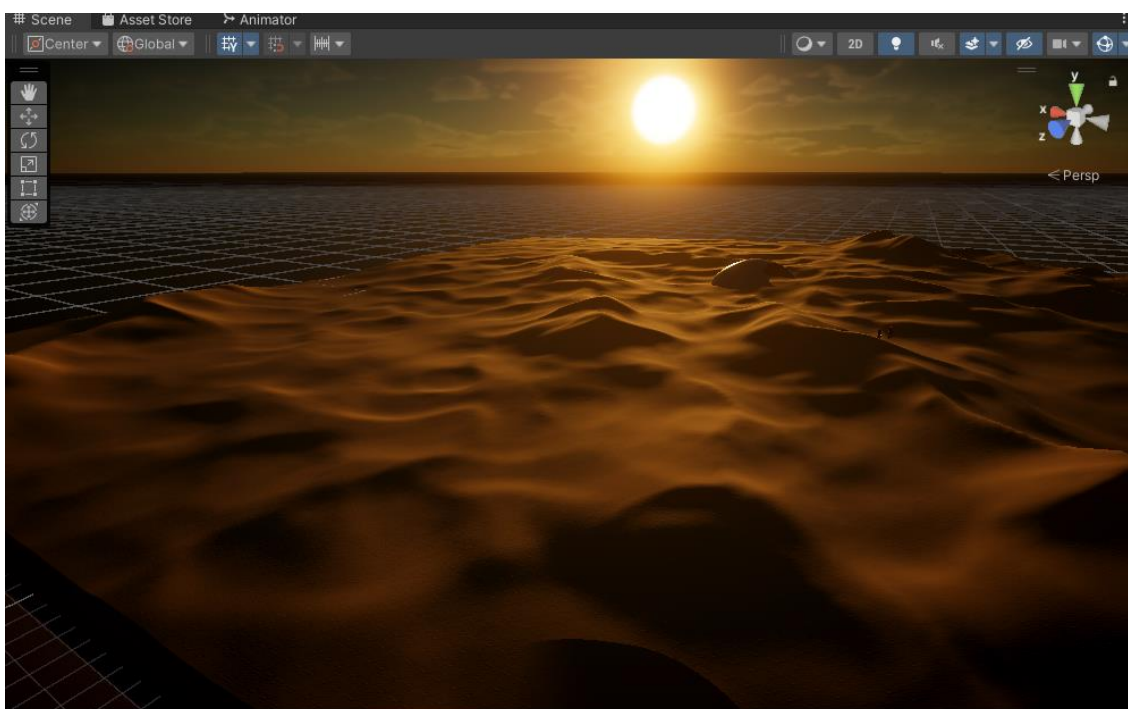


Figur 29. Projektet efter molninställningarna är aktiverade.

För att ändra på scenens belysning lägger jag en “Light Anchor” komponent till scenens sol, inom denna komponent ändrar jag på solens bana samt höjden enligt mina preferenser. Jag ändrar också på solens temperatur för att få scenen att se varmare ut.

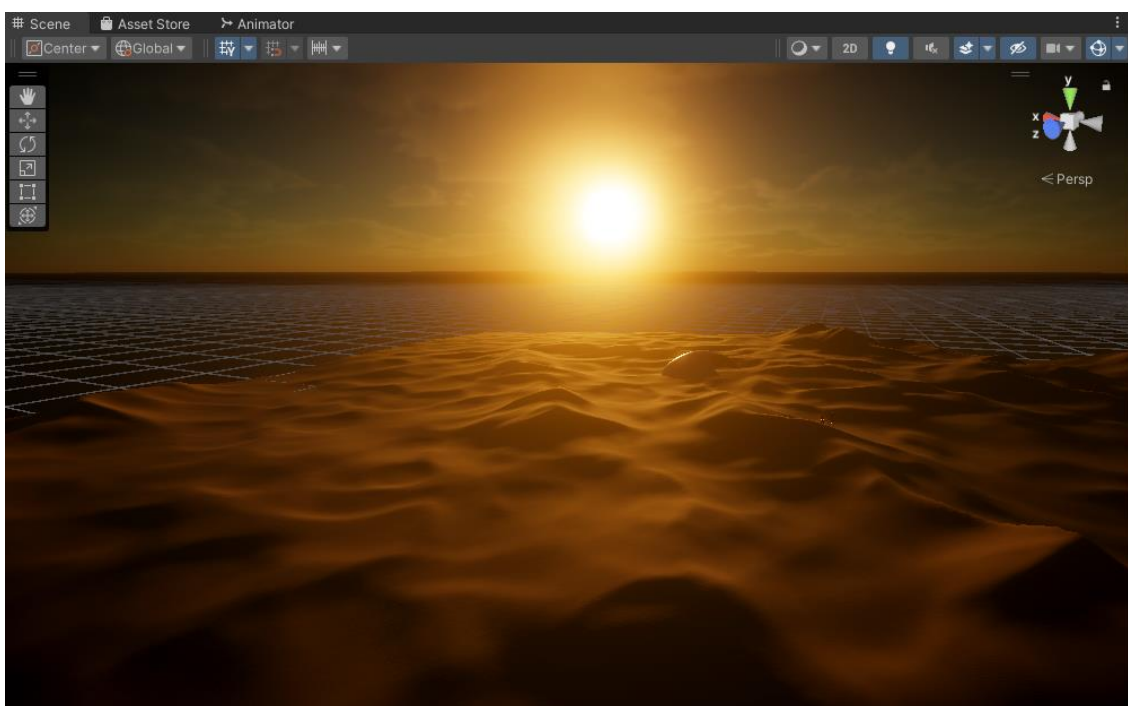


Figur 30. Projektet med Light Anchor komponenten.

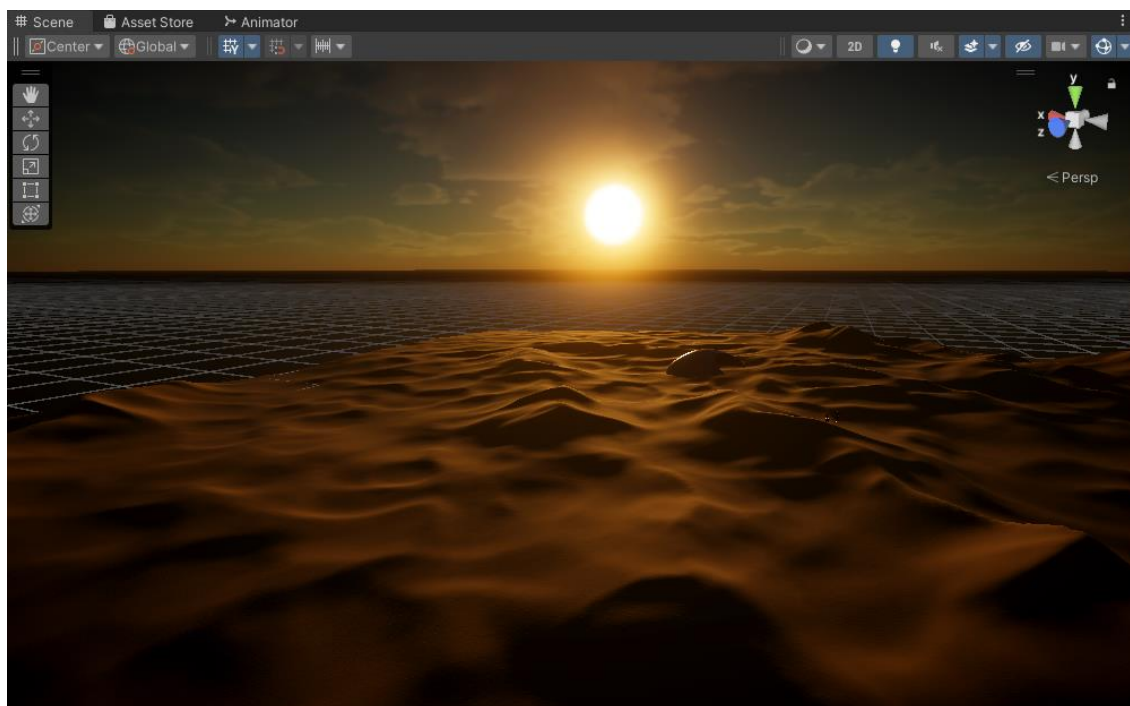


Figur 31. Projektet efter att jag ändrat på Light Anchor komponentens egenskaper.

Jag skapar en "Global Volume" resurs i vilken jag lägger en "Vignette" komponent som tillåter mig att ändra intensiteten på skuggningen av kameraramarna, jag sätter en till komponent, "Split Toning", i vilken jag kan ändra färgen av skuggorna och till sist sätter jag en "Bloom" komponent i vilken jag ändrar på intensiteten av solens strålningskraft samt ändrar jag på färgskalan av strålarna.



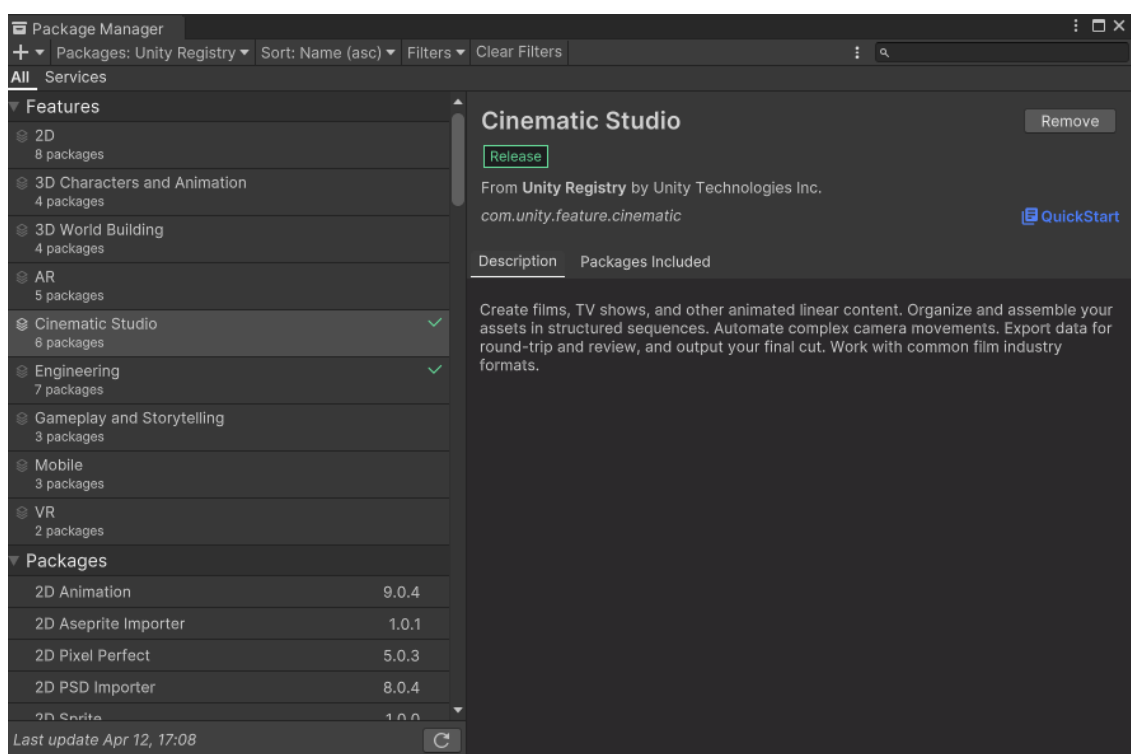
Figur 32. Global Volume resursen med Vignette, Split Toning och Bloom komponent.



Figur 33. Global Volume resursen efter förändringar på Vignette, Split Toning och Bloom komponenterna.

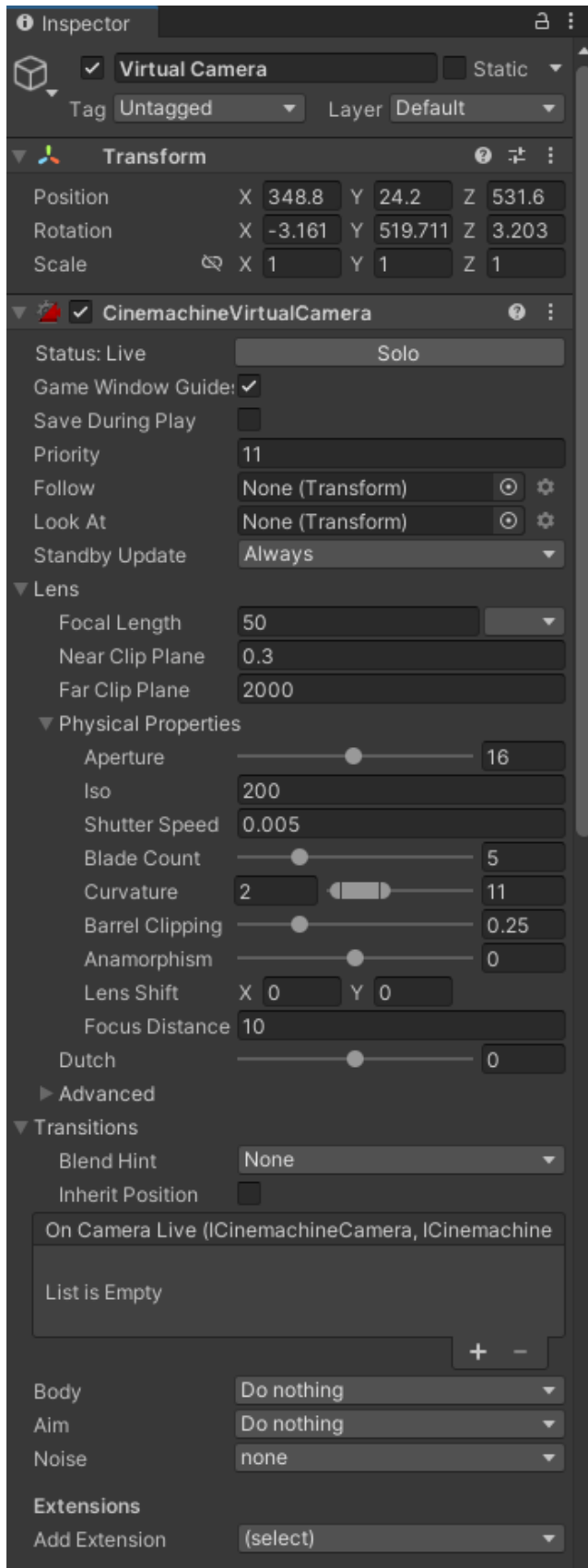
För att simulera ökenvind använde jag mig av ett partikelsystem tillgång som jag editerade att se ut som flygande sand och lagg det in i scenens terräng. Efter att scenens miljö är enligt mina preferenser så börjar jag importera de 3D-modeller som jag använde i min UE5 scen. Jag importerar klot modellen som jag använde i UE5, jag följer instruktionerna enligt (UGuruz, 2021) och importerar den som en FBX-fil in i Unity. Jag söker upp från Unitys Asset Store en gratis textur som ser lika ut som rost texturen i UE5 och lägger den på kloten i scenen. Efter det importerar jag samma karaktär jag använde i UE5. Jag aktiverar scenen för att försäkra mig om att karaktärens “Walk animation” fungerar, vilket den gör dock är animationstiden för kort. Med Youtube handledingsvideon (Imphenzia, 2022)) som förklarar mig hur jag får animationen att loopa oändligt. Efter resursen fungerar loopen, dock den har samma problem i UE att loopen repeterar sig alltid från start X koordinaten så jag går igenom olika forums och youtube videos för att åtgärda detta. Efter mycket sökande och misslyckande får jag loopen att fungera med hjälp av Unitys egna manual gällande modeller med “humanoid animations”, enligt resursen skulle jag ändra animations typen till humanoid inom inspectorn av FBX-filen, vilket fixade problemet och nu hade jag en karaktär som går oändligt framåt som i UE5 scenen. Jag försöker replikera UE5 och Unity scenen så pass lik som

möjligt, under processen märker jag att inställningarna påverkar på olika sätt i de olika programvarorna, till exempel i UE5 när jag ökar temperaturen blir miljön “varmare” medan i Unity när jag ökar temperaturen blir miljön “kallare”, vilket leder till att jag inte kan använda samma inställningar för att uppnå samma resultat. Att replikera himmelen med molnen från UE5 i Unity krävde mycket av ändringar och test på parametrarna gällande molnlagrens opacitet, och det var speciellt viktigt eftersom det hade en stor betydelse för scenens belysning och skuggor. Jag påbörjade filmprocessen av scenen efter att alla modeller, miljön och animationer var i skick. Unity har inte färdigt inbyggda verktyg för att skapa filmer, för att få dessa verktyg så söker jag upp ”Cinematic Studio” tilläggs paketet i Unitys paket hanterare. Cinematic Studio paketet innehåller Sequences verktyget, Cinemachine verktyget och Recorder verktyget.



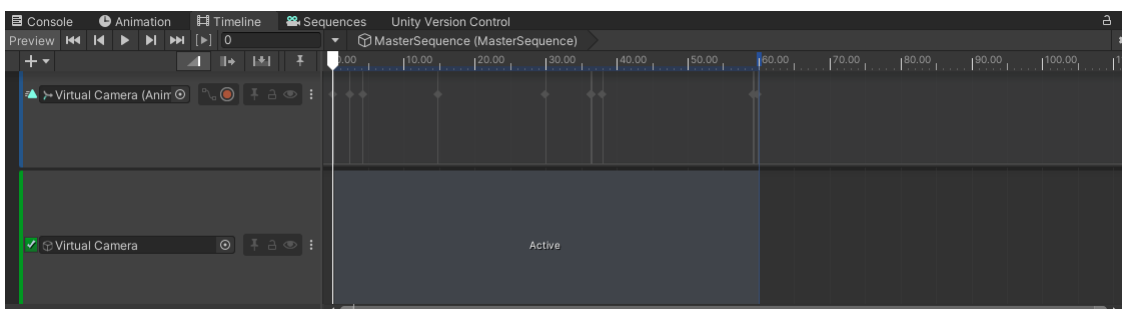
Figur 34. Cinematic Studio tilläggs paketet.

Med Cinemachine verktyget kan jag skapa en virtuell kamera in i scenen. I Unity kan man inte välja en färdig inställd super 35mm kamera så jag blir tvungen att manuellt ändra på bildformatet till 19.10 och brännvidden till 50mm. Eftersom scenen har olika belysningsinställningar kan jag inte använda samma bländaröppningsvärde som i UE5 projektet så jag prövar mig framåt med olika värden och beslutar att använda bländarvärdet 16.



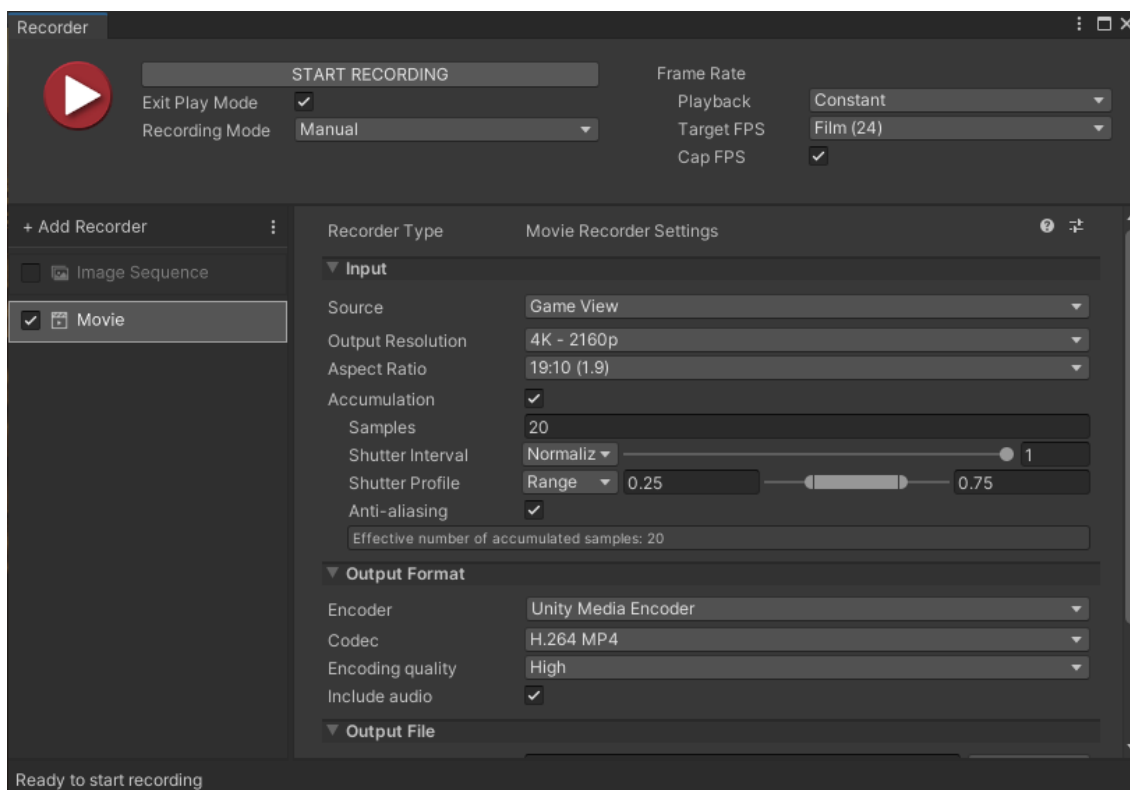
Figur 35. Cinemachine Virtual Cameras inspector i Unity.

Jag leder den virtuella kameran inom scenen och sparar nyckelrutor inom Sequencer så att programmet vet vilka x,y och z koordinater som den skall banda in.



Figur 36. Sequencer verktyget för att styra nyckelbildrutor och animationer vid inspelning.

För inspelning av filmen använder jag Recorder verktyget, som filmar scenen och sparar den i vald format.



Figur 37. Unitys Recorder verktyg som styr output formatet av inspelningen.

Recorder verktyget var lätt att förstå och det krävde inte mycket att få de önskvärda inställningarna för film outputten. För att resultatet ska vara så lik filmindustri standarden som möjligt har jag satt renderingsinställningarna enligt följande. “Target FPS” ska vara 24, outputresolutionen skall vara 4k, “Aspect Ratio” skall vara 19:10, “Samples” på 20 (ökar mängden delbilder, “sub-frames”, som fångas och ackumuleras bland två

påföljande bilder. Gentemot UE5, i vilket jag behövde använda mig av ett externt program (Da Vinci) för att sammansmälta bildsekvensen till en video, så Unitys recorder gör det för mig, d.v.s du behöver inte en extern resurs för att skapa videor av materialet som är renderat.

6 Resultat

Jag har färdiga kortfilmer som är skapade och renderade med Unreal Engine 5 och Unity. Målet var att skapa så pass likadana virtuella miljöer med lika rekvisita, omgivning, tekniska inställningar och inspelningsvinklar som möjligt för att sedan kunna jämföra de slutliga resulterade kortfilmerna och jämföra de båda spelmotorernas diverse styrkor och svagheter mot varandra för att ge svar till frågan att vilken spelmotor tillämpar sig bättre för användning inom film.

6.1 Jämförelse av filmresultaten - Resultat

Scenerna renderades med högsta tillgängliga grafikinställningar inom båda spelmotorerna och scenerna försöktes replikera så lik varandra som möjligt, så att man kunde göra en så pass objektiv jämförelse av de båda filmerna. Spelmotorernas slutprodukt jämförs med följande kriterier: filmatisk färgskala, objektens grafiska kvalité, skuggnings kvalité, och filmatisk överblick. Som stöd för jämförelsen används data i form av resultat från en enkätundersökning med 32 personers svar. Enkäten är formulerad så att svaranden väljer ett av de två alternativen för varje fråga. Alternativen i formulären är skärmdumpar från de filmer som inspelats tidigare i arbetet. Svaranden vet inte från vilken spelmotor alternativen är.

Målet för scenens färgskala var att den atmosfärliga effekten tillsammans med ljuskällan (solen) skulle ge en varm solnedgångs bild till en utomjordisk omgivning. För jämförelsen användes figur 38 och 39.



Figur 38. Unity jämförelsebild för filmatisk färgskala.

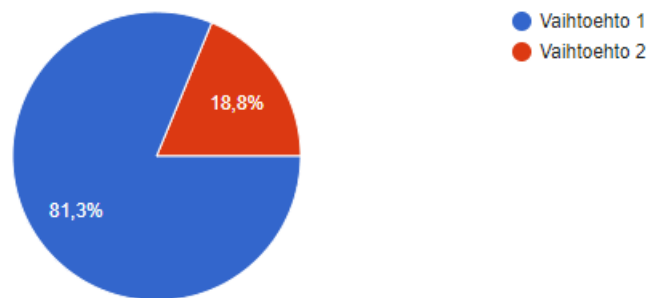


Figur 39. UE5 jämförelsebild för filmatisk färgskala.

Enkäten resulterade i följande resultat. Alternativ 1 är röster för Unity och alternativ 2 är röster för UE5. Enligt enkäten är färgskalan mera filmatisk i Unity med 81,3% av rösterna, vilket förmedlas i figur 40.

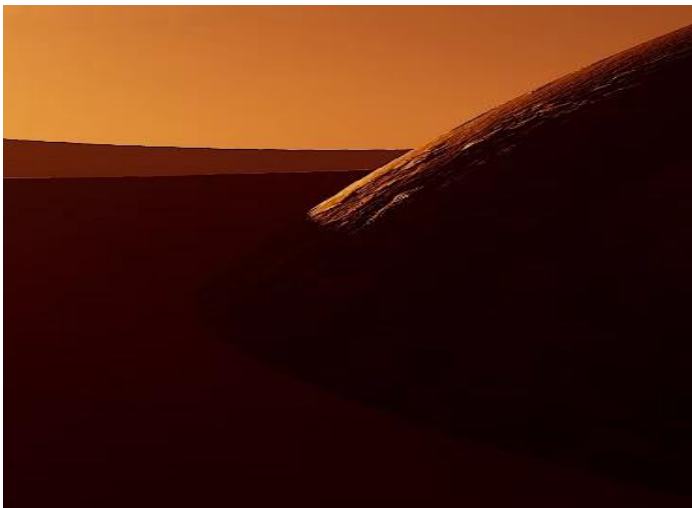
Vilken av dessa alternativ har en mera filmatisk färgskala?

32 vastausta



Figur 40. Enkätresultat för vilken spelmotor som har en mer filmatisk färgskala.

För att jämföra objektens grafiska kvalitet så användes figur 41 och 42



Figur 41. UE5 jämförelsebild för objektens grafiska kvalitet.

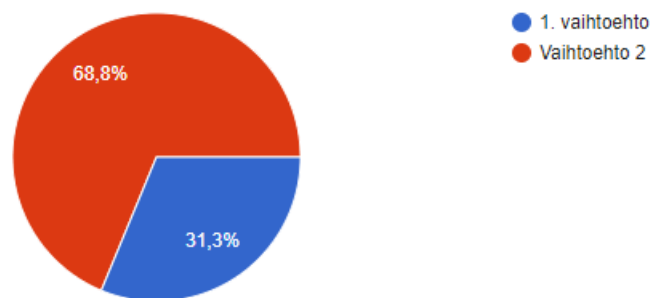


Figur 42. Unity jämförelsebild för objektens grafiska kvalitet.

Enkäten resulterade i följande resultat. Alternativ 1 är röster för UE5 och alternativ 2 är röster för Unity. Enligt enkäten är grafiska nivån av högre kvalitet i Unity med 68,8% av rösterna, vilket förmedlas i figur 43. Detta är mycket intressant resultat eftersom objektet i UE5 är uppbyggd av 14 343 826 stycken nanitevertexar och 7 535 406 stycken nanitetrianglar i kontrast till objektet i Unity som är uppbyggd av 6 614 237 stycken vertexar och 8 332 054 stycken trianglar.

Vilken bild av dessa alternativ har bättre grafisk kvalitet?

32 vastausta



Figur 43. Enkätresultat för vilken spelmotor som har bättre grafisk kvalitet.

För att jämföra skuggningskvaliteten i scenen så användes figur 44 och 45.



Figur 44. UE5 jämförelsebild för skuggningskvaliteten.

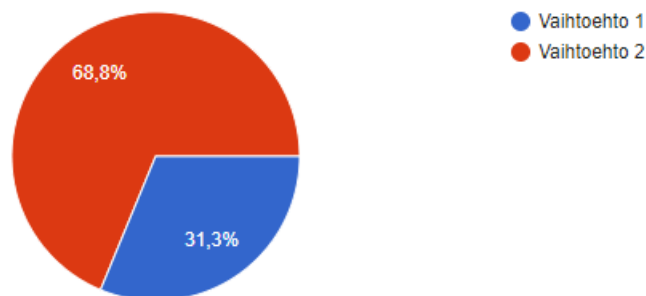


Figur 45. Unity jämförelsebild för skuggningskvalitén.

Enkäten resulterade i följande resultat. Alternativ 1 är röster för UE5 och alternativ 2 är röster för Unity. Enlig enkäten är skuggningskvalitén av högre kvalité i Unity med 68,8% av rösterna, vilket förmedlas i figur 46.

Vilken bild av dessa alternativ har bättre skuggningseffekt?

32 vastausta



Figur 46. Enkätresultat för vilken spelmotor som har bättre skuggningskvalité.

För att jämföra vilken spelmotor som har en mera fotorealistic överblick så användes figur 47 och 48.



Figur 47. Unity jämförelsebild för fotorealism.

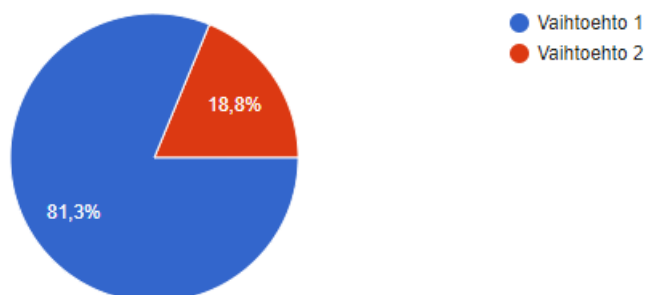


Figur 48. UE5 jämförelsebild för fotorealism.

Enkäten resulterade i följande resultat. Alternativ 1 är röster för Unity och alternativ 2 är röster för UE5. Enligt enkäten har Unity en mera fotorealistic syn med 81,3% av rösterna.

Vilken bild av dessa alternativ har en mera fotorealistisk syn?

32 vastausta



Figur 49. Enkätresultat för vilken spelmotor som har en mera fotorealistisk syn.

Alla dessa komponenter bygger upp tillsammans den stora helhetsbilden av hur filmatisk scenen är. Även om båda spelmotorerna lyckas bra med att skapa en filmatisk överblick på scenen, så med all kritik sammansatt kan man komma till den slutsatsen att med samma mängd arbete och kunskap inom båda spelmotorerna så gör Unity ett bättre jobb över UE5 i att skapa en virtuell filmscen som påminner en äkta fysisk scen.

7 Egna funderingar efter att ha använt Unity och UE5

När jag påbörjade detta arbete hade jag knappt någon erfarenhet när det kommer till användning av spelmotorer och ingen kunskap gällande konsten av cinematografi. Jag var tvungen att lära allting från grunden med Unity och Unreal Engine 5 vilket gav till mig en bra insikt om hela processen att bygga upp en miljö med aktörer och animationer från början till slut och möjligheten att jämföra de båda spelmotorerna sinsemellan från en nybörjares perspektiv.

I detta avsnitt kommer jag från mitt perspektiv att jämföra skillnaderna i mjukvarornas egenskaper som är relevanta inom tillämpning för cinematografi och spelmotorns användarvänlighet, för att kunna svara på frågan att för hurudant filmprojekt tillämpar sig någondera av spelmotorerna bäst. Spelmotorerna kommer att jämföras med varandra enligt följande kriterier: Användargränssnittets användarvänlighet, tekniska funktioner, spelmotorns prestanda, renderingstiden, den tillgängliga externa stöd från internet-resurser, virtuella kamerans prestanda och filminspelningsverktygets funktioner.

Användargränssnittet och den allmänna vyn har en modernare och klarare blick i UE5 över Unity, men UE5 har även fler verktyg och andra tekniska funktioner inom gränssnittet vilket kan överväldiga en oerfaren utvecklare till en början gentemot Unitys mera karga men enkla användargränssnitt.

Tekniska funktioner i spelmotorerna är på grundliga nivån lik varandra. Bägge spelmotorerna har möjligheten att tillämpa egen kod i projektet för att göra mera skräddarsydda lösningar för ens egna projekt. Att bygga likadana scener med komponenter som belysning, atmosfärliga effekter, ljusankare och 3D-objekt var möjligt och det fanns ingen funktion som ena mjukvaran skulle ha sakna som skulle ha gjort det omöjligt att replikera någonting från den andra mjukvaran. Någonting som Unity inte lyckades med på samma nivå som UE5 var deras Bolt tilläggsprogram som inte var lika självklar att använda som Unreal Engines blueprint-system. Det var till exempel enklare att skapa en simulerad ökenvind genom att utnyttja ett importerat partikelsystem istället för att visuellt skriptat det såsom i UE5.

Prestandan för spelmotorerna varierade mycket, i Unreal Engine kunde man skapa och bygga scenen med medelnivåns grafiker och sedan enkelt byta upp till de högsta tillåtna grafikerna, då man skulle rendera filmen, vilket var mycket behändigt i utvecklingsstadiet. Dock när det kom till själva renderingen av filmen med min PC:s uppbyggnad så var den högsta grafiska kvaliteten mycket krävande för PC:ns prestanda och under renderingen kraschade datorn ofta. Gentemot i Unity var det inte lika stora prestandaproblem med de högsta grafikerna aktiverade, dock krävde det mera arbete att ändra på tillgångarnas grafiska nivå eftersom det inte fanns en inställningsalternativ som automatiskt satt kvaliteten till det högsta som i UE5, själva renderingen var smärtfriare och kraschade inte. Även om UE5 hade större krav på min PC:s prestanda med renderingen, så blev den snabbare på att rendera scenen efter varje ny renderingskörning. Första renderingen av scenen kunde ta upp till 4 timmar medan den sista renderingen kunde vara så snabb som ~16 minuter, vilket blev den slutliga genomsnittliga renderingslängden. I Unity tog renderingen kring ~45 minuter men mjukvaran tycktes inte lära sig att köra renderingen snabbare efter tidigare renderingar. När det kommer till rent prestandakrav

av spelmotorn så är Unity mer förlåtande. Jag körde 4 stycken experimentrenderingar i både UE5 och Unity för att mäta den genomsnittliga renderingstiden, detta visas i figur 50.

	Unreal Engine 5	Unity
Rendering 1	17 min. 25 sek.	43 min. 6 sek.
Rendering 2	16 min. 7 sek.	43 min. 30 sek.
Rendering 3	16 min. 41 sek.	46 min. 23 sek.
Rendering 4	16 min. 13 sek.	45 min. 7 sekunder.
Genomsnittlig renderings- tid	~16 min. 36 sekunder.	~44 minuter. 31 sekunder.

Figur 50. Figuren beskriver renderingstider och den genomsnittliga renderingstiden för att rendera en 59 sekunder lång inspelning.

Tillgänglig stöd via internet-resurser fanns i mycket stora grad för bägge spelmotorer och för detta arbete fanns det ingen del för vilken man inte kunde finna stöd online.

Ett av nyckelverktygerna för detta arbete var den virtuella kameran i spelmotorerna. Både Unity och UE5 hade möjligheten att sätta in de inställningarna i de virtuella kamerorna som rekommenderas för film, men i UE5 så kunde man direkt välja kameramodellen super 35mm från ett flertal olika kameror, medan i Unity var jag tvungen att mera manuellt införa de inställningarna som krävs för att simulera super 35 mm kamera. Utdatan från de virtuella kamerorna var den samma och kvalitén påverkades bara av spelmotorns grafiska kapabilitet.

Verktygen för att inspela och rendera sekvensen inom både Unity (Recorder) och UE5 (Movie Render Queue) hade möjligheten för samma filminställningar vilket underlättade jämförelsearbetet. Den största skillnaden var att Unitys recorder kunde rendera i färdig video format (H.264 MP 4), medan i UE5 var jag tvungen att rendera i exr-format och sedan med det externa Da Vinci Resolver programmet göra exr-filen till samma H.264 Mp4-format som i Unity.

Med alla dessa bidragande faktorer och efter min personliga arbetsprocess anser jag att det är effektivare att skapa filmer med Unity, speciellt för lågproduktioner och kortfilmer inom spel eftersom de tekniska kraven på ens PC:s prestanda är lägre för att utnyttja spelmotorn fullt än i UE5 som var mycket prestationstung. Om man däremot vill åstadkomma en högre fotorealistic nivå bör man använda sig av UE5 för filmskapande. Som jag i tidigare kapitel har behandlat om tillämpning av spelmotorteknik inom filmbranschen, och hur filmbranschen utvecklas i riktning mot att bygga 3D-miljöer i vilken tekniken skall samarbeta med det praktiska över att fullkomligt ersätta den, så tillämpar sig UE5 både tekniskt sett och grafiskt sett bättre för denna agenda.

	Unreal Engine 5	Unity
UI användarvänlighet		x
Tekniska funktioner	x	
Spelmotorns prestanda		x
Renderingstid	x	
Tillgänglig online stöd	x	x
Virtuella kamerans prestanda	x	
Filminspelningsverktygets funktioner		x

Figur 51. Jämförelse mellan av funktioner och egenskaper mellan UE5 och Unity. X i respektive spalt betyder att spelmotorn i fråga presterade bättre i detta sammanhang. X i samma rad på bägge spelmotor betyder att spelmotorerna presterade lika bra.

8. Slutsatser

Syftet med detta arbete var att studera hur filmbranschen inom de senaste åren har implementerat spelmotorteknik för VFX-effekter, och att jämföra de två spelmotorerna Unreal Engine 5 och Unity i deras egenskaper att skapa fotorealistic filmer. Jag anser att jag har lyckats med att få en grundlig bild över orsakerna till att filmbranschen använder sig av spelmotorer för 3D-miljöer och andra VFX-effekter, samt även insikt om teknikens tillämpning och sammansmältning inom praktisk scenografi i teatervärlden. Jag anser också att jag lyckats göra en neutral och opartisk jämförelse av Unreal Engine

5 och Unitys tekniska egenskaper och kapabilitet när det kommer till att skapa filmer från en nybörjares perspektiv och med begränsade resurser. Det jag inte lyckades i detta arbete var att få en intervju av någon finsk företag som använder sig av denna spelmotor teknik för VFX-effekter i praktiken. Jag tycker att det skulle ha gett god information som skulle ha vidare förstärkt detta arbetes forskningresultat. Resultatet från enkäten var överraskande, min personliga åsikt från filmresultatet är att Unreal Engine 5 hade bättre kvalitet gällande grafik, belysning, skuggning och filmatisk överblick, jag skrev tidigare i arbetet med basis på olika artiklar om hur UE5 har en brantare inlärningskurva så kanske detta reflekteras också bra med dessa resultat. För framtida forskning skulle man kunna kontakta aktörer inom både film och marknadsföringsbranschen för att få en djupare analys av möjligheterna att tillämpa denna teknik inom respektive bransch.

Källor

Boundless Entertainment. (2022). Unreal Engine for Filmmakers - Cinematic Camera Settings & Setting up Virtual Camera.

<https://www.youtube.com/watch?v=gFO0qhdLKec>

Cinemyscope. (2023). Unreal 5.2 Cinematic Camera Tutorial [video]. Youtube.

<https://youtube.com/watch?v=hftxCXvcJ9g>

CGHERO. (u.å). What is Unreal Engine?

<https://cghero.com/glossary/what-is-unreal-engine>

Dealessandri, M. (2020). What is the best game engine: is Unreal Engine right for you?, Games Industry.

<https://www.gamesindustry.biz/what-is-the-best-game-engine-is-unreal-engine-4-the-right-game-engine-for-you>

Epic Games. (u.å). Features.

<https://www.unrealengine.com/en-US/features>

French, J. (2022, Augusti). Is Unity free? (and will you ever need Unity Pro?) - 2024 Update

<https://gamedevbeginner.com/is-unity-free/>

Geniuscrate. (u.å). The Unity vs Unreal Engine Dilemma: Exploring Pros and Cons.

<https://www.geniuscrate.com/the-unity-vs-unreal-engine>

Imphenzia. (2022) Animated Character from BLENDER 2.93 To UNITY 2021 with correct Rotation. [video] Youtube.

<https://www.youtube.com/watch?v=B12xBW-L7uU>

Interesting Engineering. (2016, November). How Do Game Engines Work?

<https://interestingengineering.com/innovation/how-game-engines-work>

Kevuru Games. (2023, Mars). Unity VS Unreal Engine: Pros and Cons.

<https://kevrugames.com/blog/unity-vs-unreal-engine-pros-and-cons/>

Latvis. (2023, Mars). Welcome to the Extraordinary World of LED Volume Walls.

<https://www.bhphotovideo.com/explora/video/news/intro-to-using-led-volume-walls>

MDN. (2023, Juni). Texel.

<https://developer.mozilla.org/en-US/docs/Glossary/Texel>

Modo. (u.å). UDIM Workflow.

https://learn.foundry.com/modo/901/index.html#help/pages/uv-ing/udim_workflow.html

Muir, M. (2022, Juni). Alembic.

Epic Games.

<https://dev.epicgames.com/community/learning/tutorials/ryax/unreal-engine-asset-pipeline>

Olav3D Tutorials. (2023). How to Make Mixamo Animations Last Forever Using Blender [video]. Youtube.

<https://www.youtube.com/watch?v=hMUzHlo1faE>

Openusd. (u.å). What is USD?

<https://openusd.org/release/intro.html>

Painter, H. (2021, December). What is the Moiré Effect in Photography & How to Avoid it.[Blogg inlägg]. Hämtad från

<https://wavelength.focuscamera.com/what-is-the-moire-effect-in-photography-how-to-avoid-it/>

Roland C. (2022). Setting up a Desert Scene in 4 minutes [video]. Youtube.

<https://www.youtube.com/watch?v=SJ4XPxADa1g>

Stream. (2022, Januari). Unreal vs. Unity for Beginners: Choosing the Best Gaming Engine.

<https://getstream.io/blog/unreal-unity-engine/>

The University of Michigan Center for Academic Innovation. (2020). How Disney's the Lion King became a pioneer in the use of virtual reality. [Blogg inlägg]. Hämtad från

<https://ai.umich.edu/blog-posts/how-disneys-the-lion-king-became-a-pioneer-in-the-use-of-virtual-reality/>

Unit Led. (2023). The Mandalorian LED Wall: Ultimate Introduction is Here!

<https://www.unit-led.com/mandalorian-led-wall>

Unity Technologies. (2020, Juni). Visual Scripting with Bolt.

<https://docs.unity3d.com/2019.3/Documentation/Manual/VisualScripting.html>

Unity Technologies. (2024, Maj). Introduction to render pipelines.

<https://docs.unity3d.com/Manual/render-pipelines-overview.html>

UnrealityBites. (2022). Create a desert landscape procedurally [video]. Youtube.

https://www.youtube.com/watch?v=0vDnDERh_2Q

UnrealityBites. (2023). Desert Sand Blowing in the Wind [video]. Youtube.

<https://www.youtube.com/watch?v=iyS9hU1XWP4>

Vection. (2022 November). Everything You Need to Know About FBX Files: A Comprehensive Guide.

<https://vection-technologies.com/blog/Everything-You-Need-to-Know-About-FBX-Files-A-Comprehensive-Guide/>

Wikipedia. (u,å) Cave automatic virtual environment.

https://en.wikipedia.org/wiki/Cave_automatic_virtual_environment

Wikipedia. (u,å). Powerwall.

<https://en.wikipedia.org/wiki/Powerwall>

YR3Design. (2022). Exporting from Blender to Unreal Engine 5 [video].

<https://www.youtube.com/watch?v=IRbEkhs5L58>

Zenva. (2023, November). What is Unity? - A Top Game Engine for Video Games.

<https://gamedevacademy.org/what-is-unity/>