Tomi Rämö

PELIOHJELMOINTIA UNITYLLA

Tieto- ja viestintätekniikan koulutusohjelma

2019

samk
Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

.

PELIOHJELMOINTIA UNITYLLA

Rämö, Tomi
Satakunnan ammattikorkeakoulu
Tieto- ja viestintätekniikan koulutusohjelma
Marraskuu 2019
Sivumäärä: 27
Liitteitä: -

Asiasanat: peliohjelmointi, videopelit, pelisuunnittelu

Opinnäytetyössä tutkittiin, miten Unity-pelimoottorilla suunnitellaan peli C#-ohjelmointia käyttäen. Pelinkehitysvaiheet kuvattiin opinnäytetyössä.

Työssä osoitettiin, että nykypäivän pelinkehitys on helpottunut pelimoottoreita käyttämällä, etenkin Unity-pelimoottorilla.

Työn teoriaosuuksissa kerrotaan videopelien historiasta, Unity-pelimoottorista, vertaillaan muita pelimoottoreita ja katsotaan, miten kahdella eri käyttöjärjestelmällä kehitellään peli.

GAME PROGRAMMING WITH UNITY

Rämö, Tomi
Satakunta University of Applied Sciences
Degree Programme in Information and Communication Technology
November 2019
Number of pages: 27
Appendices: -

Keywords: Game Programming, Videogames, Game Development

How to design a game with the Unity game engine using C# programming was examined in the thesis.

It is shown that developing a game has been eased by the use of game engines, especially the Unity game engine.

The history of video games, the Unity game engine, the comparison of other game engines, and how two different operating systems are developed for the game are described in the thesis theory sections.

LYHENNELUETTELO

2D = Second Dimension

3D = Third Dimension

API= Application Programming Interface

APK= Android Application Package

AR = Augmented Reality

ESA = Essential Facts

FPS = Frames Per Seconds

HDRP= High Definition Render Pipeline

JDK = Java Development Kit

LWRP = Lightweight Render Pipeline

MMORPG = Massive Multiplayer Online Role Playing Game

NDK =Native Development Kit

OS X = Apples Operating System

RPG = Role Playing Game

SDK = Software Development Kit

SRP= Scriptable Render Pipeline

VR = Virtual Reality

SISÄLLYS

1 JOHDANTO

Tässä opinnäytetyössä ohjelmoidaan Unity-pelimoottorilla peli C# -ohjelmointikielellä ja käydään pelintekoprosessi vaihe vaiheelta läpi. Syy opinnäytetyön aiheen valintaan oli selkeästi se, että pelinkehitys on mielenkiintoinen ja ajankohtainen aihe ja se, oli myös syy miksi allekirjoittanut haki opiskelemaan Satakunnan ammattikorkeakouluun tieto- ja viestintätekniikkaa. Työ, osoittaa kuinka helppoa ja nopeaa Unity-pelimoottorilla on tehdä peli.

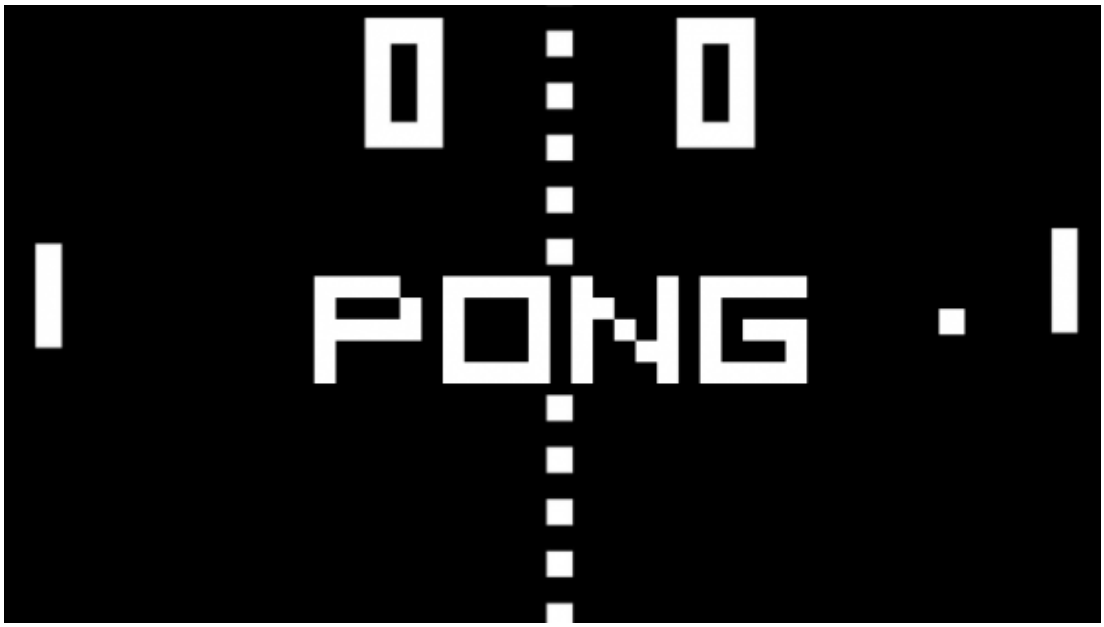
Työssä esitettään syitä, miksi valita Unity-pelimoottori muista useista pelimoottoreista vertailemalla niitä. Pelinkehitysprosessia on helpotettu nykypäivänä, sillä pelejä tehdään paljon ja kilpailu alalla on kovaa. Pieni kokemus ohjelmoinnista vaaditaan etenkin C#- ja JavaScript-ohjelmointikielillä, että tietää miten Unityn koodiskriptit toimivat pelissä. Allekirjoittanut tutustui Unity-pelimoottoriin vertaistensa ja oman kiinnostuksen kautta, käyttäen Unity-pelimoottoria ja opiskellen sen käyttöä.

2 VIDEOPELIEN HISTORIA

. ”Vuonna 1889 Fusajiro Yamauchi perusti pelikortteja valmistavan yrityksen, josta tuli myöhemmin Nintendo”. (Ranta. 2010.)

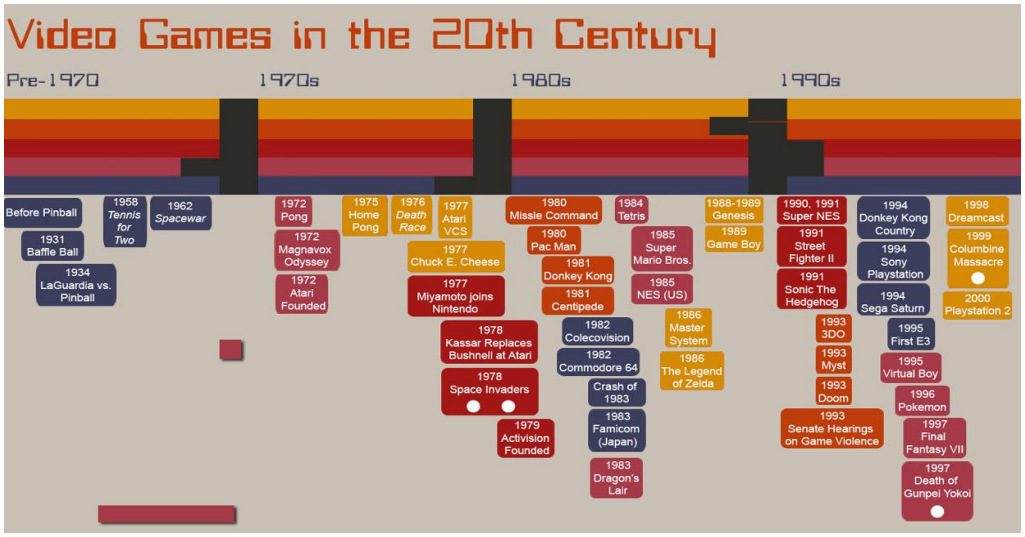
2.1 Videopelien kehitys

Videopelit saivat alkunsa rahalla toimivista peliautomaateista, joita säilytettiin halleissa. Näissä pelihalleissa nuoret ja lapset pelasivat taskut täynnä kolikoita. Näitä pelejä valmisti muun muassa Atari jo vuodesta 1970 lähtien. Vuonna 1972 Atari kehitti pelin nimeltä Pong (Kuva 1), jossa oli kaksi valkoista viivaa mailoina ja valkoinen neliö pallona, jota nämä kaksi mailaa löivät ruudulla edestakaisin, kunnes toinen mailoista ei osunutkaan palloon. Pong oli jo iso hitti tulessaan markkinoille Vuonna 1983 Pohjois-Amerikan videopeliteollisuus koki ison romahduksen lukuisten seikkojen takia, muun muassa ylikyllästetyt pelikonsolimarkkinat, tietokonepelien kilpailu ja huonolaatuisten pelien ylimainonta, kuten esimerkiksi Atarin kehittämä E.T-peli, joka pohjautui samanimiseen elokuvaan. Tätä peliä pidettään huonoimpana pelinä, jota koskaan tehty. Muutaman vuoden kestänyt romahdus johti usean kotikone- ja videopelikonsoliyhtiön konkurssiin. Vuonna 1985 videopelikotiteollisuus alkoi toipumaan, kun Nintendo Entertaiment System (NES), jota Japanissa kutsutiin Famicom nimellä tuli Yhdysvaltoihin. NES paransi muiden edellisten konsoleiden 8-bitin grafiikkaa, värejä, ääntä ja pelattavuutta. (History.com Editors, 2017.)



Kuva 1. Atarin kehittämä Pong peli (Sheehan, 2017).

Vuonna 1991 Nintendo julkaisi Super NES-konsolin Yhdysvalloissa ja sen hinta oli 249.95 dollaria. Vuonna 1995 Sony toi Playstation konsolin Yhdysvaltoihin ja myy sitä 299 dollarilla. Samaan aikaan Japanissa Nintendo julkaisee Nintendo 64-konsolin, joka julkaistiin Yhdysvalloissa 1996 vuonna. Playstation-konsolia pidettiin suosituimpana pelikonsolina, sillä sitä myytiin 20 miljoonaa kappaletta vuonna 1997.(Kudler,2017.) Tässä on 20-vuosisadan videopelit (Kuva 2)



Kuva 2. 20 vuosisadan videopelit (Rochelle ,2014)

2000-luvun alusta lähtien internetin kyvykkyydet ovat kehittyneet räjähdysmäisesti ja tietokoneprosessoriteknologia on parantunut niin nopealla tahdilla, että jokainen uusi joukko pelejä, grafiikoita ja konsoleita näyttää päihittävän edellisen sukupolven pelit mennen tullen. Teknologian, servereiden ja internetin hinnat ovat tippuneet niin alas, että salamannopea internet on nyt saatavilla tavanomaisille ihmisille ja 3.2 miljardia ihmistä ympäri maailmaa on pääsy internettiin. ESA Tietokone ja videopeliteollisuuden raportin mukaan 1.5 miljardia ihmistä pelaa videopelejä internetyhteydellä. Siitä asti, kun älypuhelimet ja sovelluskaupat astuivat markkinoille 2007 vuonna pelaaminen on käynyt läpi vielä toisen nopean evoluution, joka muutti ei ainoastaan sen miten ihmiset pelaavat pelejä, mutta toivat pelaamisen, myös yleiseen pop-kulttuuriin ennen näkemällä tavalla. Nopeat kehitykset mobiiliteknologiassa viimeisen vuosisadan aikana on luonut räjähdysmäisen kehityksen mobiili pelaamisessa (Chikhani. 2015.)

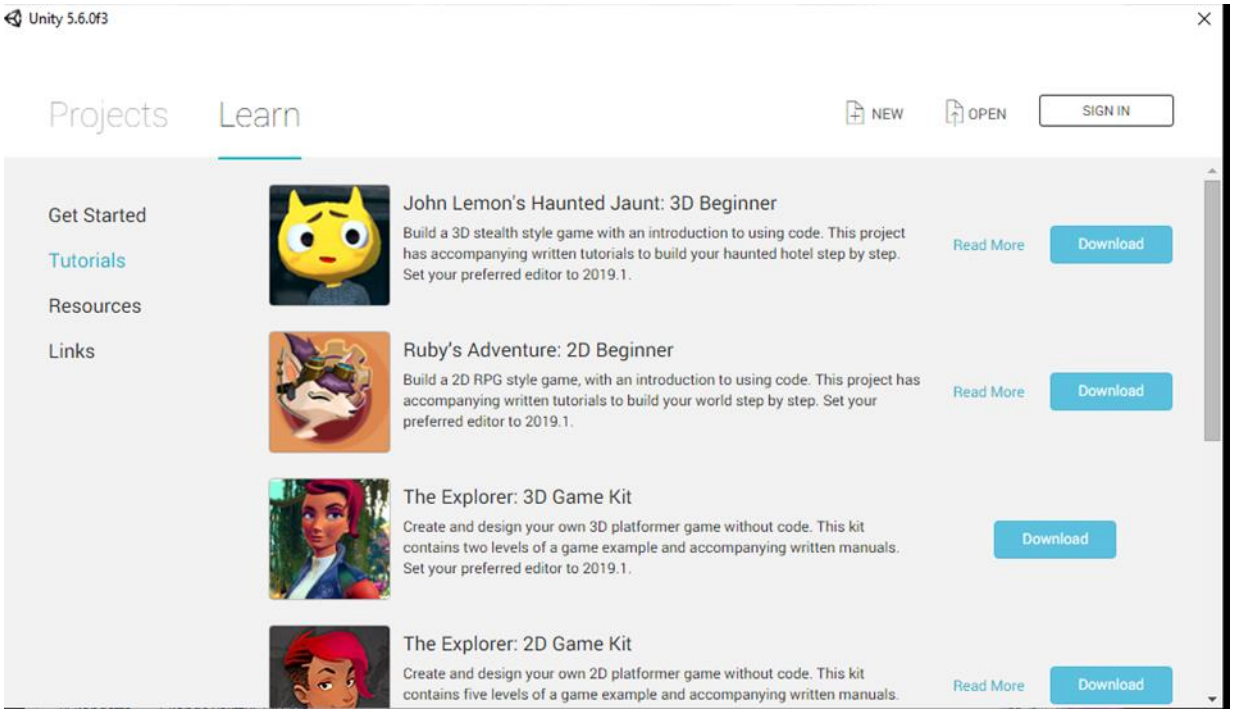
2.2 20-vuosisadan videopelit

- v.1931 ilmestyi Baffle Ball- Pinpal tyyppinen peli.
- v.1934 LaGuardia vastaan Pinpal
- v.1958 Tennis for Two peli
- v.1962 Spacewar-peli
- v.1972 Pong-peli, Magnavox Odyssey-peli ja Atari perustettiin
- v.1975 Pong-pelin kotiversio
- v.1976 Death Race-peli
- v.1977 Atari VCS-konsoli, Chuck E Cheese-peli ja Miyamoto liittyy Nintendo-yritykseen
- v.1978 Ray Kassarista tulee uusi Atarin presidentti syrjäyttäen Nolan Busnellin
- v.1978 Space Invaders- peli
- v.1979 Activision perustettiin
- v.1980 Missile Command ja Pacman
- v.1981 Donkey Kong ja Centipede
- v.1982 Colecovision ja Commondore 64
- v.1983 Famicon-konsoli japanissa ja Dragon’s Lair-peli
- v.1984 Tetris

- v.1985 Super Mario Bros-peli ja NES-konsoli USA
- v.1986 Master System-konsoli ja Legend Of Zelda-peli
- v.1989 Genesis ja Gameboy konsolit
- v.1990 Super NES konsoli
- v.1991 Street Fighter 2 ja Sonic The Hedhog pelit
- v.1993 3DO, Myst, Doom Ja Videopeli väkivalta kuulustelu Yhdysvaltojen Senaatissa
- v.1994 Donkey Kong Country, Sony Playstation ja Sega Saturn
- v.1995 Ensimmäinen E3 pelitapahtuma ja Virtual Boy- konsoli
- v.1997 Final Fantasy 7-peli ja Gunpei Yokoi Gameboyn kehittäjä kuolee auto
- onnettomuudessa
- v.1998 Dreamcast-konsoli SEGA-yhtiöltä
- v. 1999 kolumbialainen teurastus tuo uudelleen videopeli väkivalta puheeksi
- v.2000 Sony julkaisee Playstation 2-konsolin

(Kuva 2)

3 UNITY



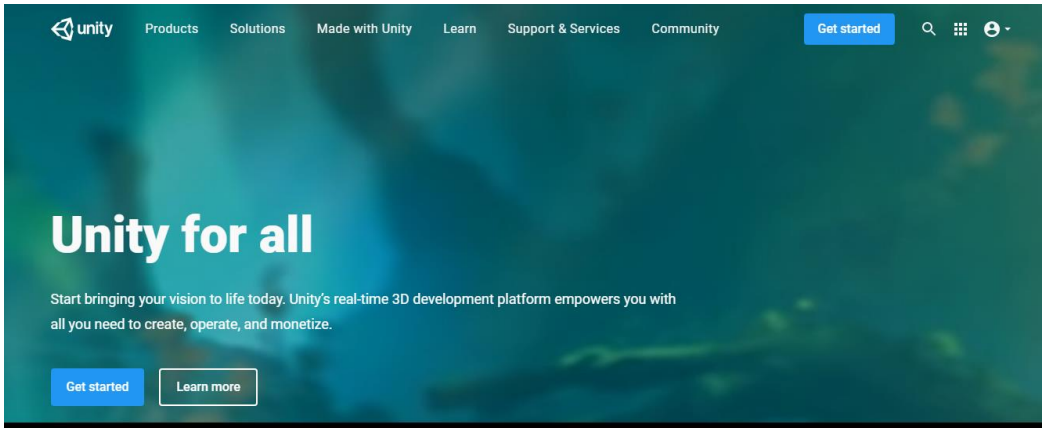
Kuva 3. Unity ohjelman Learn Osio

Unityn ohjelma näkymässä on learn-osio, jossa on erilaisia opetteluun tarkoitettuja projekteja(Kuva 3.)

Unity on Unity Technologiesin kehittämä pelimoottori. Se on tunnettu varsinkin indiekehittäjien (yksityinen pelinkehittäjä) ja harrastelijoiden suosimana ohjelmistona, vaikkakin sitä ovat käyttäneet myös jotkin suuremman budjetin pelistudiot. Unityn historia alkaa vuodesta 2002, kun kaksi pelikehityksestä kiinnostunutta tanskalaista tapasivat keskustelufoorumilla ja päätyivät yhdessä tekemään uudenlaisen pelimoottorin. Kolmen vuoden päästä, vuonna 2005, julkaistiin Unity 1.0. Suuremman yleisön huomion se saavutti kuitenkin vasta myöhemmin, kun selainpelien suosio räjähti ja Unity oli ensimmäisiä ohjelmistoja, joka tarjosi hyvät työkalut niiden kehittämiseen (Sinkkonen, 2017, 4)

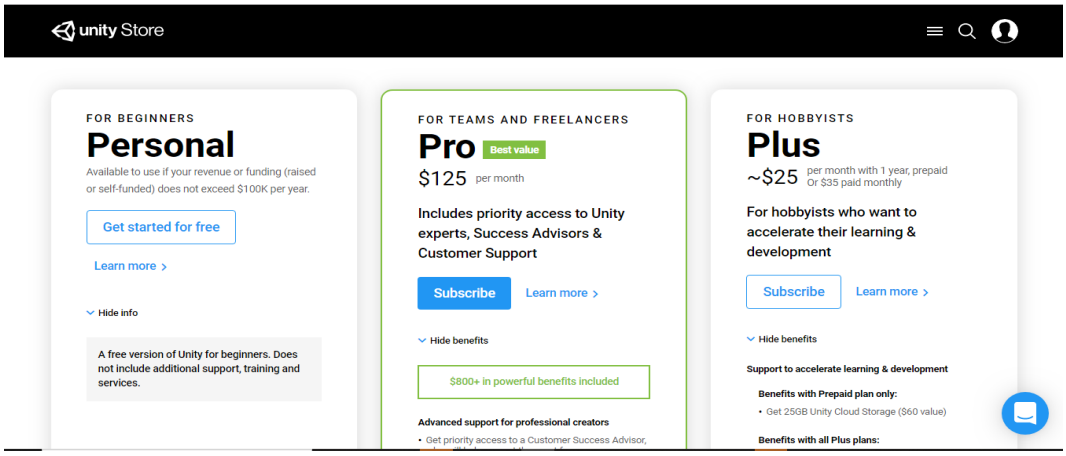
Unity on 2Dimension/3Dimension-moottori ja -viitekehys, jonka avulla voi suunnitella pelin tai sovelluksen kohtauksia 2D-, 2.5D- ja 3D-malleilla. Unitylla on kehitetty harjoitussimulaattoreita, hätäensiapu-sovelluksia, jotka soittavat hätänume-

roon automaattisesti kutsuen apua nopeasti ja muita bisnekseen keskittyviä sovelluksia, joilla oli tarve olla 2D/3D tilan kanssa tekemisessä. Unity mahdollistaa vuorovaikutuksen niiden sovellusten kanssa ei ainoastaan koodin kautta, mutta myös visuaalisten komponenttien kanssa ja viedä ne jokaiselle suurelle mobiiliympäristölle ja paljon muutakin ilmaiseksi. Unity-Pelimoottorista on myös olemassa maksullinen pro-versio mutta ilmaisversiolla pystyy tekemään yllättävän paljon.
(Tuliper,2014)



Kuva 4. Unity-pelimoottorin www-sivut (Unity3D, 2019)

Unity:n reaaliaikainen 3D-kehitysalusta antaa kaiken tarvittavat työkalut luomiseen, käyttämiseen ja rahan ansaitsemiseen (Kuva 4). Unity Personal on ilmaisversio, joka on suunnattu aloittelijoille. Unity Pro on maksullinen versio, joka on suunnattu tiimeille ja freelancereille. Unity plus on myös maksullinen versio suunnattu harrastelijoille.(Kuva 5)



Kuva 5. Unity-pelimoottorin 3 versiota Unity kauppa www-sivulla (Unity Store. 2019)

Unity on äärimäinen pelinkehitysalusta. Unity-ohjelmaa voi käyttää rakentamaan korkealaatuisia 3D- ja 2D-pelejä, jotka voit levittää mobiili-, tietokone-, VR/AR- ja konsoliympäristössä. Unity on myös rinnakkaisalusta pelimoottori, mitä ensisijaisesti käytettään videopelien ja simulaatioiden luomiseen. Ensin julkaistiin vain OS X:lle Apple-yhtiön maailmanlaajuisessa kehittäjien konferenssissa v.2005. Unity on sen jälkeen laajennettu 27 alustalle. Unity on yleinen tehtäväsäiliö, jota käytetään ihan minkä tyyppisessä Microsoft.NET-viitekehykseen perustuvassa sovelluksessa. Unity käyttää kahta skriptikieltä C# ja JavaScript, joka Unityssä tunnetaan myös nimellä UnityScript (Quora.2017)

Unity on pelimoottori, joka mahdollistaa luodon pelin ajamisen eri ympäristöissä. Se on myös sovellus, missä pelin näkyvät palat laitetaan yhteen graafisella esikatselulla ja ohjatulla pelausfunktiolla. Unity myös on koodieditori. Unityn suosio ei ole hehkutuksen tai markkinoin ansiota, vaan siihen on yksinkertaisia ja käytännöllisiä syitä. Unity tuli mahdolliseksi kun muisti ja resurssit lisääntyivät niin paljon, että muistin parantaminen menetti ensisijainen asemansa pelin kehitystyössä. Unity helpottaa pelin nopeaa kehitystä mahdollistamalla jatkuvaa julkiasemista ja nopeaa prototyypitystä, kun pääset jyvälle miten se toimii (Polsinelli,2013)

3.1 Mikä on pelimoottori?

Pelimoottori on sovellus, joka varustaa pelienkehittäjät tärkeillä ominaisuuksilla, joilla he voivat rakentaa pelejä nopeasti ja tehokkaasti. Pelimoottori toimivat pelinkehittämisen viitekehyksenä, joka tukee ja tuo yhteen useita keskeisiä alueita. 2D/3D taidetta ja esineitä voi tuoda toisista sovelluksista kuten Maya, 3s Max tai Photoshop ja koota nämä esineet erilaisiksi kohtauksiksi ja ympäristöiksi. Tähän lisäämällä valaistus, ääni, erikoistehosteet, fysiikka ja animaatio, interaktiivisuus ja pelauslogiikkaa voidaan näiden avulla muokata debugoida ja optimoida kohdesisältöä. (Unity. 2019)

Pelimoottori syntyy viiden eri komponentin pääohjelmasta, joka sisältää pelilogiikan, grafiikkamoottorin mitä voidaan käyttää 3D animoidun grafiikan luomiseen, Audiomoottorin, joka sisältää ääneen liittyviä algoritmeja,

fysiikkamoottori toteuttamaan fysiikan lait järjestelmän sisällä ja tekoälymoduuli suunniteltu ohjelmistoinsinöörien käytettäväksi. Työkalut ja ohjelmat tänä päivänä on tehneet pelin tekoprosessin aloituksen helpoksi. (Interesting Engineering, 2016)

3.2 Pelimoottoreiden vertailu

Seuraavassa vertaillaan Unitya tunnettuihin pelimoottoreihin Corona SDK, GameMaker Studio, Unreal Engine. Vertailun perusteella Unity on helppokäyttöisin ja monipuolisin erityisesti Indie-pelinkehittäjän kannalta.

Unity tarjoaa kaiken mitä tarvitaan pelin rakentamiseen yhdessä paketissa. Se on integroitu kehitysviitekehys, joka tarjoaa rikkaita ratkaisuja ja valmiita toimintoja pelin luomiseen. Sen avulla voi koota esineet ja taidetta kohtauksiin sekä ympäristöihin. Sen lisäksi voi lisätä ääntä, erikoistehosteita, valaistusta ja animaatiota. Unity on myös kaikkein suosituin pelimoottori maailmalla kattaen noin 45% markkinoista ja kosketaen 600 miljoonaa pelaajaa ympäri maapalloa.

Hyvät puolet:

- Unity tukee 25 eri alustaa ja parhaimmat alustat jolla tehdä rahaa on iOS, Android, Nintendo Switch, Steam, VR/AR jne.
- Voimakas grafiikka moottori joka optimoitu useammalle laiteelle. Johdonmukainen FPS saidoilla laiteilla.
- Tukee JavaScript ja C# ohjelmointikieliä
- Vedä ja tiputa rajapinta
- Iso pelikehittäjien yhteisö
- Reilu hinnoittelu
- Valtava voimavara kauppa, joissa on esirakennettuja, kiinnitä ja pelaa malleja
- Tukee 2D- ja 3D-grafiikkaa

Huonot Puolet:

- Pelimoottorin opettelu on hieman hankalaa

- Uudet pelinkehittäjät voivat häkeltyä pelimoottorin vaikeudesta
- Graafisesti intensiivisten pelien optimointi voi olla hankalaa
- Mobiili API:en integrointi ja Mobiili mainonta on hankalampaa kuin muilla pelimoottoreilla

Corona SDK on erinomainen 2D pelimoottori aloittelijoille, jotka haluavat kehittää järjestelmästä riippumattomia pelejä iOS, Android, Kindle, Windows ja muille käyttöjärjestelmille. Corona on täysin ilmainen.

Hyvät puolet:

- Lua skriptikieli joka on helppo aloittelija koodareille
- API:t ovat erittäin helppoja käyttää ja suoraviivaisia
- Helppo käyttää simulaattoria ilman että tarvitsee uudelleen rakentaa koodia joka muutoksen jälkeen
- Kolmannen osapuolen API:t ovat helppoja integroida, jos ne ovat Coronan kanssa kumppaneita

Huonot puolet:

- Todella vähän esirakennettuja malleja muihin pelimoottoreihin verrattuna
- Vanha pelimoottori, jolla pieni pelikehittäjien yhteisö
- Kohdistaa iOS ja Android laitteiden kaupallistamiseen. On olemassa työpöytä rakenne, joka ei ole kaupallistettavissa yhtä hyvin kuin mobiili
- Ei tue 3D:tä

GameMaker Studio on yksi vanhoista pelimoottoreista maailmalla. Mutta GameMaker Studio 2 saavuttua markkinoille, niin pelimoottorista tuli entistä vahvempi. Nopea, ystävällinen ja järjestelmäriippumaton. GameMaker antaa sinun rakentaa pelin yhdellä ainoalla koodausalustalla ja julkaista se ajettavaksi useammalla järjestelmällä.

Hyvät puolet:

- Yksinkertainen käyttää ja eikä liian ylivoimainen
- Erinoaminen aloittelijoille, joilla on rajalliset ohjelmointi tiedot

- Hauska ja helppo harrastelijoiden projekteille

Huonot puolet:

- Käyttää omaa ohjelmointi kieltä
- Rajoiteet kaupalistamisessa ei ole kolmanen osapuolen API:ja
- Ei tue 3D:tä

Unreal Engine on toinen nykypäivänä suosituista pelimoottoreista ja on tunnettu erityisesti ensimmäisen persoonan ampumispeleistä, mutta se on myös hyvä MMORPG-, RPG-, tappelu- ja hiiviskelypeleihin. Järjestelmäriippumaton Unreal on yhteensopiva todella monen käyttöjärjestelmän ja laitteen kanssa. On erittäin suuri todennäköisyys että olet pelannut Unrealilla tehtyä peliä.

Hyvät puolet:

- Sisältää Default-versiossa profiloinnin
- Graafiset kapasiteetit ovat edellä kilpailijoita
- Voimavara kaupassa on hyviä malleja

Huonot puolet:

- Käyttää C++ mikä vaatii enemmän ohjelmointi kokemusta kuin C# tai JavaScript’
- Epic Games Unrealin kehittäjä saa 5% rojalteja joka ansiostasi
- Rajatut kolmannen osapuolen APIt muihin pelimoottoreihin verrattuna
- Rakenteet eivät ole hyvin optimoituja pienille laitteille

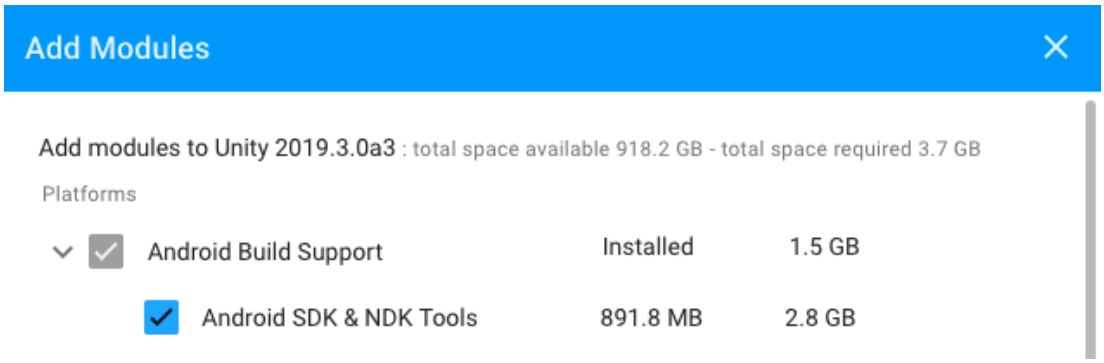
Loppujen lopuksi päätöksen tekoa helpottamaan sinun on tutkittava mihin alustoihin aiot kohdistua ja miten aiot kaupallistaa tuotteesi. (Scheutz, 2018)

Kuva 6: Pelimoottoreiden logot (Scheutz, 2018)

4 UNITY PELINKEHITYS KÄYTTÖJÄRJESTELMILLE

4.1 Android

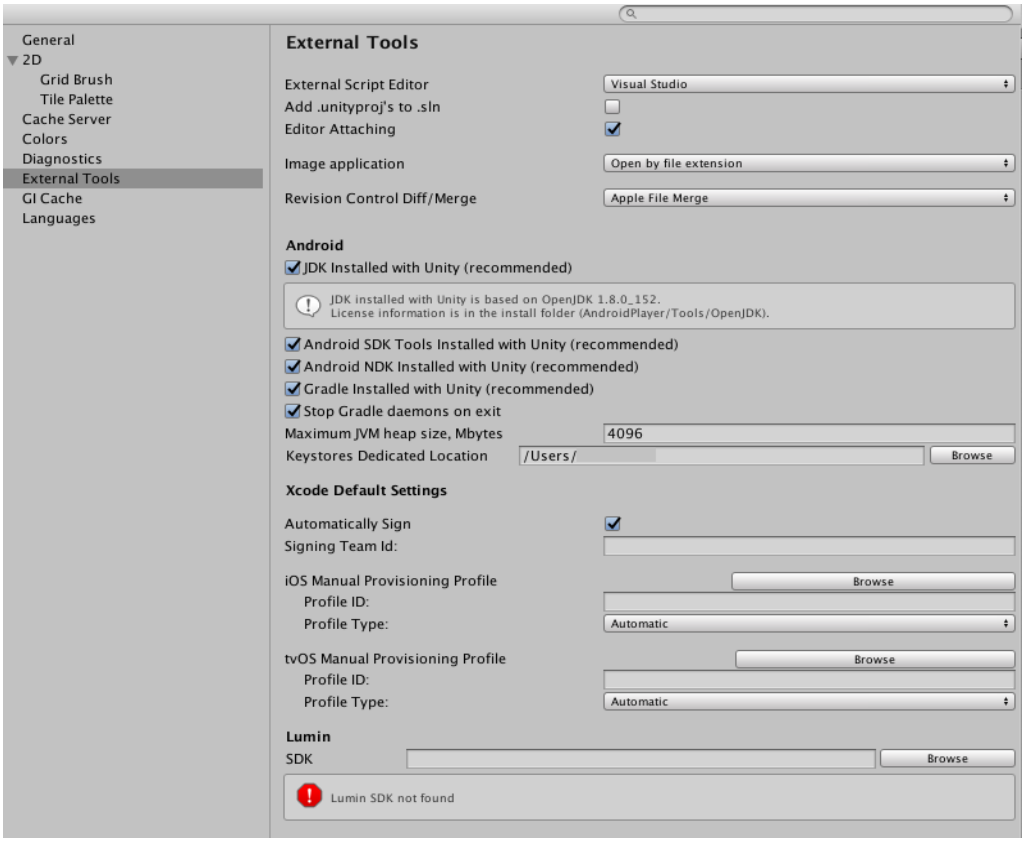
Android-käyttöjärjestelmän pelinkehitys alkaa asettamalla Androidin ympäristön, joka on aihe Unity-manuaalissa, sisältäen yksinkertaisen rajauksen tehtävistä, jotka on pakko suorittaa ennen koodin ajamista Android-emulaattorilla tai Android-laitteellasi. Unity tarjoaa komentosarjan API:eille, jotka antavat useita syöttödataa ja asetuksia Android-laitteesta. Android-ympäristön asennus alkaa asentamalla Unity Android Build Support- tukialustamoduuli(Kuva 7). Tämän lisäksi on asennettava Android **Software Development Kit (SDK)** ja **Native Development Kit (NDK)** ohjelmistot, joilla voit rakentaa ja ajaa minkä tahansa koodin Android laitteellasi. Oletuksena Unity asentaa **Java Development Kit,** joka pohjautuu OpenJDK-ohjelmistoon(Unity User Manual Android/Getting started with Android development, 2019)



Kuva 7 Unity Hubin kautta asennetaan Android Build Support ja Android SDK ja NDK työkalut(Unity User Manual Android, 2019)

Seuraavaksi on sallittava USB-testaus laitteessa, tämä onnistuu sallimalla Developer asetukset laitteella. Build numero on löydettävä laitteen Asetuksista. Sen paikka vaihtelee eri laiteilla mutta yleisillä Androidilla se sijaitsee **Asetukset>Tietoja puhelimesta> Build numero**. Tietyt tiedot laitteesta ja Android-version saat laitteen valmistajalta. Build numeron löydyttyä on napautettava sitä seitsemän kertaa ja seitsemännellä näpäytyksellä Developer asetukset avautuvat. Siirry asetuksiin > kehittäjä Asetukset tai jos tämä ei toimi, niin joillakin laitteilla polku on Asetukset > System > Developer Options. Tarkista **USB Debugging** valinta ruutu. Android siirtyy nyt de-

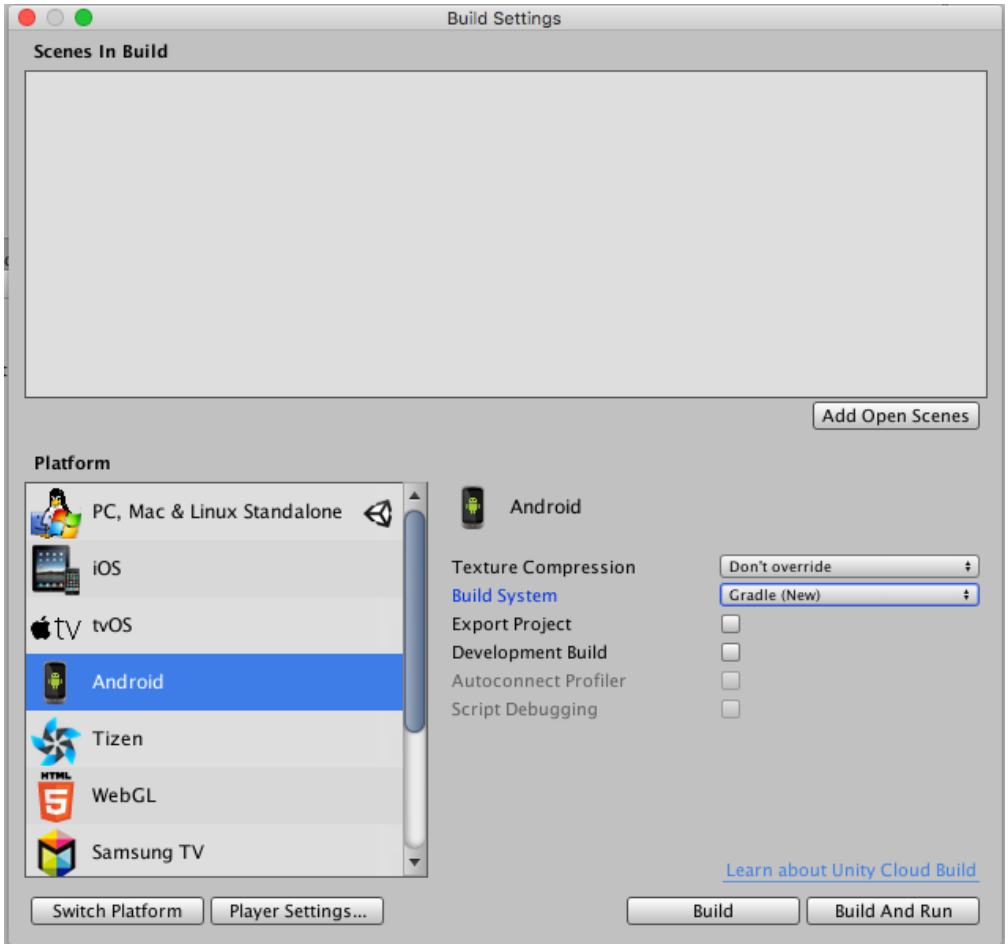
bug-tilaan, kun se on kytketty tieto koneeseen USB-yhteydellä. Liitä laite tietokoneeseen USB - kaapelilla, jos olet kehittämässä Windows-tietokonetta, sinun on ehkä asennettava laitekohtainen USB-ajuri. Katso lisätietoja laitteen valmistajan verkkosivustolta. Asennusprosessi eroaa Windowsille ja MacOS:lle, ja se selitetään yksityiskohtaisesti Android - kehittäjän verkkosivustolla. Lisätietoja Android-laitteen yhdistämisestä SDK:ssa löytyy Running Your App-osiossa Android Developer Documentation. Unity suosittelee, että käytät Unity Hub-työkalua Android SDK & NDK-työkalujen asentamiseen, jotta saat oikeat versiot ja määritykset. Unity asentaa Android SDK & NDK-työkalut vastaavasti SDK-ja NDK-kansioihin kohdassa Unity/Hub/Editor/[EditorVersion]/Editor/Data/PlaybackEngines/AndroidPlayer/.
Jos asennat manuaalisesti Android SDK & NDK-työkalut toiseen paikkaan, etkä halua kopioida asennusta, voit määrittää sijainnin Unity-asetukset-ikkunassa. Voit tehdä tämän siirtymällä kohtaan **Preferences** > **External tools** (Kuva 8) ja kirjoittamalla hakemisto polut SDK-ja NDK-kenttiin. (Unity User Manual Android, 2019)



Kuva 8. Preferences > External tools valikko Unityssa(Unity User Manual Android, 2019)

Unity käyttää Androidille Gradle-ohjelmistoa, joka automatisoi useita prosesseja. Tämä automaatio tarkoittaa, että monia yleisiä virheitä esiintyy vähemmän todennäköisesti. Erityisesti ykseydessä se vähentää menetelmän referenssien määrää DEX (Dalvik-tiedosto muoto) -tiedostoissa, mikä tarkoittaa, että olet vähemmän todennäköisesti törmännyt DEX-raja-ongelmiin. Joidenkin projektien konvertointi Gradlelle saattaa olla hankalaa Gradlen ja Androidin eroavaisuuksien takia. Voit joko rakentaa tuotospaketin (APK) käyttämällä Gradle-koontijärjestelmää yhtenäisyydessä tai viedä Gradle-projektin ja rakentaa sen ulkoiseen työkaluun (kuten Android Studio).

Gradlen rakentaminen Unityssa on kolmivaiheinen. Ensimmäinen vaihe on mennä Unity Editoriin ja avata **Build Settings**-ikkuna(Kuva 9), joka löytyy Unityn Menu-valikosta. Toinen vaihe on valita **Platform** listasta Android vaihtoehto. Viimeinen vaihe on asettaa **Build System-** valikossa **Gradle(New),** sitten klikataan **Build.** (Unity User Manual Android, 2019)



Kuva 9. Build Settings-ikkuna Unityssa (Unity User Manual Android, 2019)

Android Manifest on XML-tiedosto, joka sisältää tärkeitä metatietoja Android-sovelluksesta. Tämä sisältää paketin nimen, aktiviteettien nimet, pääaktiviteetin (sovelluksen aloitus kohta), kokoonpanot, Android-versio tuen, laitteiston ominaisuuksien tuen ja käyttö oikeudet. (Unity User Manual Android, 2019)

4.2 PlayStation/Xbox

Pelin kehittäminen konsoleille vaatii **Devolopment Kit** nimisen ohjelmiston. Tämä **Dev Kit** on pohjimmiltaan konsolin kehittäjä versio. Se on laite, joka toimii kuten todellinen konsoli, mutta sillä on enemmän ominaisuuksia virheen korjausta, käyttöönottoa ja sertifioimista varten (jotkut konsoli valmistaja yritykset vaativat, että olet "yhdistetty", jotta voit käyttää dekit-pakettia) jne. Useammat modernit pelimoottorit tukevat konsoleita. Mutta vaikka käytät yhtä näistä pelimoottoreista, niin tarvitset devkit-paketin ja kaikki oikeat sertifikaatit konsolisvalmistaja yritykseltä (Microsoft ja Sony) kääntääkseen niitä. (Arias, 2017)

Unity sisältää ominaisuuden nimeltä **Platform Dependent Compilation.** Tämä koostuu joistakin esikäsittelydirektiiveistä, jotka antavat sinun jakaa skriptisi jotka voit kääntää ja suorittaa osa koodista yksittäisesti yhdelle tuetulle alustalle. Voit ajaa tämän koodin Unity editorissa kääntää sen haluamalle alustalle ja testata sitä editorissa (Unity User Manual Platform dependent compilation, 2019)

UNITY_PS4	#define directive for running **PlayStation 4** code.
UNITY_XBOXONE	#define directive for executing **Xbox One** code.

Taulukko 1: PlayStation ja Xbox alusta #define direktiivit koodin ajaamiseen Unitylla (Unity User Manual Platform dependent compilation, 2019)

Scriptable Render Pipeline (SRP) on vaihtoehto Unity built-in-render pipeline ominasuudelle. SRP:N avulla voit hallita ja räätälöidä renderöintiä C#-komentosarjojen avulla. Näin voit joko hieman muokata tai täysin rakentaa ja mukauttaa render pipeline ominaisuuden tarpeidesi mukaan. Unityssa on kaksi SRP:tä **High Definition Render**

Pipeline (HDRP) ja **Lightweight Render Pipeline (LWRP).** Kukin SRP kohdistuu tiettyihin käyttötapaus skenaarioihin ja laitteiston tarpeisiin.
(UnityUserManual Scriptable Render Pipeline, 2019)

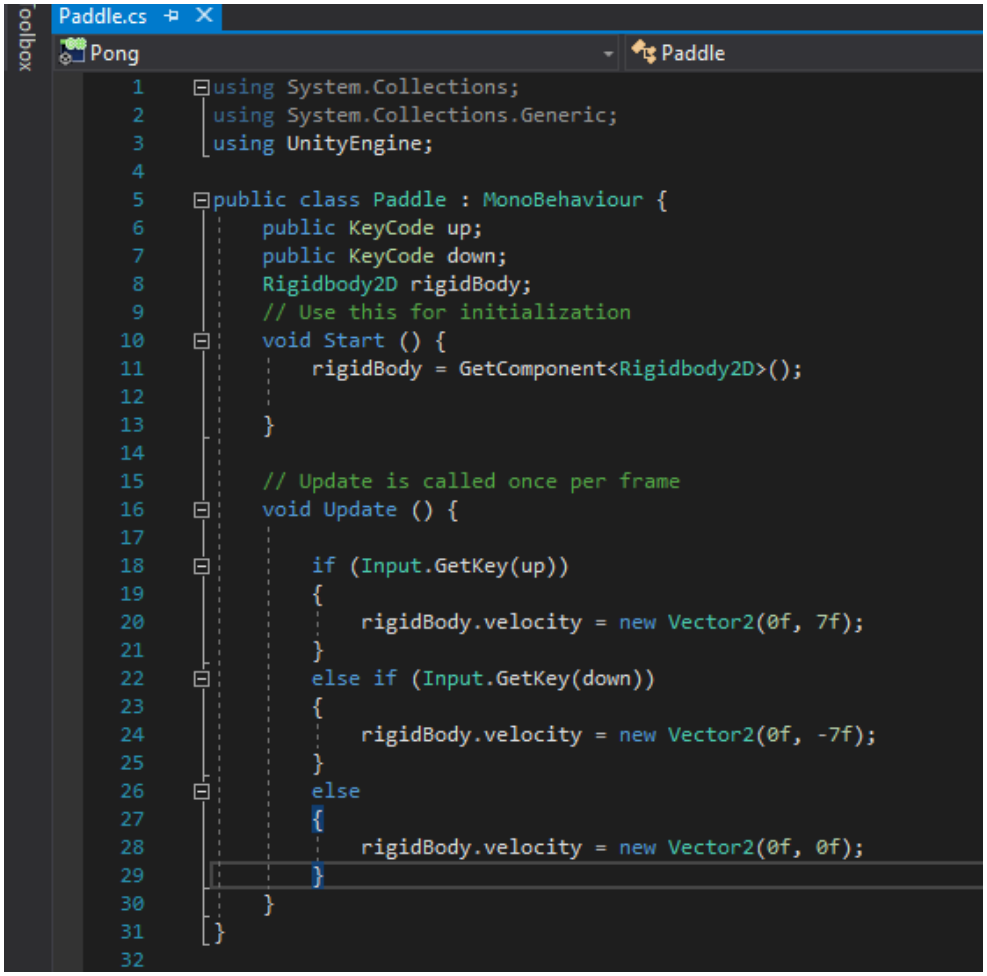
Platform	Minimum Unity version required
PlayStation 4	2018.2
Xbox One DirectX 11	2019.2
Xbox One DirectX 12	2019.1

Taulukko 2. Minimi Unity versio vaatimukset Playstation 4 ja Xbox One konsoleille (Unity User Manual Graphics Scriptable Render Pipline SRP Batcher, 2019)'

Näistä kahdesta konsolista Xbox One on helpoin päästä pelinkehitykseen mukaan. ID@Xbox -ohjelmaan haku on erittäin helppoa ja tämä todella hieno ohjelma sillä jos pääset osaksi sitä niin saat kaksi Xbox One Devkit-ohjelmistoa ja Unity lisenssin ilmaiseksi. Haettuasi tähän ohjelmaan sinä saat automaattisen sähköpostiviestin jota joudut odottamaan noin 1-2 viikkoa. Tämän jälkeen sinulle tulee toinen viesti sähköpostiin jossa sanotaan että olet liittynyt ID@Xbox -ohjelmaan. Mutta et vielä saa DevKit-ohjelmistoja, sillä on paljon muita pelinkehittäjiä, jotka odottavat samaa ohjelmistoa. Joudut siis jonottamaan ja odottamaan vuoroasi, mutta Microsoft on reilu ja se on yhteydessä pelinkehittäjiin kuukausittain. Tullaksesi Playstation pelinkehittäjäksi sinun täytyy liittyä Playstation Partners jäseneksi ja täyttää hakemus, mutta SONY on se joka asettaa vaatimukset. Kaksi ehtoa pitää täyttyä: ensimmäinen ehto on, että sinulla pitää olla liitetiedosto, joka todistaa, että olet todellakin rekisteröitynyt yritykseen ja toinen ehto on liitteenä oleva tiedosto, joka kuvaa peliä, jota haluat kehittää PlayStation-alustoille. Odotusaika siihen, että SONY vastaa viesteihin on 1-2 viikkoa, SONY on reilu, mitä tulee kehittäjäpaketteihin, jos sinulla on rahaa niin voit ostaa ne, mutta SONY lainaa myös niitä sinulle.(Deisling, 2015)

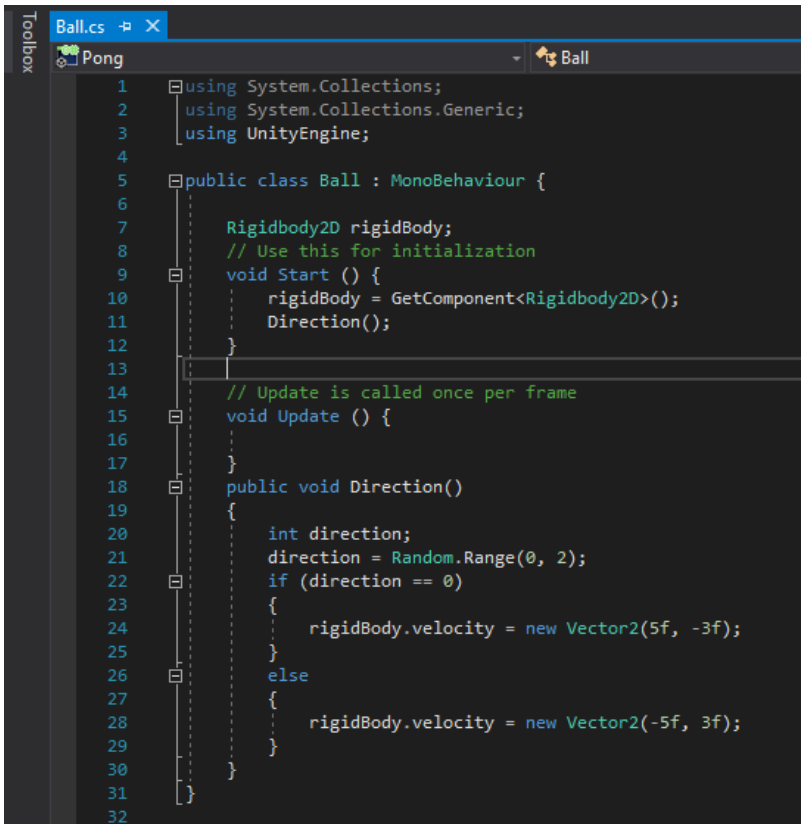
5 PONG PELIPROJEKTI UNITY-PELIMOOTTORILLA

Pelin, jonka luon Unity-pelimoottorilla, on yksinkertaistettu versio Atarin kehittämästä pelistä nimeltä Pong. Aloitin peliprojektini luomalla Unity-pelimoottorilla uuden 2D-peliprojektitiedoston nimeltä Pong. Tiedoston luomisen jälkeen aloin työstämään itse peliä Unity-pelimoottorissa. Peliprojektini ensimmäinen vaihe oli luoda Sprite-objekti nimeltä paddle ja sitten loin pelaajat lisäämällä BoxCollider2D-komponentin, Rigidbody2D-komponentin ja C# koodiskriptin nimeltä Paddle (Kuva 8). Nimesin pelaajat nimellä Player 1 ja Player 2.

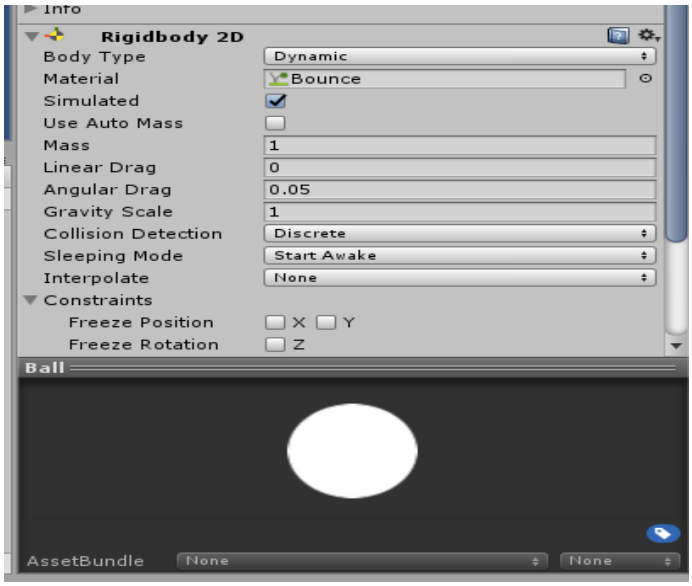


Kuva 8. Paddle C# koodiskripti Visual Studio näkymä luotu Unitylla

Pelaajien luomisen jälkeen loin toisen Sprite-objektin nimeltä Ball ja tästä objektista loin pallon pelaajille lisäämällä Ball-objektiin komponentit CircleCollider2D ja Rigidbody2D ja C# koodiskripti nimeltään Ball (Kuva 9). Pallon toimiakseen nimensä mukaisesti oli lisättävä Rigibody2D-komponentiin 2D fysikaalinen materiaali nimeltä Bounce (Kuva 10) tämä mahdollisti pallon pomppimisen.

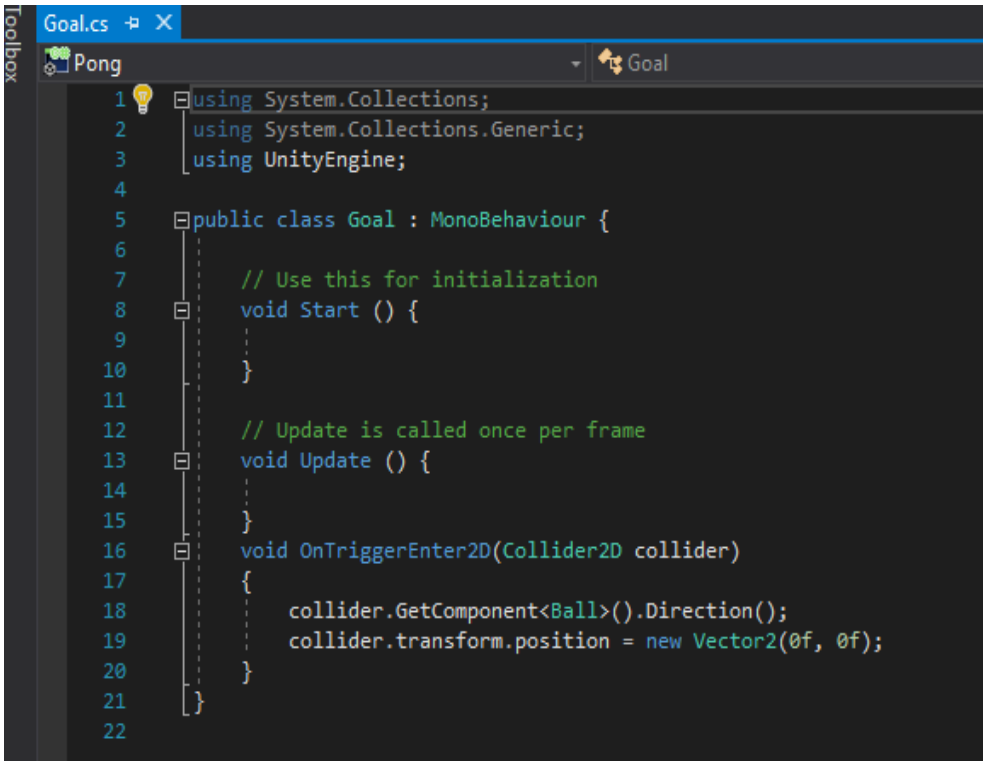


Kuva 9. Ball C# koodiskripti Visual Studio näkymässä luotu Unitylla.



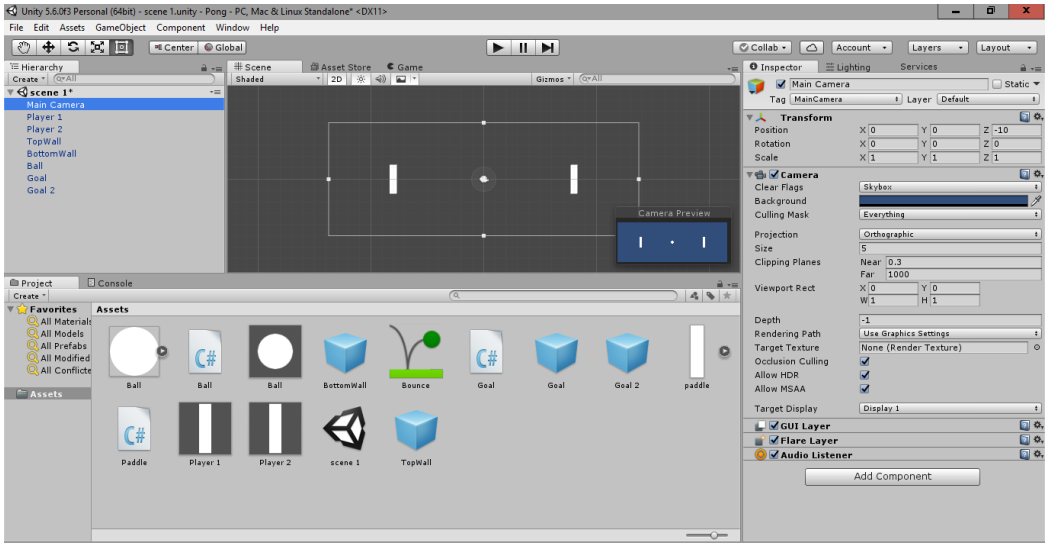
Kuva 10. Ball-objektin Rigidbody 2D valikko, jossa Material kohtaan lisätiin Bounce

Pallon luomisen jälkeen oli aika lisätä maalit, jotka tulivat pelaajien taakse. Ne loin Prefab-objekteista, jotka nimesin Goal 1 ja Goal 2. Toimiakseen nämä kaksi objektia tarvitsivat BoxCollider2D-komponentin ja C# Koodiskriptin nimeltä Goal (Kuva 11).



Kuva 11. Goal C# Koodiskripti Visual Studiossa luotu Unitylla.

Viimeiseksi loin Peliprojektiini ylä- ja alaseinät, jotta pallo ei menisi peli alueen ulkopuolelle. Nämä seinät tein myös Prefab objekteista, johon lisäsin BoxCollider2D-komponentin. Näin Peliprojektini oli valmis (Kuva 12) ja peliä pystyi nyt pelaamaan Unity-pelimoottorissa.



Kuva 12. Valmis peliprojekti Unity-pelimoottorissa

6 YHTEENVETO

Tein Pong-pelin Unitylla netin ohjeistuksen avulla ja käytin siihen C#-ohjelmointia kolmen skripitin verran, mikä mielestäni on hyvin vähän. Näissä Unitylla luoduilla C#-skripteissä koodia ei ole paljon ja tämä teki tästä peliprojektistani yksinkertaisen. Syy miksi valitsin Unity muiden Pelimoottoreiden joukosta on se, että olen opinnoissa opiskellut paljon C#-ohjelmointia, joten oli luontevaa valita Unity. Unity myös oli ilmainen ja sopiva yksin pelin kehittämiseen. Muutenkin Unity on tuttu ohjelmisto itselleni. Aion tulevaisuudessa tehdä vähän isomman ja haastavamman pelin. Tehdessäni tätä työtä huomasin, että pelien tekeminen Unityn kanssa oli suhteellisen helppoa loppujen lopuksi, kun osasi yhdistää oikeat komponentit ja koodiskriptit yhdeksi kokonaisuudeksi. Tein johtopäätöksen, että pelien kehittämien Unitylla kannattaa, sillä muihin pelimoottereihin verrattuna se on varsin nopea ja helppo omaksua. Unity soveltuu erityisesti itsenäiselle pelinkehittäjälle, jolla ei ole omaa tiimiä.

LÄHTEET

Deisling, 2015. How to Get on consoles As indie https://www.gamasutra.com/blogs/SlawaDeisling/20150223/237040/How_to_get_on_consoles_as_an_Indie.php

Scheutz, 2018 Comparing Game engines Viitattu: 24.9.2019 https://www.pubnub.com/blog/comparing-game-engines-unity-unreal-corona-gamemaker/´

Sinkonen S, 2017. Unity-pelimoottorin tarkastelu aloittelevan Indie-kehittäjän näkökulmasta. Tietotekniikan kandidaatintutkielma. Jyväskylän Yliopisto Viitattu 17.9.2019. https://jyx.jyu.fi/bitstream/handle/123456789/56591/1/URN%3ANBN%3Afi%3Ajyu-201801041046.pdf

Chikhani. 2015. The History Of Gaming: An Evolving Community Viitattu: 10.9.2019 https://techcrunch.com/2015/10/31/the-history-of-gaming-an-evolving-community/?gucounter=1&guce_referer_us=aHR0cHM6Ly93d3cuZ29vZ2xlLmZpLw&g

History.com Editors. 2017. Video Game History. Viitattu 23.5.2019 https://www.history.com/topics/inventions/history-of-video-games

Interesting Engineering, 2016. How Do Game Engines Work? Viitattu 12.9.2019 https://interestingengineering.com/how-game-engines-work

Kudler A. 2017. Timeline: Video Games. Viitattu 23.5.2019 https://www.infoplease.com/spot/timeline-video-games

Polsinelli,2013 Why is Unity so popular for videogame development? Viitattu 12.9.2019 https://designagame.eu/2013/12/unity-popular-videogame-development/

Quora, 2017. What is Unity Game Engine Viitattu 11.9.2019 https://www.quora.com/What-is-unity-game-engine

Arias, 2017 How Can I Become A game dev Viitattu 14.10.2019 https://www.quora.com/How-can-I-become-a-PlayStation-xbox-or-PC-game-developer

Ranta. P. 2010. Tietokonepelien lyhyt historia. Viitattu 23.5.2019 http://pranta.mbnet.fi/ptimeli.htm

Rochelle C. 2014. Ultimate History of Video Games Timeline. Viitattu 23.5.2019 https://www.thinglink.com/scene/444741687617519616

Sheehan G. 2017 Atari Celebrates Pong's 45th Anniversary With "Pong Day" & New Products. Viitattu 23.5.2019 https://www.bleedingcool.com/2017/11/29/atari-celebrates-pongs-45th-anniversary-with-pong-day-new-products/

Tuliper A. 2014 Unity: Developing Your First Game With Unity and C#. Viitattu 24.6.2019 https://msdn.microsoft.com/en-us/magazine/dn759441.aspx

Unity Store. 2019. Unity kauppa www-sivu. https://store.unity.com/

Unity. 2019. What is a game Engine https://unity3d.com/what-is-a-game-engine

Unity3D. 2019. Unity www-sivu Viitattu: 10.9.2019 https://unity.com/

Unity User Manual Android, 2019 Viitattu 10.10.2019 https://docs.unity3d.com/Manual/android.html

Unity User Manual Platform dependent compilation, 2019 Viitattu 14.10.2019 https://docs.unity3d.com/Manual/PlatformDependentCompilation.html

Unity User Manual Graphics Scriptable Render Pipline SRP Batcher, 2019 Viitattu 15.10.2019https://docs.unity3d.com/Manual/SRPBatcher.html

Wikihow. 2017 Create Pong in Unity Viitattu 13.8.2019 https://www.wikihow.com/Create-Pong-in-Unity-2017