Karelia-ammattikorkeakoulu
Tietojenkäsittely, Tradenomi (AMK)

Peliohjelmoijana 3D-tasohyppelypelin kehitysprojektissa

Matti Hänninen

Opinnäytetyö, marraskuu 2023

www.karelia.fi

Karelia AMMATTIKORKEAKOULU	**OPINNÄYTETYÖ** **Marraskuu 2023** **Tietojenkäsittelyn koulutus** Tikkarinne 9 80200 JOENSUU +358 13 260 600

Tekijä
Matti Hänninen

Nimeke

Peliohjelmoijana 3D-tasohyppelypelin kehitysprojektissa

Tiivistelmä

Tässä opinnäytetyössä seurataan peliohjelmoijan arkea ja työskentelyä 3D-tasohyppelypelin kehitysprojektissa. Päiväkirjatyöskentelyä seurattiin 65 työpäivän eli 13 työviikon ajan. Opinnäytetyön tavoitteina oli kirjoittajan osaamisen kehittäminen ja ammatillisen kehittymisen seuraaminen erilaisissa peliohjelmoinnin työtehtävissä. Tämän lisäksi tavoitteena oli tarkastelujakson aikana esiintyvien työtehtävien tehokas edistäminen.

Opinnäytetyössä käydään läpi työn aloitustilanne, joka sisältää kuvauksen kirjoittajan pääasiallisista työtehtävistä, niihin vaaditusta osaamisesta sekä kehittymisen arvioinnista lähtö- ja tavoitetilanteessa ja työntekoon vaikuttavista sidosryhmistä. Päiväkirjaosiossa tarkastellaan kirjoittajan työtehtäviä. Päiväkirjaosiossa keskityttiin peliohjelmoijan arjessa esiintyviin työtehtäviin, niissä ilmenneisiin haasteisiin sekä onnistumisiin ja erilaisten ratkaisumallien kehittämiseen. Työssä analysoitiin 3D-tasohyppelypelien perus-elementtejä ja niiden esiintymistä kehitysprojektissa. Työssä käsiteltiin pienen pelistudion ainoan peliohjelmoijan ajatuksia ja kyseisen roolin tuomaa vastuuta.

Opinnäytetyön kirjoittajan ammatillisessa osaamisessa havaittiin konkreettista kehitystä. Tämä ilmenee uusien työkalujen käyttöönotossa, uusien toimintamallien löytämisessä ja oman ammatillisen osaamisen tunnistamisessa.

Kieli suomi	Sivuja 121

Asiasanat
päiväkirja, peliala, peliohjelmointi, tasohyppelypelit

THESIS
November 2023
Degree Programme in Business Information Technology

Tikkarinne 9
80200 JOENSUU
FINLAND
+ 358 13 260 600

Author
Matti Hänninen

Title

Working as a Game Programmer in a Development Project of a 3D-Platformer Game

Abstract

This thesis follows the everyday work of a game programmer in a development project of a 3D-platformer game. The diary-based thesis covers 65 workdays, that is, 13 work weeks. The goals of the thesis were to enhance the author's skills and to follow his professional development in various game programming tasks. In addition to this, the goal was to promote the efficient progress of work tasks during the review period.

At first, the initial starting situation of the thesis is reviewed, which includes a description of the author's focal work tasks, the skills required, as well as an evaluation of professional competence in the initial phase, as well as the goals for professional development during the thesis and the stakeholders influencing the work. The diary section examined the author's work tasks. The diary section focused on the daily tasks of a game programmer, the challenges and successes and the development of different solution models. The basic elements of 3D platforming games and their appearance in the development project were analysed. The thoughts of the only game programmer of a small game studio were considered, as well as the responsibility of that role.

On the basis of the results of this thesis, it can be concluded that substantial development was identified in the author’s professional competence. This can be seen in the introduction of new tools, the discovery of new solution models and the recognition of one's own professional competence.

Language
Finnish

Pages 121

Keywords
diary, game sector, game programming, platform games

Sisältö

1 Johdanto 5
2 Aloitustilanne 6
2.1 Työtehtävät 6
2.2 Työtehtävissä vaadittava osaaminen ja osaamisen karttuminen 9
2.3 Työtehtävien tarkoitus ja tavoitteet 10
2.4 Opinnäytetyön tietoperusta ja keskeiset käsitteet 11
3 Kehittymisen arviointi 12
3.1 Oman osaamisen arviointi 12
3.2 Ammatillisen kehittymisen arviointi 12
3.3 Työtehtävissä kehittymisen arviointi 13
3.4 Tiedon ja tietojen kartuttaminen 13
3.5 Arvioinnin perustelu ja suhde tietoperustaan 14
4 Sidosryhmät ja vuorovaikutus työpaikalla 14
4.1 Sidosryhmien listaus 14
4.2 Sidosryhmien luokittelu 15
4.3 Kaavio sidosryhmistä 15
4.4 Sidosryhmien mielipiteet ja intressit 16
4.5 Vuorovaikutustilanteet työtovereiden kanssa 17
4.6 Vuorovaikutustilanteet sidosryhmien kanssa 18
4.7 Mahdolliset haasteet sidosryhmien kanssa 19
4.8 Arvioinnin perustelu ja suhde tietoperustaan 19
5 Päiväkirjatyöskentely 20
5.1 Seurantaviikko 1 (12.6.2023-16.6.2023) 20
5.2 Seurantaviikko 2 (19.6.2023-22.6.2023) 27
5.3 Seurantaviikko 3 (26.6.2023-30.6.2023) 33
5.4 Seurantaviikko 4 (3.7.2023-7.7.2023) 40
5.5 Seurantaviikko 5 (10.7.2023-14.7.2023) 48
5.6 Seurantaviikko 6 (7.8.2023-11.8.2023) 56
5.7 Seurantaviikko 7 (14.8.2023-18.8.2023) 63
5.8 Seurantaviikko 8 (21.8.2023-25.8.2023) 71
5.9 Seurantaviikko 9 (28.8.2023-1.9.2023) 77
5.10 Seurantaviikko 10 (4.9.2023-8.9.2023) 82
5.11 Seurantaviikko 11 (11.9.2023-15.9.2023) 89
5.12 Seurantaviikko 12 (18.9.2023-22.9.2023) 93
5.13 Seurantaviikko 13 (25.9.2023-29.9.2023 + 2.10.2023-4.10.2023) 99
6 Pohdinta 109
6.1 Työtehtävistä suoriutuminen, osaamisen kehittyminen ja arviointi 109
6.2 Tasohyppelypelien peruselementtejä ja subjektiivinen vaikeustaso 114
6.3 Peliohjelmoijan vastuu 116
6.4 Loppusanat 118
Lähteet 119

1 Johdanto

Tässä päiväkirjamuotoisessa opinnäytetyössä seurataan peliohjelmoijan arkea 3D-tasohyppelypelin kehitysprojektissa. Päiväkirjatyöskentely kattaa 13 viikon mittaisen tarkastelujakson. Päiväkirjan ensimmäinen merkintä on 12.6.2023 viikolla 24, ja viimeinen 4.10.2023 viikolla 40. Opinnäytetyössä käydään läpi työn aloitustilanne, jossa esitellään tekijän pääasialliset työtehtävät ja niihin vaadittu osaaminen, tarkoitus ja tavoitteet. Aloitustilanteessa esitellään myös työn tietoperusta ja keskeiset käsitteet, arvioidaan osaamisen lähtö- ja tavoitetasoja sekä käydään läpi erilaiset työskentelyyn vaikuttavat sidosryhmät. Arvioinnin suhdetta tietoperustaan perustellaan työn aloitusvaiheessa. Aloitustilanteen kuvaamisen jälkeen siirrytään 13 viikkoa kestävään päiväkirjatyöskentelyyn. Tarkastelujakson aikana seurataan tekijän jokapäiväisiä työtehtäviä ja niiden etenemistä. Päiväraporteissa kuvataan lyhyesti päivän työtehtävät ja niiden parissa ilmenneitä asioita sekä arvioidaan osaamistavoitteiden täyttymistä. Viikkoraporteissa käydään läpi viikon aikana esiintyneiden työtehtävien teemoja ja perustellaan käytettyjä toimintamalleja tietoperustaan verraten.

Opinnäytetyön tavoitteena on oman ammatillisen osaamisen kehittäminen, ja kehittymisen seuraaminen erilaisissa peliohjelmoinnin työtehtävissä. Tämän lisäksi tavoitteena on tarkastelujakson aikana esiintyvien työtehtävien tehokas edistäminen, ja niissä käytettyjen toimintamallien arviointi. Työ tarjoaa myös lukijalleen mahdollisuuden tutustua peliohjelmoijan arkeen. Opinnäytetyössä analysoidaan erilaisia 3D-tasohyppelypelien peruselementtejä ja tarkastellaan niiden esiintymistä kehitettävässä projektissa. Tämän lisäksi työssä avataan pienen pelistudion ainoan peliohjelmoijan ajatuksia ja kyseisen roolin mukanaan tuomaa vastuuta. Tämä opinnäytetyö kertoo, millaista on työskennellä peliohjelmoijana 3D-tasohyppelypelin kehitysprojektissa.

2 Aloitustilanne

2.1 Työtehtävät

Työnkuvani on "peliohjelmoija", joten työtehtäviini kuuluu luonnollisesti peliohjelmointia, mutta myös paljon design puolen suunnittelua. Pääasiallisena työkaluna toimii Unreal Engine 4 -pelimoottori, ja työstämme 3D-tasohyppely tyylistä peliä. Työtehtäviini kuuluu oikeastaan kaikki, mitä kyseisen pelityylin peliltä voi odottaa, joita olen listannut alla olevaan luetteloon.

-Versionhallinta

Huolehdin versionhallinnasta omalta osaltani (kuten muutkin työntekijämme) Perforce ohjelman kautta. Unreal Engine -pelimoottorin puolella merkkaan tiedostot, joita työstän Check Out -merkinnällä, eli lukitsen ne itselleni niiden työstämisen ajaksi. Tiedostot pysyvät lukittuna minulle, kunnes saan ne valmiiksi. Tehtyäni tarvittavat muutokset (tai vaihtoehtoisesti luotuani täysin uusia ominaisuuksia tai asset-objekteja peliin), teen Perfoce ohjelman kautta Submit komennon, eli pusken työstämäni tiedostot etärepositorioon muille saatavaksi. Perforce ohjelman avulla saan myös muiden tekemät muutokset itselleni. Versionhallintaa tulee suoritettua koko ajan projektin kanssa.

-Oman työn dokumentointi

Dokumentoin tekemäni työtehtävät Trello- ja, Miro- palveluihin sekä Perforce ohjelman Submit-käskyjen kommenttikenttiin. Trello palvelun Task-kortteihin kirjoitan kuvauksen tekemistäni muutoksista, tai liitän siihen linkin Miro palveluun. Dokumentointi Trello kortilla voi olla työtehtävän mukaan vain yksi lyhyt lause (esimerkiksi nopea bugikorjaus tai muuttujan arvojen vaihto), tai suurempi kokonaisuus, jossa kerrotaan kohta kohdalta tehdyt muokkaukset/lisäykset. Miro palveluun kirjoitan erilaisia lyhyitä tutoriaaleja erilaisista mekaniikoista, ja niiden toiminnallisuuksista. Perforce ohjelman Submit kommenttikenttään kirjoitan aina myös mahdollisimman tarkan

kuvauksen tekemistäni muutoksista, ja uusista ominaisuuksista. Unreal Engine -pelimoottorin puolella pyrin myös kommentoimaan kirjoittamaani koodia mahdollisimman paljon ja tarkasti. Dokumentaatiota tulee suoritettua paljon, koska se on äärimmäisen tärkeä osa-alue työssäni.

-Peliohjelmointi: blueprint ohjelmointi

Työtehtäväviini Unreal Engine -pelimoottorin blueprint ohjelmoinnin kanssa kuuluu oikeastaan kaikki, mitä 3D-tasohyppely tyylisen pelityylin peliltä voi odottaa; esimerkiksi hahmon liikkumista ja kykyjä, animaatioiden kanssa työskentelyä, tekoälyn kanssa työskentelyä, dialogijärjestelmiä, UI-elementtien luontia, ja mitä vain muuta aiheeseen liittyvää. Vietän blueprint ohjelmoinnin parissa suurimman osan työpäivistäni.

-Peliohjelmointi: animaatioblueprintit

Animaatioblueprintit hallitsevat pelihahmojen animaatioita. Animaatioblueprinteissä luon hahmoille animaatiotilakoneen, jossa voidaan siirtyä eri tiloista toiseen. Esimerkkejä ovat esimerkiksi kävely, hyppy, hyökkäys ja niin edelleen. Jokaisen animaatio tilan välille luon siirtymäehdot, joiden avulla valitaan, mitä animaatiota halutulla hetkellä pyöritetään. Joihinkin animaatioihin liitän kiinni myös ääniä ja erilaisia partikkeliefektejä.

-Peliohjelmointi: dialogijärjestelmät

Dialogijärjestelmät luodaan toimimaan yhdessä UI-elementtien kanssa. Luon Data Table -objektin, johon kenttäsuunnittelija voi kirjoittaa dialogeja halutuille pelihahmoille. Data Table -objektissa määritetään, kuka puhuu, missä, koska, ja pitääkö keskustelun avaamiseen olla suoritettuna jokin esiehto. UI-elementtien koodin puolella nämä tiedot, sekä kirjoitetut dialogilauseet parsitaan yhteen tietyillä avainsanoilla, joiden avulla oikeat vastinkappaleet löytyvät. Pelistämme löytyy jo varsin hyvä dialogijärjestelmä, ja tähän mennessä olen tehnyt siihen hieman muutoksia ja parannuksia. Dialogijärjestelmä vaatii kuitenkin lisäyksiä ja muokkauksia vielä tulevaisuudessa.

-Peliohjelmointi: UI-elementit

User Interface -elementit, ovat ruudulle ilmestyviä tekstejä, kuvia, ikkunoita tai jotain muita visuaalisia elementtejä. UI-elementtien kanssa työskennellessä luon ensin visuaalisen ulkoasun, esimerkiksi pelaajan terveysmittarin. Visuaalisen ulkoasun luotua siirryn UI-elementin koodin puolelle, ja liitän toiminnallisuudet visuaalisiin elementteihin, esimerkiksi terveysmittarin tapauksessa vähennän pelaajan terveyspisteitä halutun määrän tämän ottaessa osuman viholliselta. UI-elementit vaativat paljon välillä paljonkin koodia, jotta visuaaliset elementit pelaavat yhdessä haluttujen toimintojen kanssa.

-Peliohjelmointi: tekoäly

Työskennellessäni pelihahmojen tekoälyn kanssa, pääsääntöisiä kohteita on muutama: AI-Controller -luokka, johon luon logiikkaa tekoälyhahmon havainnoista ympärillä olevasta maailmasta, esimerkiksi tekoälyn näköyhteys pelaajaan. Behavior Tree -käytöspuu, jolla ohjaan tekoälyhahmon käytöstä puumaisella rakenteella eri suuntiin. Käytöspuut vaativat todella paljon ohjelmointilogiikkaa, jotta hahmojen toiminta tuntuu luontevalta, ja säilyttää pelaajan immersion pelimaailman kanssa. AI-Controller -luokan ja Behavior Tree -käytöspuun lisäksi yhdistän niiden logiikkaa pelihahmojen omien blueprintien kanssa.

-Peliohjelmointi: bugien löytäminen ja korjaaminen

Bugien havaitseminen ja korjaaminen on tärkeää työtä, koska ne voivat pahimmillaan rikkoa tai lukita koko pelin. Bugit ovat hyvin yksilöllisiä, ja niihin kuluvaa työaikaa on vaikea arvioida. Bugin mukaan, saattaa joskus riittää muutaman koodirivin siirtäminen eri paikkaan, tai mahdollisesti täysin uuden logiikan luominen ja tekeminen bugin poistamiseksi. Etsin ja korjaan itse havaitsemiani bugeja koko ajan työskennellessäni, ja korjaan myös muiden jälkikäteen havaitsemia bugeja.

-Pelidesign: Asioiden ja mekaniikkojen ideointi
Suoritan työssäni myös paljon design puolen asioita. Tämä pitää sisällään esimerkiksi mekaniikkojen ja asioiden suunnittelua sekä ideointia. Mekaniikkoja työstäessä on tärkeää kokeilla ja iteroida erilaisia vaihtoehtoja, ja työhöni kuuluu myös ideoiden esittäminen ja vaihtoehtojen tarjoaminen. Suunnittelemme myös välillä yhdessä tiimin kanssa pelimekaniikkoja, ja kenttien kulkua.

2.2 Työtehtävissä vaadittava osaaminen ja osaamisen karttuminen

Edellä mainitut työtehtävät vaativat perustasoa vaativampaa osaamista yleisistä ohjelmointilogiikoista, vahvaa ongelmanratkaisukykyä sekä luovaa ajattelua ja ymmärrystä pelikehityksen periaatteista. Ne vaativat ymmärrystä versionhallinnan merkityksestä projekteissa ja sen sujuvaa hallitsemista. Ne vaativat kykyä artikuloida ja dokumentoida koodia, design puolta ja prosesseja sekä sujuvaa tiedon välittämistä muille tiimin jäsenille. Työtehtävä vaativat osaamista Unreal Engine -pelimoottorin käytöstä, ja sen eri osa-alueiden ymmärtämisestä.

Työtehtävät kartuttavat mm. seuraavanlaista ammatillista osaamista: Monipuolisen ammatillisen osaamisen pelikehitykseen keskittyen erityisesti ohjelmointiin, design puoleen ja versionhallintaan. Ne tarjoavat tekijälleen taitoja ohjelmoida erilaisia pelimekaniikkoja, dokumentoida omaa työnjälkeä laadukkaasti, hallita koodikantaa tehokkaasti, suunnitella ja toteuttaa interaktiivisia kokemuksia sekä etsiä ja korjata ongelmia. Työtehtävien monipuolisuuden vuoksi niiden kautta on mahdollisuus saavuttaa todella laajan skaalan osaamisen taso pelikehittäjänä.

2.3 Työtehtävien tarkoitus ja tavoitteet

Pelin visio: Peliprojektin kokonaisvision ja tavoitteiden ymmärtäminen on tärkeää. Tämä sisältää kohdeyleisön, pelin tyylin tai teeman, ja suunnitellun pelaajakokemuksen tuntemisen. Nämä tiedot auttavat sovittamaan työtehtävät projektitavoitteiden kanssa.

Projektin laajuus ja vaatimukset: Projektin laajuuden ja vaatimusten selkeä ymmärtäminen on välttämätöntä. Tämä sisältää pelin haluttujen ominaisuuksien, mekaniikkojen ja toimintojen tuntemisen. Se auttaa priorisoimaan tehtäviä, asettamaan odotuksia ja varmistamaan, että työ on linjassa projektin tavoitteiden kanssa.

Suunnittelu ja tekninen dokumentaatio: Pääsy suunnittelu- ja teknisiin dokumentaatioihin tarjoaa yksityiskohtaista tietoa pelin mekaniikasta, järjestelmistä ja työnkuluista. Dokumentaatioissa esitellään suunnitellut toiminnot, käyttäytyminen ja toteutusohjeet, joiden avulla työntekijät voivat sisäistää työtehtäviensä erityiset tavoitteet ja vaatimukset.

Yhteistyö ja viestintä: Tehokas viestintä tiimin jäsenten ja sidosryhmien kanssa on välttämätöntä. Säännölliset tapaamiset, keskustelut ja palauteistunnot auttavat selventämään työtehtävien tarkoitusta ja tavoitteita, jolloin yksilöt voivat kohdistaa tekemisensä projektin laajempiin tavoitteisiin.

Käyttäjäpalaute ja testaus: Käyttäjien palautteen kerääminen ja testausistuntojen suorittaminen voivat antaa käsityksen aiotusta pelaajakokemuksesta. Pelaajien vuorovaikutuksen pelin ja odotusten ymmärtäminen voi auttaa sisäistämään työtehtävien tarkoituksen ja ohjaamaan pelin ohjelmointiin, suunnitteluun ja projektin yleisiin tavoitteisiin liittyviä päätöksiä.

2.4 Opinnäytetyön tietoperusta ja keskeiset käsitteet

Tietoperustana opinnäytetyössä hyödynnetään teosta: Game Design Deep Dive: Platformers, jonka on kirjoittanut Joshua Bycer (2019). Teoksessa käsitellään tasohyppelypeleissä esiintyviä elementtejä, ja toimii erinomaisena lisänä asioiden analysoimisessa.

Bycerin (2019) teoksen lisäksi tietoperustana toimii Epic Games Unreal Engine Documentation -dokumentaatiot (2023). Dokumentaatiot pitävät sisällään viralliset tiedot ja kuvaukset Unreal Engine -pelimoottorin ominaisuuksista, ja niiden toiminnasta.

Päiväkirjatyöskentelyn aikana sekä opinnäytetyön pohdintaosiossa esiintyvät keskeiset käsitteet ovat työtehtävien tehokas edistyminen, sekä osaamisen kehittyminen peliohjelmoijana. Näitä teemoja on tärkeää seurata siksi, koska ne kulkevat käsi kädessä peliohjelmoinnin asiantuntijaksi kehittymisen ja projektin edistymisen kanssa. Yrityksen tällä hetkellä ainoana ohjelmoijana minulla on paljon vastuuta, ja työtehtäviä peliohjelmoinnin eri osa-alueilta. Työtehtävien tehokas, itsenäinen ja omatoiminen eteneminen on siis koko projektin kannalta erityisen tärkeää.

Lisäksi tarkasteltavana käsitteenä on asioiden tarkastelu pienen pelistudion näkökulmasta, eli mitä kaikkea ohjelmoijan vastuulla on, mitä tulee huomioida ja mitä toteuttaa eri vaiheissa. Näiden asioiden huomioiminen tuo mielenkiintoista näkökulmaa edellä mainittuihin teemoihin.

Näiden käsitteiden lisäksi työskentelyn aikana tutkitaan 3D-tasohyppelypelien peruselementtejä, ja miten suositut kyseisen pelityypin pelit ovat toteuttaneet asioita, millaisia elementtejä niistä löytyy, ja miten niitä on toteutettu meidän pelissämme. Valitsemani Joshua Bycerin (2019) teos tulee toimimaan apuvälineenä näissä käsitteissä.

3 Kehittymisen arviointi

3.1 Oman osaamisen arviointi

Opinnäytetyön osaamisen kuvaamiseen ja arviointiin on annettu kolme eri tasoa: aloitteleva toimija, taitava suoriutuja, ja kokenut asiantuntija. (Karelia-ammattikorkeakoulu 2021, 2.) Työtehtävien osaamisvaatimukset työpaikallani ovat kaikki Kokenut asiantuntija -tasoisia työtehtäviä. Koen tämänhetkisen osaamiseni seuraavanlaiseksi:

-Versionhallinta – Taitava suoriutuja
-Oman työn dokumentointi – Taitava suoriutuja
-Peliohjelmointi: blueprint ohjelmointi – Taitava suoriutuja
-Peliohjelmointi: animaatioblueprintit – Taitava suoriutuja
-Peliohjelmointi: dialogijärjestelmät - Taitava suoriutuja
-Peliohjelmointi: UI-elementit - Taitava suoriutuja
-Peliohjelmointi: tekoäly - Taitava suoriutuja
-Peliohjelmointi: bugien löytäminen ja korjaaminen - Taitava suoriutuja
-Pelidesign: Asioiden ja mekaniikkojen ideointi - Taitava suoriutuja

Työskentelin jo harjoitteluni aikana paljon itsenäisesti, jonka jälkeen olen ollut palkkatöissä jo yli puolen vuoden ajan. Tänä aikana olen kartuttanut kokemustani valtavasti. Tästä johtuen koen olevani jokaisella osa-alueella jo Taitava suoriutuja -tasolla.

3.2 Ammatillisen kehittymisen arviointi

Koen olevani varsin pitkällä ammatillisen osaamiseni kanssa suhteessa työpaikan osaamistavoitteisiin. Muutamalla osa-alueella, kuten blueprint ohjelmoinnissa olen jo monen asian suhteen lähes Kokenut asiantuntija -tasolla, mutta koen kuitenkin tarvitsevani vielä hieman tietoja ja taitoja voidakseni sanoa yltäväni täysin kyseiselle tasolle. Uskon että tulevan 13-viikon aikana tulen

kasvattamaan osaamistani melkoisesti, toivon mukaan saamaan kaikki, tai ainakin suurimman osan tasolle Kokenut asiantuntija.

3.3 Työtehtävissä kehittymisen arviointi

Pystyn työskentelemään omatoimisesti ja tuottamaan laadukasta työnjälkeä. Ymmärrän työtehtävieni tarkoituksen, ja osaan tehdä tarvittavia toimenpiteitä tavoitteiden saavuttamiseen, ja työtehtävien toteuttamiseen. Olen selvinnyt kaikista minulle tähän mennessä osoitetuista työtehtävistä mallikkaasti. Osaan myös soveltaa osaamistani, luomalla uusia ja omia ratkaisuja ja toteutuksia. Olen myös kehittänyt monia jo olemassa olevia ominaisuuksia, ja tehnyt niihin tarvittavia parannuksia. Olen myös ohjeistanut jo kahta työharjoittelijaa. Kuten aiemmin mainitsin, koen olevani monen asian suhteen jo lähes Kokenut asiantuntija -tasolla, ja hyvin vahvasti vähintään Taitava suoriutuja -tasolla.

3.4 Tiedon ja tietojen kartuttaminen

Kehitys työssäni jatkuvasti, ja opin joka viikko uusia asioita. Peliohjelmoinnissa on todella paljon erilaisia osa-alueita, varsinkin kun olen tällä hetkellä ainoa ohjelmoija työpaikallani. Olen vasta hiljattain päässyt työstämään pelihahmojen tekoälyä, joten se on yksi osa-alue, jolla koen tarvitsevani vielä lisää kokemusta. Onneksi kyseiselle osa-alueelle on tulossa lähiaikoina lisää työtehtäviä, joten niiden avulla voin kehittää taitojani tekoälyn kanssa. Koen myös tarvitsevani lisää kokemusta animaatioblueprintien kanssa työskentelyyn. Peliimme on tulossa melko lailla lisää hahmoja, jotka tarvitsevat omat animaatioblueprintinsä, joten niiden kanssa työskennellessä taitoni karttuvat varmasti. Yleistä blueprint ohjelmoinnin osaamista, asioiden muuntelua ja parantelua minulle tulee vastaan joka viikko. Uskon voivani myös hyödyntää Joshua Bycerin (2019) teosta tasohyppelypelien elementeistä tietojeni kartuttamiseksi, ja toimivien ratkaisujen kehittämiseksi. Näiden tietojen ja taitojen avulla kehityn mukavasti kohti osaamistavoitteita.

3.5 Arvioinnin perustelu ja suhde tietoperustaan

Minulla on kertynyt työkokemusta peliohjelmoijana toimisesta harjoittelujaksoni jälkeen jo yli puoli vuotta, joten koen olevani lähtötasoisesti jo jossakin Taitava suoriutuja- ja Kokenut asiantuntija -tasojen välimaastossa. Työskentelyni on ollut jo pitkään hyvin itsenäistä ja itseohjautuvaa, ja työtehtäväni etenevät mallikkaasti. Koen omaavani myös melko laajan tietämyksen 3D-tasohyppelypelin avainelementeistä, koska kyseisen genren pelit ovat minulle entuudestaan hyvinkin tuttuja. Joshua Bycer (2019) käsittelee kirjoittamassaan teoksessa hienosti tunnettujen ja uraauurtavien tasohyppelypelien elementtejä, avaa niiden toteutustapoja sekä perustelee asioiden toimivuutta. Uskon voivani hyödyntää kyseistä teosta laajentamaan käsitystäni 3D-tasohyppelypelien maailmasta, valjastamaan hankitut tiedot ja taidot, ja hyödyntämään oppimaani kehittämässämme peliprojektissa. Viralliset Epic Games Unreal Engine -dokumentaatiot (2023) toimivat työskentelyn tukena, ja auttavat perustelemaan, ja kuvaamaan työtehtävissä käytettyjä ratkaisuja. Ne toimivat myös opetusmateriaalina osaamisen syventämisessä.

4 Sidosryhmät ja vuorovaikutus työpaikalla

4.1 Sidosryhmien listaus

Työpaikaltani löytyvät sidosryhmät:

-Työtoverit/kehitystiimi
-Ulkoiset alihankintapalveluiden asiakkaat
-Mahdolliset julkaisijaehdokkaat
-Ulkopuoliset palveluntarjoajat (äänipalvelut, sosiaalisen median markkinointi)
-Oheistuotteiden valmistajat/toimittajat
-Pelin testaajat
-Pelin fanikunta

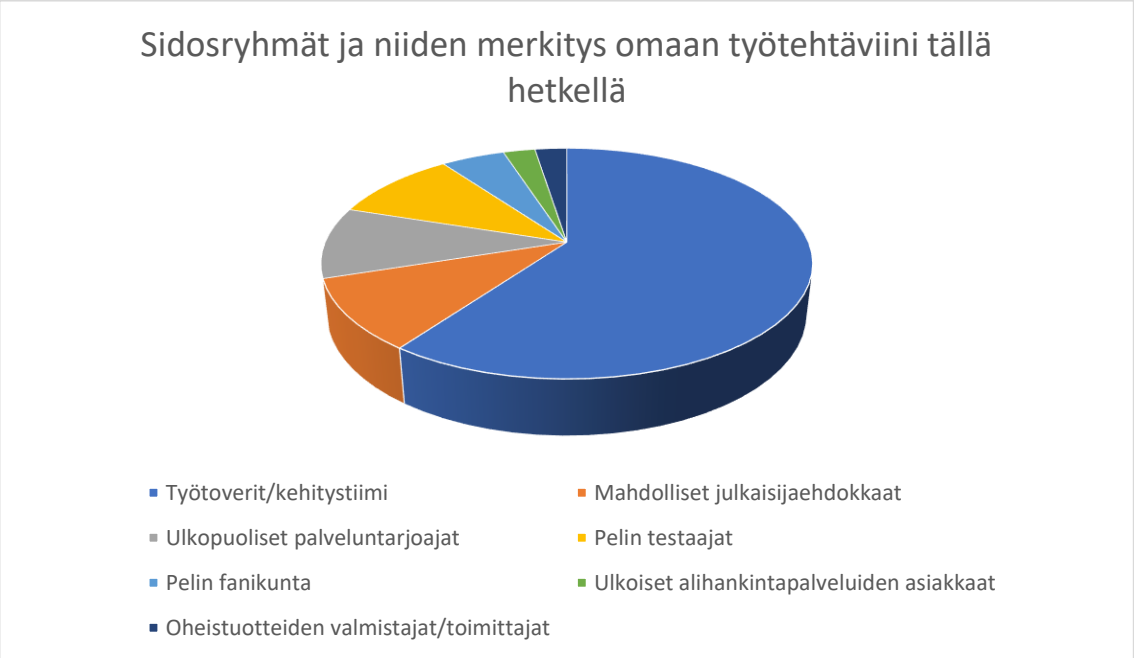
4.2 Sidosryhmien luokittelu

Sisäiset sidosryhmät:
Työtoverit/kehitystiimi

Ulkoiset sidosryhmät:
Ulkoiset alihankintapalveluiden asiakkaat, mahdolliset julkaisijaehdokkaat, ulkopuoliset palveluntarjoajat (äänipalvelut, markkinointi), oheistuotteiden valmistaja/toimittaja, pelin testaajat, pelin fanikunta

4.3 Kaavio sidosryhmistä

Kuvio (kuvio 1) kuvaa eri sidosryhmiä, ja niiden vaikutusta jokapäiväisiin työtehtäviini tällä hetkellä:



Kuvio 1. Sidosryhmät ja niiden vaikutus.

Mikäli julkaisijaehdokas löytyy, tulee heidän osionsa kaaviossa kasvamaan huomattavasti. Heidän osionsa kaaviosta kasvaisi arviolta 30-prosenttiin, ja vastaavasti työtoverit/kehitystiimi osio putoaisi 40-prosenttiin.

4.4 Sidosryhmien mielipiteet ja intressit

Kehitystiimin ja työtovereiden mielipiteet ja yhteiset intressit ovat äärimmäisen tärkeitä työn ja projektin kannalta, koska työtä tehdään pienessä kehitystiimissä, jossa jokaisella on oma roolinsa projektissa. Pelissä esiintyviä ja siihen tulevia mekaniikkoja mietitään yhdessä, niitä testataan ja niitä iteroidaan tiimin jäsenten kesken.

Ulkopuolisia palveluntarjoajia meillä on tällä hetkellä äänipalveluiden tarjoajat (pelin musiikit ja ääniefektit ovat tilattu yrityksen ulkopuolilista henkilöiltä), sekä yrityksen sosiaalisesta mediasta vastaava henkilö (hoitaa ja tekee päivityksiä pelimme sosiaalisen media kanaville, kuten Twitter, Instagram, Tiktok, Facebook, ja Discord). Heidän mielipiteensä reflektoivat heille antamiamme kriteerejä ja ohjeita, ja heille itselleen syntyneitä visioita näiden kriteerien pohjalta. Mikäli iteroitavaa ilmenee, asioista keskustellaan heidän kanssaan.

Pelin testaajat, ja fanikunta on myös omalta osaltaan melko suuressa roolissa työmme kannalta. Olemme jakaneet sosiaalisessa mediassa paljon kuvia ja videoita pelistämme, ja saamme sen kautta paljon palautetta. Pyrimme ottamaan nämä palautteet huomioon hienosäätäessämme mekaniikkoja, ja luodessamme lisää sisältöä peliin. Olemme suorittaneet myös virallisia pelitestauksia, kuvanneet pelitestaukset sekä saaneet suulliset ja kirjalliset kommentit testaajilta. Nämä pelitestauksissa esiintyneet mielipiteet ja asiat ovat kullanarvoisia projektin kannalta. Mikäli esimerkiksi suuri osa testaajista on sitä mieltä, että jokin asia ei toimi (tai vaihtoehtoisesti toimii todella hyvin), tulee se huomioida kehitysprosessin edetessä. Pelitestauksen kautta on myös saattanut ilmetä uusia bugeja, tai vaihtoehtoisesti löytynyt syitä jo olemassa oleville bugeille, joiden korjaamisen on ensiarvoisen tärkeää.

Mahdolliset julkaisijaehdokkaat ovat tärkein ulkoinen sidosryhmä. Meillä ei tällä hetkellä ole pelille julkaisijaa, mutta olemme koko ajan yhteydessä mahdollisiin julkaisijaehdokkaisiin. Mikäli joku heistä päättää julkaista ja rahoittaa pelimme loppukehityksen, ja pääsemme asiasta sopimukseen, tulee yhteistyömme heidän kanssaan olemaan todella tiivistä. Julkaisijan löytymisen jälkeen, tulee heidän mielipiteensä olemaan todella suuressa roolissa pelin jatkokehityksen kannalta. Heiltä voi mahdollisesti tulla aikataulutusta työtehtävien etenemiseen, ja työn tulosten esittelyyn. Mikäli julkaisija löytyy, saattaa projektin kehitysprosessi muuttua melkoisesti heiltä tulevien ohjeistusten ja toiveiden mukaan.

4.5 Vuorovaikutustilanteet työtovereiden kanssa

Olemme paljon vuorovaikutuksessa työtovereidemme kanssa. Jokaisella on oma roolinsa projektin työtehtävissä, joten hyvät vuorovaikutustaidot ovat projektin kannalta tärkeitä.

Pääsääntöisesti kuvio menee seuraavanlaisesti: Yrityksemme CEO on myös projektimme Creative Director, joten hänellä on projektin vision kannalta suurin sananvalta. Hän antaa visioidensa pohjalta muulle tiimille työtehtäviä. Hän antaa 2D artistillemme aiheita suunnitella konsepteja erilaisista asioista, esimerkiksi pelialueista, hahmoista tai pelimekaniikoista. Näiden konseptien pohjalta 3D artistit luovat 3D mallit tarvittavista esineistä, asioista tai pelihahmoista. Luotujen 3D mallien tai konseptikuvien pohjalta ohjelmoija luo kyseiset mekaniikat toimimaan peliin. Kenttäsuunnitteluja suunnittelee kenttiä hyödyntäen konseptikuvia ja Creative Directorin visioita. Kenttäsuunnittelija tekee myös paljon design -puolen työtehtäviä, eli myös häneltä saattaa tulla ohjelmoijalle toiveita erilaisista mekaniikoista ja asioista. Ohjelmoijan ja kenttäsuunnittelijan tekemiä toteutuksia arvioidaan yhdessä tiimin kanssa, ja niitä iteroidaan tarvittaessa.

Pidämme tiimin kanssa viikkopalaverin kerran viikossa, jossa viikon aikana tehdyt työtehtävät esitellään muulle tiimille. Vaikka vuorovaikutusta tapahtuukin työpaikalla koko ajan, toimii tämä viikkopalaveri aina eräänlaisena kulmakivenä projektin etenemisen arvioinnissa.

4.6 Vuorovaikutustilanteet sidosryhmien kanssa

Sidosryhmien kanssa esiintyvät vuorovaikutustilanteet ovat hieman erilaisia sidosryhmästä riippuen. Asiakaspalvelua sidosryhmien kanssa esiintyy tällä hetkellä ainoastaan alihankintapalveluiden asiakkaiden kanssa. Yrityksemme tarjoaa muille yrityksille erilaisia grafiikkapalveluita alihankintana. Viestintä alihankintapalveluiden asiakkaiden kanssa tapahtuu sähköpostitse, tai videopuheluiden välityksellä. Tällä hetkellä yrityksemme CEO vastaa täysin ulkoisten asiakkaiden kanssa kommunikoinnista.

Julkaisijaehdokkaiden kanssa viestitään sähköpostitse, tai videopuheluiden välityksellä. Tällä hetkellä yrityksemme CEO vastaa täysin ulkoisten asiakkaiden kanssa kommunikoinnista. Heidän kanssaan vuorovaikutustilanteita esiintyy tällä hetkellä lähettämiämme hakemuksia käsitellessä. Mikäli julkaisija pelille löytyy, tulee vuorovaikutustilanteet heidän kanssaan kasvamaan melkoisesti.

Tällä hetkellä yrityksen sosiaalisesta mediasta vastaava henkilö työskentelee muun tiimin kanssa paikan päällä toimistolla. Hänen kanssaan vuorovaikutus toimii siis täysin samalla tavalla, kuin kehitystiimin kanssa, ja hänen työtehtäviään arvioidaan samoilla periaatteilla.

Ulkoiset äänipalveluiden tuottajat työskentelevät tilaustyönä, ja heidän kanssaan kommunikoidaan pääasiassa sähköpostin välityksellä. Ääniefektien tuottajat ovat kotimaisia palveluntarjoajia, mutta pelin musiikkien tuottaja on japanilainen säveltäjä. Hänen kanssaan vuorovaikutus tapahtuu englanniksi.

Oheistuotteiden valmistajia ovat tällä hetkellä pelin fanituotteiden valmistajat ja toimittajat. Olemme tilanneet t-paitoja, pinssejä, magneetteja, tarroja ja pehmoleluja ulkoisilta oheistuotteiden valmistajilta. Heidän kanssaan viestiminen tapahtuu sähköpostitse, tai mikäli kyseessä on vain yksinkertainen tilaus verkkokaupan kautta, ei vuorovaikutusta ole lähestulkoon ollenkaan.

Pelin testaajien kanssa vuorovaikutustilanteet tapahtuvat paikan päällä toimistollamme. He kirjaavat pelitestaus sessionsa jälkeen palautteensa heille esitetylle lomakkeelle, jonka jälkeen suullinen palaute käydään läpi kasvotusten.

4.7 Mahdolliset haasteet sidosryhmien kanssa

Haasteita sidosryhmien kanssa oman osaamisen kannalta esiintyy tällä hetkellä hyvin vähän. Omaa työskentelyäni koskettavat sidosryhmät ovat tällä hetkellä ainoastaan oma kehitystiimi, sosiaalisen median markkinoinnista vastaava henkilö, ja mahdollisissa suoritettavissa pelitestaus sessioissa esiintyvät henkilöt.

Oman tiimimme kanssa viestintämme on sujuvaa, ja asioista on helppo keskustella kasvotusten paikan päällä. Yksi teoriassa mahdollinen haaste on, mikäli minua (ohjelmoijaa) pyydetään toteuttamaan jokin mekaniikka tai ominaisuus, mitä en yksinkertaisesti osaa tehdä. Tällaisissa tapauksissa tarvitsee miettiä, miten asian kanssa on järkevää edetä.

Mikäli julkaisija pelille löytyy, saattaa haasteita yhtäkkiä alkaa esiintyä. Tällaisia haasteita voivat olla esimerkiksi aikataulujen tiukentuminen työtehtävien suorittamisessa, tai jo olemassa olevien asioiden tai mekaniikkojen hienosäätö.

4.8 Arvioinnin perustelu ja suhde tietoperustaan

Omaan työhöni sidosryhmät vaikuttavat tällä hetkellä varsin vähän. Tämä johtuu siitä, että omaan työnkuvaani ei kuulu asiakkaiden kanssa työskentely ja heidän

kanssaan kanssakäyminen, koska ulkoisille asiakkaille menevät työt ovat 3D artistiemme työnkuvia. Työtehtäväni ohjelmoijana kuuluvat omaan peliprojektiimme, ja koska meillä ei tällä hetkellä ole ulkoista julkaisijaa, ovat aikataulut ja työtehtävät oman kehitystiimimme sisällä määriteltyjä.

Oman tiimimme sisällä viestintä on avointa ja sujuvaa, ja koska työskentelemme lähes täysin paikan päällä toimistolla, ei meillä ole tarvetta etäpalaverien järjestämiselle. Tiimimme on hyvin kartalla 3D-tasohyppelypelien peruselementeistä, ja erityisesti kenttäsuunnittelijallamme on paljon kokemusta mm. Mario- ja Zelda- pelisarjojen peleistä. Mario pelejä ja niiden mekaniikkoja käsitellään myös valitsemassani tietoperustan teoksessa Game Design Deep Dive: Platformers, Joshua Bycer (2019), joten teoksessa esiintyvät asiat ja elementit tulevat varmasti jossain muodossa vastaan kehittäessämme omaa peliämme. Viralliset Epic Games Unreal Engine -dokumentaatiot (2023) toimivat tukena käytettyjen ohjelmointiratkaisujen perustelussa oman tiimin kesken.

5 Päiväkirjatyöskentely

5.1 Seurantaviikko 1 (12.6.2023-16.6.2023)

MA 12.06.2023

Tämän päivän työtehtäviini kuuluu dokumentointi, lyhyen tutoriaalin teko, ja koodin lataaminen versionhallintaan viime viikon perjantaina valmiiksi saadusta työtehtävästä. Työtehtävä oli pelin Signal Plant -vastustajien toteutus. Tämän jälkeen alan tutustumaan pelissä olemassa olevan, mutta keskeneräiseksi jääneen seinäkiipeilymekaniikan toteutukseen. Seinäkiipeilymekaniikka on tarkoitus tehdä loppuun, ja sitä on tarkoitus hienosäätää seuraavaksi.

Tämän päivän osaamistavoitteisiin kuuluu oman työn kattava dokumentointi ja tiedostojen siirtäminen versionhallinnan kautta repositorioon.

Osaamistavoitteisiin kuuluu myös toisen ohjelmoijan tekemän koodin tulkitseminen, ymmärtäminen ja sisäistäminen.

Dokumentointi, tutoriaalin teko ja työn siirtäminen versionhallintaan sujui vaivattomasti. Näitä asioita olen tehnyt Hologram Monsterilla jo vuoden ajan, joten ne ovat minulle jo täysin arkipäivää. Seinäkiipeily- mekaniikkaan tutustuminen sujui myös varsin hyvin. Löysin kiipeilyyn liittyvät osat koodista, ja sain selville, miten kyseinen mekaniikka tällä hetkellä toimii. Sain myös jo ideoita, miten sitä voisi lähteä kehittämään eteenpäin.

Seinäkiipeily käyttää samaa Collision kanavaa kuin Rope kanava, ja seinän tunnistus tapahtuu Longtail hahmon Event Hit -tapahtuman kautta. Collision kanavan täsmätessä käynnistetään itse tehty Event Rope Hit event, jolloin hahmon action state asetetaan RopeClimbIdle tilaan, movement mode Flying tilaan, ja AttachToActor-solmulla hahmo kiinnitetään Actor-objektiin, jolla kyseinen pinta löytyy. Tämän jälkeen aloitetaan liike pystysuoralla pinnalla. Climb Event -tapahtuman aikana suoritetaan lukuisia erilaisia TraceForObjects raycast -tarkistuksia, jotka käyttävät Rope Collision -kanavaa. Longtail hahmon liikettä ohjataan näiden raycast tarkistusten perusteella.

Vaikka olenkin tehnyt dokumentaatiota ja työskennellyt versionhallinnan kanssa, on siihen aina hyvä saada lisää kokemusta. Päivän aikana sain tulkittua seinäkiipeilymekaniikan toteutusta, ja löysin koodista paljon hyödyllisiä toteutuksia erilaisista raycast tarkistuksista, joiden avulla hahmon liikettä on mahdollista ohjata haluttuun suuntaan.

TI 13.6.2023

Tämän päivän työtehtäviin kuuluu seinäkiipeily- mekaniikan hienosäätöä. Osaamistavoitteisiin kuuluu toisen ohjelmoijan kesken jääneen työn jatkaminen, ja eteenpäin vieminen.

Totesin että seinäkiipeily vaatii päivitetyt animaatiot, ennen kuin sitä kannattaa työstää enempää. Ehdin kuitenkin tehdä hieman parannuksia animaatioihin,

lisäämällä interpolaatio aikoja Blend State -tilaan. Lisäsin myös Longtail hahmon blueprintiin CornerTimer Timer -ajastimen, jonka aikana ei voi antaa lisää liike syötettä MovementClimb-tapahtuman aikana. Timer käynnistyy, kun MoverAroundCorner-tapahtuma alkaa, ja loppuu sekunnin päästä.

Tämän jälkeen minun oli tarkoitus aloittaa ohjattavan Spirit Animal -mekaniikan työstämistä. Minun tarvitsi kuitenkin alkaa tutkia yrityksessämme olevan harjoittelijan kanssa pelin ääniin liittyviä asioita. Harjoittelija on äänipuolen harjoittelija, ja hänen tarkoituksensa oli liittää FMOD niminen ääniohjelma Unreal Engine -pelimoottoriin, ja saada se toimimaan versionhallinnan kautta. Kyseinen asia aiheutti kuitenkin melko paljon päänvaivaa, koska sen toiminta versionhallinnan kanssa ei ollutkaan niin yksinkertaista, mitä alun perin ajattelimme. Saimme ohjelman toimimaan harjoittelijamme koneella pelimoottorin kanssa osittain, mutta versionhallintaan yhdistämistä emme saaneet toimimaan kunnolla. Asiaan tarvitsee palata huomenna.

Tämän päivän oppimistavoitteet toteutuivat siinä mielessä, että osasin tulkita seinäkiipeily mekaniikan tarvitsevan päivitetyt animaatiot, ennen kuin sitä on järkevää jatkaa. Yllättävä työtehtävä ääni asioiden kanssa muutti hieman päivän osaamistavoitteiden kulkua.

Versionhallinta osoittautui yllättävän haastavaksi linkittää FMOD ääni ohjelmaan, ja joudumme palaamaan asian pariin huomenna. Osaamiseni kehittyi jonkin verran FMOD ohjelman linkittämisen suhteen.

KE 14.6.2023

Tämän päivän työtehtäviini kuuluu FMOD ääniohjelman liittäminen Unreal Engine -pelimoottoriin, sekä sen yhdistäminen versionhallintaan Perforce ohjelman avulla. Tämän jälkeen minun on tarkoitus korjata muutama pieni hienosäätö Longtail hahmon Actor-objekteihin. Osaamistavoitteisiin kuuluu eri ohjelmien yhdistäminen, ja niiden onnistunut käyttöönotto, sekä Longtail hahmon ominaisuuksien parantelu.

FMOD ääniohjelman lisääminen onnistui. Longtail projektiin piti lisätä FMOD lisäosa, joka mahdollistaa FMOD ohjelmalla tehtyjen äänien käytön projektissa. Projekti piti sijoittaa versionhallinnan repositorioon, jonka jälkeen linkittäminen FMOD ohjelman omien asetusten kautta onnistui. FMOD osaa linkityksen jälkeen suoraa tehdä submit-kommennon Longtail projektiin, päivittää muuttuneet äänipankit, jonka jälkeen äänet toimivat kuten on ajateltu. Longtail hahmon blueprintiin lisäsin Collision Channel -tarkistuksen WallTrace tapahtumaan, joka tarkistaa onko seinä kiivettävä, ja onko pelaaja käyttämässä Dash kykyä. Mikäli on, Dash kyky lopetetaan, kiinnitetään pelaaja seinään ja aloitetaan seinäkiipeily. Tein myös pari siirtymä tarkistusta Longtail hahmon animaatioblueprintiin (koskettavatko jalat seinää, ja onko yläpuolella vapaata tilaa), joka mahdollistaa nopeamman reuna kiipeämisen, mikäli ehdot täyttyvät. Mikäli eivät, käytetään jo olemassa olevaa hitaampaa kiipeämistä. Lisäsin myös oman Animation Notify -tapahtuman reunalta kiipeämiselle, joka palauttaa kontrollin takaisin pelaajalle nopeammin kuin aiemmin.

Päivän osaamistavoitteet onnistuivat hyvin. Oli hienoa saada FMOD toimimaan onnistuneesti Unreal Engine pelimoottorin kanssa, ja nyt osaan jatkossakin tehdä yhdistämisen uudestaan tarpeen tullen. Tarvittavat hienosäädöt Longtail hahmon ominaisuuksiin sain myös toteutettua onnistuneesti.

TO 15.6.2023

Tämän päivän työtehtäviini kuuluu pitkään mysteerinä olleen Ground Punch -bugin selvittäminen ja korjaaminen. Bugin korjaamisen jälkeen on tarkoitus parannella ohjattavan laama hahmon liikkumista ja possession-haltuunottoa. Päivän osaamistavoitteisiin kuuluu bugien selvittämistä ja korjausta, sekä ohjattavuuden hienosäätöä. Tavoitteisiin kuuluu myös omien työtehtävien esittelyä.

Ground Punch -bugin selvittäminen oli varsin nopea toimenpide, koska löysimme eilen paikan, jossa bugi on mahdollista replikoida toistuvasti. Sain bugin poistettua, säätämällä animaatioblueprintin Transition Rule -sääntöjä,

lisäämällä tarkistuksen, että kyseinen lyönti on oikeasti Ground Punch -lyönti, eikä vain mikä tahansa lyönti.

Kontrolloitavien hahmojen ja objektien (kuten ratsastettava Laama) Camera Blend -siirtymiä oli tarve saada eri mittaisiksi, joten loin uuden enumeraation enumPossessionClass, joka kertoo kontrolloitavan asian luokan, ja se määritellään jokaisessa possession tapahtumassa erikseen. Tietoa käytetään uudessa luomassani makrossa, joka valitsee siirtymäajan kontrolloitavan asian mukaan. Laamalle loin oman pidemmän siirtymäajan, sekä säädin sen Dismount animaation siirtymäaikoja sulavamman näköisen lopputuloksen saavuttamiseksi. Ohjeistin myös äänipuolen harjoittelijaamme, mistä paikoista hänen tekemänsä ääni tapahtumien kutsuminen tapahtuu valituissa blueprinteissä. Kuvasin myös videoita viikon aikana tekemistäni työtehtävistä. Pidimme viikkopalaverin, jossa jokainen esitteli viikon aikana tekemänsä työtehtävät.

Huomasin osaamiseni kehittyneen, koska makron tekeminen kontrolloitavia asioita varten, on todella hyödyllinen jatkoa ajatellen, koska siihen on helppo lisätä asioita jälkikäteen, kun uusia kontrolloitavia asioita ilmenee. Tämänlaiset monikäyttöiset koodin osat ovat aina toimivia ratkaisuja, ja helpottavat asioiden käsittelyä jatkossa.

PE 16.6.2023

Tämän päivän työtehtäviini kuuluu laaman viimeistely tältä erää, harjoittelijan ohjeistusta, ja mikäli aikaa jää, uuden vihollisen mekaniikkojen aloittelu. Päivän osaamistavoitteisiin kuuluu ohjelmointi logiikasta kertominen alan harjoittelijalle, ja ohjattavan hahmon liikkeiden ja animaatioiden parantelu.

Lisäsin laamalle lyhyen Cooldown ajan, jonka aikana ei voi antaa uutta liike syötettä. Tämä tapahtuu tiettyjen animaatio tilojen jälkeen, ja tekee hahmon ohjaamisesta realistisempaa. Lisäsin laamalle automaatti “kaasu” vaihtoehdon, jonka aikana hahmo liikkuu eteenpäin automaattisesti. Ominaisuus on mahdollista säätää päälle tai pois yhdellä napilla. Tein laaman Dash

tapahtuman (eli eteenpäin pyrähdys) täysin uudenlaiseksi. Dash-kyky toimii nyt, kuten on ajateltu, ja sen käyttö on paljon sulavampaa. Lisäin laamalle kameran seuraamis tapahtuman, joka on samantyylinen kuin Longtail hahmolla. Tein laaman hyppäämisen toiminnon myös uusiksi. Hyppääminen toimii yhdessä animaatioblueprintin kanssa, ja sen tuntuma parani huomattavasti. Tein laaman animaatiotilakoneen uusiksi, lisäämällä sinne eri vaihtoehtoja hyppyanimaatioihin laskeutumisen tilanteesta riippuen. Loppujen lopuksi laaman animaatiot, sen ohjaustuntuma ja ominaisuudet paranivat huomattavasti. Kävin myös läpi äänipuolen harjoittelijamme kanssa lisää ääniin liittyviä asioita. Laaman viimeistelyyn ja äänipuolen asioiden läpikäymiseen meni loppujen lopuksi niin kauan aikaa, että en ehtinyt vielä aloittaa uuden vihollisen työstämistä.

Osaamiseni kehittyi jälleen animaatioblueprintien puolella, kun sain lisättyä laamalle vaihtoehtoja erilaisiin laskeutumisanimaatioihin päätymiseen. Sain myös laaman Character blueprintin toimimaan mukavasti yhdessä animaatioblueprintin kanssa.

Seurantaviikko 1 – Viikkoraportti: 12.6.2023 – 16.6.2023

Tämän viikon aikana työskentelin melko paljon äänipuolen harjoittelijamme kanssa, häntä ohjeistaen, sekä FMOD ohjelman käyttöönoton kanssa. FMOD on audio ohjelma, jonka kautta ääniä on mahdollista hallita monipuolisesti. Ohjelma piti integroida Unreal Engine -pelimoottoriin, sekä versionhallintaan käyttämämme Perforce työkalun avulla. Kyseinen toimenpide oli minulle täysin uusi asia, mutta sen selvittäminen onnistui, ja saimme kaikki ohjelmat toimimaan yhdessä onnistuneesti. FMOD projekti piti siirtää versionhallinnan repositorio kansion alle, jonka jälkeen Perforce ohjelma tunnisti projektin. Unreal Engine tarvitsi myös lisäosan (plugin), jotta FMOD äänien käyttö oli mahdollista. Tämä lisäosa tuli ladattua, ja lisättyä projektikansioon, jonka jälkeen se on käytettävissä kaikille tiimin jäsenille. (Firelight Technologies 2023)

Viikon aikana työskentelin myös Longtail hahmon ominaisuuksien parissa. Alkuviikosta aloitettu seinäkiipeilymekaniikan jatkotyöstäminen jäi vielä

keskeneräiseksi, koska totesimme että kiipeilyanimaatiot tulee päivittää, ennen kuin mekaniikasta on mahdollista tehdä sulavaa. Sain paranneltua hahmon reunalta ylös kiipeämistä, korjattua Ground Punch -kykyyn liittyvän bugin, ja luotua hyödyllisen makron, jota on mahdollista hyödyntää lisää tulevaisuudessa. Ratkaisin kiipeämiseen ja bugin selvittämiseen liittyvät asiat tekemällä muutoksia hahmon animaatioblueprintin siirtymiin, lisäämällä niihin erilaisia lisätarkistuksia, jolloin animaatioihin päädytään täsmälleen halutuissa tilanteissa. Työstin myös toisen ohjattavan hahmon, eli laaman kykyjä, animaatioita ja liikkeitä. Laaman animaatioiden parantelun toteutin lisäämällä animaatioblueprintiin vaihtoehtoja eri animaatioihin siirtymiseen, ja laitoin niille tarvittavat säännöt eri tilanteita varten. Laaman hyppy kykyä parantelin tekemälle siitä täysin oman toteutuksen, sen sijaan että käytettäisiin pelimoottorin omaa Jump komentoa. Lisäsin hienosäätöjä hahmon ilma kontrolliin, ja lisäsin hyppyyn luonnollisuutta lisäämällä siihen velocity arvoa hahmon liikkeestä. Dash mekaniikkaan lisäsin Cooldown ajan, säädin sen antamaa liikettä yhdessä animaatioblueprintin kanssa, ja lisäsin siihen tarvittavat tapahtumat, mikäli Dash kyky pysäytetään manuaalisesti.

Osaamiseni kehittyi todella paljon FMOD ohjelman integroimisen suhteen. Koska kyseessä oli minulle täysin uusi operaatio, tarvitsi asiaa selvitellä erityisesti versionhallinnan puolelta melko paljon. Integrointi onnistui kuitenkin hyvin, ja nyt osaan tehdä sen myös tarvittaessa jatkossakin. Kehitin osaamistani myös animaatioblueprintien hienosäätöjen ja tapahtumien osalta. Sain animaatiot toimimaan sujuvasti hahmojen Character- ja animaatioblueprintien välillä.

Erilaisten tekniikkojen hyödyntäminen pelin kentissä etenemiseen on pelimme avainelementtejä. Esimerkiksi laaman hyppy mekaniikan kontrollin parantaminen loi lisää luontevuutta hahmon ohjaamiseen. Joshua Bycerin (2019) mainitsema “variable jumping”, eli hyppääminen jossa pelaajalla on enemmän hallintaa hahmonsa kanssa ilmassa, tekee 3D ympäristössä liikkumisesta vaivattomampaa ja miellyttävämpää. Sen sijaan, että pelaaja olisi lukittu mihin suuntaan tahansa aloitushyppy alkoi, hän voi liikkua vasemmalle tai oikealle hahmon suurimman hyppyalueen sisällä. Toisin kuin yhteen

suuntaan sitoutunut hyppy, “variable jumping“ mahdollistaa paljon enemmän vaihtelua eri muuttujien perusteella. (Bycer, 2019, 14.) Koen sen olevan toimiva toimintamalli, ja sopivan myös tähän tilanteeseen erinomaisesti.

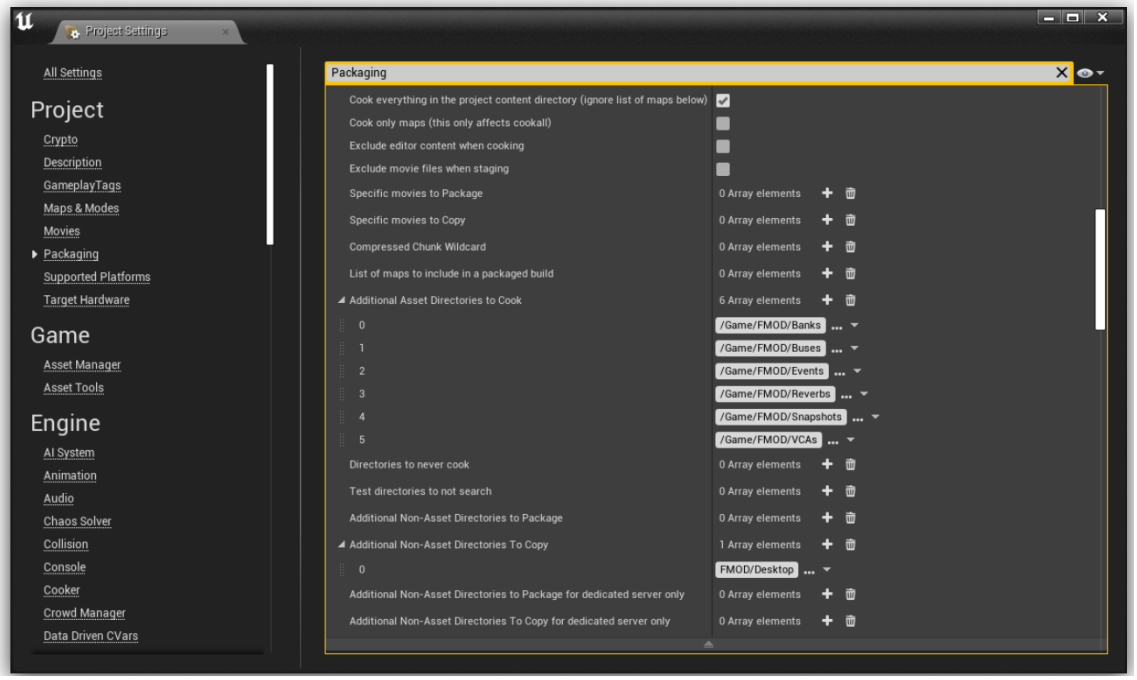
Dash mekaniikan puolesta olisi teoriassa ollut mahdollista tehdä mekaniikasta toisenlainen, tekemällä siitä vain luotisuoraa tietyn matkan eteenpäin tapahtuva pyrähdys. Koin kuitenkin sen tuntuvan paremmalta, antamalla pelaajalle kontrollia kyseisen mekaniikan aikana. Päädyin kyseiseen toimintamalliin, koska tavoite oli luoda ohjastettava ja melko luonnolliselta tuntuva mekaniikka. Erilainen pyrähdystekniikka voisi toimia eri tilanteessa ja erimerkiksi ilmassa tehtävälle pyrähdykselle. Tällaista ei kuitenkaan ole tällä hetkellä suunniteltu toteutettavaksi laama- hahmollemme, ja tekemäni toteutus toimii paremmin ratsastettavalle pelihahmolle.

5.2 Seurantaviikko 2 (19.6.2023-22.6.2023)

MA 19.6.2023

Tämän päivän työtehtäviini kuuluu pelin pakkaaminen, pakatun pelin testaaminen ja mahdollisten ilmenneiden bugien korjaaminen. Päivääni kuuluu myös täysin uuden apina tyylisen vihollis- hahmon aloittaminen. Päivän osaamistavoitteisiin kuuluu muun muassa Character-, ja animaatioblueprintin luonti, AI-Controller -luokan luonti ja Behavior Treen -käytöspuun aloittelu.

Pelin pakkaaminen sujui tuttuun tapaan, koska tarvittavat asetukset on asetettu jo etukäteen aiemmin. Peliä testatessa huomasin heti, että FMOD äänet eivät toimineet lainkaan pakatussa versiossa. Asiaa tutkiessani selvisi, että FMOD tiedostot täytyy lisätä pelin projektin asetuksiin "Additional Asset Directories to Cook"- ja "Additional Non-Asset Directories to Copy" -listoille (kuva 1). Tämän jälkeen FMOD äänet lisättiin pakattuun peliin, ja äänet toimivat normaalisti.



Kuva 1. Projektiasetukset ja FMOD äänien lisääminen pelin pakattuun versioon Unreal Engine -pelimoottorissa. (Kuva: Firelight Technologies).

Testaamisen aikana ilmeni myös muita pieniä korjattavia asioita. Kerättävien kristallien keräämis- liike tuntui liian hitaalta, joten lisäsin siihen hieman kerrointa. Samalla poistin keräämis- liikkeen Energy Vessel -objekteilta, koska niihin se oli erään aiemman päivityksen myötä tullut mukaan. Lisäsin peliin CheckPoint eli välitallennus pisteisiin ominaisuuden, joka parantaa Longtail hahmon elinvoiman täysiin.

Ehdin myös aloitella uutta apina tyylistä vihollista. Tein hahmolle Character- ja animaatioblueprintin, AI-Controller-luokkan ja Behavior Tree -käytöspuun. Animaatioblueprintiin ehdin tehdä animaatiotilakoneen, muutaman animaatiotilan ja siirtymät niiden välille. AI-Controlleriin-luokkaan ehdin luoda jo hieman logiikkaa, ja yhdistin kaikki edellä mainitut toimimaan yhdessä.

Päivän tavoitteet toteutuivat hyvin. Ehdin luoda hyvän pohjan apina hahmolle, ja pelin pakattuun version ehdin tehdä siinä ilmenneet korjaukset. Opin myös taas lisää FMOD ohjelman integroimisesta, ja muistan jatkossa liittää FMOD tiedostot polkuineen pelin projektitiedostoon.

TI 20.6.2023

Tämän päivän työtehtäviini kuuluu lisätä uimismekaniikkaan ylös ja alaspäin mahdollisuus. Tämä ominaisuus oli käytössä aiemmin, ennen kuin uimismekaniikka uusittiin. Nyt se olisi kuitenkin tarkoitus vielä lisätä nykyiseen mekaniikkaan lisä ominaisuudeksi. Tämän jälkeen minun on tarkoitus jatkaa apina vihollishahmon työstämistä. Päivän osaamistavoitteisiin kuuluu ominaisuuden lisääminen jo olemassa olevaan kokonaisuuteen, sekä tekoäly logiikan luomista vihollis hahmolle.

Lisäsin uuden Action Mapping -vaihtoehdon SwimUpDownMode, joka on mahdollista käynnistää painamalla L2- tai vasen shift -näppäimiä. Napin ollessa pohjassa Longtail siirtyy SwimUpDownMode tilaan ja voi liikkua ylös ja alas veden alla ylös- ja alas (ohjaimen vasen sauva tai W/S näppäimistöllä) näppäimillä. Kun näppäin (L2 tai vasen shift) vapautetaan, SwimUpDownMode tila päättyy. Jos pelaaja alkaa uida eteenpäin (R1- tai hiiren vasen painikke) SwimUpDown tilassa, SwimUpDownMode tila poistetaan automaattisesti. Jos pelaaja pitää edelleen L2- tai vasen shift -näppäintä painettuna, vapauttaessaan eteenpäin unti painikkeen, SwimUpDownMode-tila aktivoituu automaattisesti uudelleen. Longtail hahmon pitch-kierto pidetään nollassa, kun SwimUpDownMode-tila on aktiivinen ja pelaaja antaa liike syötettä.

Apina vihollishahmoon sain myös luotua paljon lisää logiikkaa. Sain Behavior Treen -käytöspuun kasattua jo lähestulkoon valmiiksi, käyttäen AI-Controller-luokkaa yhdessä siirtymien kanssa. Lisäsin logiikkaa myös apinan Character-blueprintiin. Apina on siis valmis hyökkäysmekaniikan työstämisen aloittamista varten.

Päivän osaamistavoitteet onnistuivat hyvin. Sain uimismekaniikkaa laajennettua entisestään, sekä lisättyä paljon logiikkaa apina hahmon tekoälyyn.

KE 21.6.2023

Tämän päivän työtehtäviini kuuluu apina hahmon puusta putoamis- mekaniikan teko, ja mikäli aikaa jää, myös hyökkäysmekaniikan aloittelu.

Päivän osaamistavoitteisiin kuuluu vihollis- hahmon ominaisuuksien laajentaminen.

Tarvitsin jotain, jonka avulla tunnistaa, että tietyn Static Mesh -objektiin on yhdistetty apina hahmo. Päädyin tekemään erillisen Hit Detection -komponentin, jonka voi liittää mihin vain Static Mesh -objektiin. Lisäsin Longtail hahmon lyönti tapahtuman tunnistamaan Static Mesh -objektit, joihin on liitetty kyseinen komponentti. Komponenttiin on mahdollista lisätä erilaisia käsittelyjä osumalle, ja lisäsinkin sinne jo käsittelyt äänelle, sekä objektin fyysiselle pinnalle, jonka mukaan haluttu ääni soitetaan (esimerkiksi puu, kivi, ja niin edelleen). Komponentin kautta on mahdollista käynnistää nyt myös apinahahmon puusta putoamistapahtuma. Kyseinen komponentti on todella hyödyllinen, koska sitä on mahdollista käyttää jatkossa tekemään erilaisia haluttuja käsittelyjä Static Mesh -objektien kanssa, joilla ei ole erikseen omaa blueprintiä.

Apina hahmolle tehty putoamistapahtuma, joka käynnistyy, kun Longtail lyö puuta (tai muuta Static Mesh -objektia) johon apina on liitetty. Apina putoaa maahan, on hetken pökertyneenä, jonka jälkeen hyppää takaisin puuhun. Hyppy tapahtumaa täytyy vielä säätää myöhemmin, kun hahmo saa animaatiot. Hyökkäysmekaniikkaa en ehtinyt vielä jatkaa.

Päivän osaamistavoitteet onnistuivat hyvin, ja olen erityisen tyytyväinen tekemästäni Hit Detection -komponentista. Kyseinen komponentti on todella hyödyllinen myös muihin asioihin kuin apina hahmon käyttämiseen. Tällaiset yleishyödylliset ja monikäyttöiset asiat ovat tarpeellisia pelin toiminnallisuuksien rakentamisessa. Laajensin osaamistani entisestään monikäyttöisten komponenttien käytön suhteen.

TO 22.6.2023

Tämän päivän työtehtäviini kuuluu apinahahmon työstämisen jatkaminen. Päivän osaamistavoitteisiin kuuluu vihollishahmon ominaisuuksien laajentaminen.

Lisäsin apina hahmon animaatioblueprintiin Falling- ja HitGround -animaatiotilat, ja lisäsin siirtymät niiden välille. Lisäsin samat tilat myös Behavior Treeh -käytöspuuhun. Animaatioblueprint ja Behavior Tree -käytöspuu toimivat nyt yhdessä, ja vaikka hahmolla ei ole vielä valmiita animaatioita, vaihtuvat tilat taustalla oikein. Tästä on oikein hyvä jatkaa, kun saan hahmon animaatiot käyttööni. Jatkoin myös hahmon hyökkäysmekaniikan tekoa hieman. Hahmo luo nyt heitettävän asian, joka liitetään hahmon käden Socket sijaintiin. Kun saan hahmon animaatiot käyttööni, luon heitettävälle asialle täysin oman Socket sijainnin, johon asia kiinnitetään. Ehdin myös aloittaa hahmon heitto tapahtumaa. Hahmo heittää asian nyt Longtail hahmoa kohti, tosin itse heiton tarkkuutta ja nopeuttaa tarvitsee säätää. Kuvasin myös videoita viikon aikana tekemistäni työtehtävistä.

Sain laajennettua apina hahmon ominaisuuksia, ja luotua mainion pohjan animaatioiden käyttöönottoa varten. Päivän osaamistavoitteet täyttyivät, ja opin lisää Behavior Treen -käytöspuun ja animaatioblueprintin toiminnasta. Viikkopalaveri jäi minun osaltani tällä viikolla väliin, koska vähensin kertyneitä ylityötunteja, ja tein pari tuntia lyhyemmän työpäivän.

Seurantaviikko 2 – Viikkoraportti: 19.6.2023 – 22.6.2023

Tämän viikon aikana työskentelin muutaman erilaisen asian parissa. Aloitin täysin uuden vihollistyypin tekemisen, joka piti sisällään monta eri osa-aluetta, kuten Character- ja animaatioblueprintit, AI-Controller-luokan, ja Behavior Tree-käytöspuun. Hieman haasteellista kyseisestä tehtävästä teki animaatioiden puuttuminen, joten jouduin seuraamaan AI- ja animaatiotilakoneen -tilojen vaihtumista koodin tasolla. Lisäsin itselleni paljon Debug tarkoitukseen käytettyjä tekstejä, ja käytin Unreal Engine -pelimoottorin omia AI Debugging -

työkaluja. Sain kaikki logiikat toimimaan yhdessä, ja uusi hahmo on valmiina odottamassa animaatioita. Hahmon ominaisuuksista kesken jäi vielä hyökkäysmekaniikka.

Pakkasimme pelin jälleen testaamista, ja julkaisija ehdokkaille lähettämistä varten. Huomasimme että uudet FMOD äänet eivät toimineet pakatussa pelissä. Tutkiskelin asiaa, ja ratkaisu ongelmaan oli seuraavanlainen: FMOD tiedostot täytyi manuaalisesti lisätä pelin projektin asetuksiin "Additional Asset Directories to Cook"- ja "Additional Non-Asset Directories to Copy" -listoille. Lisäyksen jälkeen FMOD äänet lisättiin pakattuun peliin, ja äänet toimivat normaalisti. (Firelight Technologies 2023)

Uimismekaniikka kaipasi hieman laajennusta, useaan otteeseen saadun palautteen vuoksi. Ajattelimme kokeilla, miltä aiemmin pelissä ollut ylös- ja alaspäin uimismekaniikka tuntuisi nykyisen erillisellä “kaasulla” toimivan mekaniikan lisäksi. Lisäsin ylös- ja alaspäin- uimismahdollisuuden tekemällä sille oman Action Mapping -tapahtuman, joka on mahdollista säätää päälle ja pois painamalla ja vapauttamalla kyseiselle toiminnolle asetettu nappi. Mekaniikan lisääminen vaati jonkin verran huomiointia eri puolella koodia, jotta erilaiset näppäinyhdistelmät tuli huomioitua. Ominaisuuden lisääminen onnistui, ja uimismekaniikka laajentui.

Tarvitsin jonkun tavan, jonka avulla tunnistaa tiettyyn Static Mesh -objektiin yhdistetty apina hahmo, ja kyseiseen objektiin kohdistuneet iskut. Päädyin luomaan uuden Actor-komponentin, joka on mahdollista liittää mihin vain Static Mesh -objektiin, ja joka liitettäessä asettaa päälle kyseisen objektin Generate Overlap Events -valinnan päälle. Tämän jälkeen lisäsin Longtail hahmon lyönti tapahtuman raycast -tarkistuksiin tunnistuksen objekteihin, joihin kyseinen komponentti on liitetty. Komponentti tuli loppujen lopuksi todella hyödylliseksi, koska sitä voi käyttää myös esimerkiksi äänien soittamiseen iskujen yhteydessä, tunnistamalla kyseisen objektin fysiikkapinta, ja soittamalla haluttu ääni pinnan materiaalin mukaan. Komponenttia on myös mahdollista laajentaa tulevaisuudessa. Komponentit ovat hyödyllisiä jaettaessa yhteisiä käyttäytymismalleja eri luokkien kesken, kuten kyky näyttää visuaalinen tehoste,

toistaa ääniä tai käsitellä käyttäytymislogiikkaa. Actor-komponenteilla ei ole Transform-ominaisuutta, mikä tarkoittaa, että niillä ei ole fyysistä sijaintia tai rotaatio arvoa maailmassa. (Epic Games 2023)

Tämän viikon työtehtävistä olin erityisen tyytyväinen luomaani komponenttiin, koska mitä monikäyttöisempi jokin tekemäni ominaisuus on, sen arvokkaammaksi kyseisen ominaisuuden arvo kasvaa. Päädyin ratkaisemaan asian komponentin kautta, koska se on mahdollista liittää asioihin, joilla ei ole omaa blueprintiä. Voidaan siis välttää niin sanotusti “ylimääräisten” blueprintien luonti, mikäli halutaan seurata esimerkiksi vain kyseisen asian Hit Event -tapahtumia. Minun tarvitsee tosin miettiä, tarvitseeko komponentille kehittää jonkinlainen tapa lisätä se esimerkiksi kaikkiin Static Mesh -objekteihin pelin latausruudun aikana. Tällä tavalla vältyttäisiin siltä, että komponentti tarvitsisi lisätä joka ikiseen haluttuun asiaan manuaalisesti.

5.3 Seurantaviikko 3 (26.6.2023-30.6.2023)

MA 26.6.2023

Tämän päivän työtehtäviini kuuluu aiemmin tekemiemme Music Trigger -objektien muuttaminen siten, että niillä on mahdollista käynnistää ja pysäyttää FMOD tapahtumia. Mikäli aikaa jää, myös apina hahmon hyökkäyksen työstäminen. Päivän osaamistavoitteisiin kuuluu olemassa olevien itse tehtyjen Trigger Volume -objektien muokkaaminen yhteen sopivaksi FMOD tapahtumien kanssa. Mikäli aikaa jää apina hahmon työstämiseen, osaamistavoitteisiin kuuluu hyökkäysmekaniikan työstämistä, ja heittokaaren laskemista kohteeseen.

Lisäsin Music Trigger -objekteihimme vaihtoehdon käyttää FMOD ääniä. Käytin valitsimena enumeraatio arvoa pelimme musiikkikappaleista, joka taas johtaa Switch To Enum -solmuun, jonka avulla voidaan valita haluttu FMOD tapahtuma. Harjoittelijamme tutkiskelee miten musiikkia ja ambience ääniä

voidaan soittaa FMOD ohjelman kautta, ja hän voi hyödyntää Music Trigger -objekteihin tekemääni pohjaa äänien kokeilemiseen.

Työstin myös apina hahmon hyökkäysradan laskemista. Tutkiskelin pelimoottorin “Predict Projectile Path”- ja “Suggest Projectile Velocity” -solmuja. Sain niiden avulla laskettua kohtalaisen lentoradan heitettäville asioille, mutta kyseinen ominaisuus vaatii vielä hienosäätöä muun muassa Collision Channel -kanavien säätämisessä.

Päivän osaamistavoitteet onnistuivat. Tein FMOD musiikin soittamiselle pohjan aiemmin tekemiimme Music Trigger -objekteihin, ja sain työstettyä apina hahmon hyökkäys tapahtumaa eteenpäin. Opin hyödyntämään heitto radan laskemiseen liittyviä solmuja, ja ymmärtämään niiden toimintaa.

TI 27.6.2023

Tämän päivän työtehtäviini kuuluu apina hahmon heittomekaniikan jatkaminen. Osaamistavoitteisiin kuuluu heittämisen hallintaa, ja hienosäätöä, sekä vahingon lisääminen heitettäville asioille.

Sain hiottua heittoradat sopiviksi, ja apinan heittämät objektit lentävät nyt sulavasti kohteeseen, eli pelaajan sijaintiin heitto hetkellä. Lisäsin heitettävien asioiden Hit Event -tapahtumaan vahingon lisäämisen Longtail hahmoille, sekä pienen Knockback efektin, joka työntää hahmoa taaksepäin. Heitettävät objektit myös tuhoutuvat niiden osuessa johonkin. Sain ratkaistua heitto radan vinoon lentämisen säätämällä Collision Channel -kanavia, jolloin heitettävä asia ei osu itse apinaan, vaan se säädetään osumaan Pawn Channel -kanavan objekteihin vasta asian lentäessä apinan kädestä. Tällöin heitto lentää suoraa, ja osuma tarkistetaan vasta sen osuessa Longtail hahmoon tai muihin asioihin, kuten seiniin tai vastaaviin. Ehdin aloittaa myös hieman uutta tapahtumaa, jonka tarkoitus on saada kaikki lähellä olevat apina hahmot nauramaan, mikäli heitettävä asia osuu Longtail hahmoon.

Työstimme yhdessä äänipuolen harjoittelijamme kanssa musiikin soittamisen siirtämistä FMOD ohjelman puolelle. Tekemiimme Music Trigger -objekteihin lisättiin ominaisuus vaihtaa parametria FMOD tapahtumien raidoilla, joiden avulla voidaan vaihtaa kyseisellä hetkellä soiva kappale. Game Instance -objektiin loimme uuden tapahtuman, joka lähtee soittamaan musiikkia sillä hetkellä aktiivisena olevan kentän mukaan, ja valitsee kyseiselle kentälle valitut musiikit. Tämä event taas käynnistetään, kun suljetaan Loading Screen -UI-elementti.

Päivän tavoitteet onnistuivat hyvin. Olen mielissäni apinan heitto tapahtuman onnistumisesta, kuten myös siitä, että musiikkia on nyt mahdollista soittaa FMOD ohjelman kautta. Opin taas ymmärtämään lisää FMOD ohjelman integroimisesta Unreal Engine -pelimoottorin kanssa, ja hyödyntämään Projectile Velocity -solmuja heitto ratojen laskemisessa.

KE 28.6.2023

Tämän päivän työtehtäviini kuuluu apina hahmon nauramismekaniikan jatkaminen, putoamismekaniikan laajentaminen ja musiikin soittamisen implementointia FMOD ohjelman kautta harjoittelimme kanssa. Osaamistavoitteet pitävät sisällään Behavior Tree -käytöspuun työstämistä, erilaisten raycast tarkistusten toteuttamista sekä FMOD parametrien säätämistä Unreal Engine -pelimoottorin kautta.

Lisäsin apina- hahmojen heitettävien asioiden osuma tapahtuman raycast tarkistukseen, joka tarkistaa läheisyydessä olevat apinat, ja käynnistää niiden nauramistapahtuman, mikäli heitettävä asia osuu Longtail hahmoon. Nauramistapahtuma estää uusien heitettävien asioiden luomisen, ja pudottaa maahan jo luodut heitettävän asian, mikäli sellainen jo löytyy. Nauramistapahtuman jälkeen apina palaa takaisin perus ohjauskäyttäytymiseen. Laajensin putoamistapahtumaa toimimaan myös apinoiden kanssa, jotka eivät roiku, vaan seisovat oksan päällä. Loin uuden makron, joka tarkistaa apina hahmon ympäriltä vapaata, jonka suuntaan on mahdollista pudota. Makro aloittaa tarkistuksen hahmon takaa, ja suorittaa

uusia raycast tarkistuksia 30 asteen kulmassa suhteessa alkupisteeseen. Mikäli vapaa tila löytyy, eli mikään objekti ei estä raycast tarkistuksen Visibility Channel -kanavaa, käännetään hahmo poispäin kyseisestä suunnasta, tönäistään kohti vapaata tilaa, ja käynnistetään putoamistapahtuma.

FMOD musiikin soittamista varten harjoittelijamme FMOD ohjelman puolelle parametrit, joiden avulla musiikkikappaleita voidaan vaihtaa, vaihtoehdon käyttää Fade In- tai Fade Out -efektejä ja säätää musiikiin äänenvoimakkuutta. Ohjeistin häntä, miten kyseisiä ominaisuuksia voidaan säätää Unreal Engine -pelimoottorin. Laitoimme pelin Options valikon Volume Slider -säätimen säätämään FMOD musiikin voimakkuus parametria, sekä hyödyntämään musiikki päälle- tai pois nappia. Musiikit soivat nyt täysin FMOD ohjelman kautta, ja niiden säätäminen sen kautta on huomattavasti dynaamisempaa.

Päivän osaamistavoitteet sujuivat mainiosti, ja sain asioita hyvin eteenpäin. Laajensin osaamistani vektorien kiertämisessä raycast tarkistusten kanssa, ja opin hyödyntämään FMOD ohjelmaan asetettuja parametreja Unreal Enginen -pelimoottorin kautta.

TO 29.6.2023

Tämän päivän työtehtäviini kuuluu apina hahmon hienosäätö ja viimeistely tältä erää, sekä musiikin soittamisen käsittelyä FMOD tapahtumilla kenttää vaihtaessa. Osaamistavoitteisiin kuuluu hahmon hienosäätö, koodin siistiminen ja toiminnallisuuden testaaminen, sekä FMOD ohjelman kautta soivan musiikin käsittely kenttien välisissä siirtymissä.

Tein apina hahmolle oman DropProjectile tapahtuman, jota käytetään tarvittavissa paikoissa. Tilanteen vaatiessa voidaan siis kutsua suoraa kyseistä tapahtumaa, ilman että kirjoitetaan sama koodi useaan otteeseen. Hienosäädin vielä apinan arvoja, ja varmistin tämänhetkisten ominaisuuksien toimivuuden. Siistin ja kommentoin koodin apinan osalta, ja puskin tekemäni asiat versionhallintaan.

FMOD musiikin osalta ohjeistin harjoittelijaamme, miten tehdään käsittelyn musiikin pysäyttämiselle ja soittamiselle kenttien välisissä siirtymissä. Lisäsimme Game Instance -objektin FMODPlayMusic tapahtumaan vaihtoehdon MainMenu kentälle, ja musiikin lopettamissolmut LoadingScreen- DemoEndScreen- ja MainMenu -UI- elementeille. Kyseisten käsittelyiden hoitaminen aiheutti jonkin verran päänvaivaa, koska kyseisiä asioita käsitellään usean eri blueprintin kautta. Saimme kuitenkin sopivat käsittelyt tehtyä, ja musiikin toimimaan suunnitellusti.

Sain apina hahmon viimeisteltyä tältä erää, ja hahmo jää odottamaan animaatioita. Laitoin tuotokseni jakoon versionhallintaan muille saatavaksi. FMOD äänet vaativat taas työstöä, mutta saimme ne harjoittelijamme kanssa onnistuneesti toimimaan. Päivän osaamistavoitteet siis täyttyivät hyvin.

PE 30.6.2023

Tämän päivän työtehtäviini kuuluu kokeilla, saisimmeko projektimme toimimaan Unreal Engine -pelimoottorin versiossa 4.27. Osaamistavoitteisiin kuuluu projektin päivittämisen opiskelu, projektitiedostojen uudelleen luominen, ja mahdollisten lisäosien uudelleen käyttöönotto.

Projektimme on alun perin tehty Unreal Engine -pelimoottorin versiolla 4.26.2. Kokeilin projektin päivittämistä, tein hieman asioiden selvittelyä, ja loppujen lopuksi sain projektin toimimaan versiossa 4.27. Tein seuraavanlaiset toimenpiteet projektin päivittämiseen onnistuneesti: Poistin projektin kansiosta .vs, binaries, build, intermediate, saved, .sln kansiot ja tiedostot. Avasin projektitiedoston tekstieditorilla ja otin manuaalisesti pois käytöstä sellaiset lisäosat, jotka eivät ole tuettuna versiossa 4.27, meidän tapauksessamme se oli FMOD. Tämän jälkeen loin Visual Studio -projektitiedostot uudestaan, avasin uudelleen luodun Solution tiedoston, ja painoin Rebuild painiketta projektin kohdalla. Tämä prosessi siivosi ja loi tiedostot uudelleen. Lopuksi tarvitsi vielä ajaa projekti Visual Studio -ohjelman kautta. Unreal Engine -pelimoottori käynnistyi, ja projekti aukesi onnistuneesti versiossa 4.27.

FMOD tarvitsi vielä päivittää manuaalisesti. Latasin FMOD ohjelman Unreal Engine -pelimoottorin versio 4.27 lisäosan, ja kopioin sen uuden 4.27 projektin Plugins kansioon. Avasin Unreal Engine -pelimoottorin, otin FMOD lisäosan käyttöön, ja käynnistin ohjelman uudelleen. Tämän jälkeen koko projekti toimi normaalisti versiossa 4.27. En ole ennen tehnyt kyseistä projektin päivittämistä täysin manuaalisesti itse, joten tämä oli todella opettavainen prosessi. Olin mielissäni, kun sain projektin onnistuneesti päivitettyä, ja osaan tehdä sen nyt tarvittaessa jatkossakin. Tämän jälkeen voimme harkita kokeilla päivittää projekti jopa Unreal Engine 5 -versioon, tosin se vaatii hieman enemmän toimenpiteitä.

Seurantaviikko 3 – Viikkoraportti: 26.6.2023 – 30.6.2023

Tämän viikon aikana työskentelin muun muassa uuden vihollishahmon parissa, kehittelin tapoja pelin musiikin soittamiseen FMOD ohjelman kautta, ja kokeilin, onnistuuko projektimme päivittäminen uudempaan Unreal Engine -pelimoottorin versioon.
FMOD asioiden parissa sai työskennellä tälläkin viikolla, ja keksiä tapoja, joita äänipuolen harjoittelijamme voi hyödyntää musiikin soittamisessa FMOD ohjelman kautta. Loin jo olemassa oleviin tekemiimme Music Trigger -objekteihin vaihtoehdon hyödyntää musiikkia FMOD ohjelman kautta. FMOD ääniä käytettäessä, Game Instance -objekti tarkistaa sillä hetkellä auki olevan pelikentän, Music Trigger -objekti tarkistaa kyseiseen objektiin parametrina asetetun musiikkikappaleen, ja syöttää parametrin FMOD ohjelmalle, vaihtaen aktiivisen musiikkikappaleen indeksin. Lisäsin Music Trigger -objekteihin myös vaihtoehdon käyttää Fade In- tai Fade Out -efektejä, jotka harjoittelijamme on tehnyt FMOD ohjelman puolella. Säädin myös pelimme Options valikoiden musiikki säätimet vaihtamaan parametreja FMOD ohjelman kautta, jonka kautta musiikki nyt soi. Asia saattaa kuulostaa yksinkertaiselta, mutta kyseiset toimenpiteet vaativat yllättävän monta mutkaa, koska asioita saatetaan ohjata muun muassa. UI-elementtien puolelta, Music Trigger -objekteista, tai Game Instance -objektista. Ratkaisut löytyivät kuitenkin laittamalla parametrien säädöt oikeisiin paikkoihin, ja tilanteisiin. Hyödyntämäni ratkaisut valitsin sen vuoksi, että pystyin hyödyntämään paljon jo entuudestaan luotua logiikkaa.

Sain työstämäni apina hahmon valmiiksi tältä erää. Hahmon peruslogiikka toimii nyt kuten Design dokumentissa on kirjoitettu. Pelaajan huomattuaan hahmo menee Alert tilaan, ja mikäli pelaaja saapuu määritetyn hyökkäysrajan sisälle, alkaa se heitellä objekteja pelaajan sijaintiin. Mikäli apina osuu pelaajaan heittämällään objektilla, kaikki ympärillä olevat apinat alkavat nauraa. Pelaajan kadotessa hahmon näkökentästä, se katselee ympärilleen hetken, jonka jälkeen palaa Idle tilaan. Apina hahmot on mahdollista liittää valittuihin puihin tekemäni Hit Detection -komponentin avulla, ja puuta lyödessä hahmo putoaa alas, ja menee hetkeksi Stunned tilaan. Mikäli pelaaja ei ehdi lyödä hahmoa ajoissa (lyönnillä voi listiä apinat), se hyppää takaisin puuhun pienen tauon jälkeen. Apina hahmoilta puuttuu vielä kaikki animaatiot, joten jouduin käyttämään paljon erilaisia Debug työkaluja, ja seuraamaan Behavior Tree -käytöspuun toimintaa tarkasti, saadakseni varmistettua oikeat ohjaus käyttäytymiset. Hahmon heittoradat sain laskettua hyödyntämällä kohteiden etäisyyttä ja “Suggest Projectile Velocity” -solmua (kuva 2).

Kuva 2. Suggest Projectile Velocity -solmu laskemassa heittorataa Longtail pelissä (Kuva: Hologram Monster).

Päädyin kyseisiin toteutus tapoihin siksi, koska hahmoilta puuttuu vielä animaatiot. Joudun tekemään hahmoihin vielä lisää hienosäätöä, kun saan animaatiot käyttööni. Silloin tulen hyödyntämään animaatioiden Animation Notify -tapahtumia mekaniikkojen sulavoittamiseen ja immersion luomiseen. Projektin päivitys uudempaan versioon onnistui, ja sain ratkaistua luomalla uudestaan projektin taustalla toimivat C++ tiedostot manuaalisesti Visual Studion kanssa (Z-Axis Up, 2019). Tällä kerralla en joutunut onneksi korjailemaan mahdollisia virheitä, joita projektin päivitys saattaa joissakin tiedostoissa aiheuttaa. Mikäli päätämme kokeilla projektin päivittämistä Unreal Engine 5 -versioon, on hyvin mahdollista, että eteen tulee korjattavia ja päivitettäviä asioita. Mahdolliset virheet tulee myös tuolloin korjata manuaalisesti. Ongelmien ratkaisemiseksi joudun tekemään selvitystyötä, ja tarkistamaan ratkaisuja mahdollisiin ilmeneviin virhekoodeihin. (Epic Games 2023)
Kaiken kaikkiaan viikko oli oikein onnistunut. Musiikin soittamisen integrointi FMOD ohjelman kautta auttaa äänipuolen harjoittelijaamme etenemään musiikkipuolen asioissa. Olin mielissäni saatuani apina hahmon tältä erää valmiiksi odottamaan tulevia animaatioita. Oli myös hyvin hyödyllinen ja mielenkiintoinen kokemus päivittää projekti uudempaan Unreal Engine -pelimoottorin versioon.

5.4 Seurantaviikko 4 (3.7.2023-7.7.2023)

MA 3.7.2023

Tämän päivän työtehtäviini kuuluu uuden kontrolloitavan hahmon luominen. Kyseessä on pieni possu, jota pelaaja voi tietyissä tilanteissa ohjata. Possuhahmosta on ollut olemassa Character- ja animaatiobluerintit, mutta sen Skeleton Rig -objekti on mennyt jossain vaiheessa sekaisin. Minun tarvitsee siis tehdä sekä Character- että animaatioblueprint uudestaan päivitetyn Skeleton Rig -objektin kanssa. Hahmoa on tarkoitus päästä myös ohjaamaan, joten se pitää voida ottaa haltuun, ja sitä tulee voida ohjata pelaajan syötteen avulla.

Päivän osaamistavoitteisiin kuuluu Character- ja animaatioblueprintien luominen hahmolle, hahmon haltuun otto sekä liikkeen mahdollistaminen.

Loin hahmolle uuden Character blueprintin, lisäsin sinne tarvittavia muuttujia, muutin kameran asetuksia ja loin uusia tapahtumia. Lisäsin hahmolle Possessed- ja Unpossessed -tapahtumat, jotta hahmo voidaan ottaa pelaajan haltuun. Lisäsin kyseisiin tapahtumiin Camera Blend -siirtymät, ja Player Controller -ohjaimen siirrot hahmojen välillä. Loin tarvittavat muuttujat hahmon liikettä varten, lisäsin uuden Event Graph -kaavio välilehden Input komentoja varten, ja lisäsin liikkeeseen liittyvät tapahtumat kyseiselle välilehdelle. Hahmoa on nyt mahdollista ohjata sekä ohjaimella, että hiirellä ja näppäimistöllä. Loin hahmolle myös hyppy tapahtuman. Hahmon perusominaisuudet ovat siis jo varsin hyvällä mallilla.

Loin possu hahmolle myös uuden animaatioblueprintin. Lisäsin muuttujia, lisäsin uuden animaatiotilakoneen, ja loin uuden animaatio Blend Spacen -siirtymän Idle-, Walk- ja Run -tiloille. Lisäsin animaatiotilakoneeseen tilat ja siirtymät liikkumiselle, hypylle ja toiselle Idle tilalle, joka tulee käyttöön, mikäli hahmo ei ole pelaajan kontrolloitavana, tai pelaaja ei anna mitään Input syötettä muutamaan sekuntiin. Hahmon animaatioblueprint on jo lähestulkoon valmis, niiltä osin mitä animaatioita minulla on tällä hetkellä käytössäni.

Päivän osaamistavoitteet onnistuivat hienosti, sillä sain possu hahmon jo lähes valmiiksi. Hahmo vaatii enää perus ominaisuuksiltaan pientä hienosäätöä. Hahmon Design puoli vaatii vielä hieman hiomista, ja siltä uupuu muutama animaatio, kuten puskeminen ja vahingon vastaanottaminen. Kyseisten animaatioiden ja Design puolen viimeistelyn jälkeen voin päivittää hahmon loppuun. Kehitin osaamistani erityisesti hahmon haltuun ottamisessa, ja hahmon ohjattavuuden hallinnassa.

TI 4.7.2023

Tämän päivän työtehtäviini kuuluu possu hahmon jatkaminen. Haltuun otto tapahtumaa tarvitsee vielä hienosäätää, sekä possulle on tarkoitus tehdä

Charge kyky. Kerättävien asioiden Group Actor -objekteja tarvitsee myös säätää FMOD äänien soittamista varten.

Possu hahmon Unpossessed tapahtuma sai aikaan sen AI-Controller -luokan “irtoamisen”, mikä aiheutti hahmon jäävän paikoilleen, kun sitä ei enää ohjata. Sain ongelman ratkaistua lisäämällä “Run Physics with No Controller” -vaihtoehdon päälle, jolloin fysiikka toimi myös ilman AI-Controller -luokkaa, mikä tapahtuu juuri kun hahmoa ei enää kontrolloida. Laitoin myös oletusarvona olevan AI-Controller -luokan palautumaan kontrolloinnin loputtua. Työstin hahmolle Spyro The Dragon -pelien henkistä Charge kykyä. Kyky on mahdollista aktivoida pitämällä määritettyä nappia pohjassa, ja lopettaa vapauttamalla kyseinen nappi. Charge kyvyn ollessa aktiivinen, kamera siirtyy hahmon taakse, ja sivulle liike kääntää sekä hahmon rotaatiota sekä kameran kulmaa. Kamera mukautuu hahmon eteenpäin akseliin kyvyn ollessa aktiivinen. Charge kyvyn aikana hahmo liikkuu eteenpäin nopeammin, ja kykyyn saatetaan myöhemmin lisätä puskemistyylinen ominaisuus.

Työstin myös kerättävien asioiden Group Actor -objekteja, jotta äänipuolen harjoittelijamme saa liitettyä niihin ääniä. Lisäsin objekteihin mahdollisuuden soittaa ääni silmukkaa, sekä tarkistuksen, joka tarkistaa onko kaikki Group Actor -objekteissa olevat kerättävät esineet kerätty, jonka jälkeen soiva äänisilmukka lopetetaan. Pelaajalle vahinkoa tekeviin Damage-komponentteihin lisäsin myös vaihtoehdon määritellä vahingon tyyppi (esimerkiksi tuli, kivi ja niin edelleen), jonka kautta on mahdollista soittaa lisää ääniä. Komponenttia käytetään esimerkiksi pelaajan osuessa liekkeihin.

Päivän osaamistavoitteet täyttyivät. Sain possu hahmon haltuun otto tapahtumat toimimaan paremmin, ja liikkeen jatkumaan sulavasti. Sain toteutettua myös Charge kyvyn, jota on mahdollista säätää myöhemmin, mikäli siihen halutaan esimerkiksi mahdollisuus tehdä vahinkoa. FMOD äänien soittamista varten tehdyt lisäykset kerättävien asioiden Group Actor -objekteihin onnistuivat myös hyvin. Osaamiseni kehittyi erityisesti haltuun otto tapahtumien hallitsemisessa.

KE 5.7.2023

Tämän päivän työtehtäviini kuuluu kerättävien asioiden Group Actor -objektien hienosäätöä, ja eilen ilmenneen bugin selvittämistä ja korjaamista. Mikäli aikaa jää, voin jatkaa possu hahmon työstämistä. Päivän osaamistavoitteisiin kuuluu bugien korjaamista, ja ominaisuuksien hienosäätöä ja parantelua.

Harjoittelijamme ilmoitti minulle, että kerättävien asioiden kanssa on ilmennyt bugi, jonka vuoksi Group Actor -objekteihin asetetut kerättävät asiat kerääntyvät monta kertaa. Aloin selvittämään mikä bugin voisi aiheuttaa, ja loppujen lopuksi huomasin, että objektit ovat jostain syystä ajaneet Construction Script -osion useaan kertaan, ja luonut duplikaatteja kerättävistä asioista täysin samoihin sijainteihin kuin alkuperäiset. Oli siis vaikeaa havaita, että kerättäviä olisi ollut useampi, koska ne olivat täysin päällekkäin. Kyseinen bugi on ilmennyt joskus aiemminkin, enkä löytänyt sille täysin selkeä syytä, miksi Construction Script -osio on ajettu useampaan kertaa. Ratkaisu ongelmalle oli poistaa duplikaatti versiot kerättävistä. Ne oli helppo paikallistaa, koska ne eivät olleen kiinnitettynä Group Actor -objektien Parent objektiin. Niiden manuaalinen poistaminen oli kuitenkin melko työlästä.

Huomasin myös, että vaikka kerättävien asioiden Group Actor -objektit pysäytettiin, kun kaikki objektin kerättävät on kerätty, ne lähtivät uudestaan käyntiin, mikäli Begin Play -solmu käynnistyi uudelleen, esimerkiksi pelitallennusta ladatessa, tai takaisin Level Streaming -alueelle saapuessa. Loin Group Actor -objekteille uuden makron, joka tarkistaa, ovatko sen Child Actor -objektit (eli kerättävät) jo kerätty. Mikäli kaikki on kerätty, objektin Tick funktiota ei käynnistetä. Tämä on pelin optimoimisen kannalta erittäin hyvä lisäys.

Päivän osaamistavoitteet täyttyivät kerättävien asioiden osalta, mutta niiden työstämisessä kului sen verran aikaa, että en ehtinyt enää työstää possu hahmoa. Kehitin osaamistani bugien paikallistamisessa ja korjaamisessa, sekä asioiden optimoinnissa.

TO 6.7.2023

Tämän päivän työtehtäviini kuuluu possu hahmon kykyjen laajentaminen. Päivän osaamistavoitteisiin kuuluu uuden kaivautumisominaisuuden kehittäminen.

Viimeistelin possu hahmon Charge kyvyn, ja sain kameran toimivaan sulavasti kyvyn aktivoimisen ja lopettamisen kanssa. Lisäsin myös hieman viivettä haltuun otto tapahtuman jälkeisille Input syötteille, jotta hahmon haltuun otto ehtii tehdä kaikki tarvittavat toimenpiteet. Possu hahmolle on tarkoitus tehdä kaivamiskyky, jolla voi kaivaa esineitä maan alta, sekä liikkua maan pinnan alla. Loin hahmolle uuden Input tapahtuman, jolla on mahdollista käynnistää kyky, ja joka etsii tiettyjä kaivamiseen liittyviä Actor -objekteja alapuoleltaan. Loin uuden “Digging Spot” Actor -objektin, ja liitin siihen jo olemassa olevan Item Spawner Actor -objektin. Kenttäsuunnittelija voi määrittää, mitä esineitä haluaa Digging Spot Actor -objektin luovan, ja kyseiset esineet ilmestyvät kaivamis kohdasta possu hahmon käyttäessä kaivamiskykyä kyseisen Digging Spot Actor -objektin päällä. Aloin myös hieman miettiä ratkaisuja maan pinnan alla liikkumiseen, eli niin sanottuun “Burrowing” kykyyn. Kokeilin miten Spline Mesh- tyyppinen -objekti toimisi Burrowing alueiden luomiseen. Loin Spline Mesh -objektilla tietyn näköisiä alueita, joissa kaivaminen olisi mahdollista. En ennättänyt kuitenkaan vielä työstää kyseistä ominaisuutta tuon pidemmälle.

Pidimme myös perinteisen torstain viikkopalaverin. Esittelin palaverissa tällä ja viime viikolla tekemäni työtehtävät. Viime viikolla palaveri jäi pitämättä, koska niin moni tiimin jäsen oli joko kesä- tai sairaslomalla. Palaverissa olikin siis paljon läpikäytävää.

Päivän osaamistavoitteet täyttyivät ihan kohtalaisesti. Olisin toivonut pääseväni pidemmälle Burrowing kyvyn kanssa, mutta muut työtehtävät veivät enemmän aikaa, mitä olin ajatellut. Osaamiseni kehittyi kuitenkin hieman uuden kaivamiskyvyn toiminnallisuuden suunnittelun ja toteuttamisen myötä.

PE 7.7.2023

Tämän päivän työtehtäviini kuuluu possu hahmon Burrowing kyvyn jatkaminen. Mikäli aikaa jää, voin päivittää apina hahmolle animaatiot, jotka ovat nyt valmiit. Päivän osaamistavoitteisiin kuuluu keksiä ratkaisu hahmon sijoittamiseen ja pitämiseen maan alla, sekä pinnan tunnistamiseen.

Tein possu hahmolle väliaikaisen animaation kaivamiselle pilkkomalla toista animaatiota, ja säätämällä sen aloituskohtaa ja nopeuta. Liitin animaation kiinni kaivamistapahtumaan. Totesin eilen kokeilemani Spline Mesh -objekti kokeilun olevan liian vaivalloinen kaivautumisalueiden käsittelyyn, joten päätin vaihtaa toteutustapaa. Toteutin pinnan alla kaivamisen raycast tarkistusten avulla, joka tunnistaa tietyn pinna Landscape Actor -objektista. Kun pinta on asetettu Mud materiaaliksi, ja pelaaja käyttää Burrowing kykyä, hahmo siirretään hieman maan alle Flying tilassa, ja sen Collision Channel -kanavat otetaan pois käytöstä. Samalla asetetaan kaivamiseen liitetty partikkeli efekti aktiiviseksi. Hahmon liikkuessa pinnan alla, raycast tarkistusta tehdään koko ajan, ja kun tunnistettu pinta ei ole enää Mud-tyyppiä, hahmo nousee automaattisesti takaisin maan pinnalle. jolloin Collision Channel -kanavat palautetaan, ja Movement Mode -solmun tila vaihdetaan takaisin Walking tilaan.

Siistin ja dokumentoin koodini possu hahmoon liittyen. Tämä pitää sisällään kaikki hahmoon itseensä liittyvät blueprintit, ja myös kaivamisen Digging- ja Burrowing -mekaniikkoihin liittyvät asiat.

Päivän osaamistavoitteet onnistuivat oikein hyvin, koska sain Burrowing kyvyn sen verran hyvin valmiiksi, että minun oli mahdollista siistiä ja dokumentoida tekemäni koodit, ja laittaa ne muille jakoon versionhallintaan. Kyseiset työtehtävät veivät kuitenkin sen verran aikaa, että en ehtinyt enää paneutua apina hahmon animaatioihin.

Seurantaviikko 4 – Viikkoraportti: 3.7.2023 – 7.7.2023

Tämän viikon aikana työtehtäviini kuului täysin uuden pelattavan hahmon luominen, ja hahmon kykyihin liittyvien asioiden ratkaiseminen. Tämän lisäksi tehtäviini kuului pienten laajennusten tekemistä kerättävien esineiden Actor Group -objekteihin, ja vahinkoa aiheuttaviin Actor Volume -objekteihin.

Uusi ohjattava hahmo oli pieni possu, jota pelaaja voi tietyissä tilanteissa ohjata. Loin hahmolle uuden Character- ja animaatioblueprintin, ja implementoin animaatiot hahmon animaatiotilakoneeseen. Tein Character blueprintiin tapahtumat hahmon haltuun ottamista, liikuttamista ja kykyjä varten. Tämä on hyvin perinteinen tapa luoda uusi hahmo peliin Unreal Engine -pelimoottorilla. (Epic Games 2023). Loin hahmolle hieman samantyylinen Charge kyvyn kuin Spyro The Dragon -peleissä, sekä kaivamis kyvyn, jolla on mahdollista kaivaa esiin maahan piilotettuja esineitä. Loin hahmolle myös Burrowing kyvyn, jolla se voi kaivautua maan alle, ja liikkua eteenpäin mullan alla.

Hahmon mekaniikkoja rakentaessa hankalin selvittämäni asia oli, miten saan hahmon kaivautumaan maan alle vain tietyillä alueilla. Aluksi lähdin kokeilemaan Spline Mesh- objekteja, joiden avulla olisin luonut eri muotoisia alueita, joiden sisällä hahmon on mahdollista kaivautua. Kyseinen tapa osoittautui kuitenkin melko vaivalloiseksi, ja alueiden määrittäminen oli varsin työlästä, joten päätin kokeilla muita ratkaisuja. Päädyin loppujen lopuksi käyttämään raycast tarkistuksia, joka tunnistaa Landscape työkalulla luodun maan pinnan, ja siihen asetetun materiaalin, ja mahdollistaa kaivautumisen määritetyllä materiaalilla. Olin tähän ratkaisuun äärimmäisen tyytyväinen, koska se helpottaa kenttäsuunnittelijan työtä huomattavasti, koska hän voi määrittää Landscape työkalulla helposti juuri halutun kokoisen alueen kaivautumiselle. Kyseinen ratkaisu tekee alueiden luomisen helppouden lisäksi myös itse mekaniikan hallitsemisesta huomattavasti helpompaa.

Valitsemani toimintamalli kaivautumismekaniikan toteuttamisessa oli mielestäni niin toimiva, että aion käyttää sitä tulevaisuudessa myös eräällä toiselle hahmolle, jolla tulee olemaan hieman samantyylinen kaivautumismekaniikka.

Olen melko varma, että käyttämäni toteutustapa pinnan tunnistamisen kanssa, tulee toimimaan myös AI hahmojen Pathfinding-metodien kanssa, jolloin pystyn hyödyntämään sitä myös hahmojen tekoälyn toteuttamisessa.

Eräs toinen työtehtäväni liittyi kerättävien asioiden Actor Group -objekteihin. Pelistämme löytyy kerättäviä kristalleja, jotka pyörivät ryhmissä asioiden ympärillä. Aiemmin tällä viikolla lisäämäni Hovering, eli leijumisäänen yhteydessä ilmennyttä bugia korjatessa huomasin, että kyseisiä kristallien Group Actor -objekteja ei ikinä pysäytetty, kun kaikki ryhmän kristallit olivat kerätty. Tämä siis johti täysin turhaan performanssia vievään liikkeeseen, joka ei ollut pelaajalle millään tavalla nähtävissä. Liitin aiemmin luotuun leijumisäänen pysäyttämisen hoitavaan funktioon myös itse ryhmien pysäyttämisen, sekä uuden makron, joka tarkistaa, ovatko ryhmän Child Actor -objektit (eli kerättävät kristallit) jo kerätty kun kyseinen Actor objekti luodaan uudestaan. Mikäli kaikki on kerätty, Group Actor -objektin Tick funktiota ei käynnistetä enää kentän latautuessa uudestaan joko pelitallennuksen tai Level Streaming -alueen ominaisuuden yhteydessä.

Olin mielissäni löytäessäni kyseisen ominaisuuden puuttumisen, koska pelin optimoiminen on äärimmäisen tärkeä osa-alue pelikehityksessä. Kaikki ylimääräistä laskentatehoa vievät prosessit tulisi pysäyttää, tai laittaa tauolle, kun niitä ei tarvita. Vaikka blueprintit ovat tehokkaita ja helppokäyttöisiä, ne voivat myös vaikuttaa merkittävästi suorituskykyyn, mikäli niitä ei optimoida oikein. Usein itseään päivittävät blueprintit voivat aiheuttaa merkittäviä suorituskykyongelmia, varsinkin jos ne on liitetty useaan näkymässä esiintyvään objektiin. (Carrington 2023). Olen tyytyväinen käyttämääni ratkaisuun, koska sen liittäminen jo olemassa olevaan pelin tallennusmekaniikkaan kävi helposti, joten minun ei tarvinnut käyttää sen toteuttamiseen kovin paljoa aikaa.

5.5 Seurantaviikko 5 (10.7.2023-14.7.2023)

MA 10.7.2023

Tämän päivän työtehtäviini kuuluu animaatioiden liittäminen apina hahmolle, ja hahmon ominaisuuksien säätäminen toimimaan yhdessä animaatioiden kanssa. Tarkoitus on myös implementoida pelin ääniefektit käyttämään FMOD ääniä yhdessä harjoittelijamme kanssa. Osaamistavoitteisiin kuuluu Character- ja -animaatioblueprintien sekä AI-Controller -luokan yhdistäminen toimivaksi kokonaisuudeksi.

Implementoin apina hahmon animaatioblueprintin käyttämään uusia animaatioita. Animaatiot itsensä vaihto oli varsin simppeliä, koska olin tehnyt Behavior Tree -käytöspuun valmiiksi jo aiemmin. Minun tarvitsi kuitenkin tehdä melko paljon hienosäätöä hahmon Character blueprintiin, jotta mekaniikat toimivat sulavasti animaatioiden kanssa. Hahmon heitto tapahtuma koki eniten muutoksia, koska heitto piti saada lähtemään oikealla hetkellä. Tein heittoanimaatiosta Animation Montage -animaation, ja lisäsin Animation Notify -tapahtumat sopiville avainkehyksille. Ensimmäinen Notify tapahtuma luo heitettävän objektin apina hahmon käteen, ja toinen Notify tapahtuma aloittaa itse heiton, eli kaaren laskemisen ja objektin irrottamisen hahmosta. Toteutus onnistui varsin hyvin. Minun tuli huomioida myös Animation Montage -animaation pysäyttäminen tarvittavissa kohdissa (esimerkiksi apinan ottaessa osuman Longtail hahmon iskusta), jotta heittämistä ei jatketa loppuun asti. Sain animaatioiden implementoinnin melkein valmiiksi, huomiselle jää vain hieman hienosäätöä ja eri tilanteiden tarkistelua.

Implementoin myös pelin ääniefektit käyttämään FMOD ääniä yhdessä harjoittelijamme kanssa. Pelin Options valikko säätää nyt Global Sound Mix -kanavien sijaan FMOD tapahtumien parametreja, joiden avulla on mahdollista säätää ääniefektien voimakkuustasoja. Kyseinen toteutus tuli tehtyä jo aiemmin pelin musiikeille, joten tällä kerralla prosessi oli varsin yksinkertainen.

Päivän osaamistavoitteet onnistuivat hyvin. Ääniefektien implementointi FMOD ohjelman puolelle onnistui, sain apina hahmon animaatiot päivitettyä ja melkein kokonaan hienosäädettyä toimimaan mekaniikkojen kanssa. Osaamiseni kehittyi entisestään animaatioiden ja hahmon mekaniikkojen implementoinnin saralla.

TI 11.7.2023

Tämän päivän työtehtäviini kuuluu animaatioiden implementoinnin viimeistely apina hahmolle, sekä pelin Ambience äänien vaihtaminen FMOD ääniksi. Päivän osaamistavoitteisiin kuuluu lisää animaatioiden käsittelyä, sekä FMOD tapahtumien ja komponenttien kanssa työskentelyä.

Loin uuden makron laskemaan jäljellä olevan ajan heittoanimaatiosta, ja apinan aloittamaan nauramistapahtuma vasta kyseisen animaation jälkeen. Viimeistelin käsittelyt myös heittämisanimaation lopettamiselle tietyissä tilanteissa. Loin animaatioblueprintiin uusia siirtymiä animaatiotilojen välille, jotta hahmo palaa oikeisiin animaatioihin, mikäli esimerkiksi heittämisanimaatio keskeytyy. Hienosäädin myös hieman hahmon Behavior Tree käytöspuuta, jotta hyökkäys tapahtuma on sulavampi. Kommentoin tekemäni muutokset ja siirsin päivitetyt animaatiot versionhallintaan.

Pelin Ambience äänet aiheuttivat jonkin verran päänvaivaa. Oikean parametrin välittäminen halutulle FMOD tapahtumille oli hieman haastava operaatio, mutta sain ongelman ratkaistua tekemällä Ambience Trigger -objektiin ja Game Instance -objektiin muuttujat FMOD audio komponentille, valitsemalla haluttu FMOD tapahtuma Ambience Trigger -objektissa, jolloin sain välitettyä sen parametrina Game Instance -objektissa olevalle audio komponentille. Tällä tavalla Ambience ääni soitetaan vasta Game Instance -objektin kautta, jolloin vältetään useamman Ambience äänen päällekkäin soittaminen.

Päivän osaamistavoitteet onnistuivat ihan kohtalaisesti. Apina hahmon suhteen kaikki sujui hyvin. Sain siihen liittyvät työtehtävät valmiiksi ja muutokset siirretty onnistuneesti versionhallintaan. Pelin Ambience äänet aiheuttivat jonkin verran

haasteita, koska FMOD tapahtumien oikeaoppinen käynnistäminen ja lopettaminen vaatii vielä jonkin verran miettimistä, ja kyseiseen asiaan meni paljon enemmän aikaa, mitä alun perin ajattelin. Tämä oli kuitenkin hyvin opettavainen kokemus, ja sain loppujen lopuksi äänet toimimaan oikein.

KE 12.7.2023

Tämän päivän työtehtäviini kuuluu possu hahmon Burrowing kyvyn hienosäätö. Osaamistavoitteisiin kuuluu Burrowing mekaniikan hienosäätäminen vastaamaan enemmän annettua referenssiä.

Toive oli saada Burrowing mekaniikasta sulavampi ja yhtenäisemmän tuntuinen. Referenssinä ja esimerkkinä toimi Super Lucky’s Tale -pelin kaivatumismekaniikka. Jaoin Burrowing tapahtumaa osiin, jolloin sen hallitseminen oli helpompaa, ja erilaisten väli tarkastusten tekeminen onnistui. Säädin possun tekemään pienen hypähdyksen kaivamis nappia painattaessa Mud materiaalin päällä. Mikäli laskeutuessa hahmon On Landed -tapahtuma tunnistaa materiaaliksi edelleen Mud materiaalin, suoritetaan maan alle kaivautuminen, jota hienosäädin myös toimimaan nopeammin. Liitin On Landed -tapahtumaan myös mahdollisuuden aloittaa kaivautuminen tekemällä hahmon perushyppy, ja painamalla ilmassa Dig näppäin pohjaan, jolloin laskeutuessa hahmo aloittaa kaivautumisen automaattisesti. Liitin tarkistukseen myös käsittelyn, mikäli hahmo on käyttämässä Charge kykyä, jolloin kyseinen kyky lopetetaan, ja siirrytään yhtä lailla automaattisesti kaivautumaan. Säädin näppäinkomennot toimiaan myös toisin päin, eli mikäli hahmo on kaivautumassa maan alla, ja pelaaja pitää Charge näppäintä pohjassa, aloitetaan Charge kyky automaattisesti kaivautuessa takaisin maan pinnalle. Muutin maan pinnalle siirtymistapahtumaa myös jonkin verran, ja se on toteutettu nyt Character Movement -komponentin Launch Character -solmulla, jolloin saan hahmon loikkaamaan takaisin pinnan alle ilman että pelaajan kontrollit lukkiutuvat, joka tekee siirtymästä todella sulavan tuntuisen. Asetin muutaman kyvyn uudelleen aktivoimiseen liittyvän Cooldown ajan käyttämällä Set Timer by Event -solmuja. Kyseisten käsittelyjen avulla pystyn estämään pelaajan käynnistämästä kykyjä uudelleen kesken kaiken näpäyttämällä

nappeja nopeasti. Kaiken kaikkiaan sain Burrowing kyvyn tuntumaan hyvin miellyttävältä, ja vastaamaan myös hyvin paljon annettua referenssi materiaalia.

Päivän osaamistavoitteet onnistuivat hienosti. Sain hienosäädettyä Burrowing kyvyn oikein sulavaksi, ja mukavan tuntuiseksi. Työtehtävien kanssa ei ilmennyt suurempia haasteita, vaan sain asiat toteutettua varsin sutjakasti.

TO 13.7.2023

Tämän päivän työtehtäviini kuuluu viimeistellä possu hahmo. Minulla on vielä muutama huomioitava ja lisättävä asia hahmon Burrowing kykyyn liittyen. Hahmon tulee voida liikkua objektien ali maan alla, ja sen liikettä maan alla tulee voida rajoittaa tarvittaessa. Minun täytyy keksiä myös ratkaisu hahmon animaatioiden Stretching, eli venymisongelmalle. Osaamistavoitteisiin kuuluu keksiä ratkaisu, joka mahdollistaa objektien ali kulkemisen maan alla, sekä hahmon animaatioiden Stretching venymisongelmaan tutustuminen ja sen rajoittaminen.

Minun tuli saada hahmo liikkumaan erilaisten objektien, kuten puunrunkojen tai muiden esteiden ali ollessaan kaivautuneena maan alla. Olin aiemmin toteuttanut maata tarkastelevan raycast tarkistuksen hahmon maanpinnan yläpuolelta. Tämä aiheutti objektien ali mentäessä ongelman, koska raycast tarkistus osui maan pinnalla olevaan objektiin, ja sai hahmon putoamaan maan läpi. Päätin toteuttaa Burrowing kyvyn raycast tarkistukset tulemaan maan pinnan alta kohti pintaa, jolloin se pysähtyi maan pintaan, eivätkä pinnalla olleet esteet häirinneet sitä. Tämän avulla sain hahmon liikkumaan esteiden ali ollessaan maan alla. Minun tuli kuitenkin huomioida tilanteet, joissa pelaaja saattoi vapauttaa kaivautumisnapin ollessaan esteen alla, jolloin hahmo saattoi nousta pinnan alta esteen “sisään”, ja jäädä jumiin. Tein uuden raycast tarkistuksen tarkistamaan onko maan pinnalla esteitä. Mikäli pinnalla on este, hahmo jatkaa matkaansa eteenpäin, kunnes maan pinnalle nousulle ei ole enää esteitä. Kyseinen tarkastus lopetetaan, mikäli pelaaja painaa kaivautumisnapin pohjaan uudelleen, ollessaan vielä maan alla. Näin ollen hahmoa ei nosteta

pinnalle automaattisesti, ja sulavuus Burrowing mekaniikan liikkeessä säilytetään (kuva 3).

Kuva 3. Possuhahmo kaivautumassa puunrungon ali Longtail pelissä (Kuva: Hologram Monster).

Minun tuli myös huomioida tilanteet, joissa pelaaja yrittää liikkua maan alla esteen ali sivusuunnassa, joka saattaisi aiheuttaa hahmon liikkumaan maan alla ei halutuille alueille. Ratkaisin ongelman laittamalla maan alle Blocking Volume Actor -objektin mukailemaan Mud materiaalin reunoja, jolloin hahmo liikkuu Mud materiaalin mukaisesti eteenpäin, ja estää hahmon liikkumisen alueen ulkopuolelle. Säädin myös Camera Boom -komponentin asetuksia, ja rajoitin hahmon kameran maksimi View Pitch -asetusta hahmon ollessa maan alla. Näillä muutoksilla sain varmistettua, että kamera ei törmää maan pinnalle oleviin objekteihin, ja että pelaaja ei voi kääntää kameraa täysin maan alle. Kommentoin tekemäni muutokset ja siirsin päivitetyt blueprintit versionhallintaan. Kuvasin myös viikko palaveria varten videoita tekemistäni työtehtävistä viikon ajalta. Pidimme viikkopalaverin iltapäivällä.

Toinen asia, jonka kanssa työskentelin, oli Stretching, eli venymisongelma hahmon animaatioiden kanssa. Unreal Engine -pelimoottori mahdollistaa Squash & Stretch- ominaisuudet animaatioiden kanssa. Tämä luo animaatioihin eloa, venyttämällä hahmon Mesh objektia mukailemaan animaatioiden liikettä. Ominaisuus sai kuitenkin meidän possuhahmomme silmät pullistumaan ulos

hahmon päästä sen hyppäämisanimaation lopussa. Jouduin selvittämään asiaa melko paljon. Aluksi kokeilin luoda Constraint rajoitus arvot silmän ja pään seudun luille, mutta ne eivät ratkaisseet ongelmaa. Loppujen lopuksi päädyin uudelleen kohdistamaan hiusten ja silmien luut hahmon Skeleton-objektiin, animaatioiden sijaan hahmon Skeleton Rig -objektissa. Vinkin tähän ratkaisuun löysin Epic Games Unreal Engine -foorumeilta (2022), jonka kautta päädyin lukemaan virallista dokumentaatiota animaatioiden uudelleen ohjauksesta.

Päivän osaamistavoitteen onnistuivat erinomaisesti. Ehdin toteuttaa päivän aikana paljon eri asioita, kuten laajentaa ja viimeistellä possu hahmon Burrowing kyvyn, ratkaista Stretching venymisongelman hahmon Skeleton Rig -objektissa, ja kuvata videoita tekemistäni työtehtävistä viikko palaveria varten. Osaamiseni kehittyi erityisesti Skeleton Rig -objektin uudelleen kohdistamisessa, joka oli minulle täysin uusi asia.

PE 14.7.2023

Tämän päivän työtehtäviini kuuluu äänipuolen harjoittelijamme auttamista, ja pienten korjailujen tekemistä. Harjoittelijamme oli lisännyt parin päivän aikana peliin paljon uusia ääniä, joista muutamassa esiintyi pieniä ongelmia. Päivän osaamistavoitteisiin kuuluu asioiden selvittelyä, harjoittelijan ohjeistamista, ja korjausten tekemistä.

Ensimmäisenä otin versionhallinnasta uudet muutokset ja tutustuin niihin. Ne sisälsivät paljon äänipuolen harjoittelijamme tekemiä lisäyksiä ääniin liittyviin asioihin. Niihin tutustuminen, ja toteutusten tarkistaminen sujui ongelmitta.

Muutama asia, joka vaati minulta huomiota äänipuolen asioihin liittyen, oli aiemmin tekemieni Signal Plant -vihollisten ohjaamien puunjuurien äänet, jotka soivat aina kun peli käynnistettiin. Tutkin asiaa, ja huomasin juurien kasvattamistapahtuma (ja näin ollen äänen soimisen kyseisestä tapahtumasta) käynnistyvän pelin alussa, koska Signal Plant -hahmot käynnistivät sen mennessään Idle tilaan. Korjasin ongelman muuttamalla kyseisen tapahtuman toimintoja hieman, jolloin se käynnistetään vasta kun Signal Plant -hahmo

menee takaisin Idle tilaan, oltuaan ensin jossain toisessa tilassa. Oli hyvä, että tämä asia tuli esille, koska korjasin samalla puun juurien Collision Channel -kanavien Default arvot oikeiksi. Seuraava asia oli Longtail hahmon erilaiset Idle animaatiot. Kyseiset animaatiot saattoivat alkaa pyörimään tilanteissa, joissa se ei ollut haluttua, kuten ratsastaessa laamalla tai Sandboard-objektilla. Loin Longtail hahmolle uuden muuttujan IsPossessingSomething, ja asetin sen käytäntöön hahmon ratsastaessa. Kyseinen muuttuja estää animaatioblueprintissä Idle animaatioiden pyörittämisen, ja korjaa esiintyneen ongelman. Toinen ongelman Idle animaatioiden kanssa, oli animaatioihin liitettyjen äänien jatkuminen taustalla, vaikka animaatio keskeytyisi. Loin uuden funktion, joka ajetaan aina Idle animaatioiden keskeytyessä. Funktio tarkistaa Idle animaatioihin sidotut äänet, ja lopettaa ne, mikäli soivat animaation keskeytyessä. Tein myös päivän aikana muita hyvin nopeita ja pieniä korjauksia asioihin siellä täällä, sekä vastailin äänipuolen harjoittelijamme kysymyksiin.

Päivän työtehtävät onnistuivat hyvin, koska sain ratkaistua kaikki äänien kanssa esiintyneet ongelmat. Osaamiseni kehittyi päivän aikana ongelmanratkaisu kyvyissä, joita sain harjoittaa erinäisten asioiden korjaamisen ja ratkaisemisen parissa.

Seurantaviikko 5 – Viikkoraportti: 10.7.2023-14.7.2023

Tämän viikon aikana työskentelin pelin ääniefektien ja Ambience äänien parissa, implementoin animaatiot apina hahmolle, säädin possu hahmon Burrowing mekaniikkaa ja ratkaisin Stretching venymisongelmat hahmon animaatioissa. Lisäsin myös ratkaisun Longtail hahmon Idle animaatioiden pyörimiselle ei halutuissa tilanteissa.

Pelin ääniefektit ja Ambience äänet tuli implementoida käytettäväksi FMOD ohjelman kautta. Pelin Options valikko säätää nyt FMOD tapahtumien parametreja aiemmin käytössä olleiden Global Sound Mix -kanavien sijaan. Minun piti selvittää oikeat parametrit FMOD tapahtumille, ja niiden ratkaiseminen sujui varsin ongelmitta, koska kyseinen toteutus tuli tehtyä jo aiemmin perin musiikeille. Olikin mukava huomata, että aiemmin tehdyn

toteutuksen kautta saatu osaaminen nopeutti tämän tehtävän implementoimista huomattavasti. Ääniefektien vaihtaminen FMOD ääniksi sujui siis varsin mainiosti.

Pelin Ambience äänien implementointi FMOD ohjelman kautta käytettäväksi aiheutti hieman enemmän päänvaivaa. Suurin ongelma oli usean FMOD äänen soiminen samaan aikaan, joten ne piti saada soimaan vain yhdestä paikasta, joka tässä tapauksessa oli Game Instance -objekti. Parametri välitetään Ambience Trigger -objektiin asetetun FMOD audio komponentin muuttujan arvon kautta Game Instance -objektissa olevan muuttujan arvoksi, ja laitetaan soimaan vasta siellä. Tarkistukset saman äänen soimisesta, ja uuden aloittamisesta tuli kuitenkin tehdä jo Ambience Trigger -objektissa. Huomaan edelleen, että FMOD parametrien ja tapahtumien käynnistäminen vaatii minulta enemmän huomiota, kuin muiden muuttujien kanssa työskentely. Olen kuitenkin huomannut osaamiseni kehittyneen FMOD parametrien välittämisen, ja niiden kanssa työskentelyn kanssa.

Apina hahmo oli saanut muutaman uuden animaation, ja minun oli määrä implementoida animaatiot hahmolle, ja säätää hahmon mekaniikat toimimaan niiden kanssa. Itse animaatioiden liittäminen oli simppeliä, koska olin luonut hahmon animaatiotilakoneen jo valmiiksi. Säädin hahmon heitto tapahtuman toimimaan yhdessä heittämisanimaatioihin liittämieni Animation Notify -tapahtumien kanssa. Hahmon käteen luodaan heitettävä esine, ja itse heitto tapahtuu animaatioiden oikeilla avainkehyksillä. Loin myös uuden makron laskemaan jäljellä olevan ajan heittoanimaatiosta, ja apinan aloittamaan nauramistapahtuma vasta kyseisen animaation jälkeen. Toisin kuin funktioissa, makroissa voi olla useampi kuin yksi ulostulon suorituspinni, joten on mahdollista luoda suoritusjärjestys, jossa eri lähtösuorituspinnit aktivoidaan graafilogiikan tuloksista riippuen. Lisäksi on mahdollista luoda makro, jossa ei ole suorituspinniä ja joka käsittelee vain tietoja, kunhan makron solmut eivät ole suoritussolmuja. (Epic Games 2023). Tällä tavoin sain varmistettua sulavan käyttäytymisen hahmon mekaniikkojen välillä makroja hyödyntämällä.

Erikoiset ongelmat possu hahmon Squash & Stretch -venymisominaisuuden kanssa sain ratkaistua tutkimalla virallista dokumentaatiota aiheesta. Animaatioiden uudelleenkohdistaminen on ominaisuus, jonka avulla animaatioita voidaan käyttää uudelleen hahmojen välillä, jotka käyttävät samaa Luurankoa, mutta joilla voi olla huomattavasti erilaiset mittasuhteet. Uudelleenkohdistamisen avulla on mahdollista estää animoituja luurankoja menettämästä mittasuhteitaan tai vääristymästä turhaan. (Epic Games 2023). En tässä tapauksessa käyttänyt eri kokoista hahmoa, mutta uudelleenkohdistuksella sain estettyä tiettyjen luiden venymisen ja vääristymisen.

Olin erityisten tyytyväinen tekemääni possu hahmon Burrowing mekaniikan laajennukseen, joka mahdollisti esteiden ali kulkemisen. Ratkaisin asian vaihtamalla raycast tarkistuksen tapahtumaan maan alla, lisäämällä uuden raycast tarkistuksen tarkistamaan vapaata tilaa maan pinnan päällä, ja ohjaamaan hahmon liikkeen tapahtumaan haluttuun suuntaan Blocking Volume -objektien avulla. Kyseinen toteutustapa mahdollisti sulavan liikkumisen mekaniikan kanssa, ja estää ongelmat, joita olisi voinut esiintyä maan pinnalle nousemisen yhteydessä, ilman että sitä edes välttämättä tiedostaa.

5.6 Seurantaviikko 6 (7.8.2023-11.8.2023)

MA 7.8.2023

Tämän päivän työtehtäviini kuuluu pienten ja nopeiden korjausten tekemistä. Osaamistavoitteisiin kuuluu bugien selvittämistä, ja toimintojen parantelua. Päivän työtehtävät ovat melko nopeita korjauksia, joten tämän päivän päiväkirjasta saattaa tulla hieman luettelomainen.

Pelaajan oli mahdollista tehdä erilaisia liikkeitä, kuten lyödä, hypätä, ja käyttää Dash- ja Interact -ominaisuuksia, sen jälkeen, kun Longtail hahmon elinvoima oli tippunut nollaan. Asetin jokaiselle edellä mainitulle ominaisuudelle tarkistuksen, että elinvoimaa on jäljellä enemmän kuin nolla. Lisäsin lyhyen

Cooldownin ajan lyömisen sallimiselle, sen jälkeen, kun pelaaja on ottanut vahinkoa. Tämä teki vahingon ottamisesta vaarallisempaa ja luontevampaa. Lisäsin myös käsittelyt lyöntien käsittelyyn, mikäli pelaaja ottaa vahinkoa lyömisen aikana. Erityisesti ilmassa tehtävään Ground Punch -kykyyn tämä käsittely oli todella tarpeellinen, jotta partikkeli efektit eivät jääneet päälle ei toivotulla tavalla. Longtail hahmon Ground Punch kyvyn kanssa oli myös ongelma, joka mahdollisti yläpuolella olevien pintojen läpi siirtymisen. Tämä johtui siitä, että Ground Punch -kykyä aloittaessa hahmoa nostetaan hieman ilmaan, joka saattoi aiheuttaa pelaajan siirtymisen kattojen läpi. Lisäsin Longtail hahmolle uuden makron, joka tarkistaa onko yläpuolella tyhjää tilaa. Mikäli tyhjää tilaa ei löydy, ei nostoa toteuteta, vaan animaatio suoritetaan paikallaan. Korjasin kameran lukkiutumisen pieneksi hetkeksi, Longtail hahmon käyttäessä Ledge Grab -kykyä. Lisäsin teleportteihin hahmon Gravity Scale -arvon palautumisen takaisin oletusarvoksi, lähinnä varmistukseksi. Lisäsin pelin Settings valikon SetHUDReferences makroon, joka luo referenssit pelin aikana käytettäviin HUD elementteihin, näin pystyin säätämään valikon avaamisen ja sulkemisen tehokkaammin, ilman että kyseinen valikko luotaisiin, ja tuhottaisiin joka kerran uudestaan. Lisäsin myös ominaisuuden avata Settings valikko myös muita hahmoja tai objekteja ohjattaessa, esimerkiksi ratsastaessa Laamalla, tai ohjatessa possu hahmoa.

Päivän osaamistavoitteet onnistuivat hyvin, ja suunnitellut tehtävät onnistuivat hienosti. Sain tehtyä paljon hyödyllisiä korjauksia, ja kitkettyä pois muutamia bugeja. Osaamiseni kehittyi bugien selvittämisessä, sekä erilaisten tilanteiden, kuten animaatioiden ja kykyjen keskeytymisen käsittelemisessä.

TI 8.8.2023

Tämän päivän työtehtäviini kuuluu selvittää, miksi Longtail hahmon lyönti-iskut eivät aina osu vihollisiin ja objekteihin. Mikäli aikaa jää, lisää työtehtäviä löytyy myös muista Combat aiheisista asioista. Päivän osaamistavoitteisiin kuuluu pelin Combat ominaisuuksien hienosäätäminen ja korjailu.

Tutkin Longtail hahmon lyömistapahtumaa, ja sain selville, että ohi menevät iskut ovat framerate, eli ruudunpäivitysnopeus -ongelma. Tämä tarkoittaa sitä, että mikäli peli pyörii hitaasti, on ruudunpäivitysnopeus hitaampi. Hidas ruudunpäivitysnopeus taas saa aikaan sen, että törmäystarkastelut iskujen kanssa tapahtuvat harvemmin, joka voi saada aikaan iskujen Hit Box -tarkistusten menevän ohi kohteistaan. Yritin korjata ongelmaa vaihtamalla Longtail hahmon iskujen raycast tarkastelut käyttämään Timer objekteja, Event Tick -solmun sijaan. Tätä toteutusta olin käyttänyt aiemmin muutaman vihollishahmon kanssa, ja sain niiden avulla aikaan luotettavampia ja tasaisempia tuloksia. Jostain syystä kyseinen ratkaisu ei kuitenkaan toiminut Longtail hahmon kanssa yhtä hyvin. Veikkaan sen johtuvan siitä, että hahmolla on paljon enemmän koko ajan tapahtuvia tarkasteluja, kuin vihollishahmoilla. Toisena ratkaisuja kokeilin liittää Longtail hahmon kaksi Sphere Collison -komponenttia, joiden kautta osuma tarkistukset tehtäisiin. Kyseinen toteutustapa toimi aluksi oikein hyvin, kunnes huomasin, että Overap tapahtumiin liitetty Sweep ominaisuus ei palauta mitään arvoja. Sweep ominaisuutta käytetään katsomaan osuman sijainti, jonka avulla soitetaan osuman ääni-, ja partikkeliefekti. Myöskään tämä toteutustapa ei loppujen lopuksi ratkaissut ongelmaa. Kyseisten toteutustapojen tekemiseen ja testailuun, sekä yleiseen asian selvittämiseen kului koko työpäiväni, joten en ehtinyt muiden työtehtävien pariin ollenkaan.

Tämän päivän osaamistavoitteet onnistuivat osittain, koska sain selvitettyä syyn ongelmalle. Toteutuspuoli ei kuitenkaan onnistunut, koska en saanut ongelmaa ratkaistua, ja minun tulee jatkaa kyseisen työtehtävän parissa huomenna. Osaamiseni kehittyi kuitenkin melko hyvin ruudunpäivitysnopeus asioita selvitellessä, ja erilaisten ratkaisujen kokeilemisessa.

KE 9.8.2023

Tämän päivän työtehtäviini kuuluu lyönti tapahtuman käsittelyn jatkamista, ja parantelua. Päivän osaamistavoitteisiin kuuluu ohjelmointityötä, ja eri toteutustapojen kokeilemista.

Päätin kokeilla “Did It Hit - Trace Detection Plugin” -komponenttia, joka on tarjolla Epic Games Market -kauppapaikalla. Kyseinen laajennus pitää sisällään erilaisia vaihtoehtoja osumien tarkasteluun. Asensin laajennuksen, ja yhdistin sen projektiimme. Osuma tarkasteluja varten loin uuden Static Mesh -objektin, ja lisäsin sille muutaman Socket sijainnin. Komponentti suorittaa osuma tarkastelut hyödyntämällä kyseisiä Socket sijainteja. Longtail hahmon käsien sisällä on piilossa pienet kynnet, jotka hyökkäyksen tehdessä skaalataan suuriksi, ja hyökkäyksen loputtua taas pieniksi. Liitin näkymättömät Static Mesh -objektit hahmon käsien Claw Socket -sijainteihin, joka skaalaa myös edellä mainitut kynnet. Lyönti hyökkäyksen tehdessä Did It Hit -komponentti suorittaa nyt osuma tarkistukset lyöntianimaatioiden mukaan, ja luo hyvin kattavat raycast tarkistukset lyöntien alueille. Säädin lyönti- ja osuma tapahtumaa melkoisesti, toimimaan hyvin yhdessä animaatioiden kanssa. Sain osumia paranneltua todella paljon, mutta ne vaativat vielä hieman hienosäätöä animaatioblueprintin puolelta.

Huomasin myös, että Longtail hahmon blueprintissä oleva osumien käsittely funktiossa olevat muuttujat saattavat muuttua matkan varrella, mikäli funktio ajetaan monta kertaa pienen ajan sisällä. Tämä tarkoittaa siis useaan kohteeseen osumista samaan aikaan. Päätin luoda uuden Actor -objektin, joka luodaan osuman tapahtuessa, sisältää Longtail hahmon blueprintissä olleen osuman käsittely funktion ja suorittaa sen, jonka jälkeen tuhoaa itsensä. Näin ollen kyseisestä funktiosta voi olla useampi instanssi kerrallaan, jolloin jokainen osuma käsitellään yksilöllisesti. Kyseinen toimintatapa toimii oikein hyvin, tosin sitä kannattaa vielä myöhemmin optimoida, esimerkiksi Pooling tekniikkaa käyttämällä. Object Pooling eli objektin varastointi on luova suunnittelumalli, joka esittelee ennen pelaamisen aloittamista kaikki objektit, joita tarvitaan minä tahansa hetkenä. Tämä poistaa tarpeen luoda uusia objekteja, tai tuhota vanhoja, pelin ollessa käynnissä. Objektivarastoja käytetään ensisijaisesti suorituskyvyn parantamiseen. Joissakin olosuhteissa objektivarastot parantavat merkittävästi suorituskykyä, kun projekti luo ja tuhoaa samaa peliobjektia toistuvasti nopeasti peräkkäin. Se toimii luomalla tietty määrä peliobjekteja ennen pelin suoritusaikaa, ja yksinkertaisesti epäaktivoi, tai aktivoi tehokkaasti

kierrättämällä vaaditut peliobjektit, tuhoamatta sitä missään vaiheessa. (Unity Technologies. 2019)

Päivän osaamistavoitteet onnistuivat oikein hyvin. Sain paranneltua lyöntien osumia huomattavasti, ja kehitettyä toimivan tavan käsitellä lyöntien osumat. Osaamiseni kehittyi ulkoisen laajennuksen hyödyntämisen käyttöönotossa ja implementoinnissa.

TO 10.8.2023

Tämän päivän työtehtäviini kuuluu lyöntitapahtuman käsittelyn jatkamista, ja parantelua, sekä toivon mukaan saada se valmiiksi. Päivän osaamistavoitteisiin kuuluu ohjelmointia mekaniikkojen hienosäätämisen parissa.

Säädin Longtail hahmon animaatioblueprintin kautta jokaisen lyönnin käynnistämään ja lopettamaan halutut raycast tarkistukset tietyillä animaation animaatiokehyksillä, hyödyntämällä Animation Notify -tapahtumia. Koska hahmon molemmissa käsissä on osuma komponentit, tulee niiden käynnistämistä ja lopettamista ajoittaa tiettyjen lyöntien ja animaatioiden mukaan. Longtail hahmolla on kolmen iskun hyökkäyssarja, jonka iskut tapahtuvat seuraavasti: oikea käsi, vasen käsi, molemmat kädet. Animation Notify -tapahtumien avulla sain lyönti tapahtumasta oikein sulavan, ja osuma tarkistukset toimimaan halutulla tavalla, käyttäen tätä menetelmää. Varmistaakseni iskujen osumisen, tein myös käsissä olevien osuma komponenttien lisäksi suoraa hahmon eteen tapahtuvat BoxTrace raycast -tarkistukset, jotka suoritetaan juuri kun käsissä olevien komponenttien tarkistukset lopetetaan. Näin pystyn varmistamaan, että mikäli kohde on täysin hahmon edessä (esimerkiksi lyödessä jotain kapeaa kohdetta, kuten aitaa), osuvat iskut varmasti myös niihin. Osuma tapahtuman testaamisen jälkeen totesin sen olevan nyt huomattavasti luotettavampi, ja toimivan oikein hyvin.

Päivän työtehtävät ja osaamistavoitteet onnistuivat hyvin. Sain viimeisteltyä lyönti tapahtumaan tehdyt osuma tarkistukset, ja paranneltua lyöntien

osumatarkkuutta huomattavasti. Osaamiseni kehittyi raycast tarkistusten käsittelyssä, ja niiden hallitsemisessa yhdessä animaatioiden kanssa.

PE 11.8.2023

Tämän päivän työtehtäviini kuuluu yleisiä pelin Combat aiheisiin liittyviä työtehtäviä, joita on ilmennyt kesälomani aikana. Työtehtäviini kuuluu siis paljon ohjelmointityötä. Päivän osaamistavoitteisiin kuuluu pelin Combat ominaisuuksien hienosäätämistä ja korjailua.

Tein päivän aikana useita muutoksia eri asioihin pelin hyökkäyssarjoihin asioihin liittyen, jotka luettelen seuraavaksi. Säädin Longtail hahmon rotaation kääntymään annetun analogi syötteen suuntaan lyönti tapahtuman käynnistyessä. Tämä mahdollistaa lyöntien sulavamman suorittamisen. Lisäsin Longtail hahmon kierähdysanimaatioon hieman iFrameja (Invincibility Frames, eli kuolemattomuus kehyksiä). Hyödynsin jälleen Animation Notify -tapahtumia, joka säätää Longtail hahmon Invincible muuttujan arvoa, joka määrittelee, ottaako Longtail vahinkoa kohdistuneista iskuista. Uuteen Damage Handler objektiin lisäsin validointi tarkistukset osuttuihin Actor -objekteihin ja komponentteihin, jolloin vältetään turhat virheilmoitukset. Säädin myös Longtail hahmon lyöntisarjan viimeisen iskun tekemään vahinkoa 2-osumapistettä yhden sijaan. Hyödynsin tässä jo aiemmin implementoimaani lyöntien indeksointia. Säädin Idler-, ja Swarmer Mouse -vihollis hahmojen Stunned tilan aikaa pidemmäksi, ja vastaavasti Shake Mesh -tapahtuman (joka tärisyttää hahmon Mesh- komponenttia nopeasti) täsmäämään Stun tilan ajan kanssa. Minun tuli myös käsitellä tilanne, jolloin Shake Mesh -tapahtuma oli jo käynnissä sen käynnistyessä, koska aikaikkuna on nyt sen verran suurempi, että se on mahdollista toteuttaa.

Päivän työtehtävät sujuivat hienosti, ja osaamistavoitteisiin päästiin. Vaikka kyseiset työtehtävät olivat minulle jo varsin tuttuja, karttui osaamiseni niiden hoitamisessa aiempaa tehokkaammin.

Seurantaviikko 6 – Viikkoraportti: 7.8.2023 – 11.8.2023

Tämän viikon teemana oli hyvin vahvasti pelin Combat asioiden tarkastelu ja hienosäätäminen. Yksi suurin havainto, jonka viikon aikana tein, oli miten paljon hidas ruudunpäivitysnopeus voi vaikuttaa peliin luotuihin toimintoihin. Meidän tapauksessamme se tarkoitti Longtail hahmon lyöntien osuvan ohi kohteistaan. Tämä johtui siitä, että raycast tarkistusten piirtoja suoritettiin liian vähän suhteessa animaatioiden nopeuteen, joka sai aikaan sen, että lyöntien kohteet eivät ikinä osuneet kyseisten tarkistusten alueelle.

Olin törmännyt kyseiseen ongelmaan jo aiemmin vihollishahmojen hyökkäysten raycast tarkistusten kanssa. Niiden kohdassa sain ratkaistua ongelman käyttämällä Custom Timer -ajastinta Event Tick -solmun sijaan. Custom Timer -ajastin aloitettiin ja lopetettiin hahmojen hyökkäysanimaatioiden kautta. Vihollishahmojen kanssa Custom Timer -ajastin toimi oikein hyvin, ja raycast tarkistusten piirrot olivat tasaisia. Jostain syystä sama ratkaisu ei kuitenkaan toiminut Longtail hahmon kanssa, ja raycast tarkistukset olivat edelleen todella katkonaisia hitaalla ruudunpäivitys nopeudella. Syytä kyseiselle ongelmalle en löytänyt, mutta epäilen sen johtuvan siitä, että Longtail -hahmon blueprint on yksi pelin laajimmista blueprinteistä, ja näin ollen sisältää paljon samaan aikaan tapahtuvaa toimintaa.

Sain ratkaistua ongelman hyödyntämällä ulkoista Plugin komponenttia, joka suorittaa useita eri raycast tarkistuksia Static Mesh -objekteihin asetettujen Socket sijaintien mukaan. Raycast tarkistukset suoritetaan C++ koodin avulla, joka tekee niistä nopeampia suorittaa, kuin blueprinteissä tapahtuvat raycast tarkistukset. (Epic Games 2023). Sain siis luotua kattavamman alueen osuma tarkistuksia, ilman että pelin suorituskyky kärsii. Liitin myös kyseisten raycast tarkistusten aloittamisen ja lopettamisen tietyille animaatio avainkehyksille, jolloin niitä ei suoriteta tapahtuvaksi toistuvasti Event Tick -funtiossa.

Valitsemani toimintamalli, eli ulkoisen Plugin komponentin hyödyntäminen oli tässä tapauksessa oikein toimiva tapa, vaikkakin minulla on ulkoisten laajennusten käyttämisestä hieman ristiriitaiset ajatukset. Toisaalta haluaisin

tehdä kaiken alusta asti itse, mutta mikäli ulkoinen laajennus tai esimerkiksi asset-ominaisuuspaketti nopeuttaa työskentelyä huomattavasti, on sellaisen käyttö järkevää, mikäli implementointi omaan projektiin onnistuu ongelmitta.

5.7 Seurantaviikko 7 (14.8.2023-18.8.2023)

MA 14.8.2023

Tämän päivän työtehtäviini kuuluu hienosäätää Longtail hahmon lyönti tapahtumaa lisää. Päivän osaamistavoitteisiin kuuluu erilaisten ratkaisujen kehittäminen ohjelmointi tehtävissä hahmon lyönteihin liittyen.

Yksi työtehtäväni oli lisätä Longtail hahmon lyönteihin Hit Stop -mekaniikka. Lisäsin pelihahmojemme yläluokkaan CharacterWithHealth uudet funktiot StartHitStop ja EndHitStop, jolloin ne periytyvät kaikille vihollishahmoille, kuin myös Longtail hahmolle. StartHitStop asettaa hahmon Custom Time Dilation -arvon nollaan, ja Set Timer By Function Name -solmun avulla kutsuu EndHitStop funktiota, joka ei tee muuta, kuin palauttaa Custom Time Dilation -muuttujan arvon takaisin ykköseksi. Set Timer By Function Name -solmulle välitetään parametrina desimaaliluku, joka toimii Hit Stopin -mekaniikan pituutena. Lisäsin lluomani Damage Handler -objektin funktion loppuun kutsut StartHitStop- ja EndHitStop -funktioille, sekä Longtail hahmolle, että iskun vastaanottajalle. Perustoteutus Hit Stop -mekaniikalle on yksinkertaisimmillaan siinä. Hit Stop -mekaniikka voi vielä hienosäätää, lisäämällä esimerkiksi hahmojen Mesh komponenteille tärinäefektiä. Säädin myös Longtail hahmon lyömiseen liittyvää raycast tarkistusta, joka etsii Pawn tyyppisiä kohteita. Aiemmin kyseinen raycast tarkistus tapahtui suoraa hahmon eteen, mutta lyönti suunnan nopean muuttamisen mahdollistaminen pelaajan inputilla, vaati että hienosäätöä raycast tarkistuksen suunnan arvoihin. Lisäämällä annetun syötteen suunnan pelaajan sijaintiin ja Forward Vector -arvoon, sain laskettua halutun suunnan nopeasti. Hienosäädin lisäksi vielä Longtail hahmon jo olemassa olevaa Aim Assist -mekaniikkaa hieman tarkemmaksi ja

sulavammaksi Line Tracen -raycast tarkistuksen ja etäisyyden mittaamisen avulla.

Päivän työtehtävät onnistuivat hyvin, ja osaamistavoitteisiin päästiin. Kyseisiin työtehtäviin kului kuitenkin huomattavasti enemmän aikaa, kuin etukäteen ajattelin. Tämä johtui siitä, että Longtail hahmon lyönti tapahtumassa on jo paljon valmista koodia, ja sen muokkaaminen ja testaaminen jälkikäteen on melko työlästä. Osaamiseni kehittyi muun muassa toimivan Hit Stop -mekaniikan kehittämisellä.

TI 15.8.2023

Tämän päivän työtehtäviini kuuluu hienosäätää Longtail hahmon lyönti tapahtumaa lisää, sekä säätää vihollishahmojen osuman ottamistapahtumaa, ja animaatioita. Päivän osaamistavoitteisiin kuuluu mekaniikkojen hienosäätämistä, sekä animaatioblueprintien kanssa työskentelyä.

Lisäsin Damage Handler -objektin vahingon välittävään funktioon välitettäviksi parametreiksi Longtail hahmon Forward Vector -arvon, ja osuman Impac tPoint -vector arvon. Lisäsin myös Ground Punch -kyvyn vahingon välittävään funktioon Impact Point -arvon. Muokkasin kahden vihollishahmon vahingon vastaanottamistapahtuman suorittamaan lyönneistä aiheutuvan sysäyksen funktion parametrina välitettävän vector arvon mukaan. Laitoin myös hahmojen rotaatio arvon kääntymään kohti Longtail hahmoa näiden elinvoiman tippuessa nollaan. Käytin kyseistä menettelyä siksi, että tällä hetkellä vain kuolettavan iskun animaatioissa on käytössä Root Motion -ominaisuus, jolloin hahmot liikkuvat animaation mukaan. Tavallisia iskuja vastaanottavissa animaatioissa ei ole käytössä Root Motion -asetusta, jolloin sysäys tarvitsee suorittaa ohjelmoimalla. Lisäsin myös Ground Punch -kykyyn aiemmin luomani Hit Stop -mekaniikan. Aloin myös säätää Swarmer Mouse -vihollis hahmon animaatioblueprintiä, koska minun tarvitsee saada ketjutettua vahingon vastaanottamis -animaatioita peräkanaa, mikäli hahmo vastaanottaa useamman osuman peräkkäin. Kyseinen työtehtävä jäi kuitenkin hieman kesken.

Päivän osaamistavoitteet onnistuivat hyvin, ja Combat aiheiset työtehtävät etenivät mukavasti. Osaamiseni kehittyi vahingon tuottamisen- ja vastaanottamisen -hallitsemisessa, ja haluttujen parametrien välittämisessä funktioiden kautta.

KE 16.8.2023

Tämän päivän työtehtäviini kuuluu jatkaa eilen keskeneräiseksi jääneitä työtehtäviä, eli säätää vihollishahmojen osuman ottamistapahtumaa, ja animaatioita. Päivän osaamistavoitteisiin kuuluu animaatioblueprintien laajentaminen, ja sulavien animaatio siirtymien mahdollistaminen.

Sain viimeisteltyä Swarmer Mouse -vihollishahmon animaati blueprintin animaatiotilakoneen muutokset. Hyödynsin animaatioiden ketjuttamisessa Conduit solmuja. Asetin hahmon animaatiotilan siirtymät hahmon ottaessa vahinkoa, siirtymään ensin Conduit solmuun (muuttujina enum Action State = Damage, ja boolean IsDead = Not), jonka kautta siirrytään lyöntien indeksien perusteella oikeaan Light Damage -animaatioon. Conduit solmu on säädetty toimimaan molempiin suuntiin, joten eri Light Damage -animaatioiden välillä siirtyminen onnistuu sulavasti. Lisäsin myös Conduit solmut kaikista Light Damage -animaatioista Conduit Idle- ja Conduit Noticed -solmuihin, joiden kautta siirtymät onnistuvat kyseisiin animaatio tiloihin sujuvasti. Kyseiset Conduit solmut mahdollistavat useita eri vaihtoehtoja, kun taas tavalliset tilasiirtymät ottavat vastaan vain yhden ehdon. Ne myös pitävät animaatiotilakoneen siistimpänä. Tein vastaavan uudelleen työstön myös Idler vihollishahmolle. Molempien hahmojen animaatiot ovat myös paljon sulavampia, koska esimerkiksi Longtail hahmon kolmen lyönnin lyöntisarjassa jokaiseen iskuun reagoidaan omalla animaatiolla. Tämän lisäksi hienosäädin ja parantelin vielä Ground Punch -kyvyn aiheuttamia sysäyksiä, koska kyseinen kyky tuottaa niin sanottua Area Of Effect -tyylistä vahinkoa, jolloin sysäyksen suunnan laskeminen tuli suorittaa hieman eri tavalla, kuin tavallisten lyöntien kanssa.

Päivän osaamistavoitteet onnistuivat oikein hyvin. Sain molempien vihollishahmojen animaatioblueprintien uudelleentyöstöt valmiiksi, ja animaatioiden siirtymät toimimaan oikein. Osaamiseni kehittyi Conduit solmujen hyödyntämisessä animaatiotilakoneen kanssa.

TO 17.8.2023

Tämän päivän työtehtäviini kuuluu luoda jonkinlainen Input Buffer -järjestelmä, jonka avulla voidaan lisätä pelaajan antamia syötekomentoja talteen, ja suorittaa niitä halutuissa tilanteissa. Päivän osaamistavoitteisiin kuuluu pelaajan syötekomentojen lukeminen, niiden tallentaminen ja hyödyntäminen.

Lähdin lähestymään Input Buffer -järjestelmän tekoa komponenttina, jotta se olisi mahdollista liittää tarvittaessa eri objekteihin. Kyseinen järjestelmä tulee olemaan meidän projektissamme varsin simppeli, koska pelissä ei esiinny esimerkiksi taistelu peleistä tuttuja erikoisliikkeitä, jotka vaativat mutkikkaita näppäinkomentoja. Tarkoitus on esimerkiksi voida painaa lyönti näppäintä hieman aiemmin, kuin lyöntianimaatio on päättynyt, jonka jälkeen Longtail suorittaa iskun “automaattisesti” annetun lyönti Buffer ominaisuuden kautta. Lisäsin komponenttiin uuden tapahtuman, joka käynnistää Timer ajastimen. Timer ajastin lopettaa itsensä annetun ajan kuluttua, jonka ajaksi annoin tässä tapauksessa 0,3 sekuntia. Kyseinen tapahtuma suoritetaan pelaajan painaessa lyöntinäppäintä, ja mikäli pelaajan painaa kyseistä nappia heti uudestaan, aloitetaan Timer ajastin aina alusta. Lyöntianimaation päättyessä tarkistetaan, onko kyseinen Timer ajastin aktiivinen, ja mikäli on, suoritetaan lyönti tapahtuma “automaattisesti” uudestaan. Pelaajalla on siis tässä tapauksessa käynnistää seuraava lyönti 0,3 sekuntia aiemmin, kuin edellinen lyönti loppuu. Tämä antaa mukavasti vapautta ja sulavuutta hahmon liikkeisiin. Komponenttia on mahdollista laajentaa eri syötekomennoille, esimerkiksi hyppyyn tai muihin hahmon kykyihin, mikäli tarve sellaiselle ilmenee. Tämän lisäksi selvittelin syitä eräälle lyönti aiheiselle bugille, joka saa lyönnit lähtemään joskus liian aikaisin. Kuvasin myös hieman materiaalia viikon aikana tekemistäni työtehtävistä viikko palaveria varten, jonka pidimme perinteisesti kello 14.

Päivän osaamistavoitteet onnistuivat hyvin. Sain toteutettua varsin simppelin, mutta toimivan Input Buffer -järjestelmän, jonka avulla pelaajan voi antaa eri syötekomentoja varastoon, ja joita on mahdollista hyödyntää halutuissa tilanteissa. Osaamiseni kehittyi komponenttien hyödyntämisessä syötekomentojen tukena.

PE 18.8.2023

Tämän päivän työtehtäviini kuuluu tutustua Unreal Engine -pelimoottorin tarjoamaan Control Rig -työkaluun. Päivän osaamistavoitteisiin kuuluu osaamiseni laajentaminen Control Rig -työkalun käytössä.

Luin Unreal Engine -pelimoottorin tarjoamia virallisia dokumentteja, ja katsoin tutoriaali videoita Control Rig -työkalun toiminnasta, käytöstä ja hyödyntämisestä. Tutustuin myös YouTube sivustolta löytämiini tutoriaali videoihin aiheesta. Sain hyvin pintapuolisen perustiedon Control Rig -työkalun toiminnasta, ja kokeilin sen käyttöä hieman. Lisäsin muutamia New Control -solmuja hahmon Rig Hierarchy -osioon, liitin hahmon käden luut kyseisiin kontrolli pisteisiin, jonka avulla pystyin liikuttamaan hahmon kättä haluamiini suuntiin. Kokeilin myös hieman, miten Full Body Inverse Kinematic (FBIK), liikuttaa valittuna kontrolli pisteitä suhteessa hahmon koko luurankoon. Minulla on vielä paljon opittavaa Control Rig -työkalun toiminnasta, mutta saada tämän päivän aikana jo hieman perustietoa sen toiminnasta.

Päivän osaamistavoitteiden mukaan sain kartutettua osaamistani Control Rig -työkalun käyttämisessä. Vielä on kuitenkin paljon opittavaa kyseiseen työkaluun liittyen.

Seurantaviikko 7 – Viikkoraportti: 14.8.2023 – 18.8.2023

Tällä viikolla työskentelyssä esiintyi seuraavanlaisia teemoja: Longtail hahmon lyömistapahtuman hienosäätäminen, vihollishahmojen vahingon vastaanottamistapahtuman hienosäätäminen, animaatioblueprintien

laajentaminen ja hienosäätäminen, Input Buffer -systeemin luominen ja Control Rig -työkaluun tutustuminen.

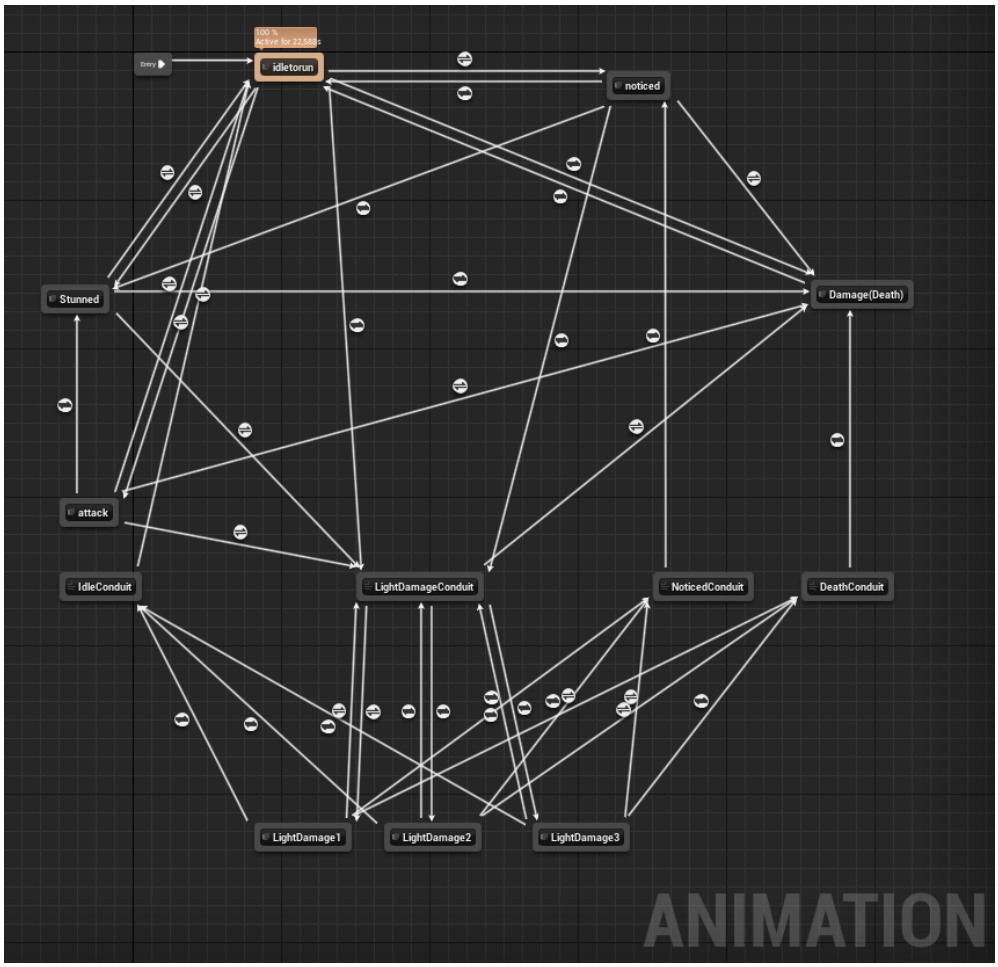
Suurimman osan viikosta työskentelin Longtail hahmon iskuista aiheutuvan vahingon ja reaktioiden aiheuttamiseen ja vastaanottamiseen. Aiemmin hahmon lyödessä, suoritettiin hahmon eteen ympyrän muotoinen raycast tarkistus, jonka ensimmäinen tulos valittiin hyökkäyksen kohteeksi. Valitun kohteen suuntaan annettiin Input Vector -arvoa, joka työnsi hahmoa kohdetta kohti. Tämä toteutus aiheutti välillä hieman erikoisia tuloksia, koska ensimmäinen tulos ei välttämättä aina ollut täysin paras mahdollinen vaihtoehto. Esimerkiksi mikäli Longtail on jo suorittamassa lyönti sarjaansa kohti vihollista, mutta toinen hahmo tekeekin nopea liikkeen Longtail hahmon ja aiemman kohteen vierestä tai välistä, saattoi Longtail kääntää rotaationsa nopeasti uutta kohdetta kohti, ja liikkua tasaisesti kohti uutta kohdetta, joka saattoi aiheuttaa dis orientaation tunnetta hahmon liikkumisen tuntumaan. Ratkaisin ongelman hienosäätämällä lyönti tapahtuman aloittamista seuraavanlaisesti: raycast tarkistus valitsee nyt saaduista tuloksista Longtail hahmoa lähimmän kohteen, suorittaa kyseistä kohdetta kohti Line Trace raycast -tarkistuksen, kääntää Longtail hahmon rotaation Line Tracen raycast -tarkistuksen osumakohtaa kohti, ja antaa Input Vector -arvoa kyseistä osumakohtaa kohti, kunnes kohde saavutetaan, tai lyönti tapahtuma päättyy. Tämä sai lyönneistä luotettavampia kohteiden valitsemisessa, ja esti hahmon yhtäkkiset nopeat rotaatio muutokset kesken lyömisen.

Vihollishahmojen iskujen vastaanottaminen vaati myös hienosäätöä. Aiemmin Longtail hahmon iskujen aiheuttama sysäys vihollishahmoihin tehtiin iskun Impact Location -arvon suuntaan. Tässä toteutuksessa ongelmana oli se, että iskun Impact Location -arvo saattoi sijaita myös hieman hahmon sivulla, koska lyönnin raycast tarkitus suoritettiin Longtail hahmon käsien mukaisesti. Päädyin antamaan sysäyksen Longtail hahmon Forward Vector -arvon mukaan, jolloin vastaanottavaa hahmoa työnnetään aina Longtail hahmosta poispäin. Tämä toteutus toimi hyvin animaatioissa, joissa ei ole mukana Root Motion -ominaisuutta, ja sysäys tarvitsee antaa matemaattisten arvojen perusteella. Vihollishahmojen vastaanottaessa kuolettava isku, nämä lentävät taaksepäin animaatioilla, joissa on käytetty Root Motion -ominaisuutta. Tällöin hahmoja ei

ole tarve sysätä mihinkään suuntaan koodin avulla, vaan animaatioiden Root Motion -ominaisuus hoitaa taaksepäin tapahtuvan liikkeen.

Yksinkertaisesti sanottuna Root Motion on hahmon liike, joka perustuu animaatioon luurangon juuriluusta. Useimmat pelin sisäiset animaatiot käsitellään jaksoilla, joissa hahmon juuri pysyy paikallaan. Näin ei kuitenkaan aina voi olla. Tämän käsittelemiseksi täytyy ottaa juuren liike pois hahmosta ja soveltaa sitä sen sijaan hahmon kapseliin. Tämä on Root Motion -ominaisuuden ydin Unreal Engine -pelimoottorissa. (Epic Games 2023)

Työskentelin myös animaatioblueprintien kanssa, ja tein hahmojen vahingon ottamisesta visuaalisesti sulavampaa. Koska Longtail hahmon iskusarja on mahdollista suorittaa melko nopeasti, tuli vihollishahmojen voida reagoida jokaiseen eri osumaan nopeasti. Hyödynsin animaatioblueprintin animaatiotilakoneessa Conduit solmuja, joiden avulla sain ohjattua animaatiotilakoneen toimintaa vahingon ottamis animaatioista vahingon ottamisanimaatioon. Conduit -solmut tarkistavat kyseisen iskun indeksin, ja ohjaa animaatiotilakoneen indeksin avulla tiettyyn vahingon ottamisanimaatioon (kuva 4).



Kuva 4. Vihollishahmon animaatioblueprintin Locomotion graafi (Kuva: Hologram Monster).

Tällä tavalla sain pidettyä animaatiot sulavana, ja kontrolloitua animaatiotilakoneen toimintaa selkeästi. Olisin voinut myös toteuttaa asian käyttämällä Animation Montage -animaatioita vaihtoehtona, joka olisi myös ollut toimiva tapa. Halusin kuitenkin pitää animaatiot pyörimässä täysin hahmon animaatiotilakoneessa, jotta pystyn seuraamaan sen toimintaa hahmojen tekoälyn Behavior Tree -käytöspuuta hallitessa. Conduit solmut ovat edistyneempi ja jaettava siirtymäresurssi. Jos tavallinen siirtymäsääntö tulee olemaan yksinkertainen 1–1-suhde tilasta toiseen, Conduit solmulla suhde voi olla 1-moneen, monta-1 tai monta-moneen. Conduit solmut näkyvät animaatiokaaviossa, ja näyttävät samanlaisilta kuin animaatiotilat, mutta niiden nimen edessä oleva kuvake on erilainen. (Epic Games 2023)

5.8 Seurantaviikko 8 (21.8.2023-25.8.2023)

MA 21.8.2023

Tämän päivän työtehtäviini kuuluu jatkaa Control Rig -työkaluun tutustumista. Päivän osaamistavoitteisiin kuuluu osaamisen laajentaminen kyseisestä työkalusta.

Loin Control Rig -työkalun avulla systeemin, joka saa Longtail hahmon katsomaan tiettyjä kohteita kohti, näiden ollessa tietyllä etäisyydellä. Lisäsin Control Rig -työkaluun muuttujan vector arvolle, joka toimii katseen kohdistuspisteenä. Loin uuden LookAtActor objektin, jolla on säädettävä vector piste, johon Longtailin katse kohdistetaan. LookAtActorin objektin Begin Overlap -tapahtuman kautta, asetetaan tarvittavat muuttujat Longtail hahmon blueprintiin, jotka taas välitetään animaatioblueprintiin, ja johon luomani funktio käsittelee kohdistuspisteen käsittelyn. Loin animaatioblueprintiin uuden Blend Poses By Bool -solmun, joka otetaan käyttöön, mikäli kohteeseen katsomiseen tarvittavat boolean arvot toteutuvat. Mikäli arvot pysyvät epätotena, käytetään hahmon perinteistä Locomotion Cache -animaatiotilaa. Luomiani LookAtActoreita on mahdollista liittää esimerkiksi NPC hahmoihin, jolloin Longtail kääntää katseensa hahmoja kohti, tai erikseen, jonka avulla on mahdollista luoda niin sanottuna “point of interest” -pisteitä pelimaailmaan. Olen melko tyytyväinen tähän kohtalaisen nopeaan kokeiluun, ja lopputuloksesta tuli varsin hyvä.

Päivän osaamistavoitteet onnistuivat mukavasti. Sain kartutettua taitojani Control Rig -työkalun kanssa, ja luotua sen avulla uuden toiminnallisuuden.

TI 22.8.2023

Tämän päivän työtehtäviini kuuluu työskennellä Control Rig -työkalun kanssa, ja laajentaa eilen aloittamani systeemi, jonka avulla Longtail hahmo kohdistaa katseensa valittuihin kohteisiin. Päivän osaamistavoitteisiin kuuluu Control Rig -

työkalun hyödyntäminen ja katseen kohdistamisen laajentaminen eri Actor objekteihin.

Päätin muuttaa eilen luomaani systeemiä siten, että en enää kiinnitä LookAtActor objekteja kiinni NPC, eli Non Playable Character -hahmoihin, jotta pystyn vähentämään kenttiin luotujen Actor objektien määrää. Lisäsin NPC hahmoille oman Sphere Collision- ja Billboard -komponentin, jotka käyttävät samaa toiminnallisuutta, kuin LookAtActor objekti. Billboard komponentti toimii katseen kohdistuspisteenä, ja sen oletus sijainti asetetaan NPC hahmon Skeletal Mesh -komponentin Head Socket -sijaintiin. Kohdistuspistettä on mahdollista kuitenkin siirtää manuaalisesti, joka tulee tarpeeseen erityisesti isojen hahmojen kanssa. Lisäsin saman toiminnallisuuden jo olemassa olevaan Interactable objektiin, jonka avulla Longtail voi olla vuorovaikutuksessa erilaisten asioiden kanssa. Kuten NPC hahmojen kanssa, Interactable objektien kohdistuspisteen sijaintia ja Sphere Collison -komponentin kokoa on mahdollista säätää manuaalisesti. Eilen luomani LookAtActor objektin jätin omaksi kokonaisuudeksi, joten kyseisiä objekteja voi lisätä pelimaailmaan, esimerkiksi niin sanottujen “point of interest” -pisteiden luomiseen. Jokaisella äsken mainitulla Actor objekteilla on myös mahdollisuus säätää kohdistuspiste päälle ja pois. Tämä on hyödyllinen ominaisuus, koska voi tulla tilanteita, jolloin kenttäsuunnittelija haluaa säätää kohdistuspisteiden toimintaa eri tilanteiden mukaan. Lopuksi laajensin Longtail hahmon animaatioblueprintissä olevan kohdistuspisteitä käsittelevän funktion käsittelemään erityyppiset LookAtActor objektit.

Päivän työtehtävät onnistuivat hyvin. Sain laajennettua luomaani LookAtActor objektien toiminnallisuutta, ja optimoimaan sen toimintaa. Osaamiseni kehittyi toiminnallisuuksien optimoinnissa, ja Control Rig -työkalun hyödyntämisessä.

KE 23.8.2023

Tämän päivän työtehtäviini kuuluu lisää Control Rig -työkaluun tutustumista, ja Longtail hahmon jalkojen sijainnin testailua. Päivän osaamistavoitteisiin kuuluu Control Rig -työkalun osaamisen laajentaminen, ja soveltaminen projektiin.

Luin Unreal Engine -pelimoottorin virallisista dokumenteista tietoa Control Rig -työkalun Full Body IK -ominaisuudesta, ja katselin videoita aiheesta. Yritin toteuttaa muutaman ohjevideon avulla jalkojen ja lantion korkeuden muuttamista Longtail hahmolle. Lisäsin kontrollipisteet ja muutaman ylimääräisen luun hahmon Rig Hierarchy -tietoihin, ja aloin työstämään toteutusta videon ohjeiden mukaan. Haasteita tuli kuitenkin eteen, koska videoiden ohjeet olivat tehty Unreal Engine -pelimoottorin versiolla 5, ja meillä on vielä käytössä versio 4. Versiossa 4 Control Rig -työkalussa ei ole kaikkia samoja ominaisuuksia kuin versiossa 5, joka aiheutti ylimääräisiä haasteita. Esimerkiksi funktioiden tekoa ei tueta, ja muutama yksittäinen solmu ei sisällä samoja tietoja. Tämä aiheutti sen, että tulokset eivät toimineet halutulla tavalla, ja jalkojen ja lantion korkeuden muuttaminen dynaamisesti ei toiminut kunnolla.

Päivän työtehtävät eivät tällä kertaa onnistuneet halutulla tavalla. Erot Unreal Engine -pelimoottorin versioiden välillä aiheuttivat haasteita, joiden seurauksena kokeilemani toteutukset eivät toimineet halutulla tavalla. Päivän osaamistavoitteet onnistuivat vain osittain. Sain kartutettua osaamista Control Rig -työkalun kanssa, mutta käytännön soveltamisen ratkaisu ei onnistunut.

TO 24.8.2023

Tämän päivän työtehtäviini kuuluu lisää Control Rig -työkaluun tutustumista, ja Longtail hahmon jalkojen sijainnin testailua, sekä pohjan luominen uudelle pomo hahmolle ja Control Rig -testailua kyseiselle hahmolle. Päivän osaamistavoitteisiin kuuluu Control Rig -työkalun osaamisen laajentaminen, ja soveltaminen projektiin.

Kokeilin Longtail hahmon jalkojen sijainnin asettamista erään toisen tutoriaalin avulla. Kyseinen tutoriaali käytti Full Body IK -toteutuksen sijasta Two Bone IK -solmua, jonka ohjaaminen onnistuu Character- ja animaatioblueprintien kautta. Sain etsittyä raycast tarkistuksilla jalkojen sijainnit, ja toteutettua jalkojen ja lantion siirtämisen oikeille paikoilleen. Lopputulos oli kuitenkin hieman erikoinen, mikä saattaa johtua siitä, että Longtail hahmon Skeletal Mesh -objekti

ei ole täysin samanlainen kuin Unreal Engine -pelimoottorin perus “ihmisen Skeletal Mesh” -objekti. Jalat ja lantio kyllä liikkuivat maan muotojen mukaan, mutta liike ei ollut tarpeeksi sulavaa. Tämän lisäksi aloitin uuden pomo hahmon työstämisen. Loin hahmolle Character-, animaatioblueprintin ja Control Rig -työkalun. Aloin testailla hahmon lonkeroiden (hahmolla on kuusi lonkeroa), sijainnin liikuttamista proseduraalisesti ohjaamalla lonkeroiden kontrollipisteitä Control Rig -työkalun avulla. Sain hahmon lonkeron liikkumaan sulavasti hahmon yläpuolelle, ikään kuin valmistautumaan lyöntiin. Itse lyömistä ehdin myös aloittaa, mutta se jäi vielä keskeneräiseksi.

Päivän työtehtävät onnistuivat osittain, sillä jalkojen sijainnin asettamisen toiminnallisuus ei toiminut täysin halutulla tavalla. Kyseessä oli kuitenkin vain testailu, eikä kyseistä mekaniikkaa ole välttämättä tarkoitus implementoida lopulliseen peliin. Pomo hahmon osalta sain luotua pohjan hahmolle, ja aloin työstämään hahmon liikettä Control Rig -työkalun avulla. Päivän osaamistavoitteet onnistuivat osittain, sillä en ollut jalkojen liikkeen testailun tuloksiin täysin tyytyväinen. Pomo hahmoon liittyvät Control Rig -työkalun testailut alkoivat kuitenkin lupaavasti.

PE 25.8.2023

Tämän päivän työtehtäviini kuuluu parannella pomo hahmon lonkeroiden kontrolloimista Control Rig -työkalun avulla. Päivän osaamistavoitteisiin kuuluu Control Rig -työkalun osaamisen laajentaminen, ja soveltaminen projektiin.

Huomasin heti pienen ongelman Control Rig -työkalun ja lonkeroiden käytössä: Vaikka vaihdoin muuttujien arvoja välittämällä niitä animaatioblueprintistä Control Rig -työkaluun, ei CCDIK solmu osannut vaihtaa arvoja ensimmäisen valinnan teon jälkeen. Tämä tarkoitti sitä, että jos liikutin ensin lonkeroa indeksillä kaksi, tapahtuivat sen jälkeen kaikki liike kyseisellä indeksillä, vaikka vaihtaisin kyseisen solmun liikuteltavia luiden tietoja. Päädyin siis ratkaisuun, jossa jokaiselle lonkeroille tulee tehdä oma CCDIK solmu, ja liikuttaa sen avulla kontrollipisteitä haluttuihin sijainteihin. Tämä ratkaisu toimi loppujen lopuksi todella hyvin, ja sain ohjattua jokaista lonkeroa erikseen. Parantelin myös

funktioiden yleistä sulavuutta, ja säätelin interpolaatio aikoja, ja sain liikkeistä varsin sulavia. Tekemieni muutosten seurauksena, pystyin pitämään Control Rig -työkalun päällä koko ajan animaatioblueprintin kautta, ilman että minun tarvitsi vaihdella sen välillä. Tämä tekee Control Rig -työkalun animaatioista proseduraalisia, ja helpommin käsiteltäviä. Näiden parantelujen jälkeen jatkoin lyönti iskun suunnittelua, jonka pitäisi tapahtua lonkeron ylös nostamisen jälkeen, ja osua Longtail hahmon sijaintiin.

Päivän työtehtävät onnistuivat oikein mukavasti. Jouduin taistelemaan melko paljon, että sain jokaisen lonkeron kontrolloitavaksi erikseen, mutta olen kehittämiini ratkaisuihin oikein tyytyväinen. Päivän osaamistavoitteet täyttyivät myös, koska sain laajennettua osaamista Control Rig -työkalun käytön soveltamisessa.

Seurantaviikko 8 – Viikkoraportti: 21.8.2023 – 25.8.2023

Tämän viikon teemana oli hyvin vahvasti Control Rig -työkalu, ja sen käytön opiskelu. Unreal Engine -pelimoottori tarjoaa sarjan animaatiotyökaluja, joiden avulla voi luoda ja animoida hahmoja suoraan moottorissa, nimeltä Control Rig. Control Rig -työkalun avulla voi ohittaa tarpeen muokata ja animoida ulkoisissa työkaluissa ja animoida sen sijaan suoraan Unreal Enginen -pelimoottorissa. Tämän järjestelmän avulla voi luoda ja asentaa mukautettuja ohjaimia hahmolle, animoida Sequencer työkalun avulla ja käyttää monia muita animaatiotyökaluja animaatioprosessin avuksi. (Epic Games 2023)

Minulla oli paljon opiskeltavaa Control Rig -työkalun käytöstä, sillä se oli minulle täysin uusi työkalu. Yksi työkalulla suorittamani kokeilu, oli Longtail hahmomme katseen kohdistaminen erinäisiä “point of interest”, eli kiinnostuksen kohde -pisteitä kohti. Loin hahmolle uuden Control Rig -objektin, jonne asetin muokattavan vector arvon, toimii kontrollipisteenä hahmon katseen kohdistamiselle. Hahmon pään alueen luut on liitetty kyseiseen kontrollipisteeseen. Kyseiselle vector arvolle annetaan arvoja parametrina, jotka saadaan määrittämieni pisteiden sijainneista. NPC hahmoille loin tällaiset

pisteet hahmojen Skeletal Mesh -komponentin Head Socket -sijaintiin. Tein pisteen sijainnista kuitenkin tarvittaessa muokattavan (kuva 5).

Kuva 5. Longtail hahmo kohdistamassa katsettaan toista pelihahmoa kohti Control Rig -työkalua hyödyntäen (Kuva 6: Hologram Monster).

Loin myös uuden Actor objektin, joka on mahdollista sijoittaa pelikenttään, ja joka tunnistetaan tällaiseksi kiinnostuksen kohteeksi Longtail hahmon blueprintissä. Kehittämäni systeemi toimii yhdessä animaatioblueprintin kanssa. Hahmon havaittua kiinnostuksen kohteen, hahmo välittää animaatioblueprintille tiedon pisteestä. Tämän jälkeen suoritetaan funktio, joka tarkistaa, onko kyseinen piste suhteellisesti hahmon edessä. Tämä tarkistus varmistaa sen, että hahmon pää kääntyy vain tietyn verran sivu- ja pystysuunnassa. Mikäli piste löytyy, ja määritellyt arvot täyttyvät, animaatioblueprint välittää Control Rig -työkalulle tiedon pisteen sijainnista pelimaailmassa, ja ohjaa hahmon katseen osoittamaan kohti kyseistä pistettä. Tällaisella logiikalla sain hahmon katseen kohdistumaan tiettyihin pisteisiin, Control Rig -työkalua hyödyntämällä.

Tutustuin myös hieman siihen, miten saisin kontrolloitua pomo hahmon lonkeroita Control Rig -työkalun avulla. Päädyin käyttämään CCDIK (Cyclic Coordinate Descent Inverse Kinematics) Skeletal Control -solmua, ohjaamaan

lonkeron kaikkia luita. CCDIK solmu on kevyt, IK-ratkaisin (samanlainen kuin FABRIK, eli Forward And Backward Reaching Inverse Kinematics), jota käytetään tyypillisesti luuketjun ohjaamiseen. Toisin kuin FABRIK solmu, CCDIK solmu tarjoaa kuitenkin mahdollisuuden määrittää kulmarajoituksia, jotka voivat osoittautua hyödyllisiksi, jos haluat rajoittaa näiden luiden pyörimistä ratkaisun aikana. CCDIK-solmua voidaan käyttää proseduraalisten animaatioiden ohjaamiseen. (Epic Games 2023) Tällä menetelmällä määrittämällä IK-pisteelle sijainnin, sain ohjattua koko lonkeron ketjumaisesti kohti kyseistä sijaintia. Vaihtoehtoinen tapa olisi ollut myös käyttää laajemman tason käsittelijää, kuten Full Body IK -solmua. Sen käyttöä en kuitenkaan vielä hallitse riittävällä tavalla, haluttujen lopputulosten saavuttamiseksi.

5.9 Seurantaviikko 9 (28.8.2023-1.9.2023)

MA 28.8.2023

Tämän päivän työtehtäviini kuuluu jatkaa pomo hahmon lonkeroiden kontrolloimista Control Rig -työkalun avulla. Päivän osaamistavoitteisiin kuuluu Control Rig -työkalun osaamisen laajentaminen, ja soveltaminen projektiin.

Longtail hahmon sijaintiin sijoittuva isku aiheutti melko paljon päänvaivaa, koska asiassa oli todella paljon huomioitavia asioita, joita emme olleet miettineet etukäteen. Näitä olivat muun muassa mitä tapahtuu lonkerolle, jos se osuu johonkin objektiin matkalla, tai onko lonkeron iskun tarkoitus “kaatua” kohti Longtail hahmon sijaintia, vai iskeä vain tämän sijaintiin. Kokeilin erilaisia tapoja suorittamalla raycast tarkistuksia Longtail hahmon suuntaan, sekä sijaintiin, ja ohjaamalla iskuja saatujen tulosten mukaan. Loppujen lopuksi kumpikaan toimintatapa ei suoranaisesti näyttänyt kovin hyvältä. Lonkeron rotaation kiertäminen aiheutti myös hieman haasteita, koska Control Rig -työkalun sisältämät solmut ovat hyvin erilaisia, verrattuna perinteisiin blueprint solmuihin, ja esimerkiksi parametrien välittämistä tarvitsee suorittaa hieman eri tavalla. Lähdin myös kokeilemaan, olisiko CCDIK solmujen vaihtaminen Full Body IK -solmuiksi parempi ratkaisu, tosin myös niiden toiminta on melko erilainen

Unreal Engine -pelimoottorin versioiden 4 ja 5 välillä. Tämä työpäivä kului hyvin pitkälti asioiden kokeilemisessa, ja erilaisten matemaattisten toimenpiteiden suorittamisessa.

Päivän työtehtävät ja osaamistavoitteet eivät varsinaisesti onnistuneet, koska en saanut vielä toteutettua toimivaa ratkaisua iskujen toteuttamiseen. Sain kuitenkin kartutettua osaamistani jonkin verran muuan muassa Full Body IK -solmujen osalta.

TI 29.8.2023

Tämän päivän työtehtäviini kuuluu jatkaa pomo hahmon lonkeroiden kontrolloimista Control Rig -työkalun avulla. Päivän osaamistavoitteisiin kuuluu Control Rig -työkalun osaamisen laajentaminen, ja soveltaminen projektiin.

Jatkoin hieman Full Body IK -solmun testailua, ja jostain syystä en saanut Constraints rajoituksia toimimaan kunnolla, ja hahmon lonkeroihin liittämäni IK pisteet vaikuttivat kaikkiin luihin. Päädyin siirtymään takaisin CCDIK toteutukseen, ja kokeilla ratkaista ongelmaa luomalla Spline komponentin, ja ohjata sen avulla lonkeroiden liikerataa. Säädin Spline komponentin loppupisteen samaan kohtaan ja samalle etäisyydelle kuin aiemmin luomani Line Trace raycast -tarkistuksen loppupiste, ja loin Spine komponentille melko korkean kurvin ylhäältä alaspäin. Lonkeron liike liikkui ihan kohtalaisesti pitkin Spline komponettia, mutta lonkeron rotaatioissa oli vielä jotakin häikkää. Spline komponentin kurvia säätämällä voisin saada myös lonkeroiden rotaation toimimaan paremmin.

Päivän työtehtävät etenivät jonkin verran Spline komponentti toteutuksen ansiosta, mutta se vaatii vielä lisää työstämistä. Päivän osaamistavoitteet onnistuivat myös osittain, koska sain sovellettua Spline komponentin toimintaa osaksi Control Rig -työkalun toteutusta.

KE 30.8.2023

Tämän päivän työtehtäviini kuuluu jatkaa edelleen pomo hahmon lonkeroiden kontrolloimista Control Rig -työkalun avulla. Päivän osaamistavoitteisiin kuuluu Control Rig -työkalun osaamisen laajentaminen, ja soveltaminen projektiin.

Lonkeroiden kontrollointi on aiheuttanut melko paljon päänvaivaa, eikä ratkennut kunnolla vieläkään. Hienosäädin Spline komponentin arvoja, ja yritin saada lonkeron rotaatioita toimimaan paremmin. Rotaatiot tuntuivat tekevän erikoisen käännöksen koko ajan, vaikka kokeilin erilaisia ratkaisuja arvojen säätämisessä. Loppujen lopuksi päätin kokeilla luoda erillisen Spline Actor -objektin, jota pomo hahmon oma blueprint referoi ja hyödyntää. Tämä toteutus toimi paljon paremmin, ja Spline komponentin kurvi toimi sulavammin. Jostain syystä Spline komponentin lokaalit arvot eivät toimineet yhtä hyvin, kun komponentti oli liitetty itse pomo hahmon blueprintiin. Spline komponentin kurvia säätämällä alkaa lyönti näyttää jo ihan kohtalaiselta, mutta lonkeron rotaatio jää lopuksi vielä hieman erikoiseen kulmaan. Tämä minun tulee vielä selvittää ja ratkaista.

Päivän työtehtävät etenivät hieman, mutta kyseiseen työtehtävään on kulunut yllättävän paljon aikaa, ja se on tuntunut olevan haastava ongelma ratkaista. Osaamiseni kuitenkin kehittyi entisestään Spline- komponentin hyödyntämisessä osana Control Rig -toteutusta.

TO 31.8.2023

Tämän päivän työtehtäviini kuuluu jatkaa edelleen pomo hahmon lonkeroiden kontrolloimista Control Rig -työkalun avulla, ja yrittää ratkaista lonkeroiden rotaatioihin ja sijainteihin kohdistuneita haasteita. Päivän osaamistavoitteisiin kuuluu Control Rig -työkalun osaamisen laajentaminen, ja soveltaminen projektiin, sekä ratkaisujen keksimistä edellä mainittuihin pulmiin.

Lähdin lähestymään lonkeroiden rotaation korjaamista siten, että annan Control Rig -työkalun tehdä omat toimensa, ja yrittää manipuloida rotaatioita ja sijainteja

erikseen Transform Bone -solmujen avulla. Lisäsin animaatioblueprintiin uuden Cached Pose -solmun, joka suoritetaan Control Rig -solmun jälkeen. Cached Pose -solmusta suoritus kulkee Output Pose -solmuun, mikäli lonkeroissa ei tapahdu hyökkäyskäskyjä. Mikäli hyökkäyskäskyjä tapahtuu, lisäsin suorituksen kulkemaan Blend Poses by Enum -solmun avulla Transform Bone -solmuihin, joiden avulla säädän valittujen luiden rotaatiota ja sijaintia. Sain kyseisellä menetelmällä säädettyä lonkeron väärässä asennossa olevan rotaation oikein, ja myös sijaintia paranneltua. Ajattelin kokeilla toteutusta, jonka avulla lasken raycast tarkistusten avulla Transform Bone -solmujen sijaintiarvoja. Kokeilemallani toteutuksella sijainti toimi hyvin, lonkeron osuessa suoraa eteenpäin. Lonkeron iskiessä sivuille, tapahtuu vielä hieman heittoa, joka minun tulee ratkaista säätämällä vektori arvojen tietoja. Kyseinen toteutus jäi minulla kesken, mutta se vaikuttaa tässä vaiheessa varsin lupaavalta keinolta ratkaista haasteet sijaintien kanssa.

Päivän työtehtävät etenivät, koska sain paranneltua tekemälläni toteutuksella lonkeroiden rotaatio arvoja, ja mahdollisesti keksittyä ratkaisun myös sijainnin korjaamiseen. Päivän osaamistavoitteet onnistuivat myös, koska sain otettua haltuun Transform Bone -solmujen toimintaa, ja hyödynnettyä niitä tekemässäni toteutuksessa.

PE 1.9.2023

Tämän päivän työtehtäviini kuuluu jatkaa edelleen pomo hahmon lonkeroiden kontrolloimista Control Rig -työkalun avulla, ja yrittää ratkaista lonkeroiden rotaatioihin ja sijainteihin kohdistuneita haasteita. Päivän osaamistavoitteisiin kuuluu Control Rig -työkalun osaamisen laajentaminen, ja soveltaminen projektiin, sekä ratkaisujen keksimistä edellä mainittuihin pulmiin.

Lähdin korjaamaan rotaation korjaamisen laskemista FindLooktAtRotation solmun avulla, jonka avulla sain rotaation pomo hahmosta Longtail hahmon suuntaan. Syötin saadun Yaw arvon Transform Bone -solmum Pitch arvoon, jonka avulla sain sopivan arvon rotaation kääntämiseksi. Iskettäessä suoraa eteen, lonkeron rotaatio kääntyy maata kohti, kun taas hieman sivulle

iskettäessä lonkero saa pientä kulmaa iskun suuntaisesti. Parantelin myös vielä alaspäin tapahtuvia raycast tarkistuksia, ja varmistin että niistä saadaan aina jokin arvo, jotta vektori arvoihin ei jää null, eli tyhjiä arvoja, jotka lähettävät lonkeroiden luut maailman nollapisteeseen. Lisäsin tapahtuviin raycast tarkistuksiin pituutta, ja laitoin niille oletussijainnin luiden Socket sijaintien arvoihin. Saatuani rotaatiot ja raycast tarkistukset sopiviksi, lisäsin vielä pienen tärinäefektin lonkeroille, niiden osuessa maahan. Tähän tein varsin simppelin Timeline solmun, jonka aikana vaihdellaan Transform Bone -solmun Translation arvoa nopeasti eri suuntiin.

Päivän työtehtävät onnistuivat hyvin, koska sain viimein laskettua rotaation korjaamiselle sopivat arvot, sekä tehtyä lyönnistä onnistuneen näköisen. Sain myös lisätty lonkeroihin simppelin tärinä efektin. Päivän osaamistavoitteet onnistuivat, koska sain ratkaistua rotaation kanssa esiintyneet haasteet.

Seurantaviikko 9 – Viikkoraportti: 28.8.2023 – 1.9.2023

Tämä viikko oli hyvin haastava. Minulla oli todella paljon haasteita pomo hahmon luiden rotaatio arvojen kanssa. Pomo hahmolle oli tarkoitus toteuttaa hahmon sijaintiin tapahtuva isku. IK-pisteen määrittäminen kurvin kanssa oli haastavaa. Sain ratkaistua ongelman hyödyntämällä Spine komponenttia suorittamaan iskun lyönnin kaaren. Haasteita esiintyi kuitenkin edelleen tiettyjen luiden rotaatioiden kanssa, iskun päätyttyä. Tämä esiintyi ilmiönä, jossa iskun lopuksi lonkeron kärki kääntyi erikoisesti. Sain loppujen lopuksi laskettua rotaatiolle kääntyvän kulman hyödyntämällä muuan muassa FindLookAtRotation solmun Yaw akselin arvoa, ja muita laskutoimituksia. Vaikka ratkaisu toimi, en ole kuitenkaan täysin vakuuttunut, oliko tämä järkevä tapa toteuttaa rotaation kääntäminen. Parempi tapa olisi mielestäni ollut rajoittaa luiden rotaatioon liittyviä arvoja Control Rig -työkalun puolella, mutta kokeiluista huolimatta en saanut näitä rajoituksia toimimaan.

Toinen ratkaistava asia oli saada lonkerot osumaan maahan asti iskun jälkeen. Tarkoitan tällä sitä, että IK-pisteen (joka sijaitsee lonkeron kärjessä, ja ohjaa koko lonkeroa ketjumaisesti) kohteen sijaintiin siirtymisen jälkeen, vain lonkeron

kärki osui maahan. Tarkoitus oli saada koko lonkero osumaan maahan aivan “kainalosta” eteenpäin. Tällä tavoin isku kattaa koko matkan lonkeron alkupisteestä sen kärkeen. Tämän sain ratkaistua hyödyntämällä Transform Bone -solmua Control Rig -työkalun tukena. Transform Bone -solmua voi käyttää tietyn luun käännön, kierron ja skaalauksen muuntamiseen. Ohjattavan luun valinnan jälkeen Bone To Modify -ominaisuuden avulla, voi muutostilan tyypin valita omainaisuusosiossa. Transform Bone -solmu toimii komponenttitilassa, joten tilamuunnos on suoritettava solmun toteuttamiseksi hahmon animaatioblueprintissä. (Epic Games 2023). Sijainnit Transform Bone -solmun vektori arvoille sain ajamalla raycast tarkistuksen suoraa alaspäin lonkeron jokaisen luun sijainnista, itse iskun liikkeen loppumisen jälkeen. Iskun loppumisen jälkeen lonkeron luut siirretään raycast tarkistuksista saatujen vektori pisteiden sijaintiin. Tällä menetelmällä sain ohjattua lonkerot iskeytymään maahan asti lonkeron koko matkalta.

Kaiken kaikkiaan tämä viikko oli melko haastava. Asioiden selvittelyyn ja ratkaisujen kehittämiseen kului paljon aikaa, ja se oli paikoitellen varsin turhauttavaa. Control Rig -työkalu on todella haastava kokonaisuus, joten sen kunnolla haltuun ottamiseen menee vielä paljon aikaa. Onnistumiset kuitenkin piristivät, ja sain vietyä asioita eteenpäin.

5.10 Seurantaviikko 10 (4.9.2023-8.9.2023)

MA 4.9.2023

Tämän päivän työtehtäviini kuuluu luoda osuma tarkistus pomo hamon lonkeron iskuille. Päivän osaamistavoitteisiin kuuluu mahdollistaa osumien käsittely, ja näin ollen vahingon tuottaminen pomo hahmon lonkeroiden iskuilla.

Kokeilin, saisinko osuma tarkistukset toteutettua ilman Dit It Hit -komponenttia, koska halusin yrittää pitää hahmon blueprintin mahdollisimman siistinä. Mikäli olisin hyödyntänyt Did It Hit -komponenttia, olisin joutunut mitä todennäköisimmin liittämään niitä kuusi kappaletta, yhden jokaista lonkeroa

kohden. Tätä ajatus ei miellyttänyt, joten päätin kokeilla, toimivatko raycast tarkistukset Timer ajastin tyylisessä tapahtumassa. Loin uuden Timer ajastimen, jonka laitoin pyörimään 0,016 sekunnin välein, joka vastaa 60 kuvaa sekunnissa. Kokeilin hieman erilaisia raycast tarkistus vaihtoehtoja, kuten Capsule- ja Sphere Trace -tarkistusta, sekä niiden alkamis- ja loppumispisteitä lonkeroiden eri Socket sijainteihin. Loppujen lopuksi loin kaksi eri MultiSphereTraceForObjects raycast -tarkistusta, joista ensimmäinen suoritetaan lonkeron ensimmäisen- ja viimeisen Socketin sijainnin väillä, luoden pitkän putkimaisen tarkistusalueen käden mitalle. Toisen raycast tarkistuksen laitoin alkamaan lonkeron viimeisestä Socket sijainnista, ja jatkumaan hieman eteenpäin lyönnin suuntaisesti, jotta se kattaa koko lonkeron kärjen alueen. Kyseiset raycast tarkistukset kattoivat juuri sopivan alueen lonkeroiden ympäriltä, ja tuntuivat pyörivän hyvin jopa hitaammilla ruudunpäivitysnopeuksilla. Osuman käsittelyt hoidin toistaiseksi vain Longtail hahmolle, joka vastaanottaa vahinkoa ja sysäyksen poispäin pomo hahmosta, sekä itse luodulle Custom Destructible -luokalle, joka kattaa esimerkiksi rikottavat vaasit ja kivenlohkareet.

Päivän työtehtävät onnistuivat oikein hyvin, koska sain toimivat osumatarkistuksen toteutettua lonkeron iskulle. Päivän osaamistavoitteet onnistuivat myös, koska sain toteutettua käsittelyt osumille, ja mahdollisuuden tuottaa vahinkoa eri Actor objekteille.

TI 5.9.2023

Tämän päivän työtehtäviini kuuluu kehitellä vaakatasossa tapahtuva lyönti pomo hahmon lonkerolle. Osaamistavoitteisiin kuuluu aiemmin toteutetun mekaniikan soveltaminen toiseen muotoon, luoden samalla uusi ominaisuus erilaisen iskun muodossa.

Lähdin lähestymään toteutusta aiemmin toteuttamani Spline komponentin avulla. Säädin Spline- komponentin vaakatasoon, ja säädin Spline pisteiden sijainteja toimimaan vaakatasossa. Lisäsin uuden IK-pisteen pomo hahmon Control Rig -työkaluun, joka toimii vaakatasossa tapahtuvan lyönnin

aloituspisteenä. Säädin pomo hahmon Character blueprintin asettamaan molemmat Spline komponenttien sijainnit halutuille etäisyyksille hahmon Construction Script -osiossa, jotta niitä ei tarvitse säätää joka kerran manuaalisesti. Sain hyödynnettyä aiemmin tekemääni toteutusta todella paljon, ja aiemmin luomani pystytasossa toimiva Spline komponentti toimi vain muutaman muutoksen jälkeen hyvin myös vaakatasossa. Minun tarvitsi luonnollisesti luoda omat muuttujat vaakatasossa tapahtuville arvoille, ja muuttaa hieman rotaatio arvoja. Control Rig -työkalun puolella säädin boolean muuttujien mukaan haluttujen IK-pisteiden toimintaa. Animaatioblueprintissä säädin Transform Bone -solmun avulla hieman luiden Z-arvoja, jotta sain lonkeron kulkemaan tasaisesti maata pitkin.

Päivän työtehtävät ja osaamistavoitteet onnistuivat hyvin. Sain luotua pomo hahmolle uuden vaakatasossa tapahtuvan iskun, jonka toteutuksessa sain hyödynnettyä aiemmin luomaani pystytasossa tapahtuvan iskun ohjelmakoodia.

KE 6.9.2023

Tämän päivän työtehtäviini kuuluu pomo hahmon vaakatasossa tapahtuvan lyönnin osumatarkistusten tekemistä. Tämän lisäksi minun tulee asentaa työkoneeseeni uusi kovalevy, siirtää projektin paikallinen kopio uudelle asemalle, ja siirtää versionhallinnan Depot kansioni uuteen sijaintiin. Päivän osaamistavoitteisiin kuuluu tietokoneen ja versiohallinnan päivittämistä, sekä raycastien kanssa työskentelyä.

Asensin uuden kovalevyn työkoneeseeni kiinnittämällä sen emolevyyn, ja ottamalla sen käyttöön Computer Management -asetusten Disk Management -osiosta. Annoin uudelle asemalle nimen, ja alustin aseman. Tämän jälkeen tein peliprojektista kopion uudelle asemalle. Versionhallinta tuli myös päivittää uuteen sijaintiin, joten säädin Perforce ohjelman asetuksista oman Workspace työasemani Depot sijainnin uudella asemalla sijaitsevaan projektikansioon. Tämän jälkeen versionhallinta toimi oikein, ja uusi asema on käytössä oikeaoppisesti. Tämän lisäksi työskentelin pomo hahmon vaakatasossa tapahtuvien iskujen kanssa. Sain lonkeron myötäisesti tapahtuvat raycast

tarkistukset toimimaan hyvin, asettamalla ne alkamaan tietystä lonkeron alkupäässä olevan luun sijainnista, lonkeron päässä olevaan luun sijaintiin, antamalla sille hieman Offset arvoa alaspäin, jotta lonkeron ja raycast tarkistuksen yli on mahdollista hypätä. Lonkeron päässä oleva pallon muotoinen raycast tarkitus tarvitsi hieman enemmän hienosäätöä, koska luun Socket sijainti kääntyy Spline komponentin myötä toiseen kulmaan, jolloin raycast tarkistuksen sijaintia tarvitsee säätää. Tämä jäi minulla vielä hieman keskeneräiseksi.

Päivän työtehtävät ja osaamistavoitteet onnistuivat osittain. Sain uuden kovalevyn asennettua ongelmitta, ja siirrettyä versionhallinnan uuteen sijaintiin. Pomo hahmon vaakatasossa tapahtuvien iskujen osuma tarkistukset jäivät vielä hieman kesken, koska lonkeron päässä tapahtuvat raycast tarkistukset vaativat enemmän hienosäätöä.

TO 7.9.2023

Tämän päivän työtehtäviini kuuluu uuden Build version tekeminen pelistä, sen testaaminen, ja sen myötä mahdollisesti ilmenneiden asioiden ja bugien korjaaminen. Päivän osaamistavoitteisiin kuuluu testaamista, ja bugien korjaamista.

Tein pelistä uuden Build version, eli pakatun version, jota voi pelata ilman Unreal Engine -pelimoottorin käynnistämistä. Pelin pakkaamisen jälkeen aloin testaamaan sitä, ja pelaamaan pelin demo kenttää alusta loppuun. Testaamisen aikana ilmeni muutama bugi, ja korjattava asia. Pelin päävalikossa sijaitseva Options HUD -valikko oli vielä päivittämättä käyttämään FMOD ääniä. Tein valikkoon tarvittavat toimenpiteet, eli korvasin vanhat Global Sound Mix -solmut uusilla FMOD Set Global Parameter by Name -solmuilla. Huomasin myös, että aiemmin toteutettu Collectable Group Actor -objektien passiivinen leijumisääni (ääni kuuluu, kun ryhmä kerättäviä esineitä leijuu ilmassa, ja katoaa kun kaikki esineet on kerätty) palasi soimaan sen jälkeen, kun peliä jatketaan tallennuksen kautta. Ongelma oli makrossa, joka tarkistaa kyseisen kerättävien esineiden ryhmän pelin alkaessa. Makro ei nollannut ryhmän lukua, jonka vuoksi

tarkistettavat luvut eivät täsmänneet. Ongelma oli ratkaistu nopeasti lisäämällä makron alkuun luvun nollaus, jonka jälkeen kokonaislukuarvon tarkistus suoritetaan oikein. Huomasin myös, että mikäli pelaaja ostaa kentässä esiintyvältä kauppiaalta esineen keräämillään kristalleilla, kerättävien kristallien arvo ei täsmää oston jälkeen, eivätkä kerättävien kristallien luvut täsmää enää ollenkaan, Longtail hahmon kerätessä uusia kristalleja. Tässä tapauksessa ongelma oli se, että kauppatapahtuman suorittavassa HUD elementissä oli jäänyt päivittämättä käytettävä lukuarvo, jolloin luku ei päivittynyt oikein. Kerätyt kristallit näyttävään HUD elementtiin oli tehty muutoksia muutama kuukausi sitten, jolloin kyseinen lukuarvo oli jäänyt päivittämättä, joten sain ongelman ratkaistua varsin nopeasti vaihtamalla lukuarvon päivittämisen oikeaan lukuun. Vastaavasti koska lukuarvo ei päivittynyt oikein oston tapahtuessa, eivät tapahtuman jälkeen kerättyjen kristallien lukuarvot päivittyneet enää oikein. Tämän ratkaisin välittämällä kauppatapahtuman suorittavasta HUD elementistä oikean lukuarvon parametrina kristallien määrän esittävään HUD elementtiin. Tämän jälkeen luvut päivittyivät taas normaalisti.

Päivän työtehtävät ja osaamistavoitteet onnistuivat hyvin. Sain etsittyä testaamisen aikana ilmenneitä bugeja, ja korjattua ne oikeaoppisesti.

PE 8.9.2023

Tämän päivän työtehtäviini kuuluu lisää pelin pakatun version testaamista, ja sen myötä ilmenneiden asioiden ja bugien korjaaminen. Päivän osaamistavoitteisiin kuuluu testaamista, ja bugien korjaamista.

Pelissä ilmeni todella erikoinen bugi tietyissä muutamissa pelin objekteissa, joita ohjataan Sequencer työkalun avulla. Eräässä luolassa on kaksi kivipilaria, joiden sijaintia muutetaan painamalla kentässä esiintyviä nappeja. Jotain on kuitenkin tapahtunut pilareille, koska pelaajan painaessa hyppy nappia pilarin päällä, Longtail hahmo ei hyppää kunnolla, vaan leijuu ilmassa pienen hetken, jonka jälkeen pyörittää korkealta laskeutumisanimaation. Kyseinen bugi on esiintynyt myös aiemmin eräässä sillassa, jolle tapahtuu jotakin kummallista sen jälkeen, kun siltaa on liikutettu, peli tallennetaan ja ladataan tallennuksesta.

Tämän jälkeen sillalla hyppääminen heittää Longtail hahmon kovalla vauhdilla sivullepäin. Olen yrittänyt selvittää silloissa esiintyvää bugia aiemminkin, mutta en silloin löytänyt ratkaisua. Tällä kertaa pilareissä esiintyvät bugit kuitenkin haittaavat peliä jo sen verran paljon, että niihin tuli löytää ratkaisu. Pitkän selvittämisen jälkeen huomasin, että Level Sequence Actor -objekti, joka ohjaa pilarien sijaintia välianimaation aikana, oli asetettu alkuarvo pilareille Sequencer sekvensserin Frame kuvalle numero nolla. Olen lukenut bugeista, jotka johtuvat juuri siitä, että asioita tapahtuu nolla Frame kuvalla, koska nolla Frame kuvalla olevaa tietoa ei välttämättä lueta ollenkaan. Kokeilin siirtää pilareiden alkuperäisen Transform arvon määrittävän avainkehyksen ensimmäiselle Frame kuvalle, nolla Frame kuvan sijaan. Tämä näytti ratkaisseen ongelman, ja pilarit toimivat taas normaalisti. Ajattelin korjaavani ongelman myös silta objektista samalla toimintatavalla, mutta sille taas oli asetettu arvo ensimmäiselle Frame kuvalle, mutta bugi ilmenee sillan kanssa silti. Tämä johtuu todennäköisesti pelin tallentamiseen ja lataamiseen liittyvistä asioita, koska bugi ilmenee vasta ladatessa peli uudestaan tallennustiedostosta. Asia jäi kuitenkin edelleen mysteeriksi, enkä ehtinyt selvittää sille syytä.

Päivän työtehtävät ja osaamistavoitteet onnistuivat ihan kohtalaisesti. Sain selvitettyä ja ratkaistua pilareihin liittyvät ongelmat, ja korjattua ne oikeaoppisesti. Vanha sillassa esiintyvä bugi jäi kuitenkin edelleen selvittämättä, vaikka pääsinkin hieman paremmin sen jäljille, kuin aiemmin.

Seurantaviikko 10 – Viikkoraportti: 4.9.2023 – 8.9.2023

Tällä viikolla työskentelin muun muassa pomo hahmon iskujen parissa. Loin hahmolle myös vaakatasossa tapahtuvan iskun. Pystyin hyödyntämään toteutuksessa aiemmin luomaani Spline komponenttia pystysuunnassa suoritettavan iskun kanssa. Raja-arvot ja tarvitsivat vain hieman säätöä, jonka jälkeen vaakatasossa suoritettava isku liikkui hyvin. Sain toteutettua molemmille iskuille myös raycast tarkistukset osumia varten, tosin vaakatasossa tapahtuvan iskun tarkistukset jäivät vielä hieman kesken. Tällä kertaa päädyin hyödyntämään raycast tarkistusten piirtämiseen Timer ajastimia. Unreal Engine 4 -pelimoottorin ajastimet ovat hyödyllinen työkalu, jota voidaan käyttää

toiminnon suorittamiseen kerran tai toistuvasti tietyin väliajoin. Ajastimia voidaan käyttää toimintojen asettamiseen viiveellä tai tietyn ajanjakson aikana. (Epic Games 2023). Laitoin ajastimen suorittamaan tarkistusta 0,016 sekunnin välein, joka vastaa 60 kuvaa sekunnissa. Tällä tavoin sain piirrettyä tarkistuksia tasaisesti hyödyntäen Timer ajastinta.

Tein myös pelistä uuden pakatun version, ja kokeilin sitä mahdollisten bugien varalta. Muutama pieni korjattava asia löytyikin, ja sain ne ratkaistua varsin nopeasti. Pääasiassa ne olivat huolimattomuusvirheiden tai unohtuneiden asioiden korjaamista ohjelmakoodissa. Yksi suurempi haaste oli kuitenkin omituiset ilmiöt Sequncer sekvensserin kanssa. Tiimimme on huomannut muutamaan otteeseen, että sekvensseriin liitetyt objektit saattavat joskus toimia omituisella tavalla. Esimerkiksi silta, joka liikkuu sekvensserin aikana kahden pisteen välillä, saattoi laukaista pelaajan kohti sillan alkuperäistä sijaintia kovalla voimalla, mikäli pelaaja hyppää sillan päällä. Tätä ilmiötä on esiintynyt vain, mikäli sekvensseri on kelattu loppuun manuaalisesti, tai pelaaja on käynyt tallentamassa pelin toisessa Level Streaming -alueella. Tällä kertaa vastaava ilmiö oli ilmestynyt kivipilareihin, jotka liikkuvat sekvensserin aikana ylös- tai alapäin. Sekvensserin pyörittämisen jälkeen, hyppääminen pilareiden päällä ei enää onnistunut, vaan pelaaja lukkiutui pilareiden päälle. Tämä on äärimmäisen erikoinen ilmiö, ja valitettavasti en ole löytänyt sille mitään virallista syytä. Itse epäilen ilmiön johtuvan siitä, että objektin Transform arvo jää jonkinlaiseen välitilaan, ikään kuin se olisi edelleen liikkeessä sekvensserin päätyttyä. Tämä taas lisäisi velocity-arvoa pelaajalle, tämän suorittaessa hypyn. Tämä on kuitenkin vain spekulaatiota, ja en ole pystynyt todentamaan tätä teoriaa todeksi. Sain ongelman ratkaistua pilareiden kanssa lisäämällä niiden alkuperäisen Transform arvon Frame kuva 1:seen, Frame kuva 0:n sijaan. Kyseinen ratkaisu ei kuitenkaan toiminut sillan kanssa. Sain vinkin kyseisen ratkaisun kokeilemiseen YouTube-videosta “Fix the Broken First Frame in Sequencer / Movie Render Queue! - Unreal Engine”, jossa William Faucher (2021) puhuu Frame kuva 1:n aiheuttamista haasteista.

5.11 Seurantaviikko 11 (11.9.2023-15.9.2023)

MA 11.9.2023

Tämän päivän työtehtäviini kuuluu apina hahmon uusien animaatioiden implementointi peliin, mahdollisten uusin pelin pakatussa versiossa ilmenneiden bugien korjaaminen, ja mikäli aikaa jää, palata pomo hahmon lonkero hyökkäysten pariin. Päivän osaamistavoitteisiin kuuluu animaatioiden kanssa työskentely, bugien korjaaminen ja raycast tarkistusten hallinta.

Otin päivitetyt animaatiot versionhallinnasta, jotka freelance animaattorimme oli saanut viime viikolla tehtyä. Aloin implementoida uusia animaatioita aiemmin luomiini apina hahmon mekaniikkoihin, aloittaen animaatioblueprintin animaatiotilakoneesta. Huomasin heti, että joudun muuttamaan luomiani mekaniikkoja, ja Behavior Tree -käytöspuun toimintaa melko paljon, jotta määrittämäni ajat täsmäävät uusiin animaatioihin. Muutama animaatio oli pilkottu osiin, esimerkiksi hypyn aloitus, hypyn Idle, ja hypyn laskeutuminen. Minun tuli jakaa animaatiotilakone myös useampaan osaan näiden osalta. Muutama animaatio, jonka tarvitsisin myös pilkottuna, oli vielä yhtenä animaationa. Animaattori lupasi kuitenkin jakaa tarvitsemiani animaatioita useampaan osaan. Esimerkkinä tästä on hypystä laskeutuminen, joka voi tilanteesta johtuen päätyä eri AI tiloihin, kuten Idle-, Noticed-, tai Attacking -tilaan. Animaatioiden implementointi oli yllättävän työlästä, ja minulla on paljon säädettävää Behavior Tree -käytöspuun, ja hahmon Character blueprintin mekaniikoissa, jotta saan ne toimimaan hyvin uusien animaatioiden kanssa. Pelin pakatussa versiossa ei onneksi ilmennyt uutta korjattavaa, joten sen suhteen ei ilmennyt korjattavia asioita. Apina hahmon animaatioiden implementointi jäi vielä sen verran kesken, että en ehtinyt palata pomo hahmon pariin ollenkaan.

Päivän työtehtävät ja osaamistavoitteet onnistuivat osittain. Sain osan animaatioista implementoitua peliin, mutta suuri osa vaatii vielä lisää hienosäätöä. Uusia korjattavia bugeja pakatusta pelistä ei ilmennyt, enkä tehnyt tänään pomo hahmoon liittyviä työtehtäviä.

TI 12.9.2023 / KE 13.9.2023

Tiistain ja keskiviikon työtehtäviini kuuluu työmatka Helsinkiin Pocket Gamer Connect -tapahtumaan, ja pelimme esittely ja demoaminen kyseisessä tapahtumassa. Päivien osaamistavoitteisiin kuuluu pelialan tapahtumassa edustaminen, pelimme demoaminen ja verkostoituminen.

Matkustimme CEO Tonyn kanssa aikaisin aamulla tiistaiaamuna Helsinkiin Pocket Gamer Connect -tapahtumaan. Olin todella innoissani tapahtumasta, koska tämä on minulle ensimmäinen isompi pelialan tapahtuma. Vaikka kyseessä on pääasiassa mobiilipelien tapahtuma, oli siellä erillinen puoli Indie peleille. Puoli sisälsi noin kaksikymmentä pelistudiota, jotka olivat tulleet esittelemään pelejään. Meidän firmallamme kävi tuuri, koska voitimme arvonnassa pöydän Indie puolelle. Kasasimme pisteemme kuntoon, johon laitoimme kahdelle läppärille pelin pelattavaksi, ja pelin trailerin pyörimään. Tapahtuma vaikutti olevan todella suosittu, ja kävijöitä oli valtavasti. Pisteellämme oli lähes koko ajan innokkaita ihmisiä kokeilemassa peliä, ja keskustelemassa kanssamme. Keskustelimme paljon kehittämästämme pelistä, sekä myös Hologram Monsterista yrityksenä, ja toimintatavoistamme pelialan yrityksenä. Oli hienoa huomata, kuinka paljon kiinnostusta pelimme herätti, ja kuinka paljon kehuja siitä saimme. Tapasin tapahtumassa paljon muita pelikehittäjiä, pelialan opiskelijoita ja markkinointialan ihmisiä, ja kävin heidän kanssaan mielenkiintoisia keskusteluja. Sain heistä myös paljon uusia LinkedIn-kontakteja, joista voi olla hyötyä tulevaisuudessa. Kaiken kaikkiaan tapahtuma oli oikein onnistunut, ja mieleenpainuva kokemus.

Päivien työtehtävät ja osaamistavoitteet onnistuivat hienosti. Sain esiteltyä peliämme todella monelle tapahtuma kävijälle, sekä edustettua Hologram Monsteria ja sen toimintaa. Sain luotua uusia kontakteja, ja laajennettua verkostoani pelialan ihmisistä. Tapahtuma oli meidän osaltamme oikein onnistunut ja positiivinen kokemus. Minulle kertyi työtunteja kahden päivän aikana sen verran paljon, että sain pitää torstain vapaapäivänä.

PE 15.9.2023

Tämän päivän työtehtäviini kuuluu apina hahmon uusien animaatioiden peliin implementoinnin jatkaminen. Päivän osaamistavoitteisiin kuuluu animaatioiden kanssa työskentely, ja niiden yhdistäminen olemassa oleviin pelimekaniikkoihin.

Loin animaatiotilakoneeseen uusia tiloja animaatioiden jaksottamista varten. Loin putoamisen aloittavalle osuma-animaatiolle Animation Montage -animaation, joka pyöritetään pelaajan lyödessä puuta, jossa apina on. Animation Montage -animaation jälkeen siirrytään suoraa putoamisanimaatioon, ja maahan osuessa Stunned tilaan. Määritellyn Stunned tilan ajan jälkeen, suoritetaan pystyyn hypähtämisanimaatio, jonka jälkeen hypyn aloittaminen. Hypyn aloittavaan animaatioon liitin Animation Notify -tapahtuman, ja käynnistin aiemmin toteuttamani hyppäämismekaniikan alkamaan kyseisen Animation Notify -tapahtuman avainkehyksellä, jolloin hyppy takaisin puuhun alkaa oikeaan aikaan. Hypyn Idle osaa jatketaan, kunnes hahmon sijainti täsmää sen aloituspisteeseen, jolloin voidaan aloittaa laskeutumisanimaatio, joka on tällä hetkellä käytössä vain oksan päälle laskeutuvilla apinoilla. Blend Poses by Bool -solmun avulla saan mainiosti kontrolloitua animaatioita roikkuvien, ja oksien päällä sijaitsevien apinoiden välillä. Näin ollen voin hyödyntää samaa animaatiotilakonetta molemmille apina hahmo tyypeille. Oksan päältä alas putoava apina tarvitsee vielä hieman työstämistä, koska taaksepäin lentäminen täytyy saada toimimaan hyvin animaation ja liikkeen kanssa.

Päivän työtehtävät onnistuivat ihan kohtalaisesti. Sain apina hahmon animaatioiden implementointia jatkettua, mutta osa animaatioista ja mekaniikoista vaatii vielä lisää hienosäätöä. Kyseiseen työtehtävään on kulunut enemmän aikaa, mitä alun perin ajattelin, koska joudun säätämään mekaniikkoja ja niiden käynnistämistä eri tavoin uusien animaatioiden kanssa. Päivän osaamistavoitteet animaatioiden implementoimisesta etenivät, mutta jäivät vielä hieman keskeneräiseksi.

Seurantaviikko 11 – Viikkoraportti: 11.9.2023 – 15.9.2023

Tämä työviikko oli hieman tavallista erilaisempi. Työskentelin maanantain ja perjantain apina hahmon uusien animaatioiden implementoinnin parissa. Tiistain ja keskiviikon olin Helsingissä Pocket Gamer Connect -tapahtumassa esittelemässä peliämme, ja edustamassa Hologram Monster Oy:n toimintaa.

Onnistuneeseen pelialan tapahtumaan kannattaa valmistautua hyvin, ja huomioida suunnittelussa monenlaisia asioita. Yleissuunnitelmaan on suositeltavaa sisällyttää tarkat tapahtumien määräajat ja jättöpäivät. Kattava tarkistuslista kaikista tarvittavista materiaaleista helpottaa markkinointitehtäviä. Nämä materiaalit sisältävät yleensä pelin pelattavan version, pelin kuvataiteen, hyvän tekstipohjaisen presentaation, trailerit, logot ja joskus jopa hyviä markkinointikuvia tiimin kanssa. Materiaalien luominen uudelleenkäytettävyyden tavoitteella on tärkeää huomioida, ja tehdä materiaalin päivityksestä mahdollisimman helppoa. On tärkeää olla hyvin perillä osaston mitoista etukäteen. Joissakin Indie tapahtumissa olosuhteet eivät välttämättä ole ihanteelliset, ja kehittäjillä saattaa olla rajallisesti tilaa ja resursseja. On hyvä tapa räätälöidä ulkonäkö jokaiseen tapahtumaan, mutta jokaisella Indie kehittäjällä olisi hyvä olla myös "selviytymispakkaus", joka varmistaa vähintään kohtuullisen edustuksen missä tahansa tapahtumassa. Käyntikortit ovat olennaisia, ja niiden edustavuuden ja riittävyyden varmistaminen on tärkeää. Lentolehtisiä ovat myös hyvä olla, mutta on hyvin mahdollista, että suuri osa niistä päätyy roskakoriin päivän päätteeksi. Lisäksi mikä tahansa jaettavaksi tarkoitettu esine on suuri yleisön miellyttäjä. Ymmärrys siitä, miten peli näkyy messuolosuhteissa, kuten heikossa tai keinotekoisessa valaistuksessa, on olennaista. Näkyvyyden parantaminen suuremmilla fonteilla ja selkeillä ohjeilla voi houkutella enemmän kävijöitä. Interaktiivisten elementtien, kuten trailerien toistaminen hiljaisina hetkinä kannattaa hyödyntää, koska se voi kutsua ihmisiä pelaamaan. Ihmiset eivät myös välttämättä tiedä pelin näppäimiä luonnostaan. Ne kannattaa esitellä jollain tavalla, vaikkapa tulostamalla ne jonnekin, missä ne ovat helposti nähtävillä peliä pelatessa. Tämä voi auttaa luomaan sujuvamman pelikokemuksen. Peliajan rajoittamista tulee myös harkita, koska ei ole mukavaa, jos vain yksi henkilö valtaa esittelytilan eikä anna kenenkään muun

kokeilla peliä. Puhtauden priorisointi säännöllisesti ylläpitämällä laitteita ja ohjaimia on oleellista. Käsidesin, puhdistusliinojen ja roskapussien kaltaisten helposti saatavilla olevien puhdistusmateriaalien tarjoaminen on tärkeää, ottaen huomioon, että koska messuilla on paljon ihmisiä, ja heidän kosketuksensa näppäimistöön tai peliohjaimeen voi olla vaihdella suuresti. (Bularca 2016.)

Tapahtuma oli oikein onnistunut. Pisteellämme kävi paljon ihmisiä kokeilemassa peliämme, keskustelemassa kanssamme ja sain luotua tapahtuman aikana paljon uusia kontakteja. Tapahtumasta jäi kaikin puolin todella positiivinen mieli. Työpäivät tosin venyivät sen verran pitkiksi, että sain pitää torstaina vapaapäivän.

5.12 Seurantaviikko 12 (18.9.2023-22.9.2023)

MA 18.9.2023

Tämän päivän työtehtäviini kuuluu apina hahmon uusien animaatioiden peliin implementoinnin jatkaminen. Päivän osaamistavoitteisiin kuuluu animaatioiden kanssa työskentely, ja niiden yhdistäminen olemassa oleviin pelimekaniikkoihin.

Apina hahmon animaatioiden implementointiin ja mekaniikkojen hienosäätöön on kulunut yllättävän paljon aikaa. Minun tuli säätää myös heittämismekaniikkaa, koska uusien heittoanimaatioiden myötä heitot eivät olleet yhtä tarkkoja. Kiinnitin aiemmin lisäämäni Sphere Collision -komponentin hahmon käden Socket sijaintiin, jolloin heitettävän esineen kädestä poistuminen havaitaan paremmin. Siirsin esineen heittoradan laskevan logiikan hahmon blueprintistä heitettävän esineen blueprintiin. Lisäsin heitettävälle esineelle Projectile Movement -komponentin, ja hyödynsin sitä velocity arvon laskemiseen. Sain myös hyödynnettyä komponentin Rotation Follows Velocity -ominaisuutta, jolloin esineen rotaatio seuraa heittorataa, eikä pyöri ilmassa, miten sattuu. Nämä muutokset paransivat heittojen tarkkuutta huomattavasti. Keskustelimme tiimin kanssa myös hahmon toiminnallisuudesta, ja tuli ilmi, että hahmon ei ollut tarkoitus heitellä esineitä oksan päältä. Hahmon tulisi laskeutua

oksan alle roikkumaan, ja heitellä esineitä oksan alapuolelta. Minun tulee siis tehdä tällainen mekaniikka mahdolliseksi, kunhan saan siihen tarvittavat animaatiot. Tein tätä varten animaatiotilakoneen tilat ja siirtymät valmiiksi.

Päivän työtehtävät etenivät ihan kohtalaisesti, ja päivän osaamistavoitteet täyttyivät. Sain korjattua hahmon heittomekaniikan toimimaan hyvin uusien animaatioiden kanssa, ja tehtyä heitoista tarkempia. Opin hyödyntämään Projectile Movement -komponenttia heitettävien esineiden hallinnassa.

TI 19.9.2023

Tämän päivän työtehtäviini kuuluu apina hahmon uusien animaatioiden peliin implementoinnin jatkaminen. Päivän osaamistavoitteisiin kuuluu animaatioiden kanssa työskentely, niiden yhdistäminen olemassa oleviin pelimekaniikkoihin, ja päivitettyjen animaatioiden tuominen käytettäväksi pelimoottoriin.

Apina hahmon uusia animaatioita oli taas saatavilla, ja minun oli tarkoitus muistutella mieleeni niiden oikeaoppinen pelimoottoriin tuonti. Sain animaattorilta hahmon fbx-tiedoston, joka pitää sisällään hahmon Skeletal Mesh -objektin, sekä animaatio tiedostot. Etsin hakemistorakenteesta hahmon Skeletal Mesh -objektin, ja päivitin sen uuteen Re-import valinnalla. Jokaiselle jo olemassa olevalle animaatiolle tarvitsi tehdä sama toimenpide. Muutaman uuden animaation tuominen pelimoottoriin onnistui siten, että raahasin fbx-tiedoston pelimoottorin hakemistoon, jolloin Import valikko aukesi. Import-valikosta valitsin Import Mesh -valinnan, ja valitsin käytettäväksi hahmon olemassa olevan Skeletal Mesh -objektin. Tämän jälkeen painoin Import painiketta, jolloin kaikki uudet fbx-tiedostoon liitetyt animaatiot lisättiin pelimoottoriin. Muutamaan animaatioon oli lisätty Root Motion -ominaisuus, joka minun tuli vielä laittaa päälle kyseisistä animaatioista, sekä itse hahmolle Root Motion From Everything -valinta käyttöön. Apina hahmolla on kaksi perustilaa; pää alaspäin oksan alla roikkuva, ja oksan päällä makoileva. Tilasta riippuen, minun tuli säätää hahmon Mesh objektin sijaintia, jotta se sijaitsee oikeassa paikassa Capsule komponenttiin nähden. Liitin kyseisen operaation hahmon konstruktoriin, ja ohjelmakoodin kohtiin, joissa kyseinen tila vaihtuu toiseen.

Laitoin myös oksan päältä laskeutumiseen tehdyn animaation toimimaan molempiin suuntiin, hyödyntäen enumeraatio arvoa. Itse liikkeen toteuttavaa ohjelmakoodia tulee vielä työstää huomenna.

Päivän työtehtävät onnistuivat ihan kohtalaisesti. Sain otettua hahmon uusia animaatioita onnistuneesti käyttöön, vaikkakin prosessi eteni hieman hitaasti. Tämä johtui hienosäätöjen tarpeesta aiemmin luomassani koodissa, jotta se toimii uusien animaatioiden kanssa. Päivän osaamistavoitteet onnistuivat myös, koska sain kerrattua, miten päivitettyjä animaatioita tuodaan käytettäväksi pelimoottoriin.

KE 20.9.2023

Tämän päivän työtehtäviini kuuluu apina hahmon uusien animaatioiden peliin implementoinnin jatkaminen, erityisesti oksalta laskeutumista ja nousemista, puusta putoamista ja sinne takaisin hyppäämistä. Päivän osaamistavoitteisiin kuuluu animaatioiden kanssa työskentely, ja niiden yhdistäminen olemassa oleviin pelimekaniikkoihin.

Tänään hienosäädin apina hahmon oksalta alas laskeutumista, ja takaisin oksan päälle nousemista. Animaatio käyttää Root Motion -ominaisuutta, joka ohjaa hahmoa liikkeen aikana hieman taakse- ja eteenpäin X- ja Y- akseleilla. Säädin hahmon korjaamaan rotaationsa sivuttain oksan mukaisesti aina kyseisen animaation alkaessa, ja ohjasin hahmon liikkumaan tietyn etäisyyden alas- tai ylöspäin Z-akselilla, riippuen animaation suunnasta. Pienen hienosäädön jälkeen, sain liikkeen toimimaan varsin sulavasti yhdessä animaation kanssa. Hienosäädin myös takaisin puuhun hyppäämisen suorittavaa logiikkaa, lisäsin siihen hahmon Mesh objektin sijainnin korjaamisen, ja asetin hypyn loppupisteen sen mukaan, onko hahmon tarkoitus jäädä roikkumaan oksasta, tai kiivetä sen päälle. Säädin myös hahmon puusta putoamista, joka tapahtuu pelaajan lyödessä puuta, jonka oksalla apina istuu. Säädin iskusta tapahtuvaa lentorataa, ja jouduin lisäämään siihen manuaalisesti hieman lisää taaksepäin kompensaatiota, koska Root Motion -ominaisuus ei

liikuttanut hahmoa tarpeeksi pitkälle pudotakseen sulavasti. Luonnollisesti hienosäädin samalla animaatioiden siirtymäsääntöjä ja sekoitusaikoja.

Päivän työtehtävät onnistuivat hyvin. Sain toteutettua onnistuneesti oksalta laskeutumis-, ja takaisin kiipeämistapahtumat, sekä takaisin puuhun hyppäämisen suorittavan-, ja puusta putoamisen toteuttavan logiikan. Päivän osaamistavoitteet siis täyttyivät.

TO 21.9.2023

Tämän päivän työtehtäviini kuuluu apina hahmon uusien animaatioiden peliin implementoinnin jatkaminen. Päivän osaamistavoitteisiin kuuluu animaatioiden kanssa työskentely, ja niiden yhdistäminen olemassa oleviin pelimekaniikkoihin.

Apina hahmon työstäminen jatkui edelleen. Lisäsin jälleen käyttöön muutaman uuden animaation; oksasta kiinni ottaminen ja pökerryksissä oleminen puusta putoamisen jälkeen. Loin animaatiotilakoneeseen uudet tilat kyseisille animaatioille, ja loin siirtymät ehtoineen niiden välille. Päivitin samalla hieman Behavior Tree -käytöspuun toimintalogiikkaa; pelaajan hyökkäysalueen sisällä olo ei enää estä siirtymistä Investigate tilaan, kun apina hahmo ei näe pelaajaa. Aiemmin hyökkäysalueella oli suurempi prioriteetti, mutta muutoksen jälkeen se on tekoälyhahmon näköaistilla. Integroin oksasta kiinni ottamisen osaksi oksaa kohti hyppäämisen suorittavaa logiikkaa. Tein myös hieman pieniä korjauksia animaatioiden siirtymien sääntöihin, jotta ne toimivat sulavammin. Tämä pitää sisällään animaatioiden välisten sekoittumisten aikoja, ja ehtojen muuttamista tarkemmiksi. Pidimme myös perinteisen viikkopalaverin, jossa esittelimme tekemiämme työtehtäviä.

Päivän työtehtävät etenivät kohtalaisesti, vaikka muutama asia vaatii vielä hienosäätöä. Päivän osaamistavoitteista onnistuivat, sillä sain yhdistettyä lisää animaatioita toimimaan yhdessä pelimekaniikkojen kanssa.

PE 21.9.2023

Tämän päivän työtehtäviini kuuluu toivon mukaan viimeistellä apina hahmon animaatioiden implementointi, ja niihin kuuluvien mekaniikkojen hienosäätö. Päivän osaamistavoitteisiin kuuluu animaatioiden kanssa työskentely, ja niiden yhdistäminen olemassa oleviin pelimekaniikkoihin.

Säädin hahmon puusta putoamisen aloittavaa logiikkaa sen mukaan, onko hahmo oksan päällä, siirtymässä ylös tai alaspäin oksalla, tai onko hahmo jo roikkumassa oksan alla. Tilanteen mukaan hahmo suorittaa valitun animaation, jonka jälkeen jatkaa putoamistaan kohti maata. Lisäsin tapahtumaan myös tarkistuksen siitä, onko hahmolla kädessään heitettävä esine. Mikäli on, irrotetaan esine hahmon kädestä, ja pudotetaan se maahan. Hienosäädin myös hahmon kuolemismekaniikan, joka tapahtuu pelaajan lyödessä hahmoa onnistuneesti. Käänsin hahmon rotaation pelaajan suuntaan iskun tapahtuessa, jonka jälkeen hahmo lentää taaksepäin animaation Root Motion -ominaisuuden ja manuaalisen kompensaation yhdistelmällä. Lisäsin animaatiotilakoneeseen uuden Conduit solmun, jonka kautta on mahdollista siirtyä kuolemisanimaatioon. Kyseiseen Conduit solmuun taas on mahdollista siirtyä jokaisesta animaatiotilasta. Kävin vielä lopuksi läpi kaikki animaatiotilakoneen siirtymät, tarkistin ne, ja tein pieniä hienosäätöjä, mikäli tarvetta esiintyi. Siistin ja kommentoin hahmoon tekemäni ohjelmakoodit, jonka jälkeen dokumentoin tekemäni muutokset, ja lähetin kaiken versionhallintaa.

Päivän työtehtävät ja osaamistavoitteet onnistuivat hyvin. Sain viimeinkin kaikki apina hahmon animaatiot implementoitua pelimekaniikkoihin, ja hahmon toimimaan halutulla tavalla. Kyseisiin työtehtäviin meni paljon enemmän aikaa kuin alunperin ajattelin, josta olen hieman harmissani. Pääasia kuitenkin on, että hahmon mekaniikat toimivat, ja animaatiot on implementoitu onnistuneesti.

Seurantaviikko 12 – Viikkoraportti: 18.9.2023 – 22.9.2023

Tällä viikolla työskentelin vahvasti animaatioiden parissa. Otin apina hahmon uusia animaatioita käyttöön, ja yhdistin niihin pelimekaniikkoja. Sain myös

kertausta siihen, miten uusia animaatioita tuodaan onnistuneesti Unreal Engine -pelimoottoriin.

Apina hahmon heitettävien objektien kanssa sain paranneltua niiden heittologiikkaa hyödyntämällä Projectile Movement -komponenttia. Komponentin Rotation Follows Velocity -ominaisuus; joka saa objektin rotaation seuraamaan heittorataa, paransi sen liikettä ilmassa mukavasti. Heittoradan laskevan logiikan siirtäminen heitettävään objektiin oli myös mielestäni hyvä ratkaisu, koska heiton suorittavan liikkeen logiikka suoritetaan nyt vain yhdessä paikassa.

Animaatiotiedostojen lisääminen pelimoottoriin tapahtuu FBX-tiedostoina. FBX on Autodeskin omistama ja kehittämä tiedostomuoto, ja sitä käytetään yhteen toimivuuden tarjoamiseen digitaalisen sisällön luontisovellusten välillä. Vaikka FBX-tiedostojen tuominen Unreal Engine 4 -pelimoottoriin on suhteellisen yksinkertainen toimenpide, tuodun omaisuuden säätämiseen on olemassa useita vaihtoehtoja. Tuodessa uutta sisältöä Content valikon kautta FBX-liukuhihnan avulla, FBX-tuontiasetukset-valintaikkuna tulee näkyviin. Tuontiohjelma tunnistaa automaattisesti, millaista tiedostoa yritetään tuoda, ja säätää käyttöliittymänsä oikeaoppisesti. FBX-tuontiohjelmassa on käytettävissämme kaksi vaihtoehtoa. Ensimmäinen vaihtoehtomme on Import painike, jolla tuodaan sillä hetkellä valitun FBX-tiedoston. Toinen vaihtoehtomme on Import All -painike, jonka avulla voi tuoda kaikki sillä hetkellä valitut FBX-tiedostot. Uusi tuotu Skeletal Mesh -objekti ja tuodut animaatiot ilmestyvät Content valikkoon, mikäli prosessi suoritettiin onnistuneesti. (Epic Game 2023) Mikäli uutta Skeletal Mesh -objektia ei ole tarvetta tuoda, vain ainoastaan uudet animaatiot, voidaan Skeletal Mesh -valinta jättää tyhjäksi, jolloin tuontiohjelma tuo Content valikkoon ainoastaan tiedoston sisältämät uudet animaatiot.

Olen viikon saavutuksiin tyytyväinen, koska sain implementoitua kaikki apina hahmon animaatiot onnistuneesti, toimimaan yhdessä pelimekaniikkojen kanssa.

5.13 Seurantaviikko 13 (25.9.2023-29.9.2023 + 2.10.2023-4.10.2023)

MA 25.9.2023

Tämän päivän työtehtäviini kuuluu uimismekaniikan hienosäätäminen, ja tuntuman parantelu. Päivän osaamistavoitteisiin kuuluu olemassa olevan uimismekaniikan hienosäätäminen, ja hahmon liikkumisen tuntuman parantaminen veden alla liikuttaessa.

Päivän aikana muokkasin uimisen tuntumaan vaikuttavia arvoja melkoisesti. Säädin Braking Deceleration Swimming -arvon hahmon Character Movement -komponentista 0,1 -> 1000, joka saa vedenalaisen liikkeen pysähtymään lähes välittömästi sen jälkeen, kun pelaaja lopettaa syötteen antamisen. Kameran palautumisen interpolaatio ajan laskin puoleen alkuperäisestä arvosta. Vaihdoin vedenalaisen uimisen suunnan käyttämään Control Rotation -arvon sijaan hahmon Forward Vector- arvoa, jolloin hahmo liikkuu eteenpäin oman akselinsa mukaisesti, kameran akselin sijaan. Tämä saa aikaan sen, että kamera ei enää seuraa pelaajan jokaista liikettä. Sivusuuntaan kääntymisen määrittävän muuttujan vaihdoin käyttämään suoraa -1 tai 1 arvoa, sen sijaan että käytettäisiin desimaali arvoja. Lisäsin myös interpolaation kameran Target Arm Length -arvolle, jota ei aiemmin säädetty veden alla ollenkaan. Koska arvoa ei säädetty ollenkaan, oli mahdollista, että kamera saattoi jäädä jumiin todella lähelle pelaajaa. Nyt kyseinen ongelma on kuitenkin korjattu. Kaikkien tekemieni muutosten jälkeen vedenalaisen uimisen tuntuma parani huomattavasti, ja tekee liikkumisesta helpompaa ja tarkempaa. Hahmon liike tuntui aiemmin melko liukkaalta, joka sai erityisesti ahtaissa tiloissa uimisen tuntumaan hankalalta.

Päivän työtehtävät onnistuivat hyvin, ja osaamistavoitteet täyttyivät. Sain hienosäädettyä hahmon vedenalaisen uimismekaniikan tuntumaa huomattavasti, ja tehtyä liikkumisesta veden alla tarkempaa ja sulavampaa.

TI 26.9.2023

Tämän päivän työtehtäviini kuuluu uimismekaniikan hienosäätämisen, ja tuntuman parantelun viimeistely, partikkeliefektin lisääminen sukeltamiseen, ja seinäkiipeilyn hienosäätämisen aloittaminen. Päivän osaamistavoitteisiin kuuluu olemassa olevan uimismekaniikan hienosäätämisen viimeistely, ja hahmon liikkumisen tuntuman parantaminen veden alla liikuttaessa. Tämän lisäksi tavoitteisiin kuuluu partikkeliefektin hallinta, ja seinäkiipeilymekaniikan kanssa työskentely.

Tein vielä pari pientä säätöä vedenalaiseen uimiseen vaikuttavaan tuntumaan. Säädin hahmon rotaatioon vaikuttavan interpolaatio ajan vakio arvoksi 2, sen sijaan että käytettäisiin vaihtuvaa arvoa. Lisäsin myös nopeamman uimisen aloittavan funktion aluksi hahmon liikkeen pysäyttämisen, jolloin nopean uimisen aloittavan ponkaisun suuntaan on helpompi vaikuttaa. Hienosäätöjen tekemisen lisäksi lisäsin hahmolle myös partikkeliefektin, joka käynnistetään hahmon siirtyessä veden alle, ja poistetaan käytöstä hahmon tullessa takaisin pinnalle. Kyseinen partikkeli efekti saa kuplia tulemaan hahmon suusta, jonka sijainti on määritelty hahmon Skeletal Mesh -objektiin lisäämäni Socket sijainnin mukaan (kuva 6).

Kuva 6. Longtail hahmo uimassa veden pinnan alla Longtail pelissä (Kuva 7: Hologram Monster).

Seuraavaksi aloin työstää seinäkiipeilymekaniikan hienosäätämistä, ja uusien animaatioiden implementointia. Aluksi vaihdoin uudet animaatiot Blend Space -animaatio-objektiin, joka yhdistää eri suuntiin tapahtuvat kiipeämisanimaatiot. Poistin myös animaatioiden alusta ylimääräiset animaatiokehykset, joita ilmestyy joskus animaatioiden pelimoottoriin tuomisen yhteydessä. Kyseisten animaatiokehysten aikana satunnaiset hahmon luut, esimerkiksi jalat, saattavat olla väärässä paikassa, eivätkä täsmää animaation viimeiseen kehykseen. Näiden kehysten poistamisen jälkeen animaatioiden aloitus- ja lopetuskehykset täsmäävät jälleen, ja animaatioita on mahdollista ketjuttaa. Ehdin myös kasvattaa seinäkiipeilyn nopeutta.

Päivän työtehtävät ja osaamistavoitteet onnistuivat hyvin. Sain uimismekaniikkaan tehdyt hienosäädöt valmiiksi, ja paranneltua sen tuntumaa. Lisäsin partikkeliefektin onnistuneesti hahmolle, tämän ollessa veden alla Ehdin myös aloittaa hieman seinäkiipeily mekaniikan hienosäätämistä, ja uusien animaatioiden implementointia.

KE 27.9.2023

Tämän päivän työtehtäviini kuuluu eilen aloittamani seinäkiipeilymekaniikan hienosäätämisen jatkaminen. Päivän osaamistavoitteisiin kuuluu olemassa olevan seinäkiipeilymekaniikan hienosäätäminen ja tuntuman parantelu, sekä uusien seinäkiipeilyyn liittyvien animaatioiden implementointi.

Ensiksi loin uuden funktion, joka tarkistaa pelaajan antaman syötteen, ja määrittelee sen perusteella annetun syötteen suunnan pelimaailmassa enumeraatio arvona. Syötin Make Vector -solmulle hahmon Move Forward- ja Move Right -arvot X- ja Y-akseleihin, ja otin luodusta vektorista Get Yaw Pitch From Vector -arvon. Määrittelin kyseisen arvon 90 asteen kvarteeleihin, jonka avulla saan syötteen suunnan pelimaailmassa. Loin animaatiotilakoneeseen uuden Blend Poses By Enum -solmun, joka valitsee annetun syötteen mukaan seinäkiipeilyn aikana tapahtuvan hypyn suunnan. Hahmon sivuille loin raycast tarkistukset, jotka tarkistavat onko sivuilla tarpeeksi tilaa suorittaa hyppy. Mikäli

hyppy on mahdollista suorittaa, käynnistetään hyppyanimaatio, jonka aikana laukaistaan Animation Notify -solmu, joka käynnistää sivusuunnassa liikkeen suorittavan funktion. Loin uuden animaatiotilan seinää pitkin liukumiselle, joka on mahdollista aloittaa painamalla hyppy nappia alaspäin. Tilaa varten loin uuden muuttujan hahmon Action Struct -solmuun, ja hyödynsin sitä tilan sääntöjen määrittämiseen. Lisäsin alaspäin tapahtuvan liikkeen suorittavan logiikan, kuin myös liikkeen lopettavan tapahtuman pelaajan syötteen päätyttyä. Sain vietyä seinäkiipeily mekaniikkaa oikein mukavasti eteenpäin.

Päivän työtehtävät etenivät hyvin, ja osaamistavoitteet täyttyivät. Sain vietyä seinäkiipeilymekaniikkaa, ja siihen liittyviä seinällä suoritettavia hyppyjä mukavasti eteenpäin, ja implementoitua uusia animaatioita käyttöön mekaniikkojen kanssa. Päivän osaamistavoitteet siis onnistuivat tavoitteiden mukaisesti.

TO 28.9.2023

Tämän päivän työtehtäviini kuuluu eilen aloittamani seinäkiipeilymekaniikan hienosäätämisen jatkaminen. Päivän osaamistavoitteisiin kuuluu olemassa olevan seinäkiipeilymekaniikan hienosäätäminen ja tuntuman parantelu, sekä uusien seinäkiipeilyyn liittyvien animaatioiden implementointi.

Hahmon sivuille on jo aiemmin toteutettu raycast tarkistukset, jotka tarkistavat löytyykö hahmon sivuilta lisää kiivettävää seinää. Päivitettyjen seinäkiipeilyanimaatioiden myötä minun tarvitsi säätää raycast tarkistusten etäisyyttä kauemmaksi hahmosta, jotta hahmon kädet ja jalat eivät mene seinien pintojen ulkopuolelle. Säädettyäni arvoja kyseisestä tarkistuksesta, minun tuli säätää myös seinien kulmissa suoritettavia tarkastuksia pidemmälle, koska hahmon liike pysäytetään jo aiemmassa vaiheessa. Lisäsin kulmissa tapahtuvien käännösten animaatioihin Animation Notify -solmut, jotka käynnistävät funktion, joka antaa hieman lisää Offset arvoa hahmon liikkeeseen. Tällaisilla yhdistelmillä sain seinäkiipeilyn reunojen ja kulmien etäisyydet säädettyä sopivan mittaisiksi. Lisäsin hahmolle myös uuden tilan, jonka aikana hahmo nojaa taaksepäin ollessaan seinäkiipeily tilassa, pelaajan

painaessa ohjaimen oikeaa liipaisin painiketta. Tämä aktivoi Action Struct -solmuun lisäämäni muuttujan. Mikäli pelaajan painaa hyppy nappia muuttujan ollessa aktiivinen, suoritetaan aiemmin toteutettu Wall Jump kyky, joka saa hahmon hyppäämään pois päin seinästä. Taaksepäin nojaamiselle lisäsin uuden tilan myös animaatiotilakoneeseen, jonka välille loin tarvittavat siirtymäehdot. Minun tuli myös huomioida se, että ohjaimen oikealla liipaisin painikkeella on mahdollista suorittaa sekä Crouch- että Air Dash -kyvyt, joka saa molemmat funktiot laukeamaan nappia painettaessa. Lisäsin siis tarvittavat tilannekohtaiset tarkistukset molempien kykyjen funktioihin.

Päivän työtehtävät onnistuivat hyvin. Sain paranneltua ja laajennettua seinäkiipeily mekaniikkaa entisestään, otettua uusia animaatioita käyttöön. Päivän osaamistavoitteisiin siis päästiin myös erinomaisesti.

PE 29.9.2023

Tämän päivän työtehtäviini kuuluu eilen aloittamani seinäkiipeilymekaniikan hienosäätämisen jatkaminen, sekä uuden peli-idean miettiminen yhdessä tiimin kanssa. Päivän osaamistavoitteisiin kuuluu olemassa olevan seinäkiipeilymekaniikan hienosäätäminen ja tuntuman parantelua, uusien seinäkiipeilyyn liittyvien animaatioiden implementointia, sekä ideoimista uudesta peli-ideasta.

Tänään hienosäädin ja parantelin seinäkiipeilyn alaspäin tapahtuvaa liikettä, ja hyppäämistä. Lisäsin alaspäin tapahtuvaan hyppyyn raycast tarkistuksen hahmon alapuolelle. Mikäli alapuolelta löytyy tarpeeksi tyhjää tilaa, voidaan hyppy suorittaa. Alaspäin tapahtuvan hypyn animaatioon lisäsin Animation Nofity -solmun, joka käynnistää Wall Slide -funktion, joka alkaa suorittaa alaspäin tapahtuvaa liikettä. Kyseisen funktioon lisäsin myös uuden raycast tarkistuksen, joka tarkistaa milloin seinä loppuu hahmon alapuolelta. Mikäli seinä päättyy, alaspäin tapahtuva liike lopetetaan, ja palataan takaisin tavalliseen seinäkiipeily tilaan. Meillä tuli tiimimme kanssa puheeksi, että peliin on toteutettu myös jonkinlainen katto kiipeily mekaniikka, ja olisi hienoa, jos seinällä liukumisesta olisi mahdollista siirtyä suoraa katto kiipeily tilaan. Ehdin

tehdä jo esiversion tällaisesta mekaniikasta, lisäämällä aiemmin mainitsemaani funktion loppuun uuden raycast tarkistuksen. Kyseinen tarkastus suoritetaan hieman seinän alapuolelta ylöspäin kohti mahdollista kattoa. Mikäli tarkistus havaitsee kiivettävän kattopinnan, siirtyy hahmo suoraa katto kiipeily tilaan. Kyseiselle siirtymälle ei kuitenkaan ole vielä animaatiota, joten minun tulee päivittää funktiota sitten, kun saan siihen tarvittavat animaatiot. Pidimme iltapäivällä myös mainitsemani suunnittelupalaverin uudesta peli-ideasta. Siitä en kuitenkaan voi valitettavasti kertoa tässä raportissa enempää.

Päivän työtehtävät onnistuivat hyvin, sillä sain seinäkiipeilymekaniikan toiminnallisuutta vietyä taas onnistuneesti eteenpäin. Päivän osaamistavoitteisiin päästiin, koska työstämäni mekaniikka eteni, sekä suunnittelupalaveri tiimimme kesken oli onnistunut.

MA 2.10.2023

Tämän päivän työtehtäviini kuuluu eilen aloittamani seinäkiipeilymekaniikan hienosäätämisen jatkaminen. Päivän osaamistavoitteisiin kuuluu olemassa olevan seinäkiipeilymekaniikan hienosäätäminen ja tuntuman parantelu, sekä uusien seinäkiipeilyyn liittyvien animaatioiden implementointi.

Vaihdoin muutaman animaation päivitettyihin versioihin, ja säädin kyseisiin animaatioihin liittyviä funktioita toimimaan niiden kanssa. Tämä pitää sisällään muun muassa Transition Rule -aikoja, niihin liittyviä ehtoja, ja funktioita käynnistäviä Animation Notify -solmuja. Huomasin, että taaksepäin tapahtuvan seinähypyn animaatiosta puuttui Root Motion -ominaisuus. Tämä aiheutti ongelmia, koska animaatioiden vaihtumisen ja hypyn aloittavan koodin kanssa esiintyi nykimistä hahmon liikkeessä. Root Motion -ominaisuus on tärkeää olla päällä aina, jos hahmon animaatiot saavat sen hahmon “astumaan ulos” sille määritetystä nollapisteestä. Hahmon seinästä pois päin hyppäämisen aloittavaan funktioon lisäsin Timer ajastimen, jonka käynnissä ollessa hahmo ei vastaanota seinäkiipeilyyn liittyvää liike syötettä. Tämä varmistaa sen, että pelaajan antama syöte ei häiritse hyppyä sen jälkeen, kun se on aloitettu. Lisäsin myös uuden tarkistuksen hahmon kiivetessä kohti yläpuolella olevaa

kiinni otettavaa reunaa. Kyseinen tarkistus siirtää hahmon suoraa reunasta kiinni pitämistilaan ja animaatioon, havaitessaan reunan hahmon yläpuolella.

Päivän työtehtävät ja osaamistavoitteen onnistuivat ihan kohtalaisesti. Seinästä pois päin hyppäämismekaniikassa oli hieman haasteita, mutta se johtui loppujen lopuksi Root Motion -ominaisuuden puuttumisesta animaatiossa.

TI 3.10.2023

Tämän päivän työtehtäviini kuuluu projektin tiedostojen palauttaminen versionhallintaan. Päivän osaamistavoitteisiin kuuluu projektin hallintaa, ja versionhallinnan kanssa työskentelyä.

Tiimimme jäsen loi viikonlopun aikana lisää tilaa versionhallintamme pilveen, ja poisti vanhoja merkintöjä versionhallinnan alkupuolelta. Tämän operaation jälkeen suurin osa projektin tiedostoista oli saanut keltaisen kysymysmerkkimerkinnän, joka tarkoittaa, että kyseisiä tiedostoja ei ole ladattu versiohallintaan. Tämä johtui mitä ilmeisimmin vanhojen merkintöjen poistamisesta. Minun tarvitsi siis ladata takaisin versionhallintaan kaikki tiedostot, jotka sieltä puuttuivat. Prosessi ei ollut haastava, mutta se on erittäin pitkäkestoinen, sillä ladattavia tiedostoja oli yli 9000. Valitsin pelimoottorin Content valikosta Not Source Controlled -filtterin, ja kävin kategoriat läpi yksitellen, esimerkiksi Bluerprints, Textures, Animations, ja niin edelleen. Merkkasin jokaisen kysymysmerkillä merkatun tiedoston Mark For Add -komennolla, jonka jälkeen sain siirrettyä ne Perforce ohjelman kautta versionhallintaa. Prosessiin kului koko työpäivä, ja muutama kategoria jäi vielä odottamaan seuraavaa päivää.

Päivän työtehtävät onnistuivat, koska sain vietyä lähes kaikki puuttuvat tiedostot takaisin versionhallintaa. Päivän osaamistavoitteet toteutuivat myös, koska tiedostojen siirtäminen onnistui ongelmitta.

KE 4.10.2023

Tämän päivän työtehtäviini kuuluu Signal Plant -vihollishahmojen toiminnallisuuden muuttaminen hyödyntämään Level Sequence -objekteja. Päivän osaamistavoitteisiin kuuluu Level Sequence -objektien sisällyttäminen ja yhdistäminen vihollishahmon toiminnallisuuteen.

Signal Plant -hahmojen aiempi toteutus käynnisti niihin liitettyjen puun juurien animaation, joiden tarkoitus oli estää pelaajan kulku kentän valituissa sijainneissa. Sain kenttä suunnittelijaltamme toiveen, että Signal Plant -hahmot voisivat käynnistää Level Sequence -objekteja, puun juurten animaatioiden sijaan. Hahmojen mahdollisuus kontrolloida Level Sequence -objekteja, tekisi hahmoista paljon monikäyttöisempiä, koska Level Sequence -objektin aikana on mahdollista käsitellä ja liikuttaa mitä vain kentän objekteja. Lähdin toteuttamaan asiaa komponentin kautta. Komponentti liitetään kentässä olevaan Level Sequence -objektiin, ja se sisältää Array muuttujan, johon halutut Signal Plant -hahmot voidaan lisätä. Signal Plant -hahmoihin taas loin muuttujan haluttua Level Sequence -objektia varten. Loin komponenttiin funktiot, jotka tarkistavat kasvien tekoälyn tilan, ja pyörittävät Level Sequence -objektien aikajanaa joko eteen- tai taaksepäin. Pelin käynnistyessä Signal Plant -hahmot hakevat niihin liitetyn Level Sequence -objektin muuttujan kautta, siihen liitetyn komponentin, joka mahdollistaa funktioiden käyttämisen. Itse funktiot käynnistetään hahmon tekoälyn tilojen vaihtuessa.

Päivän työtehtävä ja osaamistavoitteet onnistuivat erinomaisesti. Sain luotua Signal Plant -hahmoille erittäin toimivan ja helppokäyttöisen toteutuksen Level Sequence -objektien hallitsemiseen. Tämä teki hahmojen toiminnasta monipuolisempaa, eikä niiden toiminta ole sidottu ainoastaan etukäteen tietyllä tavalla animoituihin objekteihin.

Seurantaviikko 13 – Viikkoraportti: 25.9.2023 – 4.10.2023

Tämän viikkoraportti kattaa tavallisen viiden päivän sijaan kahdeksan päivää. Tämä menettely on tehty siksi, että 65 kokonaisen raportointipäivän

saavuttamiseen, minulta puuttui kolme raportointipäivää kuluneen 13 viikon ajalta. Raportointiviikolla 2 minulla kertyi kokonaisia raportointipäiviä vain neljä, koska perjantai 23.6.2023 oli meillä vapaapäivä. Raportointiviikolla 11 minulla kertyi kokonaisia raportointipäiviä vain kolme, koska Pocket Gamer Connect -tapahtuma aiheutti muutoksia tavalliseen työviikkoon.

Tällä viikolla työskentelin Longtail hahmon uimis- ja seinäkiipeilymekaniikkojen parissa, suoritin versionhallintaan liittyviä työtehtäviä, ja tein muutoksia Signal Plant -hahmon toiminnallisuuteen. Uimismekaniikan kanssa tein paljon pieniä arvojen muutoksiin liittyviä hienosäätöjä, jotka vaikuttivat uimisen tuntumaan merkittävästi. Seinäkiipeilymekaniikan kanssa työskentely piti sisällään paljon eri suuntiin suoritettavien raycast tarkistusten hallintaa, ja niistä saatujen tulosten käsittelyä.

Uusien seinäkiipeilyanimaatioiden kanssa jouduin tekemään pieniä korjauksia animaatioiden alkuun. Jostain syystä joihinkin animaatioihin ilmestyy pelimoottoriin tuotaessa erikoisia ylimääräisiä frameja, eli kuvia animaation alkuun. En ole saanut selville, onko tämä jokin bugin aiheuttama efekti, vai onko tuontiasetuksissa tai tuontitiedostossa jotain häikkää. Kyseisten ylimääräisten kuvien aikana hahmon luiden sijainnit ovat hieman missä sattuu, jolloin animaatiota pyörittäessä toistuvasti, niiden alussa ilmenee häiritsevä nytkähdys. Tämä on helppo korjata poistamalla kyseisen animaation alusta ylimääräiset kuvat, jotka yleensä kestävät noin 1–5 kuvan verran. Tämän jälkeen animaatio toistuu taas normaalisti ilman nytkähdyksiä. Toki tämä tuottaa ylimääräistä vaivaa, ja ongelmaan olisi hyvä löytää pysyvä ratkaisu.

Versionhallintaan liittyvät työtehtävät olivat puuttuvien tiedostojen palauttamista takaisin versionhallintaan. Tiimimme jäsen poisti Perforce työkalulla vanhempia merkintöjä versionhallinnan alkupuolelta, vähentääkseen repositoriossa olevan datan tiedostokokoa. Tämä sai kuitenkin aikaan sen, että kauan sitten tehdyissä suorituksissa olleet tiedostot eivät välttämättä enää olleet enää repositoriossa. Kyseisiin tiedostoihin olikin pelimoottorin Content valikossa ilmestynyt kysymysmerkki-ikoni (kuva 7)

Kuva 7. Objektien eri tiloja kuvaavat ikonit Unreal Engine -pelimoottorissa. (Kuva: Epic Games).

Minun piti siis käydä läpi kaikki tiedostot, jotka eivät olleet repositoriossa, ja lisätä ne sinne uudelleen. Hyödynsin Content valikon suodattimia tiedostojen löytämiseen. Suodattimet tarjoavat nopean tavan rajata sisältöselaimen sisältönäkymässä näkyviä sisältötyyppejä. Jokainen suodatin edustaa joko tiettyä sisältötyyppiä tai lisäehtoja, joita voidaan käyttää rajoittamaan sisältöselaimessa näkyvien sisältötyyppien määrää. (Epic Games 2023). Merkkasin tiedostot Check Out -merkinnällä, ja siirsin ne takaisin repositorioon Perforce ohjelman kautta. Prosessi ei ollut vaikea, mutta todella pitkäkestoinen, sillä kaikkien tiedostojen siirtämiseen kului paljon aikaa.

Signal Plant -hahmojen toiminnallisuuden muutin käyttämään Level Sequence -objekteja. Kyseinen päivitys teki hahmoista paljon monikäyttöisempiä, kuin aiempi tiettyjen Actor-objektien animaatioita käyttävä toteutus. Sequencer-sekvensseri voidaan liittää blueprintiin sekvenssin ohjaamiseen pelin aikana. Sitä on mahdollista toistaa, keskeyttää, pysäyttää ja hypätä tiettyyn aikaväliin, ja jopa sitoa raitoja muihin kentässä oleviin Actor-objekteihin. Sekvenssejä voidaan lisätä myös Blueprinteihin käyttämällä Actor Sequence -komponenttia. Tämä mahdollistaa sekvenssien uudelleenkäytön sitomalla animaatiot Blueprint-instanssiin ja käynnistämällä ne automaattisesti tai tapahtumakaavion kautta. (Epic Games 2023)

6 Pohdinta

6.1 Työtehtävistä suoriutuminen, osaamisen kehittyminen ja arviointi

Kolmentoista viikkoa kestäneen tarkastelujakson aikana pidin päiväkirjaa päivittäisistä työtehtävistäni, ja niiden suorittamisesta peliohjelmoijana 3D-tasohyppelypelin kehitysprojektissa. Tarkastelujakson aikana seurasin ammatillista kehitystäni, työtehtävien etenemistä ja niissä kehittymistä, sekä yleistä tietojen ja taitojen karttumista peliohjelmoijana. Annoin itselleni jokaiselle päivälle erilaisia osaamistavoitteita työtehtävään viitaten, ja seurasin näiden osaamistavoitteiden täyttymistä ja niissä kehittymistä.

Arvioin ennen tarkastelujakson aloittamista osaamistasoni olevan tasolla Taitava suoriutuja määrittämissäni työtehtävien kategorioissa. Kriteeri kyseisen osaamisen tasolle on syvällinen ymmärtäminen työtehtävistä, ja suoriutumisen olevan jatkuvasti työtehtävien vaatimusten tasolla. (Karelia-ammattikorkeakoulu 2021, 2.) Koin kartuttaneeni harjoitteluni- ja puolen vuoden työkokemukseni aikana taitojani sen verran, että voin sijoittaa itseni kyseiseen kategoriaan. Tavoitteeni tämän kolmentoista viikon tarkastelujakson aikana oli saada nostettua osaamiseni Kokenut asiantuntija -tasolle suurimmassa osassa työtehtävistä. Tämä tasoa eroaa edellisestä siten, että kokenut asiantuntija kykenee kehittämään työtehtävässä vaadittavia toimintamalleja, ja pystyy ohjaamaan tai opastamaan muiden toimintaa. (Karelia-ammattikorkeakoulu 2021, 2.)

Kolmentoista viikon aikana minulla oli lukuisia eri työtehtäviä, ja osaamistavoitteita. Tämän tarkastelujakson aikana työskentelyni painottui pääasiassa eri pelihahmojen mekaniikkojen työstämiseen, animaatioiden implementointiin ja hahmojen tekoälyn kanssa työskentelyyn. Loin myös uusia toimintaratkaisuja toteuttamieni komponenttien muodossa, otin haltuun uusia

työkaluja kuten Control Rig ja FMOD, ja ehdin jopa osallistua yhteen pelialan tapahtumaan. Siinä missä osaamiseni karttui huomattavasti näillä osa-alueilla, työtehtäviini ei kuulunut tarkastelujakson aikana dialogijärjestelmien tai UI-elementtien kanssa työskentelyä. On siis luonnollisesti selvää, että en päässyt kehittämään osaamista näiden osa-alueiden parissa, jolloin ne jäivät Taitaja suoriutuja -tasolle.

Työtehtävien osa-alueet, joiden parissa työskentelin tarkastelujakson aikana eniten, olivat Blueprint-ohjelmointi, animaatioblueprintit, ja tekoäly. Loin tarkastelujakson aikana peliin kaksi täysin uutta hahmoa, joista toinen oli tekoälyn ohjaama vihollishahmo, ja toinen pelaajan ohjaama hahmo. Pelaajan ohjaaman hahmon kohdalla tälle piti luoda täysin uusi Character- ja animaatioblueprint. Vihollishahmon kanssa piti luoda character- ja animaatioblueprintien lisäksi vielä AI-Controller ja käytöspuu. Molemmille hahmoille loin niille suunnitellut toiminnallisuudet täysin alusta loppuun. Ehdin myös aloittaa kolmatta uutta hahmoa, mutta sitä en ehtinyt työstää tarkastelujakson aikana valmiiksi. Kehitin tarkastelujakson aikana monta täysin uutta ja omaa ratkaisua erilaisten pelimekaniikkojen toteuttamiseen, muun muassa seurantaviikoilla 2–3 luomani apina hahmon yleinen toiminnallisuus sekä viikoilla 4–5 toteuttamani possu hahmon Burrowing kaivautumismekaniikka. Lisäksi luomani uudet komponentit ovat jokainen monikäyttöisiä, ja niiden toimintaa on helppo jatkokehittää. Kehitin ja hienosäädin montaa pelissä ollutta mekaniikkaa, ja sain niiden toiminnallisuutta joko paranneltua, tai laajennettua. Ohjeistin tarkastelujakson aikana myös paljon yrityksessämme ollutta harjoittelijaa, jolla ei ollut aiempaa ohjelmointikokemusta. Ohjeistin häntä erilaisissa ohjelmointipuolen asioissa, ja opastin häntä Unreal Engine -pelimoottorin käytössä. Integroimme muuan muassa yhdessä projektiimme FMOD-ohjelman, jonka kautta aloimme kontrolloida pelin äänimaailmaa. Olen vahvasti sitä mieltä, että voin sanoa osaamiseni kehittyneen näiden työtehtävien osa-alueella Kokenut asiantuntija -tasolle.

Tarkastelujakson aikana kehitin uusia ratkaisumalleja muun muassa erilaisten komponenttien muodossa. Esimerkiksi luomani osuman tunnistus -komponentti

on todella monikäyttöinen, ja mahdollistaa pelaajan lyönnistä aiheutuneen osuman käsittelyn mille tahansa objektille. Pelikentistä löytyy usein erilaisia geneerisiä objekteja, kuten puita, kiviä tai vaikka aitoja, joille ei ole tarvetta tehdä erillistä toiminnallisuutta sisältävää blueprintiä. Luomani komponentti voidaan liittää esimerkiksi tällaisiin objekteihin. Komponentin liittämisen jälkeen omistajaobjektinsa tunnistaa siihen kohdistuneet pelaajan lyönnit automaattisesti, ja esimerkiksi soittaa sopivan äänen, objektin fyysisen pinnan materiaalin mukaan (puu pinnasta puisen kuuloinen ääni, ja niin edelleen), tai käynnistää halutun partikkeliefektin. Komponenttiin on helppo lisätä myös uusia toiminnallisuuksia tarpeen mukaan myöhemmin. Olen havainnut erilaisia ratkaisuja toteuttaessani, että tämänkaltaiset monikäyttöiset ratkaisumallit ovat aina parempia, kuin yksittäiset niin sanotusti “kovakoodatut” ratkaisut, joiden laajentaminen jälkikäteen saattaa olla hankalaa. Tavoitteeni kyseisessä työskentelymallissa on yrittää tehdä asioista mahdollisimman helppoa kenttäsuunnittelijalle, ja luoda hänen työskentelylleen näppärät ja helppokäyttöiset työkalut.

Päiväkirjatyöskentelyn aikana kirjasin jatkuvasti ylös tekemiäni asioita työtehtävissä. Vaikka tein sitä pääsääntöisesti päiväkirjan kirjoittamisen vuoksi, huomasin sen olevan varsin hyvä tapa jäsennellä omaa työntekoa, ja seurata tekemieni työtehtävien etenemistä. Olenkin jatkanut tätä hyväksi todettua ratkaisumallia myös päiväkirjantyöskentelyn jälkeen, ja se on helpottanut muuan muassa asioiden esittelyä viikkopalavereissa.

Kuten edellä mainitsin, työtehtävien etenemisen ylös kirjaaminen itselleni on helpottanut omaa työskentelyäni merkittävästi. Havaitsin eri vaiheiden ja käytettyjen ratkaisujen ylös kirjaamisen työskentelyn aikana helpottaneen keskeneräisiin työtehtäviin palaamista, ja niiden dokumentointia työtehtävän valmistuttua. Tämä helpotti myös viikkopalavereihin valmistautumista. Työtehtävä, jossa käsitellään useaa eri asiaa ja toiminnallisuutta, kertyy helposti todella laajaksi kokonaisuudeksi. Nämä kokonaisuudet saattavat sisältää viittauksia eri sijainteihin ohjelmakoodissa, ja olla riippuvaisia monesta eri objekteista ja työkaluista. Tämän kaltaisen kokonaisuuden jäsentely niin työn aikana, kuin jälkikäteen on huomattavasti helpompaa, kun työn eri vaiheet on

kirjattu ylös niiden edetessä. Tätä toimintatapaa aion ehdottomasti jatkaa myös tulevaisuudessa.

Tarkastelujakson aikana suoritin myös viimeisiä meneillään olevia opintoja. Kursseilla hyödynnettiin C++ koodia blueprint-ohjelmoinnin tukena. Pelihahmomme lyöntien osumatarkasteluja suorittava komponentti oli myös toteutettu C++ avulla. Huomasin kyseisten ratkaisujen olevan todella toimivia ja tehokkaita erityisesti pelin performanssin kannalta. Haluaisin hyödyntää jatkossa enemmän C++ toteutuksia kehittämissäni toteutuksissa ja oman työskentelyni tukena. Olen hankkinut jo valmiiksi itselleni kursseja aiheesta, joita aion suorittaa itsenäisesti opintojeni päättymisen jälkeen. Tällä tavalla aion pitää yllä uusien asioiden oppimista, ja jatkokehittää ammatillista osaamistani myös jatkossa.

Tarkastelujakson aikana työskentelyni tärkeimmät aihealueet olivat pelihahmojen mekaniikkojen työstäminen, animaatioiden implementointi, ja hahmojen tekoälyn kanssa työskentelyn. Muutoksia aihealueissa tapahtui jonkin verran. Animaatioiden parissa työskennellessä huomasin ymmärrykseni niiden parissa työskentelystä parantuneen. Opin esimerkiksi seurantaviikolla 7 tunnistamaan paremmin, missä tilanteissa animaatio tarvitsee Root Motion -ominaisuuden, ja miten sen käyttöä tulee hallita. Opin paremmin pyytämään muilta tarvitsemiani ominaisuuksia, kuten vaikka mainitsemaani Root Motion -ominaisuutta. Aiemmin olen saattanut yrittää ratkaista ongelmaa täysin itse, tai yrittää kompensoida jonkin ominaisuuden puuttumista koodin avulla.

Huomasin päiväkirjatyöskentelyn vaikuttaneen työtehtävistäni suoriutumiseen sekä positiivisesti, että negatiivisesti. Tieto siitä, että kaikki tarkastelujakson aikana suorittamani työtehtävät tulevat olemaan mukana opinnäytetyössäni, motivoi minua suoriutumaan niistä mahdollisimman hyvin ja tehokkaasti. Halusin päästä esittelemään osaamistani laajasti, ja pyrkimään työtehtävien tehokkaaseen etenemiseen. Koin päiväkirjatyöskentelyn tuovan sopivaa painetta itseni haastamiseen, sekä parempien ratkaisujen kehittämiseen. Se sai minut myös pitämään aiempaa tarkempaa kirjaa työtehtävieni etenemisestä, ja niissä käytettyjen ratkaisujen avaamisesta. Päiväkirjatyöskentelyssä oli

kuitenkin myös kääntöpuoli. Huomasin tunnistaneeni itsessäni ylimääräistä stressiä, ja epäonnistumisen tunnetta, mikäli johonkin työtehtävään kului paljon aikaa, tai kokeilemani toteutukset eivät heti onnistuneet. Ohjelmointityö on usein erilaisten ratkaisujen kokeilemista, ja niistä parhaaksi koetun tavan soveltamista, tai joskus jopa kompromissien tekemistä paremman ratkaisumallin puuttuessa. Huomasin joidenkin työtehtävien aikana minulla olevan välillä hieman vaikeaa keksiä kerrottavaa aiheesta, mikäli siihen tehdyt muutokset eivät näkyneet niin sanotusti päälle päin yhtä paljon kuin minulle itselleni. Tämä oli paikoitellen hieman stressaavaa, ja sai minusta itsestäni tuntumaan, että työtehtävä ei etene “tarpeeksi” nopeasti ja konkreettisesti. Suoritin päiväkirjatyöskentelyn loppupuolella samaan aikaan myös opintoja, jolloin päiväni koostuivat lähes täysin vain työstä, päiväkirjan kirjoittamisesta ja opinnoista, joka oli hyvin kuormittavaa. Huomasin tämän vaikuttavan myös työtehtävien etenemiseen, koska palautumiselle ei ollut tarpeeksi aikaa, jonka seurauksena ajatus ei kulkenut yhtä hyvin kuin tavallisesti. Loppujen lopuksi koen päiväkirjatyöskentelyn kuitenkin vaikuttaneen työtehtävistä suoriutumiseen opinnäyteprosessin aikana enemmän positiivisesti, kuin negatiivisesti. Päiväkirjatyöskentelyn seurauksena minulle kehittyi uusia hyviä toimintatapoja, joita aion noudattaa myös jatkossa. Koen sekä positiivisten, että negatiivisten kokemusten tunnistamisen ja käsittelemisen tärkeäksi osaksi ammatillisen kasvun prosessia.

Opinnäytetyön tietoperustan tieto- ja tutkimuskirjallisuus tuki työskentelyäni eri tavoin opinnäytetyön eri vaiheissa. Opinnäytetyön suunnittelua ja alkuanalyysia, sekä työn pohdintaosiota tuki hyvin Joshua Bycerin (2019) teos Game Design Deep Dive: Platformers, koska se sisälsi asioiden kuvaamista ja analysointia erilaisten peliesimerkkien kautta. Kun taas itse päiväkirjaosuuden teknisen työn toteuttamisessa apuna toimi paremmin Epic Gamesin viralliset Unreal Engine -pelimoottorin dokumentaatiot. Ajattelin ennen päiväkirjatyöskentelyn aloittamista, pystyväni hyödyntämään Bycerin teosta myös päivätyöskentelyn tukena. Tarkastelujaksolle osuneet työtehtävät olivat kuitenkin enemmän teknisempiä, jolloin Epic Gamesin materiaali toimi paremmin työskentelyn tukena. Epic Gamesin materiaalin avulla pystyin avaamaan tiettyjen

pelimoottorin ominaisuuksien toimintaa, ja kuvaamaan toteuttamiani ratkaisuja niiden avulla.

Opinnäytetyöprosessin aikana pyrin vertaamaan omaa toimintaani, työskentelyäni ja valitsemiani toimintatapoja, sekä niiden käytänteitä hyviin tieteellisiin käytäntöihin. Hyvät tieteelliset käytännöt pitävät sisällään rehellisyyden, yleisen huolellisuuden ja tarkkuuden tutkimustyössä, tulosten tallentamisessa ja esittämisessä sekä tutkimusten ja niiden tulosten arvioinnissa. Ennen tutkimuksen aloittamista sovitaan myös oikeudet, tekijyyttä koskevat periaatteet, vastuut ja velvoitteet. (Tutkimuseettinen neuvottelukunta. 2023.) Päiväkirjatyöskentelyn aikana pyrin kertomaan työtehtävistäni ja kokemuksistani mahdollisimman rehellisesti ja avoimesti, sekä perustelemaan tekemiäni valintoja tieteellisten käytäntöjen mukaisesti. Päiväkirjatyöskentelyn kirjoitusprosessia ja näiden käytäntöjen noudattamista helpotti myös huomattavasti se, että sain työpaikkani puolesta varsin vapaat kädet työtehtävistä kertomiseen, eikä minun tarvinnut jättää asioita kertomatta tietosuoja-asioiden vuoksi.

6.2 Tasohyppelypelien peruselementtejä ja subjektiivinen vaikeustaso

3D-tasohyppelypeleistä puhuttaessa, ja varsinkin niitä kehittäessä ei voi olla huomioimatta erästä tiettyä peliä. Huolimatta siitä, että se ei ole ensimmäinen lajissaan, on peli, josta tulisi vielä tänäkin päivänä 3D tasohyppelypelien esikuva, olisi epäilemättä Super Mario 64, joka oli Nintendon ensimmäinen Nintendo 64 -konsolille julkaistu peli. (Bycer 2019, 67)

Monipuolisuus tavoissa, joilla Mariota oli mahdollista liikuttaa, johti ilmiöön, jota voi kuvata subjektiivisena vaikeutena. Yksi Mario 64: kehityksen tärkeimmistä seikoista oli tasosuunnittelufilosofian muutos. 2D-peleissä tasot olivat lineaarisia ja vaativat pelaajaa suorittamaan tarkkoja hyppyjä ja liikkeitä aina samoilla tavoilla tasojen läpäisemiseksi. Koska Marion ohjaamiseen oli enemmän tilaa, ja monenlaisia tapoja, pelaajien oli mahdollista läpäistä tasot erilaisia tekniikoita käyttämällä. Tällaisesta suunnittelufilosofiasta onkin tullut 3D-tasohyppelypelien

design-ratkaisujen peruselementti. Vähemmän taitaville pelaajille tämä tarkoitti sitä, että mikäli he eivät pystyneet suorittamaan vaadittua tehtävää, he pystyivät kuitenkin aina tekemään jotain muuta edetäkseen pelissä eteenpäin. Marion erilaiset kyvyt eivät myöskään olleet suunniteltu vihollisten kanssa yhteen ottamiseen, vaan niiden tarkoituksena oli ratkaista pelin erilaisia pulmia ja haasteita. (Bycer 2019, 70) Näistä ominaispiirteistä johtuen Mario 64 kuuluu fundamentaalisesti ehdottomasti oman pelimme esikuviin.

3D-tasohyppelypeli on loistava esimerkki siitä, kuinka subjektiivista vaikeutta voidaan soveltaa peleihin. 3D Mario -peleissä ne on suunniteltu mukautumaan useille eri taitotasoille, mutta samalla auttamaan noviisi pelaajia kehittymään. Subjektiivista vaikeutta voidaan soveltaa kahdella osa-alueella - tasosuunnittelussa ja pelitekniikoiden täydessä hallitsemisessa. Pelitasoilla on kiinteitä tavoitteita tai haasteita, mutta polku niiden läpi voi vaihdella sen mukaan, kuinka hyvin pelaaja hallitsee hahmoaan. Tämä mahdollistetaan luomalla tilanne, joka tarjoaa vaihtoehdon löytää oikoteitä (joko tahallisia tai improvisoituja) esteiden läpäisemiseen. Uudet pelaajat suorittavat säännöllistä etenemistä päästäkseen tasojen läpi, vaikkakin hitaammin. Edistyneemmät pelaajat voivat hyödyntää erilaisia tekniikoita, ohittaakseen haasteet kokonaan, tai selvitäkseen niistä paljon nopeammin. Molemmat etenemistavat ovat täysin päteviä, ja tarjoavat pelaajalle mahdollisuuden edetä omaan tahtiin. Pelin edetessä vaikeuskäyrä kasvaa tason läpi kulkevan peruspolun suhteen. Myöhemmät tasot alkavat vaatia edistyneempiä tekniikoita, vaikka niitä voitaisiin käyttää jo pelin aiemmissa vaiheissa. Tässä vaiheessa myös noviisi pelaajat alkavat oppia integroimaan näitä toimintoja normaaliin pelityyliinsä. Pelin loppuun mennessä sekä kokeneilla- että alun perin noviisipelaajilla tulee olla samat taidot, joita tarvitaan viimeisten vaiheiden läpäisyyn. Tämän menetelmän kauneus on siinä, että eteneminen on orgaanista - noviisipelaajat eivät missään vaiheessa kokeneet, että heitä hoputettiin oppimaan pelin mekaniikkoja, kun taas kokeneemmat pelaajat pääsivät välittömästi heidän ymmärrykseensä sopiviin haasteisiin ja tilanteisiin. Avainelementti subjektiiviseen vaikeuteen on se, miten pelaaja pelaa pelin läpi hänen ymmärryksensä kasvaessa. (Bycer 2019, 78–79) Pyrimme noudattamaan oman pelimme kehityksessä täysin samoja elementtejä ja avaintekijöitä. Subjektiivinen vaikeus on suuressa osassa

pelimme kenttäsuunnittelua ja pelihahmojemme kykyjä, koska haluamme tarjota sopivasti haastetta eri tasoisille pelaajille. Tästä johtuen hahmoille suunnitellut kyvyt ja mekaniikat tarvitsee saada toteutettua mahdollisimman tarkasti visioiden mukaan, jotta ne yhdessä kenttäsuunnittelun kanssa täydentävät toisiaan, ja luovat miellyttävän pelikokemuksen.

6.3 Peliohjelmoijan vastuu

Peliohjelmoijan vastuu tämän kaltaisessa projektissa on siis varsin suuri. Kuten olen maininnut, työskentelen pienessä pelistudiossa, ja olen tällä hetkellä yrityksen ainoa ohjelmoija. Luonnollisesti vastuullani on siis paljon eri osa-alueiden asioita. Siinä missä pelin design-puolen valinnat tulevat pääasiassa muilta tiimimme jäseniltä, minun tehtäväni on toteuttaa nämä visiot, ja saada ne täsmäämään mahdollisimman hyvin. Tämä näkyy erityisesti pelissä esiintyvien hahmojen toteutuksissa.

On tärkeää, että jokainen erilainen suoritettava hyppy (pelistämme löytyy useita erilaisia hyppymuotoja tavallisen hypyn lisäksi, kuten tuplahyppy, seinähyppy, köysihyppy, delfiinihyppy ja niin edelleen) on juuri oikean pituinen, ja sen tulee tuntua hyvältä pelaajalle. Jokainen suoritettu lyönti tulee kohdistua viholliseen oikein, ja osumasta tapahtuvan efektin tulee tapahtua virheettömästi. Jokaisen pelissä esiintyvän hahmon animaation tulee pyöriä oikeaan aikaan, niiden tulee olla sulavia niiden eri liikkeiden välillä, eikä niihin saa tulla katkoksia. Jokaisen pelissä esiintyvän hahmon tekoälyä suorittavan logiikan tulee toimia virheettömästi, eikä hahmojen käytöksessä saa esiintyä viivettä tai kummallisuuksia. Jokaisen pelin HUD-elementin tulee näyttää reaaliaikaista tietoa pelin tapahtumista, ja niiden esiin tuominen ja piilottaminen tulee tapahtua oikealla hetkellä. Minkä tahansa mainitsemani osa-alueen epäonnistuessa, immersio pelin ja pelaajan välillä katkeaa, joka voi hyvin nopeasti saada pelaajassa aikaan turhautumisen tunnetta, joka voi johtaa pelin lopettamiseen. En missään nimessä väitä, että pelin tulee olla sataprosenttisesti virheetön, vailla ainuttakaan bugia, koska rehellisesti sanottuna se ei ole mahdollista. Pelin yleinen presentaatio ja laatu kuitenkin määrittelee, onko peliä miellyttävää

pelata, haluaako pelaaja viettää sen parissa aikaa, ja saako peli tarpeeksi suosiota. Yhden peliohjelmoijan taktiikalla ainoa vaihtoehto ohjelmoijalle on pystyä toimimaan niin sanottuna generalisti ohjelmoijana.
Generalisti ohjelmoija on joustava kehitystiimin jäsen, joka voi työskennellä usealla erikois- osa-alueella. Pelien muuttuessa aina vain monimutkaisemmiksi, on olemassa yhä enemmän ohjelmoinnin erikois- osa-alueita, kuten fysiikka, tekoäly, pelimoottori ja verkko-ohjelmointi. Generalisti ohjelmoija ylläpitää, ja pystyy heijastamaan osaamistaan useammalle kuin yhdelle erikois- osa-alueelle pelin kehitysprosessissa. Vastuut vaihtelevat pelistudioittain, joten laaja ohjelmointiosaaminen ja kyky multitasking (kyky tehdä asioita samanaikaisesti - toimimiseen on tärkeää. (Into Games 2023)

Ajatus ainoana ohjelmoijana olosta on hyvin verrattavissa tämän opinnäytetyön kirjoittamiseen. Omalla tavallaan on hienoa, että pääsen jatkuvasti tekemään erilaisia asioita, oppia uutta, ja kasvattaa osaamistaan. Se motivoi minua suoriutumaan työtehtävistäni mahdollisimman hyvin ja tehokkaasti. Pidän siitä, kun pääsen näkemään oman työni jäljen, ja esittelemään tekemiäni toteutuksia muille. Tietynlainen paine siitä, että minun tarvitsee keksiä ratkaisut kaikkiin ohjelmointihaasteisiin, tuo työtehtäviin motivaatiota, ja tietynlaista ammattiylpeyttä. Pidän työstäni valtavasti, ja menen joka päivä töihin positiivisin mielin. Kääntöpuolena ainoana ohjelmoijana olossa on luonnollisesti onnistumisen paineet. Mikäli työtehtävään kuluu paljon odotettua enemmän aikaa, koen epäonnistuneeni, ja kuluttaneeni siihen liikaa aikaa. Vastaavasti jos jokin toteutus ei vastaa tavoitetta, saattaa se aiheuttaa riittämättömyyden tunnetta. Työtehtävien priorisointi on tällaisessa tilanteessa tärkeää. Useaa työtehtävää on hankala tehdä samaan aikaan, joten luonnollisesti yhtä asiaa tehdessä, muut odottavat. Avoin keskustelu ja koordinointi tiimin jäsenten kanssa on avainasemassa, jotta tärkeimmät työtehtävät saadaan työn alle nopeammin.

Hyvin tiivistetysti ohjelmoijan vastuulla on laittaa asiat liikkumaan ja toimimaan siten, kuten niiden on suunniteltu liikkuvan ja toimivan. Työtehtäviä toteuttaessa yritän muistaa selkeän rakenteen, ja purkaa työtehtäviin tarvittavia asioita pienempiin osiin. Yritän miettiä mitä tarvitaan, missä tarvitaan, ja mitkä asiat

ovat niistä riippuvaisia. Pyrin käyttämään paljon mielikuvitusta, ja luovuutta ongelmien ratkaisuun. Joshua Bycer (2019) kertoo kirjassaan haastatelleensa nimettömäksi jäänyttä pelikehittäjää, joka oli sanonut hänelle “Pelisuunnittelu on yhdistelmä viihdettä ja ohjelmointia". Molemmat osa-alueet tarvitsevat ymmärtää videopelin luomiseksi. (Bycer 2019, 131)

6.4 Loppusanat

Kaiken kaikkiaan opinnäytetyön toteuttaminen päiväkirjamuotoisena oli hyvin kasvattava ja opettavainen kokemus. Se sai minut haastamaan itseäni, kehittämään ammatillista osaamistani, ja tarkastelemaan työskentelyäni kriittisemmästä näkökulmasta. Koen tämän prosessin kehittäneen ammatillista osaamistani, ja oppineeni sen aikana uusia toimintatapoja, joita voin hyödyntää työskentelyssäni jatkossa. Opinnäytetyön kirjoittaminen täysipäiväisten työpäivien ja opintojen ohella oli henkisesti ja fyysisesti varsin kuormittava prosessi, jota en välttämättä olisi jaksanut päällekkäin pitää yllä koko kolmeatoista viikkoa. Onneksi päällekkäisyys kesti kuitenkin vain alle puolet tarkastelujakson pituudesta. Koen päiväkirjamuotoisen opinnäytetyön olleen silti oikea valinta itselleni, koska pääsin kertomaan itselleni tärkeästä aiheesta, eli työstäni peliohjelmoijana 3D-tasohyppelypelin kehitysprojektissa.

Lähteet

Bularca, L. 2016. How to show your indie game at any expo. Game Developer. https://www.gamedeveloper.com/business/how-to-show-your-indie-game-at-any-expo. 17.9.2023.

Bycer, J. 2019. Game Design Deep Dive: Platformers. United Kingdom: Taylor & Francis Group. ProQuest Ebook Central. 18.6.2023.

Bycer, J. 2019. Game Design Deep Dive: Platformers. United Kingdom: Taylor & Francis Group. ProQuest Ebook Central. 18.10.2023.

Bycer, J. 2019. Game Design Deep Dive: Platformers. United Kingdom: Taylor & Francis Group. ProQuest Ebook Central. 19.10.2023.

Carrington, T. 2023. Tips and tricks for optimizing performance in Unreal Engine. Craving Tech. https://www.cravingtech.com/tips-and-tricks-for-optimizing-performance-in-unreal-engine.html. 8.7.2023.

Epic Games. 2023. Adding Components to an Actor. https://docs.unrealengine.com/4.27/en-US/InteractiveExperiences/HowTo/AddingComponents/. 22.6.2023.

Epic Games. 2023. Updating Projects to Newer Versions of Unreal Engine. https://docs.unrealengine.com/5.1/en-US/updating-projects-to-newer-versions-of-unreal-engine/. 30.6.2023.

Epic Games. 2023. Setting Up a Character. https://docs.unrealengine.com/4.27/en-US/AnimatingObjects/SkeletalMeshAnimation/CharacterSetupOverview/. 8.7.2023.

Epic Games. 2023. Macros. https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints/UserGuide/Macros/. 13.7.2023.

Epic Games. 2023. Animation Retargeting. https://docs.unrealengine.com/4.27/en-US/AnimatingObjects/SkeletalMeshAnimation/AnimationRetargeting/. 13.7.2023.

Epic Games. 2023. Balancing Blueprint and C++. https://docs.unrealengine.com/4.27/en-US/Resources/SampleGames/ARPG/BalancingBlueprintAndCPP/. 13.8.2023.

Epic Games. 2023. Root Motion. https://docs.unrealengine.com/4.27/en-US/AnimatingObjects/SkeletalMeshAnimation/RootMotion/. 18.8.2023.

Epic Games. 2023. Overview of State Machines. https://docs.unrealengine.com/4.27/en-US/AnimatingObjects/SkeletalMeshAnimation/StateMachines/Overview/. 18.8.2023.

Epic Games. 2023. Control Rig. https://docs.unrealengine.com/5.0/en-US/control-rig-in-unreal-engine/. 25.8.2023.

Epic Games. 2023. CCDIK. https://docs.unrealengine.com/4.27/en-US/AnimatingObjects/SkeletalMeshAnimation/NodeReference/SkeletalControls/CCDIK/. 25.8.2023.

Epic Games. 2023. Transform Bone. https://docs.unrealengine.com/5.0/en-US/animation-blueprint-transform-bone-in-unreal-engine/. 3.9.2023.

Epic Games. 2023. Using Timers. https://docs.unrealengine.com/4.26/en-US/InteractiveExperiences/UseTimers/. 8.9.2023.

Epic Games. 2023. FBX Content Pipeline. https://docs.unrealengine.com/4.27/en-US/WorkingWithContent/Importing/FBX/. 8.9.2023.

Epic Games. 2023. Source Control. https://docs.unrealengine.com/5.3/en-US/source-control-in-unreal-engine/. 4.10.2023.

Epic Games. 2023. Filtering the Content Browser. https://docs.unrealengine.com/4.26/en-US/Basics/ContentBrowser/UserGuide/Filters/. 4.10.2023.

Epic Games. 2023. Sequencer Overview. https://docs.unrealengine.com/4.27/en-US/AnimatingObjects/Sequencer/Overview/. 4.10.2023.

Epic Games Dev Community. 2022. The mannequin’s torso stretches. https://forums.unrealengine.com/t/the-mannequins-torso-stretches/546647. 13.7.2023.

Faucher, W. 2021. Fix the Broken First Frame in Sequencer / Movie Render Queue! - Unreal Engine. YouTube-video. https://www.youtube.com/watch?v=lXcerW59onA. 8.9.2023.

Firelight Technologies. 2023. Unreal Integration 2.02.19. https://www.fmod.com/docs/2.02/unreal/user-guide.html#compiling-the-plugin-optional. 16.6.2023.

Firelight Technologies. 2023. Unreal Integration 2.02.19. https://www.fmod.com/docs/2.02/unreal/user-guide.html#compiling-the-plugin-optional. 22.6.2023.

Into Games. 2023. Careers Generalist Programmer. https://intogames.org/role/generalist-programmer/. 19.10.2023.

Karelia-ammattikorkeakoulu. 2021. Päiväkirjamuotoinen opinnäyte Karelia-ammattikorkeakoulussa. https://libguides.karelia.fi/ld.php?content_id=34719751. 18.10.2023.

Tutkimuseettinen neuvottelukunta. 2021. Hyvä tieteellinen käytäntö (HTK). https://tenk.fi/fi/tiedevilppi/hyva-tieteellinen-kaytanto-htk. 18.10.2023.

Unity Technologies. 2023. Introduction to Object Pooling - 2019.3. https://learn.unity.com/tutorial/introduction-to-object-pooling-2019-3#. 9.8.2023.

Z-Axis Up. 2022. How to Properly Upgrade an Unreal Engine Project. YouTube-video. https://www.youtube.com/watch?v=DPOHRQprLIo. 30.6.2023