



Tino Kankkunen

Kauhupelin parametrisoidun teko- älyn toteuttaminen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikan tutkinto-ohjelma

Insinöörityö

19.11.2024

Tiivistelmä

Tekijä: Tino Kankkunen
Otsikko: Kauhupelin parametrisoidun tekoälyn toteuttaminen
Sivumäärä: 59 sivua + 3 liitettä
Aika: 19.11.2024

Tutkinto: Insinööri (AMK)
Tutkinto-ohjelma: Tieto- ja viestintätekniikan tutkinto-ohjelma
Ammatillinen pääaine: Pelisovellukset
Ohjaajat: Lehtori Antti Laiho

Insinööriyössä tutkittiin kauhun luomisen keinoja pelimaailmassa, kauhupeleissä käytettyjä erilaisia tekoälyratkaisuja ja niiden toimintaa pelin aikana lisäksi selvitettiin, kuinka kauhupeliipiireissä arvostetun Creative Assemblyn vuonna 2014 julkaiseman Alien: Isolation -pelin tekoäly toimii ja toteutettiin vastaavanlaisen tekoälyn rakenne pienemmällä skaalalla.

Insinööriyössä syvennyttiin tarkemmin siihen, kuinka historiallisesti hyvin toteutetuissa kauhupeleissä Alien: Isolation ja Valven Left 4 Dead tekoälykokonaisuudet on toteutettu ja miten ne ohjaavat pelaajan pelikokemusta. Samalla analysoitiin erityisesti Alien: Isolation -pelin päävihollisen laajaa käyttäytymispuuta avoimen lähdekoodin työkalulla OpenCAGE (liite 1). OpenCAGE:n avulla päästiin käsiksi tarkemmin pelin tekoälyn käyttäytymispuun solmuihin, tekoälyn toimintaa tukeviin sensoreihin ja siihen, mikä tekee kyseisestä tekoälystä niin pelottavan.

Työssä luodun tekoälytoteutuksen kehitysympäristöksi valittiin Unity-pelimoottori, sillä se oli jo ennestään hyvin tuttu kehittäjälle. Tekoälyanalyysien pohjalta rakennettiin 3D-mallinnusohjelmalla kompakti simulaatioympäristö Unityyn, jonne tekoälytoteutus tehtiin. Tekoälyn toiminnallisuus rakennettiin ensin tilakoneella, joka vaihdettiin myöhemmin käyttäytymispuulla toimivaan rakenteeseen, tilakoneen osoittautuessa monimutkaisesti skaalattavaksi haluttujen toimintojen määrän kasvaessa.

Lopputuloksena syntyi helposti skaalautuvalla rakenteella luotu käyttäytymispuulla varustettu tekoäly. Simulaatioympäristöä vihollisen ja pelaajan toimintaa parametrisoitiin ohjaajatason tekoälyn tiedonkeruuta varten. Kehitetty tekoäly on tulevaisuudessa tarkoitus julkaista julkiseen testaukseen ja Unityn verkkokauppaan pelikehittäjien saataville. Linkki luodun tekoälyn Unity-projektiin löytyy liitteestä 3.

Avainsanat: Kauhu, kauhupelit, tekoäly, Unity, tilakone, käyttäytymispuu

Abstract

Author: Tino Kankkunen
Title: Creating a parameterized artificial intelligence in a horror game
Number of Pages: 59 pages + 3 appendices
Date: 19 November 2024

Degree: Bachelor of Engineering
Degree Programme: Information and communication technology, ICT
Professional Major: Game development
Supervisors: Antti Laiho, Senior lecturer

The engineering thesis explored techniques for creating horror in game worlds, various artificial intelligence (AI) solutions used in horror games and how they operate during gameplay. It examined the AI of the highly regarded horror game *Alien: Isolation*, developed by Creative Assembly in 2014, and implemented a similar AI structure on a smaller scale.

The thesis delved into the AI systems of *Alien: Isolation* and Valve's *Left 4 Dead*, both historically well-executed horror games, analyzing how these AI systems guide the player's experience. Special focus was given to the extensive behavior tree of the main antagonist in *Alien: Isolation*, using the open-source tool OpenCAGE (Appendix 1). OpenCAGE enabled a detailed examination of the nodes in the behavior tree, the sensors supporting the AI's functionality, and the factors contributing to the AI's terrifying effect.

Unity, a game engine familiar to the developer, was chosen as the development environment for the AI prototype created in this thesis. Based on the AI analyses, a compact simulation environment was modeled into Unity where the AI implementation was carried out. Initially, the AI functionality was built using a state machine, which was later switched to a behavior tree structure when the state machine proved challenging to scale as the number of desired functions increased.

The result was an AI with a behavior tree structure designed for easy scalability. Parameters were added to the simulation environment to control the actions of the enemy and the player, enabling data collection for director-level AI. The developed AI is planned for public testing and will be made available on the Unity Asset Store for game developers. A link to the Unity project for the created AI can be found in Appendix 3.

Keywords: Horror, horror games, artificial intelligence, Unity, state machine, behavior tree

Sisällys

Lyhenteet	5
1 Johdanto	1
2 Kauhu ja kauhun luominen peleissä	3
2.1 Kauhun määritelmä	3
2.2 Kauhua herättävät visuaaliset keinot	4
2.3 Jännitystä korostava äänimaailma	12
2.4 Pelisuunnittelun rooli kauhukokemuksen rakentamisessa	13
3 Tekoäly ja sen toteutukset kauhu -ympäristössä	16
3.1 Tekoäly kauhupeleissä	16
3.2 Tekoällyn parametrisointi kauhu -ympäristössä	25
3.3 Alien: Isolation, esimerkki toimivasta kauhupelin tekoälystä	31
4 Simulaatioympäristön kehitys ja tekoälytoteutus	42
4.1 Simulaatioympäristön suunnittelu ja luominen	42
4.2 Luotu tekoälykokonaisuus ja sen käyttäytymispuu	45
5 Tekoälytoteutuksen arviointi	51
5.1 Lopullisen tekoällyn arviointi	51
5.2 Luodun tekoällyn tulevaisuus ja jatkokehitysmahdollisuudet	52
6 Yhteenveto	55
Lähteet	57
Liite 1: OpenCAGE – Alien: Isolation	1
Liite 2: MoodMe SDK	1
Liite 3: Luodun tekoällyn Unity -projekti	1

Lyhenteet

AI	Artificial Intelligence, tekoäly.
NPC	Non-Playable Character, ei-pelattava hahmo.
DDA	Dynamic Difficulty Adjustment, dynaaminen vaikeusasteen muutos.
FSM	Finite State Machine, äärellinen tilakone.
FuSM	Fuzzy State Machine, sumea tilakone.
AAS	Active Area Set, aktiivinen asetettu alue on jokaisen pelaajan ympärillä tietyn ennalta määrätyn säteen sisällä olevat aktiiviset lähialueet Left 4 Dead -pelissä.

1 Johdanto

Maailmanlaajuisen striimaukkulttuurin sekä peliteknologian kehityksen myötä julkaistujen kauhupelien suosio ja määrä on kasvanut viime vuosien saatossa nopeasti. Kauhupeligenrenä on jakautunut suurimmalta osin toimintakauhuun, joka on enemmänkin vain toimintaa kauhuympäristössä, ja hyppypelästyksien tähdittämään indie-kauhuun. Yksittäisen indie-pelikehittäjän on nyt helpompaa kuin koskaan luoda oma virtuaalinen kauhukokemus ja julkaista se ihmisten pelattavaksi Steamiin tai muuhun pelipalveluun. Kauhupelien taso on kuitenkin laadullisesti laskenut viime vuosikymmenen aikana. Suurimmat syyt tähän ovat ennalta arvattavien tai surkeiden tekoälyvastustajien käyttö sekä kömpelösti hyödynnetyt kauhun luomisen keinot, kuten huonosti ajoitetut hyppypelästyksiset tai pieleen mennyt valaistus.

Vuonna 2014 julkaistua scifi-kauhupeliä nimeltä Alien: Isolation (Creative Assembly) pidetään edelleen kauhupelimaailman yhtenä parhaista ja pelottavimmista peleistä siihen kehitetyn tekoälyn takia. Kyseisen pelin vihollisen tekoäly toimii hyvin ainutlaatuisella tavalla, kun tehdään päätöksiä ja muutoksia omaan käyttäytymiseen pelin edetessä riippuen pelaajan pelityylistä. Näin luodaan todella jännittävä ja ennalta arvaamaton pelikokemus. Samalla parannetaan pelin uudelleenpelattavuutta.

Opinnäytetyön tarkoituksena on tutkia kauhun luomista peleissä, kauhupeleissä käytettyjä tekoälyratkaisuja ja selvittää, kuinka Alien: Isolation -pelin tekoäly toimii. Lopuksi tavoitteena on toteuttaa vastaavanlainen tekoäly pienemmällä skaalalla. Työssä on tarkoituksena myös luoda tekoälylle referenssipelin mukainen testiympäristö ja parametrisoida pelaajan liikkuminen ja toiminta, minkä kautta vaikutetaan vihollisen tekoälyn päätöksiin. Tavoitteena on tekoälykokonaisuus, jonka päälle on helppo rakentaa lisää toiminnallisuutta.

Monimutkaisen ja arvaamattoman, mutta helposti implementoitavan kauhupelin tekoälyvastustajan luonti ja sen julkaisu kaikkien käytettäväksi saattaisi

parantaa kauhupelien yleistä tasoa erityisesti indie-kehittäjien kohdalla ja toisi mahdollisesti lisää suosiota selviytymiskauhugenreen.

Insinööriyön alussa pohjustetaan aihepiiriä tutkimalla kauhun luomista peleissä ja sen eri kategorioita. Yleisen pohjustuksen jälkeen syvennyttään kauhupelien tekoälyihin ja niiden toimintatapoihin. Tutkielma keskittyy erityisesti Alien: Isolation -pelin vihollisen tekoälyyn, jonka pohjalta opinnäytetyön simulaatioympäristö ja siihen kuuluva tekoäly on toteutettu. Referenssitekoälyn käsittelyn jälkeen esitellään opinnäytetyössä luodun vihollistekoälyn toimintatapa, pelaajan toiminnan parametrisointi sekä simulaatioympäristön luominen. Tutkielman lopussa arvioidaan toteutettua kokonaisuutta ja kartoitetaan luodun tekoälyn tulevaisuuden suunnitelmia.

2 Kauhun ja kauhun luominen peleissä

2.1 Kauhun määritelmä

Kauhu ilmenee pelon tunteena, ja siihen liittyy tuntemuksia jonkin asian tai tilanteen kauhistuttavuudesta tai karmivuudesta. Kauhu saattaa aiheuttaa fyysisiä oireita, kuten vapinaa, hikoilua, huimausta, sydämentykytystä ja hyperventilaatiota. Oireet ovat samat kuin voimakkaan ahdistumisen aiheuttamat fyysiset oireet. (Neurotic, stress-related and somatoform disorders (F40-F48), 2019.)

Kauhu rakentuu pelon tunteen ympärille. Pelko on ihmisen primitiivinen reaktio minkä tahansa vaaran edessä. Tämä pohjautuu ihmisen evoluutioon ja liittyy yleensä todellisen ulkoisen vaaran havaitsemiseen. Pelon oireisiin kuuluvat sisäinen jännitys, levottomuus ja kauhun tai paniikin tunne. Pelkoa voi seurata vaistomainen pelkokäyttäytyminen, joka ilmenee taistelemisena, pakenemisena tai paikalleen jähmettymisenä. (Fight Or Flight Response.)

Pelko voi kohdistua asioihin, tilanteisiin tai tunteisiin (Huttunen 2018). Tämä seikka on tärkeä ottaa huomioon pelien kontekstissa, jotta voidaan ymmärtää, mihin eri pelin elementteihin pelon ja kauhun tunteet voidaan liittää.

Kauhupelien päämääränä on systemaattisesti herättää pelaajassa pelon tunnetta monipuolisesti hyödyntäen visuaalisia, kerronnallisia, auditiivisia ja kokemuksellisia elementtejä. Lisäksi pelaajan kyky eläytyä tilanteeseen sekä pelihahmoon samastuminen ovat merkittäviä tekijöitä pelaajan immersion eli tilanteeseen eläytymisen kannalta. (Soutter 2016.) Mitä syvemmin pelaaja uppoutuu peliin, sitä todentuntuisemmalta ja voimakkaammalta pelin kauhu tuntuu.

Kauhun luominen peleissä edellyttää hienovaraista tasapainoilua pelaajan ja pelin välillä. Pelaajien yksilöllisyys, joiden mukana tuoma subjektiivisuus vaikuttaa suoraan siihen, miten kauhua koetaan.

Seuraavissa alaluvuissa syvennytään tarkemmin eri kauhun luomisen elementteihin.

2.2 Kauhua herättävät visuaaliset keinot

Kun tarkastellaan kauhupelien visuaalisia elementtejä, voidaan olettaa, että jokaisella on jokin käsitys siitä, millainen ympäristö koetaan pelottavana. Tähän mielikuvaan vaikuttavat oletettavasti ihmisen aikaisemmat kokemukset, kulttuurinen ja sosiaalinen kasvatus sekä uskomukset.

Visuaalisen ympäristön luominen koostuu useista eri komponenteista, ja niitä suunnitellessa on tärkeää pitää mielessä, miten jokainen liittyy pelin teemaan ja miten komponentteihin on liitetty pelon tunnetta. Lisäksi ympäristöjä suunniteltaessa on yleistä ottaa huomioon pelaajan mahdolliset fobiat, kuten esimerkiksi ahtaanpaikankammo, korkeanpaikankammo, pimeän pelko, tuntemattoman pelko ja yksinjäämisen pelko. Pelko voi myös kohdistua johonkin fyysiseen asiaan tai olentoon, kuten esimerkiksi araknofobia eli pelko hämähäkkejä kohtaan. Yleisimpiä fobioita käytetään hyväksi, jotta kauhu tavoittaisi mahdollisimman laajan pelaajakunnan.

Visuaalisten komponenttien joukossa ovat muun muassa valaistus, tilavuusobjektit, kuten sumu tai ilmankosteus, teema ja sitä kautta miljö, 3D-mallit, kuvakulma, näkökenttä, kameran efektit, jälkikäsitely ja animaatiot. Kenttäsuunnittelua käsitellään pelisuunnittelun osassa.

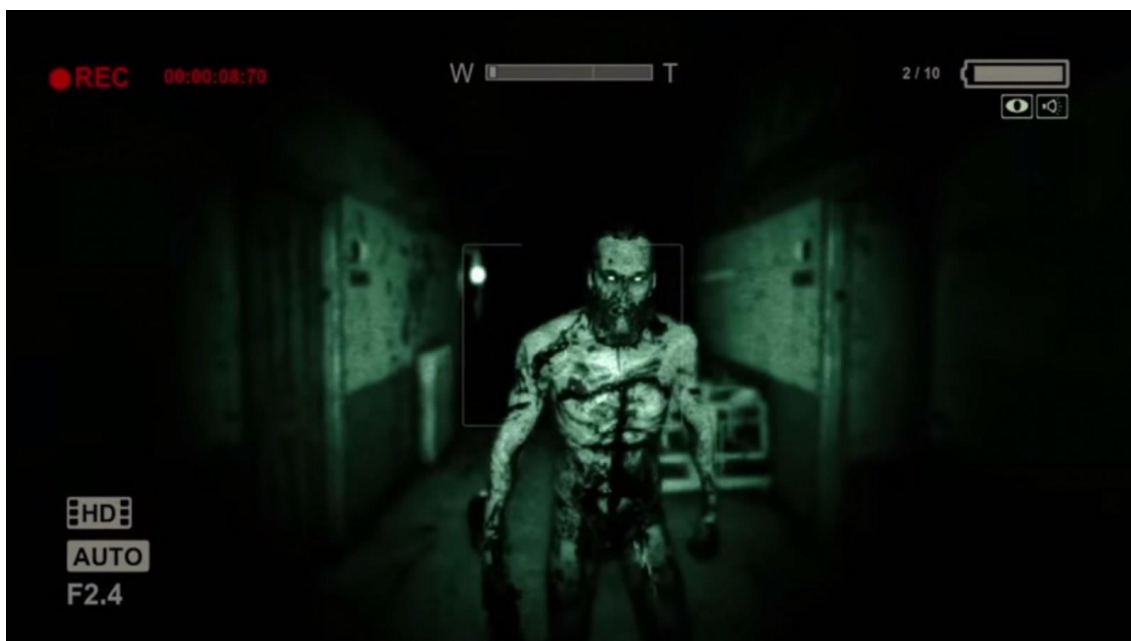
Seuraavaksi käsitellään visuaalisen kauhun komponentit yleisellä tasolla muuttaman havainnollistavan peliesimerkin kanssa.

Kauhunomainen valaistus ja sen hyödyntäminen pelimekaniikkana

Tutkimusten mukaan valaistus on yksi pelottavimmista visuaalisen ympäristön komponenteista ja erityisesti vilkkuvat valot (Graja 2020). Valaistuksen mitattiin tuottavan pelaajalle eniten stressiä ja ahdistusta audiovisuaalisessa ympäristössä. Vaikka valo tuo turvallisuuden tunnetta pimeydessä, se luo samalla myös varjoja.

Valaistusta käytetään peleissä hyvin monella eri tapaa, kuten esimerkiksi ohjaamaan pelaajaa, luomalla tunnelmaa, pelotteena tai pelimekaniikkana.

Erittäin havainnollistavana esimerkkinä valaistuksen käytöstä mainittakoon Red-Barrels kehittäjän peli "Outlast" (2013). Pelin valaistus ohjaa pelaajaa eteenpäin kiinnittämällä pelaajan huomion tärkeisiin asioihin tai paikkoihin. Valaistus luo pelissä myös paljon pimeitä alueita, joissa pelin päämekaniikkana toimivaa videokameran pimeänäkötoimintoa käyttämällä pelaaja pystyy navigoimaan näissä alueissa. Pimeänäkötoiminnolla pelaajan näkymä on vihermustavalkoinen, eteenpäin näkyy hyvin rajallinen matka, kuten kuvasta 1 voidaan nähdä. Kuluvien paristojen ansiosta pimeänäkötoiminnon rajallinen käyttö vähentää turvallisuuden tunnetta ja vaistomaisesti pakottaa pelaajan etenemään pimeiden alueiden läpi, jotta pelaaja ei jää pimeään ilman minkäänlaista tapaa nähdä eteensä. Näin pelissä luodaan valaistuksella paljon kauhua ja jännitystä käyttäen hyväksi myös eri fobioita, kuten pimeän ja tuntemattoman pelkoa.



Kuva 1. Outlast-pelin pimeänäkötoiminto (Outlast 2013).

Toinen esimerkki kekseliästä valaistuksen käytöstä kauhupelissä on Remedy Entertainmentin Alan Wake (2010). Alan Wake on toimintakauhupeli, jossa

hyödynnetään valaistusta samalla tavoin, kuin Outlast -pelissä ohjataan pelaajaa eteenpäin pelikentissä ja korostetaan mielenkiinnon kohteita ympäristössä. Alan Wake -pelissä valaistusta käytetään myös pelin päämekaniikkojen yhteydessä. Pelissä taistellaan pimeyden ympäröimiä ihmishaamuja vastaan, jotka eivät ota vahinkoa, ennen kuin pimeys niiden ympäriltä on karkotettu. Alan Wakella onkin aseistuksenaan taskulamppu, jonka avulla pimeys vihollisten ympäriltä voidaan tuhota osoittamalla valoa haamuja päin. Suojaavan pimeyden hävennyttyä viholliset ottavat vahinkoa normaalisti muista aseista, kuten revolverista. Taskulampun paristo latautuu itsestään ajan kuluessa, mutta kuluu huomattavan nopeasti sitä käytettäessä, mikä luo pelaajalle stressiä ja toivottomuuden tunnetta useampaa vihollisia vastaan taistellessa. Pelin taistelumeکانii-koista syntyy mielekäs ja sykettä nostava minipeli taskulampun ja aseiden välillä. Mainittakoon myös, että Alanin ollessa vahingoittunut, hän voi parantaa elämänpisteensä seisomalla esimerkiksi katulampun valokeilassa.

Tilavuusobjektien käyttö tunnelmaa luodessa

Tilavuusobjektit ovat olleet osana pelejä lähes alusta lähtien tavalla tai toisella. Niitä käyttäen pelimaailmaan saadaan lisää eloa ja realismia, mikä parantaa pelaajan immersiota. Yleisimpiä tilavuusobjekteja ovat sumu tai muut vastaavat savuelementit sekä ilmankosteus. Immersion lisäksi tilavuusobjekteja käytetään kauhupeleissä rajaamaan pelaajan näkökenttää. Näin luodaan laajoistakin alueista ahtaan tuntuisia. Klassisena esimerkkinä tilavuusobjektien käytöstä pidetään japanilaisen pelistudio Konamin julkaisemaa Silent Hill -pelisarjaa (1999), joka on tunnettu ikonisesta sumustaan. Alun perin sumua käytettiin pelisarjassa optimoinnin yhteydessä, jotta objekteja ei tarvitse piirtää pelaajalle näkyviin sumun takaa. Myöhemmin kuitenkin sumu jäi pelisarjan peleihin kauhun elementtinä. Sumun näkökenttää rajoittava ominaisuus voimistaa pelkoa tuntemattomasta varsinkin, jos sumun sisään sijoitetaan erilaisia ääniä tai olentoja muistuttavia hahmoja (kuva 2).



Kuva 2. Silent Hill -pelin sumu (Silent Hill, 1999).

Silmänliikkeitä seuraavassa tutkimuksessa huomattiin, että pelaaja käytti peliä pelatessaan 31,5 % ajasta sumun reunoja seuraten (Niedenthal 2009: 178). Tästä voidaan päätellä, että näkökentän rajaaminen on hyvin tehokas keino tuottaa lisää ahdistusta ja pelkoa tuntemattomasta varsinkin yhdistettynä kolmiulotteiseen äänimaailmaan.

Kauhuympäristöjen teema ja miljö

Kauhupelien teemat voivat vaihdella laajasti, ja ne hyödyntävät erilaisia kauhugenren alalajeja. Yleisimpiä teemoja ovat yliluonnolliset ilmiöt, kuten kummitukset, demonit tai muut yliluonnolliset voimat, zombiapokalypsit tai psykologinen kauhu. Psykologiset kauhupelit keskittyvät vahvasti pelaajan mielentilan järkyttämiseen ja käyttävät aktiivisesti hyödyksi fobioita ja traumoja.

Lähes kaikissa kauhupelien teemoissa korostuu myös selviytymisen elementti, jossa pelaajalta vaaditaan älykkyyttä ja resurssien hallintaa, jotta hän selviytyy pelin asettamista uhista tai vihamielisestä ympäristöstä. Tämä lisää pelaajan haavoittuvuuden tunnetta ja luo jännitystä, kun hän yrittää navigoida vaarallisten tilanteiden läpi.

Miljööön osalta kauhupelit sijoittuvat usein synkkiin, painostaviin, mutta tutunoloi-
siin ympäristöihin. Autoituneet sairaalat, rappeutuneet kartanot, hylätyt teolli-
suusalueet tai muusta maailmasta eritäytyneet kylät tarjoavat ideaalisen näyttä-
mön kauhulle (kuva 3). Pimeys ja epävarmuus liittyvät olennaisesti näihin ympä-
ristöihin, ja tämän takia pelaajat joutuvat usein kamppailemaan näkökykynsä
kanssa, mikä lisää kauhun tunnetta.



Kuva 3. Resident Evil Village -pelin ympäristöä (Resident Evil Village 2021).

Karmivat 3D-mallit

Kauhupelien 3D-mallit seuraavat miljööön tai teeman asettamia rajoitteita. Mal-
leille ominaista on progressiivinen vääristyminen. Jotta pelaaja ei kokisi ympä-
ristöä liian tutuksi tai turvalliseksi, on ympäristön muututtava pelin kuluessa jol-
lain tapaa. Tämä tuodaan usein esille 3D-mallien progressiivisena ränsistymi-
senä, lahoamisena tai muuttumisena pelin edetessä.



Kuva 4. Esimerkki jääkaapin progressiivisesta vääristymisestä (Freezer Fridge 3D Model).

Kuvassa 4 nähdään vasemmalta oikealle jääkaapin 3D-mallin progressiivinen lahoaminen ja ränsistyminen. Vasemmalla kuvassa on normaali avonainen jääkaappi, joka pelin edetessä muuttuu vaiheittain oikealla olevaan epämääräisen saastan ja roskan peitossa olevaan rikkinäiseen jääkaappiin.

Pelkoa herättävä kameratyöskentely ja jälkikäsitteily

Kameraa on käytetty hyväksi kauhun tunteen voimistamiseen peleissä jo kymmeniä vuosia. Kauhupeleissä käytetään toisinaan eriskummallisia tai rajoitettuja kameran kuvakulmia luomaan pelaajalle haavoittuvaisuuden ja epämukavuuden tunnetta. Matalat kuvakulmat, vääristynyt perspektiivi ja lähikuvat voimistavat tunnetta siitä, että jokin seuraa tai tuijottaa pelaajaa. Ikoninen esimerkki kameran käytöstä on vanha Resident Evil -peli, jossa kamera oli lukittuna tiettyyn kuvakulmaan jokaisessa ruudussa. Näin luotiin tarkoituksenmukaisesti pelkoa

tuntemattomasta rajoittamalla pelaajan saamaa informaatiota ympäristöstä (Atkinson-Bonasio) ja samalla vaikeuttamalla pelaajan toimintaa.

Jälkikäsitteilyyn kuuluu tässä tapauksessa kuvan värimäärittely ja syväterävyys. Kumpikin efekti tuo pelin visuaaliselle ilmeelle loppusilauksen. Värimäärittelyssä yleisesti kauhukontekstissa kuva desaturoidaan ja kasvatetaan kontrastia. Desaturoinnilla kuvasta viedään liika pirteys ja värikyvyys pois, mikä luo karun ja elottoman ilmapiirin. Tämä lisää rappeutumisen ja toivottomuuden tunnetta ja on samalla linjassa temaattisten elementtien kanssa. Kontrastin kasvatuksella saadaan aikaan selkeämpi ero valaistuksessa valon ja pimeän välille sekä parannetaan samalla jännittävien yksityiskohtien näkyvyyttä. Syväterävyyden rooli on ohjata pelaajan katsetta eri mielenkiinnonkohteisiin sumentamalla pelaajan katseen fokuksipisteen ympäriltä kuvaa, kuten kuvasta 5 voidaan nähdä. Syväterävyys tuo peliin realismia ja lisää jännitystä fokuksipisteen ulkopuolisten objektien sumentuessa epäselviksi muodoiksi.



Kuva 5. Alien: Isolation -pelissä syväterävyydellä ohjataan pelaajan katse kädessä pidettävään liikkeentunnistimeen (Alien: Isolation 2014).

Kauhupeleissä nähdään usein myös vinjetti-efekti (eng. vignette), jolla hämääretään kuvan ja näkökentän reunoja. Syväterävyys ja vinjetti auttavat yhdessä

luomaan eristäytymisen ja haavoittuvuuden tunteen, kun pelaajat kohtaavat tuntemattomia uhkia, jotka nousevat hämärästä taustasta ja näkökentän reunamilta.

Hahmojen uhkaavan realistiset animaatiot

Hahmojen animaatiot voidaan jakaa kolmeen eri kategoriaan: pelaajan hahmoon, ystävällisiin tai neutraaleihin ei-pelattaviin hahmoihin (englanniksi Non-Playable Character, NPC) ja vihollisiin. Pelaajan hahmon ja ei-pelattavien hahmojen animaatioiden tarkoituksena on luoda tunnelmaa ja parantaa realismia ja pelaajan immersiota. Näiden hahmojen animaatiot pyrkivät tukemaan teemaa. Niistä yleensä tehdään mahdollisimman luonnollisen oloisia liikkeiltään ja käyttäytymiseltään. Nämä hahmot eläytyvät ympäristöönsä ja välittävät kauhupelin tunnelmaa pelaajalle. Hahmot saattavat esimerkiksi olla paniikissa, liikkua kompastellen, näyttää merkkejä jännityksestä tai toimia haavoittuneena huonommalla motoriikalla.

Vihollisten animaatiot esiintyvät uhkaavina, aggressiivisina ja arvaamattomina. Animaatioiden tyyli riippuu vihollishahmon luonteesta, teemasta ja siitä, millainen olento vihollishahmo on. Tyypillisiä vihollisia ovat humanoidit ihmisen tai zombin muodossa ja kuvitteelliset hirviöt. Vihollisen toiminnan ja animaatioiden tarkoituksena on myös pitää yllä hirviön tai vihollisen mysteeriä, sillä tietty kauhu katoaa vihollisesta, jos pelaaja tietää täysin mitä odottaa.

On yleistä, että hahmojen animaatiot tuotetaan liikkeenkaappauksella. Näyttelijä pitää siis yllään liikkeenkaappauspukua ja esittää pelihahmon animaation, joka tallennetaan tietokoneelle integroitavaksi suoraan 3D-malliin. Tällä menetelmällä säästetään animaattorien aikaa ja saadaan aikaan realistisia lopputuloksia hahmojen liikkeistä ja eleistä.

2.3 Jännitystä korostava äänimaailma

Pelimaailman äänet muodostavat erittäin tärkeän ja olennaisen osan kaikkia pelejä, erityisesti kuitenkin kauhupelejä. Äänillä on merkittävä rooli pelaajan immersion ja jännityksen luomisessa: se yhdistää visuaalisen maailman yhtenäiseksi kokemukseksi.

“Siinä missä visuaalinen tieto peleissä rajoittuu näyttöön, äänet ulottuvat avaruuteen, jopa vuorovaikuttavat sen kanssa (ilmanpaineen muutokset). Äänet luovat jatkuvuutta ja läsnäoloa ja niiden seurauksena hämmentävät fiktion rajat. Tämä on osittain se, mikä tekee äänestä niin tehokkaan oikotien tunteisiin.” (Lankoski & Ekman 2009: 187–188, kirjoittajan suomennos.)

Äänimaailmaa voidaan tarkastella useista eri komponenteista, kuten interaktioäänistä, musiikista, taustäänistä ja muista ääniefekteistä. Interaktioäänillä pyritään kuvaamaan tilaa objektien kautta, kuten esimerkiksi hylätyssä mökissä äänet voivat olla narisevia, lahoja, kolkkoja ja matalia. Ääniefektien avulla pyritään säpsäyttämään pelaajaa tai ylläpitämään jännitystä. Kovat kolahdukset asioita siirrellessä, loksahdukset lukkoja avatessa tai pitkitetty narina ovia raottaessa voivat herättää turvattomuuden tunnetta pelaajassa, kun hän tiedostaa äänien kantautuvan kauas ja mahdollisesti antagonistin korviin.

Musiikilla on vaihteleva rooli kauhupeleissä. Usein äänimaailma pärjää ilman musiikkia ja keskittyy lähinnä taustääniin tunnelman luomiseksi. Musiikkia käytetään kuitenkin erityisesti jahtauskohtauksissa tai tiukoissa tilanteissa korostamaan pelaajan jännityksen tunnetta voimakkailla, nopeatempoisilla ja rytmikkäillä sävellyksillä. Kauhupelien musiikki on lähes aina progressiivista ja kehittyy kohtauksien edetessä tunnelman mukana. Se sisältää usein rytmikästä rummutusta ja korkeataajuisia jousisoittimia.

Taustäännet ja efektit esiintyvät usein matalina tai korkeataajuisina jousisoittimilla tuotettuina pätkinä. Lisäksi ne voivat olla hyvin orgaanisia sisältäen kuis-kauksia, huutoja ja hengitystä. Taustäännet seikoittavat pelaajan mieltä,

tarkoituksenaan upottaa pelaaja syvemmälle kauhun maailmaan, mikä vahvistaa kokemusta ja luo turvattoman ympäristön.

Äänimaailman suunnittelussa, erityisesti kauhun luomisessa, on syytä huomioida sen arvaamattomuus. Erityisen tärkeä ominaisuus äänisuunnittelussa on audion kolmiulotteisuus, jossa äänet ympäröivät pelaajaa joka suunnasta ja luovat syvyyttä ympäristöön ja auttamattomuuden tunnetta pelaajalle. Pelaajan ei ole tarkoitus hallita tilannetta, äänimaailma on toimiva keino pelaajan itseluottamuksen horjuttamiseen.

2.4 Pelisuunnittelun rooli kauhukokemuksen rakentamisessa

Kenttäsuunnittelu on kriittinen osa kauhupelien luomista. Pelaajaa ohjataan kohti jännittäviä ja pelottavia tilanteita erilaisilla tehtävillä tai tarkasti suunnitellulla kenttärakenteella. Kenttäsuunnittelun tarkoitus on rytmittää pelaajan etenemistä, asettaa pelaaja kauheisiin tilanteisiin ja ohjata pois niistä. Jotta kauhu ei käy toistuvaksi tai tylsäksi, on äärimmäisen tärkeää, että pelin kauhun tason tulee vaihdella. Tämä toteutetaan ohjaamalla pelaaja turvallisilta alueilta vaarallisiin ja pelottaviin tiloihin, joissa jännitys kulminoituu usein kliimaksiin eli huipentumaan. Kliimaksilla tarkoitetaan kauhupeleissä tilannetta, jossa jännitys puretaan pelkokäyttäytymisen kautta esimerkiksi jahtauskohtauksella. Jahtauskohtauksessa pelaaja altistetaan vihollisen suoranaiseksi hyökkäykselle, hän joutuu esimerkiksi pakenemaan vihollista turvalliseen tilaan.

Turvalliset tilat tai alueet ovat kliimaksin lisäksi kauhupeleissä yleisin tapa purkaa pelaajan jännitystaso. Loistavana esimerkkinä toimii The Game Awardsin vuoden peli 2023 ehdokkaaksi valitun Alan Wake 2 -pelin (Alan Wake 2, 2023) kenttiin ripotellut taukokuoneet. Kuten pelisarjan ensimmäisessä osassa pelaaja taistelee ihmismäisiä varjo-olentoja vastaan. Pimeys on lähtökohtaisesti aina uhkaava elementti, kaikki valoa tuovat asiat toimivat mekaniikkana taistella vihollisia vastaan tai turvautua niiltä. Alan Wake 2 -pelissä turvalliset alueet ovat aina valaistuja, ne erottuvat hyvin muuten varjostetun maailman siimeksestä. Tauko huoneissa on kirkkaat valot (kuva 6), joten ne pitävät pelaajan turvassa

vihollisilta ja niissä on myös mahdollista tallentaa peli manuaalisesti tai käyttää pikatallennusominaisuutta. (Rodriguez 2023.)



Kuva 6. Taukokuoneena toimiva vaja, joka on kutsuvan kodikkaasti valaistu (Alan Wake 2 2023).

Pelaaja ei vaistomaisesti aseta itseään vaaraan, joten kauhupeleissä pakotetaan pelaaja eri tavoin etenemään pelissä vaarallisille alueille. Yksi yleisimmistä tavoista toteuttaa tämä on erilaiset tehtävät. Kauhupeleissä esiintyy esimerkiksi usein tehtävä, jossa rakennuksesta sammuvat valot ja pelaajan tehtävänä on löytää generaattori, jolla ne saadaan takaisin päälle. Tehtävät toteutetaan henkisen kuorman ja paineen alla vihollisen kanssa samassa tilassa, joten ne ovat kognitiivisesti suhteellisen yksinkertaisia, mutta luovat peliin syvyyttä ja pitävät pelaajan osallisena pelimaailmaan.

Pelaajan asettaminen epä mukaviin tilanteisiin ja paineen alaisena toimiminen ovat olennainen osa kauhupelikokemusta. Turvallisten ja vaarallisten alueiden taktinen sijoittelu, kuten turvahuoneiden käyttö, korostaa tehokkaasti pelon ja turvallisuuden kontrastia. Tämä luova suunnittelu, joka nostaa pelaajan kokonaisstressitasoa ja samalla palkitsee häntä, on keskeinen elementti kauhupelien suunnittelua (Mitchell 2015). Mikäli intensiteettiä ei vaihdella eikä pelaaja saa

tarpeeksi aikaa palautua stressistä turvallisella alueella, saattaa pelikokemus tuntua turhautavalta, jolloin pelaajan mielenkiinto lopahtaa ja kauhupeli kokemuksena epäonnistuu.

Dynaaminen äänimaailma on merkittävä osa kauhupelien tunnelmaa. Se mahdollistaa äänitehosteiden säätelyn pelitilanteen mukaan, mikä edistää pelin kokonaisvaltaista immersiota. Kliimaksi on suunnittelun keskeinen hetki, joka vaatii harkittua rakentamista ylä- ja alamäkineen, paineen ja palkinnon vuorottelua niin, ettei pelaaja menetä kiinnostustaan. Dynaamisella äänimaailmalla tuetaan kenttäsuunnittelun tunnevuoristorataa ja saadaan lopputuloksena aikaiseksi koherentti jännitysnyttelmä, joka etenee progressiivisesti ja pitää pelaajan immersoituneena koko pelin ajan.

Kauhun tunnetta lisäävät komponentit, kuten arvaamattomuus, jännittyneisyyden tila, välitön kauhu, välillinen kauhu, immersion luominen, teema ja erilaisten fobioiden hyväksikäyttö. Ne ovat tärkeitä tekijöitä suunnittelussa, ja ne on syytä pitää mielessä koko prosessin ajan. Ihmiset luontaisesti etsivät kaikesta toistuvia kaavoja, joihin mieli voi tukeutua turvana, joten niiden rikkominen on pelisuunnittelun yksi tavoitteista.

Yksi erityinen pelisuunnittelun haaste on se, että useimmat kauhupelit ovat yksinpelejä. Tämä luo intensiivisen ja henkilökohtaisen kokemuksen, joka voi olla vaikea saavuttaa moninpeleissä. Kuitenkin, poikkeuksena mainittakoon Kinetic Gamesin tekemä aaveidenjahtauspeli "Phasmophobia" (kuva 7), joka on onnistunut tuomaan moninpelaamisen kauhupelikokemukseen käyttäen nokkelasti hyväksi pelin sisäistä puhetoimintoa, jonka avulla pelaajat keskustelevat toistensa kanssa. Puhetoiminnon kantavuus on rajattu hyvin lyhyelle matkalle, ja objektien lävitse puhe kuuluu vaimeasti. Näin pelaajille syntyy pelin sisällä tilanteita, joissa he ovat täysin yksin vihollisen armoilla. Myös toisten pelaajien äänten katoaminen luo epävarmuuden tilanteesta ja sitä kautta lisää jännitystä.



Kuva 7. Pelaajat keräämässä tietoa aaveesta yhdessä (Phasmophobia 2020).

Moninpelielementti on yksi harvoista asioista, jota esimerkiksi kauhuelokuvat eivät pysty tarjoamaan (Madigan 2015). Hurjien seikkailujen kokeminen toisen ihmisen kanssa samanaikaisesti niin, että molempien päätökset vaikuttavat lopputulokseen, on hyvin ainutlaatuinen kokemus ja siinä on paljon potentiaalia tulevaisuutta ajatellen.

3 Tekoäly ja sen toteutukset kauhuympäristössä

Tässä luvussa pureudutaan tarkemmin aihepiiriin käymällä läpi yleisimmät peleissä esiintyvät tekoälyn rakenteet ja miten tekoälyä käytetään kauhuympäristössä.

3.1 Tekoäly kauhupeleissä

Vaikka tekoälyn nykytila on älykkyyden suhteen varsin edistynyt, monet yritykset ja kehittäjät käyttävät edelleen yksinkertaisia toteutuksia. Kehittyneen

tekoälyjärjestelmän räätälöiminen peliksi on vaikeaa tekemättä pelistä pelaamaton tai tuottamatta surkeaa pelikokemusta. Jos tekoälyn algoritmi on liian monimutkainen tai hyvä, se saattaa olla pelaajalle ylivoimainen vastus tai vaihtoehtoisesti vain haahuilla ympäri pelikenttää tarkoituksettomasti. Toinen syy tähän on, että joskus yksinkertainen tekoäly antaa parhaan tuloksen uskottavuudessa. Tekoäly, joka on tarpeeksi monimutkainen, jotta pelaaja luulee olevansa vuorovaikutuksessa jonkin älykkään ja uskottavan kanssa, riittää useimmissa tapauksissa (Butcher & Griesemer 2002). On myös huomioitava, että kauhu on primitiivinen tunne, jota yleensä pelissä edustaa vihollisen muodossa jokin hyvin alkukantainen tai eläimellinen olento. Vaikkei se välttämättä ole suoraan syy tekoälytoteutuksen valitsemiselle, on viholliseksi valittu hahmo vaikuttava tekijä tekoälyratkaisun toteuttamisessa. Esimerkiksi kuolleista herännyt zombi, jolta on syöty pala aivoja, ei todennäköisesti ole toiminnaltaan yhtä kompleksilla tasolla, kuin tulevaisuudesta maankamaralle matkannut supersotilas.

Vihollisen ja ei-pelattavien hahmojen (NPC) käyttäytymisen manipulointia, dynaamista vaikeusasteen säätöä (Dynamic Difficulty Adjustment, DDA) ja polunhakualgoritmien käyttämistä pelihahmojen liikuttamiseen voidaan pitää yleisimpinä tekoälyn esiintymismuotoina kauhupeleissä. Tekoälyn käyttö kauhupeleissä ei ole lainkaan vieras käsite, sillä sitä on esiintynyt kautta aikojen. Vaikka Namcon Pac-Man (1980) ei ole suoranaisesti kauhupeli, sen haamut ovat varhainen esimerkki tekoälyn käyttämisestä pelissä pelaajan jahtaamiseen. Aiemmin mainittu peli Alien: Isolation on osoittanut esimerkkejä tekoälyn käytöstä tavalla, jossa vihollisen käyttäytyminen ei ole käsikirjoitettu vaan on täysin satunnaista. Tämä satunnaisuus lisää pelaajan jännitystasoa, koska hän ei voi ennustaa, mistä muukalainen ilmestyy seuraavaksi. Alien Isolation toimii tämän opinnäytetyön referenssipelinä tekoälyn toteutuksesta ja sen pohjalta on toteutettu tässä työssä luotu simulaatioympäristö. Alien: Isolation -peliä ja sen tekoälyä käsitellään tarkemmin osiossa 3.3.

Toinen nimenomaan tekoälynsä johdosta arvostettu varhainen kauhupeli F.E.A.R. on ensimmäisen persoonan psykologinen kauhuräiskintäpeli, jonka kehitti Monolithic Production vuonna 2005. Sitä voidaan pitää yhtenä

varhaisimmista kauhupeleistä, jotka sisälsivät tekoälyalgoritmeja pelattavuudessaan. Vihollisten tekoäly käyttää äärellistä tilakonetta (Finite State Machine, FSM), joka hallitsee käyttäytymistään antamalla ohjeita siitä, miten toimia kussakin tilanteessa. Pelissä on myös suunnittelujärjestelmä, joka manipuloi vihollisen tekoälyn tavoitetta pelin aikana ja antaa tekoälyn päättää, miten nämä tavoitteet saavutetaan käytettävissä olevien ohjeiden avulla (Orkin 2006). Vaikka nämä tekniikat ovat erittäin tehokkaita peleissä, jotka eivät vaadi monimutkaisella käyttäytymisellä varustettuja NPC-hahmoja, FSM- ja suunnittelujärjestelmien soveltaminen peleihin, jotka sisältävät useita käyttäytymistiloja omaavia NPC-hahmoja, voi olla monimutkaista ja vaikeaa kehittäjille. Tämän vuoksi modernit pelit suosivat käyttäytymispuiden (eng. behavior trees) käyttöä, jotta edellä mainittujen suunnittelumallien haittapuolet voidaan eliminoida.

Useimmat kauhupelit on suunniteltu kertakäyttöisiksi, eli ne pelataan läpi vain kerran. Kehittäjän näkökulmasta on kuitenkin aina kannattavaa, jos pelaaja sitoutuu peliin enemmän kuin vain minimivaatimuksen verran. Enemmän peliaikaa tarkoittaa, että pelaaja on todennäköisemmin kiinnostunut ostamaan peliin liittyviä tuotteita ja lisää todennäköisyyttä siitä, että pelaaja kertoo pelistä muille julkisten arvostelujen ja pelaajapalautteen kautta tuoden peliin lisää uusia pelaajia. Uudelleenpelattavuus on tässä avainasemassa. Jotta peli olisi uudelleenpelattava, sen täytyy tarjota jotain kiehtovaa ja mahdollisesti jotain uutta, mikä pitää pelaajan kiinnostuneena jokaisella pelikerralla. Tämä voi olla uusi pelikokemus tason suunnittelun kautta käyttämällä tekoälyllä generoituja muuttuvia tasoja, tarinan kautta tarjoamalla useita lopputuloksia pelaajan toimien mukaan tai mielenkiintoisen pelattavuuden kautta. Mielenkiintoinen pelattavuus voidaan saavuttaa esimerkiksi tekemällä tekoälyn käyttäytymisestä arvaamatonta, jolloin se voi luoda täysin uusia tilanteita pelaajalle ja näin aiheuttaa pelkoa ja jännitystä jokaisella pelikerralla. Seuraavaksi syvennyttään käyttäytymistä ohjaaviin tekoälyalgoritmeihin peleissä.

Äärellinen tilakone tekoälyn perustana

Finite State Machine eli äärellinen tilakone on konsepti, jota on käytetty videopelien tekoälyjärjestelmien rakentamisessa yli kahdenkymmenen vuoden ajan. Niiden suosion pitkälti määritteli Valven vuoden 1998 klassikko toimintapeli, Half Life. Vaikka FSM:ät eivät ole enää yhtä yleisiä kuin ennen, ne ovat edelleen loistava lähtökohta monelle tekoälyimplementaatiolle.

Äärellisen tilakoneen perustoimintaperiaate on yleisesti simuloida yksinkertaista sekventiaalista logiikkaa (Nystrom 2011). Se perustuu suurelta osin kahteen George H. Mealyn ja professori Edward F. Mooren vuonna 1955 ja 1956 tekemään työhön. FSM-järjestelmä koostuu yhdestä tai useammasta ennalta määritellystä tilasta. Kun mallinnetaan tekoälykäyttäytymistä peleissä, tila edustaa tiettyä käyttäytymistä, jonka hahmo tai muu järjestelmä pelissä suorittaa. Tämä voi olla seisominen paikallaan, pelaajaa vastaan hyökkääminen, pisteeseen siirtyminen maailmassa, esineen kanssa vuorovaikutus, mitä tahansa suunnittelija näkee sopivaksi. Usein tämä tarkoittaa, että tilassa käsitellään erilaisia pelijärjestelmien osa-alueita, kuten animaatiota, ääntä ja päätöksentekoa sille järjestelmälle, jota tilakone hallitsee. Tilakone pitää nykyisen tilan aktiivisena, kunnes se vastaanottaa ennalta määritellyn syötteen, jonka johdosta se siirtyy toiseen tilaan järjestelmässä.

```

enum State
{
    STATE_STANDING,
    STATE_JUMPING,
    STATE_CROUCHING,
    STATE_DIVING
};

void Character::handleInput(Input input)
{
    switch (state_)
    {
        case STATE_STANDING:
            if (input == PRESS_SPACEBAR)
            {
                state_ = STATE_JUMPING;
                yVelocity_ = JUMP_VELOCITY;
                setAnimation(ANIMATION_JUMP);
            }
            else if (input == PRESS_DOWN)
            {
                state_ = STATE_CROUCHING;
                setAnimation(ANIMATION_CROUCH);
            }
            break;

        case STATE_JUMPING:
            if (input == PRESS_DOWN)
            {
                state_ = STATE_DIVING;
                setAnimation(ANIMATION_DIVE);
            }
            if (isGrounded)
            {
                state_ = STATE_STANDING;
                setAnimation(ANIMATION_STAND);
            }
            break;

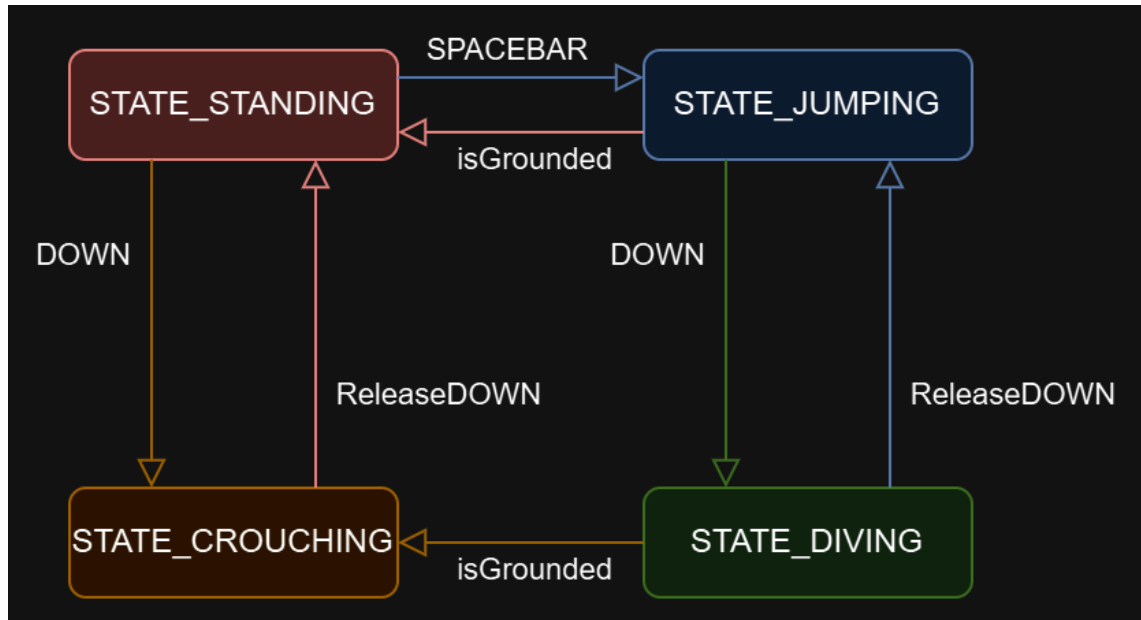
        case STATE_CROUCHING:
            if (input == RELEASE_DOWN)
            {
                state_ = STATE_STANDING;
                setAnimation(ANIMATION_STAND);
            }
            break;

        case STATE_DIVING:
            if (input == RELEASE_DOWN)
            {
                state_ = STATE_JUMPING;
                setAnimation(ANIMATION_JUMP);
            }
            if (isGrounded)
            {
                state_ = STATE_CROUCHING;
                setAnimation(ANIMATION_CROUCH);
            }
            break;
    }
}

```

Esimerkkikoodi 1. Pseudokoodi äärellisestä tilakoneesta. Funktiota `handleInput` kutsutaan jokaisella kuvan virkistyksellä (frame). Koodissa seurataan pelille annettua syötettä ja sen mukaan vaihdetaan määriteltyä tilaa eli "State":a. Tilan yhteyteen voidaan tehdä lisää haluttuja määritelmiä, kuten nopeuden tai animaation muutoksia. Kyseisessä pseudokoodissa tilan muutoksen yhteydessä

vaihdetaan hahmon animaatiota "setAnimation"-funktiolla, sekä annetaan y-suuntaista nopeutta "yVelocity", mikäli "STATE_STANDING" -tilassa annetaan syötteenä välilyönti.



Kuva 8. Esimerkkikoodin 1 mukaan hahmoteltu äärellinen tilakone. Värikoodaus auttaa hahmottamaan, mihin tilaan kukin syöte johtaa.

Pelin kehittäjä voi päättää, mitkä syötteet tila vastaanottaa, jotta siirtymä voi tapahtua, sekä mihin tiloihin se siirtyy tämän tiedon perusteella. Yhdestä tilasta voidaan siirtyä yhteen tai useampaan tilaan. Tämä johtaa joko deterministiseen äärelliseen tilakoneeseen, jossa tila lukee syötteen ja voi siirtyä vain yhteen toiseen tilaan, tai ei-deterministiseen äärelliseen tilakoneeseen, mikä tarkoittaa, että tietyllä syötteellä järjestelmä voi siirtyä useisiin eri tiloihin (What Is The Difference Between DFA and NFA?, 2024).

Äärellisen tilakoneen etuna on, että voidaan määritellä useita tiloja, jotka yhdessä muodostavat mielenkiintoisen käyttäytymiskokonaisuuden. Hyvä esimerkki on Pac-Man-pelin tekoäly, jossa kummituksen tila, jossa se jahtaa pelaajaa, on ainutlaatuinen jokaiselle viholliselle, mutta abstraktilla tasolla ne kaikki tekevät samaa asiaa. Kun pelaaja nappaa voimapillerin, kaikki kummitukset siirtyvät väistämistilaan, mikä muuttaa niiden käyttäytymistä pelimaailmassa

pakenemiseen. Voimapillerin aikana myös pelin visuaalinen ilme ja musiikki muuttuu tilan mukaan. Kun voimapillerin vaikutus lakkaa, pelin ilme palaa ennalleen ja kukin kummitus voi siirtyä takaisin jahtaamaan pelaajaa.

Äärellistä tilakonetta hyödyntäen voidaan luoda tekoäly, joka voi reagoida maailmassa tapahtuviin tapahtumiin ja pelaajan toimintaan suhteellisen monipuolisesti toimien riittävän uskottavasti monissa tilanteissa. Äärelliset tilakoneet olivat monien vuosien ajan standardi tekoälyn rakentamisessa peleihin 2000-luvun puolivälistä loppupuolelle. FSM:n heikkous piilee halutun käyttäytymisen monimutkaisuuden ja toiminnallisuuden skaalauksessa. Mitä enemmän eri tiloja pelin kehittäjä haluaa luoda tekoälylle, sen haastavammaksi tilojen välillä hyppiminen muuttuu kehittäjän näkökulmasta. Uuden tilan luominen todella kompleksiin systeemiin muuttuu työlääksi, joten äärellisiä tilakoneita suositaan yksinkertaisissa tekoälytoteutuksissa (State Machines vs Behavior Trees: designing a decision-making architecture for robotics 2023).

Sumean tilakoneen hyöty- ja haittapuolet

Fuzzy State Machines (FuSM) eli sumeat tilakoneet laajentavat perinteistä FSM:ää sisällyttämällä siihen niin sanotun sumean logiikan, mikä mahdollistaa hienostuneemman ja joustavamman tekoälyn käyttäytymisen (Reyneri 1997). Toisin kuin FSM:ssä, jossa tilat ovat toisensa poissulkevia, FuSM:ssä tilat voivat olla osittain aktiivisia samanaikaisesti, ja jokaisella tilalla on oma aktivointitaso. Tämä lähestymistapa mahdollistaa sujuvamat siirtymät tilojen välillä (Champandard, 2003) ja mukautuvamman käyttäytymisen, mikä voi parantaa pelihahmojen realismia ja monimutkaisuutta. Esimerkkinä sumeasta tilakoneesta reitti-valinnoissa olisi se, että sen sijaan, että algoritmi toimisi tiukasti tiloissa kuten ”liiku eteenpäin” tai ”käänny vasemmalle”, sumea tilakone mahdollistaa tekoälyn olevan yhdistelmätiloissa, joilla on vaihtelevia aktivointitasoja, kuten ”liiku eteenpäin” ja ”käänny vasemmalle” yhtäaikaisesti. Tämä mahdollistaa tilanteessa reitin sujuvamman mukauttamisen esteiden mukaan.

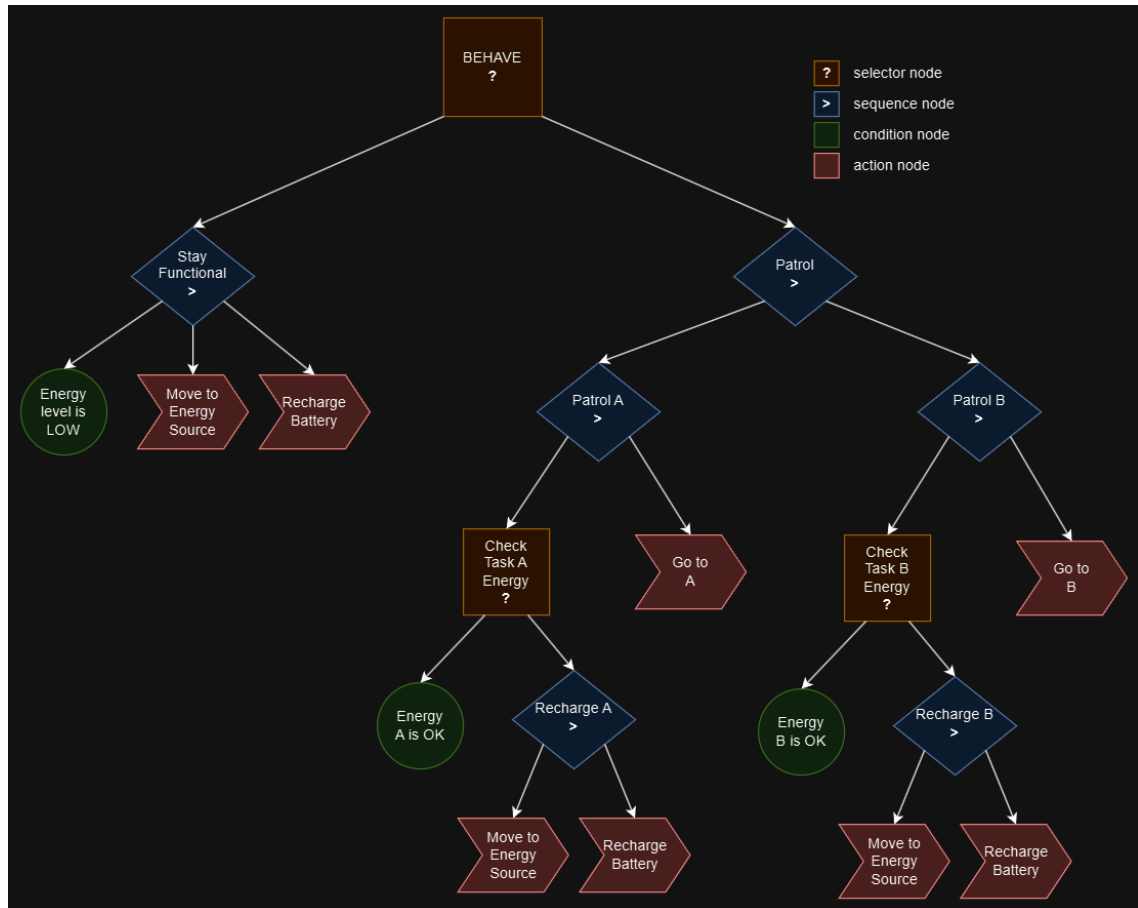
Sumean tilakoneen heikkoudet piilevät sen rakenteessa ja sen implementointi on hieman monimutkaisempaa kuin pelkän äärellisen tilakoneen. Ne ovat näistä syistä riippumatta hyvin yleisiä ja löytyvät esimerkiksi peleistä F.E.A.R, Sims 4, Left 4 Dead ja S.T.A.L.K.E.R.

Käyttäytymispuut modernissa tekoälyssä

Käyttäytymispuut ovat nykyään ehdottomasti yleisin tapa luoda hienostuneempia tekoälyimplementaatioita peleihin. Niiden selkeä rakenne ja helppous luoda lisää toiminnallisuutta olemassa olevan päälle on syy siihen, miksi ne ovat nousseet yli tilakoneista. Käyttäytymispuita voidaan kuvata "hierarkkisista solmuista koostuvaksi puuksi, joka ohjaa tekoälyentiteetin päätöksenteon kulkua" (Simpson, 2014). Yksi tämän suunnittelumallin tärkeimmistä eduista verrattuna FSM:iin on sen joustava rakenne, jonka ansiosta kehittäjät voivat lisätä uusia käyttäytymismalleja puuhun. Tämä tekee järjestelmästä huomattavasti helpommin laajennettavan samalla säilyttäen sen yksinkertaisena ja ymmärrettävänä kokonaisuutena.

Käyttäytymispuut koostuvat solmuista ja niiden juurikin on itsessään solmu. Jokaisella solmulla voi olla useampi lapsisolmu, yksi lapsisolmu tai ei yhtäkään lapsisolmuja, jolloin solmu johtaa lehtisolmuun (eng. leaf). Lehtisolmu pitää sisällään tyypillisesti jonkin toimenpiteen. On olemassa muitakin tärkeitä solmutyyppejä solmun ja lehtisolmun lisäksi, joita kutsutaan yhteisnimikkeellä komposiittisolmu. Komposiittisolmu voi olla esimerkiksi valitsija tai sekvensseri. Valitsija nimensä mukaisesti sisältää jonkinlaisen OR-lausekkeen, mikä johtaa useaan lehtisolmuvaihtoehtoon. Toinen komposiittisolmujen päätyyppi on sekvensseri. Sekvensseri on looginen AND-lauseke ja käy siis läpi jokaisen lehtisolmun, johon se johtaa järjestyksessä. Jokaisen lehtisolmun toimenpiteen on toteutettava, jotta sekvenssi toteutuu. Näiden komposiittisolmujen lisäksi lähes aina esiintyy myös konditionaalisolmu, jota käytetään ehtolausekkeen tapaan puussa etenemisen yhteydessä. Käyttäytymispuu voi olla n-solmua syvä ilman mitään rajoituksia, mutta yleisesti lehtisolmut johtavat johonkin ehtoon tai toimintoon.

Hyvä esimerkki käyttäytymispuiden käytöstä voidaan nähdä Creative Assemblyn pelissä Alien: Isolation (2014), jota tarkastellaan tämän opinnäytetyön osiossa 3.3.



Kuva 9. Esimerkkihahmotelma käyttäytymispuusta. Kyseisessä puussa tekoöly tarkistaa ensin oman energiatasonsa ja lataa akkunsaa tarvittaessa. Tämän jälkeen tekoöly aloittaa partioinnin käymällä partiointipisteet A ja B läpi. Kunkin partiointipisteen kohdalla tekoöly tarkistaa, ovatko pisteiden energiatasot OK. Jos eivät ole, se lataa kyseessä olevan pisteen akun.

Triggerit

Yleinen pelisuunnittelun tapa luoda tunnelmaa tai ohjata pelikokemusta tapahtuu niin sanotuilla laukaisimilla (eng. trigger), jotka kehittäjä tai suunnittelija asettaa manuaalisesti suorittamaan tietyn pelitapahtuman, laukaisukriteerin täytyessä. Nämä pelitapahtumat eivät suoranaisesti ole erillinen kokonaisuus tekoälystä vaan esiintyvät liitettynä yhteen tekoälyn, kuten esimerkiksi ohjaajan

tai tilakoneen, toimintamalliin. Triggerit voivat olla äänilaukaisuja, kuten salaman iskuja ja narahduksia tai toimintalaukaisuja, kuten oven avautumisia tai vihollisen hyökkäyksiä. Usein nähty esimerkki laukaisusta on vihollisen ilmestyminen pelaajan taakse pelästyttämismalliin, kun pelaaja on kerännyt jonkin tärkeän esineen maasta.

Pelitapahtuman laukaisimien manuaalinen säätäminen on hienoa, koska kehittäjä voi räätälöidä tietyn toimintasarjan käsin tai luoda tunnelmallista kauhua suhteellisen vaivattomasti ja nopeasti. Tämän lineaarisen maailmanrakentamisen haittapuoli on, että käytettynä suurissa määrin se vie paljon pois pelosta ja jännityksestä, kun pelaaja päättää pelata pelin uudelleen, ja se on yksi syy siihen, miksi useimmat kauhupelit on tarkoitettu kertakäyttöisiksi.

3.2 Tekoälyn parametrisointi kauhu -ympäristössä

Tekoälyn parametrisointi voidaan jakaa kolmeen suurempaan osa-alueeseen: käyttäytymiseen liittyvien algoritmien parametrit, pelaajakohtaiset parametrit sekä tila- ja tilannekohtaiset parametrit ja niiden hallinta.

Pelaajakohtaisten parametrien hyväksi käyttäminen

Jotta pelikokemuksesta saadaan räätälöityä mahdollisimman mielenkiintoinen pelaajalle, on suotavaa, että pelissä kerätään informaatiota pelaajan tunteista ja pelityylistä. Pelaajan tunnetilaa pystytään arvioimaan pelaajan pelityylin, etenemisnopeuden ja muun käyttäytymisen pohjalta. Unity-pelimoottorissa on esimerkiksi olemassa ohjelmistokehityspaketti MoodMe (liite 2), jonka avulla pystytään keräämään pelaajan ilmeiden perusteella dataa reaaliajassa ja käyttämään tätä dataa laukaisemaan eri sekvenssejä pelissä tai muuttamaan pelitilannetta pelaajan tunnetilan mukaan.

Japanilainen yliopisto Aizu teki tutkimuksen ilmeentunnistuksen integroimisen mahdollisuudesta kauhupelieihin ja sen tehokkuudesta parantaa pelaajien sitoutumista ja uppoutumista pelitilanteeseen (Tu & Mozgovoy 2023). Tutkimuksessa

luotiin kaksi erillistä versiota pelistä, jossa toisessa pelitapahtumia laukaistiin satunnaisesti ja toisessa laukaisut perustuivat kameran kuvasta välittyviin pelaajan tunnereaktioihin, jotka analysoitiin koulutetulla neuroverkolla. Tutkimuksen tuloksissa tunnepohjaisilla tapahtumien laukaisusysteemeillä havaittiin huomattavan suuri positiivinen vaikutus osallistujien immersiotasoon. Osallistujat pystyivät erottamaan pelin eri versiot toisistaan ja suurin osa ilmoitti tunnepohjaisella laukaisusysteemillä toimivan version olevan mukaansatempaavampi. Kuten kuvasta 10 kolmanneksi alimmasta sarakkeesta voidaan todeta, testiryhmästä 70,6 % reagoi tunnepohjaisiin pelitapahtumien laukaisuihin paremmin. 58,8 % osallistujista suosi tunnepohjaisilla laukaisimilla toimivaa versiota satunnaisesti toimivan version sijaan. Tutkimuksen testiryhmän koko oli 17 osallistujaa.

TABLE III
EXPERIMENT RESULT

Aspect	Description	Result
Perception of Difference	Noticed difference due to different triggers	29.4%
Perception of Difference	Felt some parts of the game were different	29.4%
Perception of Difference	Total participants who felt a difference	70.6%
Awareness of Own Emotions	Aware of emotions during gameplay	47.1%
Awareness of Own Emotions	Felt emotions triggered game events	41.2%
Awareness of Own Emotions	Participants not paying attention to emotions	52.9%
Evaluation of Game Immersion	Preferred game with emotion triggers	58.8%
Evaluation of Game Immersion	Preferred game with random triggers	35.3%
Evaluation of Game Immersion	Both versions equally immersive	5.9%
Evaluation of Game Immersion	Average score for more immersive version	6/10
Evaluation of Game Immersion	Most common rating for more immersive version	7/10
Emotional Responses	Higher emotional responses in game with emotion triggers	70.6%
Emotional Responses	Higher emotional responses in game with random triggers	23.5%
Emotional Responses	No significant emotional changes in either version	5.9%

Kuva 10. Yliopisto Aizun tekemän tutkimuksen tulokset.

Onnistuneiden tulosten johdolla tutkimusryhmä mainitsee kuitenkin erinäisistä haasteista kehityksessä: Jännittyneen ihmisen tunteiden lukeminen kasvoilta on haastavaa. Neuroverkon koulutusdatan tunnetilat olivat hieman liioiteltuja, joten mallilla oli vaikeuksia tunnistaa pieniä muutoksia ilmeissä. Testiryhmän osallistujat mainitsivat myös yksilöllisten pelitapahtumien määrän olleen hieman suppea.

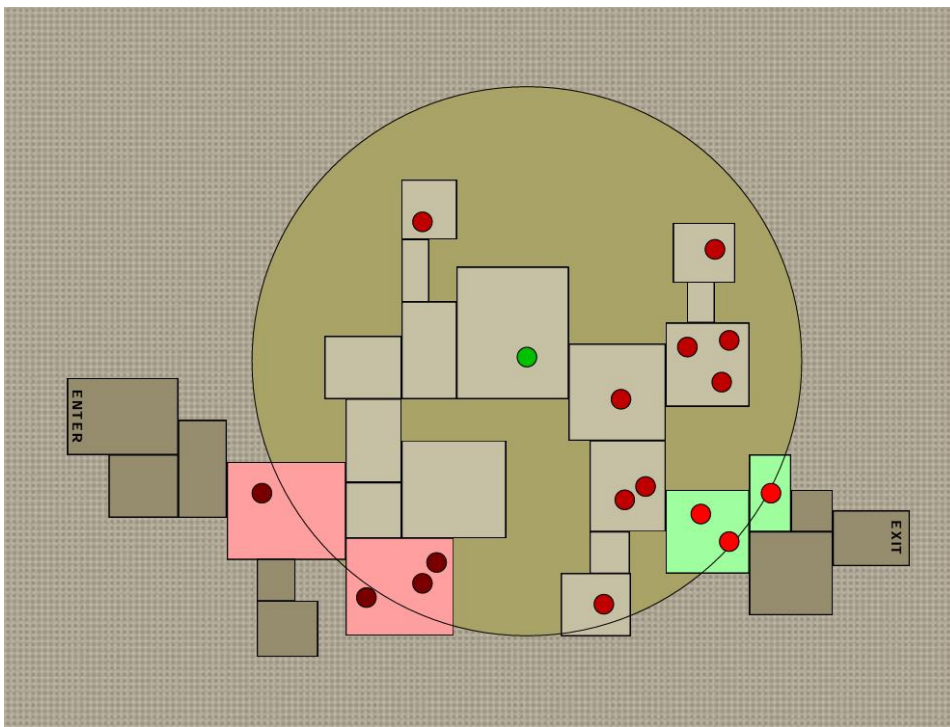
Ohjaajat hallitsemassa pelin tilannetta

Tekoälyä käytettäessä kauhupeleissä kehittäjiä on tärkeää tasapainottaa pelin kauhuelementtien parametrejä niin, että pelaaja kokee eri tason jännitystä ja pelkoa, muttei silti lamaannu ja lopeta pelisessiota kesken. Tämän tasapainon saavuttamiseksi on kehitetty kauhupeleissä tekoälysystemi nimeltään "The Director" eli ohjaaja. Ohjaajan tehtävänä on ohjata ja monitoroida pelin muita tekoälyjä ja niiden käyttäytymistä sekä pelaajan toimintaa. Ohjaaja voi pitää sisälleen informaatiota pelin tilasta, pelaajan ja vihollistekoälyn sijainnista, pelaajan pelityylistä ja muusta. Ilman ohjaajaa monimutkaisella algoritmilla toimiva tekoäly saattaa olla vaikea tasapainottaa mielekkääksi vastustajaksi. Se on helposti liian voimakas este pelaajan etenemiselle tai saattaa vain vaeltaa pelikentässä päättömästi vaikuttamatta pelaajaan millään tavalla.

Ohjaajan idea on toimia nimensä mukaan kuin elokuvan ohjaaja. Se syöttää tekoälylle ennalta määritetyllä tavalla lisäinformaatiota pelin sekä pelaajan tilanteesta ja tätä kautta vaikuttaa sen toimintaan tasapainottaen pelikokemusta tarpeen mukaan. Tällainen tasapainotus voidaan toteuttaa pelin vaikeusasteen muutoksilla. Nykyään kauhupeleissä esiintyy toistuvasti dynaaminen vaikeusasteen vaihtelu (Dynamic Difficulty Adjustment, DDA), joka nimensä mukaan muuttaa pelaajan pelikokemusta vaikeusasteen kautta pelin aikana. Tämän systeemin ideana on luoda jokaiselle henkilökohtainen omaa taitotasoa vastaava pelikokemus, joka pitää mielenkiinnon yllä.

Parhaana esimerkkinä ohjaajasta ja tekoälyn ohjaaja -konseptin esi-isänä pidetään 2008 vuonna julkaistua peliä *Left 4 Dead* (Turtle Rock Studios). *Left 4*

Dead -peli on ensimmäisen persoonan kauhu räiskintäpeli, jossa pelaajan tehtävänä on selvitä yksin kolmen NPC:n kanssa tai jopa neljän pelaajan yhteispelissä zombie-armeijan taukoamattomasta hyökkäyksestä maailmanlopun jälkeisessä Amerikassa. Left 4 Dead sai paljon positiivista palautetta sen uudelleenpelattavuudesta. Jokainen pelikerta on hieman erilainen ohjaajan toimintatavan johdosta. Ohjaajan pohjalla pyörii äärellinen tilakone (Finite State Machine, FSM), joka vaihtelee ohjaajan päätavoitetta, pelin tilaa ja tempoa tiettyjen parametrien täytyessä ja kuluessa (Booth 2009). Tärkeimmät parametrit, joita ohjaajalla on tiedossaan, ovat jokaisen pelaajan aktiivinen asetettu alue (Active Area Set, AAS) ja pelaajan ”emotionaalinen intensiteetti” eli stressitaso. Pelin taustalla pelikentät on pilkottu pieniin numeroituihin alueisiin. AAS:llä tarkoitetaan jokaisen pelaajan ympärillä olevia lähialueita.



Kuva 11. Aktiivinen asetettu alue (AAS) pelaajan ympärillä. Vihreä piste kuvastaa pelaajan sijaintia ja punaiset pisteet vihollisia.

Kuvassa 11 näkyvät kentän alueet ovat pilkottuina osiin ja pelaajan ympärillä oleva laaja ympyrä, joka on tämän tilanteen AAS. Vihreällä merkityt nelikulmiot ovat alueita, jotka ovat juuri liikkumassa AAS:n sisään, jolloin ohjaaja synnyttää

niihin hajanaisia vihollisia. Punaisella merkityt nelikulmiot ovat alueita, jotka ovat vuorostaan jäämässä AAS:n ulkopuolelle, jolloin niiden sisältämät viholliset tuhoetaan ohjaajan toimesta. Aktiivinen asetettu alue liikkuu pelaajan mukana keskitettynä pelaajaan. Ohjaaja pitää muistissaan kirjaa jokaisen jaetun alueen vihollisten ja esineiden määrästä ja synnyttää tai poistaa alueiden sisältämät asiat niiden aktiivisuuden mukaan. Tätä tietoa ohjaaja käyttää hyväksi vihollisten synnyttämiseen sekä kerättävien resurssien asetteluun.

Pelaajan stressitaso on liukuluku nollan ja yhden välillä, mikä alkaa nollost ja ollessaan yli lähtötason laskee kohti nollaa ajan kuluessa. Pelaajan stressitaso nousee, kun pelaaja ottaa osun vihollisen hyökkäyksestä, kun pelaaja on tainnutettuna maassa tai, kun vihollinen kuolee pelaajan lähetyvillä. Vihollisen kuollessa stressitason kasvumäärä vaihtelee vihollisen etäisyydestä pelihahmoon.

FSM vaihtelee pelin tilaa neljän eri vaiheen välillä (The Director AI of Left 4 Dead 2014). Nämä ovat kasvu, jatkuva huippu, laskeva huippu ja rauhoitus. Jokaisen kentän alussa ohjaaja aloittaa kasvutilasta, asettaa uhkapopulaation 100 %:iin ja asettaa seuraavan zombilauman syntymiseen lähtölaskennan 90 – 180 sekunnin väliltä määrittäen syntymähetkellä lauman suuruuden pelaajien stressitason mukaan. Myös jokaiseen pelialueen osaan syntyy haahuilevia normaali-vihollisia, joiden tarkoitus on luoda tunnelmaa. Lähtölaskennan jälkeen ohjaaja jatkaa vihollisten synnyttämistä karttaan pelaajien stressitason nostamiseksi. Tietyn väliajoin ohjaaja synnyttää pelaajien läheisyyteen kolmesta eri vaihtoehdosta erikoiszombin, jonka tarkoitus on luoda kaaosta ja lisätä pelaajien stressitasoa zombilauman lisäksi. Kun jokaiset pelihahmon stressitasot ovat ylittäneet tietyn raja-arvon, ohjaajan tila vaihtuu kasvusta jatkuvaan huippuun muutamaksi sekunniksi. Tänä aikana ilmestyy maksimaalinen määrä vihollisia pelaajien läheisyyteen, jonka jälkeen tilakone siirtyy jatkuvasta huipusta hiipuvaan huippuun.

Hiipuvan huipun aikana karttaan ei synny enää ylimääräisiä vihollisia ja tiimille annetaan mahdollisuus rauhoittaa tilanne hävittämällä viimeiset hyökkääjät

kimpustaan. Pelaajien stressitasoa monitoroidaan aktiivisesti, kun ne laskevat huipputasoon alle, pelin tila vaihtuu rauhoitustilaan. Rauhoitustila kestää 30 – 45 sekuntia ja sen aikana peliin ei synny vihollisia, vaan pelaajien annetaan rauhassa ladata aseensa, ryhmittä uudelleen, parantaa haavansa ja edetä pelin kartassa. Rauhoitustila on olennainen pelaajien henkisen hyvinvoinnin kannalta, jotta peli ei kävisi liian monotoniseksi tai raskaaksi.



Kuva 12. Vihollislauman hyökkäys pelaajien kimppuun toimistorakennuksessa (Left 4 Dead 2008).

Näiden neljän tilan välillä liikkumisen nopeus vaihtelee: jos pelaajat pelaavat tiiminä ja etenevät tasossa tasaisella vauhdilla, ohjaaja on itse asiassa vähemmän rankaiseva. Mutta jos pelaajat liikkuvat liian hitaasti tai yrittävät vain juosta tason läpi, ohjaaja on paljon aggressiivisempi ja asettaa pelaajille paljon haastavimmat olosuhteet. Se myös huomioi, pysyykö tiimi yhdessä ja lähettää yksin pelaavien pelaajien kimppuun enemmän vastustajia sekä antaa heidän läheisyyteensä vähemmän ensiaputarpeita. (Booth 2009.)

Samanlaiset tekoälyohjaajat esiintyvät monissa Left 4 Deadin kaltaisissa jatka-jissa, kuten Vermintide, Back 4 Blood tai Payday 2. Ohjaajien idea on kuitenkin sovellettavissa myös muihin genreihin. Avoimen maailman pelisarjat, kuten Far

Cry ja The Witcher, käyttävät ohjaajajärjestelmiä palvelemaan omia pelisuunnittelutarpeitaan. Tämä voi tarkoittaa sen varmistamista, että tekoälyhahmoja syntyy vain lähellä pelaajaa, jotta suorituskykykustannukset pysyvät alhaisina, tai sen tasapainottamista, kuinka monta vihollista hyökkää pelaajan kimppuun kerralla ja kuinka voimakkaita ne ovat. Ohjaajat toimivat suurissa tekoälykokonaisuuksissa hyvin kriittisessä roolissa, mikä tasapainottaa pelikokemusta. Left 4 Dead on vain yksi esimerkki muiden joukossa, jossa kompleksista systeemistä saadaan äärimäisen mielenkiintoinen uudelleenpelattava kokonaisuus.

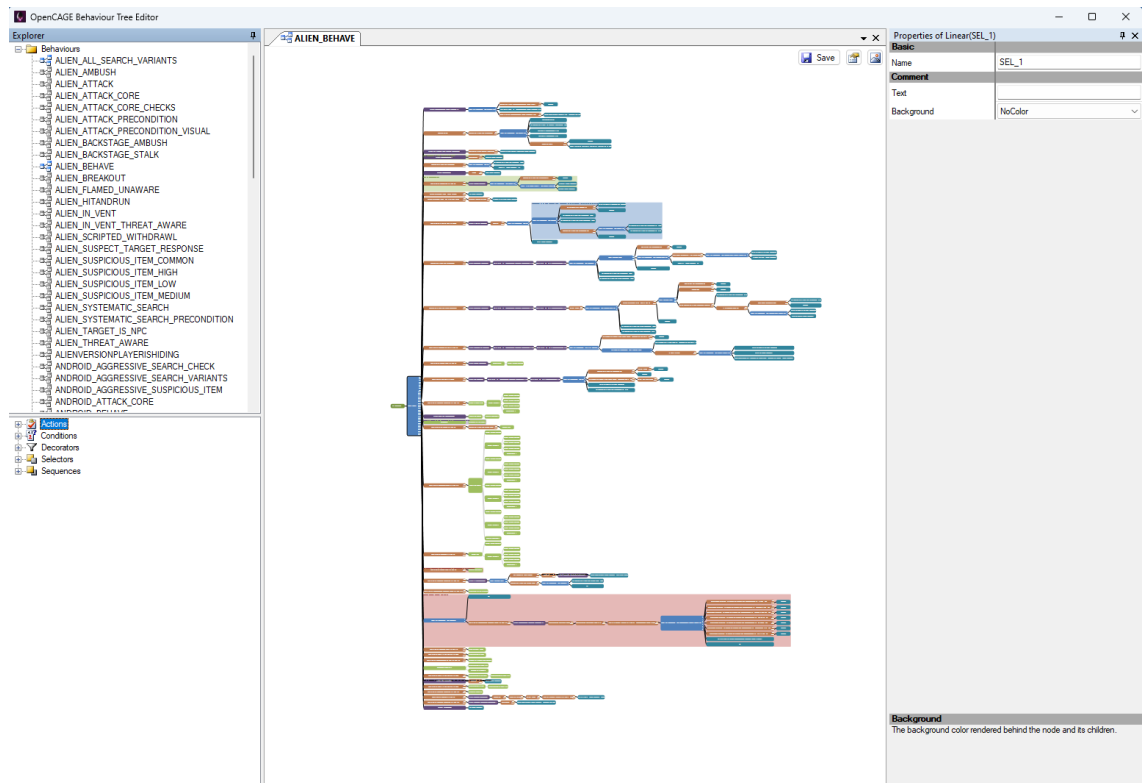
Seuraavassa osiossa tutustutaan tarkemmin Alien: Isolation -pelin tekoälytoteutukseen, jossa toimii kompleksilla käyttäytymispuulla varustettu vihollistekoäly, joka saa ohjaajatason tekoälyltä käskyjä toimintaansa näin luoden yhden arvostetuimmista ja mielenkiintoisimmista kauhupelikokemuksista pelaajan näkökulmasta.

3.3 Alien: Isolation, esimerkki toimivasta kauhupelin tekoälystä

Tämän opinnäytetyössä luodun tekoälyn referenssinä käytetään Creative Assemblyn vuonna 2014 luomaa peliä Alien: Isolation, jonka tekoälyä pidetään edelleen kauhupeligenren yhtenä parhaista ja pelottavimmista toteutuksista kautta aikain. Alien: Isolation on selviytymiskauhupeli, joka pohjautuu Ridley Scottin ohjaamaan scifi-kauhuelokuvaan Alien (1979). Peli sijoittuu 15 vuotta elokuvan tapahtumista eteenpäin seuraten elokuvan päähenkilön Ellen Ripley'n tytärtä Amanda Ripleytä, kun hän yrittää löytää kadonnutta äitiään Sevastopol -nimiseltä suurelta avaruusalukselta. Avaruusalus on kaaoksen vallassa ja Amandan tehtävä muuttuukin alukselta pakenemiseksi vapaaksi päässeeseen avaruusolio Xenomorphin verenhimoisen metsästyksen johdosta. Alien: Isolation pelikokemuksena painottuu selviytymiseen, hiiviskelyyn ja pelaajan nokkeluuteen ylivoimaista organismia vastaan erinäisten tehtävien tähdittämänä.

Avoimen lähdekoodin OpenCAGE-työkalua (liite 1) käyttäen päästään käsiksi Alien: Isolation -pelin eri tekoälyjen toimintaan. Kuvassa 13 nähdään

päävihollisen eli Alienin käyttäytymispuu kokonaisuudessaan avattuna OpenCAGE-ohjelmassa.



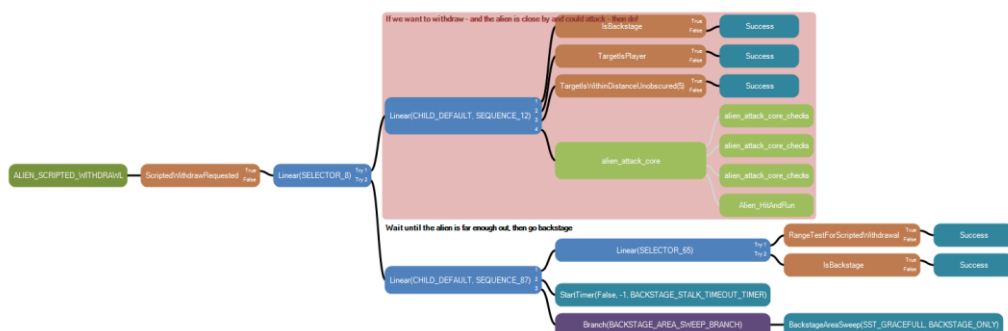
Kuva 13. Alienin käyttäytymispuu avattuna OpenCAGE-työkalulla. Ikkunan oikeassa laidassa on hieman tietoa valitusta solmusta. Vasemmalla on lista kaikista pelissä käytetyistä erillisistä käyttäytymispuista sekä Alienin käyttäytymispuun eri haaroista (OpenCAGE liite 1).

Pelissä on eri käyttäytymispuut neljälle päähahmotyypille: Alien (vihollinen), androidi, ihminen ja NPC. Työkalua käyttäen voidaan tarkkailla ja muokata jokaista pelin hahmon käyttäytymispuuta. Valitettavasti OpenCAGE:lla ei pysty näkemään tarkempaa informaatiota yksittäisistä muuttujien arvoista tai siitä, miten eri solmuihin päädytään pelin aikana. Työkalulla kuitenkin saadaan tarkka kokonaiskuva tekoälyjen päätöksenteosta, minkä pohjalta voidaan oppia paljon ja rakentaa oma pelkistetty versio simulaatioympäristöä varten. Tässä opinnäytetyössä keskitytään vain Alienin eli Xenomorphin käyttäytymispuuhun.

Alienin tekoälyn toimintaperiaate

Alien: Isolation -pelin vihollisen tekoäly koostuu kahdesta komponentista, aiemmin mainitusta ylemmän tason directorista eli ohjaajasta, jonka perusidea on samaa laatua kuin Left 4 Dead -pelin ohjaajalla, sekä käyttäytymispuulla varustetusta alemman tason tekoälystä. Nämä komponentit toimivat keskenään ohjaten pelin tekoälyä koko pelikokemuksen ajan. Lisäksi tekoälyllä on muutamassa pelin avainkohdassa manuaalisesti sijoitettuja laukaisuelementtejä, mutta tässä opinnäytetyössä niitä ei huomioida sen tarkemmin.

Xenomorphin tekoälyllä eli alemman tason tekoälyllä on kaksi eri pääkäyttäytymismallia, ”frontstage” eli etuala ja ”backstage” eli taka-ala, minkä välillä ohjaaja vaihtelee eri parametreista riippuen. Etualan vallitessa vihollinen on aggressiivinen ja vaeltaa pelikentän aluetta aktiivisesti etsien pelaajaa. Taka-alalla vihollisen käyttäytyminen muuttuu passiiviseksi, jolloin se yleensä piiloutuu pelikenttään vaanimaan tai odottamaan, jonka ansiosta pelaaja pääsee helpommin etenemään pelin tehtävien parissa. Kuvassa 14 on OpenCAGE työkalulla avatun Xenomorphin taka-alalle siirtymisen käyttäytymispuun solmut. Vetäytyessään taka-alalle Alien yrittää ensin tarkastaa, onko pelaaja hyökkäämisetäisyydellä. Mikäli on, se hyökkää ensin pelaajaan ja sitten pakenee piiloon. Jos pelaaja ei ole hyökkäyksen etäisyydellä, Xenomorph vetäytyy ”backstage”-moodiin.



Kuva 14. Xenomorphin taka-alalle siirtymisen eri vaiheet ja tarkistukset (OpenCAGE liite1).

Alienin käyttäytymispuussa pää käyttäytymismallien välinen vaihtelu on toteutettu uhkaavuusmittarilla. Uhkaavuusmittari toimii pelaajan tunteiden parametrinä ja sillä pyritään arvioimaan pelaajan stressitasoa mittaamalla vihollisteköilyn läsnäoloa pelaajan perspektiivistä. Alienin viettäessä aikaa pelaajan läheisyydessä, hyökätessään tai vaaniessaan pelaajaa uhkaavuusparametrin arvo kasvaa. Uhkaavuusarvon kasvumäärään vaikuttaa myös muun muassa Xenomorphin näkeminen, kuuleminen ja sen etäisyys pelaajasta. Kun uhkaavuusarvo ylittää ennalta määritellyn rajan, Xenomorph siirtyy taka-alalle rauhoittaakseen tilanteen ja antaa näin pelaajalle tilaa toimia. Näin pelissä saadaan aikaan jännityksen vaihteleva tasapaino. Jotta pelaaja ei tuntisi oloaan kuitenkaan täysin turvalliseksi Xenomorphin ollessa taka-alalla, se saattaa jättää pelaajalle ansan.

Ohjaajalla on koko ajan tieto sekä Alienin että pelaajan tarkasta sijainnista. Jotta pelin vastustaja ei olisi ylivoimainen, Xenomorph toimii täysin omatoimisesti omana entiteettinään vaeltaen karttaa tehden itse päätöksiä omista toimitaan. Alienilla ei ole koskaan tarkkaa tietoa pelaajasta, se luottaakin siksi omiin aistimuksiinsa. Ohjaaja syöttää sille satunnaisesti approksimoidun tiedon alueesta, jonka sisällä pelaaja on ollut. Tämän pohjalta Xenomorph voi päättää jatkaa metsästystään sille ohjaajan antamassa sijainnissa. Alemman tason teköilyn pohjana on käyttäytymispuu, joka pitää sisällään yli sata eri solmua. Alienin päätoiminnot ovat etsintä, jahtaus, äänien ja asioiden tutkiminen, hyökkäys, piiloutuminen, näkö, tunto ja kuulo.

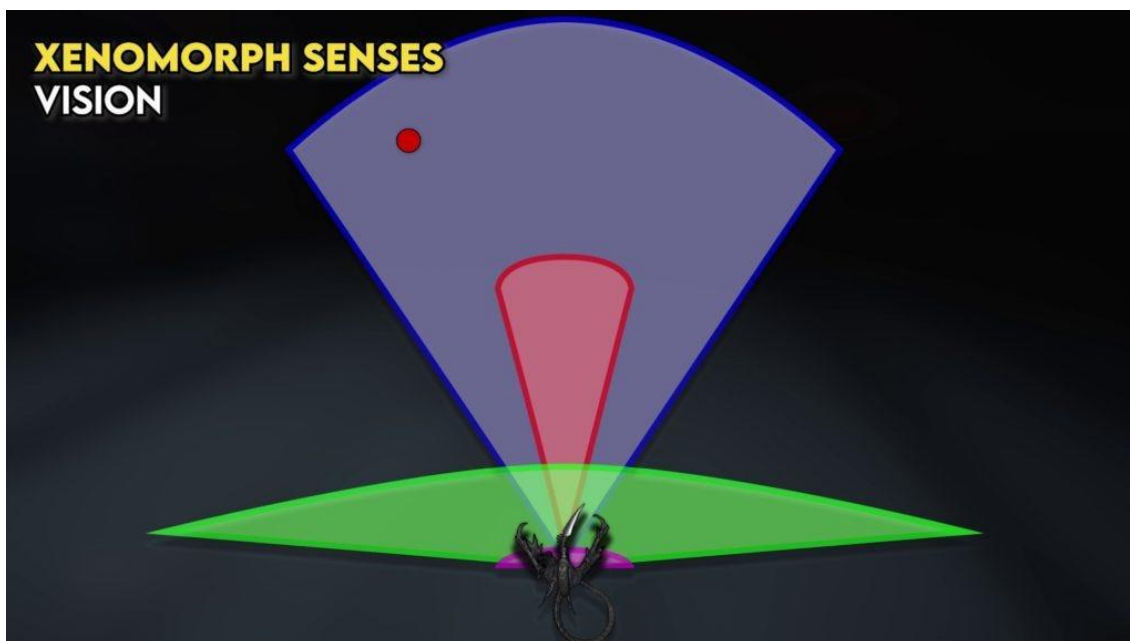
Pelin teköily tuntuu pelaajasta oppivan pelin edetessä ikään kuin se kehittäisi vastastrategioita. Tämä johtuu kahdesta eri asiasta. Käyttäytymispuun kaikki solmut eivät ole auki pelin alussa, vaan niistä osa avataan vasta myöhemmässä vaiheessa peliä. Teköily myös kerää pelaajan pelitavasta tietoa ja sen perusteella vaikuttaa Xenomorphin päätöksentekoon. Esimerkkinä näistä on ilmastointikanavien käyttö. Alien Isolationin alkupuolella pelaajalla on mahdollisuus usein paeta hirviötä piiloon esimerkiksi ilmastointikanaviin, joihin Alien ei lähtökohtaisesti kajoa. Mikäli pelaaja kuitenkin näin tekee toistuvasti, avautuu myöhemmin pelissä Xenomorphin käyttäytymispuussa solmu, joka mahdollistaa

pelaajan jahtaamisen ilmastointikanavissa ja niissä vaanimisen. Käyttäytymispuun kehitys luo illuusion pelaajalle siitä, että hän pelaa älykästä vastustajaa vastaan, joka oppii hänen pelitavastaan nostattaen pelin jännitystasoa huomattavasti, sillä mikään paikka ei tunnu enää täysin turvalliselta.

Xenomorphin näkötoiminto

Xenomorph käyttää visuaaliseen havaitsemiseen näkökartioita, jotka ulottuvat sen eteen eri muotoisina keiloina. Niiden sisällä olevat asiat luokitellaan nähdyiksi ennalta määritetyn ajan jälkeen. Näkökeilat ovat todella yleinen tapa toteuttaa tekoälylle näkösensori videopeleissä. Monissa tapauksissa pelaajalle myös indikoidaan audiovisuaalisesti, mikäli hänet on nähty, kuten Metal Gear Solid -pelissä ikoninen punainen huutomerkki vihollisen pään päällä yhdistettynä korkeaan huomioääneen. Alien: Isolationin Xenomorphilla on neljä eri näkökeilaa (kuva 15):

Alienilla on normaali näkökeila edessä olevien asioiden suhteellisen kaukaa havaitsemiseen. Pienempi ja lyhyempi fokus -näkökeila asioiden tarkempaa näkemistä ja tutkimista varten. Ääreisnäkökeila, joka on todella lyhyt, mutta äärimmäisen leveä ja toimii, kuten ihmisen ääreisnäkö. Lähinäkökeila, minkä tarkoitus on huomata heti kaikki Alienin edessä välittömässä läheisyydessä keilaan päätyvät objektit.



Kuva 15. Xenomorphin eri näkökeilat. Violetti kuvastaa lähinäköä, vihreä ääreisnäköä, punainen fokusnäkökeilaa ja sininen normaalia näkökeilaa (Revisiting the AI of Alien: Isolation 2020).

Näkökeilojen toimintaperiaatteisiin kuuluu, että kun pelaaja on keilan sisällä, eikä pelaajaan ole näköestettä, niin teoriassa Xenomorph on tällöin nähnyt pelaajan. Näkösensorit eivät kuitenkaan heti rekisteröi pelaajaa nähdyksi, vaan mitä kauemmin pelaaja on näkökeilassa, sen vahvemman signaalin Alien saa. Kun tämä signaalin arvo ylittää tietyn rajan, siitä lähetetään varmistus avaten käyttäytymispuussa solmuja, joiden mukaan se voi toimia. Jokaisella näkökeilalla on omat ominaisuutensa, millä vauhdilla pelaaja huomataan. Huomaamisen nopeuteen vaikuttaa näiden lisäksi ulkoisia tekijöitä, kuten ympäristön valaistus ja tilavuusobjektit esimerkiksi ilmankosteus tai savu. Pelin sisällä avaruusaluksella onkin terminaaleja, joista voidaan hetkellisesti deaktivoida sisäilman puhdistus, mikä puskee tilaan höyryä heikentäen Alienin näkökykyä. On kuitenkin huomioitava, että valaistuksen määrä ei vaikuta juurikaan Alienin näkökykyyn vaan enemmänkin pelin muiden hahmojen näkökykyyn.



Kuva 16. Kuvassa pelaajaan päälle hyökkää otsalampun valosta vihastunut Xenomorph (Alien: Isolation 2014).

Oleellinen näkökykyyn liittyvä lisäseikka on, että taskulampun valolla on tarkoituksen mukaisesti korostettu huomiokerroin Alienin näköaistille. Taskulampun valokeila herättää todella herkästi Xenomorphin mielenkiinnon (kuva 16) ja laukaisee käyttäytymispuussa valonlähteen tutkimisen tai jopa suoraan pelaajan jahtaamisen, joten sen käyttöä kannattaa välttää vihollisen läsnä ollessa.

Kuulo tärkeänä osana Alienin tekoälyä

Xenomorph pystyy kuulemaan valikoiman ääniä eri äänenvoimakkuuksilla. Jokaisella pelin äänellä on eri voimakkuusaste ja kantavuusalue, joiden perusteella tekoäly kiinnostuu niistä. Esimerkiksi aseella ampuminen herättää Alienin kiinnostuksen poikkeuksetta muiden äänien ylitse. Nämä äänet laukaisevat käyttäytymispuussa äänen lähteen tutkimisen, jolloin Xenomorph kävelee lähteen luokse ja alkaa tutkimaan ympäristöä etsien esimerkiksi pelaajaa. Mikäli ääni kantautui kaukaa, kävelyn sijaan Xenomorph juoksee paikalle. Vaikka

Alien olisi "backstage"-tilassa ilmastointikanavissa, kovat äänet saattavat laukaista Alienin "frontstage"-tilan.

Pelaajan liikkumisella on kolme eri ääniastetta kantaen pidemmälle riippuen liikkumistyylistä, kyykyssä, kävellen tai juosten. Äänet ovat suuressa osassa pelikokemusta, pelaajan täytyykin olla aina tietoinen omista valinnoistaan ja valmistautua piiloutumaan, mikäli hän päästää ääniä niin tahattomasti kuin tahallaan. On myös huomioitava, että pelaaja ei ole ainoa äänilähde, jota Alien kuuntelee, vaan se myös seuraa ja kiinnostuu muidenkin hahmojen päästämistä äänistä.



Kuva 17. Liikesensorin kirkkaan vihreä täplä indikoi havaitun liikkeen suuntaa pelaajan edessä (Alien: Isolation 2014).

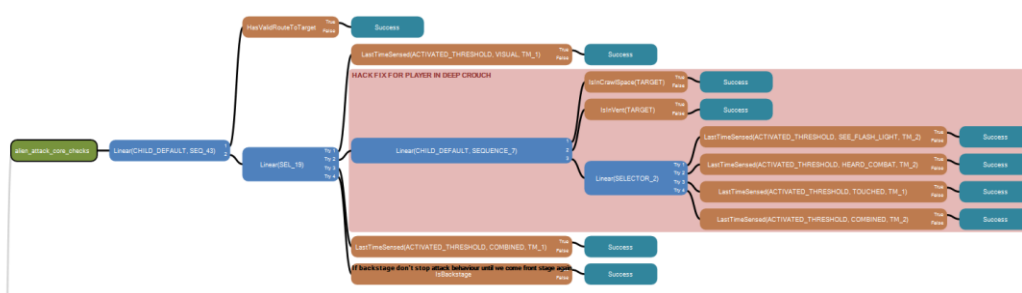
Pelaajan tukena pelissä toimii liikesensori, joka skannaa keilassa pienen alueen pelaajan edestä. Kuten kuvassa 17 voidaan havaita, skannauksen aikana liikkuvat asiat näkyvät hetken aikaa vihreinä kirkkaina pisteinä sensorin näytöllä. Näin pelaaja pystyy seuraamaan Xenomorphin ja muiden elävien olentojen liikkettä. Liikesensorin skannaus kuitenkin päästää suhteellisen kovan, kirkkaan äänen, jonka pelin viholliset pystyvät kuulemaan seinienkin läpi, joten pelaajan

on joissain tilanteissa harkittava, kuinka tärkeää hänen on yrittää saada selville vihollisen mahdollinen sijainti.

Muu käyttäytyminen

Xenomorph omaa muiden aistien lisäksi tuntoaistin, johon se reagoi aggressiivisesti ja armottomasti. Alienilla on useita metrejä pitkä häntä, jota se laahaa perässään ja mikäli pelaaja edes hipaisee häntä tai Xenomorphin kehoa, hän alustuu saman tien hyökkäykselle ja kuolee.

Tärkeässä osassa Alienin tekoälyn realismin suhteen toimii aistimuksien kuluminen. Kulumisella tarkoitetaan asioiden unohtamista ajan kuluessa. Alien unohtaa sen saamat aistimukset, kuten esimerkiksi huomiota herättävät äänet ja pelaajan näkemisen tietyn ajan kuluttua. Jokaisesta äänestä, visuaalisesta huomiosta tai muusta mielenkiintoisesta rakennetaan Alienille muistiin toimintamalli. Nämä toimintamallit järjestetään reaaliajassa mielenkiinnon ja tärkeyden mukaan järjestykseen. Tämä järjestys muuttuu ajan kanssa pääasiassa niiden tuoreuden ja kriittisyyden mukaan. Näin vihollisen toimintaa hallitaan ja sen päätöksentekoa priorisoidaan. Unohdusmekaniikka pitää huolen, ettei tekoäly jumitu tutkimaan yhtä asiaa kohtuuttoman pitkäksi aikaa vaan ylläpitää läsnäolollaan pelaajan jännitystilaa.

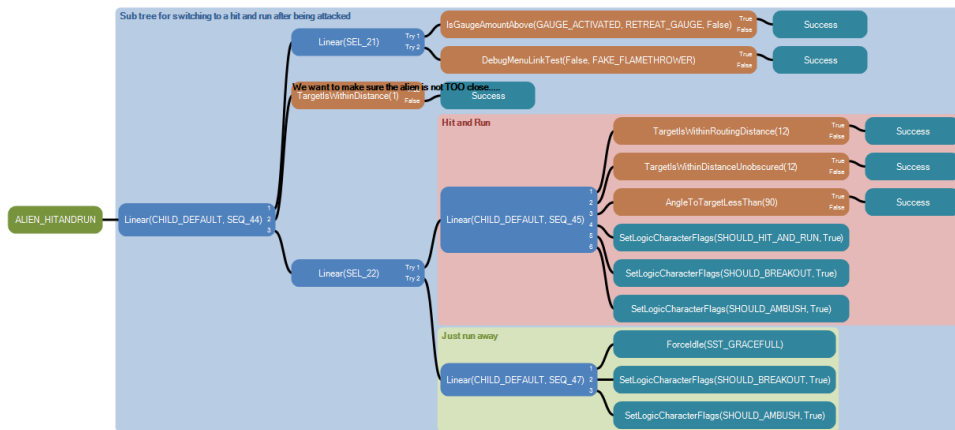


Kuva 18. Alienin käyttäytymispuun alien_attack_core_checks-solmu (OpenCAGE liite 1).

Kuvassa 18 nähdään Alienin hyökkäystä varten tehtävät eri tarkistukset. Xenomorph voi hyökätä pelaajaan vain, mikäli tarkat kriteerit täyttyvät. Sen täytyy

nähdä tai tuntea pelaaja, sillä täytyy olla esteetön reitti pelaajan luokse, se ei saa olla ilmastointikanavassa (ellei se ole yllätyshyökkäys) ja olla tarpeeksi lähellä hyökkäyksen suorittamiseksi. Peli tarjoaa jatkuvasti useita piiloja, joita pelaaja pystyy hyödyntämään liikkeessaan ympäristössä välttämällä tai yrittäessään karistaa vihollisen hänniltään. Xenomorphin hyökkäykseltä pakeneminen on äärimmäisen haastavaa, joten proaktiivinen varovaisuus ja hiiviskely on suositeltava etenemistapa.

Liekinheitin on pelaajan käytössä oleva erikoistyökalu, jonka avulla on mahdollista taistella Alienia vastaan niin kauan, kun siinä riittää polttoainetta. Liekinheitin avaa Xenomorphin käyttäytymispuussa erikoissolmuja, jotka muuttavat sen toimintaa merkittävästi. Xenomorphilla on olemassa pakenemismittari, joka täyttyy pelaajan liekittäessä sitä. Kun pakenemismittarin arvo on tarpeeksi suuri, Xenomorph poistuu tilanteesta juosten karkuun ja hypäten lähimmän kattoluukun kautta piiloon. Havaitessaan, että pelaajalla on käytössään liekinheitin, Alien ei enää suoraan hyökkää pelaajan kimppuun, vaan tunnustelee muutamaa metrin päästä tilannetta. Kuin saalistava susi se kiertää kehää pelaajan ympärillä. Ellei pelaaja liekitä sitä tarpeeksi pelottaakseen sen pois, Alien yrittää hyökätä syöksymällä pelaajan kimppuun kauempaa suorittamalla käyttäytymispuussa solmun "alien_hitandrun", joka nimensä mukaisesti tarkoittaa "lyö ja juokse". Kuvassa 19 on OpenCAGE:ssa tarkasteltuna Xenomorphin edellä mainittu solmu ja sen sisältämät lapsisolmut toiminnallisuuksineen.



Kuva 19. Alienin "alien_hitandrun"-solmu, jossa se tarkistaa, voiko se suorittaa hyökkäyksen pakeneamisen yhteydessä vai pakeneeko se vain paikalta (OpenCAGE liite 1).



Kuva 20. Xenomorph syöksymässä kattoluukusta alas (Alien: Isolation 2014).

Liekinheittimestä tietoinen tekoäly oppii käyttäytymispuussaan yllätyshyökkäyksen, jossa se lähestyy ensin pelaajaa edestäpäin, hyppää kattoluukkuun ikään

kuin paetakseen, mutta syökseekin ulos lähimmästä kattoluukusta kuvan 20 ta-
paan pelaajan takaa uuteen hyökkäykseen.

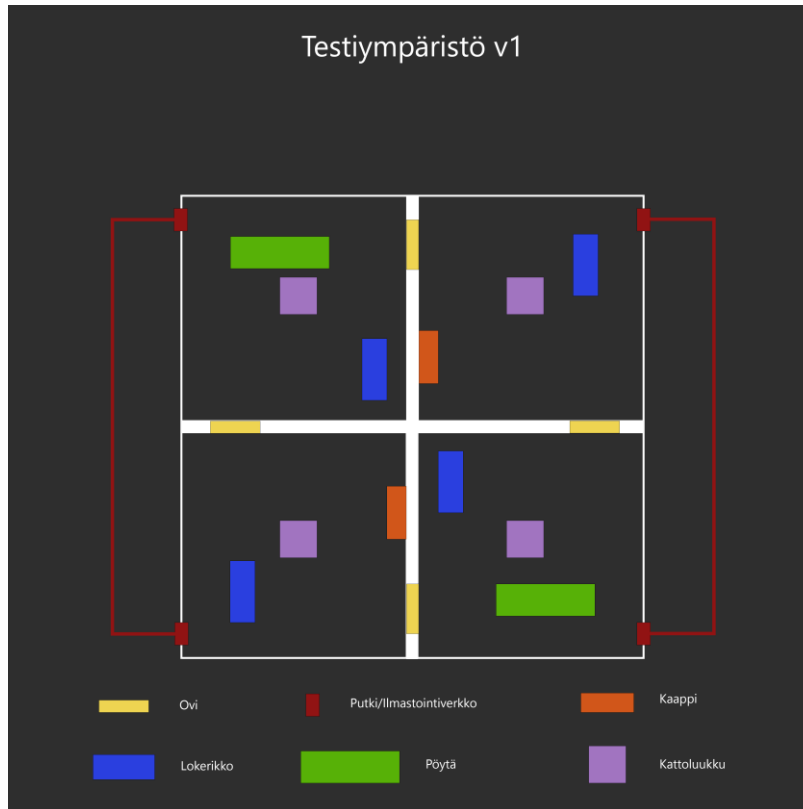
4 Simulaatioympäristön kehitys ja tekoälytoteutus

Tässä opinnäytetyön osiossa käydään läpi Alien: Isolation -pelin pohjalta luotua yksinkertaistettua tekoälykokonaisuutta, sen toimintamallia, simulaatioympäristön kehitystä ja ideoidaan sitä, kuinka tekoälyä voisi viedä pidemmälle sen alkuperäisestä esikuvasta. Tarkoituksena on luoda hyvin yksinkertainen, mutta pohjimmiltaan toimiva versio Alienin tekoälyn käyttäytymispuusta, simulaatioympäristö testaukseen, parametrisoida simulaatioympäristöä tiedonkeruuta ja tekoälyä varten. Tässä työssä ei käytetä aikaa simulaatioympäristön audiovisuaaliin kauhuelementteihin, vaan keskitytään käyttäytymispuun ja sitä tukevien toiminnallisuuksien kehittämiseen. Luotu tuotos löytyy liitteestä 3 annetun linkin takaa.

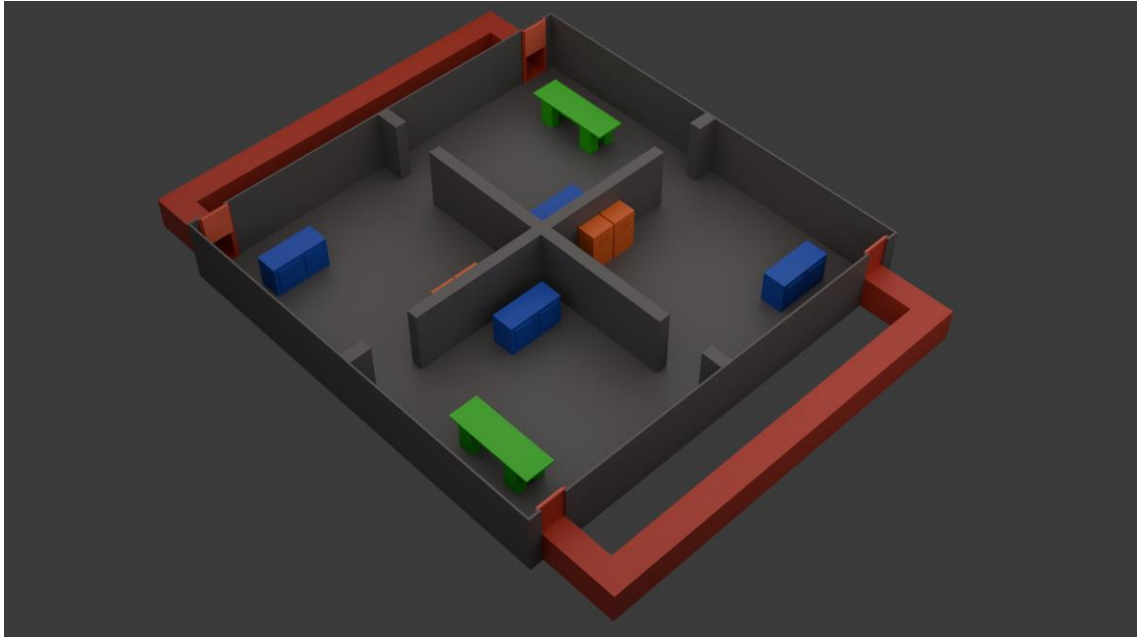
4.1 Simulaatioympäristön suunnittelu ja luominen

Simulaatioympäristön suunnittelussa kartoitetaan, minkälaisia ovat pelissä keskimääräisesti yleisimmin esiintyvät tilat ja mitä eri ominaisuuksia niillä on. Pelin läpipelauksessa opittiin, että keskimääräinen tila, jossa pelaaja liikkuu, koostuu useasta toisiinsa liitetystä huoneesta ja on suhteellisen ahdaskulkuinen, pakottaen näin pelaajan toimimaan vihollisen välittömässä läheisyydessä. Ympäristö sisältää roimasti pseudoturvallisia alueita, joita pelaaja voi käyttää hyväkseen piiloutuakseen ja edetäkseen kentässä. Tällaisia alueita ovat kaapit, joiden sisään voi piiloutua, pöytien aluset, ilmastointikanavat ja lukittavat ovet. Keskimääräisessä pelitilan katossa on useita avonaisia luokkuja, joita vihollinen käyttää yllätyshyökkäyksiin, ansoihin sekä vaihtamaan toimintaansa etu- ja takalan välillä.

Ympäristön suunnittelu alkoi kriteerien pohjalta Adobe Photoshop-ohjelmassa, jossa hahmotelma pelikentästä luotiin (kuva 21). Hahmotelman pohjalta tehtiin erittäin pelkistetty 3D-malli Blenderissä (kuva 22), joka vietiin käytössä olevaan Unity-pelimoottoriin jatkokehitystä ja toiminnallisuuden testausta varten.



Kuva 21. Simulaatioympäristön hahmotelman lopullinen versio.



Kuva 22. Simulaatioympäristöstä Blenderissä luotu 3D-malli. Lipastot ovat merkitty sinisellä, pöydät vihreällä, kaapit oranssilla ja ilmastointikanavat punaisella värillä.

Luodussa pelitilassa on neljä huonetta, joista toisiaan vastakkaisissa päädyissä olevat huoneet ovat toistensa peilikuvia. Jokaisessa huoneessa on lipasto ja joka toisessa on joko pöytä tai kaappi. Kaappien sisälle on mahdollista mennä avaamalla kaapin ovi ja kävelemällä sisään. Oven voi myös sulkea paremman näköturvan saavuttamiseksi. Ilmastointikanavat yhdistävät kaksi vierekkäistä huonetta toisiinsa. Niiden suuaukoilla on luukku, jonka pystyy sulkemaan hetkellisesti. Näin luotiin Alien: Isolation -pelin mukaan asetettujen kriteerien pohjalta pelitila, jossa luotua tekoälyä voidaan testata.

Pelaajan hahmolla on mahdollista liikkua seisaaltaan kävellen ja juosten tai kyykyssä hiipien. Pelaajan kyykkytoiminnolla voidaan piiloutua muun muassa pöytien alle, lipastojen taakse ja ilmastointikanaviin. Kyykyssä liikkuminen päästää myös vähemmän ääntä kuin seisaaltaan liikkuminen ja on siksi turvallisempaa.

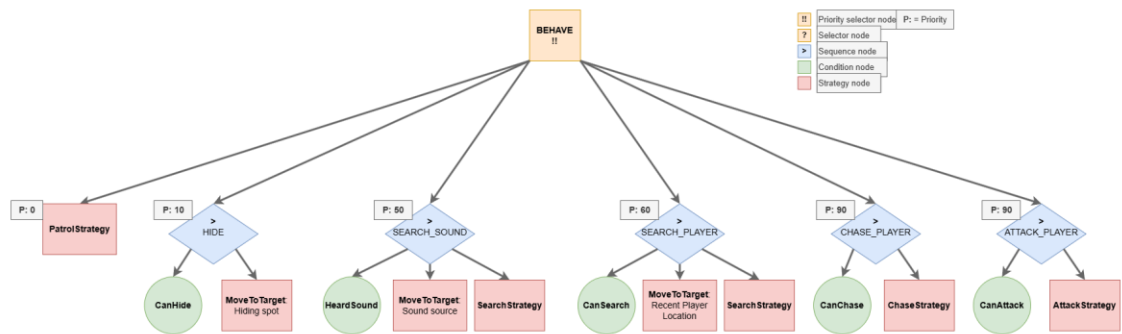
4.2 Luotu tekoälykokonaisuus ja sen käyttäytymispuu

Simulaatioympäristöön luotiin kaksi eri tekoälyä: kaikkietävä ohjaaja, joka valvoo pelitilaa ja pelin kulkua sekä käyttäytymispuulla varustettu vihollinen. Ohjaaja toimii hyvin yksinkertaisesti: se saa tietoa pelaajan ja vihollisen tarkasta sijainnista ja kummankin pelityylistä. Peliympäristö on varustettu laukaisimilla, jotka pitävät yllä tietoa siitä, missä huoneessa kukin liikkuu. Tämä tieto lähetetään ohjaajalle, joka voi halutessaan antaa viholliselle pelaajan paikkatietoa sitä auttaakseen jännitystason ylläpitämisessä. Luotu vihollisen tekoäly toimii yksinkertaisella versiolla Alien: Isolationin Xenomorphin tekoälystä.

Tekoälyn ympäristössä liikkumisen mahdollistaa Unityn NavMeshAgent-toiminto (Unity Scripting API: NavMeshAgent), jota manipuloidaan muun muassa käyttäytymispuulla. NavMeshAgent on Unity-pelimoottorin sisältämä objektin komponentti ja luokka, joka voidaan asettaa pelimoottorissa liikkuvalla hahmolla. Sen ansiosta hahmo pystyy navigoimaan ympäristössä käyttäen hyväksi ennalta laskettua navigaatioverkkoa (Unity Scripting API: NavMesh). NavMeshAgent pitää sisällään suuren kattauksen erilaisia toimintoja, joilla voidaan vaikuttaa pelihahmon liikkumiseen koodista käsin. Mainittakoon, että opinnäytetyön tekoälyn polunetsintää kehitettäessä päädyttiin NavMeshAgentin polunetsintäalgoritmiin oman A*-algoritmin (Patel, Amit) kehittämisen sijasta, sillä se käyttää jo itsessään samaa algoritmia ja on suoraan Unityn oma integraatio.

Käyttäytymispuun rakenne

Simulaatioympäristön tekoälyä varten määriteltiin käyttäytymispuun päätoiminnallisuus (kuva 23). Tärkeimmiksi toiminnallisuuteen liittyviksi ominaisuuksiksi todettiin seuraavat toiminnot: partiointi, etsintä, jahtaus, hyökkäys ja piiloutuminen pelikenttään. Toiminnallisuutta tukevat ominaisuudet kuten näkötoiminto, kuulo- ja tuntoaisti.



Kuva 23. Luodun simulaatioympäristön vihollisen käyttäytymispuu.

Toiminnallisuuden implementointi alkoi Unityssä käyttäytymispuun luonnilla. Kuten aiemmin opinnäytetyössä todettiin käyttäytymispuiden koostuvan solmuista (eng. Node), ja siksi implementaatio onkin järkevää aloittaa niistä. Käyttäytymispuun pohjalle luotiin "Node"-luokka (kuva 24), jonka erilaiset erikoissolmut ja käyttäytymispuuluokka itsessään perivät. "Node"-luokka pitää sisällään tietoa solmun tilanteesta (eng. Status), nimestä, prioriteetista, lapsisolmuista ja omaa funktiot AddChild(), Process() ja Reset(). Funktioilla pystytään lisäämään solmuille lapsisolmu, suorittamaan lapsisolmun prosessi ja resetoimaan koko solmun suoritus.

```

50 references
public class Node
{
    #region VARIABLES

    // Node status
    67 references
    public enum Status { Success, Failure, Running }

    public readonly string name;
    public readonly int priority;

    public readonly List<Node> children = new();
    protected int currentChild;

    #endregion

    // Priority determines the nodes priority in execution. Default = 0
    7 references
    public Node(string name = "Node", int priority = 0)
    {
        this.name = name;
        this.priority = priority;
    }

    // Adds child to node
    19 references
    public void AddChild(Node child) => children.Add(child);

    // Calling child and run its process
    17 references
    public virtual Status Process() => children[currentChild].Process();

    // Reset entire node
    16 references
    public virtual void Reset()
    {
        currentChild = 0;
        foreach (var child in children)
        {
            child.Reset();
        }
    }
}

```

Kuva 24. "Node"-luokka Unityn C#-koodilla toteutettuna.

Node-luokan periviä erikoissolmuja luotiin tätä projekia varten useampaa eri tyyppiä tarpeellisuuden mukaan. BehaviourTree-solmu toimii käyttäytymispuun juuri-solmuna. Sen erikoisominaisuuksiin kuuluvat testitoiminnot PrintTree() puun konsoliin printtaamista varten ja PrintNode() puun yksittäisen solmun konsoliin printtaamista varten. BehaviourTree sisältää myös IPolicy-rajapintaluokan, jonka avulla voidaan määrittää solmulle haluttu "Status", joka sen tulee palauttaa. IPolicyllä on mahdollista suorittaa jotain solmua ikuisesti, kuten esimerkiksi simulaatioympäristön vihollisella suoritetaan ikuisesti partiointia pohjimmaisena oletustoimintona.

Muita erikoissolmuja, joita luotiin käyttäytymispuuta varten ovat, sekvenssisolmu, valitsijasolmu, prioriteettivalitsijasolmu, satunnaisvalitsijasolmu, invertteri, onnistumiseen asti suorittava solmu (eng. until success), epäonnistumiseen asti suorittava solmu (eng. until fail) ja tietysti lehtisolmu. Lehtisolmulle luotiin muuttujaksi IStrategy-rajapintaluokan objekti (kuva 25). IStrategy-luokan objekti ohittaa Noden oman Process() funktion ja antaa mahdollisuuden ajaa omia IStrategy-strategioita eli toimintoja solmun Process()-funktiossa. Tämä on päätapa, millä käyttäytymispuu suorittaa aktuaalisia toiminnallisuuksia tekoälyllä.

Edellä mainituista erikoissolmuista tärkeimmät toteutetussa tekoälyssä ovat lehtisolmun lisäksi sekvenssisolmu, joka pyrkii suorittamaan kaikki lapsisolmunsa järjestyksessä, kunnes ne ovat palauttaneet onnistuneen statuksen ja prioriteettivalitsijasolmu, joka valitsee suoritettavakseen lapsistaan prioriteettimuuttujan mukaan suurimman prioriteettiaron omaavan solmun ja ajaa sen. Vihollisen tekoälyn käyttäytymispuun tukena toimii konditionaalilehtisolmu, näkö-, kuulo ja tuntoaisti.

```

// Behaviour to execute
15 references
public class Leaf : Node
{
    readonly IStrategy strategy;

    // Constructor
    13 references
    public Leaf(string name, IStrategy strategy, int priority = 0) : base(name, priority)
    {
        this.strategy = strategy;
    }

    // Execute the strategy
    9 references
    public override Status Process() => strategy.Process();

    // Reset strategy
    13 references
    public override void Reset() => strategy.Reset();
}

```

Kuva 25. Lehtisolmun -luokka Unityn C#-koodilla toteutettuna. IStrategy -rajapintaluokan objektilla voidaan luoda lehtisolmulle Process() funktioon suoritettavia toimintoja.

Konditionaalilehtisolmun tarkoituksena on toimia sekvenssisolmussa niin, että ensin tarkistetaan, onko mahdollista suorittaa seuraavia sekvenssisolmun

lapsisolmuja. Mikäli konditionaali palauttaa boolean arvon "true", voidaan siirtyä sekvenssisolmussa eteenpäin.

Simulaatioympäristöön luotu käyttäytymispuu toimii seuraavanlaisesti. Pohjalla on prioriteetilla varustettu valitsinsolmu. Se evaluoi aktiiviset lapsisolmunsa ja suorittaa niistä aina prioriteetissa korkeimmalla olevan solmun. Suunnitelluista toiminnallisuuksista kaikki kuusi toteutettiin: partiointi, jahtaus, äänen etsiminen, pelaajan etsiminen, hyökkäys ja piiloutuminen. Tekoäly ajaa partiointistrategiaa oletustoimintonaan prioriteetilla nolla, kunnes mikä tahansa muu solmu aktivoituu joko ohjaajan tai tekoälyn itsensä aistimuksen toimesta. Tekoäly kuitenkin partioi vain tietyn ajan kerrallaan piiloutuakseen hetkeksi ympäristöön ennen kuin taas jatkaa partiointiaan. Mikäli tekoäly kuulee äänen tai näkee pelaajan, mutta kadottaa tähän näköyhteyden, sen käyttäytymispuussa aktivoituu jompikumpi etsintä -solmuista. Käyttäytymispuun solmuista korkeimman prioriteetin omaavat hyökkääminen ja jahtaus. Jos tekoälyllä on näköyhteys pelaajaan, se käynnistää jahtausstrategian, minkä tarkoituksena on ottaa pelaaja kiinni. Kun se pääsee tarpeeksi lähelle pelaajaa, vihollisen on mahdollista suorittaa hyökkäys pelaajaa vastaan. Näillä yksinkertaisilla käyttäytymispuun toiminnallisuuksilla saadaan aikaan suhteellisen monipuolinen tekoäly.

Vihollisen tekoälyn aistit

Vihollisen tekoälylle toteutettiin tukiominaisuuksiksi näkö-, kuulo ja tuntoaisti. Tuntoaisti on hyvin yksinkertainen. Jokaisella pelin lasketulla ruudulla (eng. frame) tarkistetaan, onko pelaaja osunut fyysisesti vihollisen törmäystarkistukseen. Jos on, tuntoaisti on aktivoitu ja vihollinen saa pelaajan sijaintikoordinaatin suoraan tietoonsa, kääntyy katsomaan pelaajaa päin ja aloittaa jahtauksen tai mahdollisesti jopa hyökkäyksen, mikäli pelaaja on hyökkäysetäisyyden sisällä.

Näkötoiminto toteutettiin matkimalla Alien: Isolation -pelin Xenomorphin näkötoimintoa. Viholliselle toteutettiin normaali näkö. Se skannaa tietynkokoisella alueella eteensä lyhyin intervalein lisäten listaan kaikki objektit, jotka jäävät skannatun alueen törmäystarkistukseen ja ovat tietyllä ennalta määritetyllä

objektikerrostumalla (Unity Manuals: Layers). Tämän jälkeen näkötoiminto tarkistaa, onko näköyhteys objektiin suora vai onko havaittu objekti esimerkiksi seinän takana. Tämä on toteutettu ampumalla säde tekoälyn näköpisteestä havaittuun objektiin ja tarkistamalla, osuuko säde ensin johonkin muuhun. Säteenampumisfunktiolle voidaan myös määrittää, jättääkö se huomioimatta jonkin objektikerrostuman kokonaan. Lopuksi havaittujen objektien listasta tarkistetaan, sisältääkö lista pelaajan objektin; mikäli sisältää, tekoäly rekisteröi pelaajan havaituksi.

```
// Static class for calling MakeSound
1 reference
public static class Sounds
{
    1 reference
    public static void MakeSound(Sound sound)
    {
        Collider[] col = Physics.OverlapSphere(sound.pos, sound.range);

        for (int i = 0; i < col.Length; i++)
        {
            if (col[i].GetComponent(out IHear hearer))
            {
                // AIAAlertSystem
                hearer.RespondToSound(sound);
            }
        }
    }
}
```

Kuva 26. Staattisen "Sounds"-luokan toiminnallisuus Unityn C#-koodissa.

Tässä simulaatioympäristössä päädyttiin toteuttamaan kuuloaisti seuraavalla tavalla. Jokaisella objektilla, joka päästää ääntä, on MakeSound-luokka ja Unityn AudioSource-komponentti (Unity Scripting API: AudioSource). MakeSound-luokassa määritellään objektin AudioSourcen ääniefekti, äänen lähteen sijainti, sen kantavuus ja äänen parametri SoundType. SoundTypeä ei käytetä sen tarkemmin tässä työssä, mutta sen implementointi on jatkokehitystä varten. SoundType-parametrillä voidaan luetteloida äänen tyyppi esimerkiksi mielenkiintoiseksi tai vaaralliseksi ääneksi. Kun objekti, jolla on MakeSound-komponentti päästää äänen, se kutsuu staattista Sounds-luokan (kuva 26) funktiota MakeSound() antaen

paramereiksi funktiolle omat määritellyt arvonsa äänen kuvitellusta kantavuudesta, sijainnista ja äänen tyypistä. Sounds-luokan MakeSound() -funktio sitten tarkistaa, onko äänen lähtöpisteestä kantavuuden säteellä objektia, jolla on IHear-komponentti. Toteutetulla tekoälyllä on tässä tapauksessa IHear-komponentti. Mikäli IHear löytyy, sen sisäistä funktiota RespondToSound() kutsutaan, ja ääni on tällöin rekisteröity kuulluksi. Tekoälytoteutuksessa kyseistä kuulotoiminnallisuutta hyödynnetään pelaajan liikkumisen kanssa. Pelaaja päästää liikkuessaan ääntä tietyn intervallein. Kävellessä äänen kantama on pienempi, juostessa taas suurempi. Kyykyssä liikkuminen päästää vähiten ääntä ja on lähestulkoon äänetön tapa edetä.

5 Tekoälytoteutuksen arviointi

5.1 Lopullisen tekoälyn arviointi

Tarkoituksena oli luoda Alien: Isolation -pelin Xenomorphin tekoälytoteutuksesta yksinkertaistettu versio pienemmässä skaalassa Unity-pelimoottorin ympäristöön. Tekoälyn kehitys alkoi ensin tilakoneen luonnilla, jonka riittämättömyys kompleksin toiminnallisuuden toteuttamiseen paljastui myöhemmin työn edetessä. Tilakone on varsin hyvä vaihtoehto nimenomaan esimerkiksi ohjaajan toteuttamiseen tai yksinkertaisen tekoälyn luomiseen. Kuitenkin tässä tilanteessa oli lopulta helpompaa aloittaa tekoälyn kehitys alusta käyttäen käyttäytymispuumallia, joka osoittautui oikeaksi valinnaksi työn jatkokehityksenkin näkökulmasta ja on myös Creative Assemblyn valinta Alien: Isolation -pelin vihollisen tekoälyssä.

Aikapaineen alla tekoälyn toiminnallisuuteen ei käytetty määräänsä enempää resursseja, vaan työssä keskityttiin objektiivisesti tärkeämpään asiaan, käyttäytymispuun rakenteeseen ja sen jälkeen aistimuksien implementointiin. Toteutetun käyttäytymispuun rakenne on äärimmäisen toimiva ja hyvin skaalautuva. Siihen on suhteellisen vaivatonta lisätä erilaisia toiminnallisuuksia strategioiden tai

uusien solmujen muodossa. Tällaisia uusia strategioita käydään läpi jatkokehitysmahdollisuuksiin keskittyvässä osiossa myöhemmin. Käyttäytymispuun rakenteen luominen oli tekoälyn kehityksessä haastavin osa. Tilakoneen luominen verrattuna käyttäytymispuuhun on helppoa ja on siksi yleisempi ratkaisu pienempien kauhupelikehittäjien joukossa. Muutaman käyttäytymispuun iteraation jälkeen päädyttiin kuitenkin nykyiseen malliin. Käyttäytymispuun rakenteen ansiosta peliin on mahdollista luoda haluttaessa lisää yksilöllisiä tekoälyllä toimivia hahmoja ja sitä voidaan jopa käyttää esimerkiksi ohjaajan toiminnan määrittämiseen.

Käyttäytymispuuta tukevat aistimukset ovat osana onnistunutta kokonaisuutta. Näköaisti toimii halutulla tavalla ja sen skannauksen intervallia pystytään vaihtamaan, mikäli halutaan säästää prosessointitehossa. Näköaistin pelaajan huomaamisesta kuitenkin mainittakoon, että testiympäristöön ei toteutettu Xenomorphin versiosta identtistä kopiota. Simulaatioympäristössä tekoäly havaitsee pelaajan suoraan, mikäli pelaaja osuu näkökeilaan, kun taas Xenomorphilla pelaajan havaitseminen näkökeilassa on liukuluku, jonka vasta ylittäessään tietyn rajan merkitään pelaaja nähdyksi. Nykyinen toteutus on toimiva, muttei silti tuo samanlaista realismia kuin Alien: Isolation -pelin versio ja saattaa siksi tuntua hieman aggressiiviselta. Kuulo- ja tuntoaisti toimivat työssä halutulla tavalla, vaikkei esimerkiksi kuuloaistia käytetä sen täydellä potentiaalilla.

Käyttäytymispuun modulaarinen rakenne on tämän työn muun tutkimuksen lisäksi sen suurin onnistuminen ja siksi toteutuksen lopputulos on havaittu toimivaksi, vaikka tekoäly onkin käyttäytymiseltään hieman karkea.

5.2 Luodun tekoälyn tulevaisuus ja jatkokehitysmahdollisuudet

Kokonaisvaltaisen kauhusimulaatioympäristön luominen

Vaikkakin tämän opinnäytetyön tarkoituksena oli luoda käyttäytymispuulla varustettu tekoälykokonaisuus, on jatkokehityksessä paljon mahdollisuuksia. Lopullisena pidemmän ajan tavoitteena työn ulkopuolella on jatkaa luodun

tekoälyn kehitystä monessa eri muodossa. Ideana on yleisesti parantaa indie-kauhupeljä kokemuksena julkaisemalla luotu simulaatioympäristö Unityn verkkokauppaan kaikkien saataville. Lähitulevaisuuden tavoitteina on tehdä simulaatioympäristöstä kauhun luomisen keinojen avulla pelottava audiovisuaalinen kokonaisuus, pellillistä testiympäristöä pienten kognitiivisesti yksinkertaisten tehtävien avulla ja hioa tekoälyn käyttäytymispuhun luotuja toiminnallisuuksia realistisiksi vastaamaan enemmän Alien: Isolation -pelin kokonaisuutta. Ajatuksena on seurata pelaajan toimintatapaa ja muuttaa käyttäytymispuuta toimimaan pelaajan pelitapaa vastaan. Esimerkiksi, jos pelaajan huomataan pakenevan usein kaappeihin piiloon, tekoäly alkaa tarkistamaan useammin kaappien sisältä.

Tekoälyn aistimuksien realismia voidaan parantaa muutamalla eri tavalla. Kuloaistin ympärille on tulevaisuudessa tarkoitus rakentaa laajempi systeemi mahdollisesti dynaamisen äänimaailman muodossa. Ympäristöön asetetaan irtaimistoa, jonka kanssa pelaaja vuorovaikuttamaan. Aiemmin mainitun SoundTypen implementointi erilaisten esineiden ja pelaajan liikkumiseen äänimaailmaan on tarkoitus tuoda priorisointia tekoälyn äänten etsintä -solmuun. Myös aiemmin mainittu näköaistin huomaamismekaniikka toisi kokonaisuuteen uuden jännittävän ulottuvuuden.

Kehittäminen Alien: Isolation -pelin tekoälystä pidemmälle

Työn referenssipelin tekoäly on jo äärimmäisen monipuolinen ja mielekäs pelaaja vastaan. Jokaisella tekoälyllä on kuitenkin niiden potentiaalia rajoittavia tekijöitä, jotka yleensä liittyvät pelin teemaan, miljööhön tai itse tekoälyn ohjaamaan hahmoon. Alien: Isolation -peliä pelatessa on usein jännittäviä tilanteita vihollisen hiiviskellessä pelaajan silmien edestä hänen ollessaan piilossa pöydän alla tai kaapin sisällä. Xenomorphin yksi potentiaalia rajoittavista tekijöistä on sen fyysinen olomuoto. Se on avaruusolio, joka ei pysty kommunikoimaan pelaajan kanssa puhumalla. Simulaatioympäristön työhön luodulla tekoälyllä ei ole samaa ongelmaa, sillä se on vasta prototyyppi vaiheessa eikä sillä ole vielä kiveen hakattua fyysistä olomuotoa. Siispä yksi jännittävistä parannusehdotuksista

tekoälyn pelikokemuksen kauhun lisäämiseksi olisi vihollisen interaktiivinen vuorovaikutus puheen avulla pelaajan kanssa. Vihollistekoäly voisi esimerkiksi etsiessään pelaajaa uhkailla sitä varoittaen pelaajaa mahdollisesti tulevasta ja antaa samalla mahdollisuuden paniikinomaiselle toiminnalle, kuten piilopaikan vaihtamiselle. Esimerkkinä tällaisesta tilanteesta on pelaajan piiloutuminen pöydän alle. Pelaajan hetki sitten nähtyään vihollinen varoittaisi uhkaavasti mutisten ”kuulen sinut” tai kysyen ”oletko täällä alhaalla?” ennen kuin se polvistuisi katsomaan pöydän alle, jossa pelaaja on piilossa. Näin luodaan pelaajalle lisää jännitystä ja mahdollisuus altistaa itsensä vihollisen jahtaukselle pienen pakenemismahdollisuuden toivossa.

6 Yhteenveto

Insinööriyön tarkoituksena oli tutkia kauhun luomista peleissä, kauhupeleissä käytettyjä tekoälyratkaisuja ja selvittää, kuinka Alien: Isolation -pelin tekoäly toimii ja toteuttaa vastaavanlaisen tekoälyn pienemmällä skaalalla.

Työssä käytiin kattavasti läpi kauhun luomisen erilaiset audiovisuaaliset elementit ja niitä tukevat pelisuunnittelun keinot. Kauhun luominen peleissä edellyttää hienovaraista jännityksen tasapainoilua pelikokemuksessa. Pelaajien subjektiivisuus ja eri fobiat vaikuttavat suoraan siihen, miten eri ihmiset kokevat kauhua ja mikä on yksilön näkökulmasta pelottavaa.

Työssä opittiin minkälaisia eri tekoälytoteutuksia peleissä ja erityisesti kauhupeleissä esiintyy, mitkä ovat erilaisten toteutuksien etu- ja haittapuolet ja kuinka valita oikea tekoälytoteutus eri tarpeisiin. Insinööriyössä käytettiin historiallisesti arvostettuja kauhupelejä tekoälytoteutuksien esimerkkeinä ja pureuduttiin tarkemmin niiden toimintaan. Toiminnallisesti yksinkertaiset toteutukset tai määrällisesti suuret kokonaisuudet on järkevää toteuttaa äärellisillä tilakoneilla, kun taas käyttäytymispuut ovat helposti laajennettavan luonteensa johdosta parempi valinta monimutkaisiin tekoälyratkaisuihin. Insinööriyössä syvennyttiin yksityiskohtaisesti Alien: Isolation -pelin internetissä laajalti kehitettyyn tekoälytoteutukseen ja opittiin, kuinka hienostuneella rakenteella toimiva käyttäytymispuulla ja ohjaajalla varustettu tekoälykokonaisuus toimii.

Lopuksi luotiin referenssipeli Alien: Isolationin pohjalta testiympäristö ja käyttäytymispuulla toimiva tekoäly simulointia varten. Monimutkaisen ja arvaamattoman, mutta helposti implementoitavan kauhupelin tekoälyvastustajan luonti osoittautui mielekkääksi haasteeksi ja työssä onnistuttiin luomaan skaalattava käyttäytymispuurakenne Unity-pelimoottorin ympäristöön. Käyttäytymispuun pohjalta luodun tekoälyn toiminnallisuus on vielä luodussa versiossa karkea, mutta se on lupaava pohjustus jatkokehitysmahdollisuuksia varten.

Tulevaisuuden tavoitteena on kehittää luotu tekoäly pidemmälle ehdotettujen muutosten pohjalta ja julkaista tuotos Unityn verkkokauppaan kaikkien

saataville näin mahdollisesti parantaa indie-kehittäjien luomien kauhupelikokonaisten tekoälytoteutuksia.

Lähteet

Alien: Isolation. 2014. Creative Assembly & Feral Interactive.

Alan Wake 2. 2023. Remedy Entertainment.

Atkinson-Bonasio, Alice. Resident Evil and the Survival Horror Experience. Verkkoaineisto.

<https://www.academia.edu/273374/Resident_Evil_and_the_Survival_Horror_Experience>. Luettu 08.12.2023.

Booth, Michael. 2009. The AI Systems of Left 4 Dead. GDC 2009. Videoitu esitelmä. <<https://www.youtube.com/watch?v=PJNQI3K58CQ>>. Katsottu 05.08.2024.

Booth, Michael. 2009. The AI Systems of Left 4 Dead. GDC 2009. Verkkoaineisto. <https://steamcdn-a.akamaihd.net/apps/valve/2009/ai_systems_of_l4d_mike_booth.pdf>. Luettu 05.08.2024.

Butcher, Chris & Griesemer Jaime. 2002. Creating the Illusion of Intelligence: Where AI and Level Design Overlap in Halo's AI. GDC 2002. Videoitu esitelmä. <<https://www.youtube.com/watch?v=kda7rz5qFtl>>. Katsottu 28.05.2024.

Champanard, Alex J. Fuzzy State Machines. 2003. Teoksessa AI Game Development: Synthetic Creatures with Learning and Reactive Behaviors. Part VI: Emotions, Chapter 40.

Fight Or Flight Response. Verkkoaineisto. Psychology Tools.

<<https://www.psychologytools.com/resource/fight-or-flight-response/#:~:text=The%20fight%20or%20flight%20response,body%20to%20fight%20or%20flee.>>. Luettu 04.12.2023.

Freezer Fridge 3D Model. Mikserart.

<<https://www.renderhub.com/mikserart/freezer-fridge-low-poly-3d-model>>

Graja, Sarra; Lopes, Phil & Chanel, Guillaume. 2020. Impact of Visual and Sound Orchestration on Physiological Arousal and Tension in a Horror Game. Institute of Electrical and Electronics Engineers Transactions on Games PP(99):1-1. Verkkoaineisto.

<https://www.researchgate.net/publication/342577233_Impact_of_Visual_and_Sound_Orchestration_on_Physiological_Arousal_and_Tension_in_a_Horror_Game>. Luettu 06.12.2023.

Huttunen, Matti. 2018. Määräkohteinen pelko (fobia). Verkkoaineisto.

Lääkärikirja Duodecim. <<https://www.terveyskirjasto.fi/dlk00394>>. Luettu 05.12.2023.

Lankoski, Petri & Ekman, Inger. 2009. Hair-Raising Entertainment: Emotions, Sound, and Structure in Silent Hill 2 and Fatal Frame. Teoksessa Perron, Bernard (ed.). Horror Video Games: Essays on the Fusion of Fear and Play. Jefferson, N.C: McFarland & Co.

Left 4 Dead. 2008. Turtle Rock Studios.

Madigan, Jamie. 2015. The Psychology of Horror Games. Verkkoaineisto. The Psychology of Games.
<<https://www.psychologyofgames.com/2015/10/the-psychology-of-horror-games>>. Luettu 11.12.2023.

Mitchell, Jared. 2015. Creating Horror through Level Design: Tension, Jump Scares, and Chase Sequences. Verkkoaineisto. Game Developer.
<<https://www.gamedeveloper.com/design/creating-horror-through-level-design-tension-jump-scares-and-chase-sequences>>. Luettu 10.12.2023.

Niedenthal, Simon. 2009. Patterns of Obscurity: Gothic Setting and Light in Resident Evil 4 and Silent Hill 2. Teoksessa Perron, Bernard (ed.): Horror video games: essays on the fusion of fear and play. Jefferson, North-Carolina & London: McFarland & Company

Neurotic, stress-related and somatoform disorders (F40). 2019. Verkkoaineisto. ICD-10. World Health Organization.
<<https://icd.who.int/browse10/2019/en#/F40>>. Luettu 04.12.2023.

Nystrom, Robert. 2011. Game Programming Patterns. Kirja. Kappale 7: State, s. 87-104.

Orkin, Jeff. 2006. Three States a Plan: The A.I. of F.E.A.R. Verkkoaineisto. Monolith Productions / M.I.T. Media Lab, Cognitive Machines Group.
<<https://www.gamedevs.org/uploads/three-states-plan-ai-of-fear.pdf>>. Luettu 04.06.2024.

Outlast. 2013. Canada: Red Barrels.

Pac-Man. 1980. Namco.

Patel, Amit. Amit's A* Pages. Verkkoaineisto. University of Stanford.
<<https://theory.stanford.edu/~amitp/GameProgramming>>.

Phasmophobia. 2020. Kinetic Games.

Resident Evil Village. 2021. Capcom.

Revisiting the AI of Alien: Isolation | AI and Games #50. 2020. AI and Games. Video. <<https://youtu.be/P7d5lF6U0eQ>>.

Reyneri, L.M. An Introduction to Fuzzy State Automata. 1997. Verkkoaineisto. Politecnico di Torino. <https://www.researchgate.net/publication/220267791_An_Introduction_to_Fuzzy_State_Automata>. Luettu 05.08.2024.

Rodriguez, Jason. 2023. Alan Wake 2 guide to surviving the horrors within. Verkkoaineisto. <<https://store.epicgames.com/en-US/news/alan-wake-2-guide-to-surviving-the-horrors-within>>. Luettu 24.10.2024.

Silent Hill. 1999. Konami.

Simpson, Chris. 2014. Behavior Trees for AI: How They Work. Verkkoaineisto. <<https://outforafight.wordpress.com/2014/07/15/behaviour-behavior-trees-for-ai-dudes-part-1/>>. Luettu 02.08.2024.

Soutter, Alistair Raymond Bryce & Hitchens, Michael. 2016. The relationship between character identification and flow state within video games. Teoksessa Computers in Human Behavior. Volume 55, part B, s. 1030–1038.
State Machines vs Behavior Trees: designing a decision-making architecture for robotics. 2023. Verkkoaineisto. Polymath Robotics. <<https://www.polymathrobotics.com/blog/state-machines-vs-behavior-trees>>. Luettu 18.07.2024.

The Director AI of Left 4 Dead | AI and Games #07. 2014. AI and Games. Video. <<https://www.youtube.com/watch?v=WbHMxo11HcU>>. Katsottu 06.08.2024.

Tu, Yuan & Mozgovoy, Maxim. 2023. Interactive Horror Gaming: Integrating Facial Expression Recognition to Enhance Player Engagement and Immersion. Verkkoaineisto. University of Aizu, Japan. <<https://mmozgovoy.dev/papers/tm23a.pdf>>. Luettu 13.06.2024.

Unity Manuals: Layers. Unity. Dokumentaatio. <<https://docs.unity3d.com/Manual/Layers.html>>.

Unity Scripting API: AudioSource. Unity. Dokumentaatio. <<https://docs.unity3d.com/ScriptReference/AudioSource.html>>.

Unity Scripting API: NavMesh. Unity. Dokumentaatio. <<https://docs.unity3d.com/ScriptReference/AI.NavMesh.html>>.

Unity Scripting API: NavMeshAgent. Unity. Dokumentaatio. <<https://docs.unity3d.com/ScriptReference/AI.NavMeshAgent.html>>.

What Is The Difference Between DFA and NFA? 2024. Verkkoaineisto. Unstop. <<https://unstop.com/blog/difference-between-dfa-and-nfa>>. Luettu 11.06.2024.

Liite 1: OpenCAGE – Alien: Isolation

OpenCAGE on avoimen lähdekoodin työkalu, jolla käyttäjä pystyy näkemään, konfiguroimaan ja muokkaamaan Alien: Isolation -pelin tekoälyjen toimintaa graafisen käyttöliittymän kautta. OpenCAGE:lla on mahdollista tarkistella ja muokata esimerkiksi vihollisten tekoälyn käyttäytymispuita.

Linkki OpenCAGE työkalun sivulle:
<<https://github.com/MattFiler/OpenCAGE>>.

Liite 2: MoodMe SDK

MoodMe SDK on ohjelmistokehityspaketti, jonka avulla esimerkiksi Unity -pelimoottorissa voidaan reaaliajassa määritellä pelaajan tunnetiloja videokameran kautta saadusta materiaalista. Ennalta määritellyt tunnetilat, joita MoodMe -työkalu kategorisoi ovat ilo, yllättyneisyys, pelko, viha, kuvotus, suru ja neutraali ilme. Tunteita mitataan asteikolla 0-100.

Linkki yleiselle MoodMe:n sivustolle:
<<https://www.mood-me.com/unity-ai>>.

Linkki MoodMe:n Unity kaupan ohjelmistokehityspaketin ilmaisversioon:
<<https://assetstore.unity.com/packages/add-ons/machinelearning/moodme-emotions-barracuda-sdk-191930>>.

Liite 3: Luodun tekoälyn Unity -projekti

Tässä työssä luodun käyttäytymispuulla toimivan tekoälyn ja siihen liittyvän simulaatioympäristön Unity -projekti löytyy GitHub sivustolta. Projekti on toteutettu Unity:n Render Pipelinellä. Käytössä ollut Unity:n versio on 2022.3.11f1 ja kuuluu Unity:n pitkäaikaistukeen.

Linkki GitHub -sivustolle ladattuun projektiin:

<<https://github.com/Mahtisaurus/AI-TESTIYMPARISTO-V3>>.