

Timo Arna

Pelaajan mallintaminen ja adaptiiviset pelit

Tietotekniikan
kandidaatintutkielma
31. toukokuuta 2012

Jyväskylän yliopisto

Tietotekniikan laitos

Jyväskylä

Tekijä: Timo Arna

Yhteystiedot: timo.arna@jyu.fi

Työn nimi: Pelaajan mallintaminen ja adaptiiviset pelit

Title in English: Modelling of Player and Adaptive Games

Työ: Tietotekniikan kandidaatintutkielma

Sivumäärä: 30

Tiivistelmä: Tässä tutkielmassa perehdytään pelaajan mallintamisen periaatteisiin ja siinä käytettäviin tekniikoihin. Lisäksi tutkielmassa esitellään, kuinka mallinnuksella saatuja tuloksia voidaan hyödyntää adaptiivisissa digitaalisissa peleissä. Työmenetelmänä on käytetty kirjallisuuskartoitusta.

English abstract: Purpose of this study is to introduce some techniques and methods of player modelling. Also this study presents how the results of these methods can be used when developing adaptive digital games.

Avainsanat: peli, pelaaja, mallinnus, adaptiivinen

Keywords: game, player, modelling, adaptive

Sisältö

1	Johdanto	1
2	Pelaajan mallintaminen	2
2.1	Yleistä	2
2.2	Pelaajan mallinnus 3D-seikkailupelissä	3
2.3	Pelaajan mallinnus 2D-ammuntapelissä	5
2.4	Mallinnus pelaajan tunteiden avulla	7
3	Tulosten analysoinnissa käytetyt menetelmät	10
3.1	Klusterointi	10
3.2	Itseorganisoituvat kartat	12
4	Pelin adaptoituvuus	16
4.1	Yleistä	16
4.2	Adaptoituvuuden tarkoitus	17
4.3	Adaptoituvuuden kohteet	18
4.4	Adaptaatio evoluution avulla	20
5	Yhteenveto	25
	Lähteet	26

1 Johdanto

Peliteollisuus on jo kauan tutkinut pelejä ja pelaajia. Useat näistä tutkimuksista ovat kuitenkin keskittyneet tutkimaan pelaajaa pelin ulkopuolella. Käyttäjälähtöisessä pelinkehitysprosessissa pelaajien pelitapojen tuntemus nousee yhä merkittävämpään rooliin. Perinteisessä kehitysmallissa peliä on testattu prosessin eri kierrosten aikana ammattitestaajilla sekä pienellä otoksella kohderyhmään kuuluvia pelaajia [3]. Näin ollen pelistä on voinut muodostua näiden ryhmien pelityyleille sopiva. Tällöin on kuitenkin mahdollista, että ulkopuolelle jää suuri joukko potentiaalisia asiakkaita. Tämän vuoksi on tärkeää tietää, kuinka peliä pelataan, ja kuinka suureksi muodostuu ero haetun pelitavan ja todellisen pelaajakäyttäytymisen välillä [3].

Nykyasuuntaus on yhä vahvemmin suuntautumassa kohti epälineaarisia pelejä, joissa pelaajalla on useita valinnaisia tapoja edetä pelissä. Tällöin myös tarve ymmärtää eri pelitapoja kasvaa entisestään. Tutkimukset [3] [12] vahvistavat käsityksen, että jokaisella pelaajalla on oma henkilökohtainen tapansa pelata pelejä. Sillä, kuinka peli sopii pelaajan henkilökohtaiseen pelityyliin, on suuri merkitys. Tämä vaikuttaa siihen, kuinka hän kokee pelin ja kuinka suureksi pelinautinto muodostuu. Käyttäen hyväksi pelaajamallinnuksen mahdollisuuksia sekä lisäämällä pelien adaptiivisuutta peliteollisuus koettaa nyt houkutellessa uusia pelaajia yrittäen pitää samalla myös vanhat pelaajat tyytyväisinä [2] [3].

Tässä tutkimuksessa perehdytään muutamiin pelaajan mallinnuksessa käytettäviin menetelmiin sekä niiden periaatteisiin. Tutkimuksessa käytetään esimerkkeinä muutamia eri pelityyleihin ja genreihin kuuluvista pelistä tehtyjä tutkimuksia. Lisäksi tutkitaan mallinnuksen avulla saadun tiedon analysointia, siinä käytettäviä menetelmiä sekä saadun tiedon hyödyntämistä. Tutkimuksessa tutustutaan myös pelien adaptoituvuuteen, sen tarkoitukseen sekä kohteisiin. Lopuksi perehdytään esimerkin avulla erääseen tapaan toteuttaa adaptoituminen.

2 Pelaajan mallintaminen

2.1 Yleistä

Pelaajan mallintamisella tarkoitetaan yleensä pelaajan, pelin sisällä tapahtuvan, toiminnan kartoitusta. Mallinnuksella tutkitaan pelaajan suorittamia valintoja sekä käytöstä eri pelitilanteissa. Toisin sanoen pyritään luomaan kuva pelaajan henkilökohtaisesta pelityylistä. Pelaaja voidaan mallintaa myös pelaajan kokemien tunteiden avulla. Tällä pyritään muun muassa varmistamaan, että mahdollinen adaptaatio tapahtuisi oikealla hetkellä. Pelaajan tunteiden mallinnusta tarkastellaan lisää luvussa 2.4.

Tässä tutkimuksessa käytetyissä esimerkeissä mallinnus tapahtuu ensin mainitulla, pelaajan toimintaa pelin sisällä tutkivalla tavalla. Tällaisessa mallinnuksessa mitataan pelaajan toimintaa pelin eri toimintojen ja parametrien avulla. Erimerkiksi taistelupelissä voidaan mitata liikkumisnopeutta, osumatarkkuutta ja asevalintoja, mittauksia voidaan kerätä tietyissä pelitilanteissa tai ajan jaksoilla [12]. Analysoimalla näitä mitattuja suureita voidaan pelaaja sijoittaa ennalta määritettyihin pelaajatyyppeihin. Analysoinnissa käytetään muun muassa tiedonlouhintaa, tekoälyä, neuroverkkomalleja ja klusterointia.

Neuroverkkomalleista tyypillisimpiä ovat itseorganisoituvat kartat (engl. self-organizing map, SOM) [5] [7]. Itseorganisoituva kartta on neuroverkko, jonka avulla lähtödatan arvot voidaan muuntaa yksinkertaisemmiksi geometrisiksi suhteiksi. Nämä suhteet voidaan esittää kaksiulotteisen verkon muodossa. Klusteroinnilla tarkoitetaan samankaltaisiksi luokiteltujen datan osasten ryhmittelyä omiksi ryhmikseen siten, että päästään mahdollisimman suureen samankaltaisuuteen ryhmän sisällä, ja vastaavasti mahdollisimman suureen erilaisuuteen eri ryhmien välillä. Klusteroinnissa käytetään esimerkiksi dendrogrammeja [3] [12]. Itseorganisoituvia karttoja sekä klusterointia käsitellään tarkemmin luvussa 3.

Seuraavissa luvuissa tarkastellaan mallinnusta muutamissa eri peligenreihin kuuluvissa peleissä. Ensimmäisenä on kolmannessa persoonassa pelattava 3D-seikkailupeli. Tässä kolmas persoona tarkoittaa, että pelattavaa hahmoa katsotaan toisen henkilön näkökulmasta. Tällöin pelattava hahmo on kokonaisuudessaan pelaajan näkyvillä. Vastaavasti toisena esimerkkinä olevassa ensimmäisen persoonan pelissä pelaaja puolestaan katsoo pelimaailmaa pelihahmonsa silmin, jolloin hahmosta on

fyysisesti näkyvissä yleensä vain kädet.

2.2 Pelaajan mallinnus 3D-seikkailupelissä

Yksi suurimmista pelaajan mallinnusta koskevista tutkimuksista on Drachenin, Cannonan ja Yannakakisin tekemä tutkimus EIDOS-studioihin kuuluvan Cristal Dynamicsin Tomb Rider: UnderworldTMpelistä [3]. Tutkimus tehtiin vuonna 2008 ja siinä kerättiin tietoa yli 25000:sta tutkimushetkellä peliä pelanneesta pelaajasta. Näistä varsinaiseen tutkimukseen otettiin mukaan 1365 pelaajan pelitiedot. Edellytyksenä oli, että peli oli pelattu tutkimuksen aloitushetkellä läpi. Merkittäväksi tutkimuksen tekee myös se, että käytetty peli oli kaupallinen versio eikä tutkimusta varten erikseen modifioitu versio. Lisäksi tiedot kerättiin verkon välityksellä pelaajien pelatessa sitä normaalioloissa Microsoft Xbox LIVETM-pelipalvelun läpi eikä erikseen järjestetyissä laboratorio-oloissa.

Tomb Raider on seikkailupeli, jossa pelaaja ohjaa pelin sankaritarta Lara Croftia. Peliä pelataan kolmannessa persoonassa (luku 2.1). Kolmannen persoonan näkökulmasta pelattava peli vaatii kykyä hahmottaa pelihahmon etenemistä ja liikkumista 3D-maailmassa, välttämällä ympäristön vaaroja kuten ansoja tai putoamista. Lisäksi peli sisältää ongelmanratkaisutehtäviä sekä kohtaamisia eritasoisten vastustajien kanssa. [3]

Datan keräyksessä käytettiin EIDOS Metrics Suite -ohjelmistoa. Ohjelmisto on erityisesti kehitetty pelidatan keräystä varten. Pelissä on useita eri tasoja ja jokainen taso oli jaettu vielä erikseen sataan pienempään osaan. Data tallennettiin jokaiselta osa-alueelta erikseen. Tallennettavia dataominaisuuksia oli kaikkiaan kuusi: Neljä ensimmäistä liittyivät hahmon menehtymiseen sekä sen eri syihin, lisäksi mitattiin pelin läpi pelaamiseen käytetty kokonaisaika sekä pelaajan pyytämien vinkkien määrä ongelmien ratkaisutilanteissa. [3]

Mitattujen suureiden arvot vaihtelivat huomattavasti eri pelaajien välillä. Seuraavassa taulukossa 1 on esitetty mitatuista ominaisuuksista niiden suurimmat ja pienimmät arvot sekä keskiarvo KA.

Taulukko 1: Mitattujen suureiden arvoja

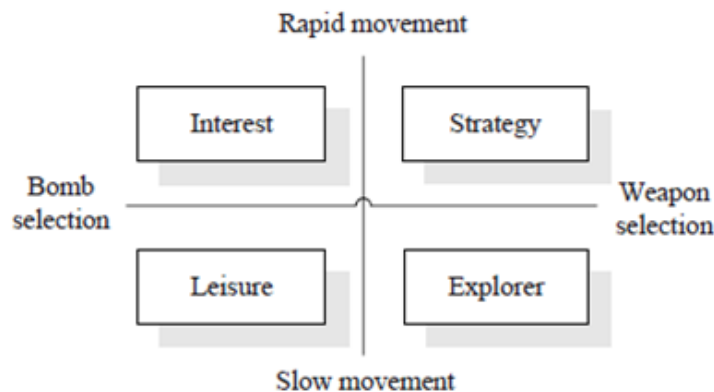
	pienin	KA	suurin
meneh.kokonaismäärä	16	140	458
meneh.vastustaja	6.32%	28.9%	60.86%
meneh.ympäristö	2.43%	13.7%	45.31%
meneh.putoaminen	27.19%	57.2%	83.33%
pelin läpäisy aika	2 h 51 min	9 h 11 min	28 h 58min
vinkkien käyttö	0	29.4	148

Kerätty data analysoitiin ja visualisoitiin luvuissa 3.1 ja 3.2 esitetyillä menetelmillä ja työkaluilla. Analyysiin tuloksena pelaajat voitiin sijoittaa neljään erilaiseen klusteriin, jokainen klusteri vastasi yhtä pelaajatyyppeä. Drachen, Canossa ja Yannakakis määrittivät pelaajat pelityylien mukaisesti seuraavankaltaisiin ryhmiin. Ensimmäistä ryhmää he kutsuvat *veteraaneiksi*. Tälle pelaajatyypille oli luonteenomaisista muun muassa nopea pelin läpäisy ja alhainen menehtymisien kokonaismäärä. Toista ryhmää kutsuttiin *ratkaisijoiksi*. Tämä ryhmä pelasi huomattavasti edellistä hitaammin ja keskittyen ratkomaan ongelmia. Tämän vuoksi heidän etenemisensä pelissä oli hidasta ja heidän pelihahmonsa menehtyi yleensä putoamalla. Kolmas ryhmä, *pasifistit*, oli suurin näistä neljästä ryhmästä. Vaikka tämän ryhmän sisällä esiintyi arvoissa suurtakin vaihtelua, he suorituivat pelistä kuitenkin keskimääräistä nopeammin. Neljättä ryhmää kutsuttiin *juoksijoksi*. Kuten ensimmäinenkin ryhmä he selvisivät pelistä erittäin nopeasti, mutta heidän menehtymisiensä määrä oli huomattavasti suurempi. Drachenin, Canossan ja Yannakakis mukaan analyysissä löytyneet neljä vaihtoehtoista tapaa pelata peliä osoittavat pelaajien olevan hyvin luovia pelityylinsä suhteen. Suhteellisen lineaarisissakin peleissä pelaajat käyttävät taitavasti hyväkseen pelin suunnittelun mahdollistamia tapoja edetä pelissä. Pelaajien toiminta ja valinnat voivat olla yllättäviäkin ja poiketa suurestikin pelin valmistajan tarkoittamasta. Juuri tällaisiin kysymyksiin Drachen, Canossa ja Yannakakis pyrkivät etsimään vastauksia tutkimuksellaan.

2.3 Pelaajan mallinnus 2D-ammuntapelissä

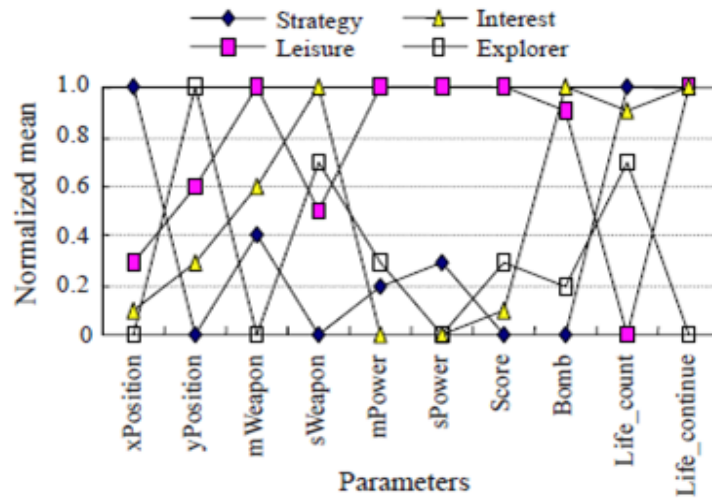
Toisena esimerkkinä pelaajan mallinnuksesta käytetään Wong *ym.* [12] tutkimusta 2D-ammuntapeleistä. Ammuntapeli ja ensimmäisen esimerkin seikkailupeli eroavat tyypiltään melkoisesti. Erilaisuudesta huolimatta pelaajan mallinnus tapahtuu niissä samojen periaatteiden mukaisesti. Suurimmaksi eroksi muodostuneekin mitattavien parametrien ja mahdollisten pelaajaluokkien määrittelyiden sovittaminen pelityypin ja peligenren mukaisiksi. Toinen merkittävä ero löytyy pelaajien luokittelusta. Ensimmäisen esimerkin Tomb Raider -pelissä pelaajaluokat muodostettiin vasta kerätyn aineiston perusteella. Wong *ym.* [12] käyttää lähestymistapaa, jossa pelaajat sijoitetaan analyysin perusteella ennalta määrättyihin pelaajaluokkiin.

Ammuntapelin pelaajaluokat muodostettiin pelaajan liikkumisnopeuden sekä käytettyjen aseiden tyypin ja käyttötavan mukaan. Luokittelu on esitetty kuvassa 1. *Interest*- sekä *Strategy*-luokan pelaajille oli yhteistä nopea liikkuminen pelissä, kun taas *Leisure*- ja *Explorer*-luokan pelaajat liikkuivat edellisiä hitaammin. *Strategy*- ja *Explorer*-pelaajat käyttivät muita luokkia monipuolisemmin pelin laajaa asevalikoimaa. [12]



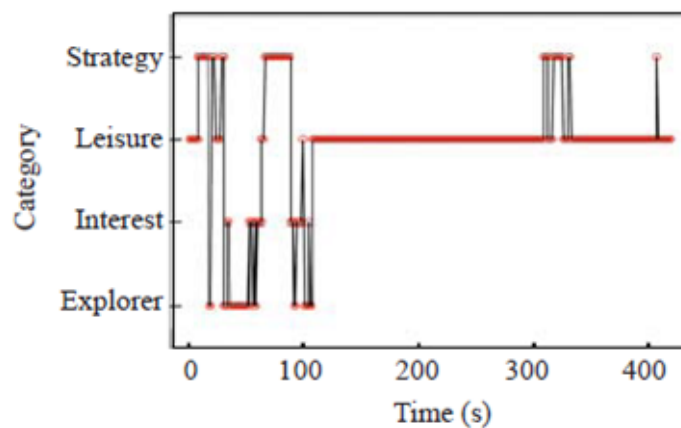
Kuva 1: Pelaajaluokat [12]

Pelaajan data muodostui yhdeksästä eri parametrasta, jotka kerättiin tietyin aikavälein. Kerätty data analysoitiin luvussa 3 esitetyn itseorganisoituvan kartan avulla. Kartta jakoi syötetyn tiedon perusteella pelaajat neljään ryhmään. Luokittelussa käytettiin apuna jokaiselle pelaajatyypille ja jokaiselle mitatulle parametrille lasketua normalisoitua keskiarvoa. [12] Nämä on esitetty kuvassa 2.



Kuva 2: Parametrien normalisoidut keskiarvot luokissa [12]

Wongin *ym.* [12] mallinnusprosessin tulos on esitetty kuvassa 3. Tulos osoittaa, että pelaajan luokka vaihtelee huomattavasti pelin alkaessa. Wongin *ym.* [12] mukaan tämä johtuu siitä, että pelaaja tutkii peliä ja etsii itselleen sopivaa pelitapaa. Pelaaja kokeilee useita vaihtoehtoisia tyylejä sekä menetelmiä ja ryhtyy sitten noudattamaan sitä, jonka tuntee sopivan itselleen parhaiten. Alun jälkeen pelaajan tyylin vaihtelu vähenee ja yksi tyyleistä muodostuu päätyylyiksi. Tyyli voi kuitenkin vaihtua ajoittain pelin tapahtumista riippuen. [12]



Kuva 3: Esimerkki pelaajatyylillä [12]

Tutkimuksensa perusteella Wong *ym.* [12] muokkasivat pelistä kaksi versiota, *strategy* ja *leisure*. Pelit annettiin koehenkilöiden pelattavaksi, ilman että pelaajilla

oli tietoa, mitä versiota pelistä he kulloinkin pelasivat. Viiden pelin jälkeen koehenkilöitä pyydettiin laittamaan pelit järjestykseen sen perusteella kuinka nautittava pelikokemus oli ollut. Wongin *ym.* [12] saamien tulosten perusteella yli 84% prosenttia peleistä sai paremman arvosanan nautittavuudessa, kun peli sopi pelaajan omaan pelityyliin.

2.4 Mallinnus pelaajan tunteiden avulla

Edellä olleissa esimerkeissä on käytetty pelistä mitattuja parametreja pelaajan mallintamisen apuna. Näillä parametreilla on pyritty tulkitsemaan sitä, mitä pelaaja kokee pelissä. Kiel ja Dix [4] kuitenkin osoittavat tutkimuksessaan, että joissain tapauksissa tämä ei johda haluttuun tulokseen vaan tulos voi olla päinvastainen. Adaptoitumisen (luku 4) tavoitteena on lisätä pelinautintoa. Yleisesti pelinautinnon tärkeimpänä tekijänä pidetään haastetta, eli peli ei ole pelaajan taitotasoon nähden liian vaikea eikä toisaalta myöskään liian helppo. Esimerkiksi mikäli pelaajaa joutuu toistuvasti suorittamaan samaa tehtävää siinä onnistumatta, voi peli jossain vaiheessa tulkita sen pelaajalle liian vaikeaksi. Tällöin voi peli muuntaa tehtävää siten että pelaaja suorituu siitä helpommin. On kuitenkin paljon pelaajia, jotka nauttivat suurista haasteista, lisäksi pelissä esimerkiksi loppuratkaisuiden tai taisteluiden oletetaan olevan hyvin haasteellisia. Mikäli peli muuntuu väärällä hetkellä tai liian paljon, voi pelaaja kokea tulleen petetyksi: "Tämä oli liian helppoa". [4]

Kielin ja Dixin [4] mukaan pelaajan mallinnuksessa tulee voida vastata seuraaviin kysymyksiin:

1. Miksi pelaajat pelaavat?
2. Kuinka he pelaavat?
3. Millon pelin tulee muuttua?

Kohdan kolme perusteella on tärkeää, että osataan tunnistaa, milloin muutos on tarpeellinen. Ratkaisuksi Kiel ja Dix [4] esittävät pelaajan tuntemusten, erityisesti turhautumisen tunteen tunnistamista. Pelaajan epäonnistuminen toistuvasti ei vielä tarkoita etteikö hän voisi nauttia pelistä, vasta kun pelaajassa alkaa esiintyä turhautumisen merkkejä pelin nautittavuus laskee. Turhautumisen merkkejä ovat muun muassa jännitys kasvoilla sekä pelivälineiden aggressiivinen kohtelu. Pelkäämään turhautumisen tunnistaminen ei riitä oikeanlaisen muutoksen aikaansaamiseksi, vaan on kyettävä tunnistamaan myös turhautumisen syy. Kiel ja Dix [4] jakavatkin turhautumisen syyt kahteen osa-alueeseen, *at-game-* ja *in-game-*turhautumiseen.

At-game-turhautumista voidaan kutsua kontrolliturhautumiseksi. Tämän tyyppinen turhautuminen johtuu pelivälineistä, kuten näppäimistöä, hiirestä tai muista ohjainlaitteista. Esimerkiksi aloitteleva pelaaja ei ole kykenevä suorittamaan useamman näppäimen painalluksia vaativaa komentoa oikeassa järjestyksessä tai oikea-aikaisesti saadakseen pelihahmonsa toimimaan tavalla, joka johtaisi onnistumiseen. Tähän vaikuttavat muun muassa komentojen monimutkaisuus, pelilaitteiden herkkyys sekä pelaajan omat taidot. [4]

In-game-turhautumista voidaan puolestaan kutsua tapahtumaturhautumiseksi. Tämän tyyppinen turhautuminen johtuu pelin tapahtumista. Esimerkiksi pelaaja ei tiedä, kuinka edetä pelissä tai joutuu käyttämään liikaa aikaa tarvittavien apuvälineiden etsimiseen tai niiden toistuvaan uudelleen hakemiseen jokaisen epäonnistuneen yrityksen jälkeen. Tähän vaikuttavat pelimaailman laajuus ja sen monimutkaisuus sekä pelaajalle mahdollisesti annettavien vihjeiden selkeys. [4]

Pelaajan tuntemusten mittaamiseen ja tunnistamiseen voidaan käyttää muun muassa sykettä, verenpainetta sekä hikoilua. Näiden ongelmaksi muodostuu kuitenkin se, että ne liittyvät sekä useisiin tunnetiloihin että erilaisiin fyysisiin tiloihin, joiden erottaminen toisistaan voi olla vaikeaa. Ratkaisuksi tarjotaan kameran käyttöä, tällä pystyttäisiin tulkitsemaan pelaajan kasvojen ilmeitä ja siten erottamaan eri tunnetiloja toisistaan. Kameran ottamista mukaan pelitilanteseen Kiel ja Dix [4] eivät pidä ongelmallisena, heidän mukaansa ongelmaksi muodostuu se, kuinka pelaajat saadaan pysymään kameran vaikutusalueella pelaamisen ajan.

Parhaaksi vaihtoehdoksi he ehdottavat turhautumisen tunteen mittaamista jo olemassa olevan pelivälineistön välityksellä. Turhautuminen voidaan heidän mukaansa tunnistaa seuraamalla seuraavia asioita:

- näppäinvirheiden määrä
- näppäinpaine
- ote ohjaimesta
- ohjaimen kulma, ravistus tai heilahdukset
- edistyminen pelissä.

Pelihahmon liikehännän sujuvuuden voimakas muuttuminen yhdessä virheellisten näppäinpainallusten kanssa voi kertoa kontrolliturhautumisesta, kun taas huono edistyminen pelissä yhdistettynä ohjaimen voimakkaisiin liikkeisiin tai kasvaneeseen näppäinpaineeseen voi ilmentää turhautumista pelin tapahtumiin [4].

Pelaajan tuntemusten mallintaminen avaa uusia mielenkiintoisia mahdollisuuksia myös adaptaatiolle. Tämän tyyppisellä mallinnuksella on myös mahdollista heijastaa pelaajan omia tunteita ja reaktioita pelihahmoon. Esimerkkinä Kiel ja Dix [4] käyttävät tilannetta, jossa pelaajan ohjaama pelihahmo hypähtää taaksepäin kun pelaaja itse säikähtää esimerkiksi äkillisesti ilmestyvää vastustajaa. Lisäksi Kielin ja Dix [4] vuonna 2004 julkaistun tutkimuksen jälkeen tekniikka on kehittynyt huomattavasti ja eri valmistajat ovat tuoneet markkinoille laitteita, jotka pystyvät tunnistamaan muun muassa pelaajan liikkeitä. Tällaisia laitteita ovat esimerkiksi Nintendo WiiTM(julk. 2006), Sony PlayStation 3 MoveTM(julk. 2010) sekä Microsoft KineticTM(julk. 2010). Tekniikka, jota voidaan hyödyntää tunnetilojen mittaamiseen esimerkiksi pelaajan kasvoilta, on jo siis olemassa.

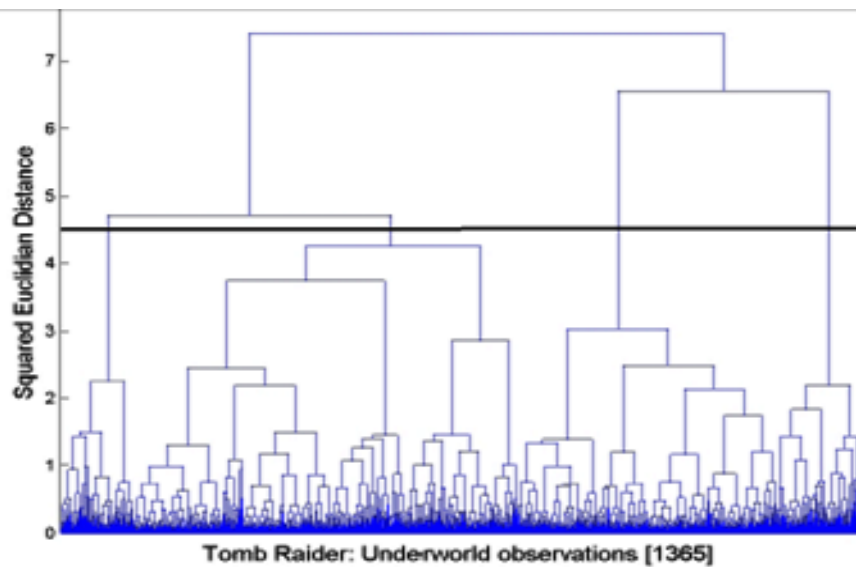
3 Tulosten analysoinnissa käytetyt menetelmät

3.1 Klusterointi

Klusterointi on tapa analysoida kerättyä dataa sen osasten samankaltaisuuden perusteella. Klusteronnissa aineisto jaotellaan pienemmiksi ryhmiksi siten että erot datassa ryhmän sisällä ovat mahdollisimman pienet ja vastaavasti erot ryhmien välillä mahdollisimman suuret. Jaottelu tapahtuu siihen valitun algoritmin tai vastaavan menetelmän avulla. Jaotteluun käytetty menetelmä mittaa datan osasten samankaltaisuutta tai erilaisuutta. Klusterointimenetelmiä on useita ja Tuononen [10] luokittelee ne seuraavasti: hierarkkiset ja osittavat menetelmät.

Hierarkkiset menetelmät Tuononen jakaa edelleen jakaviin ja yhdistäviin menetelmiin. Näillä menetelmillä klusteroinneille saadaan rakenne, joka on puumainen ja siten esitettävissä dendrogrammina (kuva 4). Jakavassa menetelmässä alkutilaksi valitaan koko analysoitava datajoukko. Tämän jälkeen alkujoukko jaetaan kahtia analysointiin käytettävän algoritmin avulla. Molemmat saadut klusterit jaetaan nyt vuorostaan kahtia ja tätä jatketaan kunnes saavutetaan haluttu klusterimäärä. Ääritilanteessa jokainen yksittäinen datan osanen voi siis muodostaa oman klusterinsa.

Yhdistävä menetelmä kulkee nimensä mukaisesti päinvastaiseen suuntaan kuin jakava menetelmä. Yhdistävässä menetelmässä lähtötilanteessa jokainen datan osanen on siis oma klusterinsa. Näitä klustereita yhdistetään analysointialgoritmin avulla suuremmiksi klustereiksi kunnes kuten edellä saavutetaan haluttu klusterimäärä tai päädytään lopputilanteeseen, jossa kaikki datan osaset kuuluvat samaan klusteriin. Yhdistäviin menetelmiin kuuluu myös luvussa 2.2 käytetty Wardin menetelmä [10].



Kuva 4: Esimerkki Wardin dendrogrammista [3]

Osittavat menetelmät Tuononen jakaa edelleen viiteen erityyppiseen menetelmään. Näitä ovat optimointi-, verkkoteoreettinen, tiheysperustainen, ristikkoperustainen ja malliperustainen menetelmä. Optimointimenetelmässä lopullinen klusterien määrä on ennalta tiedossa. Menetelmässä pyritään optimoimaan annettu funktio, joko minimoimalla tai maksimoimalla. Yleisesti funktiona käytetään niin sanottua keskineliövirhettä. Keskineliövirheelle käytetään kaavaa

$$MSE = \frac{1}{N} \sum_{i=1}^N d(X_i, C_{pi})^2 \quad (1)$$

missä N tarkoittaa datan osasten lukumäärää, X_i on datan i :s osanen, C_{pi} on sen klusterin keskipiste, johon X kuuluu, sekä $d(X_i, C_{pi})$, on i :nen osanen ja keskipisteen välinen euklidinen etäisyys. [10]

Myös luvussa 2.2 käsitellyssä tutkimuksessa käytetty K-means-algoritmi kuuluu tähän ryhmään. K-means-algoritmin periaate on seuraava:

- Ensiksi valitaan satunnaiset datan osaset, joista tulee klustereiden keskipisteitä.
- Toisessa vaiheessa muodostetaan klusterit siten että jokainen datan osanen kuuluu siihen klusteriin, jonka keskipiste on lähin.
- Kolmannessa vaiheessa muodostetaan klusterille uusi keskipiste laskemalla se siihen kuuluvien datan osasten keskiarvosta.

- Tätä toistetaan toisesta vaiheesta alkaen kunnes saavutetaan muuttumaton eli staattinen tila. [10]

Verkkoteoreettisissa menetelmissä analysoitavaan dataan voidaan käyttää pienimmän virittävää puun menetelmää. Datan osaset muodostavat puun solmukohtat ja solmut yhdistävien kaarien painona käytetään osasten etäisyyttä toisistaan. [10] Pienin virittävä puu muodostuu siis niistä kaarista, jotka yhdistävät kaikki solmut siten että kaarien paino on mahdollisimman pieni. [10]

Tiheysperustaiset menetelmät perustuvat nimensä mukaisesti datan osasten esiintymistiheyteen. Tutkitaan siis, millaisia tihentymiä datan osaset muodostavat analysoitavan aineiston sisällä ja jokainen tihentymä luokitellaan omaksi klusteriksi. [10] Tämän menetelmän eduksi katsotaan, että muodostuvien klustereiden koko ja muoto vaihtelevat enemmän kuin muilla menetelmillä tuotetuilla. [10]

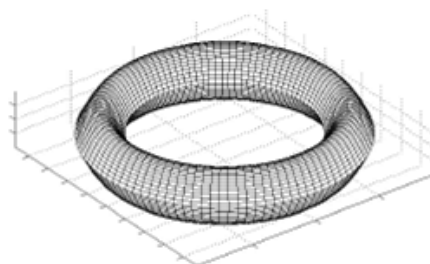
Ristikkopohjaisissa menetelmissä analysoitava data jaetaan ristikon muotoisella rakenteella ja datan analysointi tapahtuu tässä rakenteessa ja sen muodostamisessa nelöissä. [10]

Malliperustaisissa menetelmissä analysoitavaa dataa pyritään klusteroimaan jonkin mallin avulla. Yleensä mallina toimii neuroverkko. Myös luvussa 2.1 käsitellyssä esimerkkitutkimuksessa käytetyt itseorganisoituvat kartat [5] kuuluvat tähän menetelmäryhmään. [10]

3.2 Itseorganisoituvat kartat

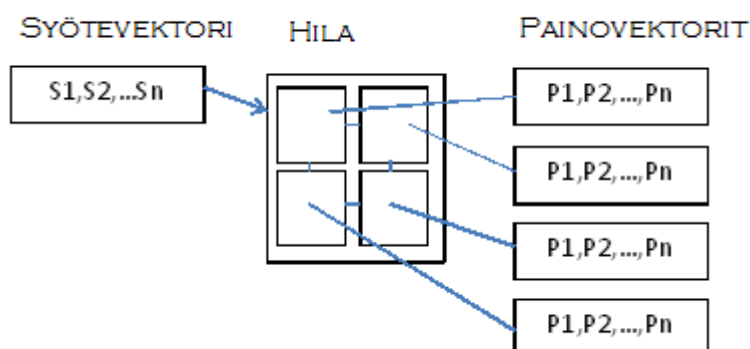
Toista tässä tutkielmassa käsiteltävää pelaajamallinnuksessa saatavan tiedon analysointimenetelmää kutsutaan itseorganisoituvaksi kartaksi tai kehittäjänsä suomalaisen Teuvo Kohosen mukaan Kohosen kartaksi. Itseorganisoituva kartta perustuu neuroverkon mallinnukseen ja ohjaamattomaan oppimiseen [5] [7] [8]. Kysymyksessä on siis keinotekoinen neuroverkko (engl. artificial neural networks, ANN).

Kartan rakenne on hilamainen ja se muodostuu neuroneista. Nämä neuronit muuttuvat prosessin aikana niihin syötettyyn dataan. Tällaisista hilaneuroneista käytetään usein nimitystä solmu. Käyttämällä renkaanmuotoista (toroidi) rakennetta, hilasta saadaan rajaton (kuva 5). Tällöin jokaisella solmulla on yhtä monta naapuria.



Kuva 5: Esimerkki toroidin muotoisesta hilarakenteesta [3]

Vierekkäiset neuronit ovat yhteydessä paitsi toisiinsa myös niin sanotuihin syöteneuroneihin. Jokaisella neuronilla on myös painovektori, joka vastaa alkiodien lukumäärältään syöteneuronien syötevektoria (kuva 6).



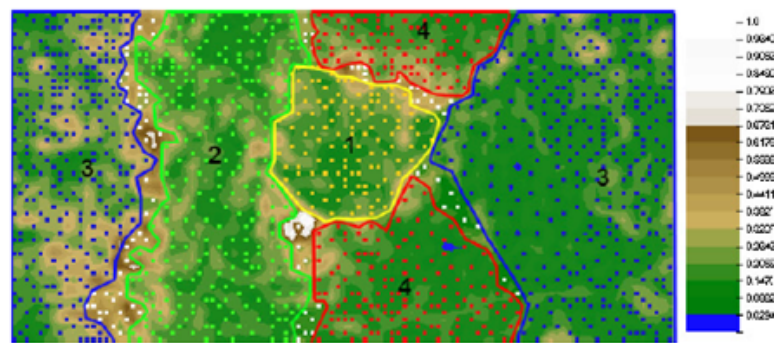
Kuva 6: Vektorit itseorganisoituvassa kartassa

Ennen käyttöä itseorganisoituva kartta alustetaan. Alustuksessa käytetään joko arvoltaan pieniä satunnaislukuja tai lukuja, jotka jo perustuvat tulevien syötevektorien arvoihin. [8]

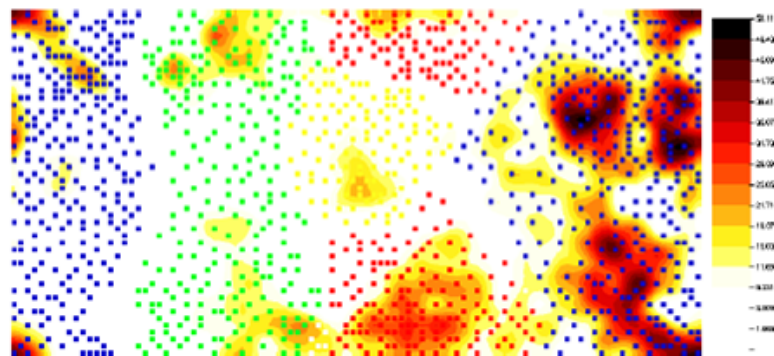
Oppimisvaiheessa käytettävässä opetusalgoritmissa [8] karttaan syötetään niin kutsuttuja syötevektoreita. Nämä vektorit sisältävät tutkittavat parametrit, eli tämän tutkielman esimerkeissä pelistä mitattuja numeerisia arvoja. Jokainen pelistä mitattu suure vastaa yhtä syötevektorin alkiota. Syötevektorit annetaan kartalle satunnaisessa järjestyksessä useita kertoja. Kun syötevektori syötetään karttaan, sille etsitään painovektori, jota se vastaa parhaiten. Tästä painovektorista käytetään myös nimitystä voittajasolmu [8] (engl. best matching unit, BMU). Tämän jälkeen suoritetaan päivitys. Tämä tapahtuu kaavoilla, jotka on tarkemmin esitetty Saarikosken työssä [8]. Päivitys koskee paitsi löydettyä solmua myös sen naapurustoa, toisin sanoen kaikkia naapurustoon kuuluvien neuronien painovektoreita päivitetään BMU:ta vastaavalla tavalla. Naapuruston koko pienenee opetusprosessin edessä. Lopputilassa kartta on järjestäytynyt vastaamaan syötetyn joukon ominaisuuksia. Mitä lähempänä kartalla olevat syötteet ovat toisiaan sitä paremmin ne vastaavat toisiaan. [8] [7]

Tässä tutkielmassa mainituissa tutkimuksissa käyetty ESOM (engl. evolving SOM, emergent SOM tai extended SOM) on itseorganisoituvan kartan eräs muunnos, jota käytetään erityisesti laajojen aineistojen analysointiin. Itseorganisoituva kartta voidaan esittää kaksiulotteisessa muodossa. Yksi esitys muodosta on niin sanottu U-matriisi (engl. U-matrix, unified distance matrix). Siinä neuronien väliset euklidiset etäisyydet estetään harmaasävyjen tai värien avulla muodostettuna kuvana. Tällaisessa kuvassa värisävyn tummus tai vaaleus ilmaisee neuronien suhteellista etäisyyttä. [8] [7]

Toinen esitysmuoto on P-matriisi (engl. P-matrix, Pareto density estimation, PDE). P-matriisi lasketaan käyttäen apuna datan tiheyden arviointia. Tämä on esitelty tarkemmin lähteessä [11].



(a) U-matrix



(b) P-matrix

Kuva 7: Esimerkki U- ja P-matriiseista [3]

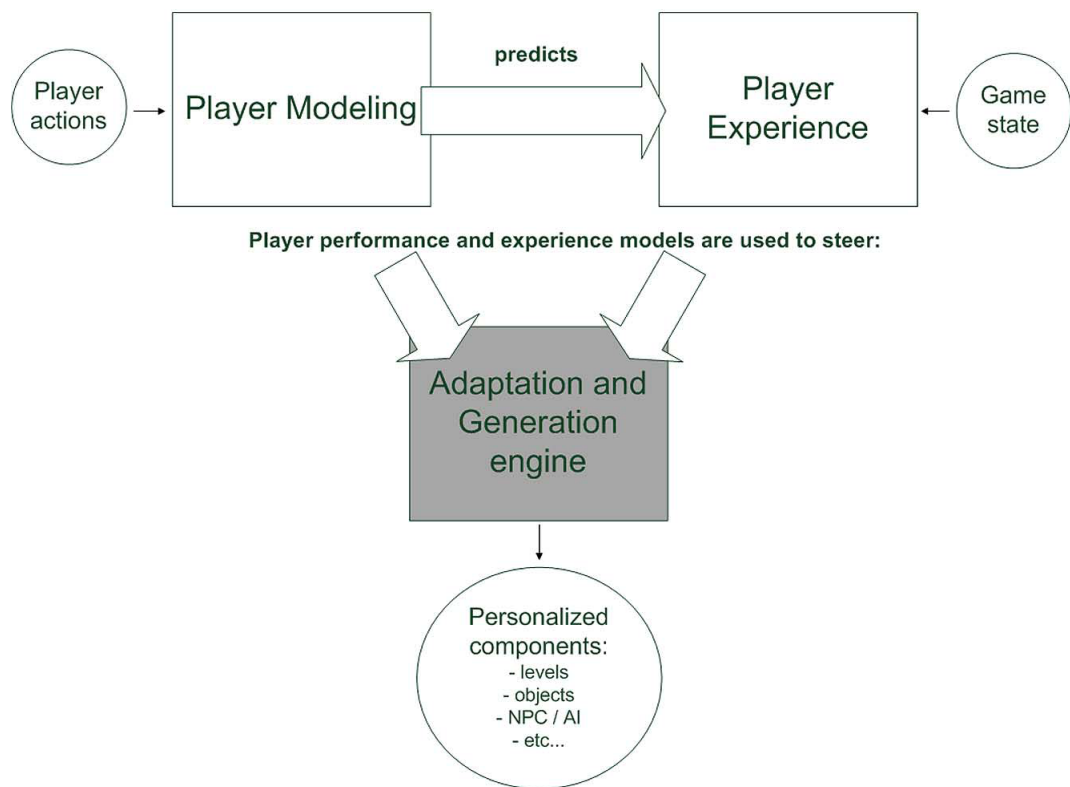
4 Pelin adaptoituvuus

4.1 Yleistä

Pelin adaptoituvuudella tarkoitetaan pelin muuntautumista pelin aikana yleensä pelaajan suosiman pelityylin tai taitotason mukaiseksi. Tarkoituksena on kasvattaa pelaajan kokemaa pelinautintoa. Opetuspeleissä ja simulaatioissa adaptoituvuudella voidaan lisätä opiskelumotivaatiota sekä parantaa oppimistuloksia muuttamalla opetusmenetelmiä ja tapoja kunkin pelaajan yksilöllisten kykyjen mukaiseksi. Lopesin ja Bidarran mukaan [6] useimmissa peleissä pelin rakenne ja sisältö, kuten säännöt, tarina, hahmot ja ympäristö on luotu kiinteäksi osaksi peliä. Pelaajalle ei näin ollen ole juurikaan annettu mahdollisuutta vaikuttaa niihin.

Pelaajan vaikutusmahdollisuudet on rajattu lähinnä yksinkertaisiin ennalta määrättyihin valintoihin, jotka suoritetaan ennen pelin alkua kuten esimerkiksi taitotason valita. Pelaajalla ei kuitenkaan välttämättä ole taitoa tai tietoa siitä, miten hänen tulisi luokitella itsensä kyseisessä pelissä [6]. Lopesin ja Bidarran tutkimuksen mukaan staattisesti käyttäytyvä peli, joko viihteellinen tai opettava, voi johtaa vastaavan tyylliseen pelaajakäyttäytymiseen. Mikäli peli ei uusia tapahtumia tai tehtäviä luodessaan tuo niihin oleellisesti mitään uutta, ei pelaajankaan tarvitse käyttää luovuuttaan niistä selvittääkseen. Tällöin pelaaja voi tyytyä toistamaan saman kaavan mukaista ratkaisua pelin tarjoamiin haasteisiin.

Kaavamainen pelaaminen voi puolestaan johtaa pelimotivaation sekä pelin kiinnostavuuden katoamiseen. Erityisesti juuri opetuspeleissä motivaatio ja kiinnostavuus olisivat erittäin tärkeitä, jotta voitaisiin saavuttaa parempia oppimistuloksia. Ratkaisumalliksi esitetään pelaajakeskeistä pelin adaptiivisuutta, toisin sanoen peli vastaa dynaamisesti pelaajan toimintaan pelissä. [6]



Kuva 8: Lopes ja Bidarra [6]: Pelin adaptoitumisen periaate

Tällaisen mallin toiminta Lopesin ja Bidarran mukaan on esitelty pääpiirteissään kuvassa 8. Pelaajan toimintaa mallinnetaan luvussa 2 esitetyllä tavalla, peli arvioi pelaajan menestystä ja sen jälkeen voi muuntaa pelin eri komponentteja vastaamaan pelaajan tyyliä ja taitoja. Pelissä on useita eri kohteita, joihin adaptatioprosessi voidaan kohdentaa. Adaptoituvien kohteiden tulee olla myös mitattavissa ja niihin kohdistuvien muutosten vaikutukset seurattavissa. Prosessin tulee tietää, mikä laukaisee muutoksen ja mihin se tulee kohdistaa. Seuraavissa luvuissa käydään läpi adaptaation tarkoitusta sekä mitä eri tyyppisiä adaptaatiokohteita on ja kuinka niitä voidaan muokata kulloisenkin pelaajan pelityyliin sopiviksi. [6]

4.2 Adaptoituvuuden tarkoitus

Adaptoitumisen tarkoituksena voidaan pitää pelaajakokemuksen parantamista. Viihteellisen pelin tarkoituksena on nimensä mukaisesti viihdyttää. Saavuttaakseen tavoitteensa pelin tulisi siis antaa käyttäjälleen miellyttävä pelikokemus. Miellyttävä pelikokemus on usean osatekijän summa, mutta yleisesti se tärkeimpänä mittarina

on pidetty haastavuutta. Haastavuuden vaativuutta tai tasoa mitataan usein tehtävään käytettyjen toistojen määrällä, suhteutettuna onnistumisten määrään. Mikäli pelaaja on joutunut toistamaan samaa tehtävää useita kertoja tuloksetta, on pelin tulkittu olevan liian vaikea ja päinvastoin. Haastavuudelle löytyy myös muita mitaustapoja, joista yksi mielenkiintoisimmista liittyy pelaajan fyysisten tuntemusten ja reaktioiden kuten turhautumisen mittaamiseen (luku 2.4). [6]

Vakavien pelien ja simulaatioiden päämäärät ovat jonkin verran erilaiset kun pelkästään viihdyttämään pyrkivillä peleillä. Näissä peleissä adaptiivisuuden tarkoituksena on selkeästi opetuksen tehostaminen ja opiskelumotivaation ylläpito. Lopesin ja Bidarran [6] mukaan tällöin on tärkeää osata kohdistaa adaptoituminen tiettyyn osa-alueeseen tai taitoon, sen sijaan että vaikutus kohdistettaisiin yleisemmin pelin haastavuuteen. Näin ollen pelaajan mallinnusprosessi tulee myös toteuttaa siten että pelaajasta voidaan luoda eräänlainen henkilökohtainen osaamisprofiili, joka muodostuu paitsi henkilön osamistasoista niin myös tavasta omaksua ja käsitellä uusia asioita. [6]

Eräs vakaville peleille ja simulaatioille tyypillinen piirre, jota ei esiinny viihteellisissä peleissä, on pelitapahtuman tulosten ja tehokkuuden arviointi. Arvioinnilla tarkoitetaan tässä joko pelaaja suullisesti tai kirjallisesti antamaa tai pelistä mitattua palautetta, jota käytetään opetuksen tehokkuuden mittaamiseen. Arviointi on pääsääntöisesti tapahtuman jälkeistä palautetta (engl. after action review, AAR). Arviointia adaptiivisuuden kannalta on alettu tutkia vasta viime aikoina, joten tutkimuksia aiheesta on vasta muutamia. Näissä tutkimuksissa pyritään muun muassa selvittämään, miten esimerkiksi tekoälyn avulla voitaisiin arvioida pelaajan oppimista ja kehittymistä ja edelleen käyttää saatua tietoa opetuksen tehostamiseksi. [6]

Kuten edellä on todettu adaptiivisuus viihteellisessä pelissä tarkoittaa hieman eri asiaa kun adaptiivisuus vakavissa pelissä tai simulaatiossa. Viihteellisissä pelissä pyritään kartoittamaan pelaajan pelityyliä ja siten vaikuttamaan positiivisesti koko pelikokemukseen, mallinnus ja adaptaatio kohdentuu siten pelaajan toiveiden ja pelimotivaation kartoitukseen. Vakavissa peleissä adaptiivisuuden kohteeksi valikoituu enemmänkin yksittäinen taito. Tällöin mallinnuksen tulee kohdentua pelaajan oppimiskyvyn ja tapojen selvittämiseen ja adaptaation puolestaan opetuksen tehokkuuden kannalta oleellisiin kohteisiin. [6]

4.3 Adaptoituvuuden kohteet

Peli voi adaptoitua usealla eri tavalla. Myös adaptaation kohteita on useita, lähes jokaisen pelin komponentin on mahdollista adaptoitua. Kohteet voidaan ryhmitellä

seuraavasti:

1. pelimekaniikka,
2. pelaajan hahmo,
3. ei-pelaajahahmot (engl. non-player character, NPC),
4. tarina,
5. pelimaailma,
6. pelin tehtävät ja skenaariot [2] [6].

Pelin mekaniikan kautta ohjataan kaikkea mitä hahmomme tekee pelissä, hyp-pää, puhuu tai ajaa autoa. Pelimekaniikalla voidaan siis muuttaa pelin toimintoja. Esimerkiksi autopelissä voidaan muuttaa mekanismin reagointia ohjausliikkeisiin, ja siten vaikuttaa ajamisen vaikeuteen.

Suoraan pelaajan hahmoon kohdistuvalla adaptaatiolla voidaan kasvattaa pelin realismin tuntua. Tämän tyyppinen adaptaatio lisää myös pelihahmon fyysisyyden tuntua. Charles *ym.* käyttävät tästä esimerkkinä tilannetta, jossa taistelupeli hidastaa pelihahmon liikkumista, mikäli tämä on haavoittunut. Hahmon realistisella reagoinnilla pelitapahtumiin on tärkeä merkitys sille, kuinka uppottava pelikokemus saadaan aikaan. [2]

Yleisin tapa vaikuttaa pelin kulkuun ovat tekoälyllä varustetut ei-pelaajahahmot. Pelin aikana kerätyn informaation avulla tekoäly pystyy muuttamaan ei-pelaajahahmojen toimintaa ja käytöstä. Näin voidaan nostaa tai laskea pelaajan vastuksen tasoa tai opastaa pelaajaa haluttuun suuntaan [2]. Ei-pelaajahahmon käytöksen muuttamista tekoälyn avulla käsitellään lisää luvussa 4.4, esimerkkinä käytetään evolutionääristä algoritmia.

Narratiivisessa eli tarinallisessa adaptaatiossa muutetaan pelin tarinaa, pelaajan suosiman pelitavan mukaiseksi. Tällöin tekoäly voi muuttaa tai vaihtaa pelissä olevia juonellisia osa-alueita pelaajan tekemien valintojen mukaisiksi. Esimerkiksi pelissä tärkeän esineen hankkimiseksi suoritettava sivutehtävä, voi vaihtua taistelusta ongelman ratkaisuksi tai päinvastoin riippuen pelaajan suosimasta pelityylistä [9].

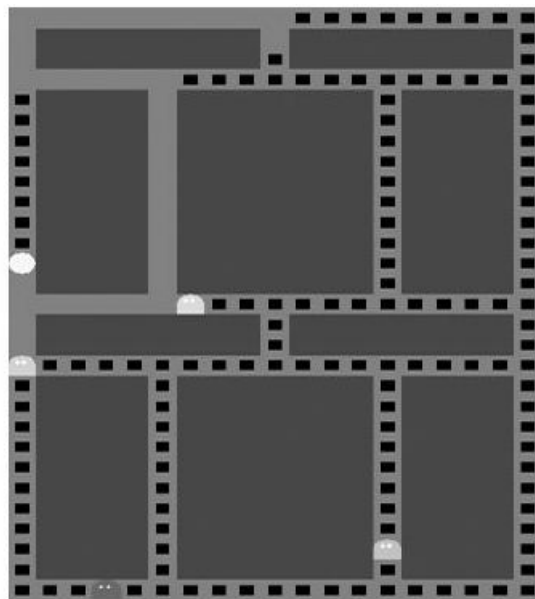
Myös kokonaisia peliympäristöjä ja maailmoja voidaan muokata adaptaation avulla. Yksinkertaisimmillaan tämä tarkoittaa tarvittavien esineiden sijoittelemista pelaajan löydettäväksi, mikäli jokin pelin kannalta olennainen esine puuttuu. Laajempien kokonaisuuksien kuten tehtävien tai kokonaisten pelitasojen (engl. level) muuntaminen on haastavampaa. Tämä onnistuu vielä suhteellisen helposti esi-

merkiksi yksinkertaisissa ASCII-pohjaisissa seikkailupeleissä, mutta mitä monimutkaisemmaksi ja yksityiskohtaisemmaksi peli tulee sitä vaikeammaksi muuntuminen tulee. [1] Skenaariot ja tehtävät ovat usein laajoja kokonaisuuksia, joiden tapahtumilla on usein selkeä syy-seuraussuhde. Tällaisten pelissä pitkälle eteenpäin vaikuttavien asioiden toteutus adaptoituvina tekee pelistä entistäkin monimutkaisemman, mikä puolestaan kasvattaa laitteistolle asetettuja vaatimuksia. Suurienkin pelikokonaisuuksien adaptoituminen on jo mahdollista 2D-ympäristössä, ja suunnittelutyökalujen sekä menetelmien parantuessa yhä laajemmat kokonaisuudet mahdollistuvat myös 3D-maailmassa [6].

Adaptaatiot voidaan toteuttaa joko pelin ollessa käynnissä (engl. online) tai ennen varsinaisen pelin käynnistystä (engl. off-line). Pienemmät ja välitöntä reaktiota pelaajan toimiin vaativat adaptaatiot toteutetaan pelin aikana. Suuremmat esimerkiksi koko pelimaailmaa koskevat muutokset on tehokkainta toteuttaa jo ennen pelin aloittamista. [6]

4.4 Adaptaatio evoluution avulla

Tässä luvussa tarkastellaan esimerkin valossa, kuinka kehittyvä vastustaja vaikuttaa pelinautintoon. Yannakakis ja Hallamin tutkimuksessa [13] käytettiin melko yksinkertaisesta ja hyvin tunnetusta Namcon PacMan-pelistä tutkimusta varten muokattua versiota (kuva 9).



Kuva 9: Yannakakis ja Hallam [13]: PacMan peli

Pelin idea on hyvin yksikertainen, pelaaja ohjaa *PacMan*-hahmoa yrittäen syödä kaikki sokkeloon asetetut nappulat, väistellen samalla neljää vastustajaa, *aaveita*. Peli päättyy kun kaikki nappulat on syöty tai joku *aaveista* onnistuu saamaan *PacManin* kiinni. Saadakseen nopeamman *PacManin* kiinni *aaveilta* vaadittiin yhteistyötä, koska pelissä ei ollut umpikujia. Alkuperäisessä *PacManissa* oli myös toiminto, jossa erityisen voimanappulan syötyään *PacMan* saattoi tuhota *aaveita* tietyn ajan. Tämä toiminto kuitenkin poistettiin kun peliä yksinkertaistettiin tutkimusta varten. [13]

Ensimmäiseksi Yannakakis ja Hallam [13] määrittivät sen mikä tekisi pelistä mielenkiintoisen. Määrittelyssään he päätyivät tulokseen, jonka mukaan pelin tuli täyttää seuraavat kolme kohtaa:

1. Vaikeustason tuli vastata pelaajan tasoa, eli peli ei saa olla liian helppo eikä myöskään liian vaikea. *PacManin* ja *aaveiden* voittojen tulee olla oikeassa suhteessa toisiinsa. Jatkovaa voittamista tai jatkuvaa häviämistä ei koeta kiinnostavana.
2. *Aaveiden* toiminta ei saa olla kaavamaisista vaan sen tulee vaihdella pelistä toiseen. Pelaajan ei pidä pystyä arvioimaan vastustajien käytöstä aiempien pelien perusteella.
3. *Aaveiden* tulee antaa vaikutelma, että ne aktiivisesti jahtaavat *PacMania*, eikä vain vaikutelmaa passiivisesta seurailusta.

Tutkimusta varten luotiin *PacManista* kolme eri tyyppiä kuvaamaan erilaisia pelitapoja. Ensimmäinen oli viereisten ruutujen painoarvoihin perustuva (engl. cost-based) *PacMan*. Ruutujen painoarvot määriteltiin seuraavasti:

- ruudun, jossa on nappula, painoarvo on 0
- tyhjän ruudun painoarvo on 10
- ruudun, jonka viereisessä ruudussa on *aave*, painoarvo on 50
- ruudun, jossa *aave*, painoarvo on 100.

Tämä *PacMan*-tyyppi pyrki löytämään painoarvoltaan pienimmän mahdollisen reitin. Mikäli ruutujen painoarvot ovat yhtä suuret, valinta suoritettiin sattuman avulla. [13]

Toista *PacMan*-tyyppiä kutsuttiin ohjeperustaiseksi (engl. rule-based) *PacManiksi*. Mikäli kaikki naapuriruudut osoittautuivat tyhjiksi, reitin valinta suoritettiin sen

perusteella, mistä suunnasta löytyi lähin nappula, muutoin valita tehtiin painoarvojen perusteella.

Kolmas, edistynyt (engl. advanced) *PacMan*, tarkasti ensin kaikki näkyvät suunnat ja mikäli jossain suunnassa oli *aave*, tuohon suuntaan liikkumisen todennäköisyys oli verrannollinen *aaveen* etäisyyteen. Muutoin valinta suoritettiin ohjeperustaisesti. [13]

Myös *aaveista* luotiin kolme erilaista tyyppiä: satunnaisesti liikkuvat (engl. random), joille minkä tahansa suunnan valitseminen oli yhtä todennäköistä, seuraajat (engl. followers), jotka pyrkivät pitämään etäisyyden *PacManiin* mahdollisimman lyhyenä ja lähes optimaaliset (engl. near-optimal), joiden algoritmi pyrki laskemaan optimaalisinta reittiä kohti *PacMania* samalla pysyen mahdollisimman etäällä muista *aaveista*. [13]

Yannakakis ja Hallam [13] kehittivät kaavan (kaava 2) *PacMan*-pelin kiinnostavuudelle. Kaava muotoiltiin muuntamalla luvun 4.4 alussa mainitut pelin kiinnostavuuden kriteerit matemaattiseen muotoon. Kaavassa

$$I = \frac{\gamma T + \delta S + \epsilon E\{H_n\}}{\gamma + \delta + \epsilon} \quad (2)$$

I on laskennan tuloksena saatava kiinnostavuusarvo. T on tässä vaikeusaste, jonka laskenta perustuu *PacManin* voittamiseen tarvittavien askelten määrän minimi-, keski- ja maksimiarvojen välisiin suhteisiin N määrässä pelejä. S on vaihteluarvo, joka lasketaan lähes vastaavalla tavalla mutta arvona käytetään askelten vaihtelua. $E\{H_n\}$ on puolestaan keskiarvo, jonka laskenta perustuu *aaveiden* vierailuihin tietyssä ruudussa ja vierailtujen ruutujen kokonaismäärään. Kaavan γ , δ ja ϵ ovat painoarvoja. Näiden eri parametrien tarkempiin laskukaavoihin voi tutustua lähteessä [13].

Tarkoituksena oli tutkia, kuinka muuttuva, pelaajan toimiin reagoiva, vastustaja vaikuttaa pelin kiinnostavuuteen. Tässä käytettiin apuna evolutionääristä algoritmia, jonka oppimisprosessi perustuu neuroverkon (luku 3.2) solmujen välisiin painoarvoihin. [13]

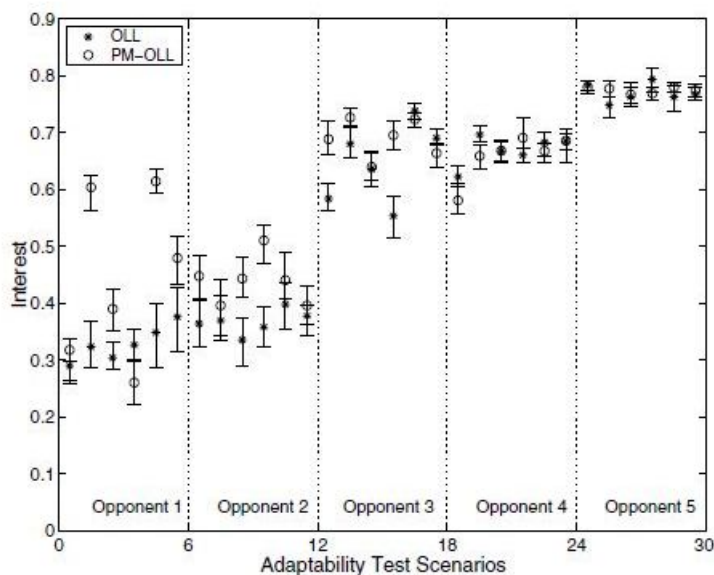
Oppimistapahtuma (engl. on-line learning, OLL) tapahtui siis evoluutioprosessin avulla. Ensimmäisessä vaiheessa *aaveista* luotiin neljän kloonin ryhmä, joka laitettiin pelaamaan valittua *PacMan* tyyppiä vastaan. Sen jälkeen tehtiin seuraavaa:

- Jokainen *aave* arvioitiin tietyin välein ja jokainen, joka oli liikkunut lähemmäs *PacMania*, nostettiin arvoasteikossa ylemmäs.
- Vain parhaiten menestyneellä oli mahdollisuus tulla kloonatuksi. Kloonauksen todennäköisyys riippui *aaveen* tehokkuudesta.

- Jälkeläisessä saattoi esiintyä mutaatio 0,01 todennäköisyydellä. Mutaation suuruus oli kääntäen verrannollinen *aaveen* tehokkuuteen, eli mitä tehokkaampi *aave* oli sitä pienempi oli haitallisen mutaation todennäköisyys.
- Jälkeläinen arvioitiin kuten kohdassa 1, peliltä näkymättömissä, ja verrattiin sen jälkeen huonoiten suoriutuvaan *aaveeseen*. Vertailun voittaja sijoittui sen jälkeen kentälle. [13]

Yannakakis ja Maragoudakis [14] kehittivät edelleen menetelmän (PM-OLL), jossa he yhdistivät pelaajan mallinnuksen sekä edellä esitetty evolutionääriseen algoritmiin perustuvan oppimismenetelmän. Menetelmän testaukseen valittiin viisi kiinnostavuudeltaan erilaista *aaveryhmää*. Jokainen ryhmä pelasi jokaista kolmea *PacMan*-tyyppiä vastaan, siten että *PacMan*-tyyppiä vaihdettiin 20 pelin jälkeen. Näin muodostui 60 pelin sarja, viisi erilaista *aave*-tyyppiä ja kuusi erilaista kolmen *PacManin* sarjaa. Tämän jälkeen sarja analysoitiin laskemalla kiinnostavuus kaavan 2 mukaan. Vertailukohtana käytettiin vastaavan testin tuloksia pelkästään OLL-menetelmällä suoritettuna. [14]

Tulokset on esitetty kuvassa 10.



Kuva 10: Yannakakis ja Maragoudakis [14]: Kiinnostavuustestin tulokset vaihteluvälineen

Tuloksena saatiin että 23:ssa tapauksessa 30:sta pelin kiinnostavuus nousi ja näistä 12:en nousu oli merkittävää luokkaa. Yannakakis ja Maragoudakis [14] päätyivät tulokseen, jonka mukaan pelaajan mallinnusprosessi yhdistettynä evolutionää-

riseen algoritmiin perustuvaan oppimiseen on tehokas tapa lisätä pelin kiinnostavuutta. Lisäksi kiinnostavuuden nousun suuruus sekä prosessin nopeus osoittautuivat hyviksi.

5 Yhteenveto

Tässä tutkielmassa on tuotu lyhyesti esille, mitä tarkoitetaan pelaajan mallinnuksella ja miten siitä saatuja tuloksia voidaan hyödyntää pelin suunnittelussa. Mallinnuksella kerätään tietoa pelaajien suosimista pelitavoista, minkä jälkeen saatua tietoa hyödynnetään pelinkehitysprosessissa. Useat tutkimukset osoittavat, että lisäämällä pelin adaptiivisuutta siitä voidaan tehdä houkuttelevampi yhä useammille pelaajille. Käytössä olevia tekniikoita ja tapoja on olemassa useita erilaisia. Lisäksi nopeasti kehittyvä tekniikka mahdollistaa yhä uusia ja entistä laajempia adaptoitumisprosesseja. Yksi adaptoitumisen tärkeimpiä sovellusalueita ovat niin sanotut hyötypelit (engl. serious games) kuten simulaatiot ja opetuspelit. Näissä adaptiivisuus on merkittävässä asemassa, silloin kun halutaan parantaa oppimistuloksia. Yleisesti ottaen pelin adaptiivisuus koetaan positiivisena asiana. On kuitenkin olemassa muutamia argumentteja, jotka ovat pelin jatkuvaa muuntumista vastaan. Eräs näistä on pelaajan halu oppia ja kontrolloida peliä. Pelin jatkuvasti muuttuessa pelaaja voi menettää kontrollin tunteen. Toinen merkittävä asia on pelaajakokemusten muodostuminen erilaisiksi eri pelaajien välillä. Tämän vuoksi kokemusten vertailusta tulee vaikeaa ja se vaikuttaa pelin ulkopuoliseen pelikokemukseen.

Lähteet

- [1] Sander Bakkes, Joris Dormans, *Involving Player Experience in Dynamically Generated Missions and Game Spaces*, 11th International Conference on Intelligent Games and Simulation, GAME-ON'2010, Leicester, United Kingdom, 2010, s. 72-79.
- [2] Darryl Charles, Michael McNeill, Moira McAllister, Michaela Black, Adrian Moore, Karl Stringer, *Player-Centred Game Design: Player Modelling and Adaptive Digital Games*, Proc. DiGRA Conf. Changing Views-World in Play, Vancouver, BC, Canada, 2005, s. 285-298.
- [3] Anders Drachen, Alessandro Canossa, George N. Yannakakis, *Player Modelling using Self-Organization in Tomb rider: Underworld*, Computational Intelligence and Games, Milano, 2009, s. 1-8.
- [4] Kiel M. Gilleade, Alan Dix, *Using Frustration in the Design of Adaptive Videogames*, Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, Singapore, 2004, s. 228-232.
- [5] Teuvo Kohonen, *Self-organized formation of topologically correct feature maps*, Biological Cybernetics 43(1), 1982, s. 59-69.
- [6] Ricardo Lopes, Rafael Bidarra, *Adaptive Challenges in Games and Simulations: A survey*, Computational Intelligence and AI in Games, 3(2), 2011, s. 85-99.
- [7] Timo Ryhänen, *Itseorganisoituva kartta*, Seminaarityö, Tietotekniikan osasto, Lappeenrannan teknillinen korkeakoulu, Lappeenranta, 1999.
- [8] Jyri Saarikoski, *Kohosen itseorganisoituvasta kartasta tekstidokumenttien tiedonhaussa* Pro gradu -tutkielma, Tampereen yliopisto, Tietojenkäsittelytieteiden laitos, 2007, s. 1-63.
- [9] David Thue, Vadim Bulitko, Marcia Spetch, Eric Wasylshen, *Interactive Storytelling: A Player Modelling Approach*, Association for the Advancement of Artificial Intelligence, 2007, s. 43-48.
- [10] Marko Tuononen, *Klusterointimenetelmät*, Joensuun yliopisto, Tietojenkäsittelytiede, 2005, s. 1-6.

- [11] Alfred Ultsch, *Maps for Visualization of high-dimensional Data Spaces*, DataBio-nics Research Lab, Department of Computer Science, University on Marburg, Germany, 2003, s. 225-230.
- [12] Chee-onn Wong, Jongin Kim, Eunjung Han, Keechul Jung, *Human-centered modeling for style-based adaptive games*, Journal of Zhejiang University - Science A , 10(4), 2009, s. 530-534.
- [13] Georgios N. Yannkakis, John Hallam, *Evolving Opponent for Interesting Interactive computer Games*, From animals to Animats, 8, 2004, s. 499-508.
- [14] Georgios N. Yannkakis, Manolis Maragoudakis, *Player Modelling Impact on Player's Entertainment in Computer Games*, User Modelling 2005, Lecture Notes in Artificial Intelligence 3538, 2005, s. 74-78.