

Mikko Kyttä

Proseduraalinen kerronnan luonti

Luonnontieteiden ala

Tradenomi

Syksy 2015

Opinnäytetyö



KAJAANIN
AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

TIIVISTELMÄ

Tekijä: Kyttä Mikko

Työn nimi: Proseduraalinen kerronnan luonti

Tutkintonimike: Tradenomi, Tietonjenkäsittely

Asiasanat: Kerronta, Peliohjelmointi, Algoritmit

Proseduraalinen luominen on ollut osa pelejä jo pitkään, ja sen käyttö on kasvanut viimeiset kymmenen vuotta. Proseduraalinen kerronta on hankalasti sovellettava proseduraalisen luomisen tyyli, jota käytetään joissakin indie-peleissä, mutta ei juurikaan isoissa AAA-peleissä.

Tässä opinnäytetyössä tutkitaan proseduraalista kerrontaa teorian ja käytännön työnä toteutetun peliprojektin avulla. Ensin tutkitaan kerrontaa ja siihen läheisesti liittyvää tarinaa pelien kannalta. Sen jälkeen käydään proseduraalisen luomisen perusteet lävitse, ja teoriaosuuden lopussa tarkastellaan proseduraalista kerrontaa.

Opinnäytetyössä käydään lävitse myös peliprojektiin tehty kerrontajärjestelmä, joka käyttää proseduraalisen kerronnan käsitteitä luodakseen peliin tapahtumia. Järjestelmän tavoitteena on sitoa tapahtumien syyt ja seuraukset pelin pelattavuuteen. Tehty järjestelmä toimii perustason tapahtumanluojana, ja on helposti laajennettavissa.

ABSTRACT

Author: Kyttä Mikko

Title of the Publication: Procedural narrative generation.

Degree Title: Bachelor of Business Administration, Business information technology

Keywords: Narrative, Game Programming, Algorithms

Procedural generation has been part of video games for a long time, and its use has increased in the last ten years. Procedural narrative is one of the hardest styles of procedural generation to implement, and it is not used in AAA-games and only rarely in indie games.

Procedural narrative is examined in this thesis, starting with the elements of which it is composed. At first, narrative and closely related story are studied from the point of view of video games. Secondly, the basics of procedural generation are dealt with, and finally the procedural narrative, which merges these two previous subjects together.

Narrative system used to create events in game project is also inspected in the thesis. The goal of this system is to integrate the causes and consequences of events created to the gameplay of the project. Completed project works as basic event creator, and is extensible.

Sisällys

SYMBOLILUETTELO	
1 JOHDANTO.....	
2 TARINA JA KERRONTA	2
2.1 Tarina	2
2.2 Kerronta	3
2.3 Tarina vastaan kerronta	3
2.4 Emergentti kerronta.....	4
2.5 Pöytäroolipelit.....	5
2.5.1 Pelimestari	6
2.5.2 Pelaajien vapaus.....	6
2.5.3 Pelimaailma ja vuorovaikutus	7
2.5.4 Erot videopeleihin.....	7
3 PROSEDURAALINEN LUOMINEN.....	8
3.1 Proseduraalisen luomisen periaatteet	8
3.2 Proseduraalisen luomisen käyttö.....	9
3.3 Luontityylit	9
3.3.1 Mallipohjainen	9
3.3.2 Muuttujapohjainen.....	11
3.3.3 Puhtaasti algoritmipohjainen	11
3.4 Esimerkkejä.....	12
4 PROSEDURAALINEN KERRONTA.....	13
4.1 Proseduraalisen kerronnan tarkastelu.....	13
4.1.1 Tärkeät ominaisuudet.....	14
4.1.2 Valinnan harha	15
4.1.3 Rajoitukset	16
4.2 Olemassaolevien järjestelmien tarkastelu	17
4.2.1 Rimworld	17

4.2.2 Minstrel remixed.....	18
4.2.3 The Director	19
4.2.4 Dwarf Fortress.....	20
5 OMAN PROJEKTIN KERRONTAJÄRJESTELMÄN LÄPIKÄYNTI	22
5.1 Projektin ja kerrontajärjestelmän kuvaus.....	22
5.1.1 Tavoite	22
5.1.2 Toimintasuunnitelma	23
5.2 Kerrontajärjestelmän läpikäynti	23
5.3 Onnistumiset ja vastoinkäymiset	25
6 POHDINTA.....	26
LÄHTEET	29

SYMBOLILUETTELO

Emergentti	Emergenssiin liittyvä, eli uusien ominaisuuksien syntyminen yksinkertaisten ominaisuuksien vuorovaikutuksesta
Hiekkalaatikkopeli	Pelityyppi, jossa pelaaja on vapaa liikkumaan ja toimimaan rajatulla alueella
Lineaarinen	Suoraviivainen
Proseduraalinen luominen	Sisältö luodaan algoritmien avulla ihmisen sijasta
Pöytäroolipeli	Pelityyppi, jossa pelaajat eläytyvät kuvitteellisiin rooleihin ja pelimestari ohjaa pelin kulkua
Pelimestari	Pöytäroolipelissä peliä johtava henkilö
Vapaa tarina	Tarina, jossa tarina seuraa pelaajaa eikä toisinpäin

1 JOHDANTO

Kerronta ja proseduraalinen luominen ovat esiintyneet videopeleissä lähes alusta asti. Yksi varhaisimmista peleistä, jossa esiintyvät molemmat on Rogue vuodelta 1980. Sen ilmestymisen jälkeen on kumpikin osa-alue kehittynyt valtavasti eteenpäin. Kerronta ja tarina peleissä on tutkinnan ja väittelyn aiheena, ja se on oma eriyvä lajinsa muista kerronnan lajeista, kuten kirjoista ja elokuvista. Proseduraalinen luominen on kokenut valtavan nousun vuonna 2011 julkaistun videopelin Minecraftin jälkeen. Huippusuositussa pelissä on valtava proseduraalisesti luotu maailma, ja monet pelit ovat ottaneet siitä mallia. (Lee 2014.)

Proseduraalista luomista voidaan käyttää videopeleissä lähes kaikkiin osa-alueisiin, perusosasista kuten musiikista ja malleista aina monimutkaisiin käsitteellisiin osiin kuten pelin sääntöihin asti. Laajimmiten sitä käytetään karttojen ja maailmojen luontiin. (Hendrikx, Meier, van der Velden & Iosup 2012.)

Kerronta on yksi pelin osa-alueista, joten sitäkin voidaan luoda proseduraalisesti. Proseduraalista luomista tehtäessä täytyy tekijän kuitenkin tietää tehtävä asia perinpohjaisesti, jotta tietokone saadaan tekemään asiat oikein. Toisin kuin karttojen luomisessa, kerronnassa ja tarinoissa ei ole helppoja mittatikkuja niiden laadun mittaamiseen. Tutkimalla tarinoiden ja kerronnan luonnetta ja koostumusta sekä ottamalla mallia vastaavista järjestelmistä saadaan lähtökohta ja suuntaviivoja, joiden avulla voidaan alkaa suunnittelemaan kerronnan luontia. (Young 2013)

Tässä opinnäytetyössä tutkitaan ensin kerrontaa ja proseduraalista luomista yleisesti ennen proseduraalisen kerronnan läpikäymistä. Näitä oppeja käytetään myös kerrontaluontijärjestelmän suunnitteluun ja tekemiseen, mikä käydään lävitse tutkinnallisten osuuksien jälkeen.

2 TARINA JA KERRONTA

Tarinaa ja kerrontaa esiintyy monessa muodossa, kuten kirjoina, elokuvina ja teatterinäytöksinä. Yksi uusimmista muodoista ovat videopelit, jotka ovat verrattain nuori laji muiden rinnalla. Kuten näytös teatterissa eroaa suuresti elokuvasta, eroaa pelienkin tarina ja kerronta aikaisemmista muodostuen omaksi tyylikseen. (Sutherland 2005.)

Videopelien tarinoita verrataan usein kirjoihin ja elokuviin. Pelissä kokonaisuus on erilainen johtuen vuorovaikutuksesta pelaajan ja pelin välillä, mutta itse tarinan rakenne on hyvin samanlainen. Videopelien eroista perinteiseen mediaan on väitely runsaasti, sillä pelien kerronta ei ole suoraan verrattavissa perinteiseen mediaan. (Jenkins 2011.)

2.1 Tarina

Tarina on pohjimmiltaan tapahtumasarja, joka selittää mitä tapahtui ja miksi. (de Boer 2014.) Tarinoilla on monia muotoja, mutta useimmat noudattavat perinteistä kaavaa, koska se on tunnettu, yksinkertainen ja toimiva. Perinteisessä kaavassa on sankari, jonka maailma muuttuu aloitustapahtuman takia. Rako muodostuu sankarin ja normaalin elämän väliin, ja normaali, varovainen yritys ratkaista ongelma epäonnistuu. Sankari ottaa riskejä ylittääkseen ensimmäisen ongelman, mutta sitten tapahtuukin mullistus, joka paljastaa sankarille jotain uutta ja luo toisen esteen sankarin eteen. Sankari ohittaa esteen isommilla riskeillä ja uusi mullistus luo kolmannen esteen. Sankarin täytyy ottaa isoin riski ylittääkseen viimeisen esteen saavuttaakseen haluamansa asian, joka on usein paluu normaaliin. (Sutherland 2005.)

Kaikki tarinat eivät noudata perinteistä kaavaa, mutta lähes kaikissa on tunnistettavia aineksia, kuten tarinan sankari eli päähenkilö ja esteitä sankarin ja tavoitteen välissä. Kirjoittajilla on käytössään monia työkaluja, joilla luodaan tunnelmaa ja kiintymystä päähenkilöön. Videopeleissä on monia muuttujia, jotka

saattavat vaikeuttaa näiden työkalujen käyttöä, kuten esimerkiksi videopelissä pelaaja itse saattaa olla päähenkilö. Toisaalta pelaajan vuorovaikutus pelimaailman kanssa antaa uusia mahdollisuuksia kertoa tarinaa. (Sutherland 2005.)

2.2 Kerronta

Pelin kerronta on pelin osa, joka osallistuu tarinan kertomiseen. Ovatko pelit kerrontaa vai onko kerronta vain yksi tapa kokea peliä, on vielä kiivaan keskustelun kohteena, ja aiheeseen onkin kaksi eriävää lähestymistapaa. Toinen näistä on narratologia, joka käyttää kerrontaa ja kirjallisuutta pohjana vuorovaikutteisen median eli videopelien teorian rakentamisessa. Ludologia taas tutkii pelin rakennetta ja pelattavuutta pelin kerrontana olemisen sijasta. (de Boer 2014.)

Kerronta on osa tarinaa. Tarinan kertominen on kerrontaa, ja sen ei tarvitse noudattaa syy-seuraussuhteita, toisin kuin tarinan. Kerronta ei tarvitse tarinaa ollakseen olemassa. Tetriksessä ei ole tarinaa, ja peli on tippuvien palikoiden asettelemista, mutta siitä voi silti muodostua kerrontaa. Pelaaja voi tarvita pitkää palikkaa tyhjentääkseen pelikentän, ja viime hetkellä oikea palikka saapuu pelastamaan tuomitulta näyttävän pelikerran. Vaikka tarinaa ei olekaan, kerrontaa syntyy silti ja edellä mainitussa tapauksessa pelaaja voi luoda kerronnasta tarinan, jota kertoa eteenpäin. (de Boer 2014.)

2.3 Tarina vastaan kerronta

Tarinaa ja kerrontaa käytetään usein tarkoittaen samaa asiaa, ja kerrontaa voidaan käyttää tarkoittaen pelin tarinaa, mutta ne eroavat silti merkitykseltään.

Ne ovat silti läheisesti sidotut toisiinsa, sillä kerrottu tarina on kerrontaa, ja pelkästä kerronnasta voi helposti muodostua tarina. (de Boer 2014.)

Yksi iso ero tarinan ja kerronnan välillä on suoraviivaisuus. Tarinassa on syy ennen seurausta, kun taas kerronnassa ei tarvitse kulkea aikajanaa pitkin, vaan kerronta voi alkaa seurauksesta ja päättyä syyhyn. (de Boer 2014.)

2.4 Emergentti kerronta

Vaikka videopelin tarina olisikin vähäeleinen tai olematon, se saattaa luoda paljonkin emergenttiä kerrontaa. Yksinkertaisista elementeistä ja mekaniikoista saattaa muodostua yhdessä monimutkaisia tapahtumia ja vuorovaikutuksia. Näistä tapahtumista muodostuu kerronta, jota ei ole etukäteen kirjoitettu eikä välttämättä edes suunniteltu. (Lee 2013.)

Emergenttiä kerrontaa esiintyy myös peleissä, joissa on kirjoitettu tarina. Tarkemmin suunniteltu kerronta vähentää emergenttiä kerrontaa, koska jää vähemmän syitä ja seurauksia mielikuvituksen varaan. (Lee 2013.)

Suunniteltuun kerrontaan verrattuna emergentti kerronta on henkilökohtaisempi. Emergentissä kerronnassa pelaaja luo itse syyt ja seuraukset tapahtumille, kun taas suunnitellussa kerronta annetaan valmiina pelaajalle. Eri pelaajille samat tapahtumat voivat muodostaa erilaisen kerronnan, sillä tapahtumien tulkinta vaihtelee henkilöstä toiseen. (Lee 2013.)

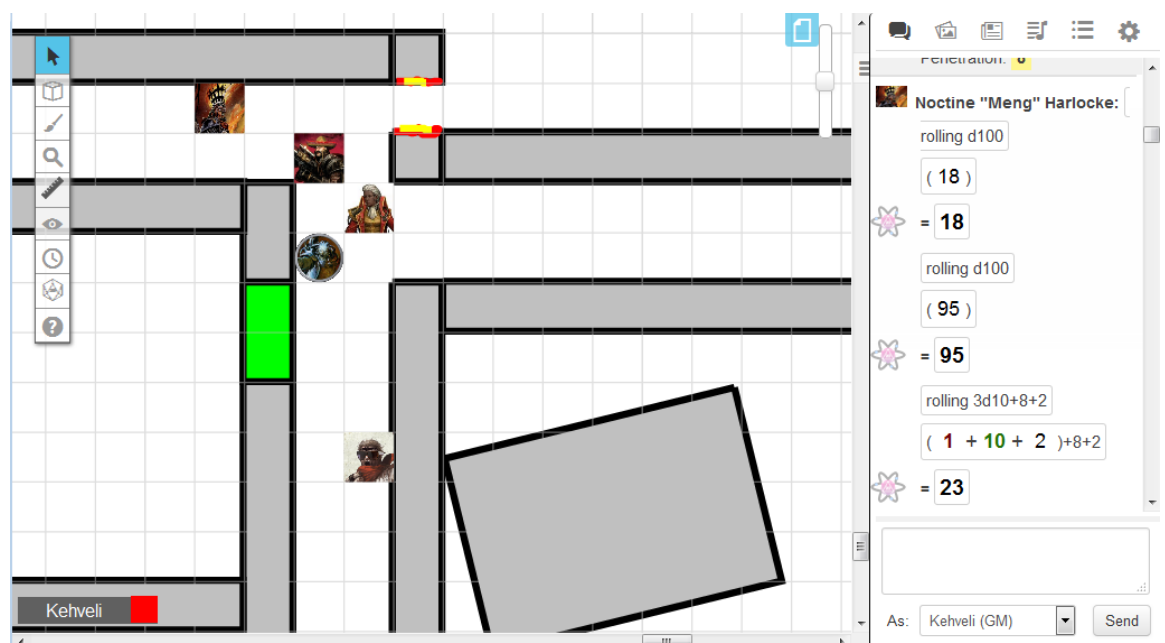
Vaikka emergentti kerronta onkin pelaajaa läheisempi, se on silti toissijainen suunniteltuun kerrontaan verrattuna. Jos pelaaja luo oman syyn pelitapahtumalle, ja myöhemmin samalle tapahtumalle annetaan pelin tarinassa toinen syy, joutuu pelaaja luopumaan omasta kerronnastaan suunnitellun hyväksi. Emergentistä kerronnasta myös puuttuu usein laajempi konteksti; usein emergentti kerronta on vain sarja hieman toisiinsa liittyviä tapahtumia, joista pelaaja itse rakentaa kerronnan. Tällöin kerronnasta jää usein myös pois merkitys, koska siinä ei

varsinaisesti ole teemaa, opetusta tai tarkoitusta. Raa'assa muodossa emergentistä kerronnasta puuttuu myös kerronta, jossa edellä mainitut konseptit esitetään. (Lee 2013.)

Vaikka emergentti kerronta saattaa jäädä helposti tyhjän oloiseksi ja sen suurin merkitys syntyy pelaajalle vuorovaikutuksesta sen kanssa, voidaan siitä silti luoda eheitä kerrontoja, joilla on laajempi merkitys. Kerrottu tapaus emergenssistä kerronnasta muuttuu kertojan käsittelyssä tarinaksi. (Dinicola 2013.)

2.5 Pöytäroolipelit

Videopeleissä on hankala saavuttaa todellista tarinallista vapautta, sillä yksinkertaisillakin valinnoilla olisi suuri kerrannaisvaikutus, ja pelintekijöillä on rajallinen määrä resursseja. Jo yksinkertainen "Haluatko pelastaa prinsessan? Kyllä/Ei" pelin alussa kaksinkertaistaisi tekemiseen tarvittavan työmäärän, elleivät pelit pakottaisi Kyllä-vastausta tavalla tai toisella. Pöytäroolipeleissä tätä ongelmaa ei ole, ja tarina on täysin vapaa liikkumaan pelaajien mukana. (Portnow 2013.)



Kuva 1. Pöytäroolipelitilanne Roll20 verkkosivustolla.

2.5.1 Pelimestari

Pöytäroolipeleissä joustava tarina syntyy pelimestarin ja pelaajan yhteistyöstä. Pelimestari ohjaa pelin tarinaa eteenpäin tapahtumilla pelimaailmassa ja pelaajat tekevät päätöksiä pelihahmonsa näkökulmasta. Yhdessä näistä syntyy vuorovaikutteinen tarina, jossa pelaajat reagoivat tarinaan ja tarina vastaa pelaajien päätöksiin. Pelillisyyttä ja satunnaisuutta antamassa on pöytäroolipeleissä usein noppa, jolla päätetään onnistuuko pelihahmojen tekemiset. Tämäkin vaikuttaa tarinaan, usein odottamattomilla tavoilla. (Koebel & LaTorre 2012.)

Pelimestari hallitsee pöytäroolipelissä kaikkea muuta paitsi pelaajien hahmoja, ja hänellä on viimeinen sana pelin säännöistä ja tapahtumista. Koska peliympäristö ja hahmot ovat vain paperilla tekstinä ja pelaajien päässä, on pelimestarin myös toimittava pelaajien ikkunana pelimaailmaan. Pelimestari kuvailee pelaajille kaiken, mitä pelissä tapahtuu ja mitä pelaajat tarvitsevat tai haluavat tietää. Nykyään käytetään myös ohjelmia tilanteen kuvaamiseen ja noppien korvaamiseen, kuvassa 1 on yksi näistä ohjelmista. (Koebel & LaTorre 2012.)

Pelimestarin rooli pelin hallitsijana on tärkeä, sillä kaikki pelin tapahtumat kulkevat hänen kauttaan. Hänen on pidettävä pelimaailma ja tarina pelinsisäisen maailman kannalta johdonmukaisina ja mietittävä pelaajille olennaisia tapahtumia ja seurauksia. (Koebel & LaTorre 2012.)

2.5.2 Pelaajien vapaus

Pelaajilla on pöytäroolipeleissä täysi vapaus tehdä mitä haluavat, vaikka pelimestarilla onkin viimeinen sana siitä, mikä on mahdollista ja mitä tapahtuu. Jos pelaajia pyydetään pelissä pelastamaan prinsessaa, ei heidän ole pakko suostua tehtävään. Päätöksellä tulisi kuitenkin olemaan seurauksia, jotka seuraisivat

pelaajia mahdollisesti pitkäänkin. Toiminnan vapaus onkin yksi pöytäroolipelien kulmakivistä. (Koebel & Lumpkin 2014.)

2.5.3 Pelimaailma ja vuorovaikutus

Pelimestarin päätehtävänä on luoda pelaajille mielenkiintoinen ympäristö ja jännittäviä päätöksiä ja tapahtumia. Vaikka pelimestari voisikin tehdä pelimaailmasta täysin totuudenkaltaisen, on helpompi tehdä vain näennäisesti johdonmukainen maailma, joka on valmis muuttumaan pelaajien päätösten mukana. (Koebel & LaTorre 2012.)

Seuraus pelimaailman olemassaolosta pelkällä ajatustasolla mahdollistaa asioiden lisäämisen maailmaan ilman, että se tuntuisi epäuskottavalta. Jos pelaaja etsii roskapönttöä kujalta, hän luultavasti löytää sen, vaikka pönttöä ei olisi etukäteen suunniteltu siellä olevaksi. Siltikään se ei ilmesty pelimaailmaan tyhjästä; sitä vain ei ollut aikaisemmin kuvailtu. (Koebel & LaTorre 2012.)

2.5.4 Erot videopeleihin

Yksi pöytäroolipelien eroista videopeleihin on tarinan joustavuus. Kummassakin on pelillisiä elementtejä, kuten nopan heittoa, ja vuorovaikutusta pelimaailman kanssa. Videopeleissä kaikki, mikä pelaajalle näytetään, täytyy valmistaa etukäteen rajoittaen laajuutta sekä pelillisesti että tarinallisesti. Videopeleissä tarina ja pelattavuus ovat usein erillään toisistaan, jolloin syntyy ristiriita näiden kahden osan välille, kun taas pöytäroolipeleissä tarina on osa pelattavuutta. (Gillen 2010.)

3 PROSEDURAALINEN LUOMINEN

Monissa videopeleissä käytetään proseduraalista luomista. Se ei rajoitu pelkästään karttojen luontiin, mihin se yleisesti mielletään, vaan esimerkiksi pelihahmon kävelyanimaatio maaston muotojen mukaisesti luodaan proseduraalisesti pelissä. (Short 2014.)

Proseduraalisesti luotu sisältö pelissä, kuten kartat ja aseet, täytyy suunnitella tarkasti, sillä muuten ne saattavat jäädä tylsiksi. Proseduraalisesti luodun sisällön sekaan voidaan lisätä käsintehtyä sisältöä. (Short 2014.)

3.1 Proseduraalisen luomisen periaatteet

Proseduraalinen luominen on tietokonealgoritmien käyttämistä automaattiseen sisällön luomiseen. Usein proseduraalista luomista kutsutaan satunnaiseksi luomiseksi, mikä on puolet totuudesta. Täysin satunnaisella luomisella voidaan saada mitä tahansa aikaan, mutta suurimman osan ajasta lopputulos on täysin käsittämätöntä. Proseduraalisessa luomisessa satunnaisuutta ohjataan joidenkin alkuarvojen mukaisesti, kuten videopeli Civilization V:ssä pelaaja voi vaikuttaa kartan luomiseen muuttamalla arvoja kuten kosteustaso, korkeusvaihtelut ja saaristojen määrä. Lisäksi luodulla sisällöllä on rajoitteita. Proseduraalisten kenttien luonnissa usein tarkistetaan kentän läpäisymahdollisuus, ettei pelaajalle tule vastaan mahdottomia esteitä. Riippuen sovelluskohteestaan, proseduraalinen luominen voi olla myös sopeutuvaa. Mikäli pelin kentät luodaan proseduraalisesti, voidaan niiden vaikeustasoa muokata pelaajan onnistumisten mukaisesti. (Kastbjerg, Schedl, Togelius & Yannakakis 2011.)

Jotkin pelit saattavat käyttää proseduraalisen luomisen sijasta tai rinnalla käsintehtyä sisältöä. Videopelissä Spore on monipuolinen työkalu erilaisten olentojen luomiseen jota pelaaja voi käyttää. Luotuja olentoja käytetään muilla pelikerroilla pelin mukana tulleiden olentojen lisäksi. Mikäli peli yhdistää internettiin, se jakaa luotuja olentoja muille pelaajille sekä ottaa muiden pelaajien

olentoja kansoittamaan maailmaa. Tällainen käsintehdyn sisällön käyttö ei ole proseduraalista luomista. (Kastbjerg, Schedl, Togelius & Yannakakis 2011.)

3.2 Proseduraalisen luomisen käyttö

Pelien valmistamisen kesto ja hinta ovat nousseet viimeisien vuosikymmenien aikana suuresti. Menestyneitä pelejä tehnyt yhtiö Id Software loi 1990-luvun puolivälissä klassikoita kuten Commander Keen, Wolfenstein 3d ja Doom, viiden tai kuuden henkilön kokoaikaisen työntekijän voimin. 2000-luvun alkupuolella pelinkehitysryhmät saattoivat koostua jo sadoista ihmisistä. Yksi tapa välttää työvoimamäärän ja kulujen kasvua on käyttää proseduraalista luomista peleissään, jolloin osan työmäärästä voi antaa tietokoneen tehtäväksi. Usein proseduraalista luomista käytetään pelin suunnittelijoiden työmäärän vähentämisessä, käyttäen algoritmeja kenttien ja muiden ympäristöjen luomisessa, mutta lähes jokaisessa pelin osa-alueessa voidaan käyttää proseduraalisen luomisen valmistamia tuotteita. (Hendrikx, Meier, van der Velden & Iosup 2012.)

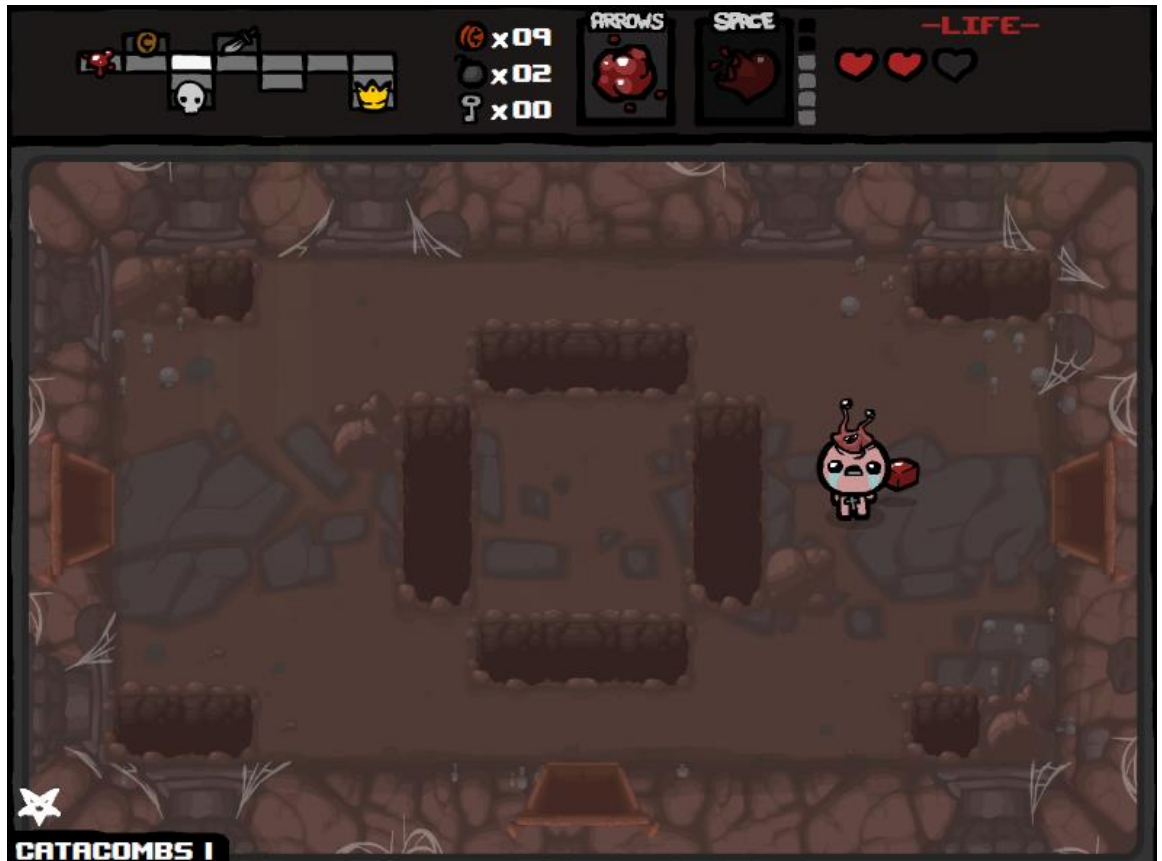
3.3 Luontityylit

Proseduraaliseen luomiseen on olemassa useita eri lähestymistapoja. Seuraavaksi läpikäytävät tavat ovat yleistyksiä useista tavoista, joita käytetään proseduraalisessa luomisessa. (Hendrikx, Meier, van der Velden & Iosup 2012.)

3.3.1 Mallipohjainen

Tässä luomistyyppissä luomisen lopputulos koostuu pienemmistä palasista, jotka on aseteltu sopivasti yhteen. Tätä käytetään usein peleissä, joissa halutaan yhdistää käsintehtyjä osia satunnaisuuteen, kuten videopeli Binding of Isaacissa.

Siinä pelimaailma koostuu huoneista, jotka on yhdistetty yksinkertaisella proseduraalisella algoritmilla toisiinsa. Huoneiden pohjapiirustus valitaan satunnaisesti yli tuhannesta käsintehdystä vaihtoehdosta. Näin saadaan pelikerrasta toiseen muuttuva kartta, joka välttää monia proseduraalisen luomisen ongelmia käyttämällä sitä vain yleisen ulkomuodon rakentamiseen. Kuvasta 2 näkee esimerkin kartasta ja kentästä. (McMillen 2011.)



Kuva 2. Bindings of Isaacin kenttä. Kartta sijaitsee vasemmassa yläkulmassa.

Mallipohjainen luominen on usein muita tässä mainittuja proseduraalisen luomisen tyylejä kevyempi toteuttaa, sillä luomista käytetään pelkästään osaan kokonaisuudesta, kuten huoneiden sijoitteluun, ja yksityiskohtaisemmat asiat, kuten huoneiden sisältö, tehdään joko käsin tai erityyppisellä proseduraalisella

luomisella. Siksi tämän tyylistä luomista käytetään usein kasvattamaan uudelleenpeluuarvoa ja helpottamaan kenttäsuunnittelijan työtä. (Young 2013)

3.3.2 Muuttujapohjainen

Muuttujapohjaisessa tyyliässä pelin muuttujia, kuten vihollisten sijainteja tai aarrearkkujen sisältöjä, muutetaan pelikerrasta toiseen. Näin saadaan vaihtelua pelikertoihin ja pystytään hallitsemaan satunnaisuutta, esimerkiksi antamalla aarrearkulle listan esineistä, joita se voi sisältää. (McMillen 2011.)

Muuttujapohjaista käytetään usein mallien kanssa, kuten aikaisemmin mainitussa Binding of Isaacissa, jossa mallihuoneiden rahamäärää ja vihollisia voidaan vaihdella. Tällöin sama huone on eri pelikerroilla eriävä aikaisemmista käyntikerroista. (McMillen 2011.)

3.3.3 Puhtaasti algoritmipohjainen

Usein peleissä käytetään proseduraalista luomista pelimaailman kartan luontiin. Jos kartan tai muun luotavan asian ei tarvitse olla erityisen tarkka tai jännittävä, on helppoa antaa tietokoneen luoda se kokonaan asetettujen alkuarvojen mukaisesti. Nämä alkuarvot voivat olla esimerkiksi maailman koko, korkeuserojen tai sateen määrä, ja niiden pohjalta luodaan jotain satunnaisesti. (Hendrikx, Meier, van der Velden & Iosup 2012.)

Kuten aikaisemmin mainittuja luontikeinoja, voidaan tätäkin yhdistää muihin luontitapoihin. Suositussa videopelissä Minecraft:ssa maailma luodaan suurimmaksi osaksi ilman ihmisen kädenjälkiä, mutta maailmaan saatetaan asetella kohteita, kuten temppeli tai kylä, jotka otetaan käsintehdyistä malleista. Alkuarvoja säätämällä voidaan maailmasta luoda esimerkiksi hyvin tasainen, aavikkovoittoinen tai vuoristoinen tundra. (Fitzpatrick 2014.)

3.4 Esimerkkejä

Proseduraalista luomista on käytetty videopeleissä pitkään, ja viime aikoina sen käyttö sekä sivu- että pääominaisuutena on lisääntynyt. Eräänlaisena lähtölaukauksena tälle kasvulle toimi Minecraft, jossa koko maailma luodaan proseduraalisesti. Sen räjähdysmäinen suosio sai monet kopioimaan pelin piirteitä, joista keskeisin on juuri maailman proseduraalinen luonti. (Lee 2014.)

Videopeli Spore käyttää proseduraalista luomista kokonaisen galaksin luomiseen ja asuttamiseen. Vaikka pelissä onkin monipuolinen luontityökalu lähes jokaiselle pelin osa-alueelle, ei peli silti luo olioita, rakennuksia ja muita muokattavia kappaleita proseduraalisesti työkaluillaan, vaan ottaa ne käsintehtyjen kappaleiden listasta. Planeetat luodaan algoritmillisesti, mutta sen mahdolliset asukkaat, eläimistöt, arkkitehtuurityylit sekä muut muokattavat asiat otetaan satunnaisesti listasta, ja ne ovat joko pelin tekijöiden tai muiden pelaajien käsin tekemiä. (Smith 2014.)

Proseduraalisen luomisen edelläkävijöitä ovat Roguelike-pelit, jotka käyttävät lähes kaikki proseduraalista luomista kenttien ja niiden sisällön luomisessa. Lajityypin nimeäjässä Roguessa pelaajan täytyy kulkea joka kerta erilaisen luolaston pohjalle hakemaan amulettia ja palata takaisin maan pinnalle. Jo vuonna 1980 tehty peli loi pohjan kokonaiselle pelilajityypille, joiden keskeinen osa on proseduraalisen luomisen käyttö pelin karttojen ja niiden sisällön luomisessa. (Lee 2014.)

4 PROSEDURAALINEN KERRONTA

Proseduraalisen luomisen käyttäminen kerronnan luomiseen on haastavaa. Kerronta ei ole helposti mitattava asia, joten algoritmit eivät kykene suoraan luomaan mielenkiintoista kerrontaa. Hyvää tarinaa ja kerrontaa on hankala tehdä omin käsinkin, joten tietokoneen kautta tehty kerronta on usein joko mielivaltaista tai kaavamaista. (Young 2013.)

Tietokone ei itsessään osaa tehdä mitään ilman täydellisiä ohjeita, joten täysin satunnaisesti luodut tarinat ovat erittäin hankalia tehdä. Jotta luotu tarina olisi hyvä ja mielenkiintoinen, tietokoneen täytyisi ensin tietää, mitä hyvä ja mielenkiintoinen tarina sisältää. Ohjelmoijan on vaikeaa saada tietokone tekemään mitään, mitä ohjelmoija ei itse osaa, sillä ohjelmointi on pohjimmiltaan asioiden selittämistä erittäin seikkaperäisesti tietokoneelle, joka noudattaa annettuja ohjeita täsmällisesti. Hyvän tarinan tekemiseen ei ole suoria ohjeita, joten tietokone täytyisi saada ymmärtämään monimutkaisia käsitteitä, joista ihmisetkään eivät ole vielä yhtä mieltä. (Young 2013.)

Yksi ratkaisu ongelmaan on tehdä tarinoita pohjista, joiden pienemmät osat vaihtuvat. Esimerkiksi pohjana voi toimia prinsessan kidnappaus, mutta kidnappaja, kidnapattu, tapa ja paikat vaihtuvat tarinakerrasta toiseen. Tällöin ongelmat tietokoneen kyvyttömyydestä luoda tarinoita tyhjästä vältetään, mutta tarinat alkavat silti toistaa itseään ja eivät salli pelaajan päätösten johtamista tarinapohjan ulkopuolelle. (Young 2013.)

4.1 Proseduraalisen kerronnan tarkastelu

Tarinan ja kerronnan luominen ovat hyvin lähellä toisiaan. Luomalla tarinan luo myös tarinan kerronnan, joskin kerronta ei suoraan luo tarinaa. Kerrontaa on myös

helpompi luoda, mutta se ei välttämättä ole merkityksellistä, jos pelaaja ei erota sitä satunnaisista tapahtumista. (Kline 2009)

Keskittymällä tiettyihin ominaisuuksiin proseduraalisen kerronnan luomisessa voidaan tehdä syntyvästä kerronnasta huomattavampaa. Monimutkaiset järjestelmät eivät välttämättä täytä tehtävänsä, mikäli pelaaja ei huomaa tekemistensä syy-seuraussuhteita. Yksinkertainen järjestelmä, joka kertoo pelaajalle selvästi, mikä asia johti toiseen, tuntuu usein paremmalta kuin monimutkainen ja epäselvä järjestelmä. (Kline 2009)

Algoritmille pitää myös opettaa pidemmissä tarinoissa ja kerronnoissa käytettäviä kikkoja, kuten rytmitystä ja tarinan rakenteita. Lyhyet kerrontapätkät saattavat olla mielenkiintoisia ilman perinteisiä rakenteita, mutta pitkät tarinat huonolla rytmityksellä tai ilman kohokohtia muuttuvat tylsiksi. Ongelmaksi muodostuu myös se, etteivät algoritmit osaa tehdä merkityksellisiä kohokohtia ilman aikaisempaa hyvää pohjaa. Jos luodun tarinan kohokohta on paljastus, joka ei oikeasti vaikuta juoneen, tai ei ole merkityksellinen pelaajalle, ei oikean kaltainen rakennekaan pelasta kerrontaa. (Thomas 2014)

4.1.1 Tärkeät ominaisuudet

Videopeleissä tärkeintä on pelaajan kokemus, joten luodun kerronnan olisi hyvä sisällyttää pelaaja tai pelaajan pelihahmo itseensä. Suuri ongelma proseduraalisesti luodussa sisällössä on sen hengettömyys. Esimerkiksi proseduraalisesti luodut tehtävät ovat harvoin pelaajalle tyydyttäviä, sillä ne ovat usein yksinkertaisia keräys- tai puhumistehtäviä. Monimutkaisempien tehtävien luominen vaatii jo monimutkaisia järjestelmiä. (Thomas 2014.)

Usein hengettömyys johtuu luodun sisällön irrallisuudesta muuhun pelimaailmaan, koska useissa peleissä proseduraalista luomista käytetään ylimääräisen sisällön

luontiin. Muusta pelistä irrallisen sisällön suorittamisen ainoa lopputulos on monesti pelaajahahmolle annettavat palkinnot, kuten kokemuspisteet tai varusteet. Joissain peleissä tällainen järjestely on riittävä, sillä peli ei ole kerrontapainotteinen, mutta mikäli pelissä kerronta on tärkeässä asemassa, rikkovat muusta pelistä irralliset osat kokonaisuutta. (Lee 2013.)

Pelaajan mielenkiinnon lisääminen luotua sisältöä kohtaan on sisällön merkitykselliseksi tekeminen. Kun luotu sisältö sidotaan pelimaailmaan vaikutuksiltaan ja käyttäen olemassaolevia osia, voidaan sisällön irrallisuutta pienentää. (Lee 2013.)

4.1.2 Valinnan harha

Koska oikeasti merkityksellisen kerronnan luominen on hankalaa, on helpompaa huijata pelaajaa. Valinnan harhaa käytetään usein tarinallisissa peleissä antamaan pelaajalle tunne vaikuttamisesta tarvitsematta tehdä oikean valinnanvapauden antamisen vaatimia suuria järjestelmiä. Isoissa peleissä valinnan tuntu on usein ainoa vaihtoehto, jonka pystyy toteuttamaan ilman valtavaa työpanostusta pelin jokaiseen osa-alueeseen. (Portnow 2013.)

Helpoin tapa luoda harha on tarjota useita valintoja, jotka esitetään eriävinä, mutta vaikuttavat vain pelin kulun kannalta merkityksettömiin asioihin ja johtavat lopulta samaan lopputulokseen. Oikeasti haarautuva tarina kasvattaisi pelin tekemisen työmäärää eksponentiaalisesti, jolloin lyhyestäkin tarinasta tulisi massiivinen. Illuusiolla saadaan sama tunne aikaan ilman valtavaa työmäärää. Videopeli Mass Effect:ssä käytetään tätä tekniikkaa mainiosti. Pelaaja pelastaa galaksin päätöksistään huolimatta, mutta onko pelaaja kaikkien rakastama sankari vai välttämätön paha riippuu valinnoista, joita pelaaja tekee. (Portnow 2013.)

4.1.3 Rajoitukset

Kuten kaikessa proseduraalisessa luomisessa, kerronnan luontia rajoittaa luontialgoritmin tekijän tieto aiheesta. Tarinankerronnassa ei ole selviä mittatikkuja joiden avulla voisi määritellä hyvän ja huonon eron, joten tietokonetta on erittäin hankala saada tekemään hyväksyttävää laatua luotettavasti. (Young 2013)

Vaikka onnistuttaisiin tekemään täydellinen kerrontalaite, nousee silti ongelmaksi pelin muut rajoitukset. Pelissä ei ehkä ole tehtynä sisältöä aavikkoalueelle, joten tarina ei silloin voi sisällyttää aavikkoa. Jotkin pelit, kuten Dwarf Fortress, kiertävät tämän ongelman pitämällä pelin ulkonäön äärimmäisen yksinkertaisena. Dwarf Fortressin tapauksessa ruohokenttä on vihreä piste ja aavikko keltainen, joten pelin kehitysaikaa ei tarvitse käyttää erilaisten ympäristöjen tekemiseen, ja pelin kerronta voi kulkea minnepäin virtuaalimaailmaa tahansa ilman rajoituksia. (Elliot 2014)

Toinen rajoitus on pelaajan huomio. Luontilaite saattaa luoda niin monimutkaisia ja hienovaraisia syy-seuraussuhteita peliin, ettei pelaaja tajua valintojensa vaikutuksia muuhun maailmaan. Tällöin iso ja tarkkakin järjestelmä tuntuu pelkästään satunnaiselta tapahtumasarjalta, joka ei seuraa normaaleja logiikan sääntöjä. (Kline 2009)

4.2 Olemassaolevien järjestelmien tarkastelu

Proseduraalista kerrontaa on alkanut ilmestymään peleissä muiden proseduraalisten luomisten rinnalla. Seuraavana on neljä esimerkkiä proseduraalisen kerronnan käytöstä.

4.2.1 Rimworld

Rimworld:ssa ohjataan kolmea kaukaiselle planeetalle haaksirikkoutunutta ihmistä. Tavoitteena on selviytyminen, haasteita ovat esimerkiksi lämpötila, alkuasukkaat, vihamieliset robotit, villieläimet ja sääilmiöt. Selviytyjien nopeasti rakentama siirtokunta kasvaa muiden paikallisten ihmisten liittyessä siihen, ja se saattaa romahtaa yhdessä hetkessä onnettomuuden tai kahden seurauksena. (Sylvester 2014.)

Yksi Rimworldin mainostetuista myyntivalteista on sen tarinankerrontatekoäly. Pelatessaan voi valita yhden kolmesta mahdollisesta tarinankerrontatyypistä itselleen parhaiten sopivan, ja pelin aikana tekoäly haastaa pelaajaa erilaisilla tapahtumilla. Tekoäly yrittää myös luoda dramaattisia hetkiä ja jännittäviä tilanteita pelaajan tekemiset huomioon ottaen. Mikäli pelaaja luottaa siirtokuntansa puolustuksessa aurinkoenergista virtansa saavilla tykeillä, saattaa tarinankertoja aiheuttaa auringonpimennyksen jonka turvissa rosvot hyökkäävät suojaattomaan siirtokuntaan. (Meer 2015.)

Kolme kertojatekoälytyyppiä ovat Cassandra Classic, Phoebe Chillax ja Randy Random. Cassandra Classic on näistä tasapainoisin, luoden jatkuvasti kasvavalla vaikeudella tapahtumia. Phoebe Chillax antaa pelaajalle enemmän aikaa onnettomuuksien ja tapahtumien välillä, kun taas Randy Random ei ota

pelitapahtumia huomioon vaan aiheuttaa onnettomuuksia ja tapahtumia satunnaisesti. (Sylvester 2014.)

Tarinankerrontatekoälyt luovat juuri kerrontaa eivätkä varsinaisesti tarinaa, sillä ne eivät luo tekstiä luettavaksi tai tarinaa seurattavaksi, vaan luovat tapahtumia jotka synnyttävät kerronnan kautta tarinoita kerrottavaksi. Tekijä Tynan Sylvester sanoo pohjauttaneensa tarinankerrontatekoälynsä Valven The Director:iin, joka on yksi myöhemmistä esimerkeistä. (Sylvester 2014.)

4.2.2 Minstrel remixed

Toisin kuin muut esimerkit tässä osiossa, Minstrel remixed ei ole videopeli, vaan projekti joka yritti luoda jännittäviä tarinoita kertovan ohjelman. Vuonna 1993 valmistettu Minstrel on yksi vaikuttavimmista tekoälyprojekteista. Scott Turnerin tohtorintutkimuksen opinnäytetyönä valmistama projekti on monien ylistämä, vaikka projektia ei tutkittu enempää mitä Turner väitöskirjassaan esitteli. Nykyään Minstrelistä ei ole olemassa toimivaa versiota tai lähdekoodia, mutta projektissa Minstrel remixed yritetään jälleenrakentaa Minstrel olemassaolevien tietojen perusteella. (Mateas, Tearse & Wardrip-Fruin 2010.)

Minstrel koostuu pääosin kahdesta komponentista. Ensimmäinen komponentti on Transform Recall Adapt Methods (TRAM), jonka tehtävänä on täyttää tarinan osasia sille annettujen vaatimusten pohjalta käyttäen apunaan tarinakirjastoa. TRAM:ejä on useamman mallisia, joista jokainen käsittelee annettujen rajoitusten ja tarinakirjastosta otettujen palojen muutosta ainutlaatuisella tavalla. Yksi yleislaatusimmista TRAM:sta on Generalize Constraints (yleistä rajoitteita), joka yleistää annettua tarinanpalaa poistamalla yhden sen rajoitteista. (Mateas, Tearse & Wardrip-Fruin 2010.)

Esimerkki, millä tavalla tämä TRAM voi toimia on seuraava: annetussa tarinapalassa vaaditaan ritari Jonin kuolemaa taistelussa. Generalize Constraints TRAM poistaa tästä yhden rajoitteista, ja päädytään vaatimukseen jossa jonkin on

kuoltava taistelussa. Tällä vaatimuksella voidaan löytää tarina tarinakirjastosta, jossa toinen ritari Pekka taistelee peikon kanssa tappaen sen lopussa. Tällöin TRAM voisi korvata peikon ritari Jonilla, palauttaen tarinapalan, jossa ritarit Joni ja Pekka taistelevat johtaen Jonin kuolemaan. (Mateas, Tearse & Wardrip-Fruin 2010.)

Toinen komponentti on Author Level Plan (ALP), mikä pitää huolen tarinasta kokonaisuutena ja sen suunnan muokkaamisesta, tavoittenaan täyttää koko tarinapohja tyydyttävällä tavalla. ALP koostuu kolmesta pienemmästä luokasta, jotka toimivat yhdessä: tarinan tuottajat, tarinan tarkastajat ja tarinan tehostajat. Tuottajan tehtävänä on antaa tarinalle paloja rajoitteilla, joiden mukaan TRAM täyttää ja palauttaa palan. Tarkastajat käyvät lävitse tarinan paloja ja lisäävät ALP:in listaan uusia vaadittavia paloja, kuten uuden hahmon esittely tai muiden puuttuvien osien lisäämistä. Tehostajien tehtävänä on lisätä uusia yksityiskohtia tarinaan lisätäkseen draamaa, jännitystä ja hahmoja kuvaavia ominaisuuksia. (Mateas, Tearse & Wardrip-Fruin 2010.)

4.2.3 The Director

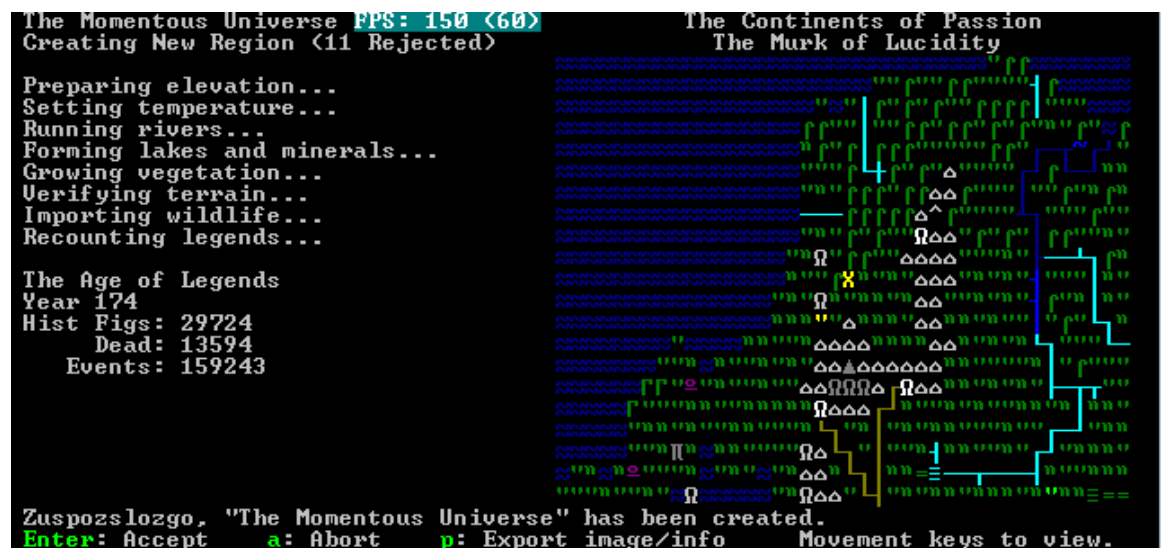
Left 4 Dead ja sen jatko-osa Left 4 Dead 2 ovat Valven tekemiä videopelejä, joissa neljä pelaajaa yrittää selviytyä elävien kuolleiden täyttämän alueen lävitse turvahuoneeseen. Matkalla pelaajat kohtaavat esteinä erikoiskyvyillä varustettuja zombeja ja valtavia zombimassoja, ja joutuvat ammuskelemaan tiensä niiden lävitse. Näissä peleissä on erikoisominaisuutena pelaajia ohjaileva The Director. (Walker 2009.)

Director on näiden pelien käyttämä kerronta- ja pelinohjausjärjestelmä, joka tarkkailee pelaajien toimintaa, ja sen pohjalta muokkaa pelin tapahtumia saadakseen aikaan mahdollisimman jännittävän pelikokemuksen. Luomalla pelaajille ammuksia ennen suurta taistelua, ohjaamalla pelaajat vaikeampaa reittiä nopean etenemisen jälkeen tai pakottamalla pelaajat säästämään resurssejaan

vihollismassoilla, se ohjaa pelin kulkua ja yrittää mukautua pelaajien taitotason mukaiseksi. (Walker 2009.)

Left 4 Deadeissa on myös The Directorin lisäksi mielenkiintoinen kerronnanluontityökalu. Peli pitää kirjaa pelaajien tekemisistä ja tietojensa mukaan aiheuttaa pelihahmojen välistä keskustelua ja puhumista. Esimerkiksi katsomalla seinällä olevaa julistetta pelihahmo saattaa sanoa jotain julisteesta ohimennen, tai jättää sanomatta jos pelitilanne ei ole järkevä julisteiden huomioimiseen. (Ruskin 2012.)

4.2.4 Dwarf Fortress



Kuva 3. Dwarf Fortressin maailmanluonti-ikkuna.

Dwarf Fortress on peli, jossa pelaaja joutuu kaitsemaan humalaisten kääpiöiden siirtokuntaa. Monimutkaisuudestaan kuuluisa, Dwarf Fortress on vuonna 2015 ollut 13 vuotta kehityksen alla, eikä pääkehittäjä Tarn Adams näe kehityksen olevan puolessa välissäkään. Pelissä ohjataan epäsuorasti komennoilla alunperin seitsemän kääpiön siirtokunnan perustamista. Ainoana tavoitteena on pitää hauskaa, sillä pelissä ei ole voittotavoitteita, ja hävitäkin voi vain menettämällä

kaikki kääpiöt. Pelin motto onkin ”Häviäminen on hauskaa!”. Pelin tyyli on lähimmillään kaupunginrakentelua. Pelaaja kääpiöitä rakentamaan tarpeellisia laitoksia kuten maatiloja, sepänpajoja ja tislamoja, ja yrittää pitää kaikki elossa ja tyytyväisinä. Erittäin monimutkaiset järjestelmät mallintavat peliä uskomattomalla tarkkuudella, olioiden ruumiin toiminnasta pelimaailman fysiikoihin, ja nämä vuorovaikutukset luovat pelistä joka pelikerralla erilaisen ja erikoisen. Maailman luomisen tarkkuutta näkee kuvassa 3. (Elliot 2014.)

Dwarf fortress on mielenkiintoinen esimerkki proseduraalisesta luomisesta. Alunperin Tarn ja Zach Adams aikoivat luoda ohjelman, joka luo proseduraalisesti fantasiatarinoita, mutta projekti paisui ja muuttui peliksi, joka luo valtavat määrät pelaajien tarinoita. Nämä tarinat ja niiden kerronta pelissä eivät varsinaisesti ole tarkoituksellisia, vaan ne syntyvät pelin monimutkaisten vuorovaikutusjärjestelmien ja tunnontarkan mallinnuksen seurauksena. Toisin kuin Rimworldissa, Dwarf Fortressissa ei ole mitään järjestelmää, joka yrittäisi luoda kerrontaa, mutta siitä huolimatta monimutkaiset järjestelmät ja niiden vuorovaikutukset luovat tapahtumia luonnonmukaisesti, jolloin pelaajan on helppoa tuntea, että peli kertoo tarinaa. (Elliot 2014.)

Erittäin tarkka pelimaailman mallinnus luo selviä suhteita asioiden välille, joten pelaajan tarvitsee vain seurata tapahtumia ja syntyvästä kerronnasta on helppo luoda tarina kerrottavaksi. Peliyhteisössä on monia tarinoita, joista kuuluisin on Boatmurdered –nimisen linnoituksen kohtalo. Siinä pelin tapahtumat pelaajien kertomana luovat tarinan, joka on useiden ihmisten mielenkiinnon herättäjänä itse peliä kohtaan. (Elliot 2014.)

5 OMAN PROJEKTIN KERRONTAJÄRJESTELMÄN LÄPIKÄYNTI

Osana tätä opinnäytetyötä on läpikäytyjen aiheiden soveltaminen oman proseduraalisen kerrontajärjestelmän suunnittelussa ja testaamisessa.

5.1 Projektin ja kerrontajärjestelmän kuvaus

Kerrontajärjestelmä tulee olemaan osa peliprojektia. Peliprojektia tehdään yhteistyössä Miika Vaetojan kanssa. Pelissä pelaaja ohjaa rikollisjoukkoa kaupungissa tavoitteenaan valloittaa kaupunginosa omalle joukolleen. Peli on jaettu taktiseen osuuteen, jossa pelaaja ohjaa yksittäisiä hahmoja vuoropohjaisessa taistelussa, ja strategiseen osuuteen, missä pelaaja hallitsee joukkionsa resursseja ja valloituksia. Kerrontajärjestelmää tullaan käyttämään strategisen osuuden sisällön luomisessa ja ohjaamisessa. Projektin tekemisessä käytetään Unity-pelimoottoria ja C#-ohjelmointikieltä.

5.1.1 Tavoite

Tavoitteena projektissa on luoda järjestelmä, joka ottaa huomioon pelaajan päätöksiä sekä pelitilanteita ja luo näiden vaatimusten pohjalta tapahtumia, jotka sopivat tilanteeseen ja ovat osa maailmaa. Suurin tavoite järjestelmälle ei ole tunnontarkka simulaatio, joka seuraisi pelin tapahtumia ja mallintaisi yksityiskohtia, joihin pelaajan päätökset vaikuttaisivat, vaan luoda harhan ja tyhjistä nyhjäisemisen voimalla vaikutelma vaikutusten päätöksestä, ja jatkaa luotua harhaa eteenpäin.

Mallia järjestelmään otetaan sekä videopelistä King of the Dragon Pass (KotDP), Valven The Director järjestelmästä sekä Minstrelistä. KotDP:ssa kerrontaa ei luoda proseduraalisesti, mutta tapahtumajärjestelmä on monipuolinen ja antaa pelaajalle mainiosti tunteen päätösten vaikutuksesta. Lisäksi se luo tunteen pelille tunnontarkasta mallinnuksesta, jota ei oikeasti ole olemassa. The Director ja

Valven Left 4 Dead:eissa käyttämä keskustelujärjestelmä antavat rakenteen ohjelmointipuolella. Keskustelujärjestelmän tilannekohtaista hakua on helppo soveltaa keskusteluista eriäviin tarkoituksiin, ja sen toimintaperiaatteet ovat hyvin dokumentoidut. Minstrel:stä otetaan mallia pääasiassa yksityiskohtaisuuden tehostamiseen.

5.1.2 Toimintasuunnitelma

Pohjana kerrontajärjestelmälle käytetään peliprojektia, jota tehdään projektikurssilla ja opinnäytetyöhön varattuna aikana.

5.2 Kerrontajärjestelmän läpikäynti

Järjestelmän ytimenä toimii EventManager-luokka, joka vuoron päättyessä arvioi, kannattaako sen tehdä vuorolle tapahtumaa. Arviointi tapahtuu yksinkertaisesti vertaamalla senhetkistä tapahtumatiheysmuuttujaa, jota tapahtumat saattavat vaihdella, ja viimeksi tapahtuneesta tapahtumasta kulunutta aikaa. Näistä saadaan todennäköisyys uudelle tapahtumalle. Mikäli on aika uudelle tapahtumalle, EventManager hakee tietopaketin PlayerData-luokasta, jossa säilötään mahdollisimman paljon muuttujia pelaajan tekemisistä.

Tietopaketti sisältää PlayerData:n tiedot dictionaryssä, jossa avaimena toimii muuttujan nimi ja sisältönä arvo. EventManager sisältää tapahtumapohjia, joilla on asetettuja vaatimuksia. Esimerkiksi asuintilan puutteesta valittava tapahtuma saattaa vaatia sen, että pelaajan henkilöstöresurssi on täynnä. Tapahtumapohjan valinta noudattaa samaa periaatetta kuin Valven Left 4 Dead:ssa käyttämä puhumisaiheen valintajärjestelmä. (Ruskin 2012.) Pohjista valitaan parhaiten

arvojen pohjalta tilanteeseen sopiva. Aina valitaan jokin tapahtumapohja, sillä tapahtumapohjalistassa on yleistapahtumia, joilla ei ole vaatimuksia.

Kun tapahtumapohja on valittu, aletaan pohjan tyhjiä muuttujia täyttää. Tyhjät muuttujat ovat tietyille tyypeille varattuja, kuten kohdemuuttuja saattaa vaatia toisen järjestön kohteeksi. Muuttujien täyte haetaan toistaiseksi yksinkertaisella switch-case järjestelmällä. Esimerkiksi `[[LowestResource]]` tekstissä korvataan pelaajan vähiten omaavalla resurssilla. Järjestelmää laajennettaessa voidaan muuttujien valintaan käyttää samanlaista järjestelmää kuin muuttujapohjan valitsemisessa.

Muuttujat ovat tapahtumapohjan tekstissä kaksoishakasulkeiden sisällä, esimerkiksi `[[PlayerFaction]]`. Nämä muuttujat etsitään tekstin lomasta ja täytetään vaaditunlaisella tekstillä. Osaa muuttujista ei välttämättä ole määritelty ennen niiden ensimmäistä esiintymiskertaa, kuten `[[EnemyFactionLeader]]`. Jos muuttujaa ei määritelty vielä, se luodaan ja lisätään `NarrativeManager`-luokkaan, joka säilöö käytettyjä muuttujia dictionaryyn myöhempiä viittauksia varten. Tapahtumilla on myös yksi tai useampi valittavissa oleva lopputulos, joka vaikuttavaa esimerkiksi pelaajan resursseihin tai suhteisiin muiden järjestöjen kanssa.

Tapahtumat saattavat myös johtaa taisteluihin, ja aikaisemmissa tapahtumissa esiintyneet hahmot voivat myös olla taisteluiden pääkohteena tai syynä. Laajentamalla seurattuja arvoja ja lisäämällä tietojen säilytystä esiintyneistä hahmoista voidaan järjestelmää laajentaa tulevaisuudessa antamaan luoduille hahmoille persoonallisuusominaisuuksia, jotka voivat vaikuttaa tapahtumiin.

5.3 Onnistumiset ja vastoinkäymiset

Rakennettu järjestelmä toimii hyvin perustasolla, mielenkiintoisten ja toisiinsa liittyvien tapahtumien luominen vaatii vielä lisää työtä. Luodulla järjestelmällä on silti hyvät mahdollisuudet saavuttaa tämä piste lisätyöllä, sillä sen rakennetta voidaan jatkaa hakemaan asiakkoitsemppaa tietoa tapahtumiin, esimerkiksi lisäämällä samanlaisen haun tapahtuman tietojen täyttöön käyttämällä NarrativeManagerin tietoja. Järjestelmän suurin rajoite on tarve valmistaa käytetyt tekstit käsin järjestelmälle käytettäväksi, mutta pitämällä osan kirjoituksista yleisluontoisina, voidaan niitä käyttää useammassa kohdassa vähentäen työmäärää.

Suurimmaksi ongelmaksi nousi ajan puute. Järjestelmän yhdistäminen peliin ei onnistunut, sillä peliprojekti ei ollut tarpeeksi pitkällä toiminnallisuudessaan jotta kerrontajärjestelmän olisi voinut yhdistää peliin. Kirjoitushetkellä järjestelmä toimii pelissä irrallisena osana, joka käyttää pelin muuttujia mutta ei vielä vaikuta pelimekaniikoihin niiden puutteen takia. Peliprojektin edetessä tarvittavaan pisteeseen on järjestelmän yhdistäminen helppoa rakenteensa ansiosta.

6 POHDINTA

Proseduraalinen sisällön luonti tarjoaa monia hyötyjä, kuten näennäisesti loputonta sisältöä peleihin tai jälleenpeluuarvoa. Lisäksi se voi säästää tekijöiltä paljon työaika oikein käytettynä. Näillä hyvillä puolilla on kuitenkin varjopuolensa, kuten luotu loputon sisältö saattaa olla merkityksettömän oloista sekä hengetöntä, ja jälleenpeluuarvo saattaakin olla vain näennäistä, sillä peli ei uudella pelikerralla olekaan merkityksellisesti muuttunut. Korjaamalla näitä ongelmia taas menettää työaikasäästöt, sillä proseduraalisten järjestelmien tekeminen ja perinpohjaisesti testaaminen vievät paljon aikaa, ja kuluva aika kasvaa kiihtyvästi monimutkaisuuden kasvaessa.

Onnistunut proseduraalinen sisältö on usein sivuosassa, kuten maailmankartan luonti Civilization –pelisarjan peleissä. Kun luotu sisältö ei ole yksi pelin pääkeskittymiskohteista, sen luominen ei vaadi niin paljoa työtä. Vaikka kartta on tärkeä osa Civilization:ia, harvaa kiinnostaa mikäli luotu kartta ei ole geologisesti ja ilmastotieteellisesti tarkka, kunhan pelilliset asiat ovat kunnossa. Luotu kartta on joka kerta erilainen ja tarjoaa vaihtelevaa pelikokemusta lisäämättä peliin tunnetta hengettömyydestä.

Toinen tapa lähestyä proseduraalista luomista onnistuneesti on siihen keskittyminen ja panostaminen. Tällöin proseduraalinen luominen ei enää ole vain työkalu jolla luodaan kenttiä tai tehtäviä ilman suunnittelijoiden työtä vaan se on keskeinen osa pelikokemusta. Kun luontiosuus nostetaan pelin pääkohteeksi, täytyy sen myös olla erittäin hyvin tehty, sillä normaalit ongelmat voivat helposti tuhota koko pelin. Jos pelin kaikki sisältö tuntuu itseään toistavalta, mielivaltaiselta ja merkityksettömältä ei menestystä ole tulemassa. Keskittämällä työaika ja vaivaa luontijärjestelmän tekemiseen ei saada hyötyä vähennyksestä työajasta, mutta on mahdollista saada lähes loputon mielenkiintoinen sisältö peliin.

Kerronta peleissä on usein oudossa tilassa. Jos kerronta ja tarina ovat sivuosassa, harva kiinnittää siihen huomiota. Pelin kylkeen lisätty juoni on usein vain löyhä syy

tehdä sitä, mitä pelissä ollaankaan tekemässä joten se ei ole erityisen mielenkiintoinen. Proseduraalista kerronnan luontia voi käyttää tälläisen juonen luomiseen, sillä helpoimmillaan tarvitsee vain kirjoittaa ”X nauraa meille! Mene potkimaan hänet Y kertaa kumoon!” ja korvata X ja Y asiaanliittyvillä muuttujilla. Tällä tasolla on silti helpompaa vain kirjoittaa syyt itse ja jättää proseduraalinen kerronta pois.

Pääkohteeksi nostettaessa taas proseduraalinen kerronta kohtaa omat ongelmansa. Mikä on hyvä tarina? Mikä on mielenkiintoista kerrontaa? Näihin kysymyksiin ei ole suoria vastauksia, ja epäsuorien suuntaviivojen ohjelmoiminen järjestelmään on vaikeaa eikä takaa mitään kerronnan laadusta. Toistaiseksi ei ole yhtään AAA-peliä, jossa proseduraalista narratiivia käytettäisiin pää- tai sivuosassa. The Elder Scrolls V: Skyrim käyttää proseduraalista luontia tekemään loputtomasti tehtäviä Radiant AI-järjestelmällään, mutta sitä syytetään juuri merkityksettömien tehtävien tekemisestä. Luodut tehtävät ovat lähes kokonaan irrallisia muusta pelimaailmasta. Jos tehtäväksi annetaan salamurha, putkahtaa tappokohteeksi esimerkiksi nimetön ”Seasoned Hunter”, joka ei ollut olemassa pelimaailmassa ennen tehtävän hyväksymistä. Järjestelmän yksinkertaisuus ja irrallisuus ei luo mielenkiintoisia tehtäviä ja ei kannusta pelaajaa keksimään kerrontaa tekemisistään.

Opinnäytetyön yhteydessä tehty järjestelmä ja muiden järjestelmien tutkiminen antaa tunteen, että hyvä proseduraalinen kerrontajärjestelmä on silti mahdollista luoda. Tarinapeleissä ei toistaiseksi uskalleta luottaa proseduraaliseen luomiseen, sillä tunnelman ja jännityksen luominen vaatii ennaltasuunnittelua ollakseen varmasti hyvää. Pelit, joissa tarina ei ole ehdoton pääasia mutta ei myöskään merkityksetön, voisivat hyötyä hyvin tehdystä kerrontajärjestelmästä. Monet nykypelit käyttävät ja luovat emergenttiä kerrontaa, joka syntyy pelimekaniikoiden

seurauksena, mutta tarkoituksellista kerronnan luomista ei esiinny monissakaan peleissä.

Mikäli proseduraalinen luominen peleissä jatkaa kasvamistaan, saattaa proseduraalinen kerronta ja tarina nousta samassa nosteessa yleisesti käytettäväksi osaksi peliä.

LÄHTEET

- Chirrick, T. 2014. The Procedural Plot in Detail. Saatavilla <https://www.youtube.com/watch?v=0nfp4pbjJ50> (Luettu 10.3.2015)
- de Boer, V. 2014. Narrative in games. Saatavilla <http://www.few.vu.nl/~vbr240/onderwijs/pim/Narrative%20in%20Games.pdf> (Luettu 7.2.2015)
- Dinicola, N. 2013. The Problem with Emergent Stories in Video Games. Saatavilla <http://www.popmatters.com/column/173580-the-problem-with-emergent-stories-in-video-games/> (Luettu 1.3.2015)
- Dyer, M. 2012. Linear Games and the Art of Control. Saatavilla <http://www.ign.com/articles/2012/04/24/linear-games-and-the-art-of-control> (Luettu 12.3.2015)
- Elliot, M. 2014. Into the deep: it's time to learn how to play Dwarf Fortress. Saatavilla <http://www.pcgamer.com/into-the-deep-its-time-to-learn-how-to-play-dwarf-fortress/> (Luettu 23.11.2015.)
- Fitzpatrick, J. 2014. The how-to geek guide to minecraft: Lesson 12: Creating Custom Minecraft Maps. Saatavilla <http://www.howtogeek.com/school/htg-guide-to-minecraft/lesson12/> (Luettu 12.11.2015)
- Franich, D. 2011. Videogames vs. Movies: Have games replaced films as the modern popular narrative medium? Saatavilla <http://www.ew.com/article/2011/08/18/movies-videogames-inception-red-dead-redemption> (Luettu 9.3.2015)
- Gillen, K. 2010. Just Die: Against "Real" Role-Playing Games. Saatavilla <http://www.rockpapershotgun.com/2010/04/08/just-die-against-real-role-playing-games/> (Luettu 12.3.2015)
- Hendriks, M., Meier S., van der Velden, J. & Iosup, A. 2012. Procedural Content Generation for Games: A Survey. Saatavilla http://www.st.ewi.tudelft.nl/~iosup/pcg-g-survey11tomccap_cr.pdf (Luettu 16.11.2015)
- Jenkins, H. 2011. Game design as narrative architecture. Saatavilla http://interactive.usc.edu/blog-old/wp-content/uploads/2011/01/Jenkins_Narrative_Architecture.pdf (Luettu 7.2.2015)
- Kastbjerg, E., Schedl, D., Togelius, J. & Yannakakis, G. 2011. What is Procedural Content Generation? Mario on the borderline. Saatavilla http://www.ccs.neu.edu/course/cs5150f14/readings/togelius_what.pdf (Luettu 16.11.2015)

- Kline, D. 2009. Bringing Interactive Storytelling to Industry: Designing a Reactive Narrative Encounter System. Saatavilla <https://dankline.files.wordpress.com/2009/10/bringing-interactive-story-to-industry-aiide-09-final2.pptx> (Luettu 4.11.2015)
- Koebel, A & LaTorra, S. 2012. Gamemastering. Saatavilla <http://www.dungeonworldsrd.com/gamemastering> (Luettu 27.5.2014)
- Koebel, A & Lumpkin, S. 2014. Being Everything Else – Ep 1: What is a GM? Saatavilla <https://www.youtube.com/watch?v=Zw17SbxcXrE> (Luettu 13.8.2014)
- Lambe, I. 2012. Procedural Content Generation: Thinking With Modules. Saatavilla http://www.gamasutra.com/view/feature/174311/procedural_content_generation.php (Luettu 16.11.2015)
- Lee, J. 2014. How Procedural Generation Took Over the Game Industry. Saatavilla <http://www.makeuseof.com/tag/procedural-generation-took-gaming-industry/> (Luettu 23.11.2015)
- Lee, T. 2013. Designing game narrative. Saatavilla <http://www.hitboxteam.com/designing-game-narrative> (Luettu 1.3.2015)
- Mateas, M., Tarse, B & Wardrip-Fruin, N. 2010. Minstrel Remixed: Procedurally Generating Stories. Saatavilla <https://www.aaai.org/ocs/index.php/AIIDE/AIIDE10/paper/download/2152/2567> (Luettu 15.11.2015)
- McMillen, E. 2011. The Binding of Isaac gameplay explained. Saatavilla <http://edmundmcmillen.blogspot.co.uk/2011/09/binding-of-isaac-gameplay-explained.html> (Luettu 22.10.2015)
- Meer, A. 2015. Early Impressions: Rimworld. Saatavilla <http://www.rockpapershotgun.com/2015/02/26/rimworld-alpha-review/> (Luettu 12.11.2015)
- Portnow, J. 2013. Extra Credits – The Illusion of Choice – How Balance Freedom and Scope. Saatavilla <https://www.youtube.com/watch?v=45PdtGDGhac> (Luettu 12.3.2015)
- Rose, M. 2012. 5 Tips for using procedurally-generated content in your game. Saatavilla http://www.gamasutra.com/view/news/181853/5_tips_for_using_procedurally_generated_content_in_your_game.php (Luettu 16.11.2015)
- Ruskin, E. 2012. Rule Databases for Contextual Dialog and Game Logic or.. How To Make Writers Even More Awesome. Saatavilla <http://www.gdcvault.com/play/1015317/AI-driven-Dynamic-Dialog-through> (Luettu 6.2.2015)

- Short, T. 2014. Level Design in Procedural Generation. Saatavilla http://www.gamasutra.com/blogs/TanyaXShort/20140204/209176/Level_Design_in_Procedural_Generation.php (Luettu 5.6.2014)
- Smith, G. 2014. "Procedural Everything": Playing Procedural Content Generation in Spore. Saatavilla <http://sokath.com/main/files/1/smith-wellplayed14.pdf> (Luettu 24.11.2015)
- Sutherland, J. 2005. What Every Game Developer Needs to Know about Story. Saatavilla http://www.gamasutra.com/view/feature/130770/what_every_game_developer_needs_to_php (Luettu 13.11.2015)
- Sylvester, T. 2014. Rimworld. Saatavilla <http://rimworldgame.com/> (Luettu 4.5.2015)
- Thomas, I. 2014. Procedural Narrative: The Experience Director: Saatavilla <http://wingedmonkeys.co.uk/procedural-narrative-the-experience-director/> (Luettu 14.11.2015)
- Walker, J. 2009. Left 4 Dead 2: Exclusive RPS Hands-On Preview. Saatavilla <http://www.rockpapershotgun.com/2009/06/01/left-4-dead-2-exclusive-rps-preview/> (Luettu 8.3.2015)
- Young, S. 2013. Procedural Stories. Saatavilla <http://www.escapistmagazine.com/articles/view/video-games/columns/experienced-points/10168-Procedural-Stories> (Luettu 4.3.2015)