

Pekka Mäntysaari

**Digitaalisten pelien dialogien kehittäminen automaattisen
tekstinluonnin avulla**

Tietotekniikan kandidaatintutkielma

21. joulukuuta 2022

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Pekka Mäntysaari

Yhteystiedot: pekka.j.mantysaari@student.jyu.fi

Ohjaaja: Tytti Saksa

Työn nimi: Digitaalisten pelien dialogien kehittäminen automaattisen tekstinluonnin avulla

Title in English: Developing dialogues of digital games through automatic text generation

Työ: Kandidaatintutkielma

Opintosuunta: Tietotekniikan

Sivumäärä: 22+0

Tiivistelmä: Tämän kandidaatintutkielman tarkoitus oli selvittää voidaanko digitaalisten narratiivisten pelien dialogeja kehittää automaattisen tekstinluonnin avulla. Kirjallisuuskatsauksessa tarkasteltiin pelikehitykseen suunnattuja automaattisen tekstinluonnin menetelmiä. Tarkasteltujen menetelmien pohjalta käsiteltiin automaattisen tekstinluonnin hyötyjä, dialogien varioimisen vaikutusta immersioon, ja eri menetelmien käyttöön liittyviä haasteita.

Avainsanat: Automaattinen tekstinluonti, dialogit, digitaaliset pelit

Abstract: The purpose of this bachelor's thesis was to find out whether dialogues of digital narrative games can be developed using automatic text generation. In the literature review, automatic text generation methods aimed at game development were examined. Based on the reviewed methods, the benefits of automatic text generation, the impact of varying dialogues on immersion, and the challenges associated with the use of different methods were discussed.

Keywords: Automatic text generation, dialogues, digital games

Sisällys

1	JOHDANTO	1
2	TEOREETTINEN TAUSTA	3
	2.1 Digitaalisten narratiivisten pelien dialogit	3
	2.2 Immersion merkitys peleissä.....	4
	2.3 Kielioppeihin perustuvat menetelmät tekstintuottamisessa	5
3	DIALOGIEN TUOTTAMISEN MENETELMÄT TUTKIMUSKIRJALLISUU- DESSA	7
	3.1 Tekstintuottamisen menetelmien käyttäminen pelikehityksessä	7
	3.2 Julkaistuja menetelmiä	8
	3.2.1 Automaattisen tekstinluonnin menetelmien tausta	8
	3.2.2 Tracery	8
	3.2.3 Expressionist	9
	3.2.4 Dialogic	10
	3.2.5 Muita menetelmiä.....	11
4	NARRATIIVISTEN PELIEN DIALOGIPERUSTAISEN IMMERSIIVISYYDEN KEHITTÄMINEN TEKSTINLUONNIN AVULLA.....	13
	4.1 Dialogien varioiminen automaattisen tekstinluonnin avulla	13
	4.2 Automaattisen tekstinluonnin vaikutus immersioon.....	14
	4.3 Automaattisesti tuotettujen dialogien käyttämisen haasteet.....	14
5	YHTEENVETO.....	16
	LÄHTEET	18

1 Johdanto

Automaattinen tekstintuotto, tai luonnollisen kielen tuottaminen, on eräs luonnollisen kielen käsittelyn menetelmä. Usein luonnollisen kielen käsittelyllä pyritään erottamaan ihmisen käyttämästä kielestä tietokoneen ymmärtämä data, ja tuottamaan tietokoneen datasta ihmisen ymmärtämää kieltä. Automaattista tekstin tuottamista voidaan toteuttaa useilla eri menetelmillä, joita ovat esimerkiksi tekoälyyn, koneoppimiseen, kontekstivapaisiin kielioppeihin ja Markovin malleihin perustuvat menetelmät.

Eräs automaattisen tekstintuottamisen sovellusalue on digitaaliset pelit. Tarinavetoisissa digitaalisissa peleissä dialogit ovat keskeinen osa pelien narratiivin kehitystä ja vaativat siten paljon huomiota pelikehityksessä. Automaattisen tekstin tuottamisen hyödyntäminen sisältää lukuisia haasteita, etenkin niissä käyttötapauksissa, joissa ohjelman antaman tulosteen on sisällettävä varmuudella määrättyjä avainasioita. Lisäksi kehittäjän on voitava luottaa siihen, että automaattisesti tuotettu teksti ei sisällä tahattomasti sopimatonta sisältöä. Myös yksinkertaisuus tai toisteisuus dialogissa voi murtaa illuusion elävästä pelimaailmasta. Eräs ratkaisu toisteisuuden vähentämiseksi on luoda proseduraalisesti dialogit pelinkehittäjän määrittämien sääntöjen sekä syöttämän datan pohjalta. Kuitenkin, dialogien luominen proseduraalisesti vaatii usein teknistä osaamista, jota ei dialogien kirjoittajilla välttämättä ole.

Dialogien tarkoituksena on usein narratiivin kehityksen ohella tuoda peliin sisältöä ja opastaa pelaajaa toimimaan pelin suunnittelijan tarkoittamalla tavalla. Immersion eli pelin maailmaan uppoutumisen ylläpitäminen vaatii peliltä syvyyttä, eli pelaajan ei tulisi kohdata tilanteita, joissa pelin rakenteet tai elementit muistuttavat staattisia lavasteita. Immersiota voidaan parantaa kehittämällä dialogeja.

Tässä kandidaatintutkielmassa tutkitaan narratiivisten digitaalisten pelien dialogien kehittämistä automaattisen tekstinluonnin keinoin. Tutkimuksen tavoitteena on tarkastella kirjallisuuden avulla, kuinka automaattisen tekstinluonnin keinoja on pyritty käyttämään pelien dialogien varioimiseen. Narratiivisten digitaalisten pelien sisältämien dialogien luomisprosessi on valtava urakka ja luonnollisten kielten käsittelyn menetelmien hyödyntäminen aputyökätluna vaatii paljon erityisosaamista (Ryan, Mateas ja Wardrip-Fruin 2016). Kirjallisuuskar-

toitus aiheesta helpottaa soveltamaan karttunutta tutkimustietoa.

Pelien dialogien luomiseen keskittyvää tutkimusta on tehty pääasiassa kahdesta näkökulmasta, kehittämällä menetelmiä luomaan kokonaisia dialogeja luonnollisen kielen käsittelymenetelmillä tai parantamalla dialogien luomisprosessia (Kerr ja Szafron 2009). Pelien dialogien luomiseen ja kehittämiseen keskittyvä tutkimus hyödyntää usein teknologioiden yhdistelmiä luoduissa järjestelmissä. Esimerkki teknologioiden yhdistelmästä on Expressio-nist, jossa hyödynnetään kontekstivapaita kielioppeja yhdessä kehittäjän antaman metadatan kanssa (Ryan ym. 2016). Automaattisen tekstintuottamisen laajasta menetelmävalikoimasta huolimatta, menetelmien käytöstä ei löydy kattavasti esimerkkejä interaktiivisten tarinoiden maailmasta (Ryan ym. 2016).

Tutkielmassa pyritään selvittämään, voidaanko narratiivisissa digitaalisissa peleissä pelielementteinä oleviin hahmoihin liitettyjä dialogeja olennaisesti kehittää lisäämällä dialogien variaatiota automaattisen tekstinluonnin avulla. Tarkastelun pääasialliseksi kohteeksi valitaan sellaiset dialogien kehittämisen keinot, joiden voidaan katsoa lisäävän pelikokemuksen immersiiivisyyttä. Tällöin vertailtaessa menetelmiä painotus on pelielementtien syvyyden lisäämisessä, ei pelaajan ohjausviestien tehostamisessa.

Tutkimuksessa rajoitutaan tarkastelemaan testin tuottamista menetelmillä, jotka tuottavat varioituja dialogeja sääntöpohjaisesti ja aihealuespesifisti. Erityisesti tutkimuksesta rajataan ulkopuolelle menetelmät, joissa luodaan täysin automaattisesti dialogeja tekoälyn ja koneoppimisen menetelmillä. Kriteerinä tarkasteltaville menetelmille pidetään mahdollisimman suurta kontrollia ja tietoisuutta tuotettujen tekstien sisällöstä.

2 Teoreettinen tausta

Tässä luvussa käsitellään automaattisen tekstinluonnin taustaa. Aluksi käsitellään pelien dialogien tutkimuksen taustaa sekä selvitetään pelien immersion merkitystä pelikokemuksessa. Tämän jälkeen esitellään kielioppien teoriaa tekstinluonnin kontekstissa.

2.1 Digitaalisten narratiivisten pelien dialogit

Tässä alaluvussa esitellään lyhyesti pelien dialogien tutkimusta ja historiaa. Aluksi määritellään tälle tutkimukselle keskeisiä käsitteitä, jonka jälkeen käsitellään dialogijärjestelmien toteuttamista sekä tietokoneroolipelien erityistä asemaa tutkimuksessa.

Dansky (2007) määrittelee narratiivin tarkoittavan niitä menetelmiä, joilla tarinalliset elementit kommunikoidaan yleisölle. Samaa määritelmää käytetään myös tässä tutkimuksessa. Narratiiviset pelit puolestaan rajataan tässä tutkimuksessa tarkoittamaan sellaisia peliteoksia, joissa narratiiviset elementit ovat elimellinen osa peliä. Siten tässä tutkimuksessa narratiivisiksi peleiksi lasketaan sellaiset pelit, joissa pelikokemus olennaisesti muuttuu, jos narratiiviset elementit poistetaan. Myös pelikokemus itsessään voi muodostaa narratiivisen kokemuksen ilman erillisiä narratiivisia elementtejä ja usein paljonkin dialogeja sisältävän pelin pelaaminen on mahdollista ymmärtämättä lainkaan dialogien sisältöjä, mutta tällaiset tapaukset rajataan tämän tutkimuksen ulkopuolelle.

Peleissä tapahtuva dialogi voi olla useiden eri toimijoiden välistä kommunikaatiota. Usein pelien dialogit ovat pelaajan ja pelissä olevan ei-pelaajahahmon välillä tapahtuvaa tiedonvälitystä. Dialogit voivat tapahtua myös pelaajan ja pelin välillä, pelaajan ja pelaajahahmon välillä, tai kahden ei-pelaajahahmon välillä (Brusk ja Björk 2009). Se, keiden toimijoiden välillä pelissä olevat dialogit käydään, vaikuttaa pelin teknisiin ja narratiivisiin sisältöihin. Esimerkiksi monen ei-pelaajahahmon välillä käytävällä dialogilla on tyypillisesti narratiivinen funktio, ja toisaalta pelaajan ja pelin välinen dialogi tapahtuu tyypillisesti peliteknisistä syistä.

Dialogeja voidaan vertailla ja tarkastella myös sen perusteella, ovatko ne diegeettisesti kon-

sistentteja, eli säilykö vuorovaikutuksessa aihepiiri tai pelin sisäinen maailma (Brusk ja Björk 2009). Esimerkkinä ei-diegeettisesti konsistentista vuorovaikutuksesta voisi olla se, että dialogijärjestelmä ilmoittaa pelaajalle peliteknisistä asioista jonkinlaisella metatekstillä, jolloin tiedonvaihto siirtyy pois pelin sisäisestä todellisuudesta.

Pelit voivat toteuttaa dialogit eriävillä tavoilla. Tyypillinen esimerkki dialogijärjestelmästä on vuorotellen etenevä ja pelaajan aloitteesta dialogeja aloittava järjestelmä, jossa voi olla kontekstista riippuvia dialogeja (Brusk ja Björk 2009). Myös muunlaiset järjestelmät ovat mahdollisia. Peli voi esimerkiksi käyttää vaihtelevan aloitteen dialogijärjestelmää käynnistämään pelinsisäiset dialogit pelaajan kanssa riippumatta pelaajan toimista (Brusk ja Björk 2009). Moderneissa narratiivisissa peleissä dialogit muodostavat usein puumaisen rakenteen, jossa pelaaja voi valinnoillaan edetä ja siten saada eteensä valintoihinsa perustuvia dialogisisältöjä.

Erityisesti tietokoneroolipelit ovat vahvasti narratiivisia elementtejä sisältäviä pelejä. Tämän lisäksi tietokoneroolipelit sisältävät usein suuria määriä dialogeja. Näistä piirteistä johtuen tietokoneroolipelit ovat erityinen sovelluskohde tutkimuksen aihepiirille. Ensimmäiset tietokoneella pelattavat roolipelit olivat tekstisyötteillä ohjattavia seikkailuja, joissa perinteisistä pöytäroolipeleistä poiketen pelinjohtajan rooli oli siirtynyt tietokoneelle. Tämänkaltaisissa peleissä pelaajan syöttämä teksti oli keino vuorovaikuttaa pelin sisäisen maailman kanssa, ja pelin muodostamat tekstitulosteet olivat pääasiallinen keino välittää pelaajalle pelin sisältö (Mäyrä 2017). Uudemmissa tietokoneroolipeleissä dialogeja käytetään harvemmin pelin tilan muuttamiseen, mutta dialogien narratiiviset tehtävät ovat lisääntyneet. Esimerkiksi suosituksissa Mass Effect-pelissä (BioWare 2008) dialogit käsittelevät pelaajan romanttisia suhteita ja taustoja (Brusk ja Björk 2009).

2.2 Immersion merkitys peleissä

Tässä alaluvussa selvitetään immersion käsitettä ja kuvataan immersion vaikuttavia tekijöitä. Lisäksi tarkastellaan miten dialogit voivat vaikuttaa pelikokemuksen immersion.

Immersion käsitettä käytetään yleisesti etenkin digitaalisten pelien tutkimuksessa kuvaamaan positiivista pelaamiseen liittyvää kokemusta, jossa pelaajan ajantaju katoaa, keskit-

tyminen lisääntyy ja pelaajan tietoisuus itsestä vähenee (Michailidis, Balaguer-Ballester ja He 2018). Immersio muistuttaa flow-tilaa. Flow ja immersio saattavatkin olla yksi ja sama käsite eri puolilta tarkasteltuna (Michailidis, Balaguer-Ballester ja He 2018).

Narratiivisissa peleissä immersio on tärkeä ja uppoutuminen pelin tarinaan voi olla pelaajalle tärkeä syy pelata. Tarinallisissa peleissä tarinasta välittäminen liittyy immersioon ja pelikokemuksen eheyteen. Tarinallisiin peleihin liittyvään immersioon kuuluu olennaisesti pelin maailman ja hahmojen uskottavuus, sekä vapaaehtoinen epäuskon viivästyttäminen (engl. suspension of disbelief).

Dialogeilla on suuri merkitys immersion kannalta. Dialogit muodostavat pelin maailmaa ja kuvaavat sen hahmojen eläväisyyttä. Dialogit eivät tue immersiota jos ne eivät ole pelin diegeettisen kehyksen sisällä, eli sisältävät elementtejä pelin sisäisen todellisuuden ulkopuolelta. Vaihtoehtoisesti tekstien toistuminen voi paljastaa pelaajalle kohdattujen elementtien pinnallisuuden. Pelaaja voi esimerkiksi vuorovaikuttaa pelimaailmassa pelimaailman sisäisen viestitekstin kanssa yksipuoleisesti siten, että immersio säilyy. Kuitenkin immersio kärsii, jos pelimaailman ajattelevia olentoja kuvaavat hahmot muistuttavat ilmoitustaulua tai kiiveen kaiverrettua tekstiä, antaen vain nopeasti toistuvan tekstikatkelman vuorovaikutuksen jatkuessa.

2.3 Kielioppeihin perustuvat menetelmät tekstintuottamisessa

Tässä aluvuossa esitellään lyhyesti kontekstivapaiden kielioppien historiaa ja keskeisiä periaatteita.

Muodolliset kieliopit koostuvat päätemerkkien (engl. terminal symbol) joukosta, välikemerkkien (engl. nonterminal symbol) joukosta, sääntöjoukosta ja alkumerkistä (Merikoski, Virtanen ja Koivisto 2004). Säännöstä, joka liittyy merkin päätemerkkiin, välikemerkkiin tai niiden yhdistelmään, käytetään tässä tutkimuksessa sanaa produktio. Kontekstivapaassa kieliopissa produktiot määrittävät symbolien uudelleenkirjoituksen muodossa $A \rightarrow \gamma$, missä γ on merkkijono, joka voi muodostua tyhjästä jonosta tai pääte- ja välikemerkkien vapaasta yhdistelmästä (Merikoski, Virtanen ja Koivisto 2004). Merkkijonojen johtaminen tapahtuu siis aloittamalla alkumerkistä ja korvaamalla merkit yksi kerrallaan merkkijonosta produk-

tioiden mukaan, kunnes lopputuloksena on merkkijono, jossa on vain päätemerkkejä.

Chomsky (1956) loi generatiivisille muodollisille kieliopeille hierarkian, jossa kielityypit määritellään formaalisti. Chomskyn hierarkiassa on neljä kielityyppiä 0-3 siten, että tyyppi 0 on vähiten rajoitettu ja tyyppillä 3 on tarkimmat rajoitteet (Merikoski, Virtanen ja Koivisto 2004). Kontekstivapaat kieliopit eli lauserakennekieliopit (engl. phase-structure grammar) kuuluvat Chomskyn hierarkiassa tyyppiin 2 (Merikoski, Virtanen ja Koivisto 2004). Nimitys kontekstivapaa (eng. context-free) tai yhteydetön kielioppi, tulee siitä, että produktiossa tuotettu merkkijono ei riipu A:n ympärillä olevista symboleista (Merikoski, Virtanen ja Koivisto 2004).

3 Dialogien tuottamisen menetelmät tutkimuskirjallisuudessa

Tässä luvussa esitellään tutkimuskirjallisuudessa esitettyjä automaattisen tekstintuottamisen menetelmiä, joiden voidaan katsoa soveltuvan digitaalisten pelien dialogien kehittämiseen. Ensiksi tarkastellaan niitä tavoitteita, joita tekstintuottamisen hyödyntämisellä pyritään saavuttamaan. Lopuksi esitellään kirjallisuudessa olevia pelikehityksessä käytettyjä tekstintuottamisen menetelmiä.

3.1 Tekstintuottamisen menetelmien käyttäminen pelikehityksessä

Tämän alaluvun tavoitteena on tarkastella eri syitä käyttää automaattista tekstintuottoa pelinkehityksessä. Automaattinen tekstintuottaminen vaatii pelien kehittäjiltä lisäpanostusta uusien työkalujen käyttöönottamisessa ja siten menetelmistä saatavien hyötyjen täytyy olla riittävän suuret.

Pelien dialogien kirjoittaminen on usein valtavan suuri urakka ja sen tekemiseen voidaan tarvita useita kirjoittajia ja paljon aikaa. Esimerkiksi paljon haarautuvia dialogeja sisältävä tietokoneroolipeli Mass effect (Bioware 2008) sisälsi arviolta 28000 riviä dialogia (Pressey ja Walters 2008). Dialogien määrän voidaan perustellusti ajatella vielä kasvavan entisestään, jos jokaisessa tilanteessa dialogit voivat olla erilaiset riippuen esimerkiksi pelin hahmojen tunnetiloista tai pelin aiemmista tapahtumista.

Automaattista tekstintuottamista voidaan pitää mahdollisena työkaluna pelinkehityksen aikana, tai se voidaan ottaa tekniikaksi tuottamaan sisältöä peliin proseduraalisesti. Yksinkertaisimmassa käyttötapauksessa automaattista tekstintuottoa hyödynnetään pienentämään sitä työmäärää, joka muodostuu peliin tarvittavien repliikkien kirjoittamisesta. Tällöin voidaan esimerkiksi luoda, kirjoittajan ohjaamana, nopeasti tekstikatkelmia, jotka kehittäjän hyväksymisen jälkeen voidaan liittää valmiiseen peliin. Tekstin proseduraalisessa luonnissa teksteistä voidaan generoida annettujen sääntöjen mukaisia variaatioita, ja siten vähentää esitettyjen dialogien toisteisuutta.

3.2 Julkaistuja menetelmiä

Tässä alaluvussa tarkastellaan automaattisen tekstinluonnin taustaa sekä esitellään valikoituja tekstintuottamisen menetelmiä kirjallisuudesta. Menetelmät valittiin tutkimuskirjallisuudesta sen perustella, mainittiinko pelikehitys niiden sovelluskohteena. Tarkempaan esittelyyn valittiin automaattisen tekstinluonnin työkalut Tracery, Expressionist ja Dialogic. Alaluvun päätteeksi tarkastellaan myös muita kirjallisuudessa esiintyviä menetelmiä.

3.2.1 Automaattisen tekstinluonnin menetelmien tausta

Eräs ensimmäisistä automaattisia dialogeja luovista järjestelmistä oli keskustelullinen ohjelma Eliza, johon toteutettu kuvitteellinen terapeutti muodosti käyttäjän antamista syötteistä uusia tekstikatkelmia luoden illuusion vuorovaikutuksesta (Weizenbaum 1966). Myöhemmin automaattista tekstinluontia on tutkittu muun muassa vuorovaikutteisen fiktion kehittämisen keinona. Osaltaan automaattisen tekstintuotannon tutkimus nojaa myös tarinakielioppien tutkimukseen. Tälle taustalle rakentuvat myös seuraavassa tarkasteltavat modernimmat pelien dialogien automaattisen tekstinluonnin menetelmät.

3.2.2 Tracery

Tracery (Compton, Kybartas ja Mateas 2015) on yksi kirjallisuudessa esiintyvistä automaattisen tekstin tuottamisen menetelmistä, joka sopii tämän tutkimuksen kysymyksenasettelun mukaisesti vaatimuksiin. Tracery on kevyt tekstintuottamisen työkalu, joka on suunniteltu helposti käyttöönotettavaksi tekniikaksi monenlaisiin tekstintuottamisen tarpeisiin. Traceryä onkin käytetty vuorovaikutteisen fiktion ja pelikehityksen lisäksi esimerkiksi twitter-bottien luomiseen (Compton, Kybartas ja Mateas 2015).

Tracery perustuu muodollisten kielioppien käyttämisen tekstinmuodostuksessa. Teksti muodostetaan rekursiivisesti kuvaamalla määritellyt symbolit joukkoon uudelleenkirjoitussääntöjä. Traceryn käyttäjä määrittelee joko raakatekstiä, johon voi sisältyä rekursiivisesti laajennettavia symboleja, tai symboleja ja niihin liitetyn joukon vaihtoehtoisia uudelleenkirjoitussääntöjä (Compton, Kybartas ja Mateas 2015). Tässä järjestelmässä käyttäjä voi määritellä uudelleenkirjoitussäännössä aina uusia viittauksia seuraavaan symboliin ja siten kieliopista

voi tulla monimutkaisia ja syviä.

Tracery tarjoaa myös funktioita, joilla luotua tekstiä voidaan muokata vastaamaan paremmin luonnollisen kielen tekstin tarpeita. Tätä toiminnallisuutta voidaan tarvita varmistamaan luodun tekstin kieliopillinen oikeellisuus, esimerkiksi kun tarvitaan oikeat taivutusmuodot sanoista. Traceryn käyttäjiä rohkaistaan myös luomaan omia määritteitä vastaamaan paremmin käyttäjien erilaisia tarpeita (Compton, Kybartas ja Mateas 2015). Esimerkkejä sisäänrakennetuista funktioista, jotka voidaan asettaa tulemaan voimaan symbolin laajennuksen jälkeen ovat sanan monikkoon muuttava `.pluralize` ja alkukirjaimen isoksi muuttava `.capitalize` (Compton, Kybartas ja Mateas 2015).

Monimutkaisissa käyttötapauksissa on tyypillistä vaatia, että jo tekstissä esiintyviä elementtejä, kuten nimiä, halutaan pitää tekstissä yhtenäisenä. Tähän vaatimukseen Traceryssä on mahdollista käyttää pinorakennetta ja sen push- ja pop-toimintoja. Tämä tapahtuu siten, että Tracery mahdollistaa ajonaikaisen kieliopin ylikirjoituksen siten, että symbolin säännöt ovatkin pino sääntöjä, joista ainoastaan päällimmäinen on käytössä (Compton, Kybartas ja Mateas 2015).

Compton, Kybartas ja Mateas (2015) ovat myös pyrkineet luomaan Tracerystä modulaarisen työkalun, jonka muokkaaminen eri käyttäjien tarpeisiin on tehty mahdolliseksi. Sen pääasiallinen komponentti on kirjasto, joka laajentaa tekstiä sääntöpohjaisesti hyödyntäen sille annettua kielioppia. Rakenteellisesti kielioppi tallennetaan JSON-muodossa (Compton, Kybartas ja Mateas 2015).

Tracery on menetelmänä kevyt ja sen soveltaminen pelikehityksessä voi riippua sen kehittyneemmän käytön ominaisuuksista. Tällöinkin sen ominaisuudet voivat osoittautua pelikehityksessä sopiviksi lähinnä luomaan nopeita variaatioita annetuista valmiista teksteistä. Katavammat vaatimukset vaativat monimutkaisempia menetelmiä, kuten esimerkiksi työkalun nimeltä Expressionist (Ryan ym. 2016).

3.2.3 Expressionist

Kuten Tracery, myös Expressionist hyödyntää kontekstivapaita kielioppeja automaattisessa tekstinluonnissa (Ryan ym. 2016). Expressionist on kehitetty monimutkaisempiin sovelluk-

siin kuin Tracery (Ryan ym. 2016) ja se sisältää kielioppien lisäksi metadatan lisäämisen luotuihin sisältöihin. Metadatan käyttö mahdollistaa välikemerkkien kuvailemisen ja jaottelun, joka puolestaan mahdollistaa laajemman tuotetun tekstin hallinnan (Ryan ym. 2016). Expressionist eroaa Tracerystä myös siinä, että Expressionistissa produktiolle voidaan määrittää todennäköisyydet eli Expressionist luo probabilistisia kontekstivapaita kielioppeja (Ryan ym. 2016). Traceryn tapaan Expressionist käsittelee kontekstivapaita kielioppeja JSON-tiedostoina (Ryan ym. 2016).

Expressionist on suunniteltu mahdollistamaan tekstintuottamisen ajonaikaisesti esimerkiksi peleissä (Ryan ym. 2016). Expressionistilla luotujen kontekstivapaiden kielioppien ajon aikainen laajentaminen vaatii kuitenkin sovelluskohtaisen laajennusmoduulin kirjoittamisen peliin. Expressionistia hyödyntävässä projektissa tarvitaan siis järjestelmä, joka osaa tulkita Expressionistin tuottamia JSON-tiedostoja ja palauttaa kehittäjän tarpeita vaativia tekstikatkelmia laajentamalla kielioppeja.

3.2.4 Dialogic

Dialogic jatkaa aiempia pyrkimyksiä luoda käyttäjäystävällinen tekstintuottamisen työkalu pelien vuorovaikutteisten tekstien kirjoittamiseen (Howe 2020). Dialogic on järjestelmä, jossa yhdistyvät skriptikieli, ajoympäristö ja verkkotyökalut. Sen tavoitteena on mahdollistaa tekstintuottamisen hyödyntäminen luovassa prosessissa, jopa ilman teknistä osaamista (Howe 2020). Kuten Tracery ja Expressionist, myös Dialogic tukee kontekstivapaiden kielioppien käyttöä tekstin tuottamisessa.

Dialogicissa voidaan lisätä metadataa tekstikatkelmiin kuten Expressionistissa, mutta määritteiden lisäksi Dialogic sisältää sumean rajoiteperustaisen haun määritteille (Howe 2020). Haku mahdollistaa tekstikatkelmien esittämisen kontekstiriippuvaisesti ja annettuihin ehtoihin perustuen. Määritteisiin liitetty metadata voi sisältää esimerkiksi lukuarvoja, joihin perustuen haetut tekstikatkelmat voidaan asettaa järjestykseen ja palauttaa parhaiten ehtoihin sopiva katkelma. Dialogiciin on lisäksi toteutettu haettavien tekstien toiston havaitseminen ja välttäminen. Järjestelmä pisteyttää noudettavat katkelmat vertaamalla niitä aiemmin esitettyihin, ja laskemalla niistä pienimmän editointietäisyyden (Howe 2020). Tekstikatkelmat,

jotka ovat liian samankaltaiset saavat alhaisemmat pisteet, jolloin haussa voidaan valita toinen tekstikatkelma samankaltaisen tilalle. Dialogicissa metadatta voidaan käyttää myös dialogien keskeytysten ja palautusten toteuttamiseen (Howe 2020). Toteuttaessa dialogin keskeytystä ja palautusta, tekstiin voidaan lisätä luonnollisia ilmauksia osoittamaan siirtymiä tai keskustelun jatkumista.

Kuten Tracery, myös Dialogic tarjoaa funktioita sanojen käsittelyyn, kuten esimerkiksi sanan monikoinnin `.pluralize()`-funktioilla tai artikkelin asettamisen `.articlize()`-funktioilla (Howe 2020). Käyttäjä voi myös laajentaa omilla funktioilla sisäänrakennettujen funktioiden koelmaa. Tämä mahdollistaa funktioiden hyödyntämisen myös muiden kielten kuin englannin kohdalla.

Dialogic on mahdollista liittää pelimoottoriin. Esimerkiksi TendAR-projektissa (2018) sitä on käytetty yhdessä Unity-pelimoottorin kanssa (Howe 2020).

3.2.5 Muita menetelmiä

Tässä alaluvussa esitellään lyhyesti muita kirjallisuudessa esiintyviä automaattisen tekstin tuottamisen menetelmiä.

Yksinkertaisimmillaan tekstinluontia voidaan tehdä käyttämällä järjestelmää, jossa varioitavasta tekstistä muodostetaan templaatti (engl. *template*), johon korvataan muuttujia halutuilla arvoilla. Tämän lähestymistavan ilmaisuvoima on heikompi kuin kontekstivapaisiin kielioppiin perustuvissa menetelmissä (Horswill 2020). Kontekstivapaita kielioppejan hyödyntävien menetelmien voidaan kuitenkin ajatella olevan myös pohjimmiltaan yhden tai useamman muuttujan rekursiiviseen korvaamiseen perustuva menetelmä. Aiemmin tässä luvussa kuvatuissa järjestelmissä on mahdollista luoda tekstiä templaatteja vastaavalla tavalla.

Blabbeur on Expressionistin kaltainen tekstinluontijärjestelmä, joka on kehitetty yhdistettäväksi Unity-pelimoottoriin (Lessard ja Kybartas 2021). Keskeinen ero Expressionistin ja Blabbeurin välillä on, että Blabbeur on toteutettu Unity-pelimoottorin käyttämällä C#-kielellä, kun Expressionist on riippuvainen Python-kielen funktioista (Lessard ja Kybartas 2021).

Bronco on Turing-täydellinen tekstinluontimenetelmä, joka on saanut inspiraationsa aiem-

pien menetelmien kuten Tracery ja Expressionist muodollisesta määrittelystä (Knochelmann ja Cardona-Rivera 2022). Broncosta on pyritty kehittämään menetelmä, jossa yhdistyvät helppo käyttöönotettavuus ja suuri kontrolli tuotettuihin teksteihin (Knochelmann ja Cardona-Rivera 2022). Muodollisesti Bronco toteuttaa attribuuttikieliopin kaltaisen järjestelmän, joka on toteutettu kokonaisuudessaan funktionkaltaisilla oliolla (Knochelmann ja Cardona-Rivera 2022).

4 Narratiivisten pelien dialogiperustaisen immersiivisyyden kehittäminen tekstinluonnin avulla

Tämä luku käsittelee pelien immersiivisyyden kehittämistä automaattisen tekstinluonnin avulla. Ensin tarkastellaan tapoja, joilla pelien dialogeja voidaan kirjoittaa generatiivisilla menetelmillä. Seuraavaksi käsitellään automaattisen tekstinluonnin merkitystä pelin immersiivisyyden kannalta. Lopuksi arvioidaan automaattisesti tuotettujen tekstien käytön haasteita.

4.1 Dialogien varioiminen automaattisen tekstinluonnin avulla

Tämä alaluku tarkastelee mahdollisuutta lisätä pelien dialogeihin variointia tai kontekstista riippuvaa vaihtelua automaattisen tekstinluonnin avulla. Automaattisen tekstinluonnin soveltuvuutta dialogien varioimiseen selvitetään valittujen kirjallisuudessa esiintyvien käyttötapauksen kautta.

Palmas, Raith ja Klinker (2020) tutkivat automaattisen tekstinluonnin käyttöä vuorovaikutteisen dialogin luomiseksi pelillistettyyn oppimissovellukseen. Tutkimuksessa käytettiin automaattiseen tekstinluontiin Expressionist-työkalua (Ryan ym. 2016) ja tekstinluonnin tukena tunteiden analyysi -järjestelmää (eng. sentiment analysis) (Palmas, Raith ja Klinker 2020). Tutkimuksessa luotiin asetelma, jossa pelaaja yrittää myydä ei-pelaajahahmolle uutta puhelinta. Pelin eteneminen tapahtui pelaajan ja hahmon välisen dialogin kautta. Dialogissa esiintyviä aikeita varten oli luotu kontekstivapaat kieliopit. Pelaaja sai valita vastauksensa keskustelussa ja siten vaikuttaa ei-pelaajahahmon tunnetilaan.

Tekstinluonnin onnistumista Palmas, Raith ja Klinker (2020) selvittivät kyselyn avulla kartoittamalla esimerkiksi kuinka uskottavina, helposti seurattavina, erilaisina ja kiinnostavina pelaajat kokivat tekstit. Tuloksissa pelaajat usein tunnistivat tekstit automaattisesti luoduiksi, mutta samalla kokivat, että tekstit säilyivät kiinnostavina ja erilaisina jopa toistuvilla pelikerroilla (Palmas, Raith ja Klinker 2020).

4.2 Automaattisen tekstinluonnin vaikutus immersioon

Pelien immersiivisyyttä voidaan kehittää automaattisen tekstinluonnin keinoin ainakin kahdella tavalla: vähentämällä toisteisuutta varioimalla pelaajalle esitettyjä tekstejä ja luomalla dialogeja eri tavoin riippuen pelimaailman muuttujista. Yksinkertaisimmillaan immersion särkymistä voidaan välttää tuottamalla vaihtoehtoisia dialogikatkelmia, jotta pelaajalle ei tarvitse esittää toistuvassa tilanteessa identtisiä dialogeja. Pienetkin muutokset tekstien sisällössä ja rakenteessa voivat esimerkiksi lisätä merkittävästi tunnetta narratiivisessa pelissä olevien ei-pelaajahahmojen elävyydestä. Monimutkaisemmassa tapauksessa ei-pelaajahahmo voi muuntaa pelaajalle esitettävää dialogia esimerkiksi asenteen tai muun pelimaailman tekijän mukaan. Vastaavien dialogien kirjoittaminen on mahdollista myös ilman tekstinluontia, mutta kyse on hyvin suuresta urakasta. Lisäksi tekstinluonnin avulla voidaan nopeasti luoda valtavia määriä variaatioita, jolloin vaihtoehtoisten dialogien kirjoittaminen on perusteltua myös pelin ei-keskeisille hahmoille.

4.3 Automaattisesti tuotettujen dialogien käyttämisen haasteet

Automaattisen tekstinluomisen käyttämiseen liittyy valitusta menetelmästä riippuen erilaisia haasteita. Haasteet voivat olla luonteeltaan menetelmien tekniseen toteutukseen tai tuotetun tekstin laatuun liittyviä.

Valitun automaattisen tekstinluonnin menetelmän integrointi pelimoottoriin voi vaatia määrätyn ohjelmointikielen käyttämistä. Esimerkiksi Expressionist käyttää Python-ohjelmointikielen Eval()-funktioita suorittamaan merkkijonoja kuten ne olisivat Python koodia (Lessard ja Kybartas 2021). Automaattisen tekstinluonnin menetelmien toteuttaminen eri ohjelmointikielillä ja eri pelimoottoreihin on selkeä jatkotutkimuksen ja -kehityksen kohde.

Palmas, Raith ja Klinker (2020) havaitsivat tutkimuksessaan automaattisesti luotujen dialogien olevan helposti tunnistettavissa koneellisesti luoduiksi. Vaikka tutkimuksen tuloksia ei voida yleistää koskemaan kaikkia automaattisesti tuotettuja dialogeja, voi silti olla perusteltua olettaa, että automatisoidessa tekstinluontia, riski immersion särkymisestä voi myös kasvaa. Kyse voi olla esimerkiksi siitä, että tuotetut tekstit eivät noudata tilanteeseen sopivaa yleisesti hyväksyttävää kieltä.

Pahimmillaan automaattisesti tuotetut dialogit voivat vaikuttaa esimerkiksi pelin ikärajan määrittämiseen, sillä hallitsemattomasti tuotettuna teksti voi sisältää luokkaavaa sisältöä. Ongelmia tekstinluonnista voi syntyä, kun kaksi tai useampia tavallisen kielenkäytön sanaa yhdistettynä muodostavat epätoivottua sisältöä. Tapausesimerkki tämänkaltaisesta ongelmasta muodostui takaisinvetoon johtaneissa KeyForge-korttipelin (Garfield 2018) proseduraalisesti luoduissa korttien nimissä, joissa raportoidusti esiintyi esimerkiksi nimi "Titanflayer, the Farmer of Racism"(Hodges 2018). Eräs ratkaisu ongelmallisten sanayhdistelmien välttämiseen on luotujen tekstien suodattaminen ennen niiden saattamista pelaajille.

5 Yhteenveto

Tässä tutkielmassa pyrittiin selvittämään, voidaanko narratiivisissa digitaalisissa peleissä peli-elementteinä oleviin hahmoihin liitettyjä dialogeja olennaisesti kehittää lisäämällä dialogien variaatiota automaattisen tekstinluonnin avulla. Tarkastelun pääasialliseksi kohteeksi valittiin pelikokemuksen immersivisyyttä lisäävät dialogien kehittämisen keinot.

Katsaus tutkimuskirjallisuuteen nosti esiin useita automaattisen tekstinluonnin menetelmiä, jotka olivat soveltuvia pelikehitykseen. Löydettyjen menetelmien teoreettinen tausta pohjasi pääsääntöisesti kontekstivapaiden kielioppien käyttöön tekstin luonnissa. Menetelmät poikkesivat toisistaan suunnittelun painotuksissa ja järjestelmien laajuudessa. Työkalujen valinnassa pelien tekijät saattavat joutua tasapainottelemaan tekstinluontijärjestelmän keveyden ja tekstien monipuolisen hallinnan välillä.

Löydetty tekstinluonnin työkalut ovat soveltuvia luomaan narratiivisiin digitaalisiin peleihin dialogeja, mutta menetelmien integroiminen osaksi peliä saattaa olla tekninen haaste. Myös menetelmästä riippuen, luotujen tekstien hallinta voi vaatia huomiota. Eräissä kirjallisuudesta löytyvissä tekstinluontityökaluissa oli kuitenkin lisätty erilaisia menetelmiä tuotettujen tekstien hallinnan parantamiseksi.

Tuottaessa automaattisesti variaatioita pelin dialogeista voidaan potentiaalisesti saada peliin lisää kiinnostavuutta ja elävyyttä. Pelin immersio voidaan ehkä säilyttää paremmin pelissä tuottamalla jopa triviaaleihin keskusteluihin aiempaan verrattuna moninkertaisesti tekstuaalista sisältöä. Tekstuaalinen variointi voi kuitenkin muodostaa riskin immersion pienentymiselle, pääosin tuotettujen tekstien laadusta riippuen. Kömpelösti tuotetut tekstit voivat olla immersiota murtava tekijä tai jopa tehdä teksteistä sisällöltään loukkaavia.

Kirjallisuuskatsaus vastasi työn tutkimuskysymykseen. Automaattisen tekstinluonnin avulla dialogeja voidaan olennaisesti kehittää käyttämällä tekstinluonnin työkaluja lisäämään dialogien variaatiota. Automaattisen tekstintuottamisen käyttäminen on tehokas keino tuottaa suuria määriä vaihtelevia dialogitekstejä. Automaattisesti luotuihin teksteihin liittyy myös riskejä, joiden huomioon ottaminen on tärkeää.

Automaattisen tekstinluonnin käyttämisestä pelikehityksessä riittäisi monipuolisesti tutkittavaa. Esimerkiksi olisi hyvä tutkia menetelmiä, joilla voitaisiin varmistaa luotujen tekstien pysyminen sosiaalisten normien mukaisina.

Lähteet

Bioware. 2008. *Mass Effect*.

Brusk, Jenny, ja Staffan Björk. 2009. “Gameplay Design Patterns for Game Dialogues.” Teoksessa *DiGRA Conference*.

Chomsky, Noam. 1956. “Three models for the description of language”. *IRE Transactions on information theory* 2 (3): 113–124.

Compton, Kate, Ben Kybartas ja Michael Mateas. 2015. “Tracery: an author-focused generative text tool”. Teoksessa *International Conference on Interactive Digital Storytelling*, 154–161. Springer.

Dansky, Richard. 2007. “Introduction to game narrative”. *Game Writing: Narrative Skills for Video Games*.

Garfield, Richard. 2018. *Keyforge: Call of the Archons*.

Hodges, Dan. 2018. “Unfortunately-Named Keyforge Decks Are Being Recalled”. Viitattu 21. joulukuuta 2022. <https://techraptor.net/tabletop/news/unfortunately-named-keyforge-decks-are-being-recalled>.

Horswill, Ian. 2020. “Generative Text using Classical Nondeterminism.” Teoksessa *AIIDE Workshops*.

Howe, Daniel C. 2020. “DIALOGIC: A Toolkit for Generative Interactive Dialog”. Teoksessa *International Conference on the Foundations of Digital Games*, 1–7.

Kerr, Christopher, ja Duane Szafron. 2009. “Supporting dialogue generation for story-based games”. Teoksessa *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 5:154–160. 1.

Knochelmann, Jonas P, ja Rogelio E Cardona-Rivera. 2022. “Bronco: A Universal Authoring Language for Controllable Text Generation”. Teoksessa *International Conference on Interactive Digital Storytelling*, 541–558. Springer.

- Lessard, Jonathan, ja Quinn Kybartas. 2021. “Blabbeur-An Accessible Text Generation Authoring System for Unity”. Teoksessa *International Conference on Interactive Digital Storytelling*, 255–259. Springer.
- Merikoski, Jorma, Ari Virtanen ja Pertti Koivisto. 2004. *Johdatus diskreettiin matematiikkaan*. 177–179. WSOY.
- Michailidis, Lazaros, Emili Balaguer-Ballester ja Xun He. 2018. “Flow and immersion in video games: The aftermath of a conceptual challenge”. *Frontiers in psychology* 9:1682.
- Mäyrä, Frans. 2017. “Dialogue and interaction in role-playing games”. *Dialogue across Media* 28:271.
- Palmas, Fabrizio, Jakob Raith ja Gudrun Klinker. 2020. “A novel approach to interactive dialogue generation based on natural language creation with context-free grammars and sentiment analysis”. Teoksessa *2020 IEEE 20th International Conference on Advanced Learning Technologies (ICALT)*, 79–83. IEEE.
- Pressey, Simon, ja Mac Walters. 2008. “Dialog Production for BioWare’s Mass Effect: Lessons Learned and Future Plans”. Viitattu 15. joulukuuta 2022. <https://www.gdcvault.com/play/206/Dialog-Production-for-BioWare-s>.
- Ryan, James, Michael Mateas ja Noah Wardrip-Fruin. 2016. “Characters who speak their minds: Dialogue generation in Talk of the Town”. Teoksessa *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*.
- Ryan, James, Ethan Seither, Michael Mateas ja Noah Wardrip-Fruin. 2016. “Expressionist: An authoring tool for in-game text generation”. Teoksessa *International Conference on Interactive Digital Storytelling*, 221–233. Springer.
- Weizenbaum, Joseph. 1966. “ELIZA—a computer program for the study of natural language communication between man and machine”. *Communications of the ACM* 9 (1): 36–45.