



Lauri Alppi

# Visuaalisen romaanin luontiprosessi

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikka

Insinöörityö

20.2.2023

## Tiivistelmä

Tekijä: Lauri Alppi  
Otsikko: Visuaalisen romaanin luontiprosessi  
Sivumäärä: 34 sivua  
Aika: 20.2.2023

Tutkinto: Insinööri (AMK)  
Tutkinto-ohjelma: Tieto- ja viestintätekniikka  
Ammatillinen pääaine: Pelisovellukset  
Ohjaaja: Lehtori Miikka Mäki-Uuro

---

Insinööriyön tarkoituksena oli perehtyä visuaalisiin romaaneihin ja niiden tekoon. Videopelejä tehdään usein pelimoottoreiksi kutsutuilla ohjelmilla, jotka ovat tärkein osa pelinkehitystä. Visuaalisten romaanien tekoon on olemassa omat pelimoottorinsa, ja niistä yhtä, Tyranobuilderia, käytettiin visuaalisen romaanin toteutuksessa.

Tarkoituksena insinööriyössä oli luoda visuaalinen romaani alusta loppuun ja samalla tutustua niiden kehitykseen. Tavoitteena oli saada käsitys visuaalisen romaanin toteutuksen vaiheista ja vaativuudesta. Insinööriyössä toteutettiin visuaalinen romaani käyttämällä Tyranobuilderin omaa ohjelmointikieltä.

Peliä työstettäessä pelimoottorilla havaittiin Tyranobuilderin useat toimintaan liittyvät ongelmat. Tästä huolimatta pelimoottori todettiin helppokäyttöiseksi ja soveltuvaksi yksinkertaisten visuaalisten romaanien tekoon. Tyranobuilderin käyttöä ei voitu suosittelua kovin monimutkaisiin projekteihin sen sisältämien ongelmien vuoksi. Insinööriyön lopputuloksena syntyi läpipelattava peli, jonka tulevaisuus jäi vielä avoimeksi.

Avainsanat: visuaalinen romaani, tuotanto, ohjelmointi

## Abstract

Author: Lauri Alppi  
Title: Process of creating visual novel  
Number of Pages: 34 pages  
Date: 20 February 2023

Degree: Bachelor of Engineering  
Degree Programme: Information and Communications Technology  
Professional Major: Game Applications  
Supervisor: Miikka Mäki-Uuro, Senior Lecturer

---

The purpose of this thesis was to orient into visual novels and ways of making them. Videogames are generally made using programs called game engines which are the most important part of game development. There are game engines specialized in making visual novels and in this thesis TyranoBuilder will be used.

The main purpose in thesis was to create a visual novel from the beginning to the end and take a look at their development. The goal was to get comprehensive understanding of steps it takes to create a visual novel and its demandingness. In the thesis project visual novel was implemented using TyranoBuilder's own programming language.

While the game was being developed using the game engine, multiple issues regarding the usability of TyranoBuilder were noticed. However, the game engine was found easy to use and suitable for simplistic visual novels. Usage of TyranoBuilder cannot be recommended for complex projects due to its internal problems. As a result of the thesis project a playable game was created but its future was left open.

Keywords: Visual novel, development, programming

# Sisällys

1	Johdanto	1
2	Visuaaliset romaanit	1
2.1	Visuaalisten romaanien historiaa	2
2.2	Visuaalisten romaanien tyylilajit	7
2.3	Visuaalisten romaanien työkaluja	9
3	Insinööriyöprojektin tavoite ja määritelmä	10
3.1	Määritelmä	10
3.2	Käytetyt teknologiat	12
4	Visuaalisen romaanin toteutus	15
4.1	Ohjelmointi TyranoScriptillä Tyranobuilder-moottoria käyttäen	15
4.2	Projektin ulkoistetut osat	19
4.3	Pelin toteutuksesta	22
4.4	Pelin testaus	23
5	Työn tulokset ja tulevaisuus	26
5.1	Insinööriyön lopputulos	26
5.2	Arvio Tyranobuilder-pelimoottorista	27
5.3	Projektin tulevaisuus	28
5.4	Visuaalisten romaanien tulevaisuus	28
6	Yhteenveto	31
	Lähteet	33

## 1 Johdanto

Videopelien alkuajoista asti on narratiivi ollut tärkeässä osassa pelejä. Videopelissä kerrontaa voidaan esittää usealla tavalla, auditiivisella tai visuaalisella tavalla. Visuaaliset romaanit ovat erikoistuneet hyödyntämään visuaalista puolta.

Insinööriyössä on tarkoitus perehtyä visuaalisiin romaaneihin ja niiden tekoon, ja lisäksi toteutetaan visuaalinen romaani. Kuten musiikissa tai elokuvissa, on visuaalisissa romaaneissa useampi tyyli.

Videopelejä tehdään usein pelimoottoreiksi kutsutuilla ohjelmilla, jotka ovat tärkein osa pelinkehitystä. Visuaalisten romaanien tekoon on olemassa omat pelimoottorinsa, joista Tyranobuilderia käytetään tässä työssä visuaalisen romaanin tekoon.

Insinööriyöraportin alussa käsitellään visuaalisten romaanien historiaa ja niiden kehittymistä vuosien varrella. Historian lisäksi kerrotaan hieman suosituimmista tyyli-tyylilajeista ja eri työkaluista, joilla visuaalisia romaaneja pääasiassa tehdään. Alun jälkeen käydään läpi projektissa käytetyt eri teknologiat ja käsitellään visuaalisen romaanin tekoa ohjelmoimalla käyttämällä kyseisiä työkaluja. Projektin tarkoitus määritellään myös osana kokonaisuutta. Lopuksi käydään läpi insinööriyön tulokset ja pohditaan projektin tulevaisuutta. Visuaalisten romaanien tulevaisuus yleisesti on myös osa lopussa tapahtuvaa pohdintaa.

## 2 Visuaaliset romaanit

Visuaalinen romaani on peli, jossa pelaajat lukevat tarinaa kuin kirjaa, kuitenkin niin, että pelaajilla on apunaan visuaaliset elementit, hahmot ja taustat ruudulla kuin sarjakuvassa. Tämä auttaa hahmottamaan paikkoja, henkilöitä ja tarinan puitteita ja on vuorovaikutuksessa niiden kanssa hiiren painalluksella (1). Kuvassa 1 on yksi pelissä esiintyvistä hahmoista, jonka kanssa voi olla kanssakäymisessä.



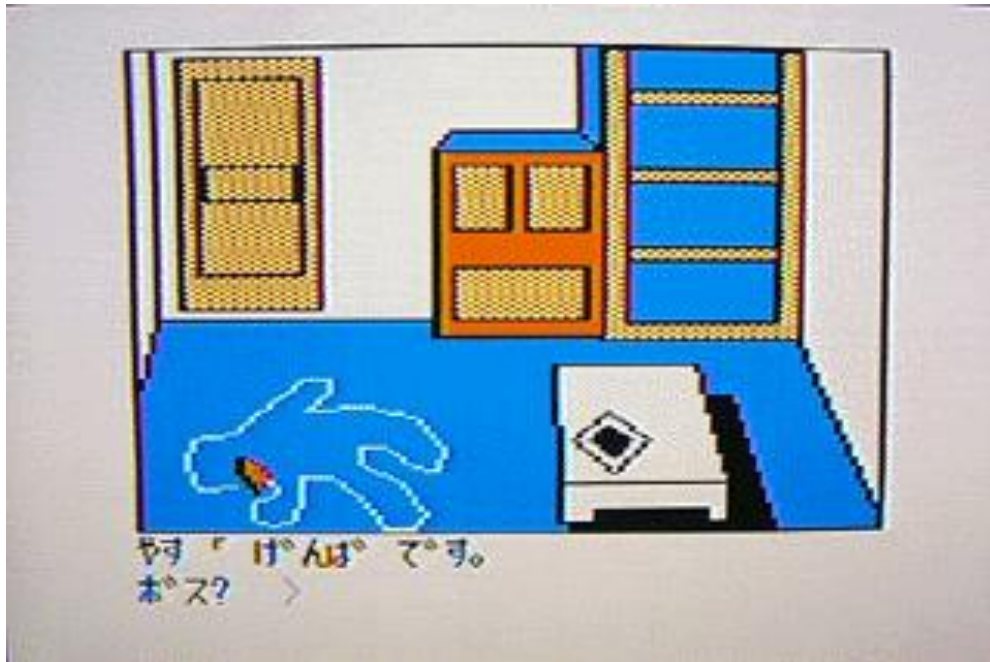
Kuva 1. Doki Doki Literature Club-pelistä (2).

Visuaalisten romaanien tyyli on usein animesta tuttua, ja peleissä käyttöliittymä on yksinkertainen ja keskittyy enemmän henkilöihin ja tapahtumiin ruudulla.

## 2.1 Visuaalisten romaanien historiaa

Visuaaliset romaanit ovat lähtöisin Japanista ajalta, jolloin tietokoneet eivät kyenneet esittämään kovinkaan monimutkaista kaksiulotteista grafiikkaa tai toistamaan monimutkaisia ääniä.

The Portopia Serial Murder Case (kuva 2) on seikkailupeli, jonka on Yuji Horiin tekemä ja julkaistiin vuonna 1983. Pelissä pelataan nimettömällä tutkijalla, jonka tehtävänä on ratkaista varakkaan pankkiirin murha yhdessä kollegansa Yasun kanssa. Peli on kuitenkin suurimmilta osin tuntematon länsimaissa, mutta Portopia pidetään yhtenä vaikutusvaltaisimpana pelinä, Japanissa. Peli on uranuurtaja seikkailupelien ja visuaalisten romaanien keskuudessa. (3.)



Kuva 2. The Portopia Serial Murder Case -pelistä (4).

Vaikka visuaaliset romaanit alkoivat pikkuhiljaa luopua pulmakeskeisestä suunnittelusta ja siirtyä narratiivikeskeisemmiksi, kaikki muut struktuurit säilyivät samanlaisina mukaan lukien visuaalinen puoli. Pelit käyttivät samaa yksittäisen taustan lähestymistapaa kuin visuaaliset romaanit nykyään. Ne eivät kuitenkaan korostaneet tätä ominaisuutta, kuten myöhäisemmät visuaaliset romaanit tekevät. Kuvat eivät liikkuneet taustan poikki, eivätkä ne liikkuneet tietyllä tavalla osoittaakseen toimintaa (5). Vuonna 1994 julkaistu Tokimeki memorial's (kuva 3) kuitenkin muutti tämän (5).



Kuva 3. Tokimeki memorial's -pelistä (5).

Mangan tavoin jokainen paneeli koottiin yhdeksi, ja jokainen kuvakulma, muodot ja muut visuaaliset elementit luovat tietyn kohtauksen tai tietyn tunnelman (5).

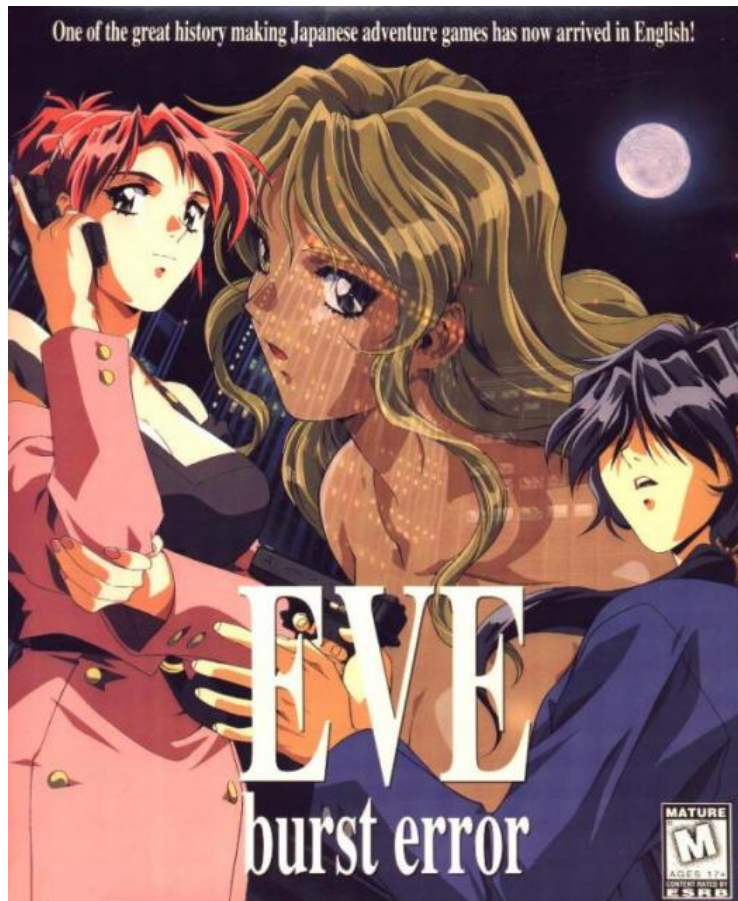
The Leaf Visual Novel Series on brändi, jota käytti 1990-luvulla pelinkehittäjä Leaf pelitrilogiassaan. Sarjan ensimmäinen osa Shizuku (kuva 4) julkaistiin vuonna 1996 ja sen jatko-osat vuosina 1996 ja 1997. (7.)



Kuva 4. Shizuku-pelistä (6).

Pelit olivat yksinkertaistettuja seikkailupelejä, jotka tehtiin muistuttamaan interaktiivisia kertomuksia haarautuvien tarinoin. Koska Leafin julkaisut keskittyivät enimmäkseen visuaaliseen puoleen, peleistä käytettiin nimitystä visuaalinen romaani totutun ääniromaanin sijaan (7). Shizuku oli siis ensimmäinen julkaistu peli, josta julkaisija käytti nimitystä visuaalinen romaani. Termi on sittemmin tullut suosituksi.

EVE Burst Error (kuva 5) on vuonna 1995 julkaistu visuaalinen romaani, joka sisältää siihen aikaan yhä yleisemmäksi tulleen kahden päähenkilön järjestelmän. Sen sijaan, että pelaaja kävisi läpi yhden päähenkilön skenaariot, ennen kuin jatkaa toisen tarinaa, huomaa pelaaja joutuvansa tekemään päähenkilöiden välillä yhteistyötä. (8.)



Kuva 5. EVE Burst Error -pelin kansikuva (9).

Jos pelaaja pelaa vain toisella hahmoista, hän jää auttamatta jumiin. Tämän tapahtuessa pelaaja joutuu vaihtamaan toiseen pelihahmoista ja pelaamaan hänen näkökulmastaan, kunnes ”avaa” toisen osapuolen tarinalle jatkoa. Tämä kahden päähenkilön välinen yhteistyö tuo peliin lisää syvyyttä.

Yu-No on ensimmäinen peli, jossa käytettiin uutta järjestelmää nimeltä automaattinen kartoitusjärjestelmä, engl. Automatic Diverge System of Mapping (ADMS). Tämä järjestelmä näyttää pelaajalle karttana, mihin suuntaan hän on tarinassa etenemässä (kuva 6) (11).



Kuva 6. Yu-No-pelissä esiintyvä kartoitus (11).

Kartta esittää pelaajalle pelin juonen haarautumiskohdat (10). Järjestelmä tallentaa pelaajan valinnat ja reitit karttaan. Kartta esittää ajan vasemmalta oikealle ja rinnakkaiset kohtaukset korkeussuunnassa (11).

## 2.2 Visuaalisten romaanien tyyllilajit

Ylivoimaisesti suosituin tyyllilaji visuaalisissa romaaneissa on romanssi ja sen yksi ilmentymä treffisimulaattorit. Näissä peleissä päähenkilöllä on usein jonkinlainen haaremi, josta pelaaja voi valita haluamansa romanssin kohteen ja pyrkiä tarinan aikana samaan romanssin syntymään. Tarinat sisältävät monia haarumakohtia ja päättyvät joko hyvään tai huonoon loppuun. Romanssi yhdistetään usein muihin genreihin. (12.) Treffisimulaattorit ovat uudelleen pelattavia useiden mahdollisten loppujen reittien takia, mikä antaa pelaajalle mahdollisuuden kokea laajan valikoiman tarinoita.

Itkemispelit tai nakige-pelit ovat visuaalisia romaaneita, joiden tarkoitus on saada pelaaja itkemään tai vähintäänkin kokemaan voimakkaita tunteita. Nämä pelit ovat usein surullisia tai suorastaan masentavia, mutta niihin sisältyy iloinen tai toiveikas loppu (kuva 7). (12.)



Kuva 7. To Heart -pelin kansikuva (13).

Utsuge on samantyyppinen kuin nakige-tyylilaji, mutta nämä pelit ovat vielä surullisempia ja niistä puuttuu kokonaan onnellinen loppu. Lisäksi utsuge-peleistä puuttuu ns. paholainen, jota syyttää kaikesta pahasta. (14.)

Tieteiskirjallisuudella on ollut vankka rooli pelialalla jo sen alkua ajoista. Romanssin tavoin scifi on laaja-alainen, ja sitä voi helposti yhdistää muihin genreihin (12). Futuristiset tarinat, joita tieteiskirjallisuudesta kumpuaa, ovat aina olleet suosittuja niin elokuvissa kuin sarjakuvissa, eivätkä visuaaliset romaanit ole tässä suhteessa poikkeus.

Nykyään kauhuteemaiset visuaaliset romaanit ovat suosittuja. Kuvitus ja ääniraita auttavat suuresti luomaan häiritsevän ja kammottavan ilmapiirin peleihin (kuva 8).



Kuva 8. Doki Doki Literature Club -pelin kansikuva (15).

Kuitenkin osassa visuaalisista romaaneista käytetään söpöä kuvitusta tai huumoria tasapainottamaan tarinan synkkää luonnetta, mikä saa kauhistuttavat kohtaukset vaikuttavammiksi (12).

### 2.3 Visuaalisten romaanien työkaluja

Pelinkehitys tehdään pääasiassa pelimoottorien avulla, joilla on mahdollista tehdä upeita ja ikimuistoisia pelejä. Visuaalisten romaanien kehittäminen on yleisimmillä pelimoottoreilla kuitenkin turhan työläs prosessi, mutta on olemassa pelimoottoreita, jotka ovat erikoistuneet visuaalisten romaanien tekoon.

Ren'py on pelimoottori, jonka avulla kuvat, sanat ja äänet saadaan kertomaan interaktiivisia tarinoita (16). Tämä moottori on avoimen lähdekoodin ohjelmisto, joka vaatii käyttäjältä vain perustason Python-ohjelmoinnin osaamista. Vaikka ohjelmointia ei osaisikaan, on Ren'pyn omissa dokumenteissa riittävä ohjeistus,

jonka avulla visuaalisen romaanin saa aikaan (17). Taitavampi ohjelmoija voi lisätä peleihin esimerkiksi esineiden hallintaa tai tilastoja.

Tyranobuilder-ohjelmalla on mahdollista tehdä visuaalinen romaani täysin ilman koodausta. Pelin jokaisen osan voi tehdä Tyranobuilderin visuaalisessa editorissa, joka koostuu, raahaa ja pudota -elementeistä (17). Tyranobuilder sisältää mahdollisuuden koodata käyttämällä JavaScriptiä tai Tyranobuilderin omaa TyranoScriptiä (18).

Yksi markkinoiden kalleimmista, mutta monipuolisimmista visuaalisen romaanin pelimoottoreista on Visual Novel Maker. Kuitenkin monipuolisuus tekee ohjelman käytöstä vähiten käyttäjäystävällisen verrattuna Tyranobuilderiin tai Ren'pyyn. Visual Novel Makerissa on laaja valikoima työkaluja, joiden avulla niin kokenut kuin aloittelija pystyy luomaan pelejä, mutta ohjelmoinnin osaaminen on kuitenkin edellytys pelimoottorin käytölle. Ohjelmointikielinä toimivat JavaScript ja CoffeeScript. Vaikka Visual Novel Maker sisältääkin Tyranobuilderin tavoin raahaa ja pudota -käyttöliittymän, suurin työ tehdään ohjelmoimalla. (18.)

### **3 Insinööriyöprojektin tavoite ja määritelmä**

#### **3.1 Määritelmä**

Idea insinööriyöhön saatiin halusta tehdä peli, joka ei olisi liian aikaa vievä tai vaativa, jolloin siitä ei olisi mitä todennäköisimmin tullut valmista vaaditussa ajassa. Projektin tarkoituksena oli perehtyä videopelien, erityisesti visuaalisten romaanien, kehitykseen ja toteutukseen. Tästä syystä aiheeksi valittiin visuaalisen romaanin teko niihin erikoistuneella pelimoottorilla. Peli toteutettiin Tyranobuilder-nimisellä pelimoottorilla. Aikaisempaa kokemusta visuaalisten romaanien tekoon oli kertynyt opintojen aikana suoritetusta pelikurssista, jossa tehtiin pelimekaniikoiltaan tavallista rikkaampi visuaalinen romaani. Aikaisemman kokemuksen lisäksi kiinnostusta visuaalisiin romaaneihin oli jo entuudestaan, mikä osaltaan helpotti ja selkeytti aiheen valintaa.

Insinööriyössä olin itse vastuussa visuaalisen romaanin ohjelmoinnista ja käsikirjoituksesta ja kuvituksesta vastasi ulkopuolinen taiteilija, jolle annettiin riittävät ohjeet vaadittujen asioiden aikaansaamiseksi.

Ennen projektin aloitusta tehtiin selväksi, ettei tarina saanut olla liian pitkä tai peliominaisuuksiltaan liian vaativa, ettei se paisuisi laajemmaksi kuin insinööriyössä haettu laajuus. Tästä syystä tarinasta tuli parissa tunnissa läpi pelattava, mikä antoi mahdollisuuden tehdä pelin kokonaisuudessaan.

Visuaalisen romaanin tyyliksi valittiin treffisimulaattori, jossa kertoja naljailee pelaajalle hänen tehdessään valintoja, ja lisäksi neljäs seinä, jolla tarkoitetaan illuusion rikkomista, on vahvasti esillä juonessa. Deadpool-peli (kuva 9) on tunnettu siitä, että se rikkoi neljättä seinää huvikseen. Pelin päähahmo on kanssakäymisissä kertojan kanssa ja uhkailee pelisuunnittelijaa sekä ohjeistaa pelaajaa koko pelin ajan (19).



Kuva 9. Pelistä Deadpool (19).

Neljäs seinä rikkomalla luotiin yhteys tarinan hahmon ja pelaajan välille, ja samalla vahvistettiin heidän välillensä olevaa suhdetta. Neljättä seinää voidaan

myös käyttää koomisena välineenä. (20.) Tarinasta haluttiin humoristinen, eikä se ota itseään tosissaan.

Ydinajatuksena pelissä oli treffisimulaattorien pinnallisuus ja se, kuinka ne ovat yleisimmin keskittyneet pelkästään siihen, kuinka nopeasti pelaaja pääsee pariutumaan haluamansa kumppanin kanssa. Lisäksi tarina otti kantaa siihen, millaisina yhteiskunnasta eristäytyneinä luusereina treffisimulaattorien pelaajia yleisesti pidetään.

Tekniselle toteutukselle annettiin takarajaksi 30.9.2022, jotta se saataisiin hyvässä ajoin valmiiksi ennen opintojen loppua. Itse tekstin kirjoittamiseen puolestaan annettiin aikaa pari kuukautta, jotta se ei venyttäisi insinööriyön valmistamista turhan pitkälle.

### 3.2 Käytetyt teknologiat

Pelinkehityksessä käytetään usein paljon erilaisia työkaluja, kuten pelimoottoreita, 3D-mallinnusohjelmia, projektinhallinnan työkaluja ja piirustustyökaluja. Pelinkehityksen työkaluista pelimoottori on ylivoimaisesti tunnetuin ja käytetyin. Pelimoottoria voi pitää pelien pohjarakenteena, jonka varassa pelit toimivat.

Ilman pelimoottoria pelinkehittäjä asettaa itsensä todellisen haasteen eteen, sillä pelimoottorin monipuolisten ominaisuuksien ansiosta pelien kehittäminen on vaivattomampaa. Pelimoottori hallitsee monia pelin komponentteja ja aikaansaa vakaan toimintaympäristön:

- pelaajan antamat komennot
- pelin fysiikat
- animaatioiden käsittely
- tekstuurien ja 3D-mallien käsittely
- äänten käsittely.

Markkinoilla on tarjolla useita erilaisia pelimoottoreita, jotka voivat olla räätälöityjä tiettyyn tarpeeseen tai suunniteltuja toimimaan kaikessa, mitä pelinkehittäjä

haluaa tehdä. Nämä pelimoottorit ovat usein kaikkien vapaasti saatavilla ja käytettävissä. Yksi suosituimmista pelimoottoreista 45 prosentin osuudella on Unity, joka mahdollistaa 2D- ja 3D-pelien kehittämisen (21).

Insinööriyöprojektin tekoon käytettiin kahta ohjelmaa, Tyranobuilder ja Notepad++, joista jälkimmäistä käytettiin sen sisältämien ominaisuuksien takia. Notepad++ ei kuitenkaan ole pakollinen ohjelma, jota olisi tarvittu visuaalisen romaanin teossa.

Tyranobuilder on maksullinen visuaalisten romaanien tekoon tarkoitettu pelimoottori. Sen kehittäjänä on Strikeworks, ja se julkaistiin vuonna 2015 Steam-kauppapaikassa (22).

Tyranobuilder sisältää raahaa ja pudota -käyttöliittymän, jonka avulla kuka tahansa pystyy tekemään visuaalisen romaanin. Kuten kuvasta 10 huomataan, on yksittäisen kohtauksen tekemiseen käytettävä useampaa elementtiä, jotta pelistä saadaan pelattava. Pidemmän päälle useamman elementin raahaaminen ja niiden muokkaaminen tulee raskaaksi ja alkaa todennäköisesti ärsyttämään käyttäjää.



Kuva 10. Tyranobuilderin näkymä (18).

Tyranobuilderilla on mahdollista luoda siinä jo valmiiksi olevat elementit koodilla, joka poissulkee jatkuvan asioiden raahaamisen ja eri asetusten muokkaamisen. Projektityö tehtiin käyttäen tätä menetelmää.

Kuten kuvasta 11 huomataan, on koodieditori kooltaan huomattavan rajallinen, vaikkakin sen kokoa pystyy hiukan kasvattamaan sen oikeassa alareunassa olevasta napista vetämällä. Tärkein asia, mikä koodieditorista puuttuu, on oikeinkirjoituksen tarkistus ja ennakoiva tekstinsyöttö, jotka ovat nykypäivän ohjelmijalle perusominaisuuksia: ne nimittäin nopeuttavat huomattavasti pelien tekoa. Tämä puutos ei kuitenkaan poissulje koodieditorin käyttöä, jos haluaa tehdä pelinsä koodaten.



Kuva 11 Tyranobuilderin koodi-ikkuna (18).

Tyranobuilder mahdollistaa valmiiden pelien julkaisun useille eri alustoille, kuten esimerkiksi selaimelle, Windows- tai Mac-alustalle ja mobiililaitteelle. Jokaisessa näistä on omat hyvät ja huonot puolensa, kuten mobiilipelien kauppapaikkojen rajoitukset (23).

Notepad++ on ilmainen lähdekoodin editoija ja tarkoitettu korvaamaan Windowsin oma muistio. Lisäksi Notepad++:aan on sisäänrakennettu useimpien ohjelmointikielten syntaksi. Tämän lisäksi ohjelmassa on ennakoiva tekstinsyöttö, joka tarjoaa jo aikaisemmin käytettyjä sanoja tai muuttujia ja näin nopeuttaa koodin kirjoittamista (kuva 12). (24.)

```
*feel
[if exp="f.cameFromGood == 'true'"]
[free_layermode]
[bg time="500" method="fadeInLeft" storage="School.png"]
[end]
endif
endmacro
endscript
```

Kuva 12. Notepad++:n ennakoiva tekstinsyöttö (24).

Jotta TyranoBuilderin käyttämät avainsanat saatiin toimimaan ennakoivasti, ne kirjoitettiin Notepad++:ssa projektin alkuun, jolloin saatiin aina haluttu ehdotus halutussa kohdassa.

## 4 Visuaalisen romaanin toteutus

### 4.1 Ohjelmointi TyranoScriptillä TyranoBuilder-moottoria käyttäen

TyranoBuilder on pelimoottori, joka on erikoitunut visuaalisiin romaaneihin. Toisin kuin esimerkiksi Ren'py, on TyranoBuilder maksullinen ohjelma. Pelimoottori tukee JavaScript- ja TyranoScript-ohjelmointikieliä. TyranoBuilderiin on mahdollista saada myös liitännäisiä (engl. plugin), jotka vaihtelevat ilmaisesta maksulliseen. TyranoBuilder ei vaadi lisenssimaksuja, joten se ei vaadi osuutta tehtyjen pelien myynnistä.

Pelimoottorin käyttö on tehty TyranoBuilderissa helpoksi erityisesti niille, joilla ei ole kokemusta ohjelmoinnista. TyranoBuilder sisältää erittäin kattavan raahaa ja pudota -käyttöliittymän. Lähtökohtaisesti visuaaliset romaanit eivät ole kovin vaikeita ohjelmoida, nimittäin ruudulla tapahtuvat asiat ovat pääasiassa seuraavia:

- tekstin eteneminen
- hahmojen lisääminen ja poistaminen
- hahmojen ilmeiden muuttaminen

- taustojen vaihtaminen
- siirtymien näyttäminen.

Ohjelmointi Tyranobuilderissa toimii siten, että halutut toiminnot kirjoitetaan tiettyillä avainsanoilla ja merkeillä. Monimutkaisempien asioiden lisääminen omaan visuaaliseen romaaniin vaatii kuitenkin ohjelmoimista JavaScriptillä. TyranoScriptin omilla avainsanoilla on mahdollista luoda kokonainen visuaalinen romaani, joka sisältää jonkin verran ominaisuuksia.

Heti ensimmäisenä tulee tekstin eteneminen. Esimerkkikoodiin 1 on otettu pelistä pari riviä koodia, joka toimii seuraavasti: Kaikki teksti yhdellä rivillä kuuluu yhteen repliikkiin, joka ilmestyy pelaajan painaessa hiiren vasenta näppäintä. Jotta seuraava repliikki ei tulisi edellisen jatkoksi vaan omanaan, täytyy jokaisen repliikin perään lisätä [p] osoittamaan ohjelmalle, että se poistaa edellisen tekstin.

```
It's a beautiful day. The sun is shining, the birds are chirping and
it fills your heart with a feeling of warmth.[p]
Having been finally accepted to the high school of your dreams, you
have a gentle skip to your step as you head to the school yard.[p]
```

Esimerkkikoodi 1. Tekstin eteneminen koodissa.

Jos ennen repliikkiä on aikaisemmalla rivillä ristikkomerkki ja nimi, merkitsee tämä repliikin kuuluvan jollekin hahmolle. Tällöin hahmon nimi tulee näkyviin osoittamaan pelaajalle, että kyseinen hahmo puhuu. Tyranobuilderissa on sisäänrakennettuna ominaisuus, joka jatkaa hahmon nimen esittämistä myös seuraavissa repliikeissä, ellei erikseen ilmoita pelkän ristikkomerkki avulla, että hahmon nimeä ei tarvita repliikissä. Esimerkkikoodissa 2 kaksi näkyy, kuinka tämä on toteutettu.

```
#???
Ueeeh! Are you a new student?[p]
#
A generic heroine type character approach you. You can sense that her
character portrait was given exactly 10 minutes of attention. Like...I
mean. Look at her! Her hair is pink for god's sake![p]
Ahem..I'm getting sidetracked. Let's see what this one has to say.[p]
```

Esimerkkikoodi 2. Hahmolle kuuluvan repliikin merkitseminen.

Tekstin lisääminen on siis erittäin vaivatonta, mutta peli jäisi todella tylsäksi, jos siinä ei olisi yhtään kuvaa. Esimerkkikoodissa 3 on käsky, joka saa BoobsMcGee-nimisen hahmon näkyviin. Ennen tätä on kuitenkin hahmo lisättävä pelin tietoihin ja annettava sille kaikki halutut ilmeet. Hahmon ilme kutsutaan halutun kuvan nimellä.

```
[chara_show name="BoobsMcGee" time="350" wait="true" storage="chara/1/Boobtocity_A.png" width="720" height="1280" left="1000" top="20"]
```

Esimerkkikoodi 3. Hahmon näyttäminen ruudulla koodin avulla.

Hahmon lisäämisen lisäksi komento sisältää ns. lisäysajan, joka on millisekunnin, ja sen, odotetaanko, että hahmo on lisätty vai jatketaanko suoraan. Lisäksi komento sisältää kuvan koon ja sijainnin vasemmasta yläkulmasta.

Esimerkkikoodissa 4 oleva "chara\_hide" poistaa hahmon ruudulta viidessäsadasssa millisekunnissa pitäen huolta hahmon positiosta sitä poistettaessa. Hahmon nimi on pakollinen tässä käskyssä, jotta pelimoottori osaa poistaa oikean hahmon.

```
[chara_hide name="BoobsMcGee" time="500" wait="true" pos_mode="true"]
```

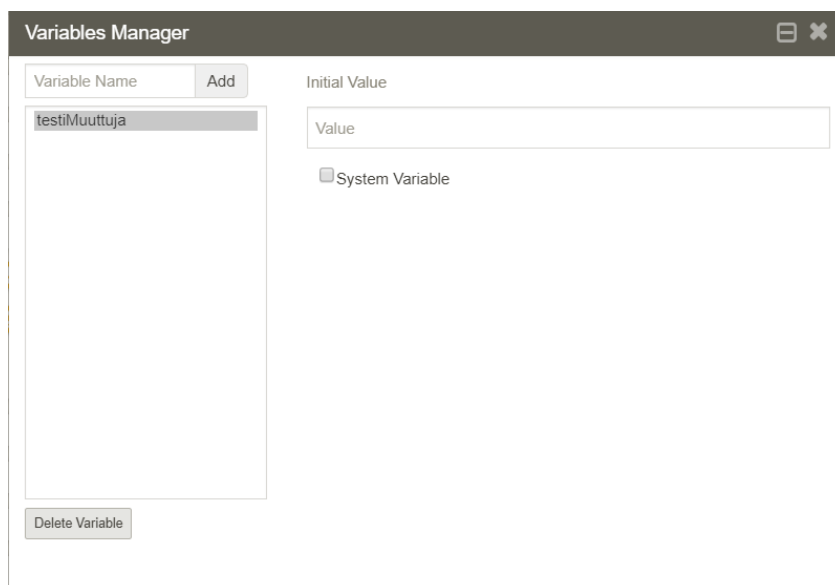
Esimerkkikoodi 4. Hahmon poistaminen ruudulta.

Jotta visuaalisesta romaanista saa mielenkiintoisemman, siihen on hyvä lisätä tarinan etenemiseen vaikuttavia tekijöitä. Esimerkkikoodissa 5 on esitelty yksi tällainen tekijä. Glink-komento luo ruudulle napin, jota painamalla siirrytään tiettyyn kohtaan kertomusta. "Storage" kertoo ohjelmalle kohtauksen sijainnin. Tekstin koko saadaan määritettyä size-komennolla, ja x ja y ovat napin sijainti. Target-komento on kohta, jonne hypätään, kun nappia painetaan. Haluttu paikka määritellään käyttämällä \*-merkkiä ja haluttua sanaa ennen repliikkiä. Jos sanoja on enemmän kuin yksi, täytyy jokaisen välilyönnin tilalle laittaa teksti &nbsp; tai pelimoottori jättää välilyönnit pois. Loppuun on laitettava [s]-komento, joka pysäyttää ohjelman etenemisen, jotta se ei jatkaisi automaattisesti ensimmäiseen vaihtoehtoon.

```
[glink storage="BUMKUNpath.ks" size="20" x="350" y="100" target="*one"
text="One?" ]
[glink storage="BUMKUNpath.ks" size="20" x="350" y="150" tar-
get="*yeah" text="Yeaah!" ]
[s]
```

Esimerkkikoodi 5. Kaksi vaihtoehtoa luotu koodilla.

Tyranobuilder tukee muuttujien käyttöä, mikä auttaa tekemään pelistä monipuolisemman. Muuttujien luominen on Tyranobuilderilla helppoa: käyttäjän ei tarvitse kuin projektin asetuksista ilmoittaa haluamansa muuttujat ja niiden alkuarvot. Muuttuja voidaan ilmoittaa järjestelmän muuttujana (engl. system variable), jonka arvo ei palaudu alkuperäiseen, vaikka pelin aloittaisi alusta (kuva 13).



Kuva 13. Tyranobuilderin muuttujavalikko.

Muuttujien arvojen muuttaminen koodissa onnistuu vaivatta käyttämällä "eval exp" -komentoa, jossa muuttujan arvoksi annetaan haluttu asia yhtäsuuruusmerkin avulla. Huomionarvoista on kuitenkin muuttujan eteen merkittävä f. tai sf., jotka kertovat moottorille muuttujien olevan joko tavallisia tai järjestelmän muuttujia. Muuttujia voidaan käyttää esimerkiksi tarkistamaan, onko jokin tietty vaatimus pelissä täytynyt käyttämällä muista ohjelmointikielistä tuttua if-lauseketta. Sen loppuun on kuitenkin laitettava [endif]-komento, joka kertoo pelimoottorille if-lausekkeen loppuvan (esimerkkikoodi 6).

```
[eval exp="f.playMusic = 'true'"]
[if exp="f.timesClicked != 11"]
[eval exp="f.timesClicked += 1"]
[endif]
```

Esimerkkikoodi 6. Muuttujien arvon muuttaminen ja if-lausekkeen käyttö.

Lopuksi käydään läpi siirtymä kohtausten välillä. Siirtymä on helppo tehdä "jump"-komennolla, jossa annetaan kohtauksen nimi ja kohde, josta halutaan aloittaa. Loppuun on vielä laitettava [s]-komento, jonka tarkoitus on estää pelimoottoria suorittamasta seuraavia koodirivejä vanhasta kohtauksesta. "Jump"-komentoa on mahdollista käyttää myös kohtauksen sisällä, jos halutaan ohittaa jokin tietty kohta kyseistä kohtausta, ja tällöin kohtauksen nimeksi asetetaan nykyinen kohta ja kohteeksi haluttu siirtymäkohta (esimerkkikoodi 7).

```
[jump storage="BUMKUNpath.ks" target="*start"]
[s]
```

Esimerkkikoodi 7. Siirtymä kohtaukseen BUMKUNpath.

Uuden kohtauksen alkuun on hyvä asettaa "pagebreak"-komento, joka poistaa ruudulta siinä mahdollisesti olleet valintanapit ja kuvat. Mahdollisen taustamusii-kin voi käynnistää "playbgm"-komennolla, johon ilmoitetaan musiikin nimi, ja lisäksi on mahdollista saada musiikki soimaan toistolla asettamalla toisto todeksi. Kaiken tämän jälkeen ruudulle tulisi jälleen uutta tekstiä (esimerkkikoodi 8).

```
[playbgm storage= "elevator.mp3" loop = "true"]
```

Esimerkkikoodi 8. Elevator-nimisen kappaleen toistaminen koodilla.

## 4.2 Projektin ulkoistetut osat

Insinööriyöprojektista jouduttiin ulkoistamaan grafiikka ja kerronta, koska omat taidot olivat niihin puutteelliset. Ulkoistuksesta on kuitenkin pelinkehityksessä hyötyä, sillä videopelien kehitys voi pahimmillaan viedä vuosia ja se vaatii muutenkin kuin pelin konseptin kehittämistä (25).

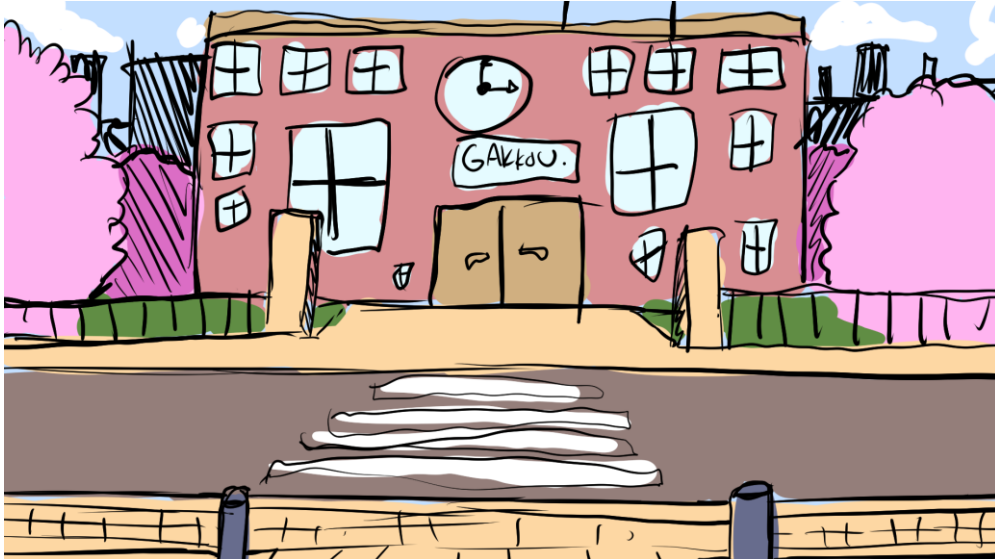
Ulkoistamisen suurin hyöty on kustannusten vähentäminen. Kun pelin grafiikan ulkoistaa, se vähentää työntekijöiden määrää ja tarvetta sekä poistaa mahdolliset palkkauskustannukset. Ulkoistaminen mahdollistaa myös halvemman työvoiman käytön. (26.)

Joustavuus on yksi syy ulkoistaa osia pelinkehityksestä, koska näin mahdollistaa väliaikaisen resurssien kasvattamisen ilman suuria kustannuksia. Tämä mahdollistaa joidenkin yksitoikkoisten asioiden siirtämistä ulkopuoliselle, jolloin pystyy itse keskittymään mielenkiintoisempiin tai tärkeämpiin asioihin pelissä. (26.)

Ajankäytön kannalta on järkevää ulkoistaa osia pelinkehityksestä. Pelien ulkoistukseen erikoistuneilla yrityksillä on valmiiksi kaikki ammattiosaaminen, jota kehityksessä vaaditaan, joten työntekijöiden etsimiseen ei kulu aikaa. Yritykset myös saavat vaaditut asiat tehdyksi määräaikaan mennessä, mikä nopeuttaa pelinkehitystä ja mahdollistaa keskittymisen muihin tärkeisiin asioihin. (25.)

Yksi iso ja mahdollisesti suurin hyöty ulkoistuksessa on yleiskustannusten puuttuminen, joita ovat esimerkiksi toimisto, tietokoneet ja kaikki mahdolliset lisenssit. Koska ulkoistukseen erikoistuneilla yrityksillä on mahdollisuus tarjota osaava ryhmä, siirtyy vastuu työntekijöiden hallinnasta palvelun tarjoajalle. Tämä tarkoittaa vapautta byrokratiasta ja tarpeettomista kuluista. (24.)

Graafiseksi tyyliksi visuaaliseen romaaniin valittiin kehoja ja kömpelöä piirtojälkeä muistuttava tyyli, jonka tarkoituksena on luoda pelille primitiivinen ja halvan oloinen ilme (kuva 14). Tämä tyyli valittiin myös sen nopean toteutettavuuden takia.



Kuva 14. Yksi pelissä käytettävistä taustoista.

Kaikki projektiin tullut grafiikka ulkoistettiin, jotta ajankäyttö saataisiin maksimoitua pelimoottorin käytön opetteluun sekä ohjelmointiin. Kaiken kaikkiaan urakka ei olisi ollut vuosia kestävä, vaikka taiteilija piirsi muutaman kuvan uudestaan:

- hahmografiikka – 3 hahmoa
- hahmojen ilmeet – 26 kuvaa
- käyttöliittymä – 5 kuvaa
- taustagrafiikka – 21 kuvaa.

Narratiivi on jo useita vuosia ollut videopelien tärkeimpiä elementtejä. Ensimmäisenä esimerkkinä voidaan pitää vuonna 1979 julkaistua Choose Your Own Adventure -kirjaa, joka oli interaktiivinen pelikirjasarja nuorille. Vuosien 1979–1999 välillä kirjasarja myi yli 250 miljoonaa kopiota (27).

Choose Your Own Adventuren ”sinä”-keskeisten valintojen uskotaan olevan syynä lukuisiin peleihin, joita julkaistiin sarjan jälkeen. Japanissa suosittu Bishoujo-videopelit, jotka yhdistävät narratiivin pelimekaniikoihin, merkitsevät alkua trendille, jossa pelaaja kykenee vaikuttamaan pelin tarinaan. (27.)

Pelin käsikirjoittajalle annettiin selkeät ohjeet luoda pelin juonesta humoristinen ja sisällyttää siihen visuaalisten romaanien tavallisimmat stereotyyppit ja kliseet.

Tarina luotiin prosessikirjoituksella, jossa tarina kirjoitetaan kokonaan loppuun ja sitten tehdään tarvittavia muokkauksia. Käsikirjoitus saatiin luettavaksi nopeasti, eikä siihen oikeastaan tullut paria pientä kohtaa lukuun ottamatta muutoksia. Nämä muutokset keskittyivät lähinnä poistamaan ominaisuuksia, joiden tekeminen olisi joko ollut turhan hidasta tai pelimoottorin rajallisuuden takia melkein mahdotonta toteuttaa.

### 4.3 Pelin toteutuksesta

Pelin käsikirjoituksesta tuli lopulta hieman yli sata sivua, ja se sisälsi kaikki pelissä olevat kohtaukset, vuorosanat ja ohjeet jokaiselle ilmeelle ja taustalle. Pelin hahmoille oli jokaiselle pienimuotoinen esittelykohtaus, jossa pelaajalle annettiin pieni tutustumismahdollisuus hahmoihin. Lisäksi jokaisella hahmolla oli oma polkunsä, ja polut oli mahdollista valita haluamassaan järjestyksessä. Jokaisen hahmon tarina oli aivan erilainen kuin muiden. Pelissä oli kolmen eri trefikumppanin lisäksi mahdollisuus saada useampi erilainen lopetus, joista jokainen vaati pelaajalta oikeat valinnat.

Dialogit noudattivat pelissä yksinkertaista kaavaa, jossa hahmojen vuorosanat tulivat ja menivät ja sekaan laitettiin vaihtoehtoja reagoida eri tilanteisiin mielekkäimmällä tavalla. Useat eri vaihtoehdot mahdollistivat saman tarinan monipuolistamisen, mikä teki uusista pelikerroista erilaisia. Monimutkaista monipolkuista karttaa ei peliin tullut, vaan peli pelataan läpi käyden kaikkien hahmojen kohtaukset läpi halutussa järjestyksessä, minkä jälkeen peli antoi loppukohtauksen tehtyjen valintojen perusteella.

Erityistä Tyrano-builderilla työskentelyssä oli vaatimus käyttää vastauksiin tarkoitetuissa painikkeissa luvussa 4.1 mainittua &nbsp;-komentoa, jolla välilyönnit saatiin tekstiin aikaiseksi. Tästä tekee erityistä se, että kyseinen komento poistui jokaisesta kohdasta koodia, kun kohtauksen käynnisti.

Helppoa projektissa oli kaikkien vuorosanojen asettaminen, mutta kuitenkin täytyi aina muistaa laittaa jokaisen puheenvuoron loppuun pysäytyskomento, jotta

pelimoottori osasi odottaa pelaajan komentoja. Lisäksi eri siirtymät kohtausten välillä olivat helposti toteutettavissa. Haastetta projektiin toi eri muuttujien käyttö, koska niitä käytettäessä pitää olla tarkkana, tuleeko muuttuja yksittäis- vai kaksoislainausmerkkien sisään. Hieman turhauttavana asiana projektissa oli eri käyttöliittymän osien asettaminen paikalleen, kun pelimoottorilla ei ollut antaa minkäänlaisia valmiita koordinaatteja, joita olisi voinut hyödyntää, vaan jokaisen napin ja painikkeen sijainti täytyi kokeilemalla löytää.

#### 4.4 Pelin testaus

Pelin testaus videopeleissä on epäjohdonmukaisuuksien, häiriöiden ja vikojen etsimistä ohjelmasta tai pelikokemuksesta. Laadunvarmistus sisältää dokumentointia, jäljentämistä ja epäjohdonmukaisuuksien arvioimista, kunnes ne ovat kunnossa (28).

Visuaalisen romaanin testaus ei ole kaikkein vaikeimmasta päästä, sillä ne yleensä sisältävät vain pelin kulun varmistuksen alusta loppuun. Näin on myös tämän projektin kanssa, vaikkakin se on aikaa vievää työtä. Tyranobuilder sisältää sisäänrakennetun debuggaus-ikkunan, jonka avulla kyetään seuraamaan koodin etenemistä pelissä sen kulkiessa eteenpäin.

Kuvassa 15 huomataan virheilmoitus, joka kertoo, ettei haluttua tiedosta löydy. Tämä johtuu virheellisestä nimeämisestä koodissa: nimittäin kuvalle, jota koetaan saada ruudulle, on annettu virheellinen nimi. Tyranobuilder ei kuitenkaan

tarkasti osaa kertoa käyttäjälle virheiden syytä, vaan se on käyttäjän itse selvitettävä.

```

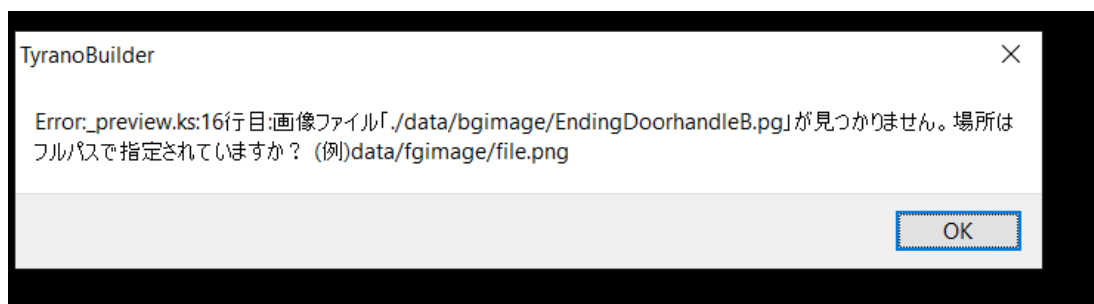
{line: 10, name: "tb_show_message_window", pm: {...}, val: "", ks_file: "_p
review.ks"}
**:1 line:4 kag.js:1
kag.js:1
{line: 4, name: "layopt", pm: {...}, val: "", ks_file: "system/builder.ks"}
**:2 line:5 kag.js:1
kag.js:1
{line: 5, name: "layopt", pm: {...}, val: "", ks_file: "system/builder.ks"}
**:3 line:6 kag.js:1
kag.js:1
{line: 6, name: "endmacro", pm: {...}, val: "", ks_file: "system/builder.k
s"}
**:9 line:11 kag.js:1
kag.js:1
{line: 11, name: "autosave", pm: {...}, val: "", ks_file: "_preview.ks"}
**:10 line:12 kag.js:1
kag.js:1
{line: 12, name: "text", pm: {...}, val: "Hrm? Oh...You're still here...? W
ell... To be fair... Well. Let me...actually do something for you! ", ks_
file: "_preview.ks"}
**:11 line:12 kag.js:1
kag.js:1
{line: 12, name: "p", pm: {...}, val: "", ks_file: "_preview.ks"}
**:12 line:13 kag.js:1
kag.js:1
{line: 13, name: "text", pm: {...}, val: "I got a good idea! Okay?! This'll
be great. Gimme a second.", ks_file: "_preview.ks"}
**:13 line:13 kag.js:1
kag.js:1
{line: 13, name: "p", pm: {...}, val: "", ks_file: "_preview.ks"}
**:14 line:14 kag.js:1
kag.js:1
{line: 14, name: "text", pm: {...}, val: "...Hmmm...hm...Hm...Okay! Her
e... ", ks_file: "_preview.ks"}
**:15 line:14 kag.js:1
kag.js:1
{line: 14, name: "p", pm: {...}, val: "", ks_file: "_preview.ks"}
**:16 line:15 kag.js:1
kag.js:1
{line: 15, name: "bg", pm: {...}, val: "", ks_file: "_preview.ks"}
✖ GET file:///C:/Program%20Files%20(x86)/Steam/steamapps/common/TyranoBuilder/myproject/ThatOneTrashyGame/data/b
gimage/EndingDoorhandleB.pg net::ERR_FILE_NOT_FOUND

```

Kuva 15. Debuggaus-ikkuna, jossa nähdään koodin eteneminen pelissä ja virheilmoitus.

Debuggaus-ikkunan lisäksi TyranoBuilder saattaa antaa käyttäjälle ponnahdus-ikkunana ilmoituksen virheestä (kuva 16). Valitettavasti virheilmoitus on

sekoitus japania ja englantia, eikä siihen ole olemassa korjausta. Näin ollen kaikki mahdollinen testaus on tehtävä debuggaus-ikkunan kanssa.

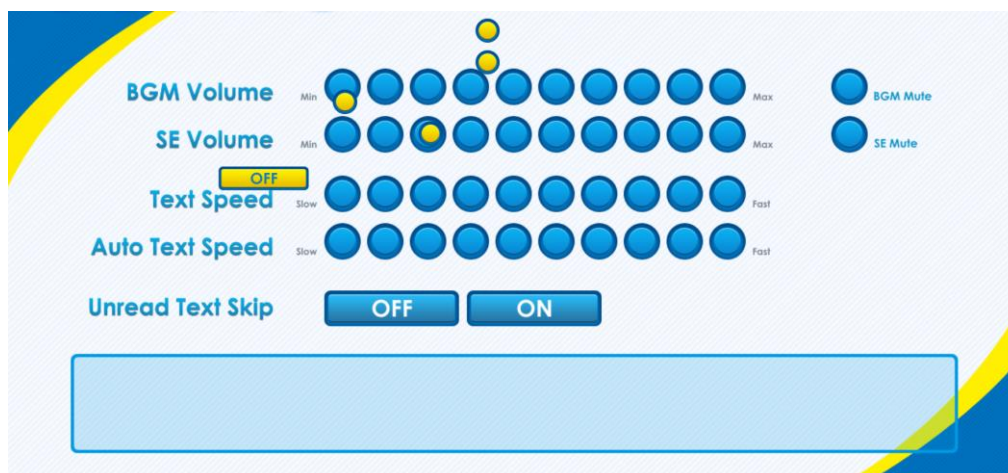


Kuva 16. Virheilmoitus, joka on osittain japaniksi ja osittain englanniksi.

Peliä testattaessa ei tullut useita virheitä, mikä johtui osittain pelin yksinkertaisuudesta ja TyranoBuilderin helppokäyttöisyydestä. Lisänä voidaankin mainita tarvittavien muuttuja lausekkeiden virheettömyys. Virheet, jotka testausvaiheessa ilmenivät, koostuivat lähinnä oikeinkirjoitusvirheestä tai väärän muuttujan käytöstä. Helppoa oli myös unohtaa asettaa jokaiselle vastausvaihtoehdolle vaaditut kohteet. Suurin ongelma peliä testattaessa oli osittain japaninkieliset virheilmoitukset, joita ei pystynyt tulkitsemaan. Osa ongelmista ei suinkaan johtunut omista virheistä tai taitojen puutteista, vaan pelimoottorin viallisuudesta.

Yhtenä esimerkkinä on asetusvalikon (kuva 17) skaalauksen puuttuminen, josta seuraa nappien väärissä paikoissa oleminen. Vastaavien ongelmien kanssa

joutuu helposti olemaan yksin, sillä Tyranobuilderilla ei ole kovin suurta käyttäjäkuntaa, joka voisi näistä ongelmista keskustella.



Kuva 17. Viallinen asetusvalikko.

## 5 Työn tulokset ja tulevaisuus

### 5.1 Insinööriyön lopputulos

Kun pelejä tehdessä onnistuu tekemään jotain toimivaa ja kunnollista, se antaa motivaatiota jatkaa pelin tekoa. Kaikissa projekteissa olisi tarkoitus saada aikaan enemmän onnistumisia kuin ongelmia, vaikkakin niitä ilmaantuu aika ajoin. Yhtenä positiivisena asiana pitäisin Tyranobuilderin helppokäyttöisyyttä ja sen tapaa tehostaa työntekoa. Työkalun avulla onnistuin luomaan pelattavan kokonaisuuden, joka on mahdollista jakaa muiden pelattavaksi.

Ongelmilta ei kuitenkaan pelinkehityksessä pysty välttymään, ja näin kävi myös tässä projektissa. Virheiden minimointi vaatii tarkkuutta, mikä tekee videopelien tekemisestä hankalaa. Tyranobuilderin sisäiset ongelmat aiheuttivat päänvaivaa peliä tehdessä, eivätkä ongelmat poistuneet projektin aikana pelimoottoriin tulneiden useiden päivitysten mukana. Jokaisen kuvan tai ääniraidan oikeinkirjoitus tuli aina tarkistaa manuaalisesti, jotta välttyttiin pelin kaatumiselta jo testausvaiheessa.

Projekti onnistui loppujen lopuksi hyvin, ja tulokseksi saatiin peli, joka nälvii pelaajaa ja rikkoo neljännen seinän, kuten aikaisemmin luvussa 3 tuli esiin. Peliin saatiin paljon huumoria, ja hiukan vakavampaakin asiaa saatiin istutettua sekaan. Pelattavuudeltaan peli toimii ja soljuu mukavasti eteenpäin, eikä se tunnu tylsältä kertojan kommenttien siivittämänä. Kun kunnolla uppoutuu pelin tarinaan, saa sen pelattua läpi parissa tunnissa. Kaikkiin lopetuksiin pääseminen kuitenkin vaatii useamman läpipeluun, mikä lisää pelin uudelleenpelattavuutta. Pelin äänimaailmasta tuli yksinkertainen mutta toimiva juuri sen helpon ymmärrettävyyden takia.

Peliin olisi kuitenkin haluttu lisätä asetusvalikon lisäksi dialogeissa esiintyviin vastausvaihtoehtoihin omat kuvitukset, mutta se olisi vaatinut jokaisen vaihtoehdon uudelleen ohjelmointia ja rajallisen ajan takia näin ei tehty. Lisäksi kohtausten väliin olisi haluttu tehdä omat animaatiot. Mahdollisia ongelmia ja virheitä voi pelistä vielä ilmetä. Ottaen huomioon, että pelin voi pelata läpi niin monella eri tapaa, jää testauksessa väkisin jotain huomaamatta.

## 5.2 Arvio Tyrano-builder-pelimoottorista

Tyrano-builder on yksinkertaisuudessaan helppo käyttää, mutta sen helppokäyttöisyys on sen heikoin lenkki. Jos pelinkehittäjä haluaa tehdä peliinsä monimutkaisia ominaisuuksia, joutuu melkein takuuvarmojen ongelmien kanssa olemaan yksin. Pelimoottori ei ole suosituimmasta päästä, ja sen seurauksena eri foorumeilta tai sivustoilta ei välttämättä löydy apua ongelmiin. Turhauttavaa on Tyrano-builderin käytössä se, että etukäteen ei pysty tietämään, mikä on vikana, ja virheilmoitukset ovat usein yhdistelmä japania ja englantia, mutta niistä tulisi pystyä päättämään, missä vika on.

Helppokäyttöisyydestä voi antaa kiitosta Tyrano-builderille. Selkeä käyttöliittymä ja perusasioiden teko ei ole pelimoottorissa vaikea toteuttaa, ja niiden ansiosta nuoremmatkin voivat onnistua toteuttamaan pelattavan visuaalisen romaanin. Pelien loppuun saattaminen on myös todella helppoa, joten ne saa nopeasti myös jaettua muille pelattavaksi. Maailman parhaasta visuaalisten romaanien

tekoon tarkoitettu pelimoottorista ei kuitenkaan ole kyse, mutta jos etsii helposti käytettävää ja riittävän hyvin toimivaa työkalua, voi Tyranobuilder olla siihen enemmän kuin riittävä. Tämä avaa mahdollisuuden myös uusien visuaalisten romaanien tekemiseen Tyranobuilderilla, nyt kun pelien tekemiseen on saatu kokemusta ja osaamista. Visuaalisten romaanien teko on suhteellisen vaivatonta työtä, jota voisi tehdä tavallisen päivätyön ohessa.

### 5.3 Projektin tulevaisuus

Insinööriyönä valmistunut visuaalinen romaani julkaistaan Steam-kauppapajassa. Peli tulee olemaan ilmainen, mikä mahdollistaa suuremman yleisön tavoittamisen, koska hinnasta johtuvaa kynnystä ei ole. Idea pelin ilmaiseen lataukseen tulee Doki Doki Literature Club -pelistä, joka on myös mahdollista ladata ilmaiseksi, mikä sai ihmiset pelaamaan peliä (28). Jos peliä ladataan riittävän monta kertaa, sitten tehdään lisäsisältöä, joka puolestaan on maksullista. Mikäli peli ei menesty toivotulla tavalla, ei siihen tehdä uutta sisältöä vaan se jätetään omaan arvoonsa.

Mahdollisen menestymisen myötä aukeaa ovi tehdä pelille jatko-osa tai mahdollisia oheistuotteita. Lisänä voidaan tehdä sosiaalisen median tilit, joiden avulla kyetään olemaan yhteydessä pelistä kiinnostuneiden kanssa.

### 5.4 Visuaalisten romaanien tulevaisuus

Visuaaliset romaanit kasvattavat suosiotaan kotitekoisten pelien ansiosta, joita löytää itse tehtyjen pelien jakoon tarkoitetuilta sivustoilta kuten itch.io. Visuaalisten romaanien tulevaisuus on hiukan kunnianhimoinen, mutta toiveikas. Peleillä on kuitenkin pitkä matka siihen, että ne tunnustetaan pelialalla laajemmin. Yksi asia, joka auttaisi visuaalisten romaanien menestystä, olisi niiden pornografisen kuuluisuuden yli pääseminen (29).

Yksi universaalisti käytetyimmistä tavoista kommunikoida ja tutustua toisiinsa on tarinankerronta, ja videopeleistä on tulossa jatkuvasti tarinavetoisempia. Yhä

useampi pelejä pelaava arvostaa tarinallista pelattavuutta ja tarinaa. Siksi visuaaliset romaanit ovat erinomainen esimerkki, sillä ne ovat erikoistuneet dialogiin ja tarinaan. Vuosien myötä videopelit ovat kehittyneet yhdeksi mukaansatempaavimmista tavoista ihmisten kokea kerrontaa lajityypistä riippumatta. Pelinkehitys on kuitenkin aikaa vievää ja kallista, mikä on esteenä monelle pelinkehittäjälle. Visuaalinen romaani voisikin olla vallankumouksellinen tapa saada näkemyksensä esille. (30.)

Yleensä visuaalisissa romaaneissa ollaan vuorovaikutuksessa pelin kanssa, mutta visuaalinen romaani voi olla myös pelkästään tekstiä. Visuaalisilla romaaneilla ei ole vakiintuneita sääntöjä, mutta tyypillisesti suurin osa ajasta kuuluu kuitenkin tekstin lukemiseen. Erityisesti japanilaisessa kulttuurissa liikutaan paljon junilla, ja niissä tällainen helposti kulutettava media on erittäin suosittua.

Visuaaliset romaanit ovat läntisessä maailmassa edelleenkin kulutukseltaan vähäisiä, mutta niihin on alettu kiinnittää huomiota ja niistä on alettu keskustella. Osana tähän huomion kasvuun on Aksys Games -nimisen yrityksen tekemä työ Japanissa suosittujen visuaalisten romaanien lokalisoinnissa (kuva 18) (31).



Kuva 18. Nine Hours -pelissä esiintyviä hahmoja (32).

Valitettavaa on se, että paras tapa saada näkyvyyttä visuaaliselle romaanille on sisällyttää peliin seksuaalisia teemoja. Nämä pelit on kuitenkin suunnattu erilaiselle yleisölle, ja yhä kasvavan kerronnallisuuden merkityksen takia on visuaalisen romaanin mahdollista menestyä. Esimerkiksi Telltale Gamesin The Walking Dead -peliä myytiin 20 päivässä yli miljoona kappaletta. Peli sisälsi erinomaisen tarinan ja oli mekaniikoiltaan erittäin rajattu, mikä teki kerronnasta merkityksellisempää. (33.)

Virtuaalisella todellisuudella tulee olemaan osansa visuaalisten romaanien tulevaisuudessa, kun teknologian kehittyessä ja halventuessa voidaan visuaalisia romaaneita tehdä virtuaaliseen todellisuuteen (kuva 19).



Kuva 19. Tokyo Chronos -pelin kansikuva (34).

Virtuaalisella todellisuudella on mahdollisuus saada kerronnasta mukaansatem-  
paavampaa ja saada pelaaja kokemaan tarina, kuin hän olisi paikan päällä (29).

Visuaalisilla romaaneilla voi olla tulevaisuus markkinoilla, mutta se voi myös ka-  
dota markkinoilta. Erilaisten mobiililaitteiden ansiosta visuaaliset romaanit ovat  
saamassa uuden mahdollisuuden, mutta vain, jos kerronta pysyy laadukkaana  
ja innostavana.

## 6 Yhteenveto

Insinööriyön tavoitteena oli toteuttaa visuaalinen romaani käyttämällä niihin eri-  
koistunutta pelimoottoria ja raportoida työvaiheista. Raportissa esiteltiin useita  
tyylilajeja, joista yhtä käytettiin pelin teossa. Visuaalisissa romaaneissa on ra-  
jana mielikuvitus ja luovuus.

Visuaalisia romaaneita on ollut jo 1980-luvulta saakka, ja ne ovat ajan kuluessa  
muuttuneet niin ulkomuodoltaan kuin pelattavuudeltaan. Kuitenkin pelien pe-  
rusta on sama: luoda mukaansatempaava tarina, jota on mukava seurata ja ko-  
kea. Yksinkertaisimmat pelit ovat vain interaktiivisia tarinoita, mutta

monipuolisimmat ovat pelimekaniikoiltaan rikkaita. Vaikka visuaalinen romaani saattaisi olla pelkkä interaktiivinen tarina, ei se estä pelin suosiota tai tee pelistä huonoa.

Romanssi on ylivoimaisesti suosituin visuaalisten romaanien lajityyppi, ja se on yhdistetty usein muihin lajityyppeihin. Siksi ei tule yllätyksenä, jos etsii muista genreistä itselleen peliä ja lopulta päätyy peliin, johon on yhdistetty romanssia.

Visuaalisten romaanien teko vaatii yllättävän paljon työtä, vaikkakin niiden tekeminen on huomattavasti helpompaa kuin perinteisten pelien. Ohjelmoinnin suoraviivaisuudesta huolimatta projektin saattaminen valmiiksi vie testauksen haastavuuden vuoksi enemmän aikaa kuin kuviteltiin.

Projekti opetti huomattavasti pelinkehityksestä ja suunnittelusta, eikä työ ole suinkaan vielä ohi, sillä peli julkaistaan myöhemmin, kunhan kaikki mahdolliset virheet ja viimeistelyt on hoidettu.

## Lähteet

- 1 Mai, Kevin. 2014. Rockmandas'h beginner's guide to visual novels. Verkkoaineisto. Reikazerambles. <<https://reikazerambles.com/posts/the-beginners-guide-to-visual-novels-1541975662/>> Luettu 29.11.2021.
- 2 Doki Doki Literature Club. Verkkoaineisto. Team Salvato. <<https://ddlc.moe/>> Luettu 11.12.2021.
- 3 Visual Novel / The Portopia Murder Case. Verkkoaineisto. TV Tropes. <<https://tvtropes.org/pmwiki/pmwiki.php/VisualNovel/ThePortopiaSerialMurderCase>> Luettu 2.12.2021.
- 4 Crimmins, Brian. 2016. A Brief History Of Visual Novels. Verkkoaineisto. Medium. <<https://medium.com/mammon-machine-zeal/a-brief-history-of-visual-novels-641a2e6b1acb>> Luettu 2.12.2021.
- 5 Totimeki memorial. 2021. Verkkoaineisto. Visual novel database. <<https://vndb.org/v19>> Päivitetty 6.1.2021. Luettu 11.12.2021.
- 6 Shizuku. 2021. Verkkoaineisto. Giantbomb. <<https://www.giantbomb.com/shizuku/3030-69917/>> Päivitetty 24.3.2021. Luettu 11.12.2021.
- 7 Leaf visual novel series. 2021 Verkkoaineisto. Giant bomb. <<https://www.giantbomb.com/leaf-visual-novel-series/3025-4357/>> Päivitetty 10.3.2021. Luettu 2.12.2021.
- 8 Chu, Ken. 2000. Eve Burst Error. Verkkoaineisto. RPG fan. <<https://www.rpgfan.com/review/eve-burst-error-2/>> Luettu 2.12.2021.
- 9 EVE Burst Error. 2020. Verkkoaineisto. Giantbomb. <<https://www.giantbomb.com/eve-burst-error/3030-2706/>> Päivitetty 14.5.2020. Luettu 11.12.2021.
- 10 YU-NO: a girl who chants love at the bound of this world download full game pc for free. Verkkoaineisto. Gaming Beasts. <<https://gaming-beasts.com/yu-no-a-girl-who-chants-love-at-the-bound-of-this-world-download-full-game-pc/>> Luettu 3.12.2021.
- 11 A.D.M.S. 2019. Verkkoaineisto. YU-NO Wiki. <<https://yu-no.fandom.com/wiki/A.D.M.S.>> Luettu 3.12.2021.
- 12 Popular Genres. 2019. Verkkoaineisto. Gettysburg. <<https://emmarie.sites.gettysburg.edu/visual-novels/most-popular-genres/>> Luettu 29.11.2021.

- 13 To Heart. 2021. Verkkoaineisto. Visual novel database. <<https://vndb.org/v18>> Päivitetty 6.3.2021. Luettu 11.12.2021.
- 14 Utsuge. Verkkoaineisto. TV tropes. <<https://tvtropes.org/pmwiki/pmwiki.php/Main/Utsuge>> Luettu 22.1.2021.
- 15 Doki Doki Literature Club. 2021. Verkkoaineisto. Visual novel database. <<https://vndb.org/v21905>> Päivitetty 24.10.2021. Luettu 11.12.2021.
- 16 What is Ren'py. Verkkoaineisto. Ren'py. <<https://www.renpy.org/>> Luettu 22.11.2021.
- 17 Vincent, Brittany. 2020. The best engines for making your own visual novel. Verkkoaineisto. PC Gamer. <<https://www.pcgamer.com/the-best-visual-novel-engines/>> Luettu 3.12.2021.
- 18 TyranoBuilder visual novel studio. 2015. Verkkoaineisto. Tyranobuilder. <<https://tyranobuilder.com/>> Luettu 3.12.2021.
- 19 D'Argenio, Angelo. 2018. Top 10 games that break the 4th wall. Verkkoaineisto. Newegg. <<https://www.newegg.com/insider/top-10-games-that-break-the-4th-wall/>> Luettu 30.3.2022.
- 20 Carlton, Tamika. 2018. Breaking the fourth wall: the ultimate guide to storytelling. Verkkoaineisto. Bottle rocket media. <<https://bottlerocketmedia.net/the-power-of-breaking-the-fourth-wall/>> Luettu 30.3.2022.
- 21 Schuetz, Jordan. 2018. Comparing Game Engines: Unity vs Unreal vs the Rest. Verkkoaineisto. Pubnub. <<https://www.pubnub.com/blog/comparing-game-engines-unity-unreal-corona-gamemaker/>> Luettu 1.5.2022.
- 22 Tyranobuilder visual novel studio. Verkkoaineisto. SteamDB. <<https://steamdb.info/app/345370/>> Luettu 5.4.2022.
- 23 All done time to export. 2015. Verkkoaineisto. Tyranobuilder. <<https://tyranobuilder.com/9-all-done-time-to-export/>> Luettu 4.5.2022.
- 24 Ho, Don. What is notepad++. Verkkoaineisto. <<https://notepad-plus-plus.org/>> Luettu 30.3.2022.
- 25 Benefits of game art outsourcing. 2017. Verkkoaineisto. Juego studios. <<https://www.juegostudio.com/blog/benefits-game-art-outsourcing>> Luettu 18.4.2022.

- 26 The benefits of outsourcing game development. Verkkoaineisto. Starloop studios. <<https://starloopstudios.com/the-benefits-of-outsourcing-game-development/>> Luettu 18.4.2022.
- 27 History of CYOA. Verkkoaineisto. Chooseco. <<https://www.cyoa.com/pages/history-of-cyoa>> Luettu 20.4.2022.
- 28 Gad, Joshua. 2019. About - QA in the Game Industry. Verkkoaineisto. Medium. <<https://joshuabgad.medium.com/about-qa-in-the-game-industry-709903d780b6>> Luettu 9.5.2022.
- 29 Present and future. Verkkoaineisto. Gettysburg. <<https://emmarie.sites.gettysburg.edu/visual-novels/future/>> Luettu 27.3.2022.
- 30 Trevino, Ashlie. 2021. How visual novels could shape the future of storytelling. Verkkoaineisto. Screenrant <<https://screenrant.com/visual-novel-storytelling-industry-diversity-indie-game-developer/>> Luettu 2.4.2022.
- 31 Wallace, Kimberly. 2014. The emerging visual novel trend: what does the future hold. Verkkoaineisto. Gameinformer. <<https://www.gameinformer.com/b/features/archive/2014/01/31/the-emerging-visual-novel-trend-what-does-the-future-hold.aspx>> Luettu 2.4.2022.
- 32 Visual novel / Nine hours, nine persons, nine doors. Verkkoaineisto. TV Tropes. <<https://tvtropes.org/pmwiki/pmwiki.php/VisualNovel/NineHoursNinePersonsNineDoors>> Luettu 2.4.2022.
- 33 Favis, Elise. 2019. The rise and fall of Telltale games. Verkkoaineisto. Gameinformer. <<https://www.gameinformer.com/feature/2019/04/09/the-rise-and-fall-of-telltale-games>> Luettu 30.3.2022.
- 34 Tokyo Chronos. 2022. Verkkoaineisto. Visual novel database. <<https://vndb.org/v23516>> Päivitetty 13.2.2022. Luettu 30.3.2022.