

KARELIA-AMMATTIKORKEAKOULU
Tietojenkäsittelyn koulutusohjelma

Janne Romppanen 1101096

VISUAALISEN NOVELLIN KEHITYSTYÖKALUT LÄNSIMAISSA

Opinnäytetyö
Huhtikuu 2015



OPINNÄYTETYÖ
Huhtikuu 2015
Tietojenkäsittelyn koulutusohjelma

Tikkarinne 9
80220 JOENSUU
Keskuksen puhelinnumero

Tekijä(t)
Janne Romppanen

Nimeke
Visuaalisen novellin kehitystyökalut länsimaissa
Toimeksiantaja
PKAMK ja Karelia-amk

Tiivistelmä
Tässä opinnäytetyössä tutkittiin, mitä visuaaliset novellit ovat ja millaisia visuaalisiin novelleihin erikoistuneita kehitystyökaluja on olemassa länsimaissa. Opinnäytetyön tavoitteena on antaa tietoa, jonka perusteella voi valita sopivan työkalun visuaalisen novellin kehittämiseen.

Sopivan työkalun valinnassa on tutkittu länsimaissa tarjolla olevia työkaluja. Työkalut on luokiteltu aktiivisiin ja ei-aktiivisiin. Työkalujen toimivuutta on tarkasteltu eri kehitystavoilla ja laitealustoilla. Tarkasteltavina olivat tekstieditorilla kehitettävä itsenäinen sovellus (standalone), tekstieditorilla kehitettävä web-sovellus sekä graafisella editorilla kehitettävä itsenäinen sovellus. Jokaisesta kategoriasta on valittu yksi työkalu, jolla on kehitetty sama visuaalisen novellin prototyyppi. Työkaluja on vertailtu toisiinsa prototyypin kehityksestä saatujen kokemusten perusteella.

Tuloksesi saatiin suositeltava työkalu jokaiselle kategorialle. Näitä ovat Ren'Py tekstieditorilla kehitettävässä itsenäisessä sovelluksessa, Belle tai WebStory Engine, tekstieditorilla kehitettävässä web-sovelluksessa ja Novelty graafisella editorilla kehitettävässä itsenäisessä sovelluksessa. IT-ala kehittyy nopeasti ja uusia kehitystyökaluja syntyy, joten kehitystyökalujen tarjonnan tutkiminen ja niiden testaaminen valintaa tehtäessä on tärkeää.

Kieli
suomi

Sivuja 43

Asiasanat
opinnäytetyö, visuaalinen novelli, kehitystyökalut, prototyyppi, vertailu



THESIS
April 2015
All Degree Programmes

Tikkarinne 9
80220 JOENSUU
FINLAND
Telephone number of the centre

Author (s)
Janne Romppanen

Title
Western Visual Novel Development Tools
Commissioned by NKUAS and KUAS

Abstract
The aim of this thesis was to study, what visual novels are and what kind of specialized tools for visual novel development exist in the western world. The goal of this thesis is to provide information which can be used for choosing the right tool for visual novel development.
For choosing the right tool, available in western world are studied. The tools are categorized in active and non-active categories. The functionality of the tools is studied with different development methods and platforms. The development methods included a standalone application developed with text editor, web-application developed with text editor and a standalone application developed with WYSIWYG-editor (What You See Is What You Get -editor). One tool is chosen from each category, which is used for developing a similar visual novel prototype. Experiences gained from developing a prototype with each tool are compared with each other.
As a result there is a recommended tool for each category. These tools are Ren'Py for developing a standalone application with text editor, Belle or WebStory Engine for developing a web-application with text editor and Novelty for developing a standalone application with WYSIWYG-editor. IT-field develops fast and new development tools are published, so studying current availability of tools and testing them when making a selection for a developing tool is important.

Language
Finnish

Pages 43

Keywords
thesis, visual novel, development tools, prototype, comparison

Sisältö

| | | |
|-------|---|----|
| 1 | Johdanto | 3 |
| 2 | Yleistä visuaalisista novelleista | 4 |
| 3 | Kehitystyökalut..... | 8 |
| 3.1 | Aktiiviset työkalut | 9 |
| 3.1.1 | Ren'py | 10 |
| 3.1.2 | Belle..... | 12 |
| 3.1.3 | WebStory Engine..... | 14 |
| 3.2 | Ei-aktiiviset työkalut | 16 |
| 3.2.1 | Novelty..... | 16 |
| 3.2.2 | JS-Vine | 18 |
| 3.2.3 | Visual Novel Toolkit | 20 |
| 3.3 | Japaninkielisiä kehitysympäristöjä | 20 |
| 3.3.1 | NScripter / ONScripter / ONScripter-EN / PONScripiter | 21 |
| 3.3.2 | KiriKiri (KiriKiri2/KAG3) | 21 |
| 3.4 | Yhteenveto työkaluista..... | 22 |
| 4 | Prototyyppi..... | 23 |
| 4.1 | Ren'py | 25 |
| 4.2 | WebStory Engine..... | 28 |
| 4.3 | Visual Novel Toolkit | 30 |
| 4.4 | Novelty..... | 31 |
| 5 | Vertailu..... | 32 |
| 6 | Pohdinta..... | 37 |

1 Johdanto

Tämän opinnäytetyön tarkoituksena on tutkia, millaisia visuaalisten novellien kehittämiseen erikoistuneita työkaluja on tarjolla länsimaissa. Tarkastelussa ovat mukana vain ilmaiseksi tarjolla olevat työkalut. Tämä rajaus tosin jätti tutkimuksen ulkopuolelle vain yhden työkalun, josta olen kertonut luvussa 6. Rajaus länsimaisiin työkaluihin jätti tutkimuksen ulkopuolelle lähinnä japanilaiset työkalut. Jouduin tekemään tämän rajauksen, koska tietoa ei ole saatavilla tarpeeksi kieltä osaamattomalle, jotta työkaluja voisi tutkia tarpeeksi syvällisesti. Visuaaliset novellit ovat kuitenkin syntyneet Japanissa ja ovat siellä erittäin suosittuja, joten olen esitellyt joitakin yleisiä japanilaisia työkaluja luvussa 3.3.

Suosion kasvun myötä olen itsekin vasta tutustunut visuaalisiin novelleihin. Visuaalisen novellin formaatti yhdistää mielestäni hienolla tavalla kertomakirjallisuuden tarinallisen syvyyden ja pelien interaktiivisuuden tarinan ja pelaajan välillä. Halu tutkia formaattia syvällisemmin kehittäjän näkökulmasta on suurin syy, että olen valinnut sen opinnäytetyön aiheeksi. Tämä opinnäytetyö onkin suunnattu etenkin sellaisille lukijoille, jotka ovat vasta tutustumassa visuaalisiin novelleihin. Tutustuminen on hyvä aloittaa luvusta 2, jossa olen kertonut mitä visuaaliset novellit ovat ja mistä elementeistä ne muodostuvat. Jos haluaa tietää millaisia työkaluja on tarjolla, kannattaa silloin lukea luku 3. Lisäksi luvussa 6 on mainittu kaksi potentiaalista mutta tutkimuksen ulkopuolelle jäänyttä työkalua. Jos on kiinnostunut visuaalisen novellin kehittämisestä, kannattaa silloin lukea luku 4, jossa tarkastelen kehittämäni prototyypin kehitystä valituilla työkaluilla. Silloin on hyvä lukea myös luku 5, jossa olen tehnyt vertailua valitsemieni prototyypin kehitystyökalujen välillä.

Tutkimuskysymyksinä olen käyttänyt seuraavia kysymyksiä:

- Mitä ovat visuaaliset novellit?
- Millaisia länsimaisia visuaalisten novellien kehitystyökaluja on olemassa?
- Kuinka visuaalisen novellin kehittäminen niillä tapahtuu?
- Kuinka kehitystyökalut eroavat toisistaan?

2 Yleistä visuaalisista novelleista

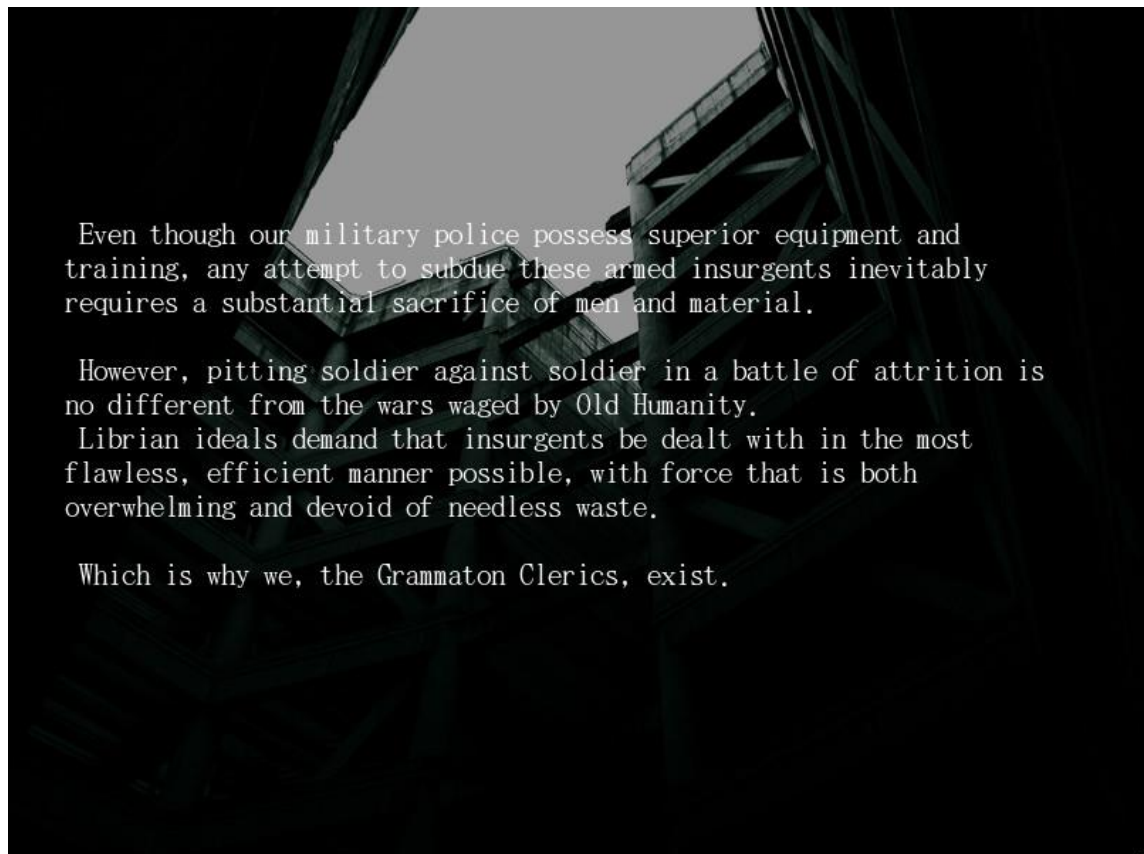
Visuaaliset novellit ovat tarinavetoisia pelejä. Novellin tyypistä riippuen pelaaja voi vaikuttaa tarinan kulkuun. Kuvassa 1 on esitetty perinteinen visuaalisissa novelleissa käytetty tyyli. Perinteisessä tyyliässä kohtaus muodostuu taustakuvasta, kohtauksen näyttelijöistä sekä tekstilaatikosta, jossa kohtauksen tekstit näytetään. Pelaajahahmoa ei näytetä vaan tarina kuvataan pelaajan näkökulmasta. Tarinalla voi olla kertoja, jota ei myöskään näytetä. Myös pelimenu voi olla esillä kuten kuvan 1 tekstilaatikon alalaidassa (The visual novel database 2015a).



Kuva1. Perinteinen tyyli / ADV-tila (Ren'Py 2015a).

Tarina näytetään kahdessa eri tilassa ADV (adventure style game) tai NVL (novel style game). Niiden välinen ero on siinä, kuinka tarinan teksti näytetään. Perinteisesti novelli esitetään ADV-tilassa, jolloin teksti näytetään ruudun alalaidassa olevassa tekstilaatikossa lause kerrallaan, kuten kuvassa 1. Tämän seurauksena tarinan ja pelaajan välinen vuorovaikutus on suuri. Myös pelin visuaalinen puoli on korostunut, koska se vie suuremman osan näytön tilasta kuin

tarinan teksti. Yleensä kohtauksen tietokonenäyttelijöiden tekstit näytetään tekstilaatikossa ja pelaajan tekstit valintalistana, jolloin se toimii samalla tarinan hahmoittajana. NVL-tilassa teksti näytetään koko ruudun pituudelta, kuten kuvassa 2. Tällöin teksti on pääosissa, visuaalisen puolen jäädessä tekstin taakse. Tarinan ja pelaajan välinen vuorovaikutus on vähäisempää kuin ADV-tilassa. (Wikipedia 2015a.)



Kuva 2. NVL-tila (The Visual Novel Database 2015b).

Tilojen käytön jakautumisesta ei ole tehty tilastoa. Molemmilla tyyleillä on tehty suosittuja visuaalisia novelleja. ADV-tilasta esimerkkinä mainittakoon Clannad, joka on rankattu seitsemänneksi suosituimmaksi The Visual Novel Database -sivustolla (The Visual Novel Database 2015c). NVL-tilasta esimerkkinä mainittakoon Fate/Stay Night, joka on rankattu suosituimmaksi samalla sivustolla (The Visual Novel Database 2015d). Tiloja voidaan myös yhdistellä, jolloin ADV-tilaa käytetään vuorovaikutteisissa kohtauksissa, kuten dialogeissa, ja NVL-tilaa silloin, kun vuorovaikutusta ei ole, esimerkiksi monologeissa. Tällaista yhdistel-

mää käyttää esimerkiksi True Remembrance (The Visual Novel Database 2015e).

Pelaaja vaikuttaa tarinan kulkuun vastaamalla hänelle esitettyihin kysymyksiin, kuten kuvassa 3. Kysymykset voivat olla kohtauksen näyttelijöiden esittämiä kysymyksiä tai pelitilanne, johon pelaajan täytyy reagoida. Valinnat vievät tarinan eri haaroihin mahdollistaen useita eri reittejä tarinan läpäisyyn. Yleensä tarina pitää pelata läpi useita kertoja, että tarina avautuu kokonaisuudessaan. (Giant Bomb 2015.)



Kuva 3. Vuorovaikutus (Ren'Py 2015a).

Yksi visuaalisten novellien kerrontatyyli on ns. Kinetic novel. Siinä pelaajan ja tarinan välillä ei ole laisinkaan vuorovaikutusta. Tarina on näin ollen täysin lineaarinen ja pelaajan tehtävänä on vain lukea tarina. (Wikipedia 2014a.)

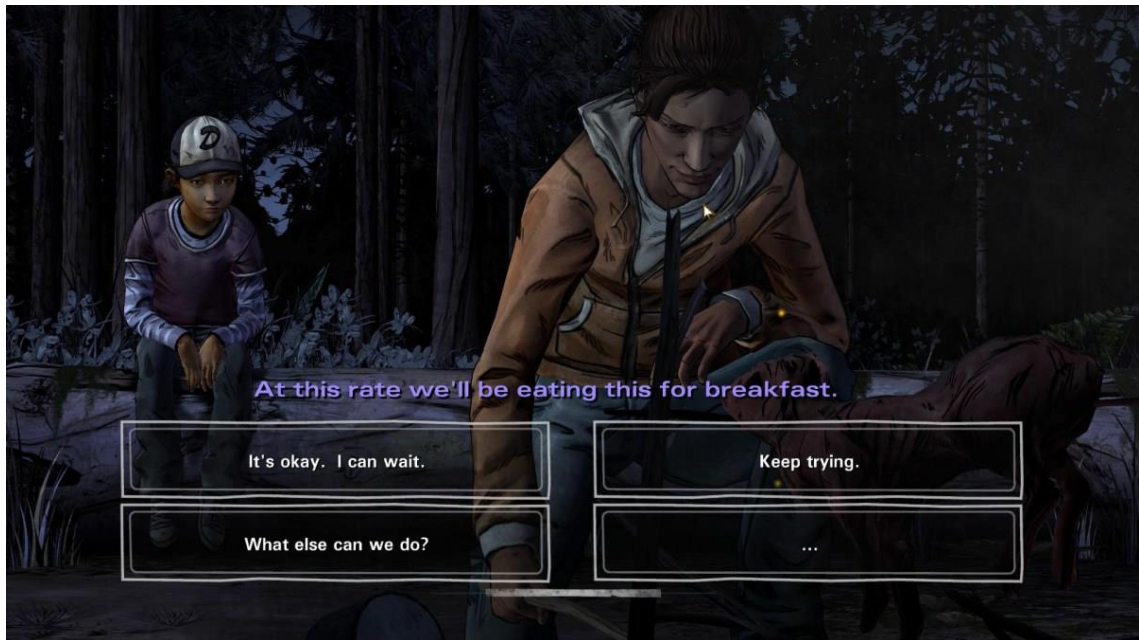
Visuaalisissa novelleissa voi olla käytetty myös muiden peligenrejen elementtejä. Yleisimpiä lisäelementtejä ovat mm. seikkailupelien puzzlet, roolipelien hahmonkehitys sekä vuoropohjaiset taistelut (kuva 4). Lisäksi novelleissa voi olla

karttanäkymä pelialueella liikkumiseen ja inventaario pelaajan tavaroiden hallinnoimiseen.



Kuva 4. Taistelukohtausta visuaalisesta novellista Planet Stronghold (Ren'Py Games List 2015).

Mitään virallista määritelmä ei ole tehty, milloin visuaalisesta novellista tulee esimerkiksi roolipeli. Visuaalisessa novellissa pääpaino on kuitenkin laajalla tekstipainotteisella tarinalla sekä tausta- ja hahmokuville, jossa pelaaja vaikuttaa tarinaan tekstin kautta (The visual novel database 2015a). Muunlaiset sisällölliset elementit tai muunlainen pelaajan tarinaan vaikuttaminen voidaan mielestäni laskea visuaalisen novellin genren ulkopuoliseksi ominaisuudeksi. Muiden peligenrejen pelit voivat käyttää myös visuaaliselle novellille tyypillisiä elementtejä. Esimerkkinä mainittakoon kuvassa 5 näkyvän The Walking Dead -pelin dialogin toteutus.



Kuva 5. The Walking Dead-pelin dialogi (Et Keegera 2014).

Eri tyyliä tehtyjä visuaalisia novelleja löytyy esimerkiksi osoitteessa <http://games.renpy.org>. Sivun novelleja voi selata mm. kategorioittain ja etsiä esim. pelkkiä roolipelejä. Kattava lista visuaalisista novelleista on koottu sivustolle <https://vndb.org>. Sivustolla voi etsiä erityylisiä novelleja tagien avulla. Käyttäjät voivat myös pisteyttää novelleja, mikä helpottaa valintaa valtavasta tarjonnasta.

3 Kehitystyökalut

Edellisessä luvussa esittelin visuaalisen novellin formaatin yleisellä tasolla. Tässä luvussa keskityn visuaalisten novellien tekemiseen erikoistuneisiin työkaluihin. Työkalut ovat yleensä genren faniin omalla vapaa-ajallaan kehittämiä. Koska työkaluja ei kehitetä niinkään ammattimaisesti, työkalujen laatu, niiden tuki sekä käyttöaste vaihtelevat. Olen yrittänyt poimia laadukkaimmat työkalut ja suurimman käyttäjäkunnan omaavat työkalut. Laadukkaimmilla työkaluilla tarkoitan sellaisia työkaluja, joilla voi tehdä visuaalisen novellin kriteerit täyttävän

sovelluksen. Koska visuaaliset novellit ovat saaneet alkunsa Japanista, esittelen myös japanilaisia työkaluja lyhyesti.

Olen jaotellut työkalut kolmeen kategoriaan: aktiivisiin, ei-aktiivisiin ja japaninkielisiin. Erottelu aktiivisiin ja ei-aktiivisiin on tehty, jotta lukija on tietoinen valintaa tehdessään, onko työkalulla aktiivista tukea tai kehitysyhteisöä, onko päivityksiä ja kehitystä odotettavissa vai onko työkalu jäänyt ns. oman onnensa nojaan mutta on vielä kuitenkin täysin soveltuva sille annettuun tehtävään. Japaninkieliset työkalut on myös eroteltu omaksi luvukseen. Vaikkei japaninkielisistä kehitysympäristöistä ole suoranaista hyötyä kieltä osaamattomalle, haluan kuitenkin mainita pari suosituinta kehitysympäristöä Japanissa olevan suuren fanipohjan takia. Vähäisen englanninkielisen tiedon takia olen jättänyt pois niiden tarkemmat ominaisuudet.

Kaikista työkaluista on esitelty tärkeimmät ominaisuudet omassa taulukoissaan. Keskeisimpiä ominaisuuksia ovat seuraavat seikat: onko kyseessä kokonaisvaltainen kehitysympäristö vai pelkkä moottori, millä ohjelmointikielillä moottori on kehitetty, mitä ohjelmointikieltä ja skriptikieltä työkalut tukevat visuaalisen novellin kehityksessä, mitä käyttöjärjestelmiä ne tukevat ja mille alustoille visuaalisen novellin voi julkaista.

Lisäksi tarkastelen sitä, mihin käyttötarkoituksiin kyseinen työkalu soveltuu. Esittelen myös työkalun kehittäjän sekä millä lisenssillä kyseinen työkalu on julkaistu. Lopuksi vertailen työkaluja keskenään ja kerron oman näkemykseni parhaiten eri tarkoituksiin soveltuvista työkaluista.

3.1 Aktiiviset työkalut

Tässä luvussa on esitelty tällä hetkellä aktiivisessa kehityksessä olevat työkalut. Olen määritellyt työkalun aktiiviseksi, jos kehittäjä sitä edelleen aktiivisesti kehittää, toisin sanoen työkalusta on julkaistu uusi päivitys viimeisen puolen vuoden aikana, tai jos kyseessä on avoimen lähdekoodin työkalu, jolla on aktiivinen kehitysyhteisö.

3.1.1 Ren'py

Ren'py on Tom "PyTom" Rothamelin kehittämä kehitysympäristö visuaalisten novellien kehittämiseen. Ensimmäinen versio julkaistiin 24.8.2004. Taulukossa 1 esitetty versionumero 6.18.1 on julkaistu 22.9.2014. Pitkä kehityskaari näkyy mm. kehitysympäristöjen tehtyjen julkaisujen määrässä, joita on yli 700 kappaletta. (Wikipedia 2014b.) Taulukossa 1 esitetystä Genret-listasta nähdään, että Ren'pyllä voidaan tehdä monen tyyliä julkaisuja. Julkaisut kattavat mm. simulaatiot, vuoropohjaisen taistelun ja roolipelit (Ren'py 2014c).

Taulukko 1. Ren'pyn ominaisuuksia (Ren'py 2014c, Ren'py 2014d).

| | |
|--------------------|---|
| Kehitysympäristö | |
| Versionumero | 6.18.1 |
| Julkaisupäivämäärä | 22.9.2014 |
| Pääominaisuudet | Pythonilla toteutettu pelimoottori Python-skriptikielellä määriteltävät novellin tekniset elementit Ren'pyn omalla skriptikielellä määriteltävät novellin tarinalliset elementit Ren'py Launcher. Graafinen työkalu projektien luomiseen ja hallintaan |
| Formaatti | itsenäinen tiedosto |
| Käyttökohteet | visuaaliset novellit kineettiset novellit (Kinetic novels) roolipelit simulaatiot sekalaiset |
| Kehitysalusta | Windows XP+ (x86) Mac OS X 10.6+ (x86_64) Linux (x86, x86_64) |

| | |
|----------------|---|
| | Sekondäärinen tuki: Android (2.0+, OpenGL ES 2.0 tuella) |
| Julkaisualusta | sama kuin kehitysalusta |

Ren'pyn päätyökalu on Ren'py Launcher. Se on graafinen projektien hallintatyökalu, jolla voi luoda uusia projekteja, hallita ja muokata jo olemassa olevia sekä suorittaa niitä. (Ren'py 2014e.) Uusi projekti sisältää tarvittavat skriptitiedostot julkaisun asetusten ja sisällön määrittelyyn. Ren'py Launcher sisältää myös mm. kääntäjän (compiler) koodin kääntämiseen, käännöstyökalun tekstisisällön kääntämiseen eri kielille sekä julkaisutyökalun taulukossa 1 määritellyille julkaisualustoille.

Ren'pyyn on kehitetty oma Pythoniin pohjautuva skriptikieli, joka vaikuttaa pohjautuvan Python-skriptikieleen (Ren'py 2014f). Ren'pyssä on Pythonille myös tuki, jolla voidaan kehittää moottorin ulkopuolisia toiminnallisuuksia. Ren'pyyn on kehitetty myös näyttökieli (screen language). Näyttökielellä määritellään julkaisun käyttöliittymät kuten päävalikko, tallennusvalikko tai puhevalikko. (Ren'py 2014g.)

Ren'pyn lähdekoodi on pääosin MIT-lisenssin alaista (Ren'py 2014d). Se on avoimen lähdekoodin ohjelmisto ja sen käyttö on ilmaista kaupalliseen ja eikaupalliseen käyttöön (Ren'py 2014b).

Yhteenvedona voi todeta, että Ren'py on avoimen lähdekoodin kehitysympäristö monelle käyttöjärjestelmälle. Siinä on tehokkaat skriptaustyökalut ja siitä löytyy kattava valikoima työkaluja visuaalisissa novelleissa yleisesti käytettävien ominaisuuksien tekoon. Tiedostotuki kattaa yleisimmät tyypit mukaan lukien videotiedostot. Lisäksi Ren'py on dokumentoitu hyvin sekä sillä on aktiivinen kehitysyhteisö. (Ren'py 2014d.) Ren'Py onkin mielestäni erittäin suositeltava vaihtoehto itsenäisten (standalone) julkaisujen työkaluksi.

3.1.2 Belle

Belle on Carlos Paisin kehittämä kehitysympäristö visuaalisten novellien kehittämiseen (Carlos 2013a). Ensimmäinen versio (0.1a) editorista ja moottorista on julkaistu 15.11.2012 (Carlos 2013b). Taulukossa 2 esitetystä versionumerosta 0.5 ei ole virallisesti ilmoitettua julkaisupäivämäärää. Viimeisin tiedostojen muokkauspäivämäärä, pelimoottorin osalta, on päivätty 13.2.2015. (Belle 2015a). Bellen GitHub-säiliöön (repository) on tehty viimeisin muokkaus 25.2.2015 (tarkistettu 27.3.2015). (GitHub 2015.) Bellen kotisivuilla kehittäjä kertoo kehittävänsä Belleä vapaa-ajallaan, joten tämä selittää mielestäni miksi viimeisimmät muokkaukset ovat yli kuukauden vanhoja. (Belle 2015b.) Voimmekin todeta, että Belle on aktiivisessa kehityksessä.

Taulukko 2. Bellen ominaisuuksia (Belle 2015a; GitHub 2015; Belle 2015c).

| | |
|--------------------|--|
| Kehitysympäristö | |
| Versionumero | 0.5 |
| Muokkauspäivämäärä | 12.9.2014 |
| Pääominaisuudet | C++:lla ja QT4:llä toteutettu WYSIWYG-editori HTML 5:llä, CSS:llä ja JavaScriptillä toteutettu pelimoottori |
| Formaatti | Web-sovellus |
| Käyttökohteet | visuaaliset novellit |
| Kehitysalusta | Windows XP+ GNU / Linux Mac OS X (10.4+) |
| Julkaisualusta | verkkoselain online- tai offline-tilassa |

Bellen kehityksessä on otettu esimerkkiä Noveltystä, joka on samantyyppinen kehitysympäristö. (Belle 2014c.) Novelty on esitelty luvussa 3 ”Ei-aktiiviset työkalut”. Bellen päätyökalu on WYSIWYG (What You See Is What You Get) editori, jossa kehitystyö tapahtuu. Editorilla voi lisätä graafiset elementit kuten pai-

nikkeet (buttons) ja dialogi-ikkunat. Tarinan sisältö tehdään lisäämällä toimintoja (actions) kohtauksiin. Toimintoja ovat esim. dialogi, tarinan haaroitus tai scriptin suorittaminen. Bellessä on myös tuki JavaScriptille, jolla voidaan toteuttaa editorin ulkopuolisia asioita. (Belle 2015c.) Kehittäjän sivuilla ei ole tietoa Bellellä kehitetyistä visuaalisista novelleista. Koska Belle on vielä sen verran varhaisessa kehitysvaiheessa, voidaan olettaa, ettei sillä ole vielä tehty merkittäviä julkaisuja.

Belle on julkaistu GNU GPLv3-lisenssillä. Se on avoimen lähdekoodin ohjelmisto ja sen käyttö on ilmaista kaupalliseen ja ei-kaupalliseen käyttöön. (Belle 2014c.)

Tiedostotuki kattaa yleisimmät tyypit mukaan lukien videotiedostot. Se soveltuu web-sovellusten kehitykseen visuaalisella editorilla. Vaikka Belle onkin vasta alpha-versiossa, siitä löytyvät työkalut yleisimpien visuaalisen novellin ominaisuuksien kehitykseen. Bellen kotisivuilla ei ole forumia, mutta sillä on oma viestiketju Lemma Soft Forumilla, joka on käytetyin kehitysfoorumi visuaalisen novellin kehittäjien parissa. (Lemma Soft Forum 2015a.) Sieltä on mahdollista saada apua kehitystyössä. Lisäksi Bellen GitHub-säiliöön voi lisätä virheraportteja (issues). (Belle 2015d.) Kehittäjälle voi myös lähettää sähköpostia, kotisivuilla ilmoitettuun osoitteeseen. (Belle 2015b.) Bellen dokumentaatio kattaa peruskäytön, mutta kotisivuilla ilmoitetaan sen olevan vanhentunutta. (Belle 2015b.).

Mielestäni Belle on varteenotettava vaihtoehto, jos halutaan kehittää visuaalisia novelleja web-sovelluksen muodossa. Etenkin, jos kehitystyö halutaan tehdä graafisella editorilla, on Belle silloin ainoa vaihtoehto, koska web-sovellusten kehitykseen ei ole julkaistu muita graafisia editoreja. Kannattaa kuitenkin muistaa, että Belle on vielä kehitysvaiheessa. Kuten kehittäjäkin on sivuillaan varoitannut, ongelmia voi ilmetä. (Belle 2015b.) Lisäksi ongelmana on jo mainittu vanhentunut dokumentaatio. Dokumentaatio on tarkastettu 27.3.2015. Lukuhetkellä se on voitu jo päivittää ajan tasalle.

3.1.3 WebStory Engine

WebStory Engine on Jonathan Steinbeckin kehittämä moottori visuaalisten novellien kehittämiseen. Ensimmäisen version julkaisupäivämäärä ei ole selvillä, mutta ensimmäinen lisäys (commit) WebStory Enginen GitHub-säiliöön (repository) on tehty 26.6.2012 (GitHub 2012). Taulukossa 3 esitetty versionumero on julkaistu jo melkein vuosi sitten. Toisaalta viimeisin lisäys GitHub-säiliöön on tehty 4.11.2014 (GitHub 2014). Lisäksi viimeisin viesti moottorin kotisivujen foorumille on lisätty 19.3.2015. (tarkistettu 27.3.2015) (iiYO 2015). Lisäksi kotisivuilta löytyy pohdintaa seuraavan sukupolven WebStory Engine-moottorin ominaisuuksista. Siinä on pohdittu nykyisen moottorin huonoja puolia ja esitetty kehittämideoita. (WebStory Engine 2014a; WebStory Engine 2014b.) Kaikki nämä seikat puhuvat sen puolesta, että moottori on edelleen aktiivisessa kehityksessä ja sillä on vahva suuntautuminen tulevaisuuteen.

Taulukko 3. WebStory Enginen ominaisuuksia (WebStory Engine 2014c; Lemma Soft Forums. 2015b).

| | |
|--------------------|--|
| Moottori | |
| Versionumero | 0.3.7 |
| Julkaisupäivämäärä | 1.11.2013 |
| Pääominaisuudet | HTML:llä, CSS:llä, XML:llä ja JavaScriptillä toteutettu pelimoottori oma XML-pohjainen merkkaukieli novellien sisällön tuottamiseen |
| Formaatti | Web-sovellus |
| Käyttökohteet | visuaaliset novellit esitelmät interaktiiviset tutoriaalit |
| Kehitysalusta | kaikki HTML, CSS, XML ja JavaScript tuen omaavat käyttöjärjestelmät |
| Julkaisualusta | verkkoselain online- tai offline-tilassa |

WebStory Enginen moottori on kirjoitettu JavaScript-kielellä. Moottori käyttää XML-kielellä määriteltyjä elementtejä, joista varsinainen julkaisu koostuu. Elementtien avulla määritellään julkaisun asetukset ja sisältö yhteen XML-tiedostoon, joka linkitetään HTML-indeksisivulle. Varsinainen julkaisu näytetään verkkoselaimessa indeksisivulla. XML-tiedosto on jaoteltu kolmeen osaan: asetukset (settings), resurssit (assets) ja kohtaukset (scenes). Asetukset sisältävät novellin teknisiä määrittelyjä kuten leveyden ja korkeuden sekä näppäinmäärittelyt. Resurssit sisältävät käytössä olevien sisällöllisten resurssien (esim. kuva ja äänitiedostot) määrittelyt kuten tiedostopolun resurssiin ja resurssin nimen. Kohtaukset käyttävät resursseja sekä niihin määritellään varsinainen leipäteksti. Kohtauksia voi olla useita ja ne erotellaan omiksi kohtauselementeiksi. Kohtauksiin myös määritellään ehdot, kuinka tarina etenee. (WebStory Engine 2014d; WebStory Engine 2014e.)

WebStory Enginen kotisivuilla ei ole tarkemmin kerrottu moottorin ominaisuuksista. Sivuilla olevassa oppaassa on esimerkiksi kohta Animations, mutta siihen ei ole lisätty vielä sisältöä (WebStory Engine 2014f). Lukijalle jää vain oletus, että moottori ehkä pystyy käsittelemään animaatioita jollain määrittämättömällä tavalla. Tämä on mielestäni suuri puute ja kertoo työkalun keskeneräisyydestä.

WebStory Engine on julkaistu BSD 3-Clause-lisenssillä (GitHub 2014d). Salliva lisenssityyppi mahdollistaa moottorin lähdekoodin muokkaamisen sekä uusien toiminallisuuksien lisäämisen, että myös uusien XML-elementtien luomisen. WebStory Enginen kotisivuilla on lista julkaistuista julkaisuista (WebStory Engine 2014f). Listassa on vasta kymmenen julkaisua mukaan lukien WebStory Enginen tutoriaali. Tämä osoittaa kyseessä olevan vielä nuori moottori, joka ei ole vielä herättänyt julkaisujen kehittäjien huomiota laajemmin. Nuoren iän lisäksi ainakin graafisen editorin puute voi olla vieraannuttava seikka sellaisille kehittäjille, jotka suosivat graafista kehitystä tekstieditorilla kirjoittamisen sijaan. Kotisivujen mukaan editorin lisäys on toivelistalla, mutta sitä ei ole vielä hyväksytty kehityslistalle (tarkastettu 27.3.2015) (WebStory Engine 2014b).

Yhteenvetona voin todeta, että WebStory Engine on avoimen lähdekoodin moottori usealle eri käyttöjärjestelmälle. Tiedostotukea ei ole mainittu kotisivuil-

la. Työkalu soveltuu web-sovellusten kehitykseen tekstieditorilla työkalun omalla XML-merkintäkielellä. Kehittäjä voi myös kehittää moottorin kieltä ja lisätä omia elementtejään siihen. Kehitystyö soveltuu ohjelmoinnista pitävälle kehittäjille. WebStory Engineä kehitetään aktiivisesti ja kehittäjillä on myös selkeitä tulevaisuuden suunnitelmia. Kotisivuilta löytyy oma foorumi, tosin se ei ole kovin aktiivinen. Visuaalisten novellien kehitystyöhön suuntautuneella Lemma Soft Forumilla WebStory Enginellä on oma sijansa yhteisön keskuudessa. (Lemma Soft Forums 2015b.) Työkalun dokumentointi on vielä vajaata. Esimerkiksi kaikista työkalun ominaisuuksista ei ole tietoa. Mielestäni WebStory Engine sopii kehittäjälle, joka haluaa kirjoittaa koodia moottorin antamassa kehityksessä ja ohjelmoida spesifejä ominaisuuksia juuri kyseistä julkaisua varten. Tukea on saatavilla ainakin Lemma Soft Forumilta.

3.2 Ei-aktiiviset työkalut

Tässä luvussa on mainitut kehitysympäristöt ovat täysin toimivia mutta niiden kehitys ja ylläpito eivät ole enää aktiivista. Työkalua voidaan kuitenkin kehittää vielä epäsäännöllisesti ja kehitysyhteisö voi olla yhä olemassa.

3.2.1 Novelty

Novelty on Erik Benerdalin kehittämä kehitysympäristö visuaalisten novellien kehittämiseen. Kehitystyö alkoi sivuprojektina vuonna 2007. Ensimmäinen alpha-versio ilmestyi heinäkuussa 2008. (Benerdal 2014a.) Taulukossa 5 esitetty versio 0.8.9 on julkaistu 14.12.2010. Viimeisin päivitys Novelty kotisivuilla kehittäjän toimesta on forumilla vuodelta 2012. Pääasiallinen uutisointi näyttääkin siirtyneen kehittäjän henkilökohtaiselle Twitter-tilille. (Benerdal 2014b.) Kotisivujen forumilla on viestejä tältä vuodelta ja muutamia kuluvalta kuulta (lokakuu), joten jonkin asteista aktiivisuutta kehitysympäristön parissa on edelleen. Aktiivisuus on kuitenkin sen verran vähäistä ja viimeisestä julkaisusta on jo niin pitkä aika, etten luokittele Noveltyä aktiiviseksi työkaluksi.

Taulukko 5. Noveltyn ominaisuuksia (Novelty 2014a; Novelty 2014b; Novelty 2014c).

| | |
|--------------------|--|
| Kehitysympäristö | |
| Versionumero | Beta 0.8.9. |
| Julkaisupäivämäärä | 14.12.2010 |
| Pääominaisuudet | Olio-ohjelmointimallin mukainen toteutus NoveltyML-merkintäkieli resurssien määrittelyyn olio-ohjelmointimallin mukaiseksi olioksi WYSIWYG-editori graafinen ohjelmointirajapinta tuki AngleScript-skriptauskielelle C++:lla ja Javalla toteutettu pelimoottori |
| Formaatti | itsenäinen tiedosto |
| Käyttökohteet | visuaaliset novellit |
| Kehitysalusta | Windows XP / Vista / 7 vaatii DirectX 9.0c API:n |
| Julkaisualusta | sama kuin kehitysalusta |

Noveltyssä on poikkeava suunnitteluratkaisu muihin työkaluihin verrattuna: se on suunniteltu olio-ohjelmointimallin mukaiseksi. Tämä tarkoittaa, että kaikki julkaisussa käytettävät resurssit määritellään olioiksi NoveltyML-merkintäkielellä. (Novelty 2014a.) Noveltyn päätyökalu on WYSIWYG-editori, jossa kehitystyö tapahtuu. Editorilla hallinnoidaan olioita sekä kehitetään tarinalle dialogi ja ehdot. Ehdot (actions) toimivat graafisina ohjelmointikomponentteina, jotka määrittävät tarinan toiminnallisuuden ja etenemislogiikan. Ehtojen lisäksi voidaan luoda omia toiminnallisuuksia AngleScript-skriptikieltä käyttäen. (Novelty 2014d; Novelty 2014b.)

Noveltyllä tehtyjen novellien määrästä ei ole virallista lukua mutta Noveltyn kotisivujen forumilla on oma keskustelualue, jossa voi esitellä Noveltylla kehitettyjä

novelleja. Keskustelualueella on yli 120 aihetta. (Novelty 2014c.) Novelty on suljetun lähdekoodin ohjelmisto mutta sen käyttö on ilmaista (Novelty 2014e). Yhteenvetona voin todeta, että Novelty on suljetun lähdekoodin kehitysympäristö Microsoft Windows -käyttöjärjestelmälle. Se soveltuu itsenäisten (standalone) julkaisujen kehitykseen. Tiedostotukea ei ole ilmoitettu kotisivuilla. Noveltyssä on tuki AngelScript-skriptikielellä sekä editorissa on työkalu graafiselle ohjelmoinnille, joten molemmat ohjelmointimuodot on katettu. Novelty on dokumentoitu kattavasti ja kotisivujen forumin julkaisukeskustelusta päätellen sillä on tehty yli sata julkaisua. Viimeisen version julkaisusta on kulunut jo kohta neljä vuotta eikä uusista julkaisuista ole ilmoitettu. Lisäksi lähdekoodi on suljettua, joten ulkopuoliset tahot eivät voi kehittää sitä. Kotisivujen forumilla on viestejä tältä vuodelta mutta vai muutama tältä kuulta (lokakuu). Tukea kehitystyöhön uskoi- sin löytyvän helposti kotisivujen forumilta ja Lemme Soft Forumilta. Novelty on mielestäni vaihtoehto Ren'pylle silloin, kun halutaan ohjelmoida vain graafisesti. Muissa tapauksissa kuten avoimessa lähdekoodissa, käyttöjärjestelmien kattavuudessa ja työkalun kehittäjän sekä kehittäjän yhteisön aktiivisuudessa Ren'py on parempi vaihtoehto.

3.2.2 JS-Vine

JS-Vine on J. David Eisenbergin kehittämä moottori visuaalisten novellien ja esitysten kehittämiseen (Eisenberg 2014a). Hän kehitti moottorin esitelläkseen työtään verkossa visuaalisen novellin muodossa (Eisenberg 2014b). Eisenbergin GitHub-tilin JS-Vine-haaraan tehdyt lisäykset alkoivat kesäkuussa 2011. Viimeisin lisäys on marraskuulta 2011. (Eisenberg 2014c.) Tämän jälkeen kehitystyö näyttää päättyneen.

JS-Vinen moottorina toimii yksittäinen JavaScript-tiedosto (Eisenberg 2014d). Varsinainen julkaisu kehitetään HTML-tiedostoon, joka sijoitetaan samaan kansioon moottorin kanssa. Resursseina toimivat kuva- ja äänitiedostot voidaan lisätä omiin kansioihinsa. HTML-tiedostolle voidaan antaa tyyliomotoiluja CSS-tyyliohjeilla. Tiedostoon lisätään sisältö moottorissa määriteltujen JavaScript-

funktioiden avulla. Moottori tarjoaa funktiot kuvien, tekstikenttien ja valikoiden näyttämiseen, dialogin lisäämiseen sekä äänien toistamiseen. Valikoiden avulla saadaan ohjattua tarinan kulkua. (Eisenberg 2014e.) Tämä antaa valmiin kehityksen julkaisun kehitykseen. Koska moottori on avointa lähdekoodia, voidaan sitä jatkokehittää vapaasti. Lisäksi HTML-tiedostoon voidaan määritellä myös moottorin ulkopuolista toiminnallisuutta JavaScript-skriptikielen avulla. Valmis julkaisu lisätään palvelimelle, jolloin sitä voidaan käyttää verkkoselaimella.

JS-Vine on julkaistu LGPL-lisenssillä (Eisenberg 2014a). Lisenssi tarjoaa käyttäjälle lähdekoodin vapaan muokkaamisen ja hyödyntämisen omassa julkaisussa (Wikipedia 2014c).

Yhteenvetona voin todeta, että JS-Vine on avoimen lähdekoodin moottori monelle käyttöjärjestelmälle. Se on hyvin pelkistetty työkalu, jolla voi koostaa yhteen ääntä, kuvaa ja dialogia sekä käyttäjän valintojen mukaan etenevän dialogipuun. Tiedostotuesta ei ole kerrottu kotisivuilla. JS-Vine on dokumentoitu kattavasti mutta minkäänlaista kehitysyhteisöä ei näytä olevan eikä kotisivuilla ole forumia eikä visuaalisen novellin kehittäjien suosimalla Lemma Soft Forumilta löytynyt keskusteluja JS-Vineen liittyen. JS-Vinellä tehdyistä julkaisuista ei ole tietoa kotisivuilla. Olettaisin ettei sillä ole tehty ainakaan kaupallisia julkaisuja. JS-Vine soveltuu mielestäni yksinkertaisten visuaalisten novellien sekä esitelmien kehitykseen web-julkaisuiksi. Jos työkalulla halutaan tehdä web-julkaisu kattavilla ominaisuuksilla, kannattaa mieluummin valita Belle tai WebStory Engine.

3.2.3 Visual Novel Toolkit

Visual Novel Toolkit (Vino) on japanilaisen Sol-Tribe-yhtiön kehittämä kehitysympäristö Unity-pelimoottorille. Visual Novel Toolkitistä on kaksi eri versiota: ilmainen ja maksullinen. Ilmainen on rajoitettu versio maksullisesta. Maksullisen hinta on tällä hetkellä 30 dollaria Unityn Asset Storessa. (Asset Store 2014a; Asset Store 2014b.) Sol-Triben kotisivuilla ilmoitettu versio 1.1.1. on julkaistu 30.1.2014. Samalla sivulla, version julkaisun yhteydessä, ilmoitetaan Vinon kehityksen lopettamisesta, sekä sen tuen lopettamisesta huhtikuussa 2014. (Sol-Tribe 2014a).

Kun lisäsin Vinon opinnäytetyöhöni, se oli vielä kehityksessä. Valitsin sen myös yhdeksi prototyypin kehitystyökaluksi, edustamaan graafista työkalua, jossa ei käytetä skriptausta. Prototyypin kehityksessä oli kuitenkin niin suuria ongelmia, että jouduin vaihtamaan sen toiseen graafisen työkaluun. Olen kertonut ongelmista kohdassa 4.3. Visual Novel Toolkit. Lisäksi käyttäjät ovat antaneet negatiivista palautetta maksullisesta versiosta palautettu Unityn Asset Storessa. (Asset Store 2015b). En siis halua suositella Visual Novel Toolkitiä missään tapauksessa. Jos kehitystyö halutaan tehdä juuri Unityllä, olen maininnut parista vaihtoehdosta kohdassa 6 Pohdinta.

3.3 Japaninkielisiä kehitysympäristöjä

Tässä luvussa on mainittu muutama suosittu, täysin tai osittain japaninkielinen kehitysympäristö. Olen jättänyt yksityiskohtaiset tiedot pois englanninkielisen tiedon puutteen takia.

3.3.1 NScripter / ONScripter / ONScripter-EN / PONScripter

NScripter on Naoki Takahashin kehittämä moottori visuaalisten novellien kehittämiseen Microsoft Windowsilla. NScripter oli suosituimmillaan yksi käytetyimmistä moottoreista Japanissa. Kehitys on ollut suurelta osin pysähdyksissä vuodesta 2003, versiosta 2.59. NScripter on suljetun lähdekoodin ohjelmisto mutta ilmainen kaupalliseen ja ei-kaupalliseen käyttöön. Lisenssi ei ole selvillä. (Wikipedia 2014d.)

NScripter-moottoria on jatkokehitetty eri klooneiksi. Yksi suosituimmista on japanilaisen ohjelmoijan nimimerkillä Opagee kehittämä ONScripter. Se on avoimen lähdekoodin ohjelmisto monelle eri käyttöjärjestelmälle, pohjautuen SDL-kirjastoon. (Wikipedia 2014d.)

ONScripter-EN on englanninkielisen yhteisön oma kehityshaara OnScripteristä. Kehitystyö keskittyy englannin kielen tuen lisäämisen sekä sitä tukevien ominaisuuksien lisäämisen moottoriin. Moottoria on käytetty monien japaninkielisten visuaalisten novellien kääntämiseen englanninkielelle. (Wikipedia 2014d.)

PONScripter on oma kehityshaaransa ONScripter-EN-haarasta. Sen tarkoituksena on viedä kehitystä länsimaisempaan suuntaan lisäämällä mm. natiivituki UTF-8 merkistökoodaukselle sekä länsimaisille fonttityypeille. Tämän takia lähdekoodiin on tehty suuria muutoksia eikä PONScripter ole täysin yhteensopiva muiden versioiden kanssa. (Wikipedia 2014d.)

3.3.2 KiriKiri (KiriKiri2/KAG3)

KiriKiri on japanilaisen W. Deen kehittämä moottori visuaalisten novellien kehittämiseen. Sitä käytetään yleensä yhdessä KAG (KiriKiri Adventure Game System) -kehitysympäristön kanssa. Viimeisintä versiota kutsutaan nimellä KiriKiri2/KAG3. Työkalusta yleisesti puhuttaessa käytetään vain termiä KiriKiri. KiriKi-

riä pidetään yleisesti Nscripterin modernina korvaajana Japanissa. KiriKiri on julkaistu GNU-lisenssillä Microsoft Windowsille. Vaikkakin lähdekoodi on avointa, ei porttausta muille käyttöjärjestelmillä ole vielä tehty. (Wikipedia 2014e.)

KiriKiri-moottoriin kehitetty skriptikieli, nimeltään TJS, on olio-ohjelmointimallin mukainen ECMAScript-toteutus (implementation). KAG-kehitysympäristöön on kehitetty XML-tyyppinen merkintäkieli. Molempia voidaan laajentaa laajennusten avulla. (Wikipedia 2014e.)

KiriKirillä tehtyjä julkaisuja voidaan käyttää suoraan länsimaisilla käyttöjärjestelmillä, jos julkaisun merkistökooodaus on UNICODE-tyyppiä. Muuten käyttöjärjestelmän paikkatiedot (locale) täytyy asettaa Japaniin tai käyttää Microsoftin AppLocale-työkalua. (Wikipedia 2014e.)

3.4 Yhteenveto työkaluista

Työkalua valittaessa kannattaa huomioida seuraavat seikat: halutaanko julkaisusta tehdä itsenäinen (standalone) vai web-sovellus, halutaanko työskennellä graafisella kehitysympäristöllä vai itse valituilla työkaluilla pelkkää moottoria käyttäen ja lopuksi, halutaanko kehitystyöhön täysin vapaat kädet alkaen lähdekoodin muokkaamismahdollisuudesta vai tehdäänkö kehitystyö ainoastaan työkalun tarjoamilla työkaluilla.

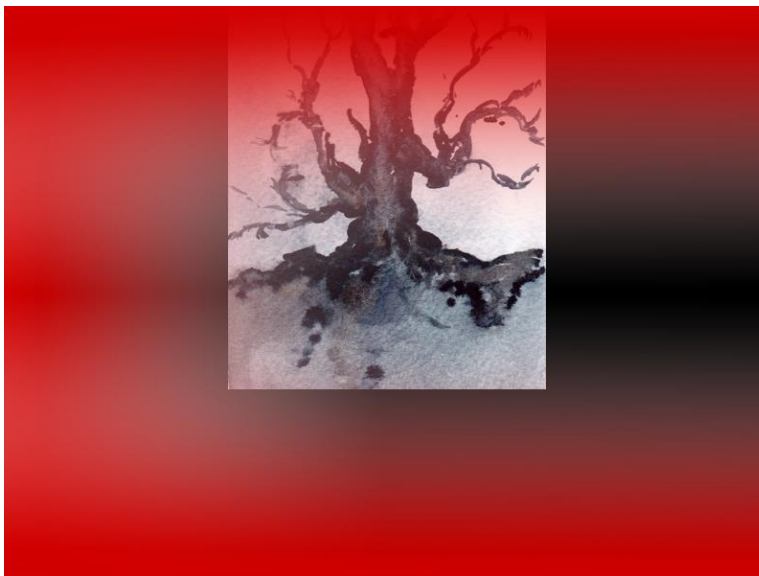
Omat suositukseni edellisessä kappaleessa mainituille kriteereille on Ren'Py itsenäiselle (standalone) julkaisuille. Ren'Py on hyvä valinta siksi, että sillä on pisin kehityskaari kaikista englanninkielisistä työkaluista. Se tarjoaa laajan työkaluvalikoiman ja täysin vapaat kädet lähdekoodin muokkaamiseen ja omien toiminnallisuuksien lisäämiseen Python-skriptikielellä. Lisäksi se on aktiivisessa kehityksessä ja sillä on aktiivinen kehitysyhteisö.

Web-sovelluksen kehitykseen suosittelen joko Belleä tai WebStory Engineä riippuen siitä, halutaanko kehitystyössä käyttää pelkästään graafista ympäristöä

vai tekstieditoria. Molemmat työkalut tukevat samaa skriptikieltä ja ovat avoimen lähdekoodin työkaluja, joten ero tulee omasta mieltymyksestä graafisen ympäristön ja tekstieditorin käytön välillä.

4 Prototyyppi

Jokaisella valitulla kehitysympäristöllä tehtiin visuaalisen novellin prototyyppi saman lähdemateriaalin pohjalta. Materiaali koostui käsikirjoituksesta, taustakuvista sekä taustamusiikista. Lisäksi skriptikieltä tukevilla työkaluilla kehitettiin kuvassa 6 näkyvä, työkalun ulkopuolinen toiminto. Toiminto on tarinan etenemiseen vaikuttava kohtausta, joka laukaistaan pelaajan dialogivalinnan perusteella. Kohtauksessa kuvaruudun kolmelle sivulle ilmestyy punaiset raamit ja pelaajan on vietävä hiiren osoitin raamittoman reunan läheisyyteen aikarajan sisällä. Onnistuessaan uudet raamit ilmaantuvat satunnaisesti valituille reunoille tai pelaaja siirtyy onnistuneeseen kohtaukseen. Epäonnistuessaan pelaaja siirretään epäonnistuneeseen kohtaukseen.



Kuva 6. Työkalun ulkopuolinen toiminto.

Varsinainen tarina kertoo pariskunnasta, joka eksyy ulkomaan matkallaan ja päätyy syrjäiseen kylään. Kylän ihmisten ja läheisellä vuorella asuvan mysteerisen heimon välillä on jännitteitä, joihin pariskunta ajautuu tahtomattaan mukaan. Pelaaja pääsee pelaamaan molemmilla pariskunnan hahmoilla pelin edetessä. Pelaaja joutuu myös valitsemaan puolensa kyläläisten ja heimon välillä ja valinnat vaikuttavat pelin lopputulokseen. Tarina ei välttämättä avaudu kokonaisuudessaan yhdellä läpipeluukerralla. Tarinassa käytetyistä kuvista vastasi Pauli Varonen. Hän maalasi kuvat öljyväreillä, joista otettiin skannaukset. Kuvassa 7 on näkymä vuorelle, jossa heimon päällikkö asustelee. Tarinassa käytetyt äänet on ladattu DL Sounds -sivustolta (DL Sounds 2015).



Kuva 7. Näkymä linnalle (Kuva: Pauli Varonen).

4.1 Ren'py

Ren'Pyllä tehtiin itsenäinen versio (standalone) prototyypistä. Työkalu valittiin sen yleisyyden vuoksi. Se on yleisin länsimaissa käytetty visuaalisten novellien kehitykseen erikoistuneista työkaluista.

Koin mielekkääksi kehittämisen Ren'Pyllä. Olen tottunut tekemään kehitystyön juuri koodaamalla eli tekstieditorilla tai IDE:n koodausikkunassa. Näin ollen koin omavalintaisen tekstieditorin käytön hyväksi tavaksi työskennellä. Ren'Py Launcher tarjoaa myös tekstieditoreita joista voi valita yhden ja asentaa osaksi Ren'Py Launcheria. Launcher suosittelee Editra-editoria ja asensin sen kokeeksi. Editori on selkeä, eikä siinä ole suurta toimintoarsenaalia vaan editori keskittyy perustoimintoihin. Tämä tekee käytöstä selkeän. Myös valmiit värimaailmat ovat mielestäni hyviä.

Asettien lisääminen oli helppoa. Käytännössä kaikki lisätään samalla kaavalla: ensin esitellään muuttuja, jolle annetaan polku tiedostoon. Sen jälkeen asettia käytetään yksittäisellä komennolla. Esimerkki kuva-asetista:

```
image village = "images/scans/village.jpg"
```

```
show village at top with dissolve
```

Kuva siis näytetään show-komennolla. Parametrit at ja with ovat myös komentoja. At määrittää kuvan paikan ruudulla ja with määrittää kuvatehosteen. Kaikilla aseteilla on omat tehosteensa käytössä. Valikoima käsittää yleisesti visuaalisissa novelleissa käytetyt tehosteet.

Ren'Pyn oma skriptikieli on mielestäni selkeää ja sillä saa tehtyä hyvin jäsennellyä skriptiä. Tästä esimerkkinä voisin mainita koodiblokin määrittelyn käyttäen sisennystä ja kaksoispistettä. Blokki alkaa kaksoispisteestä ja kaikki sisennetty skripti kuuluu kyseiseen blokkiin. Tämä pakottaa sisentämään skriptiä ja se tekee skriptistä hyvin jäsennellyä. Näytän alla esimerkin prototyypissä käyttämästäni skriptistä ja avaan sen tekstissä esimerkin jälkeen:

label husbandAwakesInForest:

scene black

show village2 at top with dissolve

pause 1

n "Husband begins to gain his consciousness opening his eyes"

n "He is badly beaten with his clothes ripped"

n "He doesn't see his wife"

n "He tries to find her and yell her name but he can't find her"

n "From the hill he sees that same village down a valley"

n "To other direction land is rising up towards a mountain"

n "There's a old castle high on mountain side"

menu:

"He returns to village hoping his wife is still there alive":

jump husbandReturnVillage

"He continues search his wife in the forest":

jump husbandSearchForest

"He starts climbing up to mountain side towards the castle":

jump husbandSearchMountainSide

Kohtaus esitellään label-komennolla. Syntaksi on "label <nimi>:". Kaksoispisteen käyttö tulee suoraan Pythonin omasta syntaksista. Kaksoispiste toimii samalla tavalla kuin kaarisulut muissa kielissä eli niillä määritellään koodiblokki, tässä tapauksessa label. Myös Pythonissa käytettävä sisennyksen käyttö koodiblokin pituuden määrittelyssä on käytössä. Kaikki sisennyksen sisällä oleva skripti kuuluu sille koodiblokille. Blokki lopetetaan sisennystä muuttamalla. Tä-

mä tuntui minusta aluksi oudolta mutta on kuitenkin yllättävän selkeä. Tyyli myös pakottaa jäsentämään tekstiä mikä tuo selkeyttä.

Scene-komennolla näytetään kohtauksen taustakuva. Show-komento näyttää kuvan taustakuvan päällä. Pause pysäyttää suorituksen numeroarvon verran sekunneissa. N on näyttelijän nimi, jolle vuorosanat kuuluvat. Näyttelijälle voidaan määritellä mm. ruudulla näkyvä nimi, vuorosanojen väri sekä näyttelijän kuva. menu tuo esille valintamenuun. Valinnassa näytettävä teksti on määritelty heittomerkkeihin. Valinnasta siirrytään jump-komennolla siihen kohtaukseen, jonka nimi on annettu.

Ren'Pyn oma skriptikieli riittää mainiosti perinteisen visuaalisen novellin tekemiseen. Jos kuitenkin halutaan lisätä perinteisestä poikkeavaa toiminnallisuutta, on Python käytössä sitä varten. Python-skripti erotetaan komennolla python: joka aloittaa koodiblokin. Alustukset tehdään init-python: -komennolla. Näin Pythonia voi heittää Ren'Pyn skriptin sekaan käyttämällä em. komentoja. Python-skripti voidaan tehdä myös omiin tiedostoihinsa. Tiedostoissa olevia toiminnallisuuksia ei tarvitse tuoda eri komennolla, vaan Ren'Py osaa etsiä ne automaattisesti, kunhan tiedosto on game-kansiossa.

Jos skriptistä löytyy virheitä, Ren'Py näyttää täyden polun (traceback) virheen aiheuttaneeseen riviin. Ren'Py Launcherissa on myös toiminto skriptin oikeellisuuden tarkistamiseen. Mielestäni nämä ovat riittäviä työkaluja virheiden jäljittämiseen. Ren'Pyssä on myös debugger-tila, jossa voi mm. seurata muuttujien arvoja, muuttaa arvoja ajon aikana tai hypätä suoraan johonkin kohtaukseen. Tämä oli riittävän kattavaa ainakin omassa projektissani. Ren'Py Launcher ei kaatunut kertaakaan kehitystyön aikana. Muitakaan teknisiä ongelmia ei ilmennyt.

Hyvä lähde Ren'Pyn ominaisuuksien tutustumiseen on Ren'Pyn dokumentaatio osoitteessa www.renpy.org/doc/html/displaying_images.html.

4.2 WebStory Engine

WebStory Enginellä tehdään web-selaimessa pyörivä versio prototyypistä. Työkalu valittiin, koska kehitystyö tapahtuu tekstieditorilla (ei-graafisesti) ja se on tällä hetkellä ainoita yleisiä työkaluja web-selaimessa pyörivien visuaalisten novellien kehitykseen.

WebStory Enginessä ei ole minkäänlaista editoria. Mukana tulevat kaikki tarvittavat tiedostot, tärkeimpänä moottorin lähdekoodin sisältävä WebStoryEngine.js -tiedosto sekä game.xml, johon kirjoitetaan varsinainen novellin skripti. Tämä karsii heti pois kaikki graafisen editorin käytöstä pitävät käyttäjät. Toisaalta tämä mahdollistaa omien lempityökalujen käytön sekä aikaa säästyy editorin opiskelulta. Kehitystyöhön tarvitaan siis pelkkä tekstieditori. Novellia voi testata paikallisesti web-selaimella offline-tilassa (Google Chrome vaatii tällä hetkellä lisäparametrin XML-tiedostojen käyttöön) tai web-selaimella internetissä lisäämällä novellin tiedostot http-palvelimelle.

Koska kehitys tehdään XML-tiedostoon, käytetään luonnollisesti XML-syntaksia. WebStory Enginessä on oma XML-skeema (schema). Skeema on tosin määritelty JavaScriptillä, joka tekee sen muokkaamisesta hankalaa tottumattomalle. WebStory Engine tukee siis JavaScriptiä. Sillä voidaan tehdä XML-skeeman (schema) ulkopuolisia ominaisuuksia. Varsinainen XML-skripti on hyvin selkeää ja olo on heti kotoisa, jos on tehnyt aiemmin HTML- tai XML-kehitystä. Samaa ei voi sanoa JavaScriptillä tehdystä skriptistä, joka näyttää mielestäni melko sekavalta ja, jonka lukeminen on välillä hankalaa. Tämä tuntuu johtuvan JavaScriptin joustavuudesta, asioita voi toteuttaa hyvin erikoisen näköisellä syntaksilla. Alla esimerkki funktion määrittelystä JavaScriptin Module Patternia käyttäen, esimerkki havainnollistaa kuinka luovasti JavaScriptiä voidaan käyttää:

```
/**
 * Mixin function for adding a drawing function
 * to the canvas.
 */
```

```

out.mixins.display = function ()
{
    var self = this;

    this.canvas.addCallback(
        this.id, function ()
        {
            self.draw();
        },
        this.layer || 0,
        this
    );
};

```

Objektille voidaan määritellä dynaaminen funktio sijoitusoperaattorilla. Myös funktion parametriksi voidaan sijoittaa funktio. Kuten jo mainitsin, tämä tekee koodista hyvin joustavaa mutta samalla vaikeasti luettavaa tottumattomalle. Tällainen joustavuus on varmasti hyödyllistä joissakin tapauksissa mutta voi mielestäni johtaa helposti erikoisiin virityksiin ja huonoon suunnitteluun. Lisätietoa WebStory Enginen skriptikielistä on saatavilla WebStory Enginen kotisivuilla olevasta dokumentaatiosta <http://webstoryengine.org/language>.

Varsinainen kehitystyö on helppoa, jos se onnistuu edellä mainitun XML-skeeman (schema) avulla. Ongelmallisempaa on lisäominaisuuksien teko JavaScriptillä. Ainakin itselläni sen käytöstä tuli ylittämätön ongelma. En saanut prototyypissäni käyttämäni lisätoiminnallisuutta toimimaan. Itseasiassa en saanut minkäänlaista Javascript-toteutusta toimimaan. Tämä johtuu itsestäni, ei WebStory Enginestä. Minulla ei ollut tarpeeksi aikaa opetella JavaScriptiä (etenkään module patterneja), jotta olisin syvällisesti ymmärtänyt, kuinka toteuttaa haluamani asiat. Tämä opetteluun vievä aika kannattaa kuitenkin huomioi-da, jos aikoo valita WebStory Enginen kehitystyökaluksi. Oma osaamiseni JavaScriptistä rajoittuu perusteisiin, toisaalta, jos JavaScriptin tuntemus on hyvä, antaa se hyvät puitteet monipuolisen novellin kehitykseen.

Koska WebStory Engine koostuu vain edellä mainitsemistani tiedostoista, ei siinä ole mitään debugaus-työkalua. Tämä vaikeuttaa ongelmien selvittämistä huomattavasti ja on mielestäni työkalun suurin puute. Internetistä löytyy XML-tiedoston validointityökaluja sekä jokaiselle selaimelle saa JavaScriptin debugaukseen soveltuvan työkalun. Näillä voi kuitenkin testata vain skriptin osan toimivuutta itsenäisenä komponenttina mutta skriptiä ei voi debugata yhtenäisenä kokonaisuutena.

4.3 Visual Novel Toolkit

Visual Novel Toolkitillä tehtiin itsenäinen versio (standalone). Työkalu valittiin tutkimuksen graafiseksi työkaluksi. Valinta tehtiin, koska editorilla novellin toiminnallisuudet (skriptit) voidaan toteuttaa käyttämällä editorin graafisia elementtejä. Lisäksi tärkeä seikka valinnassa oli, että Visual Novel Toolkit on Unitylle tehty lisäosa. Unityn suosion takia sille tehty visuaalisen novellin työkalu herättää varmasti kiinnostusta kehitysyhteisössä. Lisäksi käytettävissä on Unityn tarjoama alustatuki niin kehitys- kuin julkaisu-ympäristöille.

Valinta osoittautui huonoksi. Minulla oli työkalun kanssa niin suuria ongelmia, että jouduin vaihtamaan sen lopulta toiseen graafiseen kehitystyökaluun. Valitsin seuraajaksi Noveltyn, siitä lisää seuraavassa luvussa.

Kerron kuitenkin tässä ongelmistani Visual Novel Toolkitin (Vino) kanssa. Vino lisää oman valikon Unityn GameObject-valikkoon. Valikosta saa lisättyä Vinon elementtejä Sceneen. Elementtiä voi tarkastella Inspector-ikkunassa. Vino on toteutettu suljetulla lähdekoodilla, joten elementtien koodeihin ei pääse käsiksi. Suljettu lähdekoodi sekä skriptikielen puute rajoittavat kehitystyön käyttämään ainoastaan kehittäjien tarjoamia elementtejä sekä toiminnallisuuksia.

Myös dokumentaatio oli vanhentunutta eikä dokumentin tutoriaalia voinut noudattaa sellaisenaan. Kaikkia tutoriaalissa käytettyjä elementtejä ei enää ollut

käytössä nykyisessä versiossa. Internetissä apua on hyvin niukalti saatavilla. Ainoa paikka oikeastaan on Vinon oma viestiketju (thread) Unity-foorumilla.

4.4 Novelty

Noveltyllä tehtiin itsenäinen versio (standalone). Työkalu siis valittiin Visual Novel Toolkitin korvaajaksi. Valittavana oli kaksi graafista työkalua: Novelty ja Belle. Päädyin Noveltyyn, koska sitä on kehitetty pidempään, Bellen ollessa vielä vähän keskeneräinen.

Työkalun käyttö osoittautui todella helpoksi ja sen käytön oppii nopeasti. Ongelmiakin ilmeni kuten itse tekemäni fontin lisääminen. En onnistunut lisäämään sitä useista yrityksistä huolimatta. Apua on saatavilla Noveltyn kotisivujen dokumentaatiosta, kotisivujen forumilta tai Lemmesoft-forumilla olevasta Noveltyn viestiketjusta (thread). Fontin lisäysongelmaan löytyi vastaava kysymys Noveltyn forumilta mutta kukaan ei ollut osannut auttaa asiassa. Huomasin ongelman virheilmoituksen johtuvan käyttämästäni liian pitkästä tiedostopolusta. Tämän korjattuani lisäys ei edelleenkään onnistunut, nyt ei vaan tullut mitään virheilmoitusta. Sama tiedon puute koitui ylitsepääsemättömäksi ongelmaksi Noveltyn skriptikielen ulkopuolisen toiminnan toteutuksessa. Noveltyssä ei ole ollenkaan debuggeria, tarjolla on vain konsoli-ikkuna, jossa voi ajaa toteutettuja funktioita. Myös virheilmoitukset ovat hyvin epätarkkoja, esim. jos skriptissä on virhe niin Novelty ilmoittaa vain, että skriptissä on virhe, ei mitään sen tarkempaa tietoa. Tämä tekee virheiden ja sekä virheellisen toimintalogiikan jäljittämisen vaikeaksi, mikä onkin mielestäni Noveltyn suurin puute. Novelty on myös suljetun lähdekoodin ohjelma, joka aiheutti ongelman yrittäessäni saada AngelScriptin käyttämää ajastin-toiminnallisuutta käyttöön. Toiminnallisuus ei toiminut suoraan ja sen käyttö vaatii tiettyjen Windowsin header-tiedostojen lisäämistä. Niiden lisääminen suoraan skriptitiedostoon ei auttanut, enkä voinut katsoa onko niitä käytössä Noveltyn moottorissa. Noveltyssä on kyllä käytössä sen oma ajastin-toiminto, jota käyttämällä sain ongelman kierrettyä mutta toteutuksesta tuli paljon sotkuisempi kuin AngelScriptin vastaavalla.

Noveltyn oma tekstieditori on hyvin alkeellinen. Se muistuttaa käytännössä Windowsin Notepadia eikä siinä ole esim. värikoodausta käytössä ja esim. rivin vaihto vie aina rivin vaihtoon eikä se osaa sisentää automaattisesti edellisen rivin mukaan. Kaikki asetit on tuotava Noveltyyn sen omalla import-työkalulla. Resurssin tuonti on sinänsä tehty hyvin helpoksi, mutta resurssi ei aina ilmaantunut editoriin vaikka mitään virheilmoitusta ei tulekaan, kuten edellä mainitsemani ongelma fontin kanssa. Novelty tekee projektista vain yhden tiedoston, joten siihen tuotuja resursseja ei näe kuin Noveltyn omasta resurssi-ikkunasta. Yhden tiedoston periaate tekee toisaalta projektista helposti liikuteltavan. Noveltyssä ilmeni myös bugi käytön aikana. Kun kerran käynnistin projektin niin, joidenkin branchien ja menujen goto arvot olivat vaihtaneet paikkaa keskenään. En keksinyt mitään syytä miksi näin oli käynyt.

5 Vertailu

Tässä luvussa vertailen käyttämiäni työkaluja keskenään. Vertailemia asioita ovat tekstin lisääminen, kuvan lisääminen, äänen lisääminen, videon lisääminen, kohtauksen lisääminen, tarinan haaroittaminen (branching) ja debuggaus. Mainitut asiat kattavat kaikki visuaalisissa novelleissa käytetyt elementit kuten tekstin, kuvan, äänen ja videon sekä tarinan rakenteen. Otin mukaan myös debuggauksen, jonka koen itse tärkeäksi novellin kehitystyössä. Nämä kattavat mielestäni tärkeimmät osa-alueet visuaalisen novellin kehitystyössä. Taulukossa 6 olen listannut jokaisen em. asian omalle rivilleen ja lisännyt oman sarakkeen jokaiselle työkalulle, jossa on kerrottu kuinka kyseinen asia toteutetaan kyseisellä työkalulla. Ren’Pyn ja WebStory Enginen kohdalla olen kertonut syntaksin, jolla kyseinen asia tehdään, sekä Noveltyn kohdalla olen kertonut kaikki toiminnot, jolla toiminto tehdään. Jokaisella työkalulla on kotisivuillaan manuaali, jossa toiminnon toteuttaminen kerrotaan tarkemmin, joten en mainitse niitä tässä raportissa.

Taulukon jälkeen olen avannut sellaisia kohtia, joissa on selkeitä eroja ja niissä kohdin laitan työkalut paremmuusjärjestykseen parhaasta huonoimpaan. Perustelen myös miksi olen päätenyt kyseiseen järjestykseen. Paremmuusjärjestys ei ole syntynyt minkään tieteellisen menetelmän kautta vaan pohjautuu puhtaasti minun henkilökohtaiseen käyttökokemukseeni. Lisäksi joillakin työkaluilla ei voi toteuttaa kaikkia taulukon kohtia, jolloin olen maininnut asiasta.

Taulukko 6. Työkalujen vertailua.

| | Ren'Py | WebStory Engine | Novelty |
|---------|--|--|---|
| kohtaus | label nimi: sisältö | <scences> <scene id="nimi"> sisältö </scene> </scenes> | Add page: lisää sivun tarinaan. Add chapter: lisää kappaleen tarinaan. |
| hahmo | define skriptiNimi = Character('ruutuNimi', color="#ffffff") | <character name="skriptiNimi" text- box="tekstilkunanNimi" > <display- Name>ruutuNimi</displ ayName> </character> | "Create a new actor...": tekee hahmon valitusta kuvasta. |
| teksti | label kohtauksenNimi: näyttelijän- Nimi "teksti" | <scenes> <scene id="kohtauksenNimi"> <show as- set="tekstilkunanNimi" /> <line s="hahmonNimi"> teksti </line> | Text Object: kohde, jossa dialogi näytetään ruudulla. Dialogue / Paragraph: lisää dialogin. Dialogille määritellään hahmo, jolle se kuuluu ja varsinainen dialogiteksti. |

| | | | |
|-------|---|--|--|
| | | <pre></scene> <scenes></pre> | |
| kuva | <pre>image skripti- Nimi = "polku tiedostoon" show skripti- Nimi <sijainti ruudul- la><efektit></pre> | <pre><imagepack name="imagePackNimi" > <image name="imageNimi" src="polku tiedostoon" /> </imagepack> <set as- set="imagePackNimi" image="imageNimi" /> <show as- set="imagePackNimi" /></pre> | <p>"Import Assets...": tuo valitun kuvan projektiin.</p> <p>Tuplaklikkaa kuvaa "Asset Libray":ssä: lisää kuvan avoimna olevalle sivulle.</p> |
| ääni | <pre>play music "polku tiedos- toon" <efektit></pre> | <pre><audio name="audioNimi" <efektit> > <track ti- tle="trackNimi"> <source href="polku tiedostoon" type="tiedostotyyppi" /> </track> </audio> <set asset="audioNimi" track="trackNimi" /> <play asset="bgMusic" /></pre> | <p>"Import Assets": tuo valitun äänitiedoston editoriin.</p> <p>"Add to project resources": lisää äänitiedoston tähän projektiin.</p> <p>Audio: lisää äänen avoinne olevalle sivulle. Äänelle määritellään äänilähde, valitsemalla haluttu projektin äänitiedosto.</p> |
| video | <pre>\$ renpy.movie_c</pre> | <pre>Ei videotukea tässä ver- siossa (lisätty seuraa-</pre> | <pre>Play video: lisää vi- deon sivulle. Videol-</pre> |

| | | | |
|------------------------------|--|---|---|
| | <pre>utscene("polku videotiedos- toon") image movie = Movie(koko=(levey- s, korkeus), <lisäpara- metrit>) play movie "in- cubus.mkv"</pre> | <pre>van version toteutetta- viin asioihin) TODO: tee lähde: http://webstoryengine.or g/roadmap</pre> | <pre>le valitaan lähde määrittelemällä tie- dostopolku videoon.</pre> |
| <pre>tarinan haaroitus</pre> | <pre>menu: "virke": jump koh- tauksen nimi "virke": jump koh- tauksen nimi "virke": jump koh- tauksen nimi</pre> | <pre><choice> <option label=" virke " scene="kohtauksen ni- mi"></option> <option label=" virke " scene="kohtauksen ni- mi"></option> <option label=" virke " scene="kohtauksen ni- mi"></option> </choice></pre> | <pre>Show menu: lisää menun sivulle. Me- nulle määritellään vaihtoehtojen mää- rä, jokaiselle vaihto- ehdolle virke sekä vaihtoehdosta siir- ryttävän sivun nimi.</pre> |
| <pre>debuggaus</pre> | <pre>traceback, built-in devel- oper tools</pre> | <pre>selaimen debug- gaustyökalut</pre> | <pre>Debuggaus funktiot: FlushVideoMemory PrintVideoMemory</pre> |

Kuten taulukosta 6 huomataan, Ren'pyn käyttämä tyyli on kaikista suoraviivai-
sin. Jos pitää suoraviivaisesta ja pelkistetystä tyylistä, on Ren'pyn skriptikieli pa-
ras valinta. Toisaalta, jos pitää hyvin jäsenellystä tyylistä, on WebStory Engi-

nen käyttämä XML-syntaksi silloin paras valinta. Jos ei halua itse tehdä skrip-tausta, on Novelty graafinen tapa silloin paras vaihtoehto.

Jos halutaan käyttää videoita, rajautuu WebStory Engine silloin pois videotuen puutteen takia. Kokoruudun kattavan videon lisääminen on Ren'Pyllä ja Noveltyllä yhtä suoraviivaista, vaatien vain yhden komennon. Ren'Py tukee lisäksi elementissä pyörivää videota, jolloin video kattaa vain osan kohtauksen alasta. Sanoisin Ren'Pyn ja Noveltyn olevan yhtä hyviä vaihtoehtoja, jos halutaan tois-taa vain koko ruudun kattavia videoita. Jos kuitenkin halutaan rajata videon tois-toalue pienemmäksi, on Ren'Py silloin ainoa vaihtoehto vertailluista työkaluista. Myös, jos pitää debuggausta tärkeänä, on silloin Novelty huonoin vaihtoehto. Noveltyn debuggaustuki rajoittuu taulukossa 6 mainittuihin kahteen debuggaus-funktioon. Ensimmäinen mainittu "FlushVideoMemory" poistaa kaikki määritte-lemättömät resurssit videomuistista. Jälkimmäinen "PrintVideoMemory" tulostaa tietoa muistin käyttömääristä konsoliin. Noveltyssä ei siis ole tarjolla tyypillisiä debuggaustyökaluja kuten breakpointtien asettamista, koodin suorittamista as-keltamalla tai muuttujien arvojen tarkastelumahdollisuutta suorituksen aikana. Novelty jääkin tässä vertailussa selkeästi joukon hännille. Ren'Pyn omat de-buggastyökalut ovat mielestäni riittävät, etenkin virheen jäljityspolku (trace-back). Polun avulla virheen koko kutsupino näytetään virheilmoituksen yhtey-dessä. Näin virheilmoitus on hyvin selkeä ja virheen jäljittäminen helppoa. WebStory Enginen käyttämistä debuggaustyökaluista huolehtivat suuret, am-mattimaiset yhtiöt selainten takana ja niitä käyttävät mm. webkehitysammattilai-set, joten työkalut ovat korkeatasoisia ja niillä onnistuu mielestäni hyvin. Myös WebStory Enginen debuggaus. Ren'Pyllä ja WebStory Enginellä debuggaus onnistuu hyvin ja sanoisin valinnan niiden välillä olevan enemmänkin makuasia. Muissa vertailun kohdissa selkeitä eroja ei syntynyt ja erot tulevat enemmänkin käyttäjän mieltymysten pohjalta.

6 Pohdinta

Visuaaliseksi novelliksi voidaan siis kutsua peliä, joka koostuu tarinan muodostavasta tekstistä, tarinan kohtausten taustakuvista ja tarinan näyttelijöiden kuvista. Nämä kolme mainittua elementtiä ovat mielestäni kriittisiä visuaalisen novellin määrittelyssä. Yhdenkin puuttuessa peliä ei voida mielestäni kutsua visuaaliseksi novelliksi. Visuaaliset novellit sisältävät yleensä myös taustamusiikkia, ääniefektejä ja näyttelijöiden dialogit puhuttuina ääninä. Myös välianimaatiota voidaan käyttää. Visuaaliset novellit käyttävät myös muiden peligenrejen elementtejä ja muut peligenret voivat käyttää visuaalisen novellin elementtejä, etenkin pelin dialogissa ja tarinan haaroituksessa.

Tärkeimmäksi asiaksi visuaalisen novellin kehittämisessä muodostui hyvän työkalun valinta. Kannattaa ensin miettiä, mille julkaisualustalle visuaalisen novellin haluaa tehdä, tämä rajaa työkaluja paljon. Lisäksi erittäin suureen rooliin omassa kehitystyössäni nousi debuggaustyökalujen käyttö. Niiden pitää olla riittävän kattavat, että kehitystyö pysyy mielekkäänä, eikä ongelmien selvittäminen muodostu liian vaikeaksi.

Tarjolla on eniten työkaluja itsenäiselle (standalone) formaatille. Johtavin työkalu tälle formaatille länsimaissa on Ren'Py. Se muodostuikin omaksi suosikikseni. Ren'Py. Jos halutaan käyttää mieluummin graafista editoria, on Novelty ainoa vaihtoehto tälle formaatille. Kannattaa kuitenkin muistaa, ettei Noveltyllä ole kunnan debuggaustyökaluja käytössä.

Kehitysalustoissa suuntaus tuntuu olevan web-sovellusten suuntaan, kaikki uudet työkalut tukevat juuri tätä alustaa. Näistä työkaluista voisin suositella WebStory Engineä, jos halutaan kehittää tekstieditorilla. Sen tukemalla JavaScriptillä saadaan tehtyjä kehittyneitä ominaisuuksia, lisäksi debuggaukseen on tarjolla web-selainten omat työkalut. Graafisen editorin puolella Belle on mielenkiintoinen vaihtoehto, se tukee myös JavaScriptiä, joten se pystyy samaan kuin

WebStory Enginekin. Belle kehittyi paljon tämän opinnäytetyön kirjoittamisen aikana, joten kehitys tuntuu olevan nopeaa.

Nykyisten työkalujen rinnalle on tulossa uusia mielenkiintoisia työkaluja. Mainitakoon jo julkaistu LDC (Localized Dialogs & Cutscenes), joka on Mel Georgioun kehittämä dialogisysteemi Unitylle (LDC. 2015). Työkalun hinta on kuitenkin melko suuri. Edullisempi vaihtoehto Unitylle on esimerkiksi Fallen Productionsin kehittämä Visual Novel System. (Visual Novel System 2013.) Lisäksi yksi vaihtoehto tulevaisuudessa on Unitylle juuri kehityksen aloittanut Novelation. (Novelation 2015.) Muista kuin Unitylle suunnatuista työkaluista haluan mainita, vielä kehitteillä olevan, nimimerkin 2dt kehittämän Embeddable Visual Novel Engine EVEN (EVEN. 2015). Molemmat työkalut tukevat mobiililaitteita julkaisualustana, joten visuaalisten novellien kehitysmahdollisuudet monipuolistuvat entisestään.

Haluan myös korostaa, että kuten IT-alalla yleensä, on kehitys nopeaa ja jo yhden vuoden aikana ehtii tapahtumaan paljon. Tämänkin opinnäytetyön kirjoittamisen aikana (noin 8 kuukautta) muutoksia tapahtui paljon. Esimerkkeinä voisin mainita Bellen, joka kehittyi niin paljon, etteivät alkuvaiheessa kirjoittamani asiat pitäneet enää paikkaansa. Toisaalta Visual Novel Toolkitin kehitys ehdittiin loppua tänä aikana. Kannattaakin siis käyttää aikaa työkalujen etsimiseen ja niiden testaamiseen, jolloin löydetään uusimmat ja toimivimmat työkalut valintaa tehtäessä.

Lähteet

- Asset Store. 2015a. Visual Novel Toolkit Free. Unity.
<https://www.assetstore.unity3d.com/en/#!/content/9416>. 29.3.2015.
- Asset Store. 2015b. Visual Novel Toolkit. Unity.
<https://www.assetstore.unity3d.com/en/#!/content/9270>. 29.3.2015.
- Belle. 2015a. Download. <http://fr33mind.github.io/Belle/download.html>.
 27.3.2015.
- Belle. 2015b. Support. <http://fr33mind.github.io/Belle/support.html>. 27.3.2015.
- Belle. 2015c. About. <http://fr33mind.github.io/Belle/about.html>. 27.3.2015.
- Benerdal, E. 2014a. About Novelty.
http://www.visualnovelty.com/docs/novelty_intro.html. 29.3.2015.
- Benerdal, E. 2014b. Twitter: @NoveltyDev.
<http://www.visualnovelty.com/forum/viewtopic.php?f=7&t=292>.
 29.3.2015.
- Carlos, P. 2013a. CREDITS.txt. GitHub
<https://github.com/fr33mind/Belle/blob/master/engine/CREDITS.txt>.
 29.3.2015.
- Carlos, P. 2013b. editor/ChangeLog.txt. GitHub
<https://github.com/fr33mind/Belle/blob/master/CHANGELOG.md>.
 29.3.2015.
- DL Sounds 2015. Royalty Free Music. <http://www.dl-sounds.com>. 31.3.2015.
- Eisenberg, J. 2014a. JS-ViNE: A JavaScript Visual Novel Engine.
langintro.com. <http://langintro.com/js-vine/>. 29.3.2015.
- Eisenberg, J. 2014b. jdeisenberg. GitHub. <https://github.com/jdeisenberg>.
 29.3.2015.
- Eisenberg, J. 2014c. js-vine. GitHub. <https://github.com/jdeisenberg/js-vine/commits/master>. 29.3.2015.
- Eisenberg, J. 2014d. js-vine.js. <http://langintro.com/js-vine/js-vine.js>. 29.3.2015.
- Eisenberg, J. 2014e. JavaScript Visual Novel Engine. <http://langintro.com/js-vine/docs/index.html>. 29.3.2015.
- Et Keegera 2014. the-walking-dead-season-two-episode-one-screenshot-04.
<http://etgeekera.com/2014/01/02/the-walking-dead-season-two-episode-one-spoiler-free-review-boulevard-of-broken-dreams/the-walking-dead-season-two-episode-one-screenshot-04>. 31.3.2015.
- EVEN. 2015. <http://lemmasoft.renai.us/forums/viewtopic.php?f=57&t=25377>.
 18.3.2015.
- Giant Bomb 2015. Visual Novel. <http://www.giantbomb.com/visual-novel/3015-2029>. 31.3.2015.
- GitHub. 2015. Belle. <https://github.com/fr33mind/Belle>. 27.3.2015.
- GitHub 2012. WebStory Engine. First Commit.
<https://github.com/jsteinbeck/WebStory-Engine/commit/b6467f7b6218f27e4a894013ee379398d027abfa>.
 27.3.2015.
- GitHub 2014. WebStory Engine. <https://github.com/jsteinbeck/WebStory-Engine>. 27.3.2015.

- iiYO. 2014. WebStory Engine Support. iiYO.
<https://iiyo.org/f/categories/webstory-engine-support>. 27.3.2015.
- LDC. 2015. <http://www.mel-georgiou.co.uk/downloads/ldc-localized-dialogs-cutscene-plugin-for-unity>. 18.3.2015.
- Lemma Soft Forums. 2015a. Belle – Visual novel maker.
<http://lemmasoft.renai.us/forums/viewtopic.php?f=57&t=18741>.
 27.3.2015.
- Lemma Soft Forums. 2015b. WebStory Engine - A Visual Novel Engine for the Web.
<http://lemmasoft.renai.us/forums/viewtopic.php?f=57&t=16722&hilit=webstory+engine>. 29.3.2015.
- Novelation. 2015. <http://www.skistadstudios.com/category/novelation>.
 29.3.2015.
- Novelty. 2014a. About Novelty.
http://www.visualnovelty.com/docs/novelty_intro.html. 29.3.2015.
- Novelty. 2014b. Novelty scripting reference.
http://www.visualnovelty.com/docs/script_index.html. 29.3.2015.
- Novelty. 2014c. Expo.
<http://www.visualnovelty.com/forum/viewforum.php?f=8&sid=10fb7a5a99f7909e135866f6b5d198c9>. 29.3.2015.
- Novelty. 2014d. Using the Novelty editor.
http://www.visualnovelty.com/docs/nov_overview.html. 29.3.2015.
- Novelty. 2014e. Download Novelty.
<http://www.visualnovelty.com/download.html>. 29.3.2015.
- Ren'Py. 2015a. <http://www.renpy.org>. 27.3.2015.
- Ren'Py. 2014b. Why Ren'Py? <http://www.renpy.org/why.html>. 29.3.2015.
- Ren'Py. 2014c. License. <http://www.renpy.org/doc/html/license.html>. 29.3.2015.
- Ren'Py. 2014d. Ren'Py Games List. <http://games.renpy.org>. 29.3.2015.
- Ren'Py. 2014e. The Ren'Py Launcher.
<http://www.renpy.org/doc/html/quickstart.html#the-ren-py-launcher>.
 29.3.2015.
- Ren'Py. 2014f. Language Basics.
http://www.renpy.org/doc/html/language_basics.html. 29.3.2015.
- Ren'Py. 2014g. Screens and Screen Language.
<http://www.renpy.org/doc/html/screens.html>. 29.3.2015.
- Ren'Py. 2014h. What is Ren'Py. <http://www.renpy.org>. 29.3.2015.
- Ren'Py Games List 2015. Planet Stronghold.
<http://games.renpy.org/game/planetstronghold>. 29.3.2015
- Sol-tribe. 2014a. Release notes. http://www.sol-tribe.net/vino/vino_j.html.
 29.3.2015.
- The visual novel database 2015a. FAQ. <https://vndb.org/d6>. 31.3.2015.
- The visual novel database 2015b. Jouka no Monshou. <https://vndb.org/v1390>.
 27.3.2015.
- The visual novel database 2015c. Clannad. <https://vndb.org/v4>. 31.3.2015.
- The visual novel database 2015d. Fate/Stay Night. <https://vndb.org/v11>.
 31.3.2015.
- The visual novel database 2015e. True Remembrance. <https://vndb.org/v103>.
 31.3.2015.

- Wikipedia. 2014a. Visual Novel. Wikipedia.
http://en.wikipedia.org/wiki/Visual_novel. 29.3.2015.
- Wikipedia. 2014b. Ren'py. Wikipedia. <http://en.wikipedia.org/wiki/Ren%27Py>.
29.3.2015.
- Wikipedia. 2014c. GNU Lesser General Public License. Wikipedia.
http://en.wikipedia.org/wiki/GNU_Lesser_General_Public_License.
29.3.2015.
- Wikipedia. 2014d. NScripter. Wikipedia. <http://en.wikipedia.org/wiki/NScripter>.
29.3.2015.
- Wikipedia. 2014e. KiriKiri. Wikipedia. <http://en.wikipedia.org/wiki/KiriKiri>.
29.3.2015.
- WebStory Engine. 2014a. WSE Next Generation.
<http://webstoryengine.org/wse-next>. 27.3.2015.
- WebStory Engine. 2014b. Ideas For Features And Improvements.
<http://webstoryengine.org/ideas>. 27.3.2015.
- WebStory Engine. 2014c. WebStory Engine. <http://webstoryengine.org/start>.
27.3.2015.
- WebStory Engine. 2014d. WebStory Language Reference.
<http://webstoryengine.org/language>. 27.3.2015.
- WebStory Engine. 2014e. game.xml.
<http://webstoryengine.org/webstories/introduction/game.xml>.
27.3.2015.
- WebStory Engine. 2014f. Visual Novels Made With WebStory Engine.
<http://webstoryengine.org/showcase:visual-novels>. 27.3.2015.
- Visual Novel System 2013.
<https://fallenprogramming.wordpress.com/2013/07/16/visual-novel-system-complete/>. 29.3.2015.