



Kehitystyökalut tekstiseikkailu- pelin luomisessa

Lassi Markkinen

OPINNÄYTETYÖ
Toukokuu 2022

Tietojenkäsittely
Ohjelmistotuotanto

TIIVISTELMÄ

Tampereen ammattikorkeakoulu

Tietojenkäsittely

Ohjelmistotuotanto

MARKKINEN, LASSI:

Kehitystyökalut tekstiseikkailupelin luomisessa

Opinnäytetyö 40 sivua

Toukokuu 2022

Opinnäytetyössä vertailtiin saatavilla olevia ilmaisia työkaluja, joita käytetään tekstiseikkailupelien kirjoittamiseen. Työkaluja tutkittiin perusteellisesti ja vertailukohtia olivat esimerkiksi käytettävyys, ohjelmoinnin tuki/tarve sekä erilaiset vientiformaatit.

Tavoitteena oli vertailun kautta löytää väline, joka soveltuu parhaiten opinnäytetyön tekijän tekstipeliprojektin prototypointiin ja mahdolliseen jatkokehitykseen. Kyseessä oli kvalitatiivinen tutkimus, jossa vertailtiin välineitä ja tehtiin havaintoja niiden ominaisuuksista. Tutkimustyön aikana jokaiseen vertailtavista välineistä vietiin raportin laatijan kirjoittama materiaali ja sen pohjalta välineissä kehitettiin pelattavat prototyypit. Kaikki prototyypit sisälsivät samat ominaisuudet: yksinkertaisen tavaränäkymän, vaihtoehtoisia tarinanpolkuja sekä aikaisempiin valintoihin sidottuja kohtia tarinassa.

Tutkimustyön tuloksena jatkokehitykseen valittiin Inklen luoma Ink-skriptikieli ja editori Inky. Inkin kyky integroitua Unity-pelimoottoriin ja tehokas rakenne tarinan jäsentämiseen olivat etuja muihin kehitystyökaluihin verrattuna, vaikka nekin onnistuivat tietyillä osa-alueilla, kuten Twine hyvällä käyttöliittymällään. Loppukokemukset kaikista välineistä olivat pääosin positiivisia, vaikka dokumentaatio-kategoriassa Squiffy epäonnistui pahasti muihin kehitystyökaluihin verrattuna.

Asiasanat: tekstiseikkailupelit, interaktiivinen fiktio, kehitystyökalut

ABSTRACT

Tampere University of Applied Sciences

Business information systems

Software development

MARKKINEN, LASSI:

Development tools for writing a text adventure game

Bachelor's thesis 40 pages

May 2022

Free tools for writing text adventure games were compared in this thesis. The tools were comprehensively examined and points of comparison were usability, support/requirements for coding and available export formats among others.

The goal was to find a tool that would be the most suitable for prototyping and possibly further developing of a text game project. Research was carried out using qualitative methods by comparing the development tools and by making observations about their features. During the research, sample material was imported into each of the development tools and was used to develop playable prototypes. All of the prototypes featured the same functionalities: A simple inventory view, alternate story paths and parts of the story that were tied to previous choices.

Based on the research, the scripting language Ink and editor Inky by Inkle were selected for further development. Ink's powerful framework for structuring the story as well as the ability to integrate with the game engine Unity, were advantages over the other tools, although they also succeeded in some categories, such as Twine with its great user interface. The experiences with all the compared tools were mostly positive in the end, although Squiffy was notably unsuccessful when it came to the documentation category.

Key words: text adventure games, interactive fiction, development tools

SISÄLLYS

1	JOHDANTO	7
2	MÄÄRITELMÄT	9
3	PROSESSI	10
4	KEHITYSTYÖKALUT.....	11
4.1	Twine	11
4.2	InkleWriter, Inky ja Ink.....	11
4.3	Squiffy	11
4.4	Quest	12
5	VERTAILU	13
5.1	Asennus ja alustat.....	13
5.1.0	Twine	13
5.1.1	InkleWriter, Ink ja Inky	13
5.1.2	Squiffy	13
5.1.3	Quest.....	13
5.2	Dokumentaatio ja resurssit.....	14
5.2.0	Twine.....	14
5.2.1	InkleWriter, Ink ja Inky	14
5.2.2	Squiffy	15
5.2.3	Quest.....	15
5.2.4	Yhteenveto	15
5.3	Ominaisuudet.....	16
5.3.0	Twine	16
5.3.1	InkleWriter, Ink ja Inky	17
5.3.2	Squiffy	19
5.3.3	Quest.....	20
5.4	Multimedia.....	21
5.4.0	Twine	21
5.4.1	InkleWriter, Ink ja Inky	21
5.4.2	Squiffy	22
5.4.3	Quest.....	22
5.5	Käyttöliittymä ja käytettävyys	23
5.5.0	Twine.....	23
5.5.1	InkleWriter, Ink ja Inky	26
5.5.2	Squiffy	29
5.5.3	Quest.....	31
5.6	Ohjelmointi	33

5.6.0 Twine.....	33
5.6.1 InkleWriter, Ink ja Inky	34
5.6.2 Squiffy	35
5.6.3 Quest.....	36
5.7 Julkaisukeinot ja vientiformaatit.....	37
5.7.0 Twine.....	37
5.7.1 InkleWriter, Ink ja Inky	37
5.7.2 Squiffy	37
5.7.3 Quest.....	38
6 POHDINTA	39
LÄHTEET.....	40

ERITYISSANASTO

Interaktiivinen fiktio	Tarina, jonka kulkua lukija/pelaaja muuttaa tekemällä valintoja, tyypillisesti esimerkiksi päähahmon näkökulmasta.
IDN	Interactive digital narrative, interaktiivista fiktiota digitaalisessa ympäristössä.
ARDIN	Association for Research in Digital Narratives, järjestö, joka tukee IDN:ään liittyviä tutkimuksia.
ICIDS	International Conference on Digital Storytelling, ARDIN:in tukema vuosittainen konferenssi.

1 JOHDANTO

Tekstiseikkailupelit juontavat juurensa elektronisen peliteollisuuden alkumetreille. Pelit kuten Oregon Trail (1971) ja Colossal Cave Adventure (1976) eivät alun perin sisältäneet 2D- tai 3D-grafiikkaa koska tietokoneteknologia ei vielä mahdollistanut sitä. Vaikka 2D- ja 3D-grafiikkaa hyödyntävät pelit lopulta valtasiivat markkinat, tekstipohjaisille ja samantyyillisille peleille jäi oma yleisönsä ja niiden ilme alkoi nykyaikaistua omalla tavallaan.

Inspiraation lähde monille uusille interaktiivisen fiktion suuntauksille tuli kirjallisuudesta. 1979 alkunsa saanut Choose Your Own Adventure –kirjasarja toteutti fyysisinä kirjoina saman perusidean kuin moni tekstiseikkailupeli digitaalisena nykyään: tarinaa eteenpäin lukiessaan pelaajan oli tehtävä valintoja, jotka määrittivät tarinan suunnan ja johtivat yhteen lukuisista loppuvaihtoehdoista. Sarja on niin vaikutusvaltainen, että Choose Your Own Adventurea on alettu käyttää myös genrenimityksenä ja se esiintyy esimerkiksi Steam –pelialustalla nimikkeenä. Text-based- ja “Interactive Fiction” -tunnisteilla hakemalla Steamista löytyi hakutuloksia 487 kappaletta, ja niistä 267 sisälsi myös edellä mainitun Choose Your Own Adventure-tunnisteen (Valve Corporation 2022). Itsenäisten pelinkehittäjien keskuudessa suosituilla Itch.io-sivustolla Twine- ja Ren’Py -työkaluilla tehtyjä pelejä löytyi yhteensä 9304; Interactive Fiction –tunnisteella hakutuloksia löytyi 15956 (Itch.io 2022). CYOA-tyylistä konseptia on tutkittu jopa opetusympäristöissä interaktiivisten luentojen pitämisen välineenä (Mundy, Consoli 2013). Työkaluja tiettyihin genreihin kuuluvien pelien prototypointiin ja yleistason kirjoittamiseen on olemassa paljon, mutta kumpaankaan kategoriaan kuuluvat välineet eivät varsinaisesti ole suunnattu esimerkiksi CYOA –tyyppisiä pelejä tekeville pelikirjoittajille (Engström, Brusik & Erlandsson 2018, 156).

Interaktiiviseen fiktion liittyvää tutkimustyötä tekevä yhteisö on vuosien varrella kasvanut: 2008 kahdesta Eurooppalaisesta konferenssista muodostettiin ICIDS, jossa IDN:ään kuuluvia aiheita kuten tekstiseikkailupelejä tutkivat ihmiset jakavat vuosittain ajatuksiaan ja havaintojaan. Kun IDN nykyaikaistui ja sai tuulta alleen, sen tuottamiseen suunniteltuja välineitä alkoi ilmestyä enemmän. Laadukkaille erikoistuneille välineille on tarvetta, sillä Excelin mainitaan edelleen olevan yleinen väline pelikirjoittajille (Engström ym. 2018, 156). Opinnäytetyössä vertailtiin

neljää eri välinettä: Twine, Squiffy ja Quest sellaisinaan kun taas Inklen luomat InkleWriter, Inky ja Ink käsiteltiin kokonaisuutena niiden yhteistoiminnan vuoksi.

Välineiden määrä rajoitettiin neljään, jotta opinnäytetyön työmäärä pysyi kohtuullisena ja aikataulussa pystyttiin pitäytymään. Välineet valittiin etsimällä verkosta tunnetuimmat ilmaiset vaihtoehdot. Syvempään vertailuun pääsivät ne välineet jotka pintapuolisen katsauksen jälkeen vaikuttivat sopivilta raportin laatijan peliprojektin prototypointiin ja mahdolliseen jatkokehittämiseen. Opinnäytetyössä pyrittiin kattavasti tuomaan esiin vertailtavien välineiden vahvuuksia ja heikkouksia ja niiden välisiä eroja. Tarkoituksena oli vertailuprosessin avulla tuottaa tietoa, joka hyödyttää mitä tahansa yritystä, oppilaitosta tai yksityishenkilöä, joka haluaa tuottaa interaktiivista fiktiota. Työ toteutettiin kvalitatiivisesti käyttämällä jokaista välinettä ja vertailemalla tuloksena syntyneitä havaintoja.

2 MÄÄRITELMÄT

Tekstipohjaisista peleistä ja tekstiseikkailupeleistä näkee useita eri määritelmiä ja eri kehitystyökalutkin käyttävät samoja termejä hieman eri merkityksissä. Kummatkin termit hyväksytään yleisesti interaktiivisen fiktion alle. Lobke Kolhoff ja Frank Nack (2019, 74) Määrittelevät käyttäjien osallistumista käsittelevässä raportissaan IDN:n (Interactive Digital Narrative) ”Digitaaliseksi interaktiiviseksi viihdemediaksi, jossa käyttäjät voivat tahallisesti vaikuttaa monisyiseen tarinaan interaktiivisen tarinankerronnan järjestelmän mahdollistamana”. Tämän määritelmän alle sopivat sekä pelit, joissa pelaaja tekee valintoja kirjoittamalla, kuin myös pelit, joissa pelaaja valitsee tarjottujen valintavaihtoehtojen välillä esimerkiksi tekstiä tai nappia kursorilla painamalla. Huomionarvoista määritelmässä on kuitenkin sana ”voivat”: se kuvaa nimenomaan valinnan tekemisen mahdollisuutta, ei sen välttämättömyyttä. Felicitas Meifert-Merhard (2013, 2) määrittelee kirjassaan termin future narrative tarinaksi, jossa vähintään yksi kohta voi jatkua usealla eri tavalla. Hän painottaa juuri tätä pelaajan valinnan potentiaalia esimerkillä pelistä, jossa valinta tehdään pelaajan puolesta 30 sekunnissa, jos pelaaja ei itse sitä tee (Meifert-Merhard 2013, 2). Opinnäytetyön puitteissa rakennettiin prototyyppisiä, joissa pelaaja etenee vaihtoehtoja klikkaamalla ja peleihin lisättiin mahdollisuuksien mukaan kuvia ja ääniä. Vertailut välineet kuitenkin soveltuvat monenlaisen fiktion tuottamiseen ja vertailun aikana tehdyt havainnot ovat laajalti hyödynnettävissä myös prototyypeistä eroavien, esimerkiksi kirjoittamalla pelattavien, pelien tekijöille.

3 PROSESSI

Vertailtavat kehitystyökalut otettiin käyttöön ja niihin vietiin raportin laatijan kirjoittaman tekstiseikkailun alkuosat demomateriaalina. Tarinan kulkua tai sen tekstejä ei muutettu välineiden välillä. Tarina rakennettiin jokaisella välineellä pelattavaksi versioksi siten, että tietyt valinnat saivat pelaajan näkemään myöhemmissä tilanteissa eri tekstiä kuin toisin valinnut pelaaja. Tietyt valinnat olivat pelaajalle lukittuja ellei siihen liittyvä ehto täyttynyt. Pelaajalle tehtiin tavarajärjestelmä, johon tavaroita pystyi lisäämään tai poistamaan. Kehitystyökalun mahdollisuuksista riippuen pelaajan senhetkiset tavarat pyrittiin näyttämään omassa näkyvässä, johon pystyi palaamaan napin tai tekstilinkin kautta. Tavarajärjestelmän toteutus vaihteli kehitystyökalujen välillä, mutta sama perustoiminnallisuus saavutettiin kaikissa. Välineiden salliessa niiden käytössä suositettiin työpöytäversiota ja välineitä käytettiin Windows 10 –käyttöjärjestelmää ajavalla pöytäkoneella. Selaimena käytettiin Google Chromea.

4 KEHITYSTYÖKALUT

4.1 Twine

Twine on Chris Klimasin 2009 luoma avoimen lähdekoodin työkalu ja vertailtavista työkaluista tunnetuin. Twinellä tuotettuja pelejä löytyy itch.io –verkkopalustalta yli 6000. Se tunnetaan helppokäyttöisyydestään sekä integraatiostaan verkkotekniikoiden JavaScript, HTML ja CSS kanssa. Sen ytimessä on passaget, joista jokainen Twinessä tehty tarina koostuu. Passage voi olla hetki ajassa tai paikka, jossa pelaaja on. Tyypillisesti passaget yhdistyvät toisiinsa pelaajan tekemien valintojen kautta. Näitä valintoja kutsutaan Twinessä linkeiksi. Työkalu toimii JavaScriptillä kirjoitettujen story-formaattien varassa. Formaatteja on lukuisia eri käyttäjien tekemiä, mutta virallisessa dokumentaatiossa esitellään neljä yleisesti käytössä olevaa perusformaattia Harlowe, SugarCube, Snowman ja Chapbook. Demomateriaalina käytetty tarina tuotiin jokaiseen mainituista formaateista.

4.2 InkleWriter, Inky ja Ink

Inklewriter oli vuonna 2012 luotu maksullinen väline interaktiivisen fiktion tuottamiseen, joka poistui markkinoilta 2018 mutta avattiin uudelleen ilmaisena avoimen lähdekoodin ohjelmana vuonna 2019. Ohjelman luonut yritys Inkle on kehittänyt sen rinnalle markup-kielen Ink ja editorin Inky. Ink on mahdollista integroida Unity-pelimoottoriin, ja Inklewriterilla tehdyt projektit voi muuntaa Ink-tiedostoiksi online-muuntajan avulla. Demomateriaalin kanssa tämä prosessi tehtiin Inklewriterista aloittamalla.

4.3 Squiffy

Squiffy on Alex Warrenin 2014 luoma väline tekstiseikkailupelien tekemiseen. Se sai alkunsa 2014 Pythonilla kirjoitettuna komentoriviä käyttävänä skriptinä, mutta on sen jälkeen muuttanut muotoaan voimakkaasti. Nykyään Squiffy on kirjoitettu JavaScriptillä ja kääntäjä, joka muuntaa .squiffy-muotoiset tiedostot HTML/CSS/JavaScript-paketiksi käyttää Node.js:ää. Väline käyttää Markdownia,

HTML-syntaksia sekä omaa syntaksiaan tarinan tekstin ja osioiden formatoimiseen ja toiminnallisuuden lisäämiseen, mutta Luotujen pelien HTML-, CSS- ja JavaScript-tiedostoja on myös mahdollista muokata suoraan.

4.4 Quest

Quest on ilmainen avoimen lähdekoodin ohjelma tekstipohjaisten ja tekstiseikkailupelien tekemiseen, ja Squiffyn ohella toinen textadventures.co.uk-sivuilta löytyvistä työkaluista. Sitä mainostetaan helppokäyttöisenä, eikä se vaadi ohjelmointiosaamista. Kuten monissa muissakin ohjelmissa, mahdollisuus pelin lähdekoodin editoimiseen kuitenkin on. Questissa pelityypit on eroteltu Text Adventure- ja Gamebook –formaatteihin. Uutta peliä luotaessa on valittava näistä vaihtoehdoista toinen. Text Adventure –formaattissa pelaaja kirjoittaa tekemänsä valinnat itse, kun taas Gamebook –formaattissa mahdolliset valinnat ovat etukäteen kirjoitettuja ja peliä navigoidaan klikkaamalla. Demomateriaali tuotiin Questiin Gamebook –formaattissa.

5 VERTAILU

5.1 Asennus ja alustat

5.1.0 Twine

Twineä on mahdollista käyttää sekä selaimella verkossa että työpöytäsovelluksena. Selainkäytössä ei vaadita kirjautumista ja projektit tallentuvat selaimen välimuistiin. Työpöytäsovelluksesta on saatavilla Windows-, macOS-, ja Linux-versiot.

5.1.1 InkleWriter, Ink ja Inky

Inklewriteria voi käyttää ainoastaan selaimessa, ja kirjoitettu tarina tallentuu selaimen muistiin samoin kuin Twinessä. Inky-editorista on olemassa versiot sekä Windowsille että macOS- ja Linux-käyttöjärjestelmille. Versiot löytyvät GitHub-sivuilta.

5.1.2 Squiffy

Squiffy on saatavilla selainversiona sekä Windows-, macOS-, ja Linux-sovelluksina. Selainversiota käytettäessä pelin tallentamista varten on luotava textadventures.co.uk-tili tai kirjauduttava käyttämällä Microsoft-, Facebook- tai Google-tunnuksia. Luodun pelin voi kuitenkin myös kirjautumatta ladata itselleen ja jatkaa käyttöä paikallisesti työpöytäversion kanssa.

5.1.3 Quest

Questia on mahdollista käyttää selaimessa, mutta toisin kuin Squiffya käytettäessä, kirjautuminen vaaditaan jotta selainversiota pääsee käyttämään. Vaihtoehdot myös työpöytäkäytön suhteen ovat vertailtavista välineistä rajatuimmat, sillä välineestä on saatavilla ainoastaan Windows-versio.

5.2 Dokumentaatio ja resurssit

Tässä osiossa käsitellään vertailtavien työkalujen dokumentaatiota ja muita resursseja, joita käyttämisen opetteluun ja käytössä vastaan tulevien ongelmien ratkaisuun voi hyödyntää. Dokumentaation tarkoitus on toimia käyttöohjeena ohjelmistolle ja selittää käyttäjälle sen tarjoamat ominaisuudet. Ohjelmiston laatu ja ominaisuudet ovat toissijaisia, jos käyttäjä ei ymmärrä miten sitä käytetään (Horch 2003, 219-220). Tästä syystä kehitystyökalujen dokumentaatiot ovat merkittävä vertailukohta. Varsinaisen dokumentaation lisäksi tarkastellaan muuta vastaavaa materiaalia, kuten foorumit joilla käyttäjät voivat jakaa itsetehtyjä ohjeita ja ratkaisuja ongelmiin.

5.2.0 Twine

Työkalusta on saatavilla kattava dokumentaatio sen verkkosivuilla. Dokumentaatio on hyvin kirjoitettu ja sisältää paljon havainnollistavia esimerkkejä, jotka auttavat ominaisuuksien käytössä. Esimerkit ovat formaattikohtaisia, ja tästä oli paljon hyötyä neljän perusformaatin käytössä. Twinen virallinen foorumi suljettiin 2017 ja toiminta on siirtynyt Interactive Fiction Community Forum-sivulle, jossa Twineen liittyvä keskustelu on edelleen hyvin aktiivista.

5.2.1 InkleWriter, Ink ja Inky

Inklewriteria ei ole varsinaisesti dokumentoitu, mutta ohjeet sen toimintoihin löytyvät pelattavan tarinan muotoon kirjoitettuna. Ohjetarinaa läpi "pelaamalla" ja sen osia ohjatusti editoimalla työkalun toiminnoista saa hyvän käsityksen, vaikka etenkin ohjelmointiosaajan näkökulmasta perinteisen dokumentaation uupuminen tuntuu puutteelliselta. Ink puolestaan sisältää kattavan dokumentaation esimerkkeineen GitHubissa. Inkin avulla tehdyn esimerkkipelin voi myös ladata Inkin sivuilta.

5.2.2 Squiffy

Dokumentaatio löytyy työkalun sivuilta, mutta se ei ole kovin kattava. Monet dokumentaation kohdista ja esimerkeistä ovat erittäin lyhyitä, eikä toiminnallisuutta selitetä kovin perinpohjaisesti. Varsinaiset tekniset dokumentoinnit, kuten JavaScriptiä käyttäessä squiffy-objektin sisältämät metodit ja attribuutit listattuna ja selvitettyinä, puuttuvat täysin. Squiffylla on oma fooruminsa, tosin kategorisointia ei ole ja hakutoiminto tekee käytännössä tarkennettuja Google-hakuja.

5.2.3 Quest

Questille on sen verkkosivuilla olemassa sekä laaja dokumentaatio että askele-tain etenevät ohjeet aloittelijoille. Kummatkin sisältävät paljon esimerkkejä ja niiden ohessa on sovelluksesta otettuja kuvankaappauksia, jotka auttavat hahmot-tamaan Questin toimintoja. Questilla on oma fooruminsa samalla sivustolla kuin Squiffy.

5.2.4 Yhteenveto

TAULUKKO 1. Yhteenveto vertailtujen kehitystyökalujen resursseista.

Kehitystyökalu	Aktiivinen fo- rumi	Virallinen dokumen- taatio	Tutoriaali
Twine	Kyllä (jaettu)	Kyllä	Kyllä
InkleWriter, Ink & Inky	Ei	Ei (InkleWriter), Kyllä (Ink/Inky)	Kyllä
Squiffy	Kyllä	Kyllä (vaillinainen)	Ei
Quest	Kyllä	Kyllä	Kyllä

5.3 Ominaisuudet

Tässä kappaleessa kuvataan kehitystyökalujen käyttöä sekä niiden tarjoamia ominaisuuksia tarinan luomiseen, kirjoittamiseen, jäsentämiseen, formatointiin ynnä muihin toimintoihin. Osa kuvatuista ominaisuuksista voi liittyä kehitystyökalun ohjelmointi-integraatioihin tai ohjelmointitukeen, mutta varsinainen ohjelmointiosuus käsitellään erikseen kappaleessa 5.6.

5.3.0 Twine

Twinen erottavat muista vertailluista kehitystyökaluista erityisesti sen story-formaatit, jotka mahdollistavat suurta monipuolisuutta siinä miten esimerkiksi tarinan JavaScriptiä käyttävät osat toimivat. Vaihtamalla formaattia käyttäjä voi muokuttaa kehitysympäristön sellaiseksi, joka soveltuu tekeillä olevaan työhön parhaiten. Neljästä testatusta perusformaattista mieleen jäivät erityisesti Harlowe ja Snowman.

Harlowe on Twinen oletusformaatti uusille käyttäjille, ja pelin logiikkaa rakennetaan sen omalla ohjelmointikieltä muistuttavalla syntaksilla käyttämällä komentoja joita kutsutaan macroiksi. Macrojen syntaksi on yleisiin ohjelmointikieliin kuten JavaScriptiin verrattuna arkikielisempää ja jopa pseudokoodia muistuttavaa: tarkoituksena on todennäköisesti helpottaa niiden opettelemista ja käyttöä henkilöille joilla ei ole aiempaa ohjelmointiosaamista. Vaikka syntaksiin oli alkuun vaikea totutella, Harlowe tarjoaa pitkälti samat perustoiminnot kuin ohjelmointikielinkin ja haluttu logiikka oli helppoa saada peliin kun niitä oppi käyttämään. Tarinan tekstien formatointi kuten otsikot ja lihavointi tapahtuu Harlowen omalla Markdownia muistuttavalla syntaksilla.

Snowman on ikään kuin Harlowen vastakohta: Se on suunnattu JavaScriptiä ja CSS:ää valmiiksi osaaville, eikä sisällä mitään sisäänrakennettuja macroja. Sen sijaan käyttäjä voi lisätä JavaScript-koodia passageihin mihin kohtaan tahansa merkitsemällä kohdan skriptiksi. Snowmanissa tulee valmiiksi mukana JS-kirjastot JQuery, Underscore ja Marked. Käyttäjällä on valtavasti vapautta pelin teknisen toteutuksen suhteen Snowmanilla tehdessä ja se yhdistää Twinen siistin ulkoasun ja tarinarakenteen graafisen esitystavan tähän vapauteen mallikkaasti.

Toisin kuin Harlowessa, tekstiformatointiin käytetään HTML-tageja ja syvempään muokkaamiseen CSS. Tageja voi Snowmanissa sekoittaa leipätekstin sekaan ilman erillistä merkitsemistä.

Tarinan päänäkymässä sitä voi pelata play- tai test-toiminnoilla. Kumpikin toiminto avaa tarinan käyttäjän oletusselaimessa, mutta test-toiminto aktivoi testiominaisuuksia käyttäjän valitsemasta story-formaatista riippuen. Harlowessa tarinan lisäksi selaimeen ilmestyy testipaneeli, josta käyttäjä voi esimerkiksi seurata muuttujien statusta tarinan edetessä. Tarinaa voi myös testata mistä tahansa passagesta aloittaen, joka on hyödyllistä yksittäistä tarinan kohtaa testatessa.

Edit story JavaScript- ja Edit story Stylesheet –toiminnoilla voi lisätä koodia ja muokata tarinan oletustyyliä, mutta pääsääntöisesti koodin lisääminen tapahtuu itse passagejen sisällä jossa koodia helpompi seurata osana tarinan graafista esitystä. Passagejen ei tarvitse sisältää ollenkaan leipätekstiä, vaan ohjelmalogiikalle voi tarvittaessa tehdä kokonaan omia passageja. View proofing copy –toiminto avaa tarinasta html-sivun, jossa on jokaisen passagen teksti ilman tyyllitelyä tai pelilogiikkaa katkoviivoilla eroteltuna. Toiminnosta on hyötyä kun käyttäjä haluaa luetuttaa tarinaansa muilla ilman että lukijan tarvitsee pelata tarinaa tai ottaa pelimekaniikkoihin kantaa.

5.3.1 InkleWriter, Ink ja Inky

InkleWriterin ydintoiminnot vertailluille välineille tyypilliset: pelaaja siirtyy tekstikappaleesta toiseen tekstivaihtoehtojen kautta. Inklewriterissa siirtyminen tarkoittaa seuraavan tekstin lisäämistä viimeisen kappaleen loppuun ja siksi tarinan visuaalinen rakenne on Twineen verrattuna hyvin erilainen. Pelaaja näkee koko siihenastisen tarinan sekä kaikki tekemänsä valinnat kelaamalla ylöspäin ja voi täten halutessaan lukea vaikka pelin ensimmäisten kappaleiden tekstiä ennen sen viimeisiä valintoja. Kirjoittajan näkökulmasta tämä on huomioon ottamisen arvoista, etenkin jos tarinan aiemmista osista saa helposti vihjeitä myöhempien valintojen tekemiseen. Hyödyllinen perustoiminto jonka InkleWriter sisältää on keskeneräisten tarinanpolkujen seuranta: Jos kappale ei sisällä tarinan lopetusta tai valintaa, joka johtaa seuraavaan kappaleeseen, siitä tulee "loose end" joista

InkleWriter pitää lukua ja käyttäjä voi navigoida suoraan merkittyihin kohtiin jatkamaan kirjoittamista. Toiminto on luonnollisesti hyödyllisempi isoissa tarinoissa joissa on monta vaihtoehtoista tarinanpolkua.

Kovin syvällisiä lisätoimintoja InkleWriter ei tarjoa. Yksittäisiä kappaleita, jotka toimivat kuten Twinen passaget, ei voi nimetä mutta tarinan voi jakaa nimettyihin osioihin, jotka sisältävät tietyt kappaleet. Pelilogiikan luomiseen käytetään markkereita ja ehtolauseita. Jokaiseen kappaleeseen voi lisätä markkereita siihen tarkoitetun paneelin kautta. Markkereille annetaan nimi ja ne toimivat kuten boolean-totuusarvot ehtolauseissa. Niitä voi myös käsitellä numeroarvoina käyttämällä markerin nimeä ja merkitsemällä siihen esimerkiksi lisättävän tai vähennettävän jonkin arvon. Tällaisia numeroarvollisia markkereita kutsutaan countereiksi. Countereiden arvoa voi testata ehtolauseissa käyttämällä vertailuoperaattoreita kuten < tai >. Ehtolauseilla on InkleWriterissa kaksi toimintoa: Määrittää näkykö kappaleen teksti tai osa siitä pelaajalle ja mitä valintoja pelaaja voi tehdä.

Inky tarjoaa InkleWriteria kattavamman määrän ominaisuuksia ja muuttaa tarinan ydinrakennetta huomattavasti. Tarinaa ositetaan käyttämällä knot- ja stitch-avainsanoja. Knoteilla on niiden sisältämän tekstin lisäksi nimi jota käytetään kun pelaaja ohjataan siihen valinnan jälkeen. Stitchit toimivat samalla tavalla, mutta ovat suunniteltu toimimaan knottien alla ja pilkkomaan niiden tekstiä ja logiikkaa pienempiin osiin. Kun InkleWriterissa luodun tarinan muuntaa Inkiin, käyttäjän tekemistä osioista tehdään knoteja ja kappaleista tehdään stitchejä. Koska kappaleilla ei ole InkleWriterissa nimeä, prosessi ei ole täydellinen vaan stitchien nimiin käytetään niiden tekstien ensimmäisiä sanoja yhteen liitettynä. Luoduista markkereista tehdään muuttujia, joiden toiminnallisuus pysyy samana.

Knot- ja stitch-palasten kanssa toimiminen on hyvin joustavaa ja tarjoavat hienon enemmän vaihtoehtoja tarinan jaotteluun ja sen osien välillä siirtymiseen. Twimestä löytyy vain yhdellä tasolla toimiva passagen käsite, kun taas Inkyssä knotin ja stitchin luoma hierarkia mahdollistaa esimerkiksi samannimisiä stitchejä eri knoteissa. Yksittäisten knottien tai stitchien sisällä voi myös luoda valintoja jotka lisäävät siihen tekstiä siirtämättä pelaajaa esimerkiksi seuraavaan knottiin. Tämä on hyödyllistä esimerkiksi hahmojen välisiä keskusteluja tehdessä.

5.3.2 Squiffy

Squiffy käyttää tarinoiden jäsentämiseen samaa passage-käsitettä kuin Twine, mutta samasta nimeämisestä huolimatta tarinoiden rakenne vastaa enemmän Inkyn kaksitasoista knot/stitch-järjestelmää: Jokainen passage kuuluu osioon (section), ja ne toimivat pelaajalle valintoina jotka eivät johda uuteen osioon vaan lisäävät tekstiä edelliseen. Passageja voi käyttää esimerkiksi dialogeihin, kun taas osioiden välillä kulkemalla pelaaja edistää tarinaa ja esimerkiksi siirtyy paikasta toiseen.

Pelilogiikan rakentamiseen on olemassa sisäänrakennettu Squiffy Markdown jolla käyttäjä voi luoda muuttujia ja ehtolauseita. Sen komentoja käytetään syntaksilla, jossa komennon nimi annetaan ät-merkin jälkeen tyhjälle riville. Set-komennolla voi tehdä muuttujia ja ne voivat olla merkkijonoja, numeroita tai totuusarvoja. Muuttujien tyyppi määräytyy käyttämän kirjoittaman arvon mukaan: pelkkää muuttujan nimeä set-komennon kanssa käyttämällä Squiffy tekee siitä totuusarvon, kun taas pelkkiä numeroita sisältävä arvo tekee muuttujasta numerotyyppisen ja muissa tapauksissa käytetään oletusarvoista merkkijono-tyyppiä. Muuttujan arvon voi lisätä tekstin sekaan kaarisulkeita käyttämällä. Kaarisulkeet myös erottavat Squiffyssa ehtolauseet leipätekstistä. Samoin kuin aiemmissä välilehdissä, ehtolauseilla voi määrittää sekä mitkä tekstit että mitkä valinnat pelaaja näkee.

Tarinaa voi testata Squiffyssa run- ja build-toiminnoilla. Run-toiminto avaa tarinan editorin output-paneeliin, kun taas build rakentaa projektista vientiversio ja avaa sen selaimessa. Toiminnoista run on nopeampi pelilogiikan testaamiseen, mutta selaimessa tarkasteltava versio vastaa ulkoasultaan paremmin lopulta julkaistavaa tarinaa. Näillä toiminnoilla käyttäjä ei kuitenkaan voi testata tarinaa haluamastaan kohdasta, vaan testiversio alkaa joka kerta alusta. Varsinkin isoa projektia tehdessä tämä on ehdoton heikkous.

5.3.3 Quest

Tarinan perusyksikköinä Questissa ovat sivut, joiden kautta pelaaja kulkee tarinan läpi. Sivuille voi luoda kansioita ja kansiorakenteen avulla jakaa omiin osioihinsa. tarinat sisältävät myös valmiiksi luodun game –sivun, jolla käyttäjä voi esimerkiksi muuttaa tarinan nimeä tai lisätä kuvauksen tai kansikuvan. Pelaajalle on oma sivunsa, jonka tarkoitus on merkitä mistä tarina alkaa. Tämän sivun voi siirtää minkä tahansa muun sivun alle tarinassa jolloin siitä tulee aloituspiste.

Sivuja on useita eri tyyppisiä ja tyyppiä vaihtamalla tarinaan voi lisätä pelilogiikkaa tai efektejä. Oletusarvoisesti sivut ovat text-tyyppisiä, jolloin ne sisältävät vain tarinan leipätekstiä. Muita tyyppisiä ovat multimediallisuuden lisäämiseen tarkoitettuja sivut kuten picture ja pelilogiikan luomiseen tarkoitettuja tyyppiä script ja script + text. Script-toimintojen lisääminen tapahtuu graafisen käyttöliittymän kautta eikä vaadi koodin kirjoittamista. Kuten sivujakin, skriptityyppejä on useita ja jokainen tyyppi suorittaa yhdenlaisen toiminnon. Sisäänrakennettujen toimintojen valikoima on kattava ja sisältää välineitä sivun ulkoasun muokkaamisesta pelaajan antaman syötteiden keräämiseen. Muuttujien hallintaan ja ehtolauseiden sekä looppien kaltaisiin rakenteisiin on kiitettävästi vaihtoehtoja, myös sellaisia joita muissa kehitystyökaluissa ei sisäänrakennettuna ole kuten switch-lauseet ja for each –loopit.

Pelin testaamiseen on muiden kehitystyökalujen tapaan olemassa play-toiminto. Quest sisältää walkthrough-ominaisuuden, jossa käyttäjä voi määrittää sarjan valintoja jotta tiettyjä polkuja ja kohtia tarinassa on helpompaa testata nopeasti. Tätä ominaisuutta ei kuitenkaan tueta gamebook-tarinaformaattissa, joten tarinaa testatessa sen joutuu aloittamaan joka kerta alusta. Vahvuutena play-toiminnossa on sisäänrakennettuja lisätyökaluja, joita testatessa voi käyttää. Debugger- ja log-toiminnoilla pelilogiikan toimintaa voi seurata ja varmistaa esimerkiksi että pelaajan tavaroihin on lisätty oikea esine tietyn valinnan jälkeen. Testattavan pelin HTML-, CSS- ja JavaScript-tiedostoja voi tarkastella Questtiin integroidun Chrome devtools-ikkunan kautta, joka toimii täysin samoin kuin Chromium-pohjaisissa selaimissa.

5.4 Multimedia

Interaktiivinen fiktio oli alkujaan täysin tekstipohjaista. 1980-luvulla julkaistua Mystery House -peliä pidetään ensimmäisenä grafiikkaa sisältävänä pelinä, ja sitä seuraavien vuosikymmenten aikana grafiikka peleissä yleistyi nopealla tahdilla. Multimedian kuten kuvien yhdistäminen interaktiiviseen fiktion aiheuttaa edelleen keskustelua yhteisössä. Joidenkin määritelmien mukaan kuvien puuttuminen on juuri se tekijä, joka erottaa interaktiivisen fiktion muista peligenreistä (Kashtan 2011, 1). Termiä käytetään kuitenkin laajasti myös sellaisista peleistä, joissa on tekstin lisäksi sekä kuvia että ääniä ja animaatioita. Sekä pelkkää tekstiä sisältävissä peleissä että multimediaa käyttävissä katsotaan olevan omat vahvuutensa: Pelkkä teksti haastaa pelaajan käyttämään mielikuvitustaan, kun taas multimediaelementit voivat tehdä pelistä kiinnostavamman näköisen tai muuttaa pelaajan kokemusta (Mehta, Corradini, Ontañón & Henrichsen 2010, 228-231). Tässä kappaleessa tutkitaan vertailtavien kehitystyökalujen tarjoamia välineitä multimedian lisäämiseksi peliin.

5.4.0 Twine

Kuva-, ääni- ja videotiedostoja on mahdollista lisätä projektiin. Kuvatiedostoja tuetaan ainoastaan Base64-koodattuna tai ulkoisella sivustolla ylläpidettynä. Kuvien lisäämiseen ei ole erityisiä toimintoja testatuissa formaateissa vaan se tapahtuu täysin projektin HTML- ja CSS-tiedostoja muokkaamalla. Äänitiedostoja varten SugarCube –formaatti tarjoaa omat macronsa ja Chapbook tukee niitä insert-toiminnossaan. Harlowe ja Snowman –formaateissa ääniä voi lisätä html audio-elementteinä ja käyttää soittamiseen JavaScriptiä. Videotiedostojen lisääminen tapahtuu samaan tapaan HTML:n ja CSS:n kautta.

5.4.1 InkleWriter, Ink ja Inky

Kuvia on helppoa lisätä tarinaan Inklewriterissa add image –toiminnon avulla. Kuvatiedosto on annettava url-osoitteena, eli käyttäjän tulee varmistaa kuvien saatavuus esimerkiksi hostaamalla ne omalla verkkosivustolla. Inklewriter ei myöskään tarjoa kuvien muotoiluun välineitä. Käyttäjä pystyy määrittämään minkä

tekstin jälkeen kuva ilmestyy, mutta tarkempia määrittämiä kuten koon muuttamista ei ole mahdollista tehdä. Ääni- tai videotiedostoille ei ole tukea Inklewrite-rissa, mutta Inkissä käyttäjän on mahdollista lisätä niitä hieman vientiformaatista riippuen. Unityyn integroimalla voi käyttää sen työkaluja, kun taas web-formaattiin viettäessä voidaan hyödyntää JavaScriptiä.

5.4.2 Squiffy

Sisäänrakennettuja toimintoja kuvien tai muun multimedian lisäämiseen ei Squiffyssä ole. Koska Squiffy tukee Markdownia, kuvia voi lisätä peliin sen embedded image –syntaksia käyttämällä. Muuta multimediaa kuten äänitiedostoja on mahdollista lisätä HTML:n ja JavaScriptin tarjoamia ominaisuuksia kuten JavaScriptin HTMLAudioElement-rajapintaa käyttämällä.

5.4.3 Quest

Quest tukee kuvia sekä video- ja äänitiedostoja. Kuvatiedostojen lisäämiseen on olemassa valmiina picture-komento, jolla png- ja jpg-muotoisia kuvia voi lisätä tiedostonimeen viittaamalla. Tällä tavoin lisäämällä kuvien formatointimahdollisuutta ei kuitenkaan ole, vaan syvempää muokkaamista varten kuvat tulee lisätä HTML img-tagia käyttämällä. Img-tageina lisättyjä kuvia on mahdollista tyylitellä CSS-attribuuttien kautta kuten mitä tahansa verkkosivuille lisättyjä kuvia. Myös svg-tagia voi käyttää kuvioiden luomiseen XML-formaattia käyttämällä.

Kuten kuvien kanssa, äänitiedostojen lisäämiseen on olemassa valmiiksi rakennettuja komentoja. Skriptivalikosta käyttäjä voi luoda äänitiedostojen soittamiseen ja pysäyttämiseen komentoja. Toimintoa on helppo käyttää eikä se vaadi ohjelmointia, mutta sisältää rajoituksia. Sound scriptinä lisätyt äänet voivat olla joko mp3- tai wav-formaatissa, eikä niitä ole mahdollista soittaa kuin yhden tiedoston kerrallaan. Käyttäjä voi tässäkin tapauksessa käyttää HTML-tagia äänien soittamiseen, ja se mahdollistaa muut tiedostoformaatit sekä syvemmän muokkaamisen kuten videon hallintapalkin näyttämisen pelaajalle.

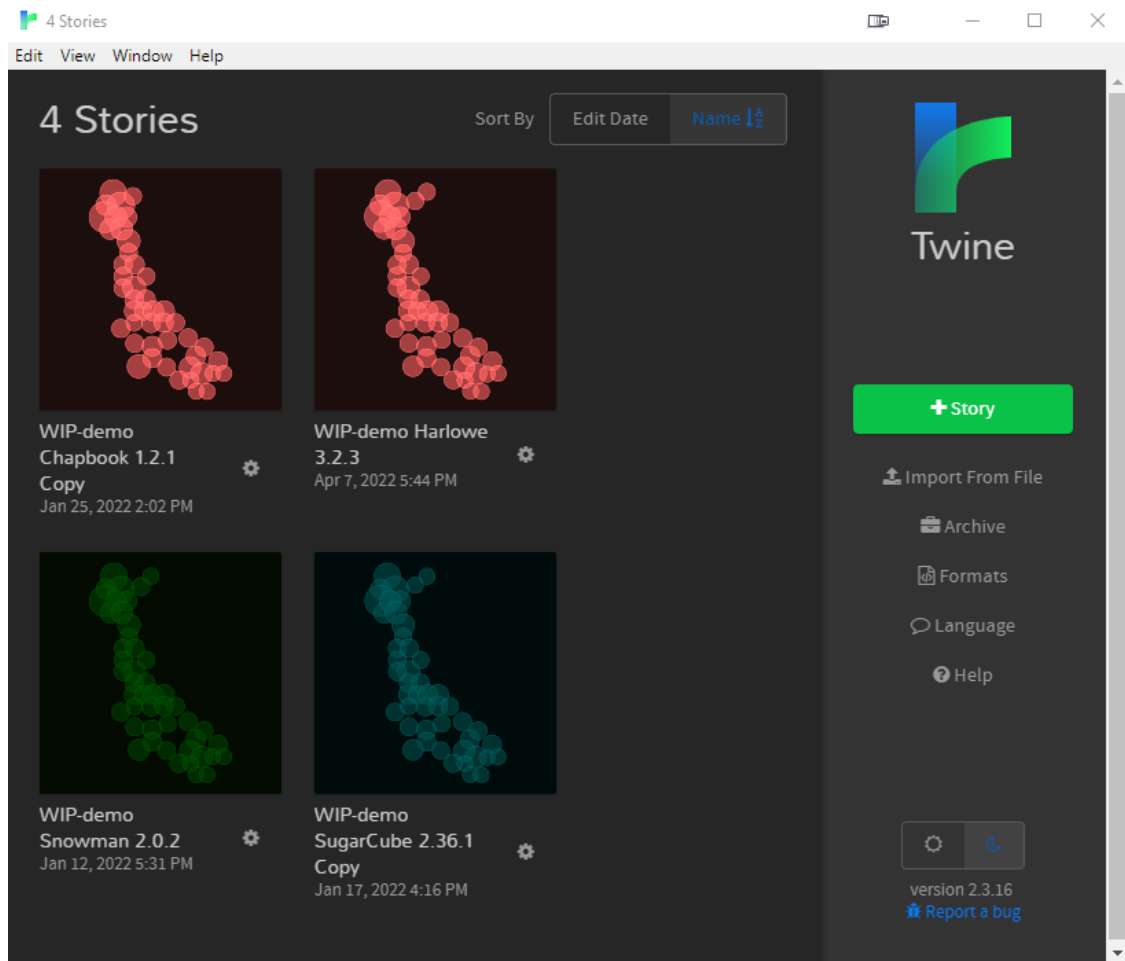
Videoiden lisäämisessä Quest tukee Youtube-linkkien kautta upotettuja tiedostoja. Play youtube video –skripti lisää projektiin HTML iframe-elementin johon linkkinä annettu video on upotettu. Tätä toimintoa käyttämällä lopputulosta voi muokata JavaScriptillä, mutta myös HTML video-tagia voi käyttää suoraan jolloin myös paikalliset videotiedostot toimivat.

5.5 Käyttöliittymä ja käytettävyys

Käyttöliittymä on tärkeä osa ohjelmistoa. Se kattaa kaikki ohjelman napit, tekstit, valikot, värit ja muut elementit. Käyttöliittymä määrittää, kuinka helppoa ja miellyttävää ohjelmistoa on käyttää. Vertailtavien kehitystyökalujen kaltaisten, sisällön tuottamiseen suunniteltujen välineiden kanssa hyvä käyttökokemus parantaa tuottavuutta: Kun käyttäjän ei tarvitse kuluttaa aikaansa esimerkiksi tärkeiden toimintojen etsimiseen vaikeaselkoisesta valikosta, hän pystyy keskittymään luovaan kirjoitustyöhön. Esimerkkejä hyvästä käytettävyydestä ovat nappien ryhmitely niiden käyttötarkoituksen mukaan ja sellaisten ikonien välttäminen, josta ei käy heti ilmi niiden merkitys (Grant 2018, 26, 228). Hyvän käytettävyyden tunnusmerkkejä ovat täten asiat kuten helppokäyttöisyys, selitysten ja ohjeiden ymmärrettävyys ja ulkoasun selkeys.

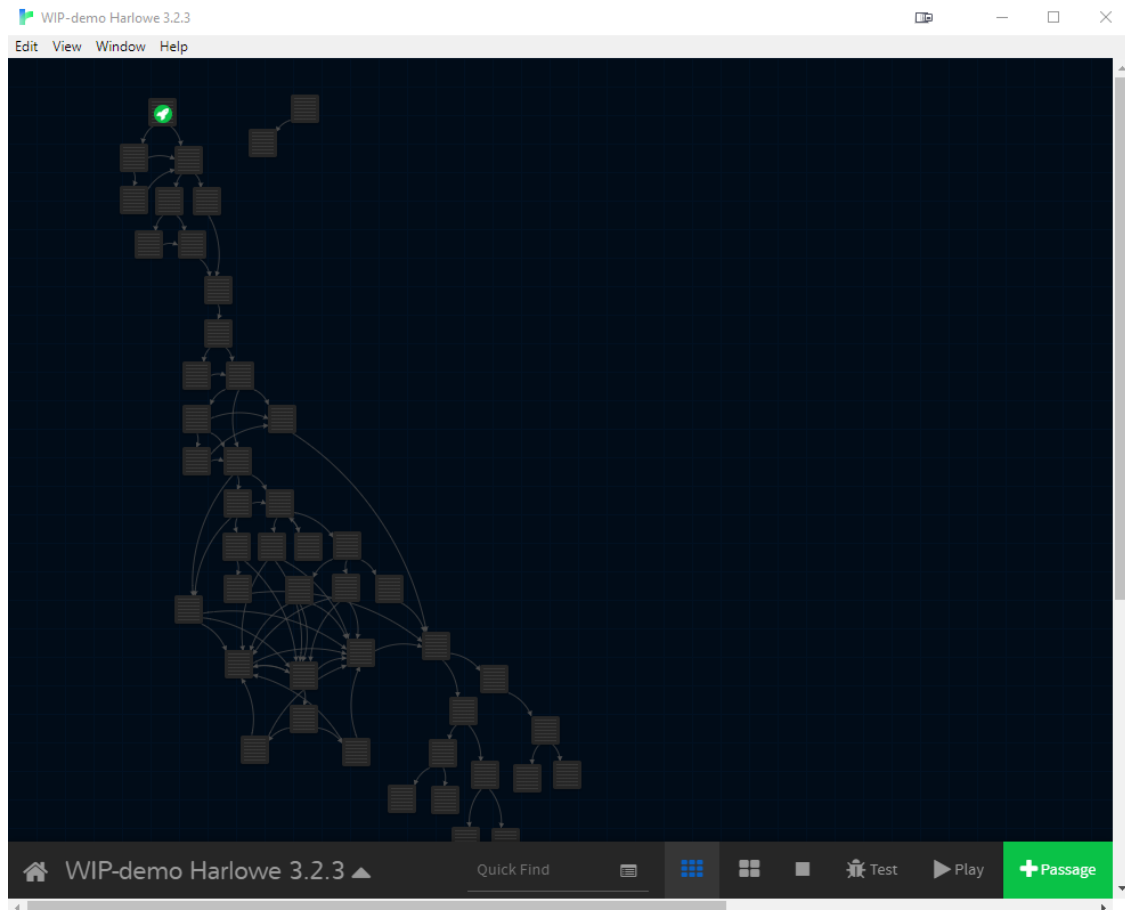
5.5.0 Twine

Käyttöliittymän perusnäkyminen on pelkistetty ja selkeä (Kuva 1). Valtaosa tilasta on varattu story-listaukselle, jossa on ruudukkomalliin asetettu kaikki käyttäjän tekemät storyt. Oikean puolen sivupalkista löytyvät toiminnot ovat helppoja ymmärtää ja niistä kaikista löytyy tyhjentävä kuvaus tooltip-tekstinä, joka aukeaa kun käyttäjä pitää kursoria toiminnon päällä. Päänäkymän tärkein toiminto eli uusien storyjen luonti on asianmukaisesti värikoodattu, järjestetty ensimmäiseksi ja samalla tehty sivupalkin toiminnoista näkyvimmäksi.



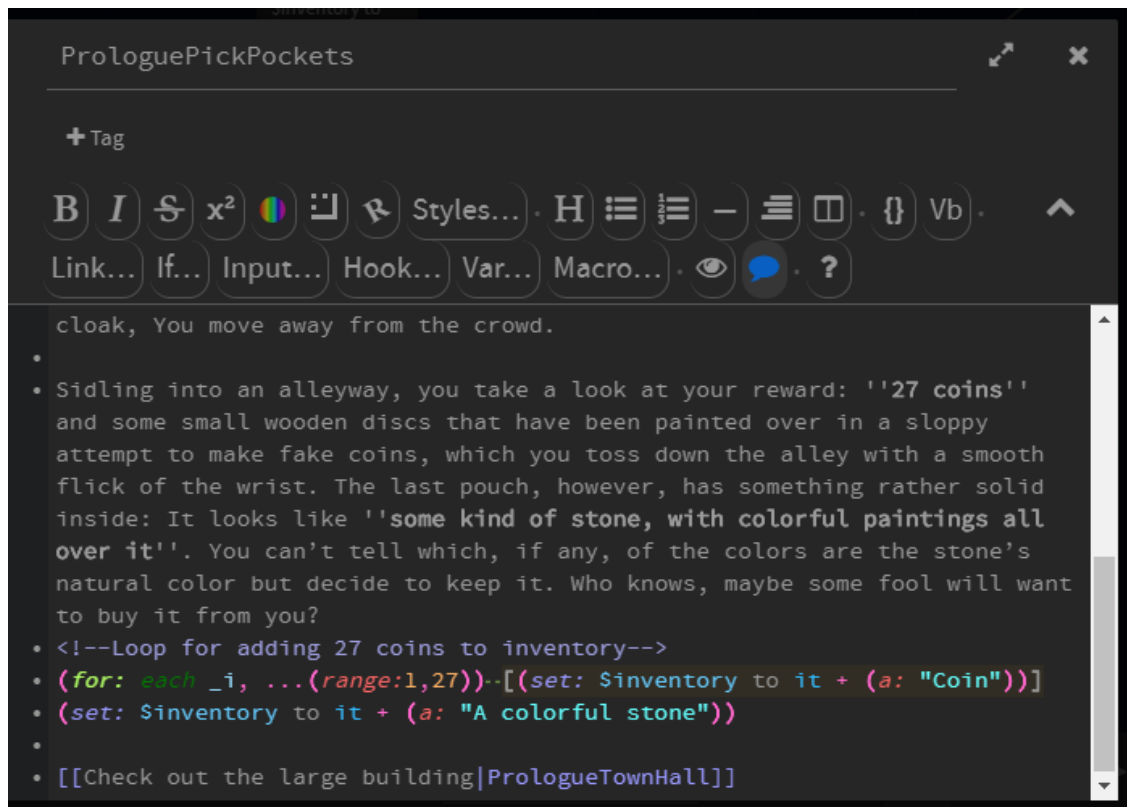
KUVA 1. Twinen päänäkymä.

Story-näkymän pääelementti on ruudukko, jossa kaikki käyttäjän kirjoittamat passaget näkyvät (Kuva 2). Jokainen passage on kuvattuna neliönä, joka sisältää oletustarkkuudella sen nimen ja osan sen mahdollisesta leipätekstistä. Passagejen väliset linkit ovat kuvattuna nuolilla. Ruudukossa on kolme tarkkuustasoa, joista uloimmalla käyttäjä voi tarkastella storyn kokonaiskuvaa ja karkeaa rakennetta, kun taas sisemmät tarkkuustasot soveltuvat storyn tiettyjen kohtien työstämiseen. Jo demotarinan kokoisessa tarinassa Twinen käyttämä rakenne on erittäin hyödyllinen: Kokonaiskuva ja pelaajalle mahdolliset polut on helppo pitää hallussa toimintakaaviota muistuttavan näkymän ansiosta.



KUVA 2. Twinen story-näkymä (Harlowe).

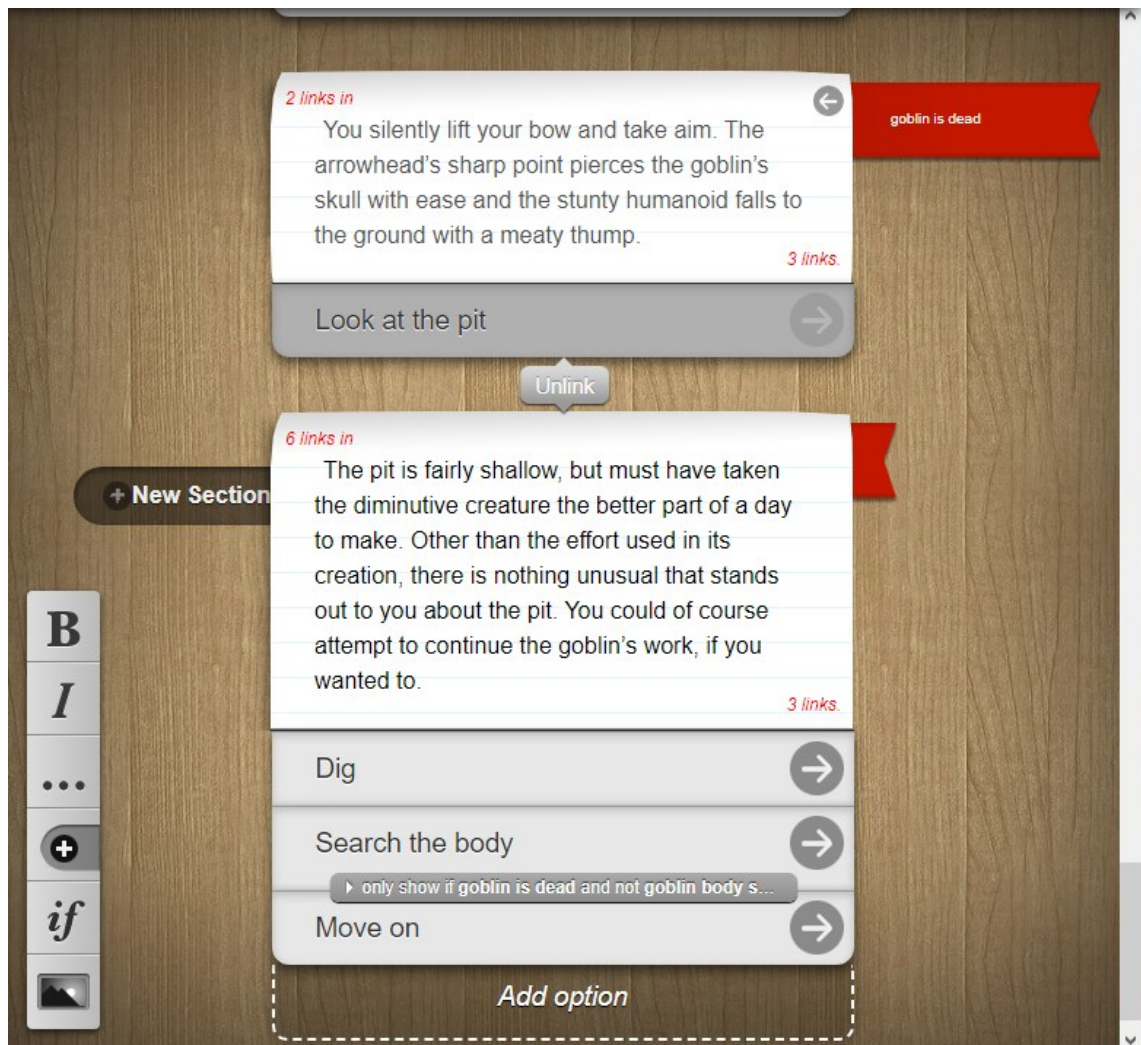
Passage-näkymä avautuu story-näkymän päälle omana ikkunanaan. Se sisältää kaikissa formaateissa ylärivin passagen nimeä varten, napin tunnisteiden lisäämiseen sekä ison tekstikentän leipätekstille ja makroille (Kuva 3). Harlowe-formaatissa tekstikentän ylle on myös koottu tekstin formatointiin ja makrojen luomiseen käytettäviä toimintoja. Yksittäistä passagea tarkastellessa syntaksin korostus on sisäänrakennettuna ainoastaan Harlowe-formaatissa eikä muissa testatuissa formaateissa ominaisuutta ole tuettu. Etenkin pidempien skriptien kirjoittamista passagejen sisään hankaloittaa syntaksin korostuksen puute.



KUVA 3. Passage-näkymä Harlowe-formaatissa.

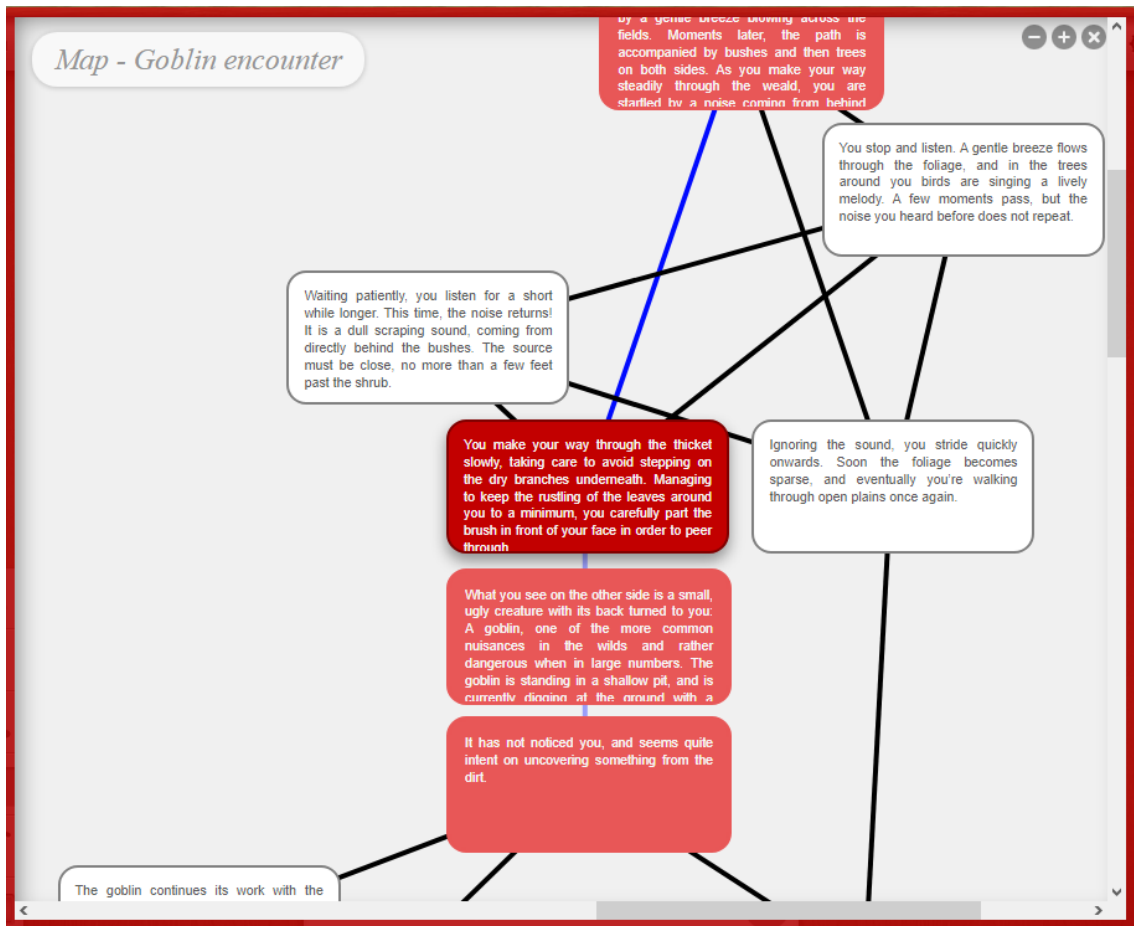
5.5.1 InkleWriter, Ink ja Inky

Inklewriterin käyttöliittymä on pelimäinen, eikä editointinäkymä ulkonäöltään poikkea kovin suuresti linkin kautta pelattavasta versiosta (Kuva 4). Keskellä käyttäjä näkee sen hetkisen tarinankohdan paperiarkkia muistuttavalla taustalla. Koko tarinan teksti ei ole näkyvässä kerralla, vaan tarinan kirjoittaja siirtyy eri valintojen kautta tekstien välillä samoin kuin pelaajakin. Mahdolliset valinnat ovat listattuna alapuolella ja niitä voi sieltä myös lisätä. Näkymän vasemmasta alareunasta löytyy sivupalkki, jonka kautta käyttäjä voi tehdä muokkauksia kuten liihooda tekstiä tai lisätä kuvia. Nappien kuvista on helppoa hahmottaa niiden toiminnot ja ne sisältävät tooltip-selitykset samoin kuin Twine.

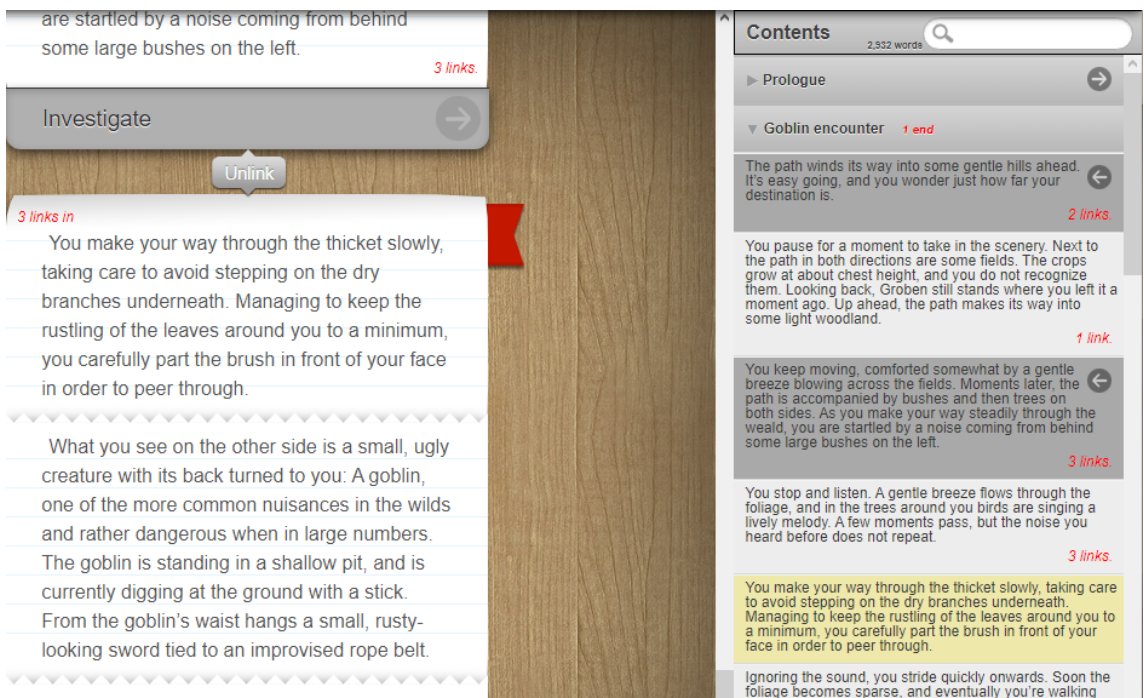


KUVA 4. InkleWriterin perusnäkökulma.

Yläpalkin oikeasta laidasta löytyvät map- ja contents –toiminnot. Map avaa kaaviomaisen näkymän jossa tarinan kohdat ovat havainnollistettu neliöillä ja niiden yhteydet kuvattu viivoilla. Näkökulma muistuttaa Twinen päänäkymää tarinaa editoitaessa, mutta on Twineen verrattuna selvästi hankalampi seurata eikä sovellu juuri muuhun kuin yksittäisten tarinapolkujen seuraamiseen graafisesti esitettynä (Kuva 5). Contents –paneelista käyttäjä näkee kaikki tarinan tekstit osioittain jaettuna. Näkökulma on hyödyllinen tarinan kohtien läpi navigoimiseen, mutta vaikeaselkoinen ja kankea käyttää (Kuva 6). Paneelia on mahdollista rajata hakutoiminnon kautta, mutta sen käyttäminen ei ole kovin luontevaa.

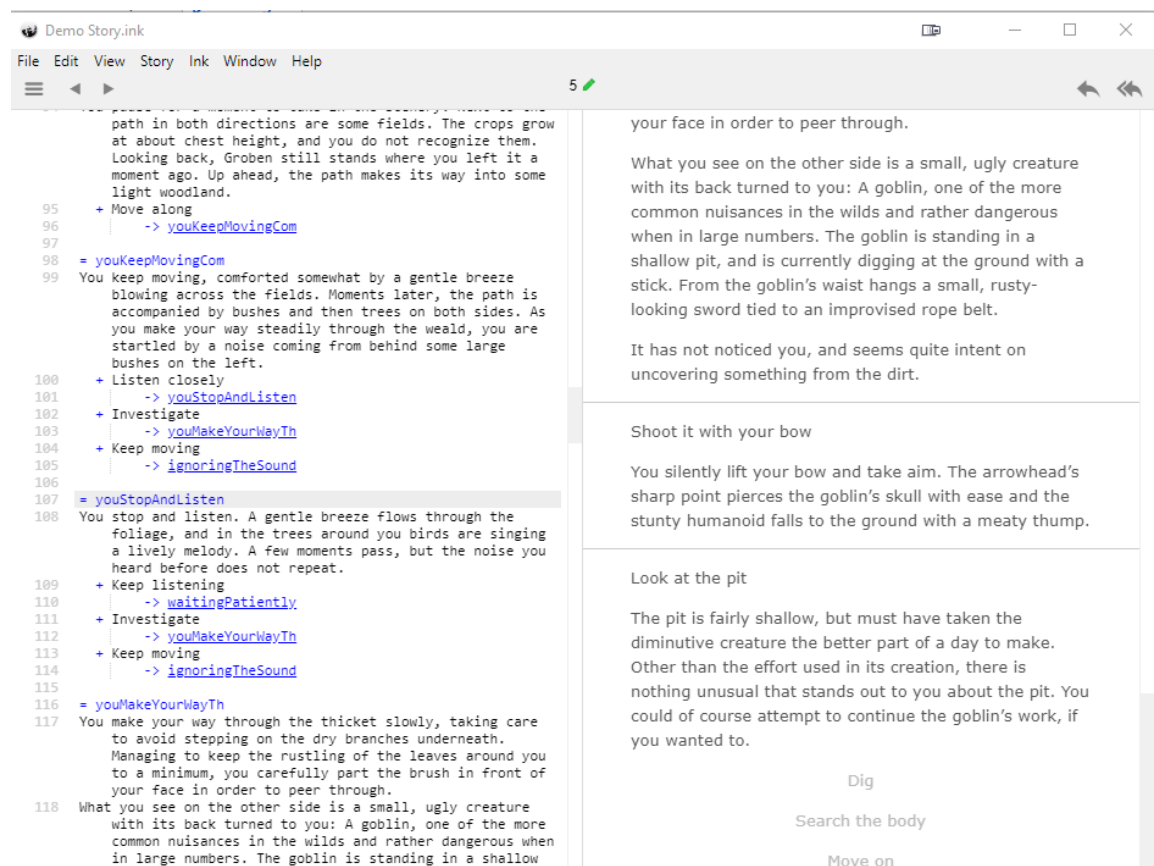


KUVA 5. InkleWriterin map-näkymä.



KUVA 6. InkleWriterin contents-valikko.

Inkyssä editorin käyttöliittymästä valtaosa on varattu tarinan tekstille ja tila on jaettu kahteen osaan: Vasemmalla puolella on kaikki tarinan teksti Ink-formaatissa ja oikealla tarinan pelattava versio (Kuva 7). Puoliskojen kokoa voi halutessaan muuttaa. Lopun tilan vie yläpalkki, joka sisältää tekstieditoreille tyypillisen jonon nappeja kuten file, edit ja help. Inkyllä uniikit story- ja ink-napit sisältävät aputoimintoja kuten tarinan sanamäärän kaltaista статистиikkaa ja valmiiksi tehtyjä ink-koodin palasia joita voi lisätä tarinaan. Vaihtoehdot on järkevästi nimetty ja valikkorakenne on toimiva. Ink-tekstinäkymässä varsinaiset Ink-koodina kirjoitetut tekstit kuten muuttujien alustukset tai knottien/stitchien nimet on värikoodattu ja tämä helpottaa luettavuutta huomattavasti.



KUVA 7. Inky:n päänäkymä.

5.5.2 Squiffy

Päänäkymä on rakenteeltaan hyvin samanlainen kuin Inkyssä: Tila on jaettu kahteen osaan jossa Inky:n tapaan vasenta puolta käytetään tarinan tekstin ja koodin muokkaamiseen (Kuva 8). Oikean puoliskon sisältöä voi muuttaa yläpalkista, joka sisältää vaihtoehdot Help, Tools, ja Output. Help ja tools näyttävät yksinkertaisia

ohjeita tarinan kirjoittamiseen tai nappeja joilla voi luoda osioita tarinaan ja piilottaa osioiden tekstit. Output-sivulle tulee näkyviin pelin testattava versio run-toimintoa käytettäessä.

Tavanomaisen tekstieditori-yläpalkin lisäksi käyttöliittymässä on ylhäällä rivi nappeja, kolme editorin vasemmalla laidalla ja kaksi oikealla. Vasemman puolen napit open, save ja run ovat jo tekstiensä puolesta selkeät ja niiden tarkoitukset ymmärtää heti. Napit erottuvat hyvin vaaleasta taustasta sinisellä värityksellään, ja testaamiseen käytettävä run-nappi on väritetty vihreäksi korostamaan toimintoa. Kaikissa napeissa on tekstin lisäksi myös pieni ikoni, joka auttaa hahmottamaan napin toiminnon nopealla silmäyksellä. Oikean laidan options- ja build-napit noudattavat muuten samaa kaavaa, mutta options-napissa ei ole tekstiä eikä se erotu väriltaan taustasta. Sen tarkoituksen kuitenkin tunnistaa klassisesta hammaspyörä-ikonista.

Squiffyssä on valmiina syntaksin korostus, joka hyödyttää koodin kirjoittamista ja squiffyn sisäänrakennettujen toimintojen käyttämistä valtavasti. Ominaisuus on erityisen hyödyllinen koska Squiffy-tarinan teksti on oletusarvoisesti kaikki yhdellä yhtäjaksoisella sivulla. Luettavuutta parantamaan osioiden tekstit voi kaataa rivinumeron vierestä löytyvästä nuolesta, jolloin vain osion nimi jää näkyviin. Tämä toiminto ei kuitenkaan toimi odotetusti passagejen kanssa vaan teksti näkyy aina.

The screenshot shows the Squiffy editor window titled "demo_story.sq - Squiffy". The interface includes a menu bar (File, Edit, View, Build, Help), buttons for Open, Save, Run, Restart, and Build, and tabs for Help, Tools, and Output. The main editor area contains code for a scene, with line numbers 55-73. The code includes a function call [Pick pockets](prologuePickPockets), a function definition [prologuePickPockets]:, and several lines of JavaScript code that update the 'coins' and 'inventory' variables. The output window on the right displays the rendered text, which includes a paragraph about leaning on a house, a paragraph about crossing the road and slipping into a crowd, and a paragraph about sidling into an alleyway. The inventory is shown at the bottom as "inventory = Bow & arrows, a colorful stone".

```
55 almost walk right into you without ever lifting their gaze
56 from the ground. Strange, you think, but easy pickings if you
57 feel like going through a couple of pockets for some
58 valuables.
59
60 [Pick pockets](prologuePickPockets)
61
62 [prologuePickPockets]:
63     squiffy.set('coins', squiffy.get('coins') + 27);
64     //coins += 27;
65
66     let inv = squiffy.get('inventory');
67     inv.push('a colorful stone');
68
69     squiffy.set('inventory', inv);
70     // inventory.push('A colorful stone');
71
72 You lean on the side of a house, watching the townsfolk slowly
73 make their way down the main road. Eventually, a larger group
of people starts to form around some wooden stalls covered
with cloth, on the opposite side of the road. It looks like
people are waiting to buy what you think are fruits in woven
baskets, and you spy leather pouches in the hands of those
closest to the stalls. With a grin on your face, you make
your move.
70
71 You swiftly cross the road and slip into the crowd by the
stalls. "15 pieces for just one Buni fruit, can you believe
him?" a voice close to you says with a surprisingly
disinterested voice. An equally bored-sounding, yet a touch
more frustrated reply comes: "Yeah well, what're you gonna do
about it? Or me, for that matter. It's not like Payton's
prices are any better." With precision and quickness, your
experienced hands dance through the bags and pockets of a few
people close to you. After stashing a couple of small pouches
and some loose coins under your cloak, You move away from the
crowd.
72
73 Sidling into an alleyway, you take a look at your reward:
<strong>27 coins</strong> and some small wooden discs that
inventory = Bow & arrows
coins = 27
inventory = Bow & arrows,a colorful stone
```

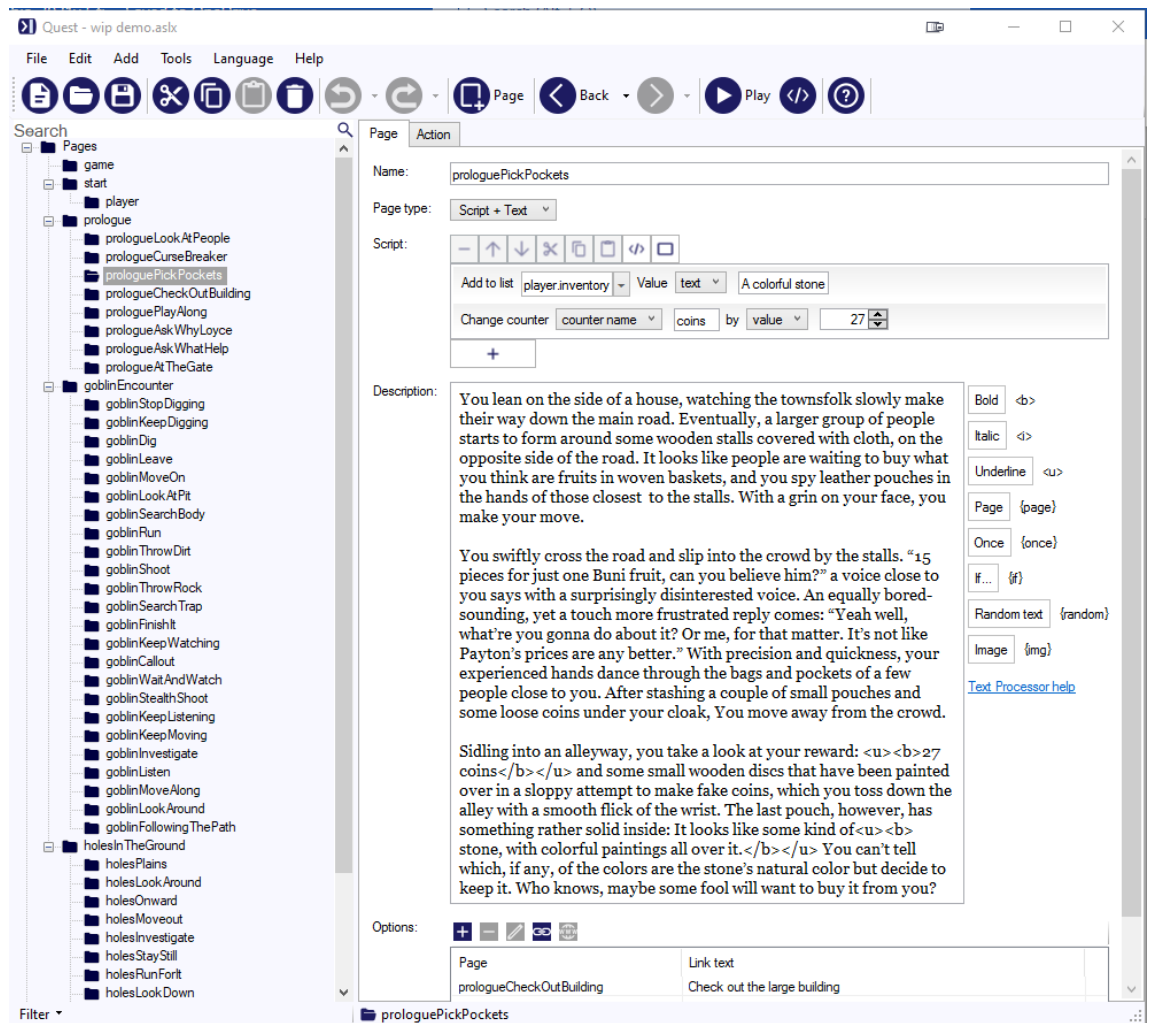
KUVA 8. Squiffyn päänäkymä.

5.5.3 Quest

Quest poikkeaa muista työkaluista merkittävästi käyttöliittymässään. Se on ulkoasultaan muita työkaluja karkeampi ja monella tapaa vanhemman, perinteisen työpöytäsovelluksen näköinen (Kuva 9). Iso osa elementeistä kuten valikoista ja napeista ovat pelkistettyjä eikä värikoodausta juuri ole.

Päänäkymän ylärivillä on nappeja erilaisille sivujen käsittelyyn liittyville toiminnolle kuten kopioinnille, leikkaamiselle ja uusien sivujen luonnille. Napit ovat kooltaan sopivan kokoisia ja erottuvat tummansinisellä taustavärillään todella hyvin editorin taustasta. Jokainen nappi sisältää valkoisen ikonin, joka kuvaa sen toimintoa. Ikonit ovat osuvia ja nappien toiminnot ymmärtää suurimman osan kohdalla välittömästi myös henkilö ilman koodaustausta. Napeista löytyy yhden sanan tooltip-teksti, ja toisiinsa liittyvät toiminnot ylärivillä ovat sijoitettu vierekkäin ja erotettu seuraavista napeista ohuella pystyviivalla.

Näkymän vasen laita sisältää kaikki tarinan sivut kansiorakenteen muodossa. Sivuja voi laittaa toistensa sisään ja siten ryhmitellä tarinan eri vaiheita jotta niitä on helpompi seurata kansiorakenteessa. Kansiorakennetta on sujuvaa selata, ja isommissa projekteissa hakutoiminnolla voi hakea sivujen nimillä tai niiden osilla pienempiä listauksia.



KUVA 9. Questin päänäkymä.

Kuten muissa työkaluissa, tarinan leipäteksti ja sen sekaan kirjoitetut skriptit vievät pääosan tilasta. Tekstin formatoimiseen käytettäville tageille tai sekaan kirjoitetuille skripteille ei ole syntaksin korostusominaisuutta. Formatointi-tagit ja napit joilla niitä luodaan löytyvät päätekstilaatikon oikealta puolelta kun taas sivujen välisten linkkien hallinnointityökalut sen alapuolelta. Nämä ominaisuudet ovat päätekstilaatikon läheisyydessä hyvin sijoitettuja, koska niitä käyttäjä tarvitsee

kirjoittaessaan eniten. Linkkien hallinnointityökalujen ikonit ovat ymmärrettävyydeltään paikoin todella huonot. Ikonien perusteella ei jokaisen napin kohdalla saa selvää käsitystä sen toiminnosta, ja tällöin joutuu tukeutumaan tooltip-vihjeisiin.

5.6 Ohjelmointi

Jokaista vertailtavaa välinettä voi käyttää ilman ohjelmointiosaamista, mutta niistä kaikista löytyy ohjelmointituki jollakin tasolla. Osassa välineistä ohjelmoinnin opettelu on jopa suositeltavaa, etenkin jos käyttäjän tavoitteena on tehdä kokonainen peli välineen avulla ja julkaista se. Yleisin kehitystyökalujen tukema ohjelmointikieli on verkkokehityksessä laajasti käytetty JavaScript, ja tässä kappaleessa tarkastellaan sekä sitä että mahdollisia muita kieliä pelien ohjelmoimiseen joita kehitystyökalut tukevat.

5.6.0 Twine

Story-formaatista riippuen Twine tukee laajasti sekä sisäänrakennettuja makroja että JavaScript-koodin kirjoittamista tarinan sisälle. Yksinomaan makroilla toimiva Harlowe on muita rajatumpi pelilogiikan rakentamisen suhteen, mutta sen makrot ja koukut (hook) sisältävät suuren määrän erilaisia toimintoja joilla on mahdollista toteuttaa myös hieman edistyneempää logiikkaa.

Makroja kirjoitetaan suoraan tarinan passageihin ja Twine tunnistaa makrojen syntaksin leipätekstin seasta. Makroja kutsutaan käyttämällä sen nimeä sulkeiden sisällä. Nimen ja kaksoispisteen jälkeen seuraa makrosta riippuen joko suoritettava toiminto tai esimerkiksi ehtolause. Makroja voi tällä tavoin myös kirjoittaa toistensa sisään, jolloin esimerkiksi for-makron loopissa jokaiselle alkiole tehdään toisen makron määrittämä toiminto.

Makrojen lisäksi toimivat koukut, joiden avulla tarinan tekstin osia voi muokata pelin aikana muokata lennosta esimerkiksi tiettyjen ehtojen täytyttyä. Koukkuja muodostetaan hakasulkeita käyttämällä, ja ne voivat kiinnittyä niitä edeltäviin makroihiin kuten for-makroon, jolloin koukun sisälle kirjoitetun tekstin voi näyttää for-loopin avulla moneen kertaan.

Perustason makroja ja koukkuja on helppo käyttää, ja ne tarjoavat pelilogiikkaa varten paljon samoja ominaisuuksia kuin suora JavaScriptin käyttö muissa story-formaateissa. Harlowen rajoittuneisuus tulee vastaan, jos käyttäjä haluaa itse muokata funktioitaan, tai jos pelilogiikka menee rakenteeltaan hyvin monimutkaiseksi.

Snowman-formaatti edustaa Harlowen vastakohtaa ohjelmointiominaisuuksiensa suhteen: Se ei sisällä makroja, mutta on suunniteltu tukemaan passagejen sisään ja tekstin sekaan kirjoitettua JavaScript-koodia. Underscore-, JQuery- ja Marked-kirjastot laajentavat JavaScriptin valmista toiminnallisuutta ja käyttäjä voi interpo-laatioksi kutsuttua ominaisuutta käyttämällä kirjoittaa koodilohkoja myös keskelle lauseita. Tällöin muuttujia voi käyttää suoraan tekstissä.

5.6.1 InkleWriter, Ink ja Inky

Koska InkleWriter ei sisällä varsinaista ohjelmointitukea, tässä kappaleessa käsitellään Ink-skriptikieltä ja Inky-editoria sen kirjoitusvälineenä. Ink eroaa muiden välineiden tukemasta JavaScriptistä hyvin paljon ulkoasultaan. Toisin kuin JavaScript, se on luotu yksinomaan interaktiivisten tarinoiden rakentamiseen ja tämä näkyy sen syntaksissa ja ominaisuuksissa selvästi. Vaikka perustason ohjelmointikäsitteet kuten loopit ja ehdollisuus kuuluvat Inkiin, se sisältää myös suuren määrän omaa termistöä kuten aikaisemmin mainitut knot ja stitch sekä shuffle ja weave.

JavaScriptillä ohjelmoitaessa käyttäjän valintoja vastaanotetaan esimerkiksi luomalla nappeja ja käyttämällä event listener –toiminnallisuutta painallusten seuraamiseen. Inkissä vastaavanlainen perusrakenne luodaan käyttämällä asteriskimerkkiä tekstirivin alussa, ja tällainen rakenne on huomattavasti selkeämpi sekä yksinkertaisempaa luoda kuin edellä mainittu JS-vastine. Valintatekstien yhdistäminen ja sekoittaminen tapahtuu kätevällä syntaksilla hakasulkeita käyttämällä. Niiden avulla samaa tekstin osaa voi käyttää sekä pelaajalle annetun valinnan tekstissä että valintaa seuraavassa osiossa.

Knot- ja stitch-rakenteiden avulla ositetun tarinan kulkua hallitaan divert-ominaisuudella. Divert luodaan tekemällä nuoli jota seuraa knotin tai stitchin nimi. Tällä

tavoin pelaajaa voidaan ohjata tarinan kohdasta toiseen ja diverteissä voidaan ottaa huomioon monimutkaisempaa logiikkaa, joka määrittää minne pelaaja seuraavaksi päättyy tarinassa.

Loop-rakenteita voidaan tehdä käyttämällä +-merkkiä asteriskin sijasta. Tällöin valintaa kutsutaan sticky-valinnaksi eli sen voi tehdä monta kertaa, ja sen yhteyteen voi liittää taustalogiikkaa kuten jonkin numeromuuttujan kasvattamisen. JS:n looppivastineisiin kuten for- ja while-looppeihin verrattuna Inkin syntaksi on tässäkin kohtaa paljon yksinkertaisempi käyttää. Huomionarvoista on että koska sticky-ominaisuus on sidottu tekstivalintaan, sitä ei voi käyttää yhtä vapaamuotoisesti kuin JS:n muuttujan arvoihin perustuvia looppeja. Myös divertejä voi käyttää looppaamiseen, eli pelaajan voi valinnan jälkeen ohjata esimerkiksi takaisin jo kerran näytettyyn knottiin, joka puolestaan voi lopulta johtaa samaan divertiin ja sitä kautta loopata.

Ink sisältää myös korkeamman tason funktioita joilla pystyy seuraamaan pelin tilaan liittyviä arvoja, kuten pelaajan tekemien valintojen määrää. Syntaksi niiden käyttämiseen on samankaltainen kuin JavaScriptin vakiomuuttujiin liittyvä nimeämiskäytäntö: Funktion nimet ovat isoilla kirjaimilla kirjoitettuja ja niitä kutsutaan samoin kuin JS:n funktioita käyttämällä sulkeita nimen jälkeen.

Etenkin suurempiin projekteihin tarkoitetut weave- ja gather-rakenteet tekevät moneen eri polkuun jakautuvan tarinan rakentamisesta tehokkaampaa: Weave muodostuu useasta gatherista, joita luodaan aloittamalla rivi yhdysmerkillä usean asteriskilla tehdyn valintaosion jälkeen. Luotu gather-rakenne yhdistää kaikki edeltävät valinnat johtamaan samaan kohtaan tarinassa eli gatherin tekstiin. Tuloksena syntynyt weave varmistaa, että tarina etenee tiettyyn pisteeseen valintojen aiheuttamista mahdollisista sivuhaaroista huolimatta.

5.6.2 Squiffy

Squiffy toimii ohjelmointituen suhteen samalla tavalla kuin Twinen Snowman-formaatti: JavaScriptiä voi kirjoittaa tarinaan ilman rajoituksia. Toisin kuin Snowman, Squiffy ei tarjoa lisäominaisuuksia kuten valmiita JavaScript-kirjastoja tai muita sisäänrakennettuja toimintoja. Koska Squiffyn sisäänrakennetut komennot ovat

melko yksinkertaisia ja rajoittavat moniulotteisen logiikan luomista, JavaScript toimii parhaiten tukemaan niitä ja laajentamaan toiminnallisuutta silloin kun se on tarpeen. Esimerkiksi Squiffyn omalla set-komennolla luodut muuttujat ovat käytävissä ja muokattavissa myös JavaScriptin kautta, jolloin yhteiskäyttö on vaivatonta.

5.6.3 Quest

Ominaisuudet-kohdassa käsiteltiin graafisen käyttöliittymän kautta luotavia skriptejä. Suora ohjelmointi Questissa toimii Code view -näkökuvan kautta. Halutessaan käyttäjä voi muokata jopa koko projektin kattavaa XML-tiedostoa, mutta se ei välttämättä ole graafista käyttöliittymää tehokkaampaa koska XML-formaattia ei ole suunniteltu interaktiivisten tarinoiden kirjoittamiseen.

Code view löytyy kuitenkin myös skriptejä sivuille lisätessä vaihtoehtona GUI:lle. Tätä kautta käyttäjä voi keskittyä XML:n sijasta yksittäisiin, sivuja koskeviin skripteihin jotka ovat Questille uniikilla ASL-kielellä kirjoitettuja. ASL on syntaksiltaan hyvin samankaltaista kuin JavaScript. Olion käsitettä käytetään osittain samaan tapaan kuin JavaScriptissa: Niille voi luoda attribuutteja jotka tallettavat kyseiseen olioon liittyviä arvoja. Jokaisessa pelissä on valmiiksi luotu player-olio, jolle tällaisia arvoja voivat olla esimerkiksi lista tavaroista joita pelaajalla on tai hahmon kykyihin liittyvä pistearvo. Olio-ohjelmoinnille keskeistä periytyvyyden käsitettä tuetaan Questissa tyyppi-avainsanalla. Gamebook-muotoisissa tarinoissa tyyppien luomiseen ei ole graafista käyttöliittymää, mutta tarinan XML-tiedostoon voi silti lisätä omia tyyppisiä suoraan. Tyyppien avulla oliot voivat periä tietyt attribuutit yhdeltä tai usealta tyyppiltä, jolloin esimerkiksi oliolle hevonen ei tarvitse kirjoittaa erikseen kaikkea eläin-tyyppiin liittyviä attribuutteja, jos hevonen perii eläin-tyypin.

Olioita luodaan create-funktiolla, joka on yksi monista Questin sisäänrakennetuista funktioista. Muihin sisäänrakennettuihin funktioihin kuuluvat ehto- ja looppirakenteet kuten if ja for, joita käytetään samalla syntaksilla kuin JavaScriptissa. Käyttäjä voi tarvittaessa rakentaa omia funktioitaan. Tämä tehdään käyttämällä graafista käyttöliittymää, josta löytyy kaikki mitä perustason funktioiden tekemi-

seen tarvitsee: Funktiolle määritetään nimi, mahdolliset parametrit sekä palautustyyppi. Näiden lisäksi funktiolle voi joko valikkoja käyttämällä tai suoraan kirjoittamalla lisätä itse suoritettavat toiminnot script-osiossa.

Questiin on mahdollista lisätä ASL-skriptien lisäksi myös JavaScript-koodia, mutta Gamebook-formaatissa tämä on haastavaa, eikä graafisessa käyttöliittymässä ole toimintoja JavaScriptin käyttämiseen.

5.7 Julkaisukeinot ja vientiformaatit

5.7.0 Twine

Publish to file -toiminnon avulla Twine luo projektista HTML-tiedoston, joka sisältää koko projektin. Twinen vientitiedoston muuttamista executable-tiedostoksi varten on olemassa käyttäjien tekemiä työkaluja, mutta niissä prosessi on hankalampi eikä virallista tukea tällaiselle toiminnolle ole.

5.7.1 InkleWriter, Ink ja Inky

Inklewriterin share-toiminnolla tarinasta saa luotua linkin, jonka kautta tarinan pelattavaa versiota voi jakaa. Samalla toiminnolla käyttäjä voi myös ladata tarinan itselleen .json- tai .ink-tiedostona. Inky-editorissa .ink-muotoisen tarinan voi viedä web-formaattiin, jolloin tuloksena on JavaScript-, HTML- ja CSS-tiedostot sisältävä kansio.

5.7.2 Squiffy

Squiffy tallentaa kirjoitetut tarinat .squiffy-tiedostoina, joista voi build-toiminnolla luoda HTML, CSS- ja JavaScript-tiedostot. Kuten Twinessä, muuntaminen web-formaatista esimerkiksi työpöytäsovellukseksi sopivaan muotoon vaatii ulkoisten työkalujen käyttöä eikä Squiffystä itsestään löydy tähän muita vaihtoehtoja.

5.7.3 Quest

Questilla kirjoitetut tarinat tallentuvat .aslx-formaatissa, eli Quest Game Source – tiedostona. Publish-toiminnolla pelistä luodaan .quest-tiedosto, jonka voi julkaista textadventures.co.uk-sivuilla. Virallista tukea muille vientiformaateille ei ole, ja Quest on vertailtavista ohjelmista ainoa jossa peliä ei ole mahdollista muuntaa HTML/CSS/JavaScript -verkkoformaattiin.

6 POHDINTA

Tekstiseikkailupelien, ja interaktiivisen fiktion ylipäänsä, kirjoittamiseen suunniteltuja välineitä on ilmestynyt valtavasti viimeisen kahdenkymmenen vuoden aikana. Valikoima on laaja, ja iso osa välineistä on käytettävissä täysin ilmaiseksi, eikä julkaisuakaan yleensä rajoiteta. Tätä selittää ainakin se että kehitystyökalut tulevat usein pieniltä tiimeiltä ja heidän pääasiallinen motivaationsa on tehdä interaktiivisen fiktion tuottamisesta saavutettavampaa. Tässä opinnäytetyössä vertailtiin vain erittäin pientä otantaa, mutta laadukkaita välineitä on varmasti vertailtuja enemmänkin. Vaikka tekstiseikkailupelit ja vastaavanlaiset pelit ovat pieni osa koko peliteollisuutta, välineiden foorumeita selaamalla paistaa selvästi läpi yhteisön intohimo genreen. Tämän valossa interaktiivinen fiktio tulee todennäköisesti jatkamaan elinvoimasena etenkin indie-kehittäjien keskuudessa, ja kehitystyökalujen parantamista jatketaan edelleen.

Kaikki vertailut kehitystyökalut olivat laadukkaita, eikä mitään niistä voi vertailun tuloksena pitää varsinaisesti huonona valintana tässä opinnäytetyössä pohjamateriaalina käytetyön peliprojektin tai muun tyypillisen tekstiseikkailupelin kehittämiseen. Valinnan tekeminen kehitystyökalujen välillä ei ollut kovin helppoa, koska jokaisella oli vahvuuksia ja heikkouksia: Twine loisti etenkin käyttöliittymällään ja sekä sillä että Questilla on todella kattavat viralliset dokumentaatiot ja ohjeet. Squiffy oli vertailtavista nopein oppia käyttämään ja on yksinkertaisuutensa ansiosta hyvä etenkin nopeaan prototypointiin henkilöille, jotka osaavat käyttää JavaScriptiä. Opinnäytetyön tuloksena valittiin Inky ja Ink, joiden isoimmat vahvuudet olivat integraation mahdollisuus Unity-pelimoottorin kanssa sekä helppokäyttöinen mutta tehokas tapa luoda tarinan rakenne Inkin syntaksilla.

LÄHTEET

Engström, H., Brusck, J. & Erlandsson, P. 2018. Prototyping tools for Game Writers. *The computer games journal* 7 (3), 153-172.

Grant, W. 2018. *101 UX Principles: A Definitive Design Guide*. Birmingham: Packt Publishing Ltd.

Horch, J. 2003. *Practical Guide to Software Quality Management*. 2. painos. Norwood: Artech House.

Meifer-Menhard, F. 2013. *Playing the text, performing the future: future narratives in print and digiture*. Berlin: De Gruyter.

Mundy, D. P., Consoli, R. 2013. Here be dragons: experiments with the concept of 'Choose Your Own Adventure' in the lecture room. *Innovations in education and teaching international*, 50 (2), 214-223.

Itch.io. 2022. Top games. Verkkosivu. Viitattu 25.4.2022. <https://itch.io/games>

Kashtan, A. 2011. Because It's Not There: Ekphrasis and the Threat of Graphics in Interactive Fiction. *Digital Humanities Quarterly* 5 (1).

Kolhoff, L., Nack, F. 2019. How Relevant Is Your Choice?: User Engagement and Perceived Agency in Interactive Digital Narratives on Video Streaming Platforms. *Interactive Storytelling*, 73-85. Cham: Springer International Publishing AG.

Mehta, M., Corradini, A., Ontañón, S. & Henrichsen, P. J. 2010. Textual vs. Graphical Interaction in an Interactive Fiction Game. *Interactive Storytelling*, 228-231. Berlin, Heidelberg: Springer Berlin Heidelberg.

Valve Corporation. 2022. Steam Search. Verkkosivu. Viitattu 25.4.2022. <https://store.steampowered.com/search/?tags=31275%2C11014%2C4486&category1=998>