

Marko Kiiskinen

Pelitestaamisen teoria sekä pelitestaaminen työnä



Tradenomi
Tietojenkäsittely
Kevät 2020



KAMK • University
of Applied Sciences

Tiivistelmä

Tekijä(t): Marko Kiiskinen

Työn nimi: Pelitestaamisen teoria sekä pelitestaaminen työnä

Tutkintonimike: Tradenomi (amk)

Asiasanat: pelitestausta, laadunvarmistus, testaus, ohjelmistotestaus

Opinnäytetyön tavoitteena on antaa ohjeistusta hyvään pelitestaukseen, havainnollistaa asia työharjoittelukokemuksien kautta sekä esitellä, millaista pelitestausta on työnä. Työn tulisi toimia eräänlaisena oppaana kaikille pelitestauksesta kiinnostuneille sekä levittää tietoa pelitestaajien työstä ja sen merkityksestä. Työn tilasi Frozenbyte-peliyritys.

Pelitestausta on videopeleille tehtävää laadunvarmistusta. Pelitestaamisella varmistetaan pelin riittävä valmius ja toimivuus, jotta kuluttaja voi nauttia siitä ilman ongelmia. Pelitestauksella on tärkeä rooli pelialan menestyksen ylläpidossa, nyt ja tulevaisuudessa. Pelitestausta tekevät pelitestaajat, jotka ovat yleensä peliyrittäjien palkkaamia. He etsivät pelistä virheitä eli bugeja ja kirjaavat nämä tietokantaan virheraportteina. Virheraportti sisältää kaikki tiedot, jotka ovat tärkeitä bugin korjaamiseen tai toistamiseen kannalta, kuten kuvauksen, koontiversion numeron sekä toisto-ohjeet. Virheraportin perusteella bugia lähdetään korjaamaan, joten on erityisen tärkeää sisällyttää tarpeeksi tietoa sekä oikeaa tietoa.

Ohjelmistotestauksessa on olemassa erilaisia periaatteita, joiden mukaan ohjelmistoa voidaan testata. Pelitestaamisessa näitä samoja teorioita voidaan käyttää tehokkaasti, sillä pelit ovat myös eräänlaisia ohjelmistoja. Esimerkiksi savutestaus, jolla varmistetaan nykyisen koontiversion riittävä vakaus ennen testaamisen aloittamista, voidaan suorittaa myös peleille. Savutestaamisen tekemisen jälkeen tiedetään, että kyseisen version testaaminen ei ole ajanhukkaa, oli paikka pelin tai ohjelmiston testauksesta.

Pelitestaamisen apuna voidaan käyttää useita erilaisia ohjelmistoja. Esimerkiksi Jenkins-ohjelmistoa käytetään savutestaamisen automatisointiin. Tietokantaohjelmisto Hansoft sekä versionhallintaohjelma Tortoise SVN ovat myös erittäin hyödyllisiä bugitietokannan ylläpitoon ja projektinhallintaan. Pelitestaajalle yleensä kaikkein hyödyllisimmät apuvälineet ovat kuitenkin itse peliprojektin sisään tehdyt työkalut. Kun työkalut räätälöidään tiettyä projektia varten, voidaan niistä hioa parhaiten sen projektin tarpeita vastaavat.

Frozenbyte on vuonna 2001 perustettu suomalainen peliyritys. Yritys on parhaiten tunnettu Trine-pelisarjastaan. Trine on fantasiaympäristöön sijoittuva pulma- ja tasohyppely-peli. Frozenbytelä on konttori Helsingin Kuusisaareessa. Frozenbytelä pelitestaaminen on erittäin vapaata ja luovaa. Yrityksessä luotetaan työntekijöiden pelitestaustaitoihin, jolloin yleisesti alalla käytössä olevaa testauslistaa ei tarvita. Testauslistassa on listattu pelin ominaisuuksia, joita testaajan tulisi sen päivän aikana saada testatuksi. Frozenbytelä on tällä hetkellä meneillään kolme projektia: Trine 4, Starbase sekä Boreal Blade.

Pelitestaaminen on erittäin luovaa työtä, jokaisena päivänä odottavat uudet haasteet ja ongelmat ratkaistaviksi. Frozenbytelä testaajat ovat onnekaassa asemassa. He ovat tasa-arvoisia muiden työntekijöiden kanssa. Kaikissa peliyrittäjissä tilanne ei ole yhtä hyvä.

Pelitestaajaksi ei tarvitse monen vuoden koulutusta, kiinnostus ammattiin ja peleihin riittää pitkälle. Edellisestä osaamisesta pelialalta on aina hyötyä. Ohjelmoimisesta, peligrafiikasta, tuottamisesta sekä suunnittelusta voi ammentaa osaamistaan pelitestaamiseen. Kielitaidot, varsinkin erinomainen englannin kielen hallinta on tärkeää. Pelitestaajan ammatissa on hyötyä, jos on luonteeltaan periksiantamaton, kärsivällinen ja sisukas, sillä pelitestaajana tehdään paljon itseään toistavia asioita. Myös looginen päättelykyky on tärkeässä osassa, sillä vastaan saattaa tulla erittäin monimutkaisia bugeja.

Abstract

Author(s): Kiiskinen Marko

Title of the Publication: Theory of game testing and game testing as a job

Degree Title: Bachelor of Information Technology

Keywords: game testing, quality assurance, testing, software testing

The objective of this bachelor's thesis is to provide guidance for good game testing, demonstrated with examples from an internship experience and to introduce game testing as a job. This thesis should function as a guide into the world of game testing, for all readers of varying experience. Spreading awareness about the importance of game testing is at the forefront. This thesis is done in collaboration with the Frozenbyte-company.

Game testing is quality assurance, that is performed on video games. Game testing is done to monitor the quality of the product, so that the consumer can get the most enjoyment out of it. Game testing is very important to the video game industry in maintaining its current and future success. Game testing is done by game testers, who usually work at the game studio, like other developers of the project. Game testers' job is to seek out bugs and inconsistencies from the game. These issues are then written down in a bug report, which is then added into a bug database, where they are easily accessible to the people responsible for fixing said issues. The bug report contains a plethora of useful information, such as reproduction steps, description of the bug, videos and screenshots and the reproduction rate. The game tester's objective is to report bugs and make sure they are fixed, so it is important to get all the information right and include everything that is relevant to the issue.

Software testing theory includes many principals, which can be seamlessly integrated into game testing, video games are a type of a software after all. A good example of this is smoke testing, which is used to determine if the current build version is functional enough, so that further testing can be performed on it. If the build is too broken, it is found out early enough, so that minimal time is wasted testing a dysfunctional build.

It is possible to use different kinds of software and tools to ease the workload of a game tester. For example, the Jenkins -software can be used to automate smoke testing. A database program Hansoft, can be used as a bug database as well as a place to store the feature tasks for other developers. Tortoise SVN is a revision management software, used to back up the project files and enable many people to work on the game simultaneously. But the most useful tools for a game tester are usually those made by the game's developers themselves, as those can be designed to fit the projects' needs the best.

Frozenbyte is a Finnish game development studio, founded in 2001. The company is best known for its platforming puzzle adventure series, Trine. A game tester working for Frozenbyte is free to test what he wants, free from test lists, which are a common practice in some companies. Test lists contain a list of what should be tested that day. Frozenbyte trusts the employees' skills, who are free to use their full creative potential when testing, instead of being restricted by the test list. Frozenbyte has three ongoing projects: Trine 4, Starbase and Boreal Blade.

To become a game tester, a high-level education is not needed, an interest in games and the profession is enough to get you started. However, knowing how game development works before applying for the job is very useful. Any experience one may have in programming, graphics design, game design or production can be useful in game testing too. Good linguistic skills are a must, especially excellent English skills. Personality traits that will prove useful in game testing are perseverance, patience and grit. Because a game testers' job requires them to do a lot of repetition to nail down the hardest issues, the after mentioned traits will prove their worth sooner than later. Logical thinking and pattern forming skills will also help greatly, in finding the reproduction steps for the hardest to crack issues.

Sisällys

1	Johdanto	1
2	Mitä on pelitestausta?	2
2.1	Pelitestaaajat.....	2
2.2	Pelitestauksen merkitys ja lyhyt historia.....	3
2.3	Bugiraportin luonti sekä sen osat.....	3
2.3.1	Otsikko	5
2.3.2	Kuvaus	5
2.3.3	Toisto-ohjeet	5
2.3.4	Koontiversion numero	6
2.3.5	Lisä -dokumentaatio.....	6
2.3.6	Bugien kriittisyystasot	6
2.3.7	Muuta huomioitavaa.....	9
3	Yleisiä testaustapoja	10
3.1	Black box- ja White Box -testaus.....	10
3.2	Toiminnallinen testaus.....	10
3.3	Regressiotestaus	11
3.4	Savutestaus	11
3.5	Apinatestaus.....	11
3.6	Ad hoc -testaus.....	11
3.7	Lokalisaatiotestaus.....	12
3.8	Hyväksyntättestaus	12
4	Pelitestauksen apuvälineet.....	13
4.1	Prowler.io	13
4.2	T Plan.....	13
4.3	Jenkins	13
4.4	Hansoft.....	14
4.5	Tortoise SVN.....	14
4.6	Frozenbyten kehittämät pelikohtaiset työkalut.....	15
5	Frozenbyte-yritys.....	18
5.1	Frozenbyten lyhyt historia	18
5.2	Frozenbyten toimisto sekä yrityksen rakenne	20
5.3	Yleiset työkäytännöt ja työyhteisö.....	20

6	Frozenbyten tapa testata ja pelitestaaminen työnä	22
6.1	Käytettävät ohjelmistot, laitteet sekä intranet	23
6.2	Tämänhetkiset projektit sekä harjoitteluni Frozenbytellä	24
6.3	Pelitestaaminen työnä	26
6.4	Mitä tarvitsee päästäkseen pelitestaajaksi?	27
7	Yhteenveto	29
	Lähteet	31

Symboliluettelo

Game design document (**GDD**), eli pelin suunnitteludokumentti. Peliprojektin suunnittelijoiden luoma dokumentti, johon kirjoitetaan ylös kaikki pelin suunnitteluun liittyvät asiat.

Nintendo Entertainment System (**NES**), on Nintendon kehittämä pelikonsoli.

Technical Certification Requirements (**TCR**), on Microsoftin käyttämä lista, jonka mukaan kaikki Microsoftin alustoilla julkaistavat pelit tarkistetaan.

Technical Design Document (**TDD**), eli tekninen suunnitteludokumentti. Tämän dokumentin kirjoittavat peliprojektin johtavat koodarit. Tiedostoon dokumentoidaan pelin koodausperiaatteet ja käytännöt.

Technical Requirements Checklist (**TRC**), on Sonyn käyttämä lista, jonka mukaan kaikki Sonyn alustoilla julkaistavat pelit tarkistetaan.

Time Stamp (**TS**) -lyhennettä tullaan käyttämään, kun kerron lähdeviittauksissa, missä kohdassa videota kyseinen asia tulee ilmi.

1 Johdanto

Pelitestauksen merkitys kasvaa jatkuvasti, sillä peleistä tulee isompia ja monimutkaisempia kuin aikaisemmin (kuva 1). Tämän suuntauksen avustuksella myös ennakkoluulot pelitestausta ammattina kohtaan ovat pikkuhiljaa hälvemässä. [1.] Haluan tuoda esiin monet pelitestauksen puolet ja kuinka kykenevä testaaja voi hioa testaustekniikoitaan asennoitumisellaan, tietämyksellään ja oikeilla työkaluilla.

Olen tehnyt paljon pelitestausta omalla ajallani vapaaehtoisena ja olen aiheesta muutenkin kiinnostunut, mikä teki pelitestaamisesta valinnan helpoksi. Tietämykseni pelitestaamisesta kasvoi suuresti, kun pääsin Frozenbyte-peliyhteykseen testaajajharjoittelijaksi. Työn tilaaja on Frozenbyte. Esittelen pelitestaamista työnä, sekä millaista se on Frozenbytelä. Opinnäytetyön tarkoitus on avata, mitä pelitestausta on, mitä siihen kuuluu ja antaa vinkkejä hyvään pelitestaukseen, sekä toimia eräänlaisena oppaana pelitestaamisen alkeisiin kaikille aiheesta kiinnostuneille. Opinnäytetyössä käsitellään, kuinka tehdä hyvä bugiraportti ja mitä kaikkea sen tulisi sisältää. Esittelen myös hyviä pelitestaamisen käytäntöjä sekä hyödyllisiä apuvälineitä pelitestaamiseen.



Kuva 1. Warframe-pelin "Fortuna"-päivitys toi toisen avoimen pelialueen kanssa 12 % lisää koodia peliin [2, TS. 02:04:05].

2 Mitä on pelitestaus?

Testauksella tarkoitetaan yleensä sitä, että suoritetaan tietyn tyyppisiä kokeita, joilla varmistetaan kyseessä olevan tuotteen tai ohjelmiston laatu. Laaduntarkistus yleensä kumpuaa siitä, että halutaan tarjota tuotteen ostajalle paras mahdollinen kokemus tai tuote. Pelitestauksella tarkoitetaan nimenomaan videopelien laadunvarmistusprosessia, jossa läpikotaisella testauksella yritetään etsiä virheitä pelin sisästä. Nykyaikana on mahdollista käyttää ohjelmistoja, jotka automaattisesti testaavat tekniseen puoleen liittyviä ongelmia, esimerkiksi palvelimien stressitestaus [3, s. 204] ja koontiversion automaattinen kokoaminen. Pelitestaukseen kuuluu monia eri osialueita, joiden painotus vaihtelee riippuen siitä, onko kyseessä tietokone, konsoli vai kannettavalle laitteelle julkaistava peli.

2.1 Pelitestaajat

Pelitestaajat ovat henkilöitä, jotka ovat yleensä peliyrityksen palkkaamia ja testaavat heille pelejä, etsien niistä virheitä tai puutteita. Näitä löytäessään he kirjoittavat virheraportin, jossa he kertovat, kuinka virheen voi toistaa. Raportit luodaan tietokantaan, jossa ne pysyvät järjestyksessä ja ovat helposti saatavilla. Testaajat myös ovat vastuussa virheiden ”elinkaaresta”; heidän tulee varmistaa, että ongelma tulee korjatuksi, ja testata, että kyseinen virhe ei enää tapahdu korjauksen jälkeen.

Pelitestaajat voi jakaa kahteen ryhmään; sisäisiin ja ulkoisiin testaajiin, sen mukaan, tekevätkö he töitä etänä, vaikkapa kodistaan, vai testaavatko he pelejä yrityksen tiloissa. Ulkoiset testaajat palkataan yleensä projektin loppuvaiheessa, testaamaan melkein valmista peliä, eivätkä he tunne pelin koodia tai sen toimintaperiaatteita, eli he tekevät usein ”Black Box” -testausta. Ulkoisia testaajia ei rajoita tietyn yrityksen konttorin sijainti tai sitoutuminen pitkiin projekteihin, sillä heidän työnsopimuksensa ovat yleensä määräaikaista, mahdollistaen vaikka jokaisen projektin jälkeen työnantajan vaihtamisen. Näin ollen ulkoisten testaajien toimeentulo ei ole niin taattua, mutta he nauttivat suuremmasta vapaudesta ja vaihtelusta kuin täysipäiväiset sitoutuneet sisäiset testaajat.

Sisäiset testaajat ovat peliyrityksen täysiä jäseniä, sillä he työskentelevät paikan päällä, kuten muutkin työntekijät. He sitoutuvat yleensä pitkäaikaisesti peliyrityksen projekteihin ja erikoistuvat yhteen tai useampaan projektin testaamiseen. Tästä johtuen he ovat yleensä läsnä projektissa

sen alusta asti ja tekevät silloin useasti ”White box” -testausta, eli he ovat tietoisia pelin suunnittelusta sekä sen sisäisten järjestelmien toiminnasta ja rakenteesta.

2.2 Pelitestauksen merkitys ja lyhyt historia

1970-luvulla pelit olivat vielä niin yksinkertaisia, että yleensä yksi henkilö riitti testaamaan ja korjaamaan mahdolliset bugit peleistä ja tämä oli lähes poikkeuksetta pelin luoja [3, s. 6]. Tilanne alkoi muuttumaan niin kutsutun 1980-luvun videopeliromahduksen jälkeen, jonka vaikutukset tuntuivat lähinnä Yhdysvalloissa. Pelialan romahdus johtui pääasiassa siitä, että pelit tekivät tekijöilleen niin paljon rahaa, että opportunistit halusivat myös mukaan tahkoamaan rahaa ja alkoivat tuottamaan kasoittain pelejä, jotka olivat laadultaan kyseenalaisia. Tämä aiheutti sen, että laadukkaat pelit, joiden tekemiseen oli nähty vaivaa, hukkuivat kaiken muun sekaan. Näin ollen kuluttajien oli yhä vaikeampi karsia jyvät akanoista, mikä johti myyntien romahtamiseen. [3, s. 8.]

Videopeliromahduksen jälkeen pelintekijöille tuli selväksi, että kuluttajat odottivat laatua peleiltään. Nintendo antoi Amerikan videopelitalouselle uuden mahdollisuuden, kun se julkaisi mullistavan NES-konsolinsa, antaen uutta toivoa mantereen pelintekijöille [3, s. 8–9]. Tästä pelitestaamisen historia alkaa. NES-pelit olivat niin aikaansa edellä, että testaajia tarvittiin, ei enää riittänyt, että koodari totesi oman koodinsa virhevapaaksi [3, s. 9–10].

Kuten historia kertoo, on tärkeää, että videopeleissä pidetään laatu riittävällä tasolla. Muuten kuluttajat menettävät uskonsa alaan ja lopettavat videopelien kuluttamisen. Tämä pätee kaikkiin tuotteisiin, jos kyseinen tuote ei täytä kuluttajan laatuvaatimuksia, ei kuluttaja tule sitä ostamaan uudelleen. Huono laatu, virheet ja muut ongelmat eivät ole hyväksi peliyhtiön maineelle, ei kukaan halua maksaa rikkinäisestä pelistä.

2.3 Bugiraportin luonti sekä sen osat

Ennen virheraportin eli bugiraportin kirjoittamista on tärkeää tietää, mikä on bugin määrittäminen: odottamaton vika, puute, virhe tai epätäydellisyys [4]. Bugin voi määrittellä myös asiana, joka rikkoo pelaajan flow-tilan ja siten häiritsee itse pelaamista [5. TS. 11:30]. Virheraportti on kirjoitus, joka sisältää kaikki oleelliset osat virheen tai bugin toistamiseen. Virheraportti ja bugiraportti ovat käytännössä toistensa synonyymejä.

Ennen kuin voi luoda virheraportteja yrityksen bugitietokantaan, tulisi tutustua projektiin, jota on määrä testata. Kuinka voisit tunnistaa pelissä piileviä virheitä, jos et tiedä, kuinka pelin kuuluisi toimia? Tutustuminen tapahtuu peliä pelaamalla, pelin suunnittelijoiden ja muiden työntekijöiden sekä testaajien kanssa puhumisella. Myös pelin GDD:n ja TDD:n läpi lukeminen on hyvä idea ennen kuin alat testaamaan, sillä niissä on dokumentoituna paljon hyödyllistä tietoa pelistä. Kun osaat erottaa, millainen kokemus pelaajalle halutaan välittää ja tunnet kaikki pelin järjestelmät ja ominaisuudet, olet valmis aloittamaan.

Bugiraportti tulisi luoda vasta, kun tietokannasta on etsitty kyseistä bugia, että ei vahingossakaan raportoida samaa bugia monta kertaa. Kun on varmistettu, että löydetty bugi on uusi, voi tehdä sille raportin. Tapauksessa, jossa bugi on jo järjestelmään dokumentoitu, on hyvä idea lukea se silti läpi ja miettiä, onko jotain tietoja, joita olemassa olevasta raportista puuttuu. Mikäli näin on, voi siihen vastata ja lisätä puuttuvia tietoja. Mitä enemmän relevanttia tietoa, sitä helpompi on bugia lähteä korjaamaan. Siinä tapauksessa, jossa löytyy uusi bugi, mutta toistaminen ei onnistu, on silti kannattavaa luoda sille oma raportti ja kirjoittaa siihen ylös kaikki tiedot, jotka tällä hetkellä tiedetään kyseisestä virheestä. Näin toimimalla varmistetaan, että kyseisen bugin olemassaolo on dokumentoitu, eikä se pääse unohtumaan.

Selkeän bugiraportin kirjoittaminen on tärkeää, koska sen avulla yritetään saada bugi toistettua. Jos bugia ei saada toistettua, raporttiin merkitään ”ei pystytty toistamaan”, jolloin virheraporttia täytyy selventää tai uudelleen kirjoittaa. Virheraporttien kirjoitustapoja on yhtä paljon kuin alalla on pelitestaajia. Tärkeintä on, että raportti on selkeä ja siitä käy kaikki oleellinen ilmi. Yritys, jossa olet töissä, määrittelee, mitä kaikkea bugiraportissa tulisi olla, sekä myös peliprojekti itse määrittää, mitä kaikkea virheraportin tulisi sisältää. Bugiraportin kolme pääkomponenttia ovat otsikko, kuvaus ja toisto-ohjeet, mutta mitä yksityiskohtaisempi raportti on, sen parempi [3, s. 120–123]. Hyviä lisäyksiä raporttiin, edellä mainittujen osioiden lisäksi, ovat koontiversion numero, toistosuhde ja ekstradokumentaatio, kuten kuvat ja videot [3, s. 123–127]. Bugin kriittisyystaso on myös tärkeä, sillä se antaa bugin korjaajalle tiedon bugin vakavuudesta yhdellä silmäyksellä. Bugista riippuen myös ”Muuta huomioitavaa” -osiota saatetaan käyttää bugiraportissa, varsinkin jos bugi on erittäin monimutkainen ja sisältää samankaltaisuuksia muihin bugiraportteihin.

2.3.1 Otsikko

Otsikon tulee olla lyhyt ja ytimekäs, kuin kirjoittaisit uutisotsikon [3, s. 120]. Otsikon tulisi myös olla riittävän kuvaava ja sisältää avainsanoja, jotta muut testaajat saavat otsikosta selville, mihin liittyvä bugi on kyseessä. Avainsanat ovat erityisen tärkeitä. Jos bugitietokantaohjelmiston haakuominaisuus tukee vain yhtä hakusanaa, mikäli hakee useammalla, tulokset eivät ole enää ollenkaan niin tarkkoja. Tämä ehkäisee samojen bugien kirjaamista moneen kertaan, kun testaajien on helppo etsiä bugia muutamalla eri hakusanalla, joiden ainakin täytyisi löytyä kyseisen bugin otsikosta. Otsikkoon saattaa myös sisältyä koontiversion numero sekä mahdollisesti myös pelialusta, jolla kyseinen bugi toistuu tai pelikohtaisia lisämääritelmiä.

2.3.2 Kuvaus

Kuvaus on yksityiskohtaisempi kuin otsikko, mutta ei vielä sisällä kaikkia toisto-ohjeen kohtia vaan avaa enemmän otsikossa kerrottua sisältöä. Kuvauksessa myös kerrotaan kaikki ekstrainformaatio, joka saattaa olla tärkeää bugin korjaamisen kannalta, jota ei bugin toisto-ohjeissa tule ilmi. Kuten kuinka monta testaaaja pelasi, kuinka usein bugi tapahtui ja kaikkea muuta, joka saattaa osoittaa, mistä bugi johtuu ja auttaa sen korjaamisessa. [3, s. 121.]

2.3.3 Toisto-ohjeet

Toisto-ohjeissa pitää käydä läpi yksityiskohtaisesti kaikki askeleet bugin toistamiseen, riittävällä tarkkuudella, jotta mitään ei jää epäselväksi. Epäselvät ohjeet tuhlaavat kaikkien aikaa ja bugiraporttisi tulee takaisin leimalla "can't reproduce", eli ei pystytä toistamaan. [3, s. 123.] On kannattavaa testata vielä bugin toistamista täysin omien ohjeittesi varassa ilman ennakkotietoja. Näin saa selvän kuvan, onko ohjeesi bugin toistamiseen riittävät. Joskus bugiraportin toisto-ohjeet saattavat mennä hyvinkin monimutkaisiksi. Näissä tapauksissa kannattaa ottaa askel taaksepäin ja arvioida bugin alkusyytä uudelleen ja yrittää toistaa samaa bugia yksinkertaisemmilla askelilla. Toisto-ohjeiden jälkeen on vielä "Result"- ja "Expected" -kohdat, eli tulos ja odotettu tulos. Tulos selventää, mitä bugissa tapahtuu ja odotettu tulos kertoo, mitä tässä tapauksessa tulisi käydä. Näitä kohtia silmäilemällä korjaaja saa nopeasti idean, millaisesta bugista on kyse ja mihin muotoon se tulisi korjata.

2.3.4 Koontiversion numero

Väärän versionumeron laittaminen bugiraporttiin aiheuttaa sen, että bugiasi ei todennäköisesti saada toistettua, sillä sitä yritetään tehdä väärässä versiossa pelistä [3, s. 123]. Version numero on erityisen tärkeä, jos pelistä testataan montaa eri versiota yhtä aikaa. Väärä versionumero saattaa ohjata testaamaan koontiversiota, jossa ei vielä edes ole sitä ominaisuutta, josta löysit bugin! Oikean koontiversion laittaminen myös auttaa korjaajaa arvioimaan, mitkä kaikki tai kenenkä muutokset ovat saattaneet bugin aiheuttaa.

2.3.5 Lisä -dokumentaatio

Bugin toistettavuusprosentti tulisi sisältyä jokaiseen raporttiin. Se auttaa myös testaajaa arvioimaan, kuinka vakava bugi on kyseessä. Bugi, jolla on 5/5-toistosuhde, on paljon vakavampi kuin 1/5-toistosuhteellinen bugi. [3, s. 124.] Myös kaikki mahdolliset virhelokit ovat tärkeitä lisätä, mikäli mahdollista, sillä ne saattavat suoraan sisältää tiedot, joita bugin korjaamiseen tarvitsee [3, s. 124– 125]. Mikäli bugin toisto-ohjeet ovat monimutkaiset, kannattaa silloin lisätä kuvankaappauksia tai videoita bugin toistaminen. Näin minimoit väärinkäsityksien mahdollisuuden [3, s. 126– 127]. Myös linkkejä muihin virheraportteihin, jotka liittyvät bugiin, voi olla hyvä lisätä. Sillä yhden bugiraporttiin liittyvän bugin korjaaminen, saattaa aiheuttaa ketjureaktion, joka korjaa kasan aiheeseen liittyviä bugeja.

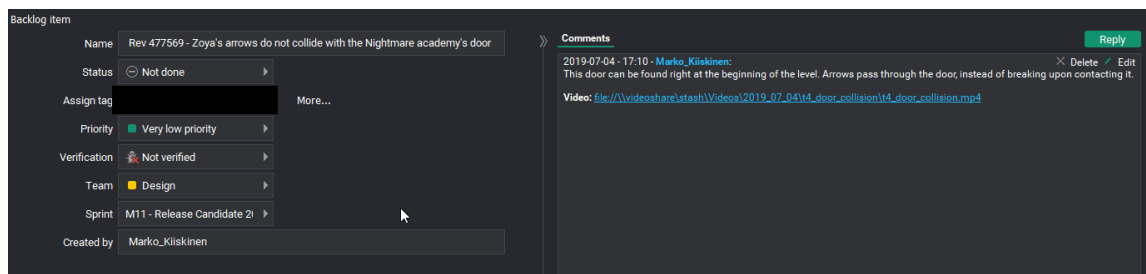
2.3.6 Bugien kriittisyystasot

Vaikka bugien kriittisyystasot ovat yleisesti ottaen Matala, Medium eli keskitaso, Korkea ja Kriittinen, voidaan projektikohtaisesti luoda uusia välitasoja, mikäli tarpeen. Esimerkiksi Frozenbytellä on Trine 4 -projektissa käytössä hyvin matala -taso, joka on vielä matalaa alempana, sekä Boreal Blade -projektissa otettiin käyttöön Korkea+ -taso, joka sijoittuu Korkean ja Kriittisen väliin.

Hyvin matala

”Hyvin matala” -kategorian alle menevät esimerkiksi pienet pelin parantamishdotukset, pienet käytettävyyteen vaikuttavat seikat ja erittäin vaikeasti toistettavat vähäpätöiset bugit. Bugit,

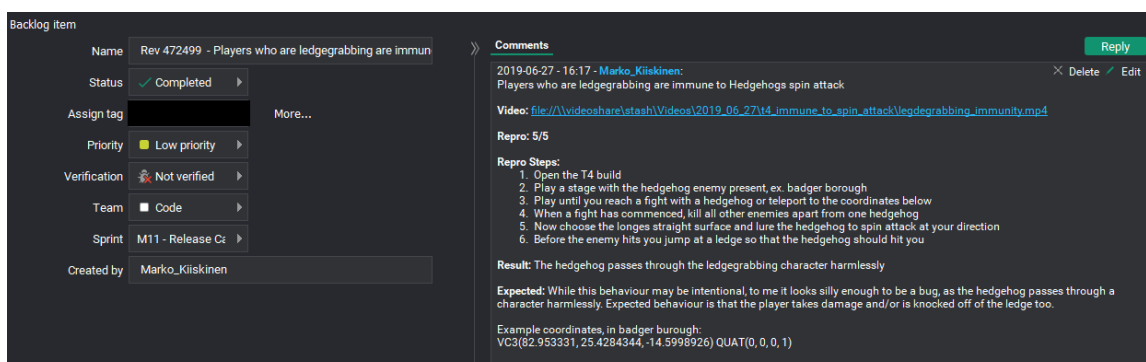
jotka olisivat normaalisti myös matalan tai keskitason bugeja, saattavat mennä tähän osioon, mikäli ne eivät ole vielä keskeisessä asemassa pelin kehittämisen kannalta, toistuvat erittäin harvoin tai ovat vain yksinkertaisesti niin lieviä, että mikäli bugi on vielä julkaistussa pelissä mukana, ei se vaikuta itse pelikokonaisuuteen merkittävästi (kuva 2).



Kuva 2. Kuvankaappaus hyvin matalasta bugista, josta olin tehnyt raportin Frozenbyten Trine 4 Hansoft -tietokantaan.

Matala

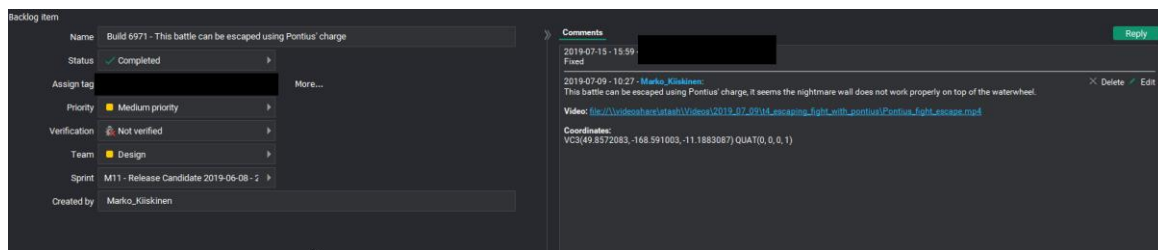
Matalan tason bugit ovat bugeja, jotka eivät vaikuta itse pelin pelattavuuteen, esimerkiksi pienet graafiset bugit, lähes huomaamaton äänen särinä tai harvoin tapahtuvat bugit. Tämän tason bugit saatetaan jättää suurimmaksi osaksi huomiotta, koska ne ovat niin merkityksettömiä, että kehittäjien aika on paremmin käytetty suurempien bugien korjaamiseen. [3, s. 50.] Tästä huolimatta matalan tason bugit tulisi silti raportoida. Matalaan tasoon voidaan myös merkitä bugeja, jotka eivät ole tällä erää kiireisiä korjattavia, mutta jotka tulisi korjata tulevaisuudessa, kun projektissa keskitytään tähän osa-alueeseen (kuva 3). Jos peliä vaikka yritetään saada ensin pelattavaan kuntoon, niin että perusmekaniikat edes toimisivat, eivät esimerkiksi käyttöliittymään liittyvät bugit ole tällöin erityisen tärkeitä.



Kuva 3. Kuvankaappaus matalasta bugista, josta olin tehnyt raportin Frozenbyten Trine 4 Hansoft -tietokantaan.

Keskitaso

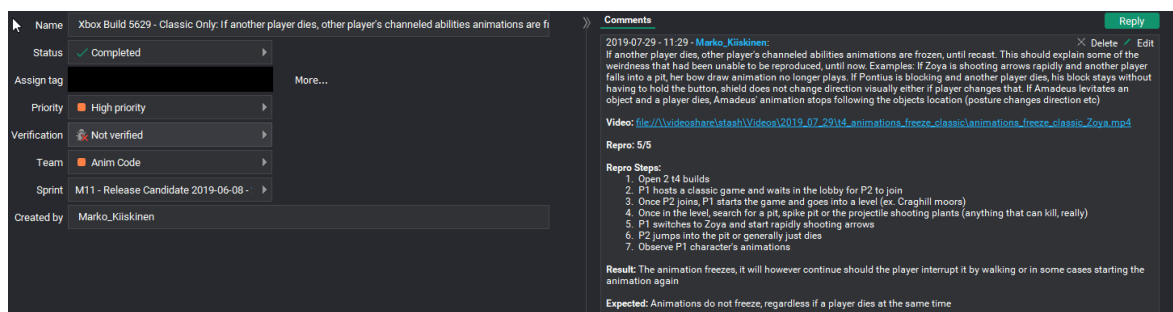
Keskitason bugit tulisi aina korjata. Ne ovat vakavampia ja tapahtuvat useammin kuin matalan tason. [3, s. 50.] Nämä bugit ovat näkyvämpiä ja on todennäköistä, että pelaaja huomaa nämä bugit julkaistussa pelissä, jos niitä ei korjata. Keskitason bugit saattavat vaikuttaa itse pelin pelaamiseen, mutta vain lievällä tavalla, esimerkiksi jos jokin mekaniikka ei toimi aivan oikealla tavalla, mutta ei myöskään estä pelaamista (kuva 4).



Kuva 4. Kuvankaappaus keskitason bugista, josta olin tehnyt raportin Frozenbyten Trine 4 Hansoft-tietokantaan.

Korkea

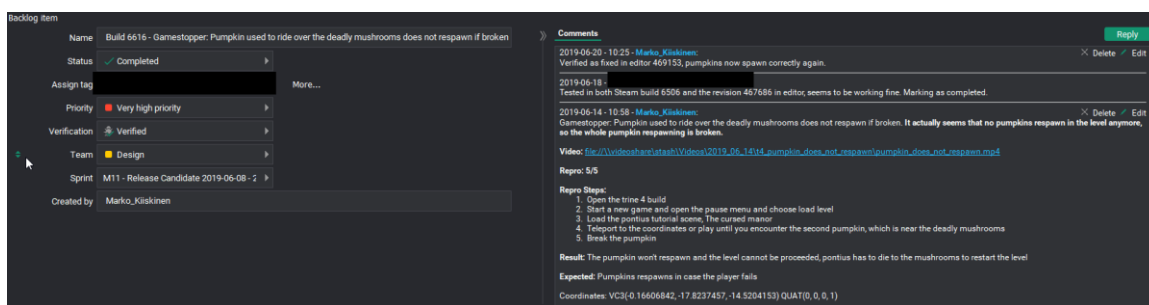
Korkean tason bugit tulee korjata. Peli, joka julkaistaan korkean tason bugien kanssa, on joko huono peli tai se julkaistiin liian aikaisin. Korkean tason bugit vaikuttavat pelaamiseen suuresti ja usein estävät jonkin pelin osan käytön kokonaan. [3, s. 51.] Esimerkiksi jos et voi hypätä, tasosta puuttuvat kaikki viholliset tai jos et voi edetä pelissä eteenpäin, pelistä riippuen nämä voisivat olla myös kriittisiä bugeja. Raja korkean ja erittäin korkean välillä voi olla häilyvä, luokitus pitää aina arvioida tapauskohtaisesti. Kuvassa 5 näkyvä animaatiobugi korkealla prioriteetilla voisi olla yhtä hyvin medium tason bugi, sillä se ei tapahdu usein normaalin pelaamisen aikana, mutta sen saa halutessaan tapahtumaan joka kerta. Tapahtuessaan se ei jää huomaamatta.



Kuva 5. Kuvankaappaus korkeasta bugista, josta olin tehnyt raportin Frozenbyten Trine 4 Hansoft-tietokantaan.

Kriittinen

Kriittinen on kaikkein vakavin bugiluokitus, ja sen luokan bugit tarvitsevat heti huomiota [3, s. 51]. Kriittiset bugit vaikuttavat käyttäjän pelaamiseen erittäin suuresti. Jos julkaisija löytää kriittisen bugin hyväksyntätestauksen aikana (esitellään osiossa ”Yleisiä testaustapoja”), julkaisuprosessi pysäytetään ja peli lähtee takaisin kehittäjälle korjattavaksi [3, s. 51]. Hyviä esimerkkejä kriittisistä bugeista ovat pelin tallennustiedoston korruptoituminen, pelin jäätyminen tai kaatuminen [3, s. 51]. Myös hyvin vakavat pelissä etenemistä estävät bugit saattavat olla kriittisiä bugeja. Trine 4:ssä tämä oli yleinen käytäntö näille niin kutsutuille ”Gamestopper”-bugeille (kuva 6).



Kuva 6. Kuvankaappaus kriittisestä bugista, josta olin tehnyt raportin Frozenbyten Trine 4 Hansoft -tietokantaan.

2.3.7 Muuta huomioitavaa

”Muuta huomioitavaa”-tai ”Notes”-osio sisältää informaatiota, joka ei sovi itse kuvaukseen. Tähän tietoon kuuluu esimerkiksi yhteys toisiin bugeihin, sekä mikäli bugin tapahtuminen johtuu toisesta bugista, jonka korjaamalla tämänkin korjaantuisi. Osiossa voi kertoa myös kyseisen bugin mahdollisia sivuvaikutuksia, jos ne eivät ole pääosassa itse raportissa. Jos vaikka peli kaatuu neljä kertaa viidestä, mutta silloin kun se ei kaadu, peli ei enää yhdistä Internettiin.

3 Yleisiä testaustapoja

Erilaisia testaustapoja ja periaatteita on yhtä paljon kuin testaajiakin, mutta tietyt avaintermit ovat yleisesti käytettyjä alalla. Koska pelit ovat tietynlaisia ohjelmistoja, voidaan ohjelmistotestaamisen testausmenetelmiä tehokkaasti hyödyntää myös pelitestaamisessa.

3.1 Black box- ja White Box -testaus

Black box -testaus on pelin lopputuotteen laadunvarmistustestausta, jossa varmistetaan, että käyttäjä saa tyydyttävän lopputuloksen [5, TS. 3:20]. Black box -testaajat eivät tiedä pelin infrastruktuurista mitään [6], vaan testaavat peliä kuin kuluttaja pelaisi sitä [5, TS. 1:50]. He etsivät lähinnä pelin immersiota rikkovia, luonnollista etenemistä estäviä asioita tai mekaniikkoja, jotka tarvitsevat hiomista, koska tässä vaiheessa pelin kehityskaarta suurimmat peliä rikkovat bugit on yleensä jo kitketty pois.

White Box -testaus tunnetaan myös Clear Box- ja Gray Box -testauksena, jolloin testaaja on läsnä peliprojektissa heti alusta asti, jolloin edes itse pelin lopullista muotoa ei ole välttämättä päätetty [7, TS. 3:08] Näin ollen white box-testaaja voi jopa olla mukana pelin suunnitteluvaiheessa ja antamassa oman mielipiteensä, mihin suuntaan ominaisuuksia pitäisi viedä, jotta ne olisi helpompia testata tai että niiden testaamiseen ei menisi tuhattomasti aikaa [7, TS. 3:50]. Toisin kuin Black box -testaajat, jotka testaavat koko peliä ja sen komponentteja yhdessä, White Box -testaajat saattavat testata yksittäisiä mekaniikkoja erillään, ennen kuin ne liitetään kiinni itse peliin [7, TS. 8:07].

3.2 Toiminnallinen testaus

Toiminnallinen testaus on nimensä mukaisesti testaamista, missä varmistetaan, että peli pystyy käsittelemään kaikki mahdolliset tavat voittaa ja hävitä sekä käsittelemään kaikki pelaajan syötämät komennot [8]. Toiminnan testausta suoritetaan niin, että peliin syötetään tietoa ja katsotaan, vastaako peli tähän tyydyttävällä tavalla [9].

3.3 Regressiotestaus

Regressiotestauksessa vanhaa sisältöä testataan siltä varalta, että pelin uusi versio olisi luonut bugeja jo toimivaan sisältöön [10]. Tällä varmistetaan, että yhden asian korjaaminen ei riko muita asioita [11]. Jos tätä testaamista ei tehtäisi, saattaisi peliin helposti jäädä bugeja alueille, jotka oli testattu jo toimiviksi. Ilman regressiotestausta löydettäisiin nämä bugit vasta siinä vaiheessa, kun koko peliä alettaisiin testaamaan kokonaisuutena. Tämä aiheuttaisi paljon turhaa päänvaivaa ohjelmoijille, joiden täytyisi mennä penkomaan kauan sitten kirjoittamaansa koodia.

3.4 Savutestaus

Savutestaus tunnetaan myös nimellä koontiversion varmistustestaus. Siinä varmistetaan, että kaikki tärkeimmät asiat pelissä toimivat. Testin tuloksena saadaan selville, onko kyseinen koontiversio tarpeeksi vakaa, jotta muiden ominaisuuksien testaus voidaan aloittaa. Jos savutestausta ei tehdä, saattaa kyseinen koontiversio olla niin rikkinäinen, että kaikki yksityiskohtainen testaus olisi vain ajantuhlausta. [12.]

3.5 Apinatestaus

Apinatestauksessa pelin syötetään satunnaisia arvoja ja komentoja, joten se suoritetaan usein ohjelmiston avustuksella, koska eri arvoja saattaa olla liikaa syöttää manuaalisesti [13]. Apinatestauksen päämäärä on saada peli kaatumaan tai tuottamaan muita epänormaaleja tuloksia, joita ei muilla testausmenetelmillä löydettäisi [14].

3.6 Ad hoc -testaus

Ad hoc -testaus on vapaamuotoista testausta. Testaaja saa pelata peliä ja testata kaikkia mahdollisia asioita, joita mieleen juolahtaa, sen sijaan, että testaisi vain asioita, jotka ovat testauslistalla [3, s. 144.]. Ad hoc antaa testaajille mahdollisuuden olla luovia ja käyttää mielikuvitustaan bugien löytämiseen. Ad hoc-testausta tapahtuu paljon pelien avoimissa alpha- ja beta- testauksissa, kun peruskäyttäjät saavat ensi kertaa pelin käsiinsä. Ajoittainen ad hoc -testaus on erityisen tärkeä

sen tyyppisille peleille, joita Obsidian Entertainment tekee, vaikeimmat bugit löytyvät ad hoc sessioiden aikana, sanoo Brandon Adler, Obsidian Entertainmentin testausjohtaja [3, s. 145].

3.7 Lokalisaatiotestaus

Lokalisaatiotestaus pitää sisällään pelin kääntämisen kohdekielelle sekä pelin läpi käymisen, että se ei sisällä mitään kohdemaita loukkaavaa sisältöä. Käännökset täytyy käydä läpi, että ne ovat edes jollain tasolla aiheeseen sopivia, sekä kaikki pelin sisältö ja kuvat, jotta ne eivät sisällä esimerkiksi uskonnollisia symboleja. [3, s. 62.] Jos loukkaavia kohteita havaitaan, ne täytyy poistaa tai korvata kyseisiin peliversioihin. Lokalisaatiotestausta tehdään yleensä hyväksyntätestauksen yhteydessä, sillä konsolivalmistajat toimivat monissa eri maissa ja näin ollen haluavat kohdemaihinsa sopivia pelejä.

3.8 Hyväksyntätestaus

Hyväksyntätestaus eli ”Compliance”-testaus on tavallista konsolipeleille, koska konsolipelien valmistajilla (Microsoft, Sony, Nintendo) on olemassa listat, joiden kaikki kohdat pelin täytyy täyttää ennen kuin se julkaistaan myyntiin kyseiselle alustalle [3, s. 61]. Hyväksyntätestaaja käy läpi näiden listojen kohtia ja testaa, että kaikki listan vaatimukset täyttyvät. Konsolien valmistajat itse käyvät nämä listat läpi jokaisen pelin kohdalla ja lähettävät pelin takaisin korjausehdotuksien kera, jos peli ei läpäissyt vaatimuksia [3, s. 62]. Kaikilla konsolivalmistajilla on omat listansa, Sonylla on TRC, Microsoftilla samantapainen TCR ja Nintendolla Lot Check -lista [3, s. 61]. Nämä sisältävät vaatimuksia, esimerkiksi pelin pitää pysähtyä, kun ohjain menettää yhteyden, peli ei saa sisältää mitään uskonnollisia kuvia ja ennen kuin käyttäjän tallennusdata pyyhitään, pitää pelin ensin varoittaa käyttäjää [3, s. 62].

4 Pelitestauksen apuvälineet

Pelitestaus on työlästä ja kärsivällisyyttä vaativaa työtä. Onneksi on kehitetty ohjelmistoja, jotka automatisoivat testauksen työläimpiä osia. Esittelen useita testausohjelmistoja, jotka voivat olla hyödyllisiä pelitestauksessa.

4.1 Prowler.io

Prowler.io on yritys, joka on erikoistunut tekemään autonomisia päätöksentekojärjestelmiä [15]. Prowler.io on myös luonut tekoälyn, jota voidaan soveltaa peliprojekteihin; se oppii menemään pelikentän läpi mahdollisimman tehokkaasti [16]. Kun kenttään tehdään muutoksia, laitetaan tekoäly toimintaan. Jos tekoälyllä menee paljon aikaisempaa pitempään kentän läpi pääsyyn tai se ei pääse enää kenttää läpi, saattaa kyseessä olla bugi. Prowler.io:n tekoäly mahdollistaa testaajien keskittämisen aikansa monimutkaisempiin testaustehtäviin, eikä vain juoksemaan pelikenttää läpi jokaisen muutoksen jälkeen siltä varalta, että jotain on mennyt rikki.

4.2 T Plan

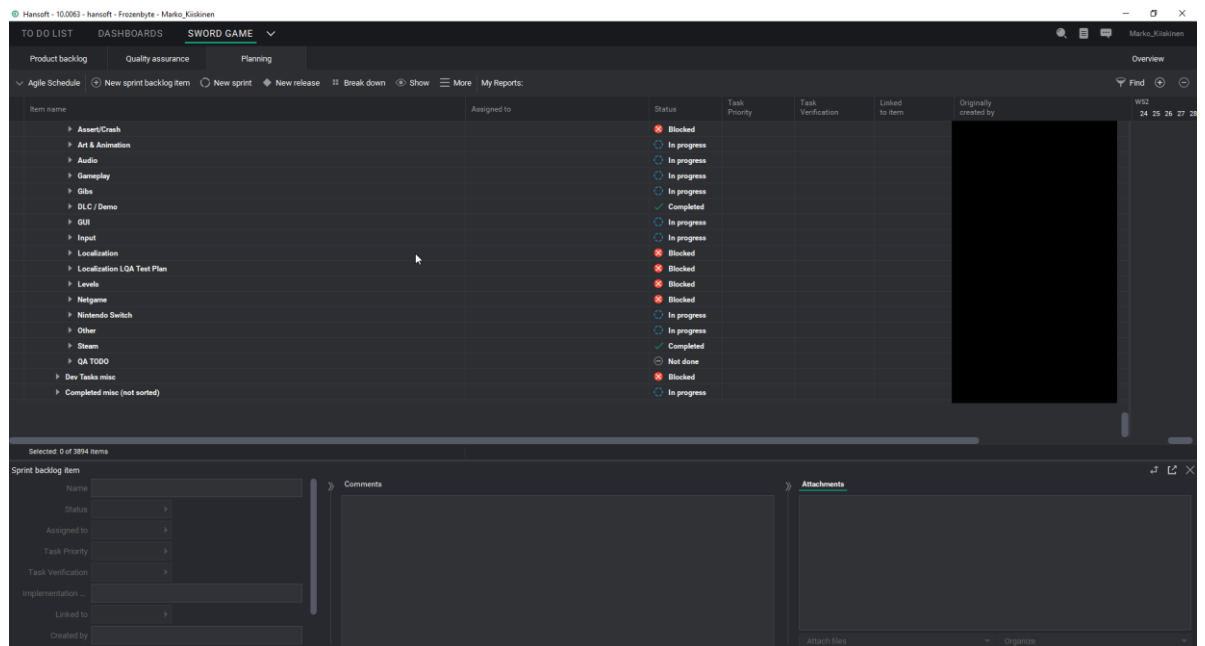
T Planilla on ohjelmisto, joka voi testata pelin visuaalista käyttöliittymää imitoimalla käyttäjän tekemiä toimintoja. T Plan ei tarvitse pääsyä pohjakoodiin ja ohjelmiston tekemät toiminnot voi ohjelmoida tapauskohtaisesti. Se on myös Java -pohjainen ja sisältää OpenGL -tuen. [17.] Näin ollen T Plan soveltuu hyvin mobiilipelien ja verkkosivuille upotettujen pelin testaamiseen.

4.3 Jenkins

Jenkins on avoimen lähdekoodin automaatio-ohjelmisto, joka tarjoaa satoja liitännäisiä koontiversion rakentamiseen, sen lähettämiseen ja projektin automaatioon [18]. Jenkinsiä voi käyttää myös automaattiseen pelitestaamiseen. Jenkins voi automaattisesti tehdä pelistä koontiversion ja testata sitä selvittääkseen, onko se tarpeeksi vakaa edes testattavaksi, eli se käytännössä automatisoi savutestauksen [19].

4.4 Hansoft

Hansoft on kehitetty vastaamaan ketterän ohjelmistokehityksen vaatimuksia [20]. Ohjelmistossa on tietokanta, johon voi luoda tehtäviä, jotka voi taas määrittää kuuluvaksi tiettyyn ohjelmalla luotuun Sprinttiin. Sprint on ketterän ohjelmistokehityksen termi, joka tarkoittaa tiettyä aikaväliä, jonka sisällä siellä olevat taskit tulisi suorittaa. Task on Hansoftissa luotu elementti, joka sisältää tehtävän, joka osoitetaan yhdelle tai useammalle henkilölle. Luodut taskit, olivatpa ne ominaisuuksia, grafiikkaa tai bugitaskeja, voidaan antaa eteenpäin niille henkilöille, joiden vastuualueeseen kyseinen taski kuuluu. Frozenbytelä bugiraportit luodaan näiden taskien sisään ja osoitetaan mahdollisille korjaajille. Kun taski vielä siirretään nykyisen Sprintin alle, ilmestyy se näiden henkilöiden tehtävlistaan. Hansoftin tietokannan ja Sprintin rakenteen voi muokata omalle projektille sopivaksi (kuva 7). Frozenbytelä käytetään Hansoftia sekä bugien että tarvittavien pelin ominaisuuksien taskien dokumentoimiseen. Näin ei aina toimita, vaan joillakin yrityksillä saattaa olla erillinen ohjelmisto bugiraportteja varten.

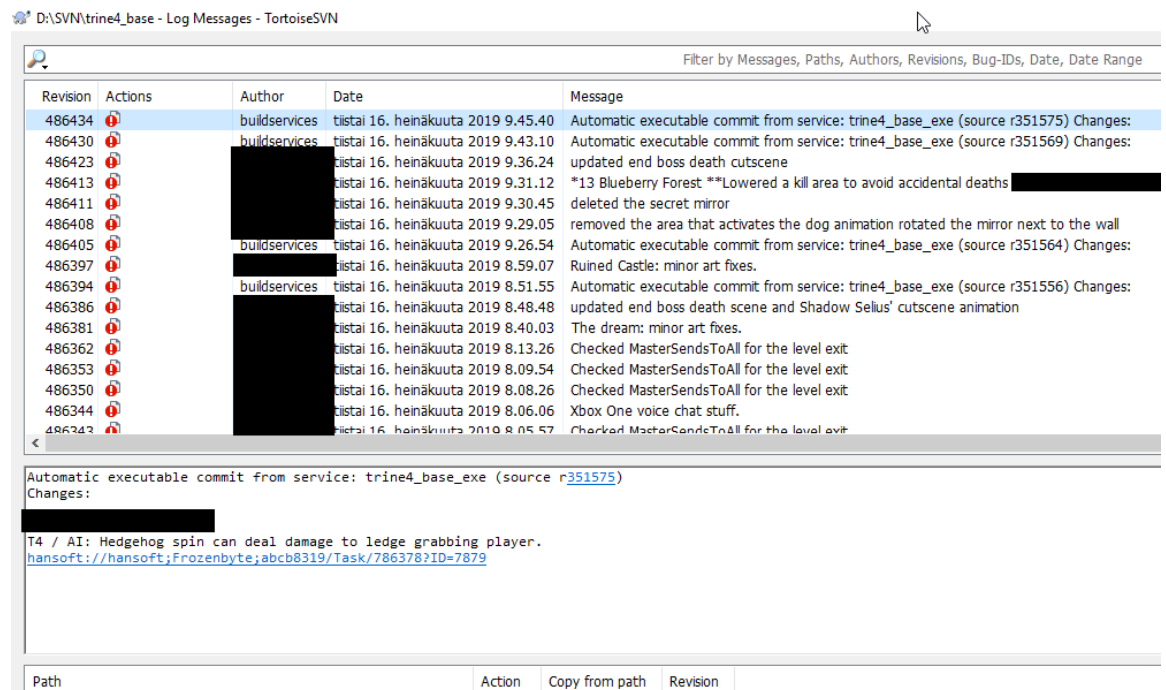


Kuva 7. Kuvankaappaus Frozenbyten Boreal Blade -projektin Hansoft-sprintin rakenteesta.

4.5 Tortoise SVN

Tortoise SVN on ilmainen versionhallintaohjelma, myös kaupallisessa käytössä [21]. Vaikka Tortoise ei ole itsessään testaustyökalu, on se erittäin helppokäyttöinen ja toimiva versionhallintaratkaisu, jonka käyttämisen oppimisesta ei tule ainakaan olemaan haittaa pelitestaajan uralla.

TortoiseSVN on kovassa käytössä Frozenbytellä. Sitä käytetään tällä hetkellä jokaisen käynnissä olevan projektin versionhallintaan, monien entisten projektien lisäksi. Käytän Tortoisea töissä päivittäin, kun päivitän projektien editorin sisältöä uusimpaan versioon, jotta voin testata uusia muutoksia, juuri kun ne on lisätty versionhallintaan tai kun käyn läpi uusimpia tehtyjä muutoksia. TortoiseSVN lisää itsensä tietokoneen hiiren oikean painikkeen valikkoon, josta voit selata kaikkia sen ominaisuuksia ja suorittaa niitä sieltä suoraan, ilman erillisen ohjelman avaamista. Tortoisella on myös oma käyttöliittymänsä, joka avautuu tarvittaessa, kun esimerkiksi päivität editoriasi uudempaan versioon tai katsot, mitä kaikkia muutoksia on projektiin viime aikoina tullut (kuva 8).

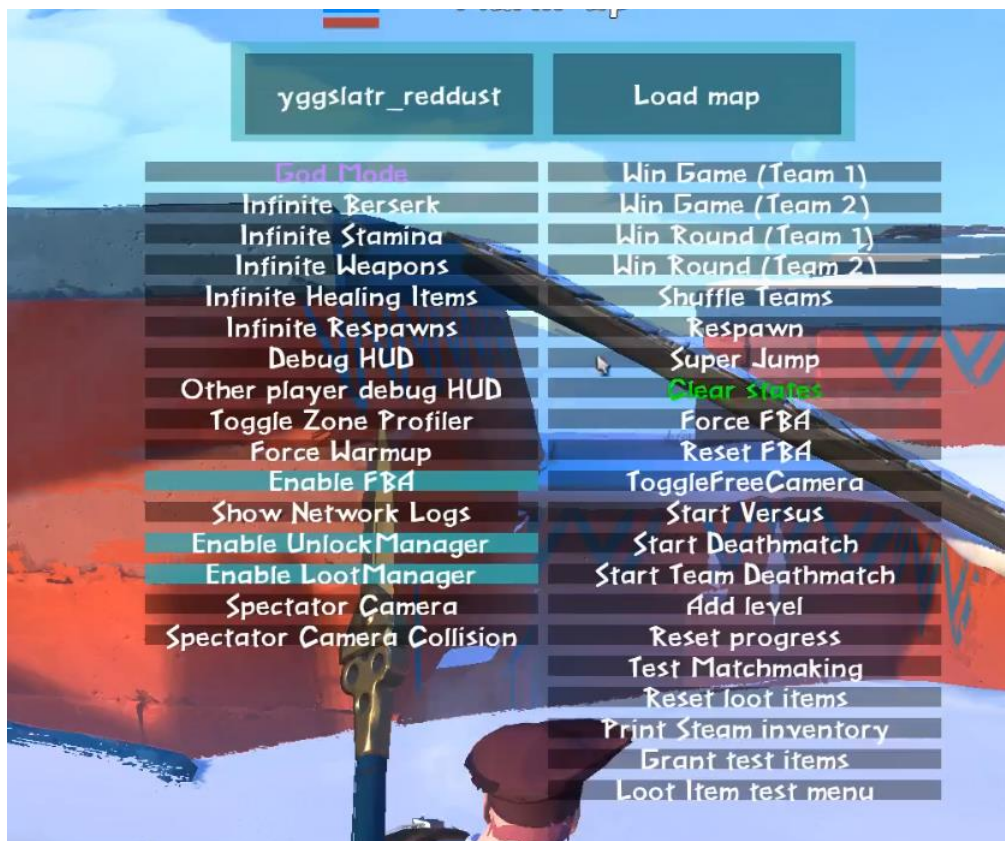


Kuva 8. Kuvakaappaus Frozenbyten TortoiseSVN-tietokantaan tehdyistä viimeaikaisista muutoksista.

4.6 Frozenbyten kehittämät pelikohtaiset työkalut

Yrityksen ulkopuolella kehitetyt apuvälineet ja ohjelmistot ovat erittäin hyödyllisiä ja tarpeellisia, mutta kaikkein tärkeimmät pelitestaajien apuvälineet kehittää yritys itse. Tehokkainta on kehittää apuvälineitä suoraan tiettyä projektia varten. Näin ne voidaan räätälöidä juuri sen projektin tarpeisiin. Pelitestaajan arki helpottuu huomattavasti, kun testatessa jotain tiettyä tason muutosta tai uutta tasoa ei tarvitse aina pelata kaikkia pelin kenttiä läpi, että pääsee edes aloittamaan itse testausta. Frozenbytellä on tapana luoda projekteihin testaamista helpottavia kehittäjän työkaluja, jotka ovat käytössä vain pelin testiversioissa. Nämä työkalut vaihtelevat erittäin paljon projektin tarpeista riippuen. Jos pelissä on paljon kenttiä, tarvitaan silloin komento, joka avaa kaikki

kentät, jotta niitä voi testata. Jos peli on taas esimerkiksi online pelaaja vastaan pelaaja-peli, jossa ei kenttiä ole samalla tavalla, tarvitaan aivan erilaisia työkaluja. Boreal Blade-projektissa oli erittäin hyödyllistä virheenkorjaus menun käyttäminen asioiden testaamiseen, joita yleensä pystyi tekemään vain kerran erässä, esimerkiksi miekan heittämistä (kuva 9). Nämä ”huijaus asetukset” mahdollistivat esimerkiksi miekkojen jatkuvan heittelyn, joka auttoi paljon tällaisten mekaniikkojen testaamisessa.



Kuva 9. Kuvankaappaus Boreal Blade-projektin sisältämästä testauskäyttöön tarkoitettu virheenkorjausmenusta.

Trine 4 -pelissä oli erittäin hyödyllistä kenttien auki saaminen, jotta koko peliä ei tarvinnut pelata joka kerta läpi asti, jos halusi testata vaikka pelin viimeistä pomovastusta. Ja koska Trine 4 on pulmapeli sekä tasohyppelypeli, tuli usein vastaan gamestoppereita eli kohtia, joissa on mennyt jotain rikki tai asioita puuttuu, joista ei voi päästä ohi vain pelaamalla peliä, tuli lyhyen matkan teleporttaustyökalut hyötykäyttöön. Näillä pystyi kätevästi pääsemään näistä gamestoppereista tarvittaessa ohi ja jatkamaan loppukentän testaamista.

StarBase-projekti käyttää Frozenbyten kehittämää automaattista testausysteemiä, jolla voi esimerkiksi testata, tekevätkö aseet oikean määrän vahinkoa, sekä materiaalien kestävyyttä niitä vastaan [22]. Tällä työkalulla voi myös testata ”Yolol”-koodin toimivuutta, joka on pelin oma si-

säinen ohjelmointikieli [22]. Testiympäristöjä voidaan rakentaa manuaalisesti StarBasen editorilla, mahdollistaen monipuolisen testaamisen [23]. StarBasen testaamiseen käytettävä automaattinen testausohjelmisto on rakennettu toimimaan Frozenbyten oman pelimoottorin editorin kanssa, joten sillä voi testata muitakin projekteja kuin vain StarBasea. Testaussysteemi tukee myös automaattisten kuvankaappausten ottamista suoritetuista testeistä, joita voidaan sitten verrata valmiiksi otettuun kuvankaappaukseen. Tästä näkee, millaiseen kuvaan pitäisi päästä, jos ominaisuus toimii halutulla tavalla. Tulevaisuudessa toivotaan, että tämä saadaan täysin automaattiseksi, ilman että ihmisen täytyy verrata kuvia keskenään. On myös mahdollista automatisoida pitempiä itse pelin pelaamiseen liittyviä tapauksia käyttämällä nauhoitettujen komentojen toistoa. On tosin yleensä kannattavampaa testata vain yksittäisiä mekaniikkoja, välttääkseen turhien muuttujien vaikuttamista testeihin. [24.]

5 Frozenbyte-yritys

5.1 Frozenbyten lyhyt historia

Frozenbyte sai alkunsa varsin vaatimattomasti 1999- vuonna, Käpylän lukiossa, kun Lauri Hyvärinen kysyi välitunnilla Juha Hiekkamäeltä, kiinnostaisiko häntä alkaa tekemään tietokonepelejä. Tällöin he eivät vielä juurikaan tunteneet toisiaan, mutta Juha vastasi myöntävästi kaikesta huolimatta. [25, s. 1.] Samana vuonna heille aukesi mahdollisuus osallistua kokeeseen, jonka kautta saattoi saada mahdollisuuden saada opinto-oikeus Helsingin yliopiston tietojenkäsittelytieteen laitokselle [25, s. 2.] He läpäisivät kokeen kuuden muun helsinkiläisen ohella [25, s. 3].

Vuoteen 2001 mennessä he olivat valmistuneet lukiosta, käyneet yliopistolla uudestaan pääsykokeissa saadakseen virallisen opinto-oikeuden ja suorittaneet asepalveluksen. Oli aika tehdä pelejä. Aloittaakseen he perustivat osakeyhtiön, jonka nimeksi tuli lyhyen miettimisen jälkeen Frozenbyte. Virallisesti yritys rekisteröitiin joulukuun seitsemäntenä päivänä. [25, s. 3.]

Vuoden 2003 keväällä oli heillä jo seitsemän työntekijää, joten oli aika hankkia lisää tilaa. Tähän Laurin äitin autotalli oli sopiva [25, s. 4]. He olivat pitkään työstäneet "Disposable Heroes"-nimistä strategiapeliä, mutta kun Juha suoritti pelille valaistustestiä muuttamalla pelin ympäristöt paljon pimeämmiksi ja antamalla yhdelle hahmolle taskulampun, jolla sohiminen tuotti seinille reaaliaikaisesti suuria varjoja, hylättiin nykyinen idea ja aloitettiin yhden demokentän tekeminen "Alien Terror"-nimiselle pelille [25, s. 4– 5].

He ottivat yhteyttä moniin julkaisijoihin, mutta saivat vastauksen vain yhdeltä, ja sekin oli kieltävä vastaus. Syksy oli koittanut, ja oli aika palata takaisin yliopistolle. Vuoden loppuun mennessä Juha ja Lauri olivat tajunneet, että he tarvitsivat vakituiset toimitilat, mikäli halusivat kehittää pelejä ammattimaisesti. He hakivat yrityshautomoon, josta he saivat 40 neliömetrin toimistotilat. Näihin aikoihin heidän korviinsa oli kantautunut, että pelitalot voisivat saada Teknologian ja innovaatioiden kehittämiskeskukselta eli Tekesiltä tukea. [25, s. 5.]

Pitkän paperisodan jälkeen, vuonna 2004 kesällä, Lauri sai puhelun, Tekes oli myöntänyt heille 66 000 euroa tuotekehitysavustusta. Tuen johdosta Frozenbyte pystyi maksamaan palkkaa työntekijöilleen, ei kovinkaan suurta, mutta parempi kuin ei mitään. [25, s. 5.]

Käytyään joulukuussa 2004 "Game Connection"-tapahtumassa Ranskan Lyonissa, esittelemässä nykyistä projektiaan, heihin otti yhteyttä "Dtp Entertainment" -niminen julkaisija Saksasta. 13 heinäkuuta Frozenbyte ja Dtp allekirjoittivat julkaisusopimuksen. [25, s. 6.] Perjantaina 11.11.2005 Frozenbyten ensimmäinen peli "Shadowgrounds", vanhalta nimeltään "Alien Terror", oli julkaistu [25, s. 7].

Julkaisun jälkeinen aika oli vaikeaa, Frozenbyte oli vailla suuntaa ja rahatilanne oli tiukka. Lyhyt aikainen pelastus pulmaan löytyi alihankintaprojekteista sekä Shadowgrounds pelin jatko-osan "Shadowgrounds Survivor" julkaisusta. [25, s. 8– 10.]

Mutta Frozenbyte ei ollut vielä kuivilla, Survivor ei ollut myynyt tarpeeksi, ja talousahdinko syveni [25, s. 10]. Ennen Survivorin kehitystä "Jack Claw"-nimistä projektia oltiin aloitettu kehittämään. Sen kehitys jatkui vielä Shadowgrounds-jatko-osan jälkeenkin, mutta projektia ei saatu kaupatuksi julkaisijoille ja sen kehitys lopetettiin [25, s. 8– 10]. Vuonna 2008 ahdinko oli kaikkein syvimmillään ja työntekijöitä alkoi irtisanoutua [25, s. 10]. Motivaatio ja ideat olivat lopussa, kunnes Frozenbyte päätti ottaa koodari-Jukan harrasteprojektin täyteen kehitykseen. Tämä projekti oli "Trine" [25, s. 10– 11]. Trine muuttui kehityksen aikana alkuperäisestä muodostaan paljon, päätyen nykyiseen kolmen hahmon moninpelin, satumaailmaan sijoittuvaan tasohyppelyseikkailuun [25, s. 11].

Heinäkuun toisena päivänä 2009 kova työ palkittiin. Tällöin julkaistiin Trine, joka sai paljon näkyvyyttä mediassa, enimmäkseen positiivisessa valossa. [25, s. 12.] Heti Trinen julkaisun jälkeen oli jatko-osan kehittäminen lyöty lukkoon, Frozenbyte nyt tiesi, että oli aina oltava projekti valmiina tehtäväksi [25, s. 12– 13].

Vuonna 2011 seitsemäs päivä joulukuuta Trine 2 ilmestyi (kuva 10). Oli mennyt tasan kymmenen vuotta Frozenbyten virallisesta rekisteröimisestä.



Kuva 10. Kuva Trine 2-pelistä [26].

Peliä kehitettiin vuolaasti kauniista maisemistaan sekä moninpelimahdollisuudesta. [25, s. 14.]
Trine 2:n suosio vakautti Frozebyten toimintaa niin, että he palkkasivat paljon lisää työntekijöitä sekä muuttivat nykyiseen toimitilaansa, Helsingin Kuusisaareen [25, s. 16].

5.2 Frozenbyten toimisto sekä yrityksen rakenne

Frozenbyten konttori sijaitsee Helsingissä, Kuusisaarella. Rakennus on kaksikerroksinen omakotitalo. Rakennuksen alakerrassa sijaitsee pelitestaajien tila, kokoustila eli ”takkis”, pukuhuoneet, sauna sekä ruokala. Ruokalassa kaikki työntekijät saavat käydä joka päivä ilmaiseksi syömässä catering-tiimin valmistamia ruokia. Rakennuksen yläkerrassa sijaitsevat kaikki muut työntekijät, jotka on järjestetty tiimeittäin, eri peliprojektien tai oman työnkuvansa mukaan. Toimiston takapihalla sijaitsevat myös asumistilat, minkä alakerrasta löytyy lisää työtilaa.

Frozenbyte koostuu yrityksenä useista eri osa-alueiden työntekijöistä, joista jokainen täyttää tärkeän roolin yrityksessä. Nämä osa-alueet ovat suunnittelijat, tuottajat, käsikirjoittajat, artistit, animaattorit, peliääni ja musiikki, koodarit, pelitestaajat, pelimoottori- ja ”renderer”-koodarit, ”IT” tai suomeksi ATK henkilöt, ”Support”-tiimi eli tuki, HR eli henkilöstöosasto sekä tietenkin siivoojat ja catering. Vaikka nämä osa-alueet saattavat vaikuttaa tiukasti rajatuilta, on mahdollista tehdä kahta eri osa-aluetta yhtä aikaa tai näiden välillä vaihtaen. Esimerkiksi artisti voi tehdä ympäristön piirtämisen välissä myös kenttä- tai pelisuunnittelua.

5.3 Yleiset työkäytännöt ja työyhteisö

Frozenbytelä on käytössä liukuva työaika, malli, joka on erityisen suosittu pelialalla. Liukuvan työajan mukaan työntekijä saa tulla töihin aamulla kello seitsemän ja kello kymmenen välissä ja lähtee sitten iltapäivällä siihen aikaan, kun kahdeksan tuntia on tullut täyteen. Tämä antaa jokaiselle työntekijälle vapautta itse määritellä työaikansa, sovittujen raamien sisällä. Poikkeuksista on mahdollista sopia erikseen, mutta jos töihin tuloaikasi poikkeaa paljonkin muista työntekijöistä, saattaa joinain päivinä esimiehesi kysyä, mikäli olisi mahdollista tulla aikaisemmin tai myöhemmin paikalle, koska joko työtoverisi eivät ole vielä silloin paikalla tai sinulle ei yksinkertaisesti riitä töitä siihen aikaan.

Frozenbyte on tiivis työyhteisö, jossa eri alojen osaajat tekevät saumatonta yhteistyötä. Kaikilla työntekijöillä on käytössä yrityksen muokkaama versio IRC-tekstikommunikointiohjelmistosta, joka on räätälöity Frozenbyten tarpeisiin. Testaajana teen paljon yhteistyötä suunnittelijoiden, artistien sekä ohjelmoijien kanssa. Kun huomaan bugin pelissä, joka on vasta ilmestynyt peliin ja joka haittaa testaamista suuresti, esimerkiksi jos pelistä ovat hävinneet kokonaan äänet, peli kaatuu, kun sen käynnistää tai se jäätyy usein testaamisen aikana. Tällöin laitan viestiä henkilölle, jonka muutoksista tämä ongelma johtuu, jotta hän voi ruveta korjaamaan tilannetta mahdollisimman pian. Muokatun IRC-ohjelman kautta on nopea laittaa viestiä eri henkilöille, jos tarvitsee kysyä suunnittelijalta, kuinka tämän ominaisuuden pitäisi toimia tai kysyä yleisellä kanavalla, kenelle tämä bugiraportti pitäisi ohjata.

Saan myös usein yhteydenottoja muilta työntekijöiltä, jossa he kysyvät tarkennusta bugiraporttiini, heillä on ongelmia itse bugin toistamisen kanssa, tai bugi ei enää toistu ja he pyytävät minua varmistamaan, tapahtuuko kyseinen bugi enää. Tavallista on myös, että saan viestiä, kun jokin tärkeä bugi on saatu korjattua.

Itse testaustiimillä on joka perjantai palaveri, jossa käydään läpi eri projektit ja kuinka ne edistyvät. Palaverin tarkoitus on pitää huoli, että kaikki testaajat ovat tietoisia myös siitä, kuinka muut projektit edistyvät sekä mikä on tällä hetkellä minkäkin projektin päämäärä. Päätestaaja eli esimies vie palaveria eteenpäin. Palaverissa tiedotetaan ajankohtaisista asioista ja peliprojektien johtavat testaajat saavat sanoa pari sanaa projektista ja sen testaamisen edistymisestä. Tämä testaajien sisäinen palaveri on kuitenkin hyvin rento ja vapaamuotoinen, jossa kuka vain saa puheenvuoron.

Perjantaisin on myös Frozenbyten koko yrityksen laajuinen perjantaitapaaminen, tai ”perjantai-paltsu”, kuten sitä yleensä nimitetään. Palaverin tarkoitus on tiedottaa yritystä koskevista asioista, kuten työterveydenhuollon muutoksista, verokorteista, peliprojekteista sekä saapuvista ja lähtevistä työntekijöistä.

6 Frozenbyten tapa testata ja pelitestaaminen työnä

Frozenbytellä testaajien ”hierarkia” mukailee yleisesti alalla vallitsevaa standardia. Ylimpänä on kaikesta pelitestaamisesta vastaava henkilö eli kaikkien testaajien esimies. Hän määrittelee kaikille peliprojekteilte, jotka ovat käynnissä, ja siinä vaiheessa, että niitä kannattaa testata, johtavan pelitestaajan, joka on vastuussa pelin testaamisesta. Näiden kahden roolin alapuolella ovat muut testaajat, jotka esimies jakaa testaamaan projekteja, pelin tarpeen mukaan.

Pelitestaajien esimies, tästä eteenpäin ”päätestaaja”, on vastuussa pelitestaamisen onnistumisesta ja vastaa yrityksen johdolle. Päätestaajalla on suuri vastuu, sillä hän päättää, kuinka paljon hän myöntää testaajia peliprojektia kohden. Näin ollen hänen täytyy seurata jokaisen projektin testaamisen edistymistä, jotta hän tietää, milloin mikäkin projekti tarvitsee enemmän tai vähemmän pelitestaajia. Päätestaaja myös tekee erittäin paljon yhteistyötä muitten osastojen kanssa, jolloin hänelle jää itse pelitestaamiseen paljon vähemmän aikaa.

Johtavat pelitestaajat vastaavat esimiehelleen ja pitävät hänet päivitettyinä peliprojektin testaamisen edistymisestä. Johtavat testaajat myös tarvittaessa ohjaavat muita projektia testaavia henkilöitä keskittymään tiettyihin asioihin ja järjestävät isompia testisessioita, jos se on pelille tarpeellista. Sessioihin osallistuu tarvittaessa muiden projektien testaajia. Johtavat pelitestaajat ovat yleensä vastuussa, että peli täyttää kaikki julkaisijoiden tai konsolivalmistajien vaatimukset, eli he tekevät hyväksyntätestausta. Näin ollen he yleensä pysyvät samassa projektissa alusta loppuun, hetkeksi vain vaihtaen projektia, mikäli on tarve, mutta aina palaten takaisin heidän vastuuprojektiinsa.

Peruspelitestaajilla ei yleensä ole ylimääräisiä osa-alueita, joista he olisivat vastuussa, vaan he pääsevät käyttämään kaiken työaikansa itse pelien testaamiseen. Toisin kuin johtavat pelitestaajat, perustestaajat saattavat testata kahtakin peliä yhtä aikaa ja heidän määräänsä saatetaan muuttaa projektikohtaisesti riippuen kunkin projektin senhetkisistä vaatimuksista ja tilanteesta. Joissain yrityksissä on käytössä niin sanottu ”Checklist” eli testauslista. Tämä lista sisältää listan pelin ominaisuuksista, jotka työntekijän täytyy käydä läpi ja testata [3, s. 97]. Näiden listojen läpikäymiseen menee suurin osa pelitestaajan ajasta, joka voi käydä hyvin yksitoikkoiseksi. Listan läpikäymisen jälkeen se palautetaan johtavalle pelitestaajalle, joka antaa seuraavan lapun läpikäytäväksi [3, s. 97– 134]. Frozenbytellä ei kuitenkaan käytetä testauslistoja, vaan testaamien on hyvin vapaata ja luovaa. Työntekijän taitoon luotetaan, että hän kyllä osaa tehdä työnsä parem-

min ilman mitään listoja tai rajattuja testauskohteita, eli suurin osa testaamisesta on ad hoc-testausta. Listasysteemiä käytetään lähinnä erittäin suurissa peli- tai testauspalveluja tuottavissa yrityksissä, joissa on tuhansia työntekijöitä. Tietenkin myös Frozenbytellä on aikoja, jolloin esimiehemme ohjeistaa meitä keskittämään testauksemme tärkeään ominaisuuteen tai pelin osalueseen, joka on tärkeää saada valmiiksi tiettyyn päivään mennessä. Myös lähellä pelin koontiversioiden eteenpäin lähettämistä julkaisijalle tai konsolivalmistajalle keskitetään testaamista kyseessä olevaan sovellusalustaan, jotta peli olisi mahdollisimman hyvässä kunnossa aikarajaan mennessä. Testauslistoja tulee vastaan lähinnä hyväksyntätestauksen parissa, kun täytyy käydä läpi konsolivalmistajan tekemää listaa siitä, kuinka pelin täytyy toimia eri tilanteissa.

6.1 Käytettävät ohjelmistot, laitteet sekä intranet

Frozenbytellä on käytössä monia ohjelmistoja: itse tehtyjä sekä lisensoituja. Näistä moni on olemassa vain yrityksen oman intra-verkon sisällä, eli yrityksen omassa sisäisessä verkossa. Intran sisällä toimivat Frozenbyten oma wiki, koontiversiokoneet sekä niiden graafinen käyttöliittymä ja työlista. Wikistä löytyvät työntekijöiden omat sivut, kaikkien projektien kehitykseen liittyvää tietoa sekä työkalujen ja ohjelmistojen käyttöohjeita. Wiki toimii Frozenbyten tietopankkina, josta kätevästi saa selville toimiston ajankohtaiset uutiset ja vanhojen projektien dokumentaatiot. Työlista on selaimen avautuva sivu, jonka kautta voi kirjata työtunteja, tehdä tilavarauksia, katsoa, missä tietyn työntekijän työpiste sijaitsee, katsoa, mihin tiimiin joku tietty työntekijä kuuluu sekä tarkistaa, ovatko he nyt paikalla töissä ja nähdä tämänhetkisen ruokalistan. Frozenbytellä on myös yleisesti käytössä heidän itse tekemänsä pelimoottori sekä sen sisällä toimiva automaattinen testausympäristö, josta kerrotaan enemmän testaustyökalujen osiossa. Käytössä on myös Hansoft ja IRC-ohjelmistosta muokattu kommunikointityökalu, joista löytyy lisätietoa samaisesta osiosta.

6.2 Tämänhetkiset projektit sekä harjoitteluni Frozenbytellä

Frozenbytellä on tällä hetkellä kolme julkista projektia. Nämä ovat Starbase, Trine 4 sekä Boreal Blade. Starbase on avaruuteen sijoittuva avoimen maailman hiekkalaatikkomainen peli, jossa pelaajat voivat hankkia resursseja ja rakentaa avaruusaluksia, joilla voi käydä taisteluun reaaliaikaisesti jopa sadan pelaajan kanssa (kuva 11). Trine 4 on Trine -sarjan neljäs osa, joka palauttaa



Kuva 11. Kuvankaappaus StarBase-pelistä [27].

pelin takaisin juurilleen: satumaisiin maisemiin yhteistyön siivittämässä tasohyppelyssä kahdessa ja puolessa ulottuvuudessa (kuva 12). Boreal Blade on monen pelaajan tiimipohjainen taistelupeli keskiaikaisilla aseilla pohjoismaisissa maisemissa.



Kuva 12. Kuvankaappaus Trine4-pelistä [28].

Harjoittelujaksoni alkuun testasin molempia: Starbasea ja Boreal Bladea. Starbase oli erittäin monimutkainen testata aloittelevana testajana. Koska se on niin laaja ja mekaniikoiltaan monimutkainen, oli vaikea keksiä, mistä aloittaa testaaminen. Boreal Blade oli aivan toista maata, se oli maksimissaan kahdeksan pelaajan pelattava online-taistelupeli, paljon yksinkertaisempi kuin sadan pelaajan avoin maailma. Kahdessa projektissa testailua jatkui pari viikkoa, jolloin aloin samalla enemmän ja enemmän lipsua tekemään Boreal Blade-testausta ihan tietoisesti. Tunsin, että sain paljon enemmän aikaan, kun testasin tätä projektia (kuva 13).



Kuva 13. Kuvankaappaus Boreal Blade-pelistä [29].

Boreal Bladesta tehtiin pääprojektini, ja aloin testaamaan sitä täyspäiväisesti. Pelin testaaminen oli hauskaa ja erittäin tuottoisaa. Projektissa ei ollut paljoa testajia sillä hetkellä, joten apuni oli tarpeellista, ja sain paljon aikaan, koska siinä oli paljon bugeja vielä löydettäväksi.

Harjoittelujaksoni jälkeen jäin Frozenbytellet töihin. Pian sen jälkeen minut siirrettiin Boreal Bladesta testaamaan Trine 4 -projektia, sillä sen julkaisuajankohta alkoi lähestymään. Vaihto peliprojektista toiseen oli kiehtova. Kaikki itsestäänselvyydet, jotka olin oppinut Boreal Bladen aikana, täytyi oppia uudelleen, kuten mitkä asiat tässä pelissä ovat bugeja, millä tavalla asioiden kuuluisi toimia, ja millaisiin pelikokemuksiin haettavaisiin asioihin tulisi kiinnittää huomiota. Vaikka nämä asiat täytyi oppia uudestaan, kaikki pelitestaustaitoni sekä ajattelutyylini, jotka opin harjoitteluni aikana, pystyin säilyttämään ja hyödyntämään myös Trine 4:ssä. Apua oli myös siitä, että olin tehnyt paljon testausta Nintendo Switch -konsolilla Boreal Bladen kanssa, joten aluksi vastuualueekseni määritettiin Trine 4:n Switch-testaaminen.

6.3 Pelitestaaminen työnä

Pelitestaaminen on hyvin luovaa työtä, joka päivä eteesi tulevat uudet haasteet ja ongelmat, jotka sinun tulee ratkaista parhaaksi näkemälläsi tavalla. Nämä ratkaisut voivat olla kovankin työn takana, mutta hankalan bugin toistaminen ja sen toistologiikan tajuaminen ovat erittäin palkitsevia tunteita, jotka tekevät tästä työstä vaivan arvoisia. Pelitestaajana olet ikään kuin portinvartija, pysäytät bugit ovelle ja päästät ehjän pelin läpi. Vaikka usein pelitestaajia pidetään arvoltaan alempina työntekijöinä, ovat he myös yhtä lailla kehittäjätiimin jäseniä. Jos pelialalla ei olisi testaajia, olisi pelien laatu ja toimivuus paljon alemmalla tasolla. Frozenbytellä testaajat ovat kunnioitettuja sekä arvostettuja yhtä lailla muiden työntekijöiden tavoin. Pelitestaajien huonompi kohtelu tuntuu onneksi olevan valloillaan enemmän muissa maissa kuin Suomessa.

Frozenbytellä testaamisen tavat voi jakaa karkeasti kolmeen kategoriaan: vapaaseen testaamiseen, osiotestaamiseen sekä ryhmätestaamiseen. Vapaa testaaminen on hyvin samanlaista kuin ad hoc -testaaminen, jota tehdään Frozenbytellä yleensä noin puolet ajasta, riippuen projektista sekä aikataulusta. Vapaassa testaamisessa sinun täytyy käyttää mielikuvitustasi ja osaamistasi mahdollisten ongelmien löytämiseen. Tämä on testaamisen kaikkein vaikein osio, varsinkin uusille testaajille, sillä voi olla aluksi vaikeaa keksiä, mitä haluaa tai kannattaa testata. Tähän on onneksi olemassa useita apukeinoja, joista pääsee alkuun. Frozenbytellä on tapana pitää jokaisen projektin Hansoftin tietokannassa osio, jossa on lista bugeja, joita ei ole saatu toistettua, joihin tarvitsisi löytää toisto-ohjeet tai muista testauskohteista, joita on pyydetty testaamaan. Listan lukemalla voit saada ideoita mahdollisista tutkimisen arvoisista kohteista tai lähteä tutkimaan listattuja ongelmia. Toinen hyvä tapa tehdä vapaata testaamista on ruveta vahvistamaan korjatuiksi merkityjä taskejasi, eli käydä testaamassa, mikäli nämä ongelmat vielä tapahtuvat. Itse pidän varsin paljon tästä lähestymistavasta, sillä siinä saa samalla vahvistettuja taskeja korjatuiksi ja usein löytää uusia bugeja tai saa idean, mitä testata seuraavaksi.

Osiotestaaminen on samankaltaista kuin vapaa testaaminen, mutta testaajan tulee rajoittaa testaamisensa esimiehen tai päätestaajan nimittämään osioon tietyksi aikaa, tai kunnes tietty ongelma on löydetty. Osiotestaamisen päämäärä on saada tietty pelin osa tai ominaisuus mahdollisimman valmiiksi ja ehjäksi ennen tiettyä päivämäärää. Osiotestaaminen on vapaa testaamisen tapaan hyvin luovaa, mutta luovuus tulee rajoittaa pienemmän alueen sisään. Esimerkiksi jos pelin Nintendo switch -versio on lähestymässä julkaisua, saatetaan ohjata osa testaajista keskittymään vain tämän alustan testaamiseen. Tätä testaamisen tapaa voidaan käyttää eri konsolien

sijaan myös tiettyjen ominaisuuksien tai mekaniikkojen massatestaamiseen, kuten käyttöliittymän tai vaikka pelissä kuoleminen testaamiseen.

Ryhmätestaaminen on pelin internetin välityksellä toimivan moninpelin testaamista tai paikallisen moninpelin testaamista monen testaajan voimin. Ryhmätestaaminen on yleistä Frozenbytellä, sillä melkein kaikissa projekteissa on paikallinen moninpeli. Tässä testaamismuodossa yleensä keskitytään testaamaan, että pelin pystyy pelaamaan loppuun ilman ongelmia. Ryhmätestaaminen, varsinkin netin välityksellä, on ainut tapa saada kunnollinen käsitys siitä, kuinka peli toimii monella pelaajalla netin välityksellä. Moninpelistä löytyy usein bugeja, jotka ovat uniikkeja tälle moodille, tehden ryhmätestaamisesta erityisen tärkeää.

6.4 Mitä tarvitsee päästäkseen pelitestaajaksi?

Jo nuorena iässä olin huomaamattani hyvin pelitestauserioitunut. Kun pelasini vapaa-aikani videopelejä, yritin etsiä niistä bugeja sekä rikkoa niitä parhaani mukaan. Mistään en saanut parempia nauruja, kuin pelien rikkoutumisesta odottamattomalla tavalla. Tajusin tämän vasta hiljattain, kun aloin miettimään miksi, halusin alun perin kokeilla pelitestaamista ja miksi nautin siitä niin paljon työnä.

Pelitestaajaksi pääsemiseksi et tarvitse monien vuosien koulutusta, kiinnostus pelitestaamiseen ja peleihin yleensä riittää jo pitkälle. Tietotekniikkaan ja peleihin liittyvät tutkinnot auttavat paljon hakemusvaiheessa sekä itse työssä. Jotta voit rikkoa pelejä mahdollisimman hyvin, täytyy sinun myös ymmärtää, kuinka niitä kehitetään. Jos olet työskennellyt videopeliartistina, suunnitellut tai tuottanut videopelejä tai ohjelmoinut niitä, voit edellisestä osaamisestasi soveltaa tietoa pelitestaamiseen. Artistit tietävät, kuinka animaatiot ja efektit toimivat ja voivat tällöin tunnistaa mahdolliset virheet helposti. He myös osaavat tunnistaa, mikä voisi mahdollisesti aiheuttaa kyseisen virheen tapahtumisen ja antaa korjausehdotuksia. Suunnittelijat osaavat antaa täsmällistä palautetta pelinsisäisistä systeemeistä, esimerkiksi eri aseiden välisestä tasapainosta, pelinsisäisestä taloudesta tai pelin reiluudesta. Tuottajat taas sopivat päätestaajiksi, sillä he osaavat laatia eri ominaisuuksien testaamiselle aikatauluja, mitä kaikkea ehditään testaamaan aikataulun sisällä ja mitä kaikkea täytyy lykätä. Entiset ohjelmoijat taas osaavat kuvitella, kuinka pelin ominaisuuksia on mahdollisesti implementoitu peliin ja täten osaavat etsiä optimaalisia tapoja, kuinka rikkoa niitä. Ohjelmoijan taidot auttavat myös ongelmatilanteissa. Kun bugin tapahtumiselle ei löydy

minkäänlaista logiikkaa, voivat he miettiä oman osaamisensa kautta, mitkä kaikki asiat voisivat yhteisvaikutuksellaan aiheuttaa tämän ongelman.

Kielitaidoista on testaajalle hyötyä, sillä suurin osa peleistä tehdään englannin kielelle ja ainakin Frozenbytellä kaikki bugiraportit tulee kirjoittaa englanniksi. Näin ollen hyvä englannin kielen taito on lähes välttämätön pelitestaajalle. Muidenkaan kielien taito ei mene hukkaan, sillä pelitestaaja voi hyödyntää kielitaitojaan lokalisaatiotestauksessa. Esimerkiksi japaniksi, saksaksi, espanjaksi ja ranskaksi lokalisoidaan paljon pelejä.

Pelitestaajan tulisi olla luonteeltaan periksiantamaton, kärsivällinen ja sisukas, sillä tulet pelamaan saman pelin useita kymmeniä kertoja läpi, useimmiten bugisena ja keskeneräisenä. Eteesi voi myös tulla bugeja, joita et vain millään saa toistumaan, jolloin saatat joutua painimaan sen kanssa useita tunteja, ennen kuin saavutat riittävän ymmärryksen, mistä se johtui. Pelitestaajan tulisi myös omata loogista päättelykykyä, että hän pystyy selvittämään toisto-ohjeet erittäin monimutkaisiinkin tapauksiin. Pitää myös osata antaa periksi, kun tilanne sitä vaatii. Ei ole järkevää tai tehokasta jahdata vaikeasti toistettavaa matalan tason bugia useita tunteja, varmasti löytyy tärkeämpiäkin työtehtäviä. Hyvät pelitaidot auttavat työssä. Nopeasti pelin pelaaminen läpi säästää aikaa ja usein löytyy bugeja, joiden löytämiseen ja toistamiseen tarvitaan erittäin tarkkoja ja nopeita näppäinkomentoja. Mutta jos menee liian nopeasti läpi, saattaa jäädä huomaamatta graafisia tai muita virheitä, jotka hitaammat pelaajat löytävät. Pitää osata hiljentää vauhtia ja pelata kuten kokemattomat pelaajat, jotta saa kaikkein kokonaisvaltaisimman kuvan ja parhaan kattavuuden bugien etsimiseen.

7 Yhteenveto

Pelitestaukseen on tehty niin pitkään kuin videopelejäkin, mutta erillistä ammattia siitä ei syntynyt ennen kuin videopeleistä tuli tarpeeksi monimutkaisia, eikä koodinkirjoittaja enää riittänyt testaamaan omaa koodiaan tarpeeksi laajasti. Laadunvarmistustyöllä varmistetaan, että kuluttaja saa käsiinsä parhaan mahdollisen tuotteen. Pelitestaaja on henkilö, joka suorittaa videopeleille laadunvarmistusta etsimällä pelistä virheitä ja bugeja. Testaaja kirjoittaa löydettyistä ongelmista virheraportin, jonka perusteella sitten vastuussa oleva henkilö pyrkii korjaamaan sen.

Bugilla tarkoitetaan odottamatonta vikaa, puutetta, virhettä tai epätäydellisyyttä. Bugiraportit kirjoitetaan jokaisessa yrityksessä hieman eri tavalla, mutta yleensä niihin sisältyy otsikko, kuvaus, toisto-ohjeet, koontiversion numero, bugin kriittisyystaso sekä lisädokumentaatio, jonka alle lukeutuu esimerkiksi videot ja kuvankaappaukset virheestä. Bugien kriittisyystasot ovat matala, keskitaso, korkea sekä kriittinen. Asteikkoa voidaan muuttaa tarpeen tullen, jolloin saatetaan lisätä projektikohtaisia välitasoja.

Pelitestauksessa voi käyttää hyödyksi ohjelmistotestauksen teoriaa ja periaatteita, sillä pelit ovat myös eräänlaisia ohjelmistoja. Esimerkiksi Black- ja white box-testaus, toiminnallinen testaus, savutestaus ja lokalisaatiotestaus ovat hyödyllisiä ohjelmistotestauksen periaatteita, jotka voidaan helposti soveltaa pelitestaukseen. Pelitestauksessa voidaan käyttää myös apuna erilaisia ohjelmistoja tehostamaan pelitestausta, esimerkiksi Hansoft, Tortoise SVN sekä yrityksen itse tekemiä työkaluja.

Frozenbyte on suomalainen peliyritys, joka perustettiin vuonna 2001. He julkaisivat ensimmäisen pelinsä Shadowgrounds vuonna 2005. Frozenbyten toimisto sijaitsee Helsingissä, Kuusisaarella. Pelitestaajien hierarkia Frozenbytelä koostuu kolmesta roolista: päätestaajasta eli pelitestaajien esimiehestä, projektikohtaisista johtavista pelitestaajista, sekä perus pelitestaajista, jotka saattavat testata useaakin projektia kerralla ja heitä saatetaan vaihdella projektista toiseen useammin.

Frozenbytelä on kirjoittamisen ajankohtana meneillään kolme eri projektia: Starbase, Trine 4 ja Boreal Blade. Työharjoitteluni aikana testasin Frozenbytelä hieman Starbasea, mutta enimmäkseen Boreal Bladea, sillä se oli helpompi testattava Starbaseen verrattuna. Harjoittelujaksoni jälkeen sain Frozenbytelä työpaikan ja aloitin Trine 4-pelin testaamisen. Frozenbytelä testaamista voi jakaa kolmeen kategoriaan: ryhmätestaamiseen, osiotestaamiseen sekä vapaaseen testaamiseen. Näistä tavoista yleisin Frozenbytelä on vapaa testaaminen.

Pelitestaaminen on hyvin luovaa työtä, sisältäen paljon ongelmanratkaisua sekä loogista ajattelua. Tästä johtuen pelitestaajaksi ryhtymiseen ei tarvitse monen vuoden koulutusta vaan, kiinnostus ammattiin sekä peleihin riittää. Hyvä englannin kielen taito on välttämätön pelitestaajalle, sillä pelit tehdään yleensä englanniksi. Mikäli olet suorittanut tutkintoja tai työskennellyt esimerkiksi koodarina tai artistina, voit ammentaa osaamista edellisestä työstäsi. Pelitestaajan työssä hyötyy periksiantamattomuudesta, kärsivällisyydestä sekä sisukkuudesta, sillä tulet testaamaan paljon keskeneräisiä pelejä sekä tekemään samaa asiaa useaan kertaan etsiessäsi tai toistaessasi bugeja.

Lähteet

- 1 Clever Sean. QA & Localization: Quality Careers. MCVuk 21.09.2017
<https://www.mcvuk.com/development/ga-localisation-quality-careers>, luettu 10.10.2018.
- 2 Steve Sinclair, de_steve's mobile stream. 08.11.2018, 02:04:05.
<https://www.twitch.tv/videos/333262776##>
- 3 Levy L, Novak J. Game development essentials: game QA & testing. Clifton Park: Delmar Cengage Learning; 2009.
- 4 Merriam- Webster dictionary. Definition of bug <https://www.merriam-webster.com/dictionary/bug>, luettu 26.11.2018.
- 5 Trendane Sparks, L2QA (Learn To Quality Assurance) Part 1 - Blackbox Testing. 12.06.2013. <https://www.youtube.com/watch?v=OLayCNOPWIo>
- 6 Software Testing Fundamentals, Differences Between Black Box Testing and White Box Testing. <http://softwaretestingfundamentals.com/differences-between-black-box-testing-and-white-box-testing/>, luettu 26.11.2018.
- 7 Trendane Sparks, L2QA (Learn To Quality Assurance) Part 2 - Embedded Testing. 14.06.2013. <https://www.youtube.com/watch?OLayCNOPWIo=&v=WUzC2tiJoE>
- 8 Tutorials Point, Functional Testing https://www.tutorialspoint.com/software_testing_dictionary/functional_testing.htm, luettu 25.11.2018.
- 9 Software Testing Fundamentals, Functional Testing <http://softwaretestingfundamentals.com/functional-testing/>, luettu 25.11.2018.
- 10 Testbytes. Game Testing tutorial- All the information is here!
<https://www.testbytes.net/blog/game-testing-tutorial/>, luettu 18.11.2018.
- 11 Software Testing Fundamentals, Regression Testing <http://softwaretestingfundamentals.com/regression-testing>, luettu 25.11.2018.
- 12 Software Testing Fundamentals, Smoke Testing <http://softwaretestingfundamentals.com/smoke-testing/>, luettu 25.11.2018.
- 13 Tutorials Point, Monkey Testing https://www.tutorialspoint.com/software_testing_dictionary/monkey_testing.htm, luettu 25.11.2018.

- 14 Software Testing Help, What is Monkey Testing in Software Testing? <https://www.softwaretestinghelp.com/what-is-monkey-testing-in-software-testing/>, luettu 25.11.2018.
- 15 Prowler.io, Careers <https://www.prowler.io/careers>, luettu 26.11.2018.
- 16 Prowler.io, AI tools for automated game testing <https://www.prowler.io/blog/ai-tools-for-automated-game-testing>, luettu 26.11.2018.
- 17 T Plan, Automation Tool for Desktop and Mobile Game Testing <https://www.tplan.com/game-test-automation/>, luettu 26.11.2018.
- 18 Jenkins, Jenkins home page <https://jenkins.io/>, luettu 26.11.2018
- 19 Saxena Mohit. Automation Testing with Jenkins: The Game Changer in Test Automation. HCLtech 19.10.2016, <https://www.hcltech.com/blogs/automation-testing-jenkins-game-changer-test-automation>, luettu 26.11.2018
- 20 Perforce, Hansoft Agile Planning Tool <https://www.perforce.com/products/hansoft>, luettu 15.07.2019
- 21 TortoiseSVN, About TortoiseSVN <https://tortoisesvn.net/>, luettu 15.07.2019
- 22 Keskustelu Henri Päärnilan kanssa Frozenbytelä käytössä olevista automaattisista testausohjelmistoista. 09.08.2019. Ei saatavilla julkisesti
- 23 StarBase automated testing, wikiartikkeli Frozenbyten sisäisessä wikissä. Ei saatavilla julkisesti, luettu 09.08.2019
- 24 Automated testing, wikiartikkeli Frozenbyten sisäisessä wikissä. Ei saatavilla julkisesti, luettu 09.08.2019
- 25 Frozenbyte Yritystarina. Ei saatavilla julkisesti, luettu 24.06.2019
- 26 Kuva 10, Trine 2: Complete Story, Steam <https://store.steampowered.com/app/35720/>, luettu 10.08.2019
- 27 Kuva 11, Starbase, Steam <https://store.steampowered.com/app/454120/>, luettu 10.08.2019

- 28 Kuva 12, Trine 4: The Nightmare Prince, Steam <https://store.steampowered.com/app/690640/> luettu 19.10.2019
- 29 Kuva 13, Boreal Blade, Steam <https://store.steampowered.com/app/598780/>, luettu 10.08.2019