

Tommi Korvola

PELITESTAUKSEN MENETELMÄT JA HAASTEET



JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2022

TIIVISTELMÄ

Korvola, Tommi

Pelitestauksen menetelmät ja haasteet

Jyväskylä: Jyväskylän yliopisto, 2022, 25 s.

Tietojärjestelmätiede, kandidaatintutkielma

Ohjaaja(t): Clements, Kati

Peliteollisuus on kasvanut vauhdilla jo vuosikymmeniä. Alan rahavirta on ylittänyt jo elokuvateollisuuden vuositulot, ja yritysten välillä on kova kilpailu julkaisujen menestyksestä. Pelikehitys on monin tavoin muun ohjelmistokehityksen kaltaista, mutta luova ympäristö, jossa pelejä kehitetään, tekee kehitystoiminnasta vaikeasti hallittavan. Monet pelit joutuvatkin hylätyksi tai epäonnistuvat, kun joko itse työ tai tulos ei vastaa vaatimuksia. Tämän kirjallisuuskatsauksen tavoitteena on selvittää pelitestauksen haasteita peliyritysten kehitystoiminnassa. Tutkimuksessa kerättiin tieteellisestä kirjallisuudesta testimenetelmiä ja niihin liittyviä haasteita. Aineistoa haettiin Google Scholarin ja JYKDOKin kautta. Käytettyjä hakusanoja olivat muun muassa *"game development"*, *"software development"*, *"game testing"* ja *"automated testing"*. Tutkimuksessa löytyi joitakin manuaalisia- ja useita automaattisia testimenetelmiä, sekä niiden haasteita. Varsinkin automaattisiin testimenetelmiin liittyi monia käytännön vaikeuksia, ja kirjallisuuskatsauksessa esitettyjen haasteiden ratkaiseminen tarjoaisikin tutkijoille kattavan tutkimusaiheen. Pelikehittäjät voivat hyödyntää kirjallisuuskatsauksessa koostettuja menetelmiä harkitessaan projekteihin sopivia testimenetelmiä.

Asiasanat: ohjelmistokehitys, pelikehitys, bugi, testaus, automaattinen testaus, manuaalinen testaus

ABSTRACT

Korvola, Tommi

Videogame testing methods and challenges

Jyväskylä: University of Jyväskylä, 2022, 25 p.

Information Systems Science, Bachelor's thesis

Supervisor(s): Clements, Kati

The games industry has grown rapidly for many decades now, and already exceeds the movie industry in revenue. This success drives hard competition and companies strive for successful releases. Game development follows many aspects and methods of traditional software development, but the creative space in which games are made adds an extra layer of challenge. Many games do fail as a result when the end product does not match the desired outcome. This literature review focuses on mapping the methods and challenges involved in testing games. The material was gathered using Google Scholar and JYKDOK using search terms such as 'game development', 'software development', 'game testing', and 'automated testing'. The findings include some manual, and several automated test methods, as well as their challenges. Many of the automated test cases were theoretical or deemed unsuitable for practical use, but present a rich research field for grounding the theory in practice. Additionally, game developers can use the findings presented in this paper to map desirable test methods for their games.

Keywords: software development, game development, bug, testing, automated testing, manual testing

KUVIOT

Kuvio 1 Pelikehityksen vaiheet (Annanpera, Yli-Kantola, Sauvola, Heinonen & Siira, 2018).....	9
Kuvio 2 Testaus peli- ja ohjelmistokehityksessä (Politowski ym., 2022).....	10

TAULUKOT

Taulukko 1 Kolme pelitestauksen pääkohdetta.....	14
Taulukko 2 Automaattisen testauksen kategorioita ja menetelmiä	17
Taulukko 3 Testimenetelmiä ja kehittäjien suhtautuminen	18
Taulukko 4 Pelitestauksen menetelmät ja haasteet	20

SISÄLLYS

TIIVISTELMÄ	2
ABSTRACT	3
KUVIOT	4
TAULUKOT	4
SISÄLLYS.....	5
1 JOHDANTO.....	6
2 PELIKEHITYS.....	8
2.1 Pelikehityksen määritelmä	8
2.2 Pelikehitys suhteessa ohjelmistokehitykseen	10
3 PELITESTAUS	12
3.1 Määritelmä.....	12
3.2 Testaus.....	12
3.2.1 Testaamisen merkitys	12
3.2.2 Mitä testataan?	13
3.2.3 Miten testataan?.....	14
4 PELITESTAUKSEN HAASTEET	19
5 YHTEENVETO JA JATKOA.....	23
LÄHTEET	25

1 JOHDANTO

Videopeliteollisuuden arvo vuonna 2021 oli 180.3 miljardia (Wijman, 2021). Vastaavasti elokuva-, TV-, ja suoratoistomarkkinoiden yhteenlaskettu arvo samana vuonna oli 328.2 miljardia (Rivkin, 2021). Lisäksi suurimmat peliprojektit vastaavat budjetiltaan menestyselokuvien budjetteja (Ahmad, Barakji, Shahada & Anabtawi, 2017). Schmalz, Finn & Taylor (2014) painottavat oikeanlaisen kehitysstrategian merkitystä onnistuneelle pelikehitysprojektille. He toteavat testauksen yhdeksi kehitysstrategian avaintekijäksi, joka jää usein kiireessä liian vähäiseksi. Testauksen osuus ohjelmistokehityksessä on yli 50 % (Politowski, Guéhéneuc & Petrillo, 2022) ja kuten muussakin ohjelmistotuotannossa, pelikehitysprojekteissa on omat vaikeutensa. Tässä kirjallisuuskatsauksessa tarkastellaan pelien testausmenetelmiä. Tutkimuksen keskeisiä käsitteitä:

- **Agentti:** Ohjelman osa, joka pystyy suorittamaan tehtäviä käyttäjän puolesta (Nwana, 1996).
- **Bugi:** Virhe ohjelmassa, joka aiheuttaa ohjelman kaatumisen, jumittumisen tai toimimisen virheellisesti (suomisanakirja, 2022). Lisäksi Lewis, Whitehead & Wardrip-Fruin (2010) jakavat bugin kolmeen osaan: vika (fault), virhe (error), epäonnistuminen (failure). Vika voi johtua ihmisestä tai laitteistosta, ja johtaa järjestelmän virheeseen. Virhe on vian esiintymä ohjelman toiston aikana ja voi johtaa epäonnistumiseen. Epäonnistuminen on havaittava eroavaisuus järjestelmän normaalitoiminnasta.
- **Ohjelmistokehitys:** Toimintaa, joka johtaa ohjelmistotuotteisiin. Kehitys voi sisältää uuden tuottamista, muokkaamista, uudelleenkäyttöä, uudelleensuunnittelua, ylläpitoa tai muuta toimintaa, jotka johtavat ohjelmistotuotteisiin (New Product Development Glossary, 2022).
- **Pelikehitys:** Ohjelmistokehitystä, jossa yhdistyy monialainen osaaminen ja jonka tarkoituksena on luoda interaktiivinen ohjelma.
- **Testaus:** Ohjelman analysointiprosessi, jossa verrataan ohjelman nykyistä tilaa määritettyyn tavoitetilään ja arvioidaan sen ominaisuuksia (Hernández Bécares, Costero Valero & Gómez Martín, 2017).

Tutkimuksen tavoitteena on selvittää pelitestauksen nykytilannetta, ja kartoittaa alueita, joissa testauskäytänteitä voisi parantaa. Tutkimuskysymys on *”Mitä haasteita pelitestauksen menetelmiin liittyy?”*.

Tutkimus toteutetaan kirjallisuuskatsauksena. Aineiston haussa on käytetty sivustoja Google Scholar ja JYKDOK. Kirjallisuuskatsauksessa käytetty aineisto koostuu tutkimuksista, artikkeleista ja konferenssijulkaisuista, sekä peli- ja viihdeteollisuuden yritysten tiedotteista ja julkaisuista.

Tutkimuksen seuraavassa luvussa määritellään pelikehityksen käsite ja kuvataan pelikehityksen suhdetta ohjelmistokehitykseen. Kolmannessa luvussa määritellään pelitestauksen käsite ja kuvataan pelitestausta tarkemmin. Neljännessä luvussa tarkastellaan testauksen haasteita. Viidennessä kootaan kirjallisuuskatsauksessa löydetyt testimenetelmät ja haasteet. Yhteenvedossa kerrataan tutkimuksen sisältö ja käsitellään jatkotutkimusaiheita.

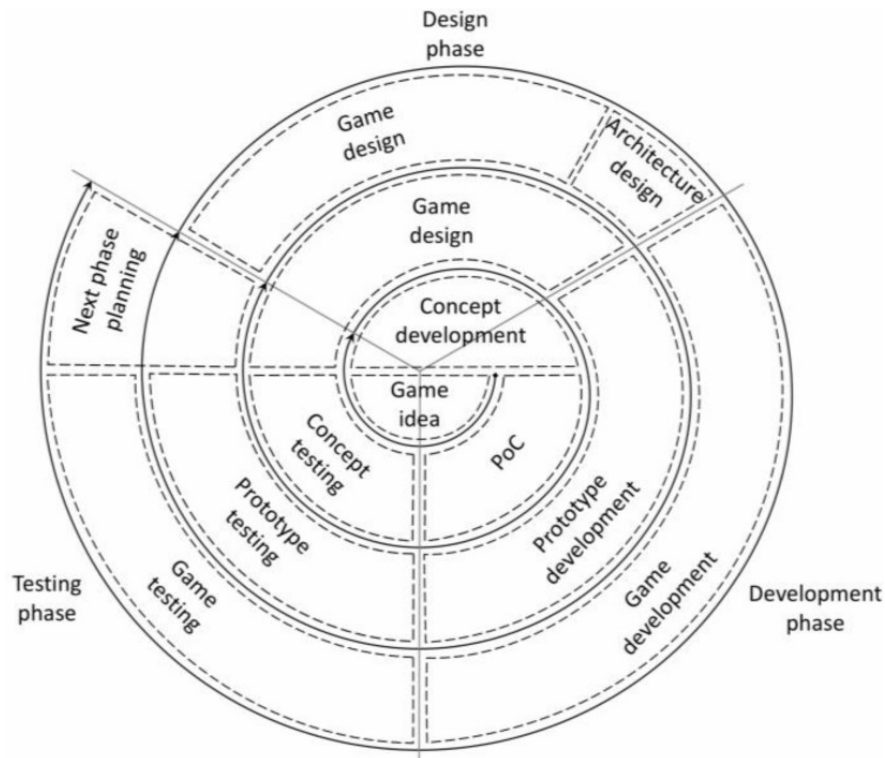
2 PELIKEHITYS

2.1 Pelikehityksen määritelmä

Tässä kirjallisuuskatsauksessa kuvatut tutkimukset kuvailivat pelikehitystä hyvin samankaltaisesti. Lähes kaikki tutkimukset peilasivat pelikehitystä ohjelmistokehitykseen jollain tavoin; pelikehitys sisältää perinteisiä ohjelmistokehityksen piirteitä, kuten ohjelmointia, dokumentointia ja testausta, mutta eroaa myös merkittävin tavoin. Eroavaisuuksia korostaessa tutkimuksissa esiintyi toistuvasti mainintoja pehmeistä osa-alueista, kuten pelisuunnittelusta, kokemuksista ja taiteellisesta puolesta. Tiettyjen samankaltaisten ominaisuuksien kuvailun lisäksi pelikehitykselle ei kirjallisuudessa esitetty yhtä selkeää ja yhtenäistä kuvausta. Yhteisen määritelmän puutteesta mainittiin myös tutkitussa kirjallisuudessa: pelikehityksen määritelmästä näyttää olevan epävarmuutta jopa pelikehittäjien keskuudessa (Berg Marklund, Engström, Hellkvist & Backlund, 2019).

Aleem, Capretz & Ahmed (2016b) määrittävät pelikehitykselle kolme päävaihetta: esituotanto (pre-production), tuotanto (production) ja jälkituotanto (post-production). Esituotantovaiheessa testataan pelin toteutettavuutta teknisten vaatimusten ja markkinoinnin kannalta. Tuotantovaiheessa pelin toteutus suunnitellaan, projektin eri osa-alueita dokumentoidaan ja peliä rakennetaan. Jälkituotannossa peliä testataan, markkinoidaan ja mainostetaan. Aleem, Capretz & Ahmed (2016a) olivat aiemmassa tutkimuksessaan määrittäneet lisäksi esituotantoa edeltäväksi vaiheeksi konseptin (concept). Kyseisessä tutkimuksessa näille neljälle vaiheelle määritettiin erilaisia toimintoja, kuten tiivistelmän kirjoittaminen, taustatutkimus, käsikirjoitus, visualisointi ja konseptitaide, kenttä- ja vuorovaikutussuunnittelu (interaction design), animointi, ohjelmointi, media-tuotanto, integrointi, testaaminen ja julkaiseminen. Lisäksi Aleem ym. (2016a) kuvaavat eri pelilajeja, joihin jokaiseen kuuluu omia vaatimuksiaan, jotka tulee ottaa huomioon esituotantovaiheessa. Tällaisia ovat esimerkiksi toiminta-, pulma-, ralli-, strategia- ja tanssipelit. Tutkijat korostavat, että pelikehitys on näistä syistä monimutkainen prosessi, joka vaatii monialaista yhteistyötä ja eri osaamisalueiden prosesseja. Valmiin pelin tuotantoon kuuluu esimerkiksi ääni- ja pelisuunnittelua, taidetta, tekoälyä, hallintajärjestelmiä ja inhimillisten tekijöiden arviointia.

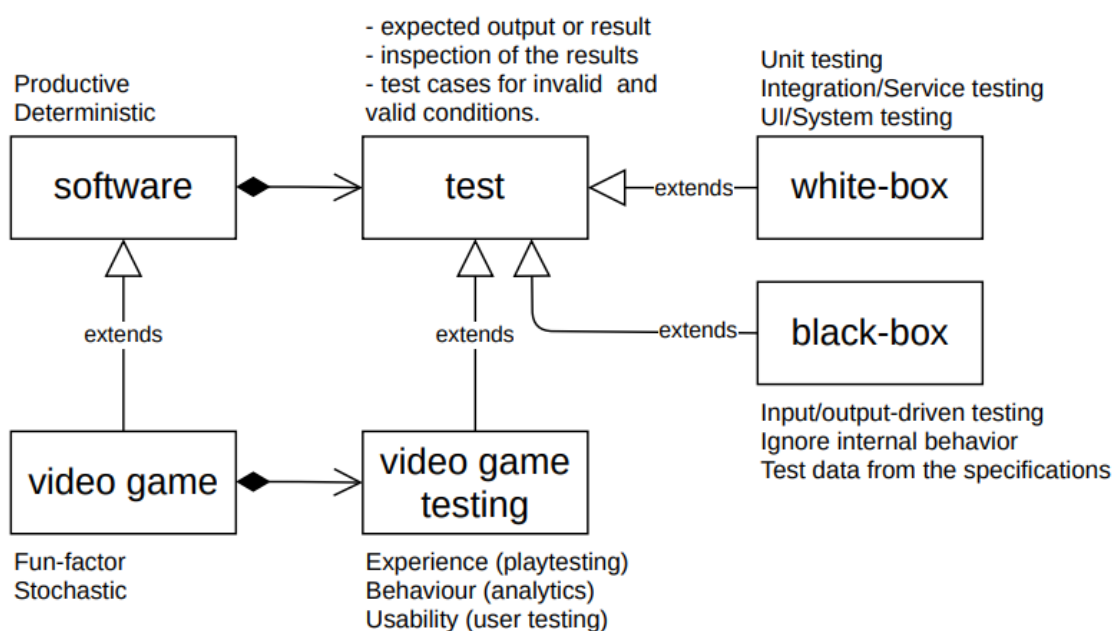
Annanpera, Yli-Kantola, Sauvola, Heinonen & Siira (2018) muodostivat pelikehitysprosesseja käsittelevän kirjallisuuden perusteella yleistetyn mallin pelikehityksen vaiheista (Kuvio 1). Malli sisältää suunnittelu-, kehitys- ja testausvaiheet. Kuvassa painotetaan kehityksen iteratiivisuutta, sekä testimenetelmien- ja tavoitteiden vaihtumista eri vaiheissa.



Kuvio 1 Pelikehityksen vaiheet (Annanpera, Yli-Kantola, Sauvola, Heinonen & Siira, 2018)

2.2 Pelikehitys suhteessa ohjelmistokehitykseen

Kirjallisuuskatsauksessa nousi esille toistuva teema: pelikehittäjät ja tutkijat pitävät pelikehitystä omana erillisenä alanaan ohjelmistokehityksestä, joka kuitenkin sisältää ohjelmistokehityksen osa-alueita. Näiden kahden alan eroilla on selkeitä vaikutuksia testaukseen. Tässä luvussa alustetaan pelitestauksen kannalta tärkeitä asioita kuvaamalla näiden kahden alan keskeisimmät erot. Esimerkiksi Politowski ym. (2022) kuvaavat testauksen luonnetta pelikehityksen ja ohjelmistokehityksen näkökulmista kuviossa 2.



Kuvio 2 Testaus peli- ja ohjelmistokehityksessä (Politowski ym., 2022)

Kirjallisuuskatsauksessa selvisi, että useimmat peliyritykset pitävät pelikehitystä luovana työnä. Kasurisen & Smolanderin (2014) mukaan monet pelikehittäjät eivät koe työskentelevänsä pelkästään ohjelmistotalalla. Tutkimus tarkasteli seitsemää pelialan yritystä haastattelemalla yhteensä 27 yritysten pelikehittäjää. Tutkimuksessa kaikki osallistuneet organisaatiot, ohjelmoijista ja testaajista lähtien kokivat pelikehityksen olevan lähempänä luovaa työtä kuin tavanomaista ohjelmistokehitystä. Tutkijat tarkensivat, että tähän löytyi vain muutamia poikkeuksia. Kasurinen & Smolander (2014) kuvailivat havaintojensa perusteella viisi peliprojektien tavoitetta:

- vaaditun toiminnallisuuden tuottaminen
- budjetissa pysyminen
- aikataulun seuraaminen
- riittävän laadun tuottaminen
- tunteisiin vetoavan tuotteen tuottaminen.

Näistä ensimmäiset neljä muistuttavat ohjelmistokehitystä, mutta viides tavoite, ”tunteisiin vetoavan tuotteen tuottaminen” eroaa ohjelmistokehityksen vaatimuksista. Kaikki tutkimuksessa haastatellut organisaatiot mainitsivat käyttäjäkokemuksen ja pelimekaniikat, ja toisin kuin ohjelmistokehityksessä, jossa painoarvo on luotettavuudessa, yhteensopivuudessa ja tehokkuudessa, pelikehityksessä painotettiin hauskuutta ja pelimekaniikkoja. Kasurisen & Smolanderin (2014) mukaan tavanomaiset ohjelmistokehitysorganisaatiot keskittyvät vaadittujen ominaisuuksien toteutukseen ja niiden tekniseen alustaan. Tällaisissa organisaatioissa on yleistä käyttää pelkästään ennakkoon suunniteltuja testitapauksia, eikä kokeiluun perustuvia testejä tavallisesti käytetä. Peliorganisaatioissa keskitytään yleensä pelattavuuteen, käytettävyyteen, pelisääntöihin ja hauskuuteen, eikä niinkään tuotteen teknisiin ominaisuuksiin. Tekniset osapuolet otetaan huomioon siinä määrin, että niistä johtuvat puutteet eivät haittaa tuotetta tai yrityksen mainetta.

Kasurisen & Smolanderin (2014) mukaan peliyrityksillä on riittävästi resursseja testauksen toteuttamiseksi, vaikka testausprosessit eivät olisikaan järjestelmällisiä. Tutkimuksessa havaittiin kuitenkin, että peliyritysten testiresurssit eivät vastanneet näiden itsemäärittelemiä optimimääriä. Keskimäärin yritysten testiresurssit olivat 70 % toivotusta, mikä vastaa ohjelmistoalan yritysten tilannetta yleisesti.

Pelikehitys ja ohjelmistokehitys eroavat myös suhtautumisessaan ohjelmistoon tehtäviin myöhäisiin muutoksiin. Peliprojekteissa muutokset ovat yleisiä ja odotettuja, kun taas ohjelmistokehityksessä myöhäisiä muutoksia pyritään minimoimaan (Kasurinen & Smolander, 2014).

3 PELITESTAUS

Tässä luvussa käsitellään pelitestauksen määritelmää, työkaluja ja menetelmiä.

3.1 Määritelmä

Hernández Bécares ym. (2017) kuvaavat testausta ohjelman analysointiprosessiksi, jossa verrataan ohjelman nykyistä tilaa määriteltyyn tavoitetilään ja arvioidaan sen ominaisuuksia. Ohjelman tavoitetilä sisältää laadun, toiminnallisuuden ja luotettavuuden vaatimukset ja tavoitteet. Kyseinen kuvaus koskee perinteistä ohjelmistotestausta, mutta pätee suurelta osin myös pelitestaukseen.

Politowski ym. (2022) mukaan pelitestaukselle ei ole olemassa yhtä selkeää määritelmää. Kyseinen tutkimus määrittääkin pelitestauksen sateenvarjoterminiksi, joka kattaa teknisen testauksen, laadunvalvonnan (Quality Assurance, QA) ja pelaajatutkimuksen (Game User Research, GUR). Pelaajatutkimus keskittyy pelien käytettävyyteen ja käyttäjäkokemukseen. Tutkimusala käsittelee kaikkia pelin osa-alueita, joiden kanssa pelaajat voivat olla vuorovaikutuksessa. Tämä kattaa muun muassa valikot, äänet, taiteen ja pelimekaniikat. Tutkimuksessa pyritään vastaamaan kysymykseen, miksi pelaaja toimii tietyllä tavalla. Pelaajatutkimuksella ja pelitestauksella on eroa. Molemmat hyödyntävät testi-istuntoja, mutta pelaajatutkimus keskittyy pelin subjektiivisiin puoliin, kun taas pelitestauksessa pyritään etsimään ohjelman virheitä (bugeja) ja arvioimaan muita teknisiä puolia.

3.2 Testaus

3.2.1 Testaamisen merkitys

Pelialalla kilpailu on kovaa ja onnistunut pelijulkaisu voi olla peliyritykselle elintärkeä ja epäonnistunut julkaisu voi kaataa yrityksen (Ahmad, Barakji, Shahada & Anabtawi, 2017). Kuten Politowski, Petrillo, Ullmann & Guéhéneuc (2021b) toteavat, pelit kärsivät usein heikosta laadusta ja peliprojektit viivästyvät, mikä viittaa mahdollisiin testausongelmiin. Kyseisessä tutkimuksessa esitetäänkin, että pelikehityksessä testausta ei suoriteta riittävästi. Aleem ym. (2016a) kuvaavat tutkimuksessaan pelitestausta hyvin tärkeäksi vaiheeksi pelikehitystä. Viivästykset testauksen aloittamisessa voivat johtaa projektien epäonnistumiseen. Pelimarkkinoiden ja kilpailijoiden määrän kasvaessa kilpailu kovenee, ja yritysten on kiinnitettävä huomiota prosesseja tehostaviin keskeisiin tekijöihin pysyäkseen kilpailukykyisinä.

Politowski, Petrillo & Guéhéneuc (2021a) kuvaavat tutkimuksessaan ongelmallisten pelijulkaisujen vaikutuksia: bugiset pelit vahingoittavat yritysten tuloista, imagoa ja asiakasuskollisuutta, kun julkaistu peli ei vastaa pelaajien odotuksia tai toiveita. Negatiiviset mielipiteet pelistä näkyvät pelien menestyksessä; bugisen pelin myynti laskee sanan kiiriessä ja jää myyntitavoitteista. Vaikutus on nähtävissä monessa julkaisussa. Melko tuore esimerkki julkisuutta saaneesta tapauksesta on *Cyberpunk 2077*, joka oli ennen joulukuun 2020 julkaisua vuoden odotetuimpia pelejä, mutta jonka tulos jäi yrityksen tavoitteesta tiedon levitessä pelin lähes pelikelvottomasta toiminnallisuudesta.

Nantes, Brown & Maire (2008) kuvaavat peliympäristöjä monimutkaisiksi interaktiivisiksi kokonaisuuksiksi, jotka vaativat laajaa analyysiä ja testausta laadun varmistamiseksi julkaisua varten. Vaadittava testauksen määrä kasvaa ohjelman koon kasvaessa. Testauksella pyritään varmistamaan pelin toimivuus, vakaus ja kestävyys, mutta myös esimerkiksi pelin hauskuus ja haastavuus. Myös Berg Marklund ym. (2019) kuvasivat tutkimuksessaan pelikehitystä samankaltaisesti: peliprojekteja on lähes mahdotonta suunnitella ennakkoon yksityiskohdallisesti, sillä tuotantovaiheessa kehittäjät ideoivat, parantelevat ja testaavat peliä jatkuvasti, jolloin uusia pehmeitä vaatimuksia nousee esille.

Kasurisen & Smolanderin (2014) mukaan pelikehittäjät suorittavat testausta lähes poikkeuksetta ja tutkimuksessa kaikki yritykset muuttivat lopputuotteitaan testauksen perusteella. Pelikehityksessä testausprosesseilla on suurempi vaikutus tuotteeseen kuin tavanomaisessa ohjelmistokehityksessä, sillä myöhäisempiä muutoksia sallitaan prosessissa ja niitä jopa odotetaan. Jopa merkittäviä pelin ominaisuuksia saatetaan muuttaa testien perusteella. Tutkimuksessa kohdeyleisön mielipiteiden perusteella sisältöä saatettiin poistaa, muokata tai suunnitella uudelleen.

Aleem ym. (2016a) tutkimuksessa esitettiin, että testauksella on tärkeä rooli jokaisessa kehitysvaiheessa, ja testauksen hallinta on tärkeää läpi kehitysprosessin. Tulokset tukivat tutkimuksen hypoteesia, että tehokas pelikehitysprosessi vaatii testaamisenhallintaa.

3.2.2 Mitä testataan?

Politowski ym. (2022) esittää tutkimuksessaan monia testikohteita sekä testaajien että kehittäjien näkökulmasta. Pelitestaajat arvioivat testisessioissa esimerkiksi pelin teknistä toimivuutta ja pelikokemusta. Toiminnallisuuden arvioinnissa pelitestaajat etsivät virheitä tai bugeja ja tarkistavat pelin normaalia toimivuutta. Pelikokemusta tarkastellessa pelitestaajat tarkastelevat muun muassa pelin hauskuutta ja puoleensavetävyyttä. Testauksessa tärkeintä on kuitenkin selvittää pelin hauskuus. Pelikehittäjät arvioivat koodin ja toiminnallisuuden lisäksi kokemusta ja käytettävyyttä. Testauksessa kehittäjät testaavat bugien lisäksi pelisisältöä ja pelimekaniikkojen tasapainotusta. He yrittävät myös paikantaa mahdollisia kaatumisen tai jumittumisen aiheuttavia kohtia, sekä visuaalisia ja suorituskykyyn liittyviä ongelmia. Pelin laatua voi arvioida myös sen grafiikoiden, äänien ja koodin pohjalta (Aleem ym., 2016a). Lisäksi voidaan testata ohjelmien

toimivuutta eri alustoilla ja niiden muutoksensietokykyä (Hernández Bécares ym., 2017).

Politowski ym. (2022) kysyivät kehittäjiltä myös heille tärkeintä testauksen aluetta. Vastaukset olivat monimuotoisia ja koskivat muun muassa kehityskohdeiden paikantamista, pelaajien odotusten täyttämistä, oikeanlaisen tuntuman hakemista, sekä pelaajien näkökulman ja kokemuksen selvittämistä. Lisäksi vastauksissa täydellinen toimivuus ja määritettyjen vaatimusten täyttäminen koettiin toissijaiseksi tai yksinään riittämättömäksi. Politowski ym. (2021b) huomasivat, että heidän tutkimissaan peliprojektien jälkeisissä analyyseissä (post-portem) oli mainittu pelitestaus, mutta ei yksikkö- tai integraatiotestausta.

Kasurisen & Smolanderin (2014) tutkimuksessa esitettiin, että peliyritykset eivät keskity samoihin testialueisiin kuin tavanomaiset ohjelmistoyritykset. Peliyritykset keskittyvät enemmän pehmeisiin testialueisiin, kuten käyttäjäkokemukseen, hauskuuteen, käytettävyyteen tai pelimekaniikkoihin teknisten ominaisuuksien, kuten vakauden, turvallisuuden tai ylläpidettävyyden sijaan. Teknisten ominaisuuksien alhaista prioriteettia selitettiin osaltaan sillä, että yritykset tapaavat ulkoistaa teknisen kehittämistyön, esimerkiksi ostamalla kolmansien osapuolien pelimoottoreita. Tutkimuksessa luokitellaan kolme pelitestauksen kohdealuetta: pelimekaniikat, tekninen toteutus ja käyttäjäkokemus. Tutkimuksessa perinteiselle ohjelmistokehitykselle tärkeät laadun ominaisuudet, kuten luotettavuus ja toiminnallisuus eivät ole pelikehityksessä yhtä tärkeitä kuin pehmeämmät ominaisuudet, kuten käyttäjäkokemus ja käytettävyys.

Kasurisen & Smolanderin (2014) kolme testauksen pääkohdetta sisältöineen on esitetty taulukossa 1.

Taulukko 1 Kolme pelitestauksen pääkohdetta

Käyttäjäkokemus	Tekninen toteutus	Pelimekaniikat
Hauskuus	Toimivuus	Pelissäännöt
Pelin vaikutelma	Vakaus	Pelin tasapaino
Palkitsevuus	Muut laatutekijät	Pelin sisältö

3.2.3 Miten testataan?

Kasurisen & Smolanderin (2014) mukaan testinhallinta on pelikehityksessä tilannekohtaisempaa (ad hoc) kuin tavanomaisessa ohjelmistokehityksessä. Tutkimuksessa haastatelluissa yrityksissä etukäteen suunniteltujen testien määrä oli 0–30 % organisaatioiden käyttämistä testitapauksista. Yritykset käyttivät vapaa-aiheisemmasta testauksesta huolimatta laadunvarmistusmenetelmiä, kuten laadunvarmistuksesta vastaavaa henkilöä tai tiimin yhteisiä päätöksiä tuotteen sisällöstä. Yleensä suurissa, niin sanotuissa AAA-studioissa yrityksen sisäiset laadunvarmistustiimit testaavat yritysten omia pelejä, kun taas pienemmät

yritykset joko ulkoistavat testauksen tai testaavat pelejään kehittäjien kesken (Politowski ym., 2022).

Aleem ym. (2016a) esittävät tutkimuksessaan, että kehittäjät tekevät testausta pääasiassa pelinkehityksen myöhäisessä vaiheessa tai vasta kehitystyön lopussa, jotta lopputuotteen laatu ja toimivuus varmistuisivat. Tavallisesti peliprojektista vastaava määrittää projektille tietyn verran aikaa laadunvarmistukselle tai testaukselle. Tutkijat kuitenkin mainitsevat, että testauksen tulisi olla osa kaikkia prosesseja ja pelin kaikki osat tulisi testata kehityksen kaikkien vaiheiden aikana. Lisäksi esituotannon aikana tulisi testata pelin perusteita, kuten kehitysympäristöjä ja alustoja. Kehittäjien tulisi siis integroida testaus osaksi tuotantovaihetta tehokkuuden parantamiseksi. Laadun varmistamiseksi kehittäjien tulisi myös miettiä keskeisimpiä testimenetelmiä tuotantovaiheessa.

Hernández Bécares ym. (2017) esittävät tutkimuksessaan erilaisia lähestymistapoja testaukselle. Näitä ovat muun muassa ohjelman toimivuutta eri alustoilla testaavat yhteensopivuustestit, muutoksensietokykyä testaavat regressio-testit, sekä manuaaliset virheitä kartoittavat alpha- ja beta-vaiheen testit.

Kasurisen & Smolanderin (2014) tutkimuksessa nousi esille muutamia yleisesti käytettyjä testimenetelmiä. Suurin osa tutkituista yrityksistä keskittyi oppimista ja helppokäyttöisyyttä testaavaan käytettävyydestestaukseen tai pelkkään ohjelman käyttöön perustuvaan kokeilulliseen testaukseen. Pieni osa tutkituista yrityksistä käytti erillisiä esirakennettuja testiympäristöjä, ja osa suoritti kattavaa ohjelmiston yksikkötestausta. Dokumentteihin perustuvat testimenetelmät tai etukäteen suunnitellut testitapaukset olivat vähemmän käytettyjä.

Aleemin ym. (2016a) mukaan pelitestauksessa on monta muutakin vaihetta kuin testitapausten määrittely, sillä suurin osa pelitestauksesta perustuu black-box -testaukseen, jossa tarkastellaan pelin ulospäin näkyvää käytöstä. Tästä syystä testimetodien hallinnalla on suuri merkitys. Esimerkiksi esituotantovaiheessa testisuunnitelmassa tulisi määrittää pelin ohjelmistovaatimukset. Kunnon toteutettu testien dokumentointi auttaa kehittäjiä korjaamaan pelissä esiintyviä virheitä nopeammin ja halvemmalla.

Albaghajati ja Ahmed (2020) keräsivät erilaisia automaattisen testauksen keinoja ja luokittelivat ne viiteen eri pelitestauksen kategoriaan niiden lähestymistavan mukaan.

Ensimmäinen kategoria kattaa hakuun perustuvat lähestymistavat, jotka keskittyvät analysoimaan pelin toiminnallista tilaa. Ne etsivät ja raportoivat mahdollisia tiloja, jotka vastaavat tai rikkovat ennalta määrättyjä kriteereitä. Mahdollisia testikohteita ovat pelin tasot, tasapaino ja pelattavuus. Malleilla voidaan myös simuloida eritasoisten pelaajien käyttäytymistä pelissä. Yksi tällainen malli on Cicero, jota käytetään apuna pelisuunnittelussa ja virheenjäljityksessä (Machado, Gopstein, Nealen, Nov & Togelius, 2018). Machadonin ym. (2018) tutkimuksessa tekoäly paransi ihmisten suorittaman testauksen tarkkuutta. Toisaalta menetelmän kerrottiin olevan vielä kaukana tavoitellusta alustariippumattomuudesta. Toisin sanoen menetelmä soveltui vain rajalliseen määrään testiympäristöjä. Changin, Aytemizin, & Smithin (2019) esittämä Reveal-More RRT-pohjainen menetelmä lukeutuu myös tähän kategoriaan. Reveal-More menetelmässä

testiagentti testaa peliympäristöjä pelaajilta tallennettujen syötteiden avulla. Menetelmän heikkoudeksi koettiin agentin päämäärättömyys ja heikko skaalautuvuus testiympäristön koon tai monimutkaisuuden kasvaessa.

Toiseen kategoriaan kuuluvat tavoitepohjaiset lähestymistavat, jotka pyrkivät asettamaan automaattisille testiagenteille toimintatapoja, palkintoja ja rangaistuksia, jotka ohjaavat agenttia tutkimaan haluttuihin tavoitteisiin johtavia mahdollisia reittejä. Bergdahl, Gordillo, Tollmar, & Gisslén (2020) käyttivät menetelmässään syvää vahvistusoppimista. Käytetyn agentin tavoite oli testata pelimekaniikkoja ja löytää ongelmia itsenäisesti tilanteissa, joissa pelkkä ihmisten suorittama testaus olisi epäkäytännöllistä tai kallista. Testimenetelmän heikkous oli monia automaattisia testimenetelmiä koskeva heikko yleistettävyyys. Menetelmän kuvattiin soveltuvan parhaiten tukemaan muita testimenetelmiä. Jaffe, Miller, Andersen, Liu, Karlin, & Popovic (2012) esittivät ”rajoitetun pelaamisen” (Restricted-play) sovelluskehityksen, joka kuuluu myös tähän kategoriaan. Menetelmän tavoitteena on seurata ja mitata tiettyjä pelin ominaisuuksia ja tarjota kehittäjille palautetta esimerkiksi sopivasta haasteellisuudesta mitattujen arvojen perusteella. Restricted-play ei kuitenkaan pysty ennustamaan pelaajakokemusta, ja sen käyttö voi olla vaikeaa laajoissa kokonaisuuksissa.

Kolmanteen kategoriaan kuuluvat ihmistä matkivat lähestymistavat, jotka pyrkivät matkimaan ihmisen käyttäytymistä. Tällaiset lähestymistavat ovat hyödyllisiä testattaessa esimerkiksi tunteita, uteliaisuutta, haasteita, vaikeutta ja aggressiivisuutta. Ariyurek, Betin-Can, & Surer (2019) käsittelivät tutkimuksessaan Monte Carlo -puuhakua ja vahvistusoppimista (reinforcement learning) hyödyntävää testiagenttia. Menetelmän tavoitteena on löytää mahdollisia pelin rikkovia tiloja. Käytetyt algoritmit lisäsivät agenttien ihmismäisyyttä, sekä tehostivat virheiden löytämistä. Testimenetelmän haasteeksi kuvattiin heikko yleistettävyyys ja hidas käyttöönotto. Stahlke, Nova, & Mirza-Babaei (2019) tutkivat asiantuntija-järjestelmiä ja tekoälyä hyödyntäviä agenteja. Tällaisen agentin tavoitteena oli suorittaa yksinkertaista kenttäsuunnittelun testausta. Tässäkin haasteena oli yleistettävyyys pelien välillä. Lisäksi käytetty alusta ei pohjautunut pelaajien tuottamaan ”asiantuntijan” dataan. Tämä auttoi mallin läpinäkyvyydessä, mutta tutkijoiden mukaan pelaajien dataan perustuva koneoppimismalli voisi toisaalta soveltua paremmin mallintamaan pelaajien käyttäytymistä.

Neljänteen kategoriaan kuuluvat tilanteeseen perustuvat lähestymistavat. Ne käyttävät testejä, jotka perustuvat ennalta määrättyihin toimintasarjoihin tai pyydettyihin toimintoihin. Tuovenen, Oussalah & Kostakos (2019) esittivät pääasiassa mobiilipelien testaukseen suunnatun MAuton, joka käyttää kuvantunnistusteknologiaa etsimään grafiikan virheitä. Testien luonti ei vaadi koodaamista ja soveltuu myös muihin sovellustyyppeihin kuin peleihin. Menetelmän heikkoudeksi mainittiin kyvyttömyys havaita kaikkia virheitä, sekä testien hitaus. Mozgovoy & Pyshkin (2017) käsittelivät tutkimuksessaan osittain automatisoitua savutestausta. Menetelmällä pyritään tarkistamaan ohjelman perustoimivuutta. Testi soveltuu heikosti graafisten elementtien testaukseen sellaisenaan, mutta kuvantunnistusteknologioiden avulla testi pystyy vastaamaan näihinkin tarpeisiin.

Viidentenä kategoriana ovat malliin perustuvat lähestymistavat, jotka esittävät tapahtumien kulkua formaalein mallein. Malleilla pyritään varmentamaan muun muassa tapahtumia, dataa tai logiikkaa. Iftikhar, Iqbal, Khan & Mahmood (2015) esittivät tutkimuksessaan pelaajan syötteitä simuloivan UML-mallin, jonka avulla pystytään mallintamaan pelin rakennetta testausta varten. Mallin avulla voidaan luoda testitapaus ja -tavoite, sekä suorittaa testi. Menetelmä vaatii toisaalta ohjelmistokehittäjän osaamista, eikä sovellu yleiseen käyttöön.

Esitettyjen kategorioiden lisäksi Albaghajati ja Ahmed (2020) mainitsivat tutkimuksessaan useita algoritmeja, jotka on kuvattu taulukossa 2.

Taulukko 2 Automaattisen testauksen kategorioita ja menetelmiä

Hakuun perustuva (search-based)	Tavoitepohjaiset (goal-directed)	Ihmistä matkiva (human-like)	Tilanteeseen perustuva (scenario-based)	Malliin perustuva (model-based)
Evoluutioalgoritmi (Evolutionary algorithm)	Vahvistusoppiminen (Reinforcement learning)	Koneoppiminen (Machine learning)	Nauhoita ja toista (Record and Replay)	Petri-verkko (Petri nets)
Kaaviohaku (Graph search)	Rajoitettu heuristiikka (Restricted heuristics)	Rajoitettu heuristiikka (Restricted heuristics)	Pelisisimulaatio (Game simulation)	UML-mallinnus (Unified Modeling Language)
RRT puuhaku (Rapidly exploring random tree search)		Sekoitetut algoritmit (Mixed algorithms)	Konenäköön perustuva virheenetsintä (Visual debugging)	GDL-kieli (Game Description Language)
				Lineaarinen ajallinen logiikka (Linear Temporal Logic)
				ModelMMORPG-kieli

Politowskin ym. (2022) tutkimuksen kirjallisuuskatsauksen osiossa 80 tutkijasta ehdotti erilaisia käytännön automaattisia testimenetelmiä. Menetelmät olivat joko täysin tai osittain automatisoituja. Niistä 53 hyödynsi koneoppimis-malleja testiagenttien opettamiseen ja pelien testaamiseen. Politowski ym. (2022) tutkivat lisäksi, kuinka hyvin kirjallisuudesta löytyvät automaattiset testaustavat sopivat pelikehittäjien tarpeisiin. Tutkimuksen haastatteluosioon osallistui 12 hyväksyttyä vastaajaa. Manuaalinen testaus oli tavanomaisempaa; kaikki vastanneet käyttivät manuaalista pelitestausta säännöllisesti. Osa vastanneista testasi pelejään pelimoottoreissa. Yksikään vastanneista ei käyttänyt testikoodeja (script) pelitestauksessaan. Tutkijat ehdottivat erilaisia kirjallisuudessa mainittuja automaattisia testausmenetelmiä. Hyväksytyimmät testausmenetelmät olivat helppoja toteuttaa ja nopeuttivat testaamista. Kuitenkin yleisesti ottaen vastaajat suhtautuivat menetelmiin epäilevästi. Taulukossa 3 on koottu Politowskin ym. (2022) esimerkkejä tutkimuksessa ehdotetuista menetelmistä. Esitetystä

menetelmistä näkee, että automaattinen testaus on tavoiteltavaa, mutta vaikeasti toteutettavissa.

Taulukko 3 Testimenetelmiä ja kehittäjien suhtautuminen

Menetelmä	Tavoite	Kehittäjien suhtautuminen
Ihmistä matkiva syväoppiva testaus	Kentän vaikeusasteen automaattinen ennustaminen.	Haluttu, mutta vaatii liikaa aikaa ja dataa, sekä testikokonaisuuden rakentamisen.
Vaikeusasteen ja pelaajakadon ennustaminen ilman pelaajia	Kenttien onnistuneiden läpäisymäärien ennustaminen pelitestaa-jilta ja automatisoiduilta testiagenteilta saadulla datalla.	Varauksellinen, koska menetelmällä ei pysty sanomaan tarkasti keskeytysten syytä ja kohtaa.
Pelitestauksen kattavuuden nostaminen koneoppimisella	Monimutkaisten 3D-ympäristöjen kattava testaus pelimoottorissa.	Halutuvin, sillä testattavaa asiaa on hidas testata manuaalisesti.
Pelitestaus eriluonteisilla koneoppimista hyödyntävillä testiagenteilla	Erilaisten reittien löytäminen kentässä.	Vaihteleva, vastaajat epäilivät metodin käytettävyyttä.
Taistelupelien automaattinen testaus koneoppimisella	Rajatapausten ja pelin tilanteiden automaattinen testaus.	Vaihteleva, vastaajat epäilivät metodin käytettävyyttä.
Valvomaton haasteiden ratkaisu	Pelien nopea kaatumisten ja jumittumisten paikannus	Haluttu, menetelmä on ehdotuksista käytännöllisin ja realistisesti toteutettavissa
Muokattu hakupuu ja yleinen peliteko-äly pelitestauksessa	Pelien testaus ja bugien löytäminen toistettavilla toimintasarjoilla	Varauksellinen, sillä testaamiseen käytettiin tunnettuja vikoja. Kykyä löytää uusia vikoja epäiltiin.

4 PELITESTAUKSEN HAASTEET

Automatisoidun testauksen toteuttaminen pelikehityksessä on vaikeaa ja usein epäkäytännöllistä, sillä pelikehitykselle on ominaista jatkuva tavoitteiden ja suunnitelmien muuttuminen (Murphy-Hill, Zimmerman & Nagappan, 2014). Lisäksi pelijärjestelmät ovat monimutkaisia, monitasoisia ja muuttuvia ympäristöjä, minkä vuoksi bugien löytäminen ja korjaaminen on vaikeaa Albaghajati & Ahmed (2022).

Kasurinen & Smolander (2014) esittivät, että testausprosessissa voi esiintyä vaikeuksia monesta syystä, kuten puute organisaation johdon tuesta, sidosryhmien heikko osallistaminen tai esimerkiksi ajan, rahan tai muiden resurssien puute.

Politowski ym. (2021a) määrittivät seitsemän erilaista automaattisen testauksen haastetta:

1. Koodin ja muiden elementtien linkittyminen (coupling)
2. Pelin suuruus
3. Syötteiden ja seurausten satunnaisuus
4. Muutokset pelin suunnassa
5. Testi-insinöörien kalleus (halvempaa palkata pelitestaajia)
6. Rajallinen aika kuluu sisällön luontiin
7. Hauskuuden mittaus automaattisesti

Hernández Bécares ym. (2017) mukaan pelitestausta vaikeuttavat entuudestaan peleihin kuuluvat muut resurssit, kuten 3D-mallit, tekstuurit, musiikki jne., sekä pelisisältö, kuten tasot, tehtävät ja haasteet. Näiden välillä on keskinäisriippuvuuksia, mikä saattaa luoda uusia virheitä muutosten tapahtuessa. Lisäksi testitilanteissa virheeseen johtaneita vaiheita on usein vaikea toistaa (Politowski ym., 2022). Automaattisen testauksen haasteet näkyivät myös Politowskin ym. (2022) tutkimuksessa haastateltujen kehittäjien mielipiteissä: kehittäjät uskoivat manuaalisen testauksen pysyvän käytössä 10 vuodenkin päästä ja painoarvo tulee olemaan testityökalujen kehitystä enemmän pelikehityksessä. Myös puutteelliset testiprosessit ja tekninen osaaminen koettiin haastavana.

Politowskin ym. (2022) mukaan viime vuosina julkaistujen tutkimusten ehdottamat koneoppimisen ratkaisut keskittyivät testauksen ongelmien ratkaisun sijaan koneoppimismalleihin. Näiden soveltuvuutta käytännön testityöhön kyseenalaistettiin:

- Käytetyt työkalut ovat liian yksinkertaisia, keskeneräisiä tai akateemisia.
- Testitavoitteita ei määritellä selkeästi.
- Testitavoitteita ei ole tai niitä ei ole määritelty ajallaan.
- Lähdekoodia ei ole usein saatavilla, minkä vuoksi ehdotettua ratkaisua ei voi toisintaa.

Politowski ym. (2022) esittivät tutkimuksessaan selkeän tarpeen yleispäteville avoimen lähdekoodin työkaluille, joita pelikehittäjät voisivat hyödyntää. Ehdotettujen työkalujen tulisi soveltua monenlaisille peleille ilman pelikohtaista muokkaamista ja toimia suoraan erilaisten pelimoottoreiden kanssa. Tutkimuksessa ei uskottu koneoppimisen korvaavan pelitestaajia lähiaikoina, sillä ihmisiä tarvitaan vielä havaitsemaan poikkeamia, pelien outoa käytöstä ja testaamaan niiden hauskuutta.

Alla on esitetty kirjallisuuskatsauksessa löytyneet testimenetelmät, sekä niihin liittyviä haasteita (Taulukko 4).

Taulukko 4 Pelitestauksen menetelmät ja haasteet

Testimenetelmä	Tavoitteet	Haasteet	Lähde
Kaikki	Laadun takaaminen	Puutteellinen johdon tuki tai sidosryhmien osallistaminen, ajan, rahan tai muiden resursien puute.	Kasurinen & Smolander (2014)
Dokumentaatio	Helpottaa projektin ja testien hallintaa	Jatkuvasti muuttuva sisältö vaatii jatkuvaa dokumenttien ylläpitämistä.	Kasurinen & Smolander (2014)
Automaattiset testimenetelmät	Koodin ja pelien toimivuuden testaus	Jatkuvasti muuttuva sisältö, sekä monimutkaiset ja -tasoiset järjestelmät vaativat jatkuvaa testien muokkaamista.	Murphy-Hill ym. (2014)
Yksikkötestaus	Testaa ohjelman yksittäisiä osia	Ei selkeää syytä vähäiselle käytölle.	Politowski ym. (2021b)
Integraatiotestaus	Testaa yksittäisten osien yhteistoimintaa	Ei selkeää syytä vähäiselle käytölle.	Politowski ym. (2021b)
Koneoppimista hyödyntävät menetelmät	Pitkäkestoisten manuaalisten testien korvaaminen, testaamisen analysointi	Nykyiset menetelmät eivät vastaa kehittäjien tarpeita, vaan ovat lähinnä tutkimusvaiheen ratkaisuja. Halutut ratkaisut voivat olla myös liian hankalia ottaa käyttöön.	Politowski ym. (2022)
Hakuun perustuva	Ennalta määritettyjen tilojen (states) testaus	Tietyt hakuun perustuvat menetelmät voivat rajoittua liikaa jo tunnettuihin virheisiin.	Albaghajati ja Ahmed (2020), Politowski ym. (2022)
Cicero (tekoälyavusteinen kaaviohaku)	Pelisuunnittelu ja virheenjäljitys,	Menetelmä rajoittuu tiettyihin kehitysympäristöihin ja käytetyt simulaatiot voivat olla hitaita.	Machado ym. (2018)

	riippumattomuus ympäristöstä		
Reveal-More (RRT-pohjainen pelaajan tallennetun toiminnan ja automaattisen haun yhdistelmä)	Peliympäristöjen testaus	Testimenetelmä ei osaa keskittyä testin kannalta keskeisiin toimintoihin ja skaalautuu heikosti kenttien koon tai monimutkaisuuden kasvaessa.	Chang ym. (2019)
Tavoitepohjainen	Pelin kenttien, tehtävien ja tavoitteiden monipuolinen testaus	Menetelmälle ei löytynyt selkeitä haasteita.	Albaghajati ja Ahmed (2020)
Syvä vahvistusoppiminen (Deep Reinforcement Learning)	Itsenäinen pelimekaniikkojen kokeilu ja käyttö tilanteissa, jotka ovat ihmiselle vaikeita	Testejä on vaikea yleistää ja soveltuu parhaiten tukemaan muita testimenetelmiä.	Bergdahl ym. (2020)
Restricted-play -sovelluskehys	Pelitasapainon mittaus	Soveltuu heikosti pelaajakokemuksen testaamiseen ja voi olla vaikea toteuttaa monimutkaisissa peleissä.	Jaffe ym. (2012)
Ihmistä matkiva	Työläiden testivaiheiden helpottaminen ja esimerkiksi sopivan vaikeusasteen automaattinen ennustaminen.	Vaatii liikaa aikaa, dataa ja ponnostusta. Lisäksi hienojakoisemat ominaisuudet tarvitsevat ihmisen luonnollista havainnointikykyä.	Albaghajati ja Ahmed (2020), Polittowski ym. (2022)
Monte Carlo -puuhakua ja vahvistusoppimista hyödyntävä agentti	Pelin rikkovien toimintojen löytäminen	Mahdollisesti heikko soveltuvuus vaihtuviin testitilanteisiin ja käytetyt testit voi olla hidasta ottaa käyttöön.	Ariyurek ym. (2019)
Asiantuntijajärjestelmiä (expert system) ja tekoälyä hyödyntävä agentti	Kenttäsuunnittelun yksinkertainen testaaminen	Yleistettävyyys eri pelien välillä voi olla haasteellista ja agenttien kouluttaminen vaatii nykyistä enemmän pelaajien tuottamaa dataa.	Stahlke ym. (2019)
Tilanteeseen perustuva	Ohjelman ja sisällön testaus tallennettujen syötteiden perusteella	Tietyt hakuun perustuvat menetelmät voivat rajoittua liikaa jo tunnettuihin virheisiin. Soveltuvuus testaukseen voi joissain tapauksissa olla heikko.	Albaghajati ja Ahmed (2020)
MAuto	Grafiikan virheidten testaus	Menetelmä ei välttämättä havaitse kaikkia virheitä ja testit voivat kestää pitkään.	Tuovenen ym. (2019)

Puoliautomaattinen savutestaus (semi-automated smoke test)	Ohjelman perustoimivuuden tarkistus	Perusversio metodista soveltuu heikosti pelin visuaalisiin osiin. Esimerkiksi käyttöliittymän testaus vaatii kuvantunnistuksen soveltamista.	Mozgovoy & Pyshkin (2017)
Malliin perustuva	Ohjelman ja sisällön testaus kaavioita ja formaaleita esityksiä seuraten	Soveltuvuus testaukseen voi joissain tapauksissa olla heikko.	Albaghajati ja Ahmed (2020), Politowski ym. (2022)
Pelaajan syötteitä simuloiva UML malli	Testitapauksen ja -tavoitteen luonti, testin suoritus	Menetelmässä käytetyt mallit vaativat asiantuntijan osaamista, eivätkä sovellu yleiseen käyttöön.	Iftikhar ym. (2015)
Manuaaliset testimenetelmät	Sisällön toimivuuden ja hauskuuden mittaus	Hitaus ja ihmisistä riippuva testaus tekevät manuaalisista menetelmistä heikosti skaalautuvan.	Politowski ym. (2022), Politowski ym. (2021b)
Kokeilullinen testaus	Sisällön toimivuuden ja hauskuuden mittaus	Käytetään paljon, mutta hitaus ja ihmisistä riippuva testaus tekevät manuaalisista menetelmistä heikosti skaalautuvan.	Kasurinen & Smolander (2014), Politowski ym. (2022), Politowski ym. (2021b)
Käytettävyydestestaus	Pelin ominaisuuksien ja toiminnallisuuden ohjaaminen pelaajille soveltuviksi.	Hidasta, suoritetaan usein ad hoc tyylillä	Kasurinen & Smolander (2014)

5 YHTEENVETO JA JATKOA

Kirjallisuuskatsauksessa pyrittiin vastaamaan tutkimuskysymykseen ”*Mitä haasteita pelitestauksen menetelmiin liittyy?*”. Tutkimuksessa kerättiin tieteellisessä kirjallisuudessa käsiteltyjä testimenetelmiä ja niihin liittyviä haasteita.

Tutkimus käsitteli aineistoa yleisellä tasolla, eikä testimenetelmiin syvennytty tarkemmin. Tästä johtuen tutkittujen menetelmien yksityiskohtaisemmat ominaisuudet ja niiden merkitys testaukselle ja kehitykselle yleensä jäivät vähäisiksi. Tutkimuksessa käytettiin myös laajasti sanaa ”hauskuus” kuvaamaan pelien puoleensavetävyyttä. Tässä tulee kiinnittää huomiota, että pelit eivät aina pyri olemaan hauskoja, vaan myös toisenlaiset tunnereaktiot voivat olla pelin kannalta tärkeitä. Sopiva rinnastus voisi olla teatterien komedia ja tragedia; monenlaiset tunnereaktiot voidaan kokea viihteessä arvokkaana.

Kirjallisuuskatsauksessa selvisi, että pelikehitys erotetaan tavanomaisesta ohjelmistokehityksestä, mutta sen piirteitä löytyy pelikehityksestä kuitenkin laajasti. Testaukselle määrätty resurssit ovat molemmilla aloilla laajalti samantasoisia suhteessa yritysten tarpeisiin, mutta testimenetelmissä ja niiden käytössä on suuria eroja. Esimerkiksi peliyritykset pitävät manuaalista pelitestausta käytännöllisimpänä tapana testata pelin sisältöä, ja testien perusteella pelien ominaisuuksia saatetaan muuttaa myöhemmässä vaiheessa kuin tavanomaisissa ohjelmistoissa.

Tutkimuksessa selvisi myös, että peliprojekteissa testaaminen painottuu enemmän pelin nautinnollisuuden kuin teknisen laadun varmistamiseen. Nautinnollisuuden tai hauskuuden takaamiseksi pelitestauksella pyritään testaamaan sisällön ja mekaniikkojen tasapainoa ja pelattavuutta.

Yrityksen näyttävät suosivan manuaalista pelitestausta testiautomaation sijaan, ja olivat epäileväisiä monien ohjelmistokehityksessä käytettyjen ja tutkimuksissa ehdotettujen automaattisten testimenetelmien suhteen. Esimerkiksi Politowskin ym. (2021b) tutkimissa peliprojektien jälkeisissä analyyseissä (post-portem) pelitestauksen termiä oli käytetty, mutta yksikkö- tai integraatiotestausta ei ollut mainittu. Kuten Aleem ym. (2016a) kertovat, testimenetelmät ovat kehittyneet, mutta niissä on vielä parannettavaa. Aikaisemmissa tutkimuksissa ehdotetut testimenetelmät osoittivat paikoitellen huomattavia puutteita. Politowski ym. (2022) esittivätkin, että pelitestauksen tutkimuksessa testauksen tavoitteita ei ole mietitty riittävästi.

Politowski ym. (2022) havaitsi tutkimuksessaan automaattisen testauksen nousseen tutkijoiden suosiossa viime vuosina. Näissä ehdotetut testityökalut keskittyivät kuitenkin enemmän koneoppimisen malleihin, kuin testien tavoitteisiin. Kyselyn mukaan pelikehittäjät suhtautuvat automaattisiin testiagentteihin varauksella.

Peliyritysten käyttämistä testausmenetelmistä ja työkaluista löytyi aikaisempaa tutkimusta, mutta aiheesta riittää vielä runsaasti tutkittavaa, varsinkin käytännön työhön soveltuvan testiautomaation osalta. Tekoälyn mahdollistamien testimenetelmien kartoitus tarjoaisi esimerkiksi laajan työmaan.

Jatkotutkimuksessa voisi siis seurata yrityksiä, jotka hyödyntävät edistyneitä testausmenetelmiä ja teknologioita. Kirjallisuuskatsauksessa ilmeni myös kehittäjien toiveita automaattiseen testaukseen liittyen. Politowskin ym. (2022) haastattelemat osallistujat toivoivat helposti ylläpidettäviä automaattisia testejä, jotka ovat erillisiä testattavasta pelistä.

LÄHTEET

Ahmad, N. B., Barakji, S. A. R., Shahada, T. M. A. & Anabtawi, Z. A. (2017). How to launch a successful video game: A framework. *Entertainment computing*, 23, 1-11. <https://doi.org/10.1016/j.entcom.2017.08.001>

Albaghajati, A. M. & Ahmed, M. A. K. (2020). Video Game Automated Testing Approaches: An Assessment Framework. *IEEE transactions on games*, 1. <https://doi.org/10.1109/TG.2020.3032796>

Albaghajati, A. & Ahmed, M. (2022). A co-evolutionary genetic algorithms approach to detect video game bugs. *The Journal of systems and software*, 188, <https://doi.org/10.1016/j.jss.2022.111261>

Aleem, S., Capretz, L. F. & Ahmed, F. (2016). Critical Success Factors to Improve the Game Development Process from a Developer's Perspective. *Journal of computer science and technology*, 31(5), 925-950. <https://doi.org/10.1007/s11390-016-1673-z>

Aleem, S., Capretz, L. F. & Ahmed, F. (2016). Game development software engineering process life cycle: A systematic review. *Journal of software engineering research and development*, 4(1), 1. <https://doi.org/10.1186/s40411-016-0032-7>

Annanpera, E., Yli-Kantola, J., Sauvola, T., Heinonen, S. & Siira, E. (2018). *Testing methods for mobile game development a case study on user feedback in different development phases*. <https://doi.org/10.1109/SeGAH.2018.8401372>

Ariyurek, S., Betin-Can, A., & Surer, E. (2019). Automated video game testing using synthetic and humanlike agents. *IEEE Transactions on Games*, 13(1), 50-67.

Berg Marklund, B., Engström, H., Hellkvist, M. & Backlund, P. (2019). What Empirically Based Research Tells Us About Game Development. *The computer games journal.*, 8(3-4), 179-198. <https://doi.org/10.1007/s40869-019-00085-1>

Bergdahl, J., Gordillo, C., Tollmar, K., & Gisslén, L. (2020, August). Augmenting automated game testing with deep reinforcement learning. In *2020 IEEE Conference on Games (CoG)* (pp. 600-603). IEEE.

Chang, K., Aytemiz, B., & Smith, A. M. (2019, August). Reveal-more: Amplifying human effort in quality assurance testing using automated exploration. In *2019 IEEE Conference on Games (CoG)* (pp. 1-8). IEEE.

Hernández Bécares, J., Costero Valero, L. & Gómez Martín, P. P. (2017). An approach to automated videogame beta testing. *Entertainment computing*, 18, 79-92. <https://doi.org/10.1016/j.entcom.2016.08.002>

Iftikhar, S., Iqbal, M. Z., Khan, M. U., & Mahmood, W. (2015, September). An automated model based testing approach for platform games. In *2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS)* (pp. 426-435). IEEE.

Jaffe, A., Miller, A., Andersen, E., Liu, Y. E., Karlin, A., & Popovic, Z. (2012, October). Evaluating competitive game balance with restricted play. In *Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Kasurinen, J., Palacin-Silva, M. & Vanhala, E. (2017). *What Concerns Game Developers? A Study on Game Development Processes, Sustainability and Metrics*. <https://doi.org/10.1109/WETSoM.2017.3>

Kasurinen, J. & Smolander, K. (2014). *What do game developers test in their products?* <https://doi.org/10.1145/2652524.2652525>

Lewis, C., Whitehead, J. & Wardrip-Fruin, N. (2010). *What went wrong: A taxonomy of video game bugs*. <https://doi.org/10.1145/1822348.1822363>

Machado, T., Gopstein, D., Nealen, A., Nov, O., & Togelius, J. (2018, July). *Ai-assisted game debugging with cicero*. In *2018 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1-8). IEEE.

Mozgovoy, M., & Pyshkin, E. (2017, March). *Unity application testing automation with appium and image recognition*. In *International Conference on Tools and Methods for Program Analysis* (pp. 139-150). Springer, Cham.

Murphy-Hill, E., Zimmermann, T. & Nagappan, N. (2014). *Cowboys, ankle sprains, and keepers of quality: How is video game development different from software development?* <https://doi.org/10.1145/2568225.2568226>

Nantes, A., Brown, R., & Maire, F. (2008). *A framework for the semi-automatic testing of video games*. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (Vol. 4, No. 1, pp. 197-202).

New Product Development Glossary. (2022). <https://www.npd-solutions.com/glossary.html>

Nwana, H. (1996). *Software agents: An overview*. *The Knowledge Engineering Review*, 11(3), 205-244. doi:10.1017/S026988890000789X

Politowski, C., Petrillo, F., & Guéhéneuc, Y. G. (2021). *A Survey of Video Game Testing*. In *2021 IEEE/ACM International Conference on Automation of Software Test (AST)* (pp. 90-99). IEEE.

Politowski, C., Petrillo, F., Ullmann, G. C. & Guéhéneuc, Y. (2021). *Game industry problems: An extensive analysis of the gray literature*. *Information and software technology*, 134, 106538. <https://doi.org/10.1016/j.infsof.2021.106538>

Politowski, C., Guéhéneuc, Y. G., & Petrillo, F. (2022). *Towards Automated Video Game Testing: Still a Long Way to Go*. *arXiv preprint arXiv:2202.12777*.

Rivkin, C. (2022). *2021 Theme Report*. <https://www.motionpictures.org/wp-content/uploads/2022/03/MPA-2021-THEME-Report-FINAL.pdf>

Schmalz, M., Finn, A., & Taylor, H. (2014, January). *Risk management in video game development projects*. In *2014 47th Hawaii International Conference on System Sciences* (pp. 4325-4334). IEEE.

Stahlke, S., Nova, A., & Mirza-Babaei, P. (2019, May). *Artificial playfulness: A tool for automated agent-based playtesting*. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (pp. 1-6).

suomisanakirja.fi (2022)

Tuovenen, J., Oussalah, M., & Kostakos, P. (2019). *MAuto: Automatic mobile game testing tool using image-matching based approach*. *The Computer Games Journal*, 8(3), 215-239.

Wijman, T. (2021). The Games Market and Beyond in 2021: The Year in Numbers. *Newzoo*. <https://newzoo.com/insights/articles/the-games-market-in-2021-the-year-in-numbers-esports-cloud-gaming/>