

KARELIA-AMMATTIKORKEAKOULU  
Tietotekniikan koulutusohjelma

Aino Markkinen

PELIOHJAIN KOSKETUSNÄYTTÖLAITTEELLE

Opinnäytetyö  
Toukokuu 2014



**OPINNÄYTETYÖ**  
**Toukokuu 2014**  
**Tietotekniikan koulutusohjelma**

Karjalankatu 3  
80200 JOENSUU  
+358 50 260 6800

**Tekijä**  
Aino Markkinen

**Nimeke**  
Peliohjain kosketusnäyttölaitteelle

**Toimeksiantaja**  
Polarbunny Oy

**Tiivistelmä**

Tässä opinnäytetyössä käsitellään peliohjaimen suunnittelua, kehitystä ja testaamista kosketusnäyttölaitteelle. Opinnäytetyön toimeksiantaja on joensuulainen pelikehitysyritys Polarbunny Oy. Opinnäytetyön tarkoituksena oli toteuttaa toimiva peliohjain eräseen yrityksen peleistä.

Teoriaosuudessa on tutkittu käytettävyyden psykologiaa toiminnan, havaitsemisen ja ongelmanratkaisun osalta. Aihealueet tukevat peliohjaimen käytettävyyden suunnittelua. Lisäksi teoriaosuudessa on tutkittu kosketusnäyttölaitteita kehitysympäristönä ja millaisia erityispiirteitä kosketusnäyttöillä on käytettävyyden kannalta. Teoriapohjan aiheita tarkastellaan pelisuunnittelun näkökulmasta.

Peliohjain on toteutettu käyttäen Unitya kehitysympäristönä ja C#:a ohjelmointikielenä. Opinnäytetyön toteutus koostuu peliohjaimen suunnitteluprosessista sekä ohjelmistokehityksen ja käytettävyydestäuksen iteroinnista käyttäjakeskeisen suunnittelun periaatteiden mukaisesti.

Opinnäytetyön tuloksena syntyi toimiva peliohjain kosketusnäyttölle, joka jää yrityksen käyttöön. Opinnäytetyössä esitellään myös jatkokehitysmahdollisuuksia ja käytettävyydestien perusteella rakennettuja parannusehdotuksia.

**Kieli**  
suomi

Sivuja 51  
Liitteet 3  
Liitesivumäärä 3

**Asiasanat**  
pelisuunnittelu, kosketusnäyttö, käytettävyys, käyttöliittymät



**THESIS**  
**May 2014**  
**Degree Programme in**  
**Information Technology**  
Karjalankatu 3  
FI 80200 JOENSUU  
FINLAND  
+358 50 260 6800

Author  
Aino Markkinen

Title  
Game Controller for Touchscreen Device

Commissioned by  
Polarbunny Ltd

Abstract

This thesis describes the design, development and testing of a game controller for touchscreen devices. The thesis was commissioned by Polarbunny Ltd, a game development company in Joensuu, Finland. The purpose of the thesis was to develop a functioning game controller for one of the company's games.

The thesis presents an overview of the psychology of usability, concentrating on the subjects of action, observation and problem solving. These subjects support the design of game controllers with high usability. The thesis examines touchscreen devices as a target platform and the characteristics they have from the usability point of view. The theoretical part also considers usability from a game design perspective.

The game controller was implemented using Unity and C#. The development of the controller consisted of three phases: planning, software development, and usability testing. These phases were used iteratively throughout the process following the principles of user-centered design. The result of the thesis is a functional game controller for touchscreen devices. The thesis also presents ideas for further development and improvements, based on the usability test results.

Language

Finnish

Pages 51

Appendices 3

Pages of Appendices 3

Keywords

game design, touchscreen, usability, user interfaces

## Sisälllys

1	Johdanto.....	5
2	Käytettävyys.....	6
2.1	Käytettävyuden psykologia.....	7
2.1.1	Käyttäjän tavoitteiden ymmärtäminen suunnittelussa.....	7
2.1.2	Havaitseminen ja tulkinta.....	8
2.1.3	Ongelmanratkaisu ja toiminnallinen kiintyminen.....	10
2.2	Käyttäjakeskeinen suunnittelu.....	11
2.3	Käytettävyystestit.....	13
3	Kosketusnäytön käytettävyyden erikoispiirteet.....	15
3.1	Hiiren ja suoran kosketuksen eroja.....	17
3.2	Luonnolliset käyttöliittymät.....	19
3.2.1	Luonnollisen käyttöliittymän suunnittelu.....	20
3.2.2	Saumaton käyttökokemus.....	21
3.3	Kosketusnäyttöohjain.....	22
4	Ohjainten suunnittelu.....	24
5	Prototyypin toteutus.....	29
5.1	Kosketuspisteestä kentän pisteeksi.....	29
5.2	Ohjaimen tilojen suunnittelu.....	30
6	Käytettävyystestien toteuttaminen.....	32
6.1	Ensimmäinen käytettävyystesti.....	32
6.2	Parannustoimenpiteet käytettävyystestin pohjalta.....	35
6.2.1	Kosketuksen tunnistusalueen laajennus.....	36
6.2.2	Reitin laskeminen.....	38
6.3	Toinen käytettävyystesti.....	41
6.4	Parannusehdotukset toisen käytettävyystestin pohjalta.....	45
7	Pohdinta.....	47
	Lähteet.....	51

### Liitteet

Liite 1	Ensimmäisen käytettävyystestin monivalintakyselyn tulokset
Liite 2	Testitarina
Liite 3	Toisen käytettävyystestin monivalintakyselyn tulokset

## 1 Johdanto

Opinnäytetyön tarkoituksena oli suunnitella, kehittää ja testata kosketusnäyttö-ohjain peliin. Opinnäytetyön toimeksiantajana oli Polarbunny Oy, joensuulainen pelialan yritys. Opinnäytetyön prosessiin kuului ohjainten suunnittelu suunnittelutiimin kanssa, parhaiden ideoiden valitseminen pelisuunnittelijan ja tuottajan kanssa, prototyyppien kehittäminen ideoiden pohjalta, parhaan prototyypin valinta käytettävyydestein, prototyypin kehittäminen toimivaksi tuotteeksi ja tuotteen käytettävyydestaus. Opinnäytetyö jakautui siis kolmeen eri osa-alueeseen, joita toistettiin sykleittäin: suunnitteluun, kehitykseen ja testaukseen. Ohjain suunniteltiin Parceliin, Polarbunnyn kehittämään peliin. Parcel on kolmiulotteinen ongelmanratkaisupeli, jossa on erilaisia hahmoja, joilla jokaisella on erityiskykyjä. Hahmojen erityiskykyjä käytetään hyväksi ongelmanratkaisussa. Pelin tarkoitus on saada hahmot maaliin välttämättä vihollisrobotteja ja kiinnijäämistä. Parcel on voittanut Start Me Up -liikeideakilpailun vuonna 2013. Opinnäytetyö on toteutettu Unitylla ja ohjelmointikielenä on käytetty C#:a. Unity on varsinkin indie-pelien kehittäjien suosima pelimoottori ja ohjelmointiympäristö. Myös Polarbunny käyttää Unitya pelikehityksessä.

Opinnäytetyön teoriaosuuden luvussa kaksi käydään läpi käytettävyyden psykologian perusteita. Käytettävyyden psykologia perustuu kognitiiviseen psykologiaan. Esille on nostettu käyttäjän tavoitteiden ymmärtämisen merkitys suunnittelussa, havaitseminen ja ongelmanratkaisu. Luvussa kaksi käsitellään myös käyttäjäkeskeistä suunnittelua ja käytettävyydestausta. Luvussa kolme käsitellään kosketusnäytön erityispiirteitä, luonnollisia käyttöliittymiä ja peliohjaimien suunnittelua. Ohjaimen suunnitteluprosessia kuvataan tarkemmin luvussa neljä. Suunnitellut ohjaimet käydään läpi ja eritellään, mitkä suunnitelmat valittiin prototyyppitasolle. Prototyypin toteutusta kuvataan luvussa viisi nostamalla muutamia luokkia ja funktioita esimerkiksi käytetystä ohjelmointityylistä. Luvussa kuusi kuvataan käytettävyydestejä, niiden tuloksien tulkintaa ja niiden pohjalta tehtyjä parannustoimenpiteitä, jälleen nostamalla esille muutamia ohjelmistoesimerkkejä. Luvussa seitsemän käydään läpi opinnäytetyön tuloksia ja jatkokehitysmahdollisuuksia sekä pohditaan opinnäytetyön suhdetta teoriapohjaan.

## 2 Käytettävyys

Pelin ohjain on käyttöliittymä, joka toimii rajapintana käyttäjän ja pelin välillä. Ohjaimen avulla käyttäjä antaa komentoja pelille ja saa palautetta toimistaan. Pelin ohjaamisen tulee ennen kaikkea olla hauskaa ja johdonmukaista, jotta käyttäjä voi keskittyä varsinaisiin pelitehtäviin. Ohjaimen huolellinen suunnittelu varmistaa, että peliohjain ei häiritse pelattavuutta ja on nopea ja miellyttävä oppia.

Peleissä käyttäjän tavoitteena on yleensä viihdyttää itseään. Tämä erottaa pelien käyttöliittymien suunnittelun tuottavuuden lisäämiseen keskittyvien ohjelmistojen käyttöliittymien suunnittelusta. Käyttäjän toiminnan tavoite tuottavuuteen keskittyvissä ohjelmistoissa on harvoin varsinainen ohjelmiston käyttö (Sinkkonen, Kuoppala, Parkkinen & Vastamäki 2002, 54). Pelin käyttöliittymän käytön haastavuus voi joissain peleissä olla suunniteltu osa pelimekaniikkaa. Pelien pelaamisen tulee kuitenkin olla ennen kaikkea hauskaa, oli käyttöliittymän käyttö haastavaa tai helppoa. Pelin käyttöliittymän käyttämisen hauskuus ei kuitenkaan kasva lineaarisesti, vaan vain tiettyyn rajaan asti. Käyttäjä voi kokea, että fyysisen ohjainlaitteen tai ohjelmisto-ohjaimen käyttö on vaivalloista, kunnes ohjaimen käytön oppimisen raja on ylitetty. Raja tulee vastaan, kun käyttäjä oppii pelin ohjaamisen niin hyvin ja intuitiivisesti, että hän osaa käyttää ohjainta ilman hankaluuden tunnetta ja voi keskittyä täysin pelimekaniikkaan. (Toprac 2013, 31.)

Pelien pelaaminen on vapaaehtoista, joten on tärkeää, että peli kaappaa käyttäjän huomion nopeasti sekä pitää hänet motivoituneena (Pagualyan, Keeker, Fuller, Wixon, Romero & Gunn 2009, 13). Hallinnan käsite on avainasemassa pelaamisessa ja peleistä nauttimisessa. Se on syy, miksi pelaaja tahtoo pelata. Käyttäjä nauttii pelistä eniten, kun pelissä tehdyt toimet johtavat haluttuun prosimaaliseen tulokseen, vaikka käyttäjälle onkin epäselvää, miten toimi tulee vaikuttamaan pelissä pidemmällä tähtäimellä. Jos käyttäjä ei odota, että toimet johtavat tulokseen, häntä ei kiinnosta pelata. (Toprac 2013, 22.)

## **2.1 Käytettävyyden psykologia**

Usein käyttäjätutkimuksen lähtökohtana on arkiymmärrys tai arkipsykologia. Tämä tarkoittaa sitä, että käyttäjän kohtaamia ongelmia koetetaan ratkaista intuition ja introspektion, itsensä tarkasteluun perustuvan ihmiskäsityksen pohjalta. Tällöin käytettävyyden ydintieto muodostuu psykologista koulutusta vailla olevien henkilöiden huomioista ihmisen toiminnasta ja mielestä. Arkipsykologian käyttö suunnittelussa on varsin yleistä, mutta ei kovin riskitöntä, sillä arkipsykologian käytänteet eivät ole yhtä täsmällisiä ja luotettavia kuin varsinaisen psykologisen tutkimuksen. (Saariluoma 2011, 48.) Opinnäytetyötä varten on siis tutkittu käytettävyyden psykologiaa ja suunnittelusääntöjä, jotka avustavat mielekkään käyttöliittymän rakentamisessa. Tässä luvussa nostetaan esille ihmisen tavoitteiden ymmärtäminen, havaitseminen ja tulkitseminen ja ongelmanratkaisu.

### **2.1.1 Käyttäjän tavoitteiden ymmärtäminen suunnittelussa**

Ihmisen toimintaa ohjaa aina jokin sisäinen tai ulkoinen päämäärä. Toiminta voi siis olla ihmisen sisäisen tarpeen täyttämisen tai jonkin ulkoisen pakotteen tai virikkeen tulosta. Ihmisen toiminnan voidaan ajatella koostuvan kolmesta eri vaiheesta: tavoitteesta, toimenpiteestä ja evaluoinnista. (Sinkkonen ym. 2002, 53.) Ihmisen tavoitteen ymmärtäminen helpottaa ihmisen toiminnan ymmärtämistä. Käyttöliittymäsuunnittelijan tehtävä on selvittää, mitä käyttäjä haluaa saada aikaiseksi jossain näkymässä ja mitä tavoitteita hänellä on yleisesti järjestelmän käytön suhteen. (Sinkkonen ym. 2002, 54.) Evaluointivaiheessa käyttäjä arvioi, pääsikö hän tekemillään toimenpiteillä haluamaansa maaliin vai pitääkö hänen vielä tehdä jotain. Käyttäjän täytyy saada tietää, missä tilassa tuote on ja mitä hän voi tehdä päästäkseen tavoitteeseensa. Tämä on tärkeää käyttäjän oppimisen ja järjestelmän ymmärrettävyyden kannalta. Toimenpiteen suorittamisen jälkeen on myös annettava palautetta siitä, toimiko järjestelmä kuten toivottiin ja onko tavoitteeseen päästy. Käyttäjä ei voi säätää tekemisiään, jos hän ei saa palautetta järjestelmän toiminnoista, vaan päätyy tekemään saman toimenpiteen uudelleen ajatellen, että kommentoa ei rekisteröity tai jokin meni vikaan. (Sinkkonen ym. 2002, 59–60.)

Palautteen tulee olla positiivista, sillä se vahvistaa käyttäjän mielikuvaa oikeasta toimintatavasta (Sinkkonen ym. 2002, 61). Positiiviset ilmaisut jäävät myös paremmin käyttäjän mieleen ja niiden ohjeistusta totellaan helpommin (Kuutti 2002, 52).

Pelin ohjaimen antama palaute tulisi suunnitella Sinkkosen ym. (2002, 63) viittaaman Normanin seitsemänvaiheisen toimintamallin (1988) mukaan. Normanin mallin mukaisesti käyttäjä muodostaa ensin tavoitteen, jonka mukaan hän aikoo toimia. Seuraavaksi käyttäjä suunnittelee toimenpiteensä ja suorittaa sen. Tämän jälkeen järjestelmä antaa palautetta, jonka käyttäjä havaitsee. Käyttäjä tulkitsee havaintonsa, jonka jälkeen hän vertaa tavoitettaan saamaansa palautteeseen. (Sinkkonen ym. 2002, 63.) Ohjaimen tilat tulee siis suunnitella niin, että ne vastaavat pelin tiloja ja esittävät ne käyttäjälle ymmärrettävällä tavalla. Ohjaimen täytyy myös antaa palautetta siitä, onko toiminta onnistunut. Erilaiset virhetilat täytyy erottaa selkeästi toisistaan, ettei käyttäjä jää arvailemaan, mikä meni pieleen. Pelien pelaaminen on motivoivaa pelaajalle, mutta vain siinä tapauksessa, että pelaaja arvostaa pelimekaniikka sekä uskoo, että pelin tehtävät ovat suoritettavissa ja peli vastaa käyttäjän antamaan syötteeseen riittäväällä tavalla. (Torpac 2013, 24.) Käyttäjätasoisella pelikäyttöliittymällä lisää pelaajan autonomian ja osaamisen tunnetta pelaamisen aikana (Torpac 2013, 26).

### **2.1.2 Havaitseminen ja tulkinta**

Havaitsemiseen tarvitaan aistimista ja tarkkaavaisuuden kiinnittämistä oikeisiin asioihin. Jotta havaitut asiat pystytään muistamaan myöhemmin, niille pitää löytää merkitys eli ne pitää pystyä tulkitsemaan. (Sinkkonen ym. 2002, 77.) Monissa kosketusnäyttökäyttöliittymissä esimerkiksi ohjelman asetukset voidaan avata laitteen fyysisten painikkeiden avulla. Aivan uusi käyttäjä ei pysty tällöin arvaamaan mistä asetuksia voi säätää. Peleissä käyttöliittymäsuunnittelija koettaa saada kaiken oleellisen esille, niin että käyttäjä pystyy välittömästi pelaamaan peliä. Suunnittelija voi myös auttaa käyttäjää navigoimaan tuotteen käyttöliittymässä esimerkiksi varmistamalla, että tuotteen alku ja loppupalaute on selkeästi näkyvissä. Toteutus voi olla vaikeaa varsinkin pienemmillä mobiililaitteilla. (Sinkkonen ym. 2002, 158.)



Ihmisen havainto kohteesta ei ole välttämättä todellisuutta vastaava, vaan ihmisen oma tulkinta. Ihmisen tulkinta taas voi riippua monista seikoista, joista osa on päivittäin vaihtuvia, kuten mieliala tai odotukset. Ihminen luo merkityksiä havainnoille yhdistämällä ne mielessään oleviin asioihin sekä muistoihin aikaisemmista havainnon tulkinnoista. (Sinkkonen ym. 2002, 91.) Ihminen ei kuitenkaan havaitse kaikkea, mitä käyttöliittymässä on, eikä suunnittelija itse pysty näkemään käyttöliittymää niin kuin aloittelija (Sinkkonen ym. 2002, 79). Käyttäjän pitää pystyä havaitsemaan käyttöliittymässä kaikki oleellinen ja pystyä seuraamaan toimiansa vaikutusta järjestelmän tilaan. Käyttäjän tulee pystyä tunnistamaan käyttöliittymän elementit ja mieltää ne joksikin ennen kun hän pystyy käyttämään järjestelmää. Käyttäjä etsii yhtäläisyyksiä käyttöliittymän ja aikaisemman kokemuspohjansa, tuotteen tarkoituksen ja ennakkokäsitysten välillä. (Sinkkonen ym. 2002, 77.) Tämän vuoksi uudenlaisten käyttöliittymien kehittäminen saattaa olla hankalaa, sillä niitä verrataan vanhoihin käyttöliittymiin ja juurtuneisiin käsityksiin siitä miten jokin asia toimii tai on toiminut aikaisemmin.

Palautteen lisäksi suunnittelija voi käyttää rajoituksia osoittamaan käyttäjälle, mitkä käyttöliittymän elementit ovat käytettävissä ja mitkä ovat epäaktiivisia. Käyttäjän ymmärrystä käyttöliittymästä lisäävät myös konventioiden hyödyntäminen sekä vastaavuuden hyödyntäminen. Konventio tarkoittaa tuttua asiaa, joka on opittu käyttämällä vastaavia järjestelmiä aikaisemmin. Vastaavuus taas tarkoittaa, että tuotteen osat vastaavat käyttäjän kuvaa maailmasta. Jos elementit ovat käyttöliittymässä siinä järjestyksessä kuin käyttäjä ajattelee niiden olevan oikeassa maailmassa, käyttöliittymä on helppokäyttöisempi. (Sinkkonen ym. 2002, 158–159.)

Ihmisen ja ulkoisen maailman toimintaa kuvataan skeeman, mallin ja käsitteen avulla. Käsite on asia, jota voidaan kuvata yhdellä sanalla tai virkkeellä. Käsitteeseen liittyy ihmisen käsiteverkosto, joka sisältää tiedon siitä, millainen joku asia on. Käsiteverkkoa kutsutaan myös skeemaksi. Se sisältää käsitteeseen liittyvät asiat, paikat ja tiedon siitä, mitä sillä voidaan tehdä. Kun käyttäjä käyttää käsitettä skeeman mukaisesti, hän olettaa jonkin toiminnon tapahtuvan. Tätä kutsutaan mentaaliseksi malliksi. (Sinkkonen ym. 2002, 215.) Käsitteen, skeeman ja mentaalisen mallin ymmärtäminen on tärkeää opittavan käyttöliittymän suunnittelussa, sillä sitä voi hyödyntää järjestelmän opetusmetaforien suunnitte-

lussa. Oppimista edistää, jos järjestelmän elementit vastaavat jotain ulkoisen maailman asiaa. Käyttäjä ymmärtää yhteyden käyttöliittymäelementtien ja ulkoisen maailman välillä juuri mentaalisen mallin avulla.

Ihminen luo mentaalimalleja selittääkseen tuotteen rakenteen ja toiminnan. Mentaalimallien avulla käyttäjä pystyy ennakoimaan tuotteen toimintaa. Mallien rakentamisessa käytetään yleistietoa tuotetyypistä. Malleihin vaikuttavat myös vanhat mentaalimallit, tuotteen näkyvät osat ja tuotteessa käytetty terminologia. Mentaalimallit rakentuvat käyttäjän kokeillessa tuotetta. (Sinkkonen ym. 2002, 241–242.) Käyttäjän käyttäytyminen ja virheet käytettävyydestitilanteessa auttavat suunnittelijaa päättämään käyttäjän mentaaliset mallit (Sinkkonen ym. 2002, 246).

### **2.1.3 Ongelmanratkaisu ja toiminnallinen kiintyminen**

Kyseessä on ongelmanratkaisutilanne kun käyttäjällä on tavoite, jonka suorittamiseen vaadittavista toimenpiteistä hän on epävarma. Ratkaistessaan ongelmaa käyttäjä käyttää erilaisia ongelmanratkaisutekniikoita, kuten kokeilua tai analogista päättelyä. Käyttäjä ottaa usein mallia aikaisemmasta kokemuksesta tai samankaltaisesta tuotteesta. (Sinkkonen ym. 2002, 233.) Ongelmanratkaisu on ihmiselle jokapäiväistä, ja tuotteiden käyttäminen vaatii ongelmanratkaisua (Kuutti 2003, 40). Ongelma jaetaan ongelman alkutilan kuvaukseen, eli ongelmarepresentaatioon, sekä laillisten toimenpiteiden joukkoon, ongelma-avaruuteen. Käyttöliittymässä ongelmarepresentaatio on kaikki käyttöliittymän näkyvät elementit. Ongelma-avaruus on kaikkien käyttöliittymän toimintojen joukko. Onnistunut ongelmanratkaisu vaatii, että käyttäjä näkee koko ongelma-avaruuden. (Sinkkonen ym. 2002, 234.) Kokeilu toimii melko hyvin ongelmanratkaisussa ja on hyvin yleistä peleissä. Myös tutoriaalit ovat osa pelien oppimista. Peliohjaimen käytön opettelu tapahtuu tutoriaaleissa mallioppimisen perusteella. Mallioppimisessa käyttäjä saa toimintamallin, jonka hän jäsentää. Jäsennetty malli muuttuu mielikuvaksi, jonka käyttäjä pyrkii toistamaan mielikuvan mukaisesti. Toiminta täytyy myös ymmärtää, että sen voi oppia. (Sinkkonen ym. 2002, 231.) Ulkoista apua monimutkaisempiin peleihin tarjoavat foorumit sekä muut internetohjeistukset.

Ongelmanratkaisua saattaa hidastaa joidenkin käyttäjien kohdalla se, että he ovat hyvin kiintyneitä vanhan järjestelmän ongelmanratkaisutapoihin. Uusien ongelmaratkaisukeinojen sijaan käyttäjä koettaa ratkoa ongelmia aina vanhalla totutulla tavalla. Tämä kasvattaa osaltaan kynnystä vaihtaa vanhat järjestelmät uusiin ja päivittää laitteistoa. (Sinkkonen ym. 2002, 235.) Tätä ilmiötä kutsutaan toiminnalliseksi kiintymiseksi. Suunnittelijoiden on tärkeä välttää toiminnallista kiintymistä kosketusnäyttölaitteelle suunnitellessaan, sillä laite ei syöttömenetelmiltään vastaa millään lailla graafista käyttöliittymää tietokoneella. Nielsen (1993) suosittelee Sinkkosen ym. (2002, 235) mukaan, että uuden käyttöliittymän suunnitteluun otetaan mukaan useampia suunnittelijoita, jotka suunnittelevat oman käsityksensä käyttöliittymästä. Näistä suunnitelmista tutkitaan hyvät ja huonot puolet. Näin kannattaa tehdä myös pelin käyttöliittymän suunnittelussa, sillä useamman suunnittelijan tiimi on parempi innovoimaan kuin yksi henkilö.

## **2.2 Käyttäjäkeskeinen suunnittelu**

Käytettävyys on määritelty ISO-standardin (ISO 9241-11:1998) mukaan niin, että se on vuorovaikutuksellisen tuotteen käytön ”tarkoituksenmukaisuutta, tehokkuutta ja miellyttävyyttä tietyissä käyttötilanteissa”. Näin ollen hyvään käytettävyyteen kuuluu paljon muutakin kuin hyvää käyttöliittymäsuunnittelua. Käyttäjäkeskeinen suunnittelu on yksi tapa lähestyä tuotteen toteutusprosessia. Käyttäjäkeskeisen suunnittelun keskeisiä osa-alueita ovat käyttökontekstin ja käyttäjän tarpeiden analysointi sekä vaatimusmäärittelyn ja suunnitteluratkaisujen iterointi asiakkaan kanssa. Prosessin tavoitteena on kehittää tuote, joka vastaa käyttäjän tarpeita ja toiveita parhaalla mahdollisella tavalla. (Väänänen-Vainio-Mattila 2011, 102.) Käytettävyyden arviointi ja testaus ovat myös oleellinen osa käyttäjäkeskeistä suunnittelua. Sitä tehdään joko prototyyppien tai valmiiden tuotteiden avulla. Käyttäjien tarkkailu, kyselyt ja haastattelut, ääneen ajattelu käytettävyydestä, yhteistyössä tapahtuva suunnittelu ja asiantuntija-arvioinnit ovat käyttäjäkeskeistä suunnittelua tukevia metodeja. Asiantuntija-arvioinnissa käytettävyyden ammattilaiset käyvät läpi tuotteen keskeiset ominaisuudet. Järjestelmää käydään läpi heuristiikkojen avulla ja tästä syntyy lista ongelmista, niiden korjausehdotuksista sekä ongelmien vakavuudesta. (Väänänen-Vainio-Mattila 2011, 110–113.) Käyttäjättestissä tuotteen kohderyhmää edustava henkilö suorittaa testitehtäviä ympäristössä, joka vastaa tuotteen käyt-

töympäristöä. Käytettävyydestaajat havainnoivat testin perusteella tuotteen käyttöongelmia. (Kuutti 2003, 68.) Käyttäjätestausta tehdään yleensä laboratoriossa tai tuotteen oikeassa käyttöympäristössä. Laboratoriossa on helpompi kerätä systemaattista käytettävyydestietoa. Todellisessa ympäristössä taas ilmenevät ongelmat tuotteen käytössä ja soveltuvuudessa todellisiin käyttötilanteisiin. (Väänänen-Vainio-Mattila 2011, 110–113.) Käyttäjätestit ja asiantuntija-arvioinnit eivät ole toisiaan korvaavia menetelmiä. Ne ovat luonteeltaan erilaisia ja antavat tuotteesta erityyppistä käytettävyydestietoa. (Kuutti 2003, 69.) Käytettävyydesteistä kerrotaan enemmän luvussa 2.3.

Käyttäjäkeskeinen suunnittelu ei ole tullut osaksi pelialaa niin paljon kun muita teknisiä aloja. Yleensä ohjelmiston tavoitteena on maksimoida käyttäjän tehokkuus vähemmällä vaivalla, paremmalla lopputuloksella ja suuremmalla käyttäytyvyvällisyydellä. Peleissä tavoitteena on taas maksimoida hauskuuden määrä, sillä pelit, kuten kirjallisuus ja elokuvat, on tarkoitettu viihteeksi. Pelin tulokset ovat enemmän käyttäjän kokemukseen keskittyviä kuin aineellisia tuloksia. Näin ollen pelitestauksessa on annettava prioriteetti ongelmille, jotka vaikuttavat käyttäjän kokemaan hauskuuden määrään. (Pagulayan ym. 2009, 6.)

Käytettävyydestaajan on helppo nähdä, milloin ohjelman tuottavuuden kanssa on ongelmia, mutta pelin hauskuuden kanssa näin ei aina ole. Pelien kannalta on tärkeää huomioida käyttäjän toimet ja verrata niitä pelisuunnittelijan aikomuksiin. Pelisuunnittelijan visio kertoo, mikä tekee hänen suunnittelemissaan peleistä hauskoja. Tekstinmuokkausohjelmassa käyttäjän ei olisi mielekästä koettaa painaa kolme kertaa painiketta päästäkseen haluttuun lopputulokseen, mutta peleissä vaativimpien vastustajien voittamisen voi olla suunniteltu vaativan useampia yrityksiä. Tämän takia on tärkeää pitää pelisuunnittelija osallisena käytettävyydestauksessa. (Pagulayan ym. 2009, 6–7.) Vain suunnittelija pystyy havaitsemaan milloin pelaaja ei koe käyttäjäkokemusta odotetulla tavalla (Pagulayan ym. 2009, 17). Pelattavuushaasteet kuuluvat pelin suunnitteluun ja niiden tarkoitus on lisätä pelin hauskuutta. Käytettävyydestauksen tulisi siis keskittyä lieventämään asioita, jotka vaikuttavat pelaamisen hauskuuteen ja varmistamaan, että haaste lisää pelaamisen hauskuutta, eikä vähennä sitä. (Pagulayan ym. 2009, 7.) Peleissä tavoitteiden oppiminen, strategiat ja onnistumisen taktikoiminen on osa hauskuutta. Valitettavasti ei ole aina selkeää,

minkä asioiden tulisi olla intuitiivisia ja minkä haastavia. Käyttäjiltä saatava palaute auttaa määrittelemään haastavuuden määrää. (Pagulayan ym. 2009, 11.)

### 2.3 Käytettävyytestit

Käytettävyytestit ovat ainoa objektiivinen mittaustapa arvioida tuotteen käytettävyyttä (Sinkkonen ym. 2002, 301). Käyttäjättestissä kohderyhmää edustava testihenkilö suorittaa ennalta määrättyjä testitehtäviä tuotteen prototyypillä tai valmiilla tuotteella. Testaajat arvioivat testistä tehtyjen havaintojen perusteella tuotteen käytettävyyttä. (Kuutti 2003, 68.) Käytettävyytestin tarkoitus on aina parantaa tai arvioida tuotteen käytettävyyttä (Sinkkonen ym. 2002, 297). Toinen tarkoitus on parantaa tuotteen suunnittelu- ja kehitysprosessia (Dumas & Redish 1993, 22). Tavoitteena on löytää tuotteen heikkoudet ja ongelmat niiden korjaamiseksi. Tämän vuoksi käytettävyystesteissä keskitytään yleensä negatiiviseen palautteeseen. (Dumas & Redish 1993, 193.)

Käytettävyystestejä tulisi tehdä koko tuotteen kehitysprosessin ajan (Sinkkonen ym. 2006, 299). Käytettävyystestejä voidaan tehdä valmiille tuotteelle tai tuotteen prototyypille. Iteratiivisessa tuotekehityksessä käyttäjättestit aloitetaan aikaisessa kehitysvaiheessa. Testien pohjalta kehitetään parempaa käyttöliittymää. (Kuutti 2003, 68.) Käytettävyystestejä voidaan käyttää osana kehitystyötä, tai niiden avulla voidaan arvioida onko tuotteen käytettävyyden taso tarpeeksi hyvä tuotteen julkaisemiseksi (Sinkkonen ym. 2002, 297).

Käytettävyystestiin kuuluu kolme osaa: suunnittelu, suorittaminen ja analysointi (Sinkkonen ym. 2002, 302). Testin suunnittelu aloitetaan testin tavoitteiden määrittelemisellä. Testin toteuttajilla on hyvä olla käsitys siitä, mitä tuotteelta odotetaan ennen kuin käytettävyystesti järjestetään. (Sinkkonen ym. 2002, 304–305.) Käytettävyystestisuunnitelmassa on suunniteltava tarkasti, mitä toimintoja halutaan testata ja mitä niistä halutaan saada selville (Kuutti 2003, 73). Käytettävyystesteihin valitaan mittareita tuotteen käytettävyyksvaatimuksien perusteella. Mittareita on kahdenlaisia: objektiivisia ja subjektiivisia mittareita. Objektiivisillä mittareilla mitataan järjestelmän toimintakyky ja ne ovat kvantitatiivisia. Niillä mitataan käyttäjän toimien määriä. ”Kuinka kauan käyttäjältä kesti suorittaa tehtävä” on yksi esimerkki objektiivisesta mittarista. Mittareiden tulisi olla vertailukelpoisia testien välillä. Subjektiiviset mittarit voivat olla kvantitatiivisia tai kva-

litatiivisia. Ne mittaavat käyttäjän mielipidettä tuotteesta. Käytettävyydestin jälkeen kerättävä monivalintakysely on esimerkki kvantitatiivisesta subjektiivisesta mittarista. Käyttäjältä vapaasti kysytyt kommentit ovat taas esimerkki kvalitatiivisesta subjektiivisesta mittarista. (Dumas & Redish 1993, 184–189.)

Käytettävyydestien testihenkilöiden valinta riippuu tuotteen käyttäjäkunnasta. Testiin valitaan tuotteen kohderyhmää tai -ryhmiä edustavia henkilöitä, jotka eivät ole olleet mukana tuotteen kehityksessä. Käytettävyydesteihin suositellaan valittavaksi kolmesta kuuteen testihenkilöä. Testihenkilöiden määrän lisääminen lisää löydettyjen käytettävyysongelmien määrää, mutta samalla kustannusten määrä lisääntyy. Vakavimmat käytettävyysongelmat löytyvät jo kolmen testihenkilön avulla. (Sinkkonen ym. 2002, 306–307.) Käytettävyydestiä varten laaditaan testitehtäviä, joilla on tarkoitus selvittää järjestelmän toimintojen käytettävyys. Testiin kannattaa valita järjestelmän helppoja, keskeisiä toimintoja, mutta kannattaa valita myös muutamia monimutkaisia toimintoja. Testitehtävät asetetaan testitarinaan, joka on kehyskertomus, jonka perusteella testihenkilöt toimivat testitilanteessa. Testitarinoiden tulee sijoittua testihenkilöiden arkeen ja puhua heidän kieltään. (Sinkkonen ym. 2002, 308–309.) Testitehtävien tulee myös pohjautua tuotteen käyttöön luonnollisessa ympäristössä (Dumas & Redish 1993, 23). Käytettävyydestin suunnittelusta kannattaa valmistaa suunnitteluraportti, jotta käytettävyydestiprosessin laatu pystytään varmistamaan vertailemalla prosessin dokumentointia ja raportointia (Kuutti 2003, 70). Yleisimpiä testausmenetelmiä ovat ääneen ajattelu, paritestit, yhteisläpikäynti, jälkikäteen haastattelu ja kommentointi sekä ryhmäläpikäynti (Sinkkonen ym. 2002, 309–311).

Ennen testin toteutusta tehdään pilottitesti. Pilottitestin tarkoituksena on varmistaa, että testi itsessään on toimiva. Pilottitestissä on tarkoitus käydä testi läpi testihenkilön kanssa. Tuotteen sijaan testillä on tarkoitus arvioida testin laatua. Pilottitestiä ei oteta huomioon tulosten tulkinnassa. Pilottitestillä varmistetaan, että tekniikka toimii ja testitarinat ovat ymmärrettäviä. (Sinkkonen ym. 2002, 313.) Pilottitestin jälkeen testi voidaan toteuttaa käyttäen valittuja menetelmiä.

Käytettävyydesteissä ei voida saavuttaa täysin luonnollista käyttöympäristöä, sillä käyttäjä tietää, että häntä tarkkaillaan (Kuutti 2003, 69). Testitilanteen tulee kuitenkin olla mahdollisimman luonnollinen, jotta se ei vääristäisi testin tuloksia.

Vaikka ympäristön tulee olla luonnollinen, ennen testiä kannattaa esitellä käyttäjälle testiympäristö, että se ei häiritse käyttäjää testin aikana. Käyttäjän neuvoa ja opastamista testitilanteen aikana tulee välttää tai siitä tulee ainakin sopia etukäteen, että testien tulokset pysyvät vertailukelpoisina. Tulee myös muistaa, että käyttäjän kohtaamat ongelmatilanteet kertovat paljon tuotteen käytöstä ja käyttäjän suhtautumisesta siihen. (Kuutti 2003, 75.)

Käyttäjätestin aikana kerätään paljon informaatiota, joka pitää analysoida. Informaatio on muunnettava käytettävään muotoon, jotta sitä voidaan käyttää tuotesuunnittelussa. Tieto tulisi saada helposti kerättävään muotoon, joten testin aikana tehdyt muistiinpanot kirjataan ylös ja samoin testin aikana äänitetyt nauhoitteet. Videot voidaan ladata koneelle ja numeroarvot tallentaa laskentatallukoon. (Kuutti 2003, 78–79.) Tietojen analysointiin käytetään tilastollisia menetelmiä. Ongelmien yleisyydestä tehdään johtopäätöksiä. Jos ongelma ilmenee vain yhdellä testihenkilöllä, on todennäköistä, että se on satunnainen häiriötilanne. Ongelman esiintyessä monella henkilöllä kyseessä on todellinen käytettävyysongelma. Ongelman selvittyä on sen alkuperä selvitettävä ja korjausehdotus tuotettava. Ongelman vakavuutta voidaan myös arvioida ongelman korjauksen priorisointia varten. (Kuutti 2003, 79–80.)

### **3 Kosketusnäytön käytettävyyden erikoispiirteet**

Ensimmäiset kosketusnäyttölaitteet, kuten taskutietokoneet ja kämmentietokoneet kehitettiin jo 1980-luvulla. 1990-luvulla kehitettiin kynäpohjaisia laitteita sekä kännyköitä. Vuonna 2007 julkaistu Applen iPhone ja sitä seuranneet älypuhelimet ja tabletit ovat tuoneet uuden ulottuvuuden ihmisen ja tietokoneen väliseen vuorovaikutukseen. Aikaisempien mobiililaitteiden pienet näppäimistöt ja kynät ovat muuttuneet kosketus- ja elekomennoiksi. Älypuhelimien käyttökokemusta laajentavat myös erilaiset laitteisiin asennetut sensorit; kamerat, kiihtyvyyssmittarit, gyroskoopit, GPS ja magneettisuuden mittarit. Nykyajan mobiililaitteet ovat monitoimitietokoneita, joilla voi yhdistää internetiin 3G:n, 4G:n ja langattomien verkkojen kautta. Ne toimivat myös mediasoittimina, organisointityökaluina, paikantajina ja pelialustoina. (McKenzie 2013, 106.)

Muutama vuosi sitten tuli selväksi, että tulevaisuuden pelialustojen kehitystä hallitsevat tabletit. Tietokoneet ja konsolit ovat tuskin kovin pian väistymässä, mutta on selkeää, että kannettavat kosketuslaitteet tulevat vaikuttamaan pelikehityksen suuntaan. (Au 2012, 185.) Kosketusnäyttölaitteille pelejä kehitettäessä on tärkeää ymmärtää, että kosketusnäyttö tarjoaa uudenlaisen vuorovaikutusympäristön, joka eroaa huomattavasti perinteisestä WIMP-käyttöliittymästä<sup>1</sup> käytettävyydeltään sekä käyttäjärajapinnaltaan. WIMP-käyttöliittymässä vuorovaikutuksen elementit ovat yksinkertaiset. Komentoja välitetään osoittimella, joka kerää x- ja y-koordinaatteja näytöltä. Osoittimen näppäinten tilat toimivat myös hallintavälineenä. Jos WIMP-käyttöliittymä siirretään suoraan kosketusnäyttölaitteelle, laitteen syöte on täysin rajoittunut näihin yksinkertaisiin osoitinkomentoihin, eli emuloimaan hiirtä. Vaikka tämä tietyllä tavalla yksinkertaistaa ohjelmistosuunnittelua, se myös rajoittaa mahdollisuuksia, joita kosketusnäytön syöttömekanismi voisi tarjota. (Wigdor & Wixon 2012, 168–169.)

Käyttöliittymätutkimuksen tavoitteena on kehittää tapoja, joilla ihmisen ja tietokoneen kommunikointi onnistuu vaivatta. Syöttölaitteet ovat tässä tärkeässä osassa, sillä kaikki vuorovaikutus tapahtuu niiden kautta. Syöttölaitteiden käyttäminen on aina kaksisuuntaista, sillä käyttäjä tarvitsee palautetta toimistaan. Tietokoneiden ulkopuolisessa maailmassa tämä palaute on monitasoista. Objektia koskettaessaan ihminen tuntee sen pinnan tekstuurin ja lämpötilan sekä vartalossaan sen painon. Samalla hän voi havaita sen liikkeitä ja kuulla siitä lähtevät äänet. Käyttöliittymässä palaute ei ole vastaavan laajuista. Tietokoneen käyttöliittymässä käyttäjä tuntee näppäimistön painallukset, kuulee näppäinten äänet sekä näkee painallusta vastaavan merkin näytöllä. (Isokoski 2011, 187.) Kosketusnäyttökäyttöliittymässä palaute on ainoastaan taktiilista palautetta, sillä käyttäjä tuntee vain oman kosketuksensa näyttöön. Taktiilinen palaute tarkoittaa kosketukseen perustuvaa palautetta. Taktiilinen palaute ei vielä itsessään varmistakaan sitä, että järjestelmä on huomannut kosketuksen ja että käyttäjä on koskettanut oikeaa paikkaa. (Wigdor & Wixon 2012, 85.) Sen takia onkin tärkeää, että käyttäjä ei voi tehdä käyttöliittymässä mitään peruuttamatonta, ja hänellä on aina mahdollisuus palata edelliseen tilaan (Wigdor & Wixon 2012, 143).

---

<sup>1</sup> Termi ”WIMP” tulee sanoista Windows, Icons, Menus, Pointers, eli ikkunat, kuvakkeet, menut ja osoittimet.



### 3.1 Hiiren ja suoran kosketuksen eroja

Kosketusnäyttölaitteella on huomattavasti vaikeampaa valita tarkkoja pisteitä näytöltä kuin hiirellä tietokoneen monitorilta. Tämä johtuu suoran- ja epäsuoran käyttäjäsyötteen eroista. Hiiri on hyvä esimerkki epäsuorasta käyttäjäsyötteestä, sillä se on erillään järjestelmästä. Kosketusnäyttölaitteella käyttäjä koskettaa suoraan näytön objektia, jota hän tahtoo käyttää. Tämän suoran objektien manipulaation takia kosketusnäyttölaitteella ei tarvita kursoria. Hiirtä käytettäessä näytöltä voidaan valita pisteitä jopa pikselin tarkkuudella ja kursorin kohdistaminen objektiin varmistaa käyttäjälle, että hän on painamassa oikeaa painiketta. Kosketusnäytöllä ei kyetä vastaavaan tarkkuuteen, sillä kosketusnäytöllä sormi muodostaa kosketuspisteen sijaan kosketusalueen. Kosketusalueen koko vaihtelee käyttäjien välillä ja siihen vaikuttaa myös kosketuksen voima sekä kulma, jossa sormi osuu näyttöön. Objekteja valitessa käyttäjän käsi peittää osan näytöstä, eikä hän siten näe tarkasti mitä hän koskettaa. (McKenzie 2013, 106–107.) Kuitenkin myös kosketuskäyttöliittymässä on kyettävä valitsemaan objekteja tarkasti. Suunnittelijan tulee siis ottaa huomioon, että joillekin käyttäjille on hankalampaa osua näytön objekteihin. Useilla laitevalmistajilla on omat standardinsa siitä, minkä kokoisia objekteja käyttöliittymässä saa minimissään olla.

Kosketus koostuu kolmesta eri vaiheesta. Ensimmäinen vaihe on kosketuksen tunnistaminen sormen laskeutuessa laitteen näytön pinnalle. Toinen vaihe on kosketuksen jatkuminen, jolla säädetään komennon parametreja. Viimeinen vaihe on kosketuksen loppuminen sormen noustessa näytöltä. Suurimmassa osassa kosketusnäyttölaitteita nämä vaiheet vastaavat suurin piirtein fyysisiä muutoksia. (Wigdor & Wixon 2012, 127.) Edellisessä kappaleessa todettiin, että kosketusnäytöltäkin tulee pystyä valitsemaan objekteja tarkasti. Periaatteessa on kolme tapaa, joilla käyttäjä voi valita objektin kosketusnäytöllä. Käyttäjä voi liu'uttaa sormensa objektin päälle ja nostaa sen, jolloin objekti aktivoituu. Hän voi liu'uttaa sormen objektin päälle, jolloin se aktivoituu heti. Objekti voi myös aktivoitua vain jos käyttäjä laskee sormensa suoraan sen päälle. (McKenzie 2013, 107.) Perinteisessä näppäimistöissä järjestelmä reagoi sormen laskeutuessa näppäimelle. Kosketusnäytönäppäimissä taas usein vaaditaan sormen laskeutumista tai liukumista näppäimen päälle ja sormen nostamista pois näppäimen päältä. Tämä antaa käyttäjälle mahdollisuuden hienosäätää valintojaan tai peruuttaa epätoivotun valinnan. (Wigdor & Wixon 2012, 77.) Esimerkiksi iP-

honen näppäimistö toimii tällä tavalla. Käyttäjän laskiessa sormensa näppäimistön näppäimen päälle, ilmestyy iso symboli, jossa lukee mitä painiketta käyttäjä painaa. Jos käyttäjä nostaa sormensa ylös, valinta vahvistetaan. Painikkeen ollessa väärä käyttäjä voi liikuttaa sormeaan oikean painikkeen päälle. (McKenzie 2013, 107.) Laitteistokannan ollessa laaja kosketusnäyttölaitteet usein havaitsevat kosketuksen eri tavalla riippuen laitteesta. Tämän vuoksi on tärkeää muistaa, että käyttäjälle on annettava visuaalista tai taktiillista palautetta kosketuksesta sekä siitä, rekisteröitiinkö se interaktiivisen objektin kohdalla vai kohdassa, joka ei aiheuta muutosta järjestelmän tilaan. (Wigdor & Wixon 2012, 57.)

Pelisuunnittelussa on tärkeää muistaa, että kun peli siirretään tietokoneelta toiselle alustalle, on huomioitava uuden alustan fyysiset ohjaimet. Ongelmia ilmenee jos hiirelle ja näppäimistölle suunnitellut pelit siirretään suoraan PAD-ohjaimelle tai joystickille. Näillä ohjaimilla on huomattavasti vaikeampaa valita tarkkoja x- ja y-pisteitä näytöltä. Sama ongelma ilmenee jos peli siirretään suoraan tietokoneelta kosketusnäyttölaitteelle. Kuten aiemmin todettiin, kosketusnäytöltä on vaikea valita tarkkaan pisteitä. Tämän vuoksi on tärkeää suunnitella peliohjain erikseen kosketusnäytölle, ja käyttää kosketusnäytön erityisominaisuuksia hyödyksi. (Wigdor & Wixon, 74–75.) Kosketusnäyttölaitteiden tarjoamat komentotyytit ovat mielenkiintoisia. Kosketusta on mahdollista käyttää hyödyksi pelissä kosketus- ja elekomennoilla. Useat tämän hetken laitteet tukevat myös monikosketusta ja tämä laajentaa mahdollisten komentojen joukkoa entisestään. Kosketusnäyttölaitteisiin asennetut sensorit luovat myös jännittäviä mahdollisuuksia pelien kannalta. Muun muassa kiihtyvyyssmittaria ja GPS-tekniikkaa on hyödynnetty pelissä *Zombies Run!*<sup>2</sup> (2012). Pelin tarkoituksena on suorittaa tehtäviä juoksemalla ulkona. Juoksun aikana kerrotaan tarinoita, jotka paljastavat lisää pelin juonesta. Juoksun aikana käyttäjä myös kerää selviytymistarvikkeita matkaltaan. Mobiiliteknologia ja kosketuslaitteet ovat mahdollistaneet uuden tekniikan käytön pelaamisessa ilman, että käyttäjän tarvitsee hankkia erityislaitteistoa. Se on myös tehnyt tekniikkoihin pohjautuvan pelisuunnittelun helpommaksi.

---

<sup>2</sup> *Zombies Run!* 2012. Six to Start with Naomi Alderman. Six to Start.

### 3.2 Luonnolliset käyttöliittymät

Tietokonetekniikan kehityksessä on tapahtunut useita murroksia, jotka ovat mahdollistaneet transistoritekniikan kehittyminen sekä Mooren lain kuvaama kehityssuunta mikropiirien kehittämisessä (Oulasvirta 2011, 14). Tietokoneiden laskemistehon noustessa ja hinnan laskiessa on kehitetty henkilökohtaiset tietokoneet, World Wide Web ja mobiililaitteet. Kaikki tämä on johtanut siihen, että tietokoneita käyttävien ihmisten määrä on tasaisesti kasvanut. Siinä missä tietokoneiden laskentateho on kasvanut hyvin tasaisesti, käyttöliittymistä ei voida sanoa aivan samaa. Käyttöliittymien kehityksen historiassa on nähtävissä kaksi hallitsevaa käyttöliittymätyyppiä: komentorivipohjaiset käyttöliittymät sekä graafiset käyttöliittymät. Graafisista käyttöliittymistä suosituin on WIMP-käyttöliittymä. Graafiset käyttöliittymät korvasivat komentorivipohjaiset käyttöliittymät, sillä komentojen muistaminen oli vaativaa ja ne piti kirjoittaa aina täsmällisesti oikein. (Wigdor & Wixon 2012, 4–5 ja 16–17.) Graafisissa käyttöliittymissä navigointi tapahtuu näytön objektien avulla yleensä jonkin metaforan kautta. Metafora on oikean elämän asia, joka toimii mallina uudelle asialle tuotteessa. Graafisessa käyttöliittymässä näitä ovat muun muassa työpöydät, kansiot ja ikkunat. (Sinkkonen ym. 2002, 252.) Hallitseville käyttöliittymätyypeille on toki koetettu kehittää vaihtoehtoisia malleja, kuten pelkästään menupohjaiset käyttöliittymät, mutta ne ovat usein jääneet pienemmän käyttäjäjoukon käyttöön tai palvelevat omaa käyttötarkoitustaan (Wigdor & Wixon 2012, 4–5 ja 16–17).

Tietokoneet ovat nykyään eri tavalla ihmisten hallinnassa kuin ensimmäiset tietokoneet, eikä tietokoneen käyttö yleensä vaadi ohjelmointitaitoja tai laajamittaista koulutusta. Voidaan ajatella, että nykyisen tietotekniikan kehityksen keskipisteessä on ihminen. Ihminen määrittelee tietotekniikan mielekkäät käyttötarpeet ja -tavat sekä näin ollen myös sen, minne tietokoneet seuraavaksi kehittyvät. (Oulasvirta 2011, 15.) Uusien käyttötapojen tulisi olla luonnollisia ja intuitiivisia, jolloin käyttäjän kognitiivinen kuorma vähenee. Luonnolliset käyttöliittymät tarjoavat monipuolisempia vuorovaikutusmekanismeja ihmisen ja tietokoneen välille. (Ailisto 2011, 192.) Luonnollinen käyttöliittymä tarkoittaa läpinäkyvää käyttöliittymää, jonka oppii huomaamattaan. Suurin osa kosketusnäyttölaiteista toimii luonnollisella käyttöliittymällä. Luonnollista käyttöliittymää on ehdotettu WIMP-käyttöliittymän seuraajaksi käyttöliittymien kehityksessä, sillä se ei vaadi erillisten syöttölaitteiden käytön opettelu. Luonnollisten käyttöliittymien suunnit-

telu pohjautuu ajatukseen, että käyttäjä oppii aloittelijasta asiantuntijaksi nopeasti. Aloittelijan ja asiantuntijan ero on se, että aloittelija on henkilö, joka käyttää järjestelmää ensimmäistä kertaa ja asiantuntija henkilö, joka käyttää järjestelmää siten, miten suunnittelijat ovat tarkoittaneet sen käytettäväksi sekä tuntee tyytyväisyyttä käyttäessään järjestelmää. (Wigdor & Wixon 2012, 53 ja 227.)

### **3.2.1 Luonnollisen käyttöliittymän suunnittelu**

Käyttöliittymä tulisi rakentaa siten, että käyttäjän itsenäistä oppimista tuetaan. Suunnittelussa tämä tarkoittaa sitä, että jokaisen kosketuksen tulisi johtaa seuraavaan askeleeseen tehtävän suorituksessa tai ainakin avustaa sitä kohti. Järjestelmän tärkeimpien elementtien tulisi olla välittömästi löydettävissä, jotta käyttäjä voi aloittaa järjestelmän käytön välittömästi ilman ohjeistusta. Mitä pidemmälle käyttäjä jatkaa järjestelmän käyttämistä, sitä enemmän hän löytää uusia ominaisuuksia. Käyttäjää tulisi myös rohkaista käyttämään järjestelmän näkymättömiä elementtejä, esimerkiksi elemamenteja. Järjestelmän käyttämien metaforien tulisi olla johdonmukaisia koko järjestelmässä; jos kahdella sormella nipistäminen suurentaa elementtiä jossain osassa sovellusta, sen tulisi tehdä niin muuallakin. (Wigdor & Wixon 2012, 56.) Näin käyttäjä oppii käyttämään sovelluksen uusia osia ilman opiskelua (Kuutti 2002, 55). Yhdenmukaisuus on järjestelmän oppimisen ja sen muistamisen kannalta keskeinen ominaisuus. Käyttäjän on myös nähtävä missä tilassa järjestelmä on ja mitä hän voi tehdä päästäkseen eteenpäin. Myös virhetiloista ja niiden syistä on informoitava. Luonnolliset käyttöliittymät keskittyvät tekemällä oppimiseen sekä opetukseen askel askeleelta. Näin käyttäjä oppii järjestelmän taidokkaan käyttämisen nopeasti, sekä tuntee onnistumisen tunnetta koko oppimisprosessin ajan. (Wigdor & Wixon 2012, 54–57.)

Käyttöliittymät tulisi suunnitella mahdollisimman yksinkertaisiksi. Jokainen ylimääräinen objekti näytöllä tarkoittaa lisää opittavaa käyttäjälle ja lisää väärintymmärryksen riskiä. (Kuutti 2002, 50.) Turhia käyttöliittymäelementtejä on hyvä välttää jos ne voidaan helposti korvata eleellä. Järjestelmän tilojen tulisi olla vähäisiä sekä eleiden, joilla niitä kutsutaan itsestään selviä. (Wigdor & Wixon 2012, 31.) Suunnittelijan ei kannata luottaa siihen, että ele on arvattavissa, sillä useinkaan se ei ole. Ainoa suoraan arvattava ele on suora manipulaatio, eli

objektin liikuttaminen koskettamalla sitä. Koska käyttäjä ei todennäköisesti tule arvaamaan, mitä hänen on tehtävä, on hyvä käyttää käyttöliittymäelementtejä, jotka ilmestyvät käyttäjän koskettaessa objektia ja näyttävät mitä hän voi tehdä. (Wigdor & Wixon 2012, 154.) Luonnollisen käyttöliittymän oppiminen muistuttaakin pelien tutoriaaleja, joissa käyttäjälle esitellään yksi pelin ohjainmekaniikka kerrallaan oikeassa, mutta usein rajoitetussa peliympäristössä.

### 3.2.2 Saumaton käyttökokemus

Saumaton käyttökokemus on yksi luonnollisen käyttöliittymän periaatteista. Käyttäjä tulee saada uppoutumaan käyttäjäkokemukseen niin syvällisesti, että hän unohtaa epäuskoisuuden tunteensa. Sen sijaan käyttäjä ymmärtää, että eleillä on suora yhteys käyttöliittymän elementtiin. ”Suspension of disbelief”, eli epäuskon lykkääminen on Samuel Taylor Coleridgen kehittämä termi, joka tarkoittaa henkilön tahtoa hyväksyä totena jotain, joka on mahdotonta todellisessa maailmassa. Ilmiö on tärkeä elokuvissa, animaatioissa ja peleissä, sillä käyttäjän on väliaikaisesti hyväksyttävä epätodelliselta tuntuvat elementit voidakseen nauttia esityksestä. Esimerkiksi peleissä on jo useiden vuosikymmenten ajan rakennettu valtavia pelimaailmoja, jotka simuloivat todellista ympäristöä, mutta lisäävät fantasiaelementtejä maailmaan. Realismi auttaa käyttäjiä uppoutumaan peliin. (Wigdor & Wixon 2012, 43.)

Wigdorin ja Wixonin (2012, 44) mukaan epäuskon lykkäämistä voi soveltaa myös luonnollisiin käyttöliittymiin. Tämä tapahtuu hämärtämällä fyysisen ja virtuaalisen rajaa. Käyttöliittymän tulee vastata kosketukseen jatkuvasti. Kun käyttäjä voi suoraan manipuloida näytön objekteja, käyttökokemuksesta tulee saumaton. Jos objektit näytöllä vielä muistuttavat todellisen elämän objekteja, todellisen ja virtuaalisen raja hämärtyy. Järjestelmän tulee reagoida jokaiseen kosketukseen odotetulla tavalla. Jos näin ei tapahdu, käyttökokemuksen saumattomuus häviää. Järjestelmän palautteen tulee olla ”maagisen tuntuista, sekä joko odotettua tai informatiivista”. (Wigdor & Wixon 2012, 44–45.) Ajatus epäuskon lykkäämisen käyttämisestä kosketusnäyttökäyttöliittymien suunnittelussa on mielenkiintoinen ja sopii erityisen hyvin tämän opinnäytetyön tehtävänantoon. Komentojen tulisi olla intuitiivisia ja läpinäkyviä, jotta käyttäjä pystyy keskittymään pelin varsinaiseen tarkoitukseen eli ongelmanratkaisuun. Ohjaimen tar-

koituksena on saada käyttäjä ajattelemaan, että hän manipuloi peliä suoraan ja vaikuttaa hahmojen liikkeisiin ja toimintoihin omin käsin eikä käyttöliittymän kautta.

### 3.3 Kosketusnäyttöohjain

Peliohjaimet ovat osallisena käyttöliittymäinnovaatioissa. Yleensä ohjelmistot pyrkivät säilyttämään käyttöliittymän samana versioista toiseen, sillä uuden käyttöliittymän opettelu vaatii rahaa yritykseltä ja vaivaa käyttäjiltä. Käyttöliittymän muuttaminen kannattaa vain silloin, jos tuotteen tehokkuus kasvaa niin paljon, että se korvaa kulut. Peliteollisuus taas kehittää uusia syöttölaitteita ja ohjelmistokäyttöliittymiä koko ajan. Pelit ovat kokeellinen alusta käyttöliittymätestaamiselle, sillä käyttöliittymäinnovaatiot peleissä voivat päätyä koskettamaan miljoonia ihmisiä. (Pagulayan ym. 2009, 5.)

Peliohjaimen suunnittelu vaikuttaa pelin pelattavuuteen ja peliohjaimen käytön hauskuudella on vaikutus pelin hauskuuteen. Jos käyttäjän täytyy ponnistella käyttääkseen ohjainta, tai hän ei kykene kääntämään omia aikomuksiaan ohjaimen komennoiksi, hän turhautuu. Turhautuminen voi johtaa siihen, että käyttäjä kokee pelin epäreiluksi tai pelin asettamat tavoitteet saavuttamattomaksi. On siis helppo ymmärtää, miksi käytettävyys on tärkeää peleissä. (Pagulayan ym. 2009, 17.) Pelimaailmoissa liikkuminen tapahtuu yleensä kolmiulotteisesti x-, y- ja z-akseleita pitkin tai kaksiulotteisesti x- ja y-akseleita pitkin. On tärkeää saada käyttäjä ymmärtämään pelimaailman tila spatiaalisesti. Tämä onnistuu liikuttamalla hahmoa ohjaimen avulla ja näkemällä hahmon interaktio pelimaailman objektien kanssa. Käyttäjän täytyy myös oppia kuinka ohjainta käytetään ja kuinka siitä saadaan oikeanlainen vastaus pelin tapahtumiin. (Gazzard 2013, 122–124.) Pelimaailman tilan ymmärtäminen on pelattavuuden kannalta tärkeää, sillä pelin tila on irrallaan käyttäjän kokemasta tilasta. Ohjaimen hyvä suunnittelu siis varmistaa, että käyttäjä pystyy tekemään pelin kannalta välttämättömät ohjainkäskyt intuitiivisesti sekä liikkumaan pelin tilassa vaikeuksista. Paras tapa välttää ongelmia ohjaimen suunnittelussa on selvittää tärkeimmät käyttäjätehtävät ja testata ne käytettävyystestein (Pagulayan ym. 2009, 18).

Kosketusnäyttölaitteet muodostavat kasvavan osan pelimarkkinoista, joten kosketusnäyttölaitteiden suunnittelun perusteita on hyvä soveltaa myös kosketus-

näyttölaitteille suunniteltujen pelien käyttäjärajapintaan. Kosketuskäyttöliittymän suunnittelu on hyvä aloittaa peruseleistä ja rakentaa monimutkaisempia komentoja niiden pohjalta. Keskeisiä komentoja valittaessa kannattaa miettiä, mitä komentoja käytetään eniten, mitä komentoja tehdään ensimmäisenä ja mitkä toimivat pohjana muille komennoille. (Wigdor & Wixon 2012, 24.) Ohjelmistosuunnittelun ajatellaan yleisesti koostuvan kolmesta osasta: informaatioarkkitehtuurista, interaktiosuunnittelusta ja visuaalisesta suunnittelusta (Wigdor, Wixon 2012, 115). Uudenlaisen ympäristön ja kosketuskomentojen kanssa toimimassa tulee laitekohtaisesti pohtia eri vuorovaikutustasoja laitteen kanssa: mikä on fyysisesti mahdollista, mitä laite pystyy käsittelemään ja miten laitteen käsittelemistä asioista voi tehdä komentoja. (Wigdor & Wixon 2012, 116.)

Primitiivikomennot tarkoittavat yksinkertaisimpia komentoja, joiden pohjalta järjestelmä toimii. Primitiivit ovat vuorovaikutuskielen rakennuspalikoita. Esimerkiksi hiiren klikkaus on tietokoneen käyttöliittymän primitiivi. Useimpien suunnittelijoiden ei ole koskaan tarvinnut harkita syöttölaitteiden primitiivikomentoja. Graafiset käyttöliittymät kasvoivat hiiren ympärille, mutta suunnitellessa kosketusnäytölle on tärkeä suunnitella myös uudenlaisia primitiivejä. Primitiivien huolellinen suunnittelu on tärkeä osa käyttöliittymän suunnittelua. Parhaimmillaan hyvin tehdyt primitiivit lisäävät käyttöliittymän helppokäyttöisyyttä ja tekevät siitä intuitiivisemmän käyttöä. Huonosti suunnitellut primitiivit taas tekevät järjestelmästä vaikeakäyttöisen. (Wigdor & Wixon 2012, 115–116.)

Idealisesti kaikki primitiivit olisivat standardoituja laitteelle. Windows käyttöjärjestelmässä klikkaus, tuplaklikkaus ja ctrl-näppäinyhdistelmät on standardoitu pikakomentoina. Kosketuslaitteiden kohdalla tähän ei ole vielä päästy. Eri laitteilla on erilaiset primitiivit, joten ohjelmistokehittäjät joutuvat usein suunnittelemaan oman ohjelmansa primitiivit. Tällä hetkellä suurin osa kosketusnäyttölaitteille suunnitelluista primitiiveistä matkii hiiren toimintaa. Tämä johtuu todennäköisesti siitä, että primitiivien kehitystyötä kosketusnäyttölaitteille ei ole vielä tehty. Kun kosketusnäyttölaitteiden määrä kasvaa, primitiivit kehittyvät ja standardoituvat. (Wigdor & Wixon 2012, 117.) Tällä hetkellä suunnittelijan kannattaa ottaa huomioon eleet, joita laitealusta tukee, monikosketusmahdollisuudet ja laitteen erityispiirteet ja käyttää näitä hyväksi suunnittelussa.

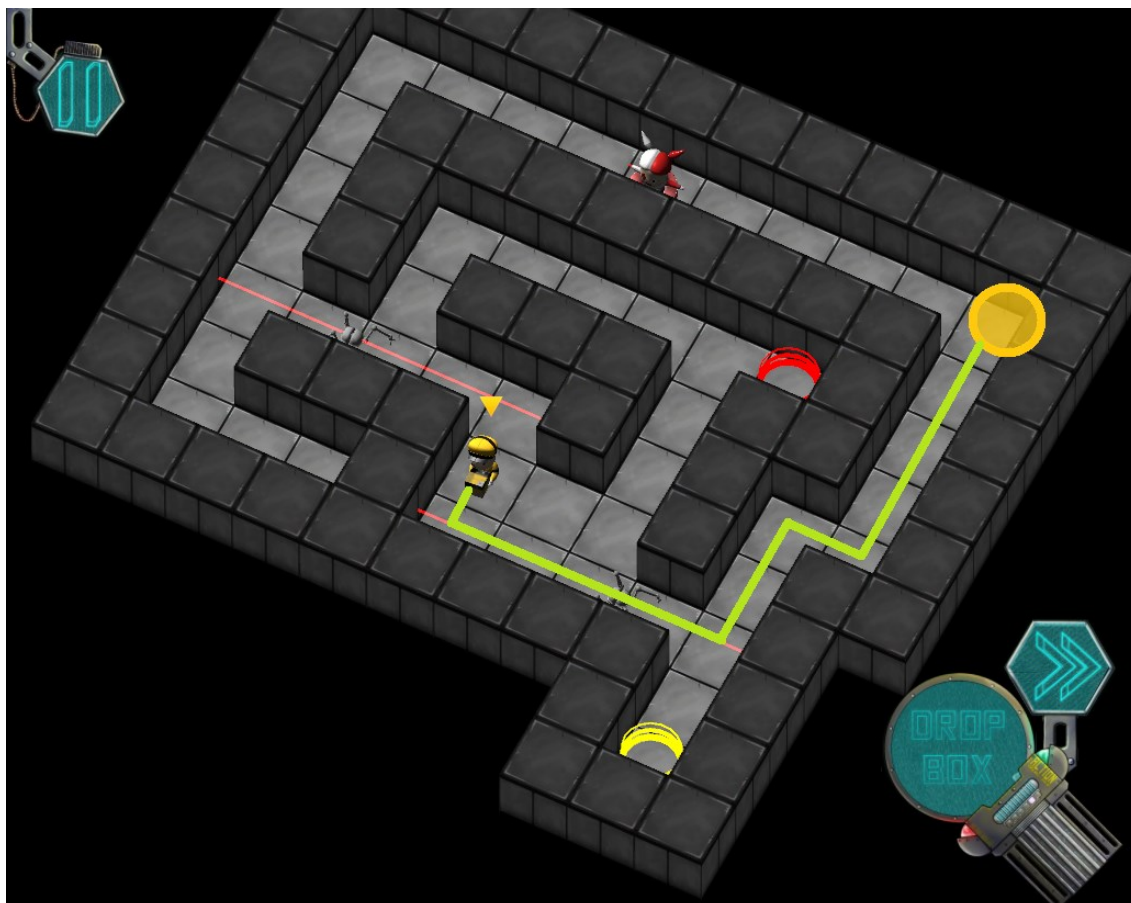
## 4 Ohjainten suunnittelu

Opinnäytetyöni lähtökohtana oli yrityksen tarve kehittää uudenlainen kosketusnäyttöohjain. Pelin mobiiliversiossa on aiemmin ollut virtuaaliset suuntapainikkeet, mutta niistä puuttuu täysin näppäinten taktiilinen palaute. Yrityksen tavoitteena oli siis kehittää uusi ohjainjärjestelmä kosketusnäyttölaitteille. Ennen varsinaisen suunnittelutyön aloitusta oli tärkeää tutustua muihin kosketusnäyttölaitteille kehitettyihin peleihin. Tarkoituksena oli selvittää millaisia ohjaimia muut pelikehittäjät olivat suunnitelleet. Testaamalla pelejä sai melko hyvän kuvan siitä, millaisia suunnitteluratkaisuja tulee välttää. Virtuaaliset suuntapainikkeet huomattiin välittömästi huonoksi ohjainratkaisuksi, sillä useissa pelissä niiden paikka on staattisesti määritetty näytölle. Ohjaimet eivät antaneet riittävästi palautetta siitä, oliko näppäimen painallus onnistunut. Onneksi myös monet suunnittelijat olivat ottaneet huomioon kosketusnäyttöympäristön tuomat uudet mahdollisuudet. Kosketusnäyttöohjainten hyvä suunnittelu teki peleistä mielenkiintoisempia ja vetoavampia. Aikaisempiin toteutuksiin tutustuminen oli hyödyllistä, sillä se antoi projektille hyvän suunnittelulähtökohdan ja korosti ohjaimen suunnittelun tärkeyttä.

Suunnitteluprosessi aloitettiin ideoimalla ohjainmalleja pienryhmässä. Työtavan tavoitteena oli löytää monipuolisempia ideoita ja saada monta eri näkökulmaa suunnitteluun. Työtapa toimi projektin kohdalla hyvin, ja pienryhmä tuotti monia mielenkiintoisia ehdotuksia, jotka vaihtelivat lauluohjaimesta perinteisten suuntapainikkeiden hiomiseen. Pienryhmän työprosessi eteni siten, että jokainen henkilö sai vuorollaan ehdottaa erilaista ohjainmekaniikkaa idean omalaatuisuudesta huolimatta. Ideoista karsittiin liian kokeelliset tai muuten toteuttamiskelvottomat ohjaimet. Tämän jälkeen ideoita hiottiin ja yksityiskohdista sovittiin ryhmän kesken. Toteuttamiskelpoisimmista ohjaimista tehtiin paperiprototyypit, jotka esiteltiin yrityksen tuottajalle ja pelisuunnittelijalle.

Aivoriihen tuloksena suunnitellut ohjaimet käytiin ensin läpi tuottajan kanssa. Hän kertoi omat mielipiteensä ideoista, paranteli niitä ja valitsi parhaat prototyyppivaihetta varten. Seuraavaksi kuvataan tärkeimpiä suunnitelluista prototyypeistä ja eritellään, mitkä niistä valittiin prototyyppivaiheeseen ja mitkä hylättiin.



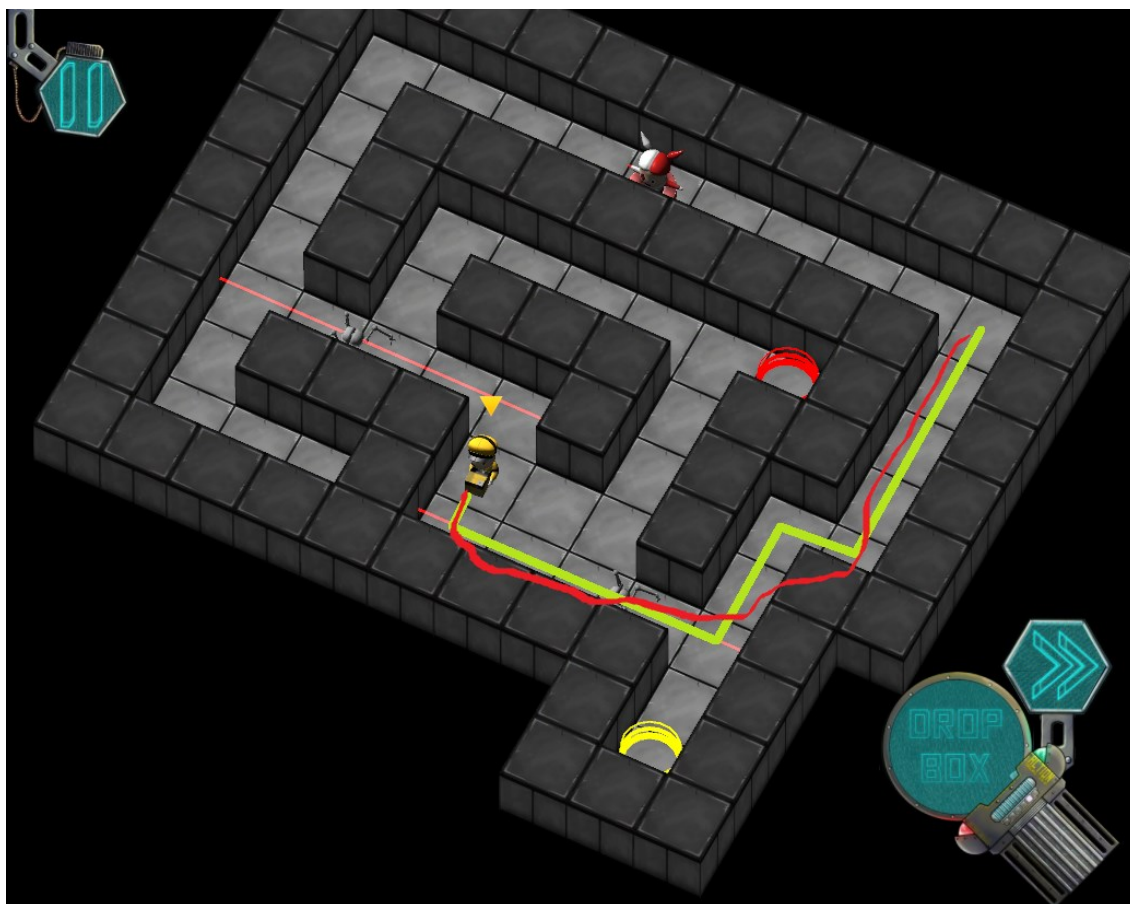


Kuva 1. Reitin laskeminen. Hahmo, jonka päällä on keltainen nuoli, on tällä hetkellä aktiivinen, ja keltainen ympyrä kuvastaa kosketuspistettä.

Ensimmäinen valituista ohjainehdotuksista perustui reitin laskemiselle (kuva 1). Tällöin käyttäjän koskettaessa pelilaudan ruutua järjestelmä laskee lyhyimmän reitin ruutuun. Sormea nostamalla reitin voi vahvistaa tai vaihtoehtoisesti vetää sormen toiseen ruutuun, jolloin reitti lasketaan uudelleen. Reitinlaskujärjestelmästä oli valittu muutamia variaatioita prototyyppiehdotuksiksi. Lyhyimmän reitin sijaan käyttäjälle voidaan näyttää kaikki mahdolliset reitit valittuun ruutuun ja käyttäjä voi valita näistä parhaan. Reitin pisteiden muokkaaminen kuului myös ehdotuksiin. Tällöin käyttäjä voi säätää reittiä mieleisekseen liikuttamalla viivaa kentällä, niin että se kulkee eri pisteiden kautta. Myös reitin vahvistus- ja perustoiminnoista oli erilaisia variaatioita. Tuottaja hylkäsi reitinlaskemisen prototyyppilistalta, sillä se helpottaa pelaamista liikaa ja vähentää pelin dynaamisuutta. Käyttäjän pitäisi odottaa hahmon liikkuvan valittuun pisteeseen, eikä se sovi pelin luonteeseen. Tuottaja kuitenkin ehdotti parannettua reitinlaskusuunnitelmaa, joka laskee reitin lyhyemmällä välimatkalla. Suunnitelman mukaan käyttäjän koskettaessa samalla rivillä tai sarakkeella olevaa ruutua, hahmo liikkuu

kosketettuun ruutuun. Jos hahmosta vedetään johonkin suuntaan, se kohdistaa hahmon kyseessä olevaan suuntaan. Tuottajan toivomuksesta suunnitelma li-  
sättiin prototyypilistaan.

Seuraava esitetty ehdotus oli reitin piirtäminen kenttään sormella. Reitti piirre-  
tään valitun hahmon päältä haluttuun ruutuun (kuva 2). Kun sormi nostetaan  
ylös, reitti vahvistetaan, ja hahmo liikkuu reittiä pitkin lopetusruutuun asti. Reittiä  
ei voi muokata kesken liikkeen, mutta sen voi perua kokonaan. Reittiä ei voi  
piirtää suoraan seinän läpi. Järjestelmän täytyy kuitenkin hyväksyä joitain piirto-  
virheitä. Esimerkiksi kulmissa käyttäjän kosketuksen osuessa seinän päälle,  
järjestelmän täytyy ymmärtää, että käyttäjä tahtoo todellisuudessa valita reitin  
kulman ympäri. Sormen piirtojälgjen ei voida ajatella olevan kovin tarkkaa tai  
täsmällistä, sillä Unity havaitsee kosketuspisteenä kosketusalueen keskustan.  
Piirto-ohjain valittiin toteutettavaksi prototyypinä.



Kuva 2. Piirtojärjestelmä. Punaisella viivalla on kuvattu käyttäjän kosketus ja  
vihreällä järjestelmän ymmärtämä reitti.

Prototyypiehdotuksiin kuului muutama variaatio piirto-ohjaimesta. Vaihtoehtona reitin vahvistamiselle nostamalla sormi näytöltä oli välitön liikkuminen. Tällöin hahmo alkaa liikkua välittömästi sen jälkeen, kun reittiä on alettu piirtää. Välitön liikkuminen tekee järjestelmästä kuitenkin virheherkän ja sitä kautta vaikeakäyttöisen, sillä käyttäjän voi olla vaikeaa korjata piirtojalkeeseen virhetilanteissa. Ongelmaa voi helpottaa pysäytystoiminnolla. Pysäytystilassa käyttäjä voi piirtää reitin ja muokata sitä rauhassa juuri oikeanlaiseksi. Tila tukee myös useiden hahmojen liikuttamista samanaikaisesti, jolloin käyttäjä voi suunnitella pelistrategiaansa kaikkien hahmojen avulla. Hyvin samankaltaista ideaa käytetään *Dragon Age: Origins*-pelissä<sup>3</sup> (2009). Peli voidaan pysäyttää kesken taistelun, ja jokaisen pelaajan joukkueessa olevan hahmon seuraava liike voidaan määrätä. Pelin palatessa normaalitilaan, hahmot siirtyvät tekemään käskyjen mukaisia toimia. Tätä ehdotusta ei valittu prototyyppiä varten, sillä se hidastaa pelin luontaista kulkua liikaa.

Ehdotus välittömästä liikkumisesta toimi kuitenkin toisen ohjainsuunnitelman perustana. Ajatellaan, että hahmoon on kiinnitetty virtuaalinen ja näkymätön talutushihna. Kun hahmoa vedetään johonkin suuntaan, se seuraa liikettä muutamana ruudun viiveellä, mutta pysähtyy kuitenkin heti kun liike pysähtyy. Periaatteessa ehdotus on hieman jalostettu versio suuntapainikkeista. Siitä puuttuvat vain visuaaliset näppäimet ja kosketuksen rajoittaminen näppäinten alueelle. Virtuaalinen talutushihna valittiin toteutettavaksi prototyyppiä. Näiden kolmen suunnitelman lisäksi tuottaja hyväksyi vielä yhden ohjainsuunnitelman. Suunnitelman mukaan ohjaus toimii siten, että käyttäjän pitäessä sormea näytöllä, hahmo liikkuu eteenpäin. Kun sormi nostetaan, hahmo pysähtyy. Kun näyttöä koskettaa hahmon kohdalta, hahmo kääntyy 90 asetetta.

Muut hylätyt ohjainjärjestelmät olivat mielikuvituksellisempia, kuten aiemmin mainitut ääni- ja laulukomennot. Äänikommentoina olisivat toimineet pelille annetut lyhyet käskyt, joilla liikkumista, hahmonvaihtoa ja hahmojen toimia hallitaan. Hylättyihin ohjainideoihin kuului myös kallistusohjain, käskyjen jonoon laittaminen ja hahmon työntäminen kentällä. Nämä suunnitelmat hylättiin pelin linjaukseen sopimattomina. Lopuksi tuottajan kanssa sovittiin vielä hahmojen vaihtamiseen, reitin perumiseen, reitin piirtämisen avustamiseen ja hahmojen erityis-

---

<sup>3</sup> BioWare. *Dragon Age: Origins*. 2009. Electronic Arts.

kykyihin liittyvistä yksityiskohdista. Sovittiin myös, että valmiissa tuotteessa tulee olemaan kaksi kameratilaa. Toinen on 3D-kamera, joka pelissä on jo aiemmin toteutettuna ja toinen piirtämiseen tarkoitettu kamera, joka kuvaa kenttää ylhäältä päin ja siten helpottaa piirtämistä. Kameroiden toteutus ei kuitenkaan kuulu tämän opinnäytetyön piiriin.

Seuraavaksi tuottajan karsima lista ohjainideoista esitettiin pelisuunnittelijalle. Hän karsi prototyypilistasta huonot ohjainsuunnitelmat omaan kokemukseensa nojaten. Pelisuunnittelijalla oli esimerkkejä samantyyllisistä ohjaimista, jotka tuntuivat pelattaessa jäykiltä. Niinpä lopullisiksi toteutettaviksi prototyypeiksi valittiin piirto-ohjain ja suunnittelijan itsensä ehdottamat parannellut suuntapainikkeet. Pelisuunnittelijan mielestä uusien ohjainten kehittäminen on hyvä asia, mutta hän haluaa tarjota pelaajille jo ennestään tutun mekanismin vaihtoehdoksi. Pelisuunnittelijan kanssa sovittiin, että opinnäytetyö keskittyy piirto-ohjaimen toteutukseen. Peliin on jo kehitetty suuntanäppäinohjain, joka pitää sovittaa uuden lähdekoodin kanssa toimivaksi. Piirto-ohjaimen toiminta sovittiin suunnittelijan kanssa tarkasti ja siitä kirjoitettiin toiminnallinen vaatimusmäärittely.

Toiminnallisen vaatimusmäärittelyn sisältö on seuraavanlainen. Käyttäjä voi liikuttaa hahmoa piirtämällä laudalle sormellaan reitin. Reitin tulee alkaa hahmosta, jota halutaan liikuttaa ja päättyä pisteeseen, jossa hahmon halutaan olevan. Sormen nostaminen näytöltä vahvistaa reitin, ja hahmo alkaa liikkua reittiä pitkin. Jos käyttäjän piirtämä reitti ei ole validi, mitään ei tapahdu. Pelaaja voi palata piirtovaiheessa polkua pitkin takaisin päin, jolloin polun pisteet poistetaan reitistä. Koko piirtoprosessi voidaan perua laskemalla toinen sormi näyttöön. Hahmoa käännetään napauttamalla hahmon päällä. Laatikon vetäminen ja työntäminen tapahtuu siten, että sormi lasketaan laatikon päälle ja vedetään siihen suuntaan, mihin halutaan vetää tai työntää. Peliin toteutetaan myös pikakelausnappi, sillä reitin piirron valmistuttua käyttäjä jää odottamaan siirron valmistumista, mikä voi olla turhauttavaa.

## 5 Prototyypin toteutus

### 5.1 Kosketuspisteestä kentän pisteeksi

Pelin kenttä koostuu kaksiulotteisesta ruudukosta. Piirto-ohjaimen tulee muuttaa käyttäjän kosketuspisteet pelikentän pisteiksi, joita pitkin hahmo voi liikkua. Ohjaimen logiikan tulee myös varmistaa, että käyttäjä ei voi kävellä seinän, muiden hahmojen tai vihollisrobottien läpi. Vihollisrobotit lähettävät myös lasersädettä, johon osuessaan pelaaja häviää.

Kosketuspisteiden muuntaminen kentän pisteiksi on toteutettu luomalla säde, joka lähetetään kamerakomponentin lähileikkaustasolta (near clipping plane) näytön pisteen kautta maailmatilaan (world space). Unityn Physics-luokan Raycast-funktiolla tarkastetaan osuuko säde kentälle. Säde ei saa reagoida kentällä olevien objektien kanssa, joten sädettä varten on luotu nelikulmion mallinen objektiin kentän tasoon. Taso-objektille on luotu oma fysiikkakerros, jonka Collider-komponenttiin osumista Raycast-funktio tarkastelee. CheckHit-funktio (kuva 3) tarkastaa säteen osumisen taso-objektiin ja palauttaa boolean-arvon sen perusteella, osuiko säde objektiin. Funktion parametrina on olioviittaus Vector3-tyyppiseen paikkakoordinaattiin, joka on funktiota kutsuttaessa näytön kosketuspiste. Funktio muuttaa olion Vector3-arvon sen pisteen arvoksi, missä säde osui pelikentän tasossa olevaan objektiin. Osuman paikkakoordinaatti ja Raycast-funktion palauttama Boolean-arvo tallennetaan muuttujiin. Funktio tarkastaa kutsutaanko sitä kahdesti saman Update-syklin aikana. Jos kutsutaan, Raycast-funktiota ei kutsuta uudelleen, vaan muuttujiin tallennetut edellisen tarkastuksen arvot palautetaan. Funktiota kutsutaan ulkoapäin overload-metodilla, joka muuttaa kosketuspisteen Vector2-arvon Vector3-arvoksi Raycast-funktiota varten.

```

private static float lastTime;
static bool lastIsAHit;
static Vector3 lastHitPoint;
private static bool CheckHit(ref Vector3 position)
{
    if(lastTime != Time.time)
    {
        RaycastHit hitPoint;
        Ray fromCamRay = Camera.main.ScreenPointToRay(position);
        if(Physics.Raycast(fromCamRay, out hitPoint, Mathf.Infinity, TouchRayLayerMask)){
            lastHitPoint = hitPoint.point;
            lastIsAHit =true;
            position = hitPoint.point;
        }
        else
            lastIsAHit = false;
    }

    lastTime = Time.time;
    position= lastHitPoint;
    return lastIsAHit;
}

```

Kuva 3. CheckHit-funktio.

CheckHit-funktiota kutsutaan reitin piirtämisen yhteydessä TrailDrawer-luokasta. Luokka toimii ohjaimen tilojen ohjauksen logiikkana. Logiikan täytyy tarkastaa, onko CheckHit-funktion avulla saatava kenttäpiste vapaa vai onko siinä seinä, muu hahmo tai vihollisrobotti. Hahmo ei voi liikkua näiden läpi. Kenttäpisteiden tulee myös olla vierekkäisissä ruuduissa, sillä muuten hahmo ei pysty liikkumaan niiden välillä. Kenttäpisteen ollessa vapaa ja saavutettavissa, se tallennetaan BoardPlan-luokan kuvaamaan reittisuunnitelmaan. Piirretty polku hahmotetaan kentälle visuaalisesti käyttäen Unityn LineRenderer-objektia, joka piirtää viivan siihen tallennettujen Vector3-tyyppisten pisteiden välille. Reittisuunnitelman reittipisteet tallennetaan suoraan LineRenderer-objektiin, ja viivan piirtäminen seuraa sormen liikettä visuaalisesti. BoardPlan-luokan metodien avulla listaan voidaan lisätä pisteitä ja sieltä voidaan myös poistaa pisteitä sen perusteella, onko uusi reittipiste jo valmiiksi listassa. Tämä mahdollistaa reitin takaisinpäin piirtämisen pisteiden poistamiseksi listasta. Pisteiden lisääminen reittisuunnitelmaan lopetetaan, kun sormi nousee ylös näytöstä. Hahmo alkaa silloin liikkua reittiä pitkin.

## 5.2 Ohjaimen tilojen suunnittelu

Kentälle piirtämisen teknisen toteutuksen jälkeen piti suunnitella, miten muita hahmoja ohjataan. Pelissä on useita hahmoja, joiden välillä käyttäjän tulee pystyä vaihtamaan. Yksittäistä hahmoa pitää myös pystyä kääntämään ruudun si-

sällä. Hahmoilla on erityiskykyjä, joita tässä ohjainjärjestelmässä hallitaan heijastusnäytöllä (HUD<sup>4</sup>) olevalla painikkeella. Erään hahmon erityiskyky vaatii kuitenkin kosketuskomentoja. Hahmo voi laskea maahan laatikon, jota se voi vetää ja työntää.

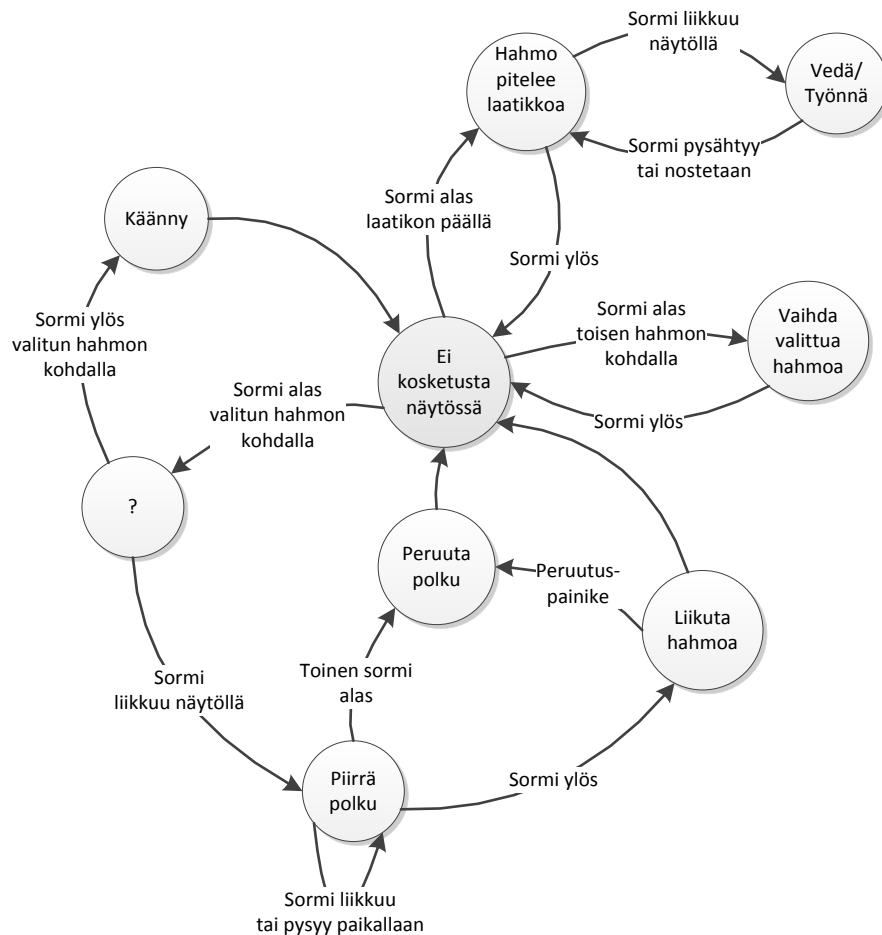
Komentoja suunnitellessa on huomioitava kehitysympäristö. Ohjaimen toteutusta varten oli siis selvitettävä, kuinka Unity tunnistaa kosketuskomennot. Unity on perustanut kosketuskirjastonsa tilojen tunnistuksen kosketusnäyttölaitteiden tunnistamiin tiloihin. Unityn tunnistamat kosketuksen tilat ovat Began (alkoi), Moved (liikkui), Stationary (paikallaan), Ended (loppui) ja Canceled (peruttu). Unity tallentaa kosketuksen paikan näytöllä kaksiulotteisen koordinaatiston pisteenä (Vector2). Kosketuksen paikka tallennetaan jokaisen Update-syklin aikana. Unity kertoo myös paikan muutoksen viimeisimmän ja sitä edeltäneen Update-syklin välillä. Kosketuksen muuttujiin kuuluu myös ajan muutos, jonka arvo kertoo, kuinka pitkä aika on kulunut edellisestä muutoksesta johonkin kosketuksen arvoon. Unity tukee myös usean kosketuksen tunnistusta. Tilojen välillä navigointi tapahtuu kuvan neljä mukaisesti.

Tilojen ja tilasiirtymien hahmottamiseksi piirrettiin tilakaavio (kuva 4). Tilakaavio on avulla oli helpompi hahmottaa suunnittelun monimerkityksellisiä komentoja. Esimerkiksi käyttäjän laskiessa sormensa valittuna olevan hahmon päälle, järjestelmä ei tiedä mitä käyttäjä aikoo, ennen kuin hän joko nostaa sormensa ylös tai liikuttaa sitä. Ohjainjärjestelmässä on koetettu vähentää monimerkityksellisiä komentoja muuttamalla tilojen siirtymät yksiselitteisiksi ja varmistamalla, että käyttäjän sormen liike rekisteröidään heti sormen laskeuduttua. Tarkoituksena oli minimoida käyttäjän eleiden määrä jonkun tietyn toiminnon rekisteröimiseksi. Monimerkityksellisissä tiloissa käyttäjän pitää tehdä enemmän vaiheita komennon suorittamiseksi. Tämä lisää järjestelmän monimutkaisuutta. Ohjaimessa on siis vähennetty samalla tavalla alkavien eleiden määrää, ja osa ohjauksesta on siirretty heijastusnäyttöpainikkeisiin. Monikosketuskomennot olivat myös yksi vaihtoehto ohjaimen suunnittelussa. Kehitetystä järjestelmästä niitä on hyödynnetty piirretyn polun perumisessa. Ohjauksen suunnittelussa on käytetty pohjana peruseleitä, joita yhdistelemällä on muodostettu monimutkaisempia komentoja. Ohjaimen teknisessä toteutuksessa on hyödynnetty Unityn tunnis-

---

<sup>4</sup> Head-Up-Display, pelin graafinen käyttöliittymä, joka antaa käyttäjälle tietoa pelin tilasta ja toiminnoista.

tamia kosketustiloja. Kosketuksen tiloja on yhdistelty merkitsemään ohjaimen eri tiloja. Tilojen tunnistuksen kautta kutsutaan ohjaimen funktioita kuvan neljä mukaisesti.



Kuva 4. Ohjaimen tiloja havainnollistava tilasiirtymäkuvaus

## 6 Käytettävyydestien toteuttaminen

### 6.1 Ensimmäinen käytettävyydestesti

Käytettävyydestesti tehtiin osana kehitystyötä peliohjaimen ollessa sellaisessa vaiheessa, että sitä voidaan käyttää. Testi tehtiin pelin piirto-ohjaimelle ja kosketusnäyttökäyttöliittymälle yhteistyössä Niko Nevalaisen (2014) kanssa. Testin tarkoitus oli osaltani tutkia, miten käyttäjät suhtautuvat uudentyyppiseen peliohjaimeen, ja selvittää sen ongelmakohdat jatkokehitystä varten. Testi tehtiin viidelle 20–25-vuotiaalle henkilölle, joilla on aikaisempaa kokemusta kosketus-



näyttölaitteista. Testiä varten kirjoitettiin testitarina, jonka mukaan testihenkilöiden oli toimittava testitilanteessa (liite 2). Testi tehtiin yksitellen jokaiselle testihenkilölle, ja testimetodi oli ääneen ajattelu. Testin jälkeen jokaiselle osallistujalle järjestettiin haastattelu ja annettiin mahdollisuus kommentoida peliä vapaamuotoisesti. Lopuksi jokaiselta osallistujalta kerättiin verkkokysely, joka koostui monivalintakysymyksistä ja avoimista kysymyksistä. Testilaitteena oli Applen iPad. Testin aikana testihenkilöitä kuvattiin kahdella kameralla, joista toinen kuvasi käyttäjän kasvoja ja toinen iPadin näyttöä. Kasvoja kuvaavalla kameralla oli tarkoitus nähdä testihenkilön reaktiot, ja toisella kameralla tarkastella mitä testihenkilö tekee tabletilla. Testihenkilön ääni tallennettiin toisella kameralla. Testi toteutettiin yrityksen toimistotiloissa, sillä kameroita ei saatu järjestettyä rauhallisempaan tilaan. Ennen käytettävyydestä toteutettiin pilottitesti, jonka tarkoituksena oli varmistaa, että käytettävyydestä ja testilaitteisto olivat toimivia.

Käytettävyydestä alkoi peliin tutustumisella. Testihenkilöille annettiin mahdollisuus tutustua järjestelmään ja opetella sen peruskomennot. Tämän jälkeen edettiin testitarinan mukaisesti. Käytettävyydestä aikana tuli ilmi, että useiden testihenkilöiden oli vaikeaa muistaa ajatella ääneen. Testin toteuttajat joutuivat usein mainitsemaan asiasta tai rohkaisemaan testihenkilöä kysymällä kysymyksiä. Kaikilta käytettävyydestä osallistuneilta henkilöiltä saatiin kuitenkin käyttökelpoista testimateriaalia. Testin tuloksia esitellään taulukossa yksi.

Taulukko 1. Testin ja haastatteluiden tulokset testihenkilöittäin.

<b>Testin tulosten kuvaus</b>	
Testihenkilö 1	<ul style="list-style-type: none"> <li>- Testaajalla oli ongelmia reitin perumisen kanssa.</li> <li>- Hahmon valinta oli vaikeaa, sillä sormen piti osua tarkasti hahmon päälle.</li> <li>- Laatikon on vetäminen vaikeaa samasta syystä kuin hahmon valinta.</li> </ul>
Testihenkilö 2	<ul style="list-style-type: none"> <li>- Laatikon vetäminen tai työntäminen oli vaikeaa, sillä laatikosta oli vaikeaa "saada kiinni".</li> <li>- Piirtämisen aikana sattui virhe: hahmo veti laatikon suoraan laserin kohdalle reitin piirtämisen sijaan. Virhe johtui siitä, että käyttäjä kosketti tabletin näyttöä kahdesta kohtaa, sillä hän käytti tablettia peukaloillaan. Käyttäjä vaikutti hyvin turhautuneelta.</li> <li>- Käyttäjä koki, että piirtäminen on vaikeaa, sillä hän "ei näe näyttöä oman sormensa läpi".</li> </ul>

Jatkuu.

Taulukko 1. Testin ja haastatteluiden tulokset testihenkilöittäin.

Testin tulosten kuvaus	
Testihenkilö 3	<ul style="list-style-type: none"> <li>- Testaaja piirsi hyvin lyhyitä reitinpätkiä, eikä käyttänyt peruuttamis-painiketta ollenkaan</li> <li>- Kulmien yli piirtäminen ei toiminut niin kuin käyttäjä odotti, joten ongelmia ilmeni etenemisessä ja reitin takaisin päin piirtämisessä. Tämä aiheutti sen, että hahmo meni paikkoihin, joihin sen ei haluttu menevän.</li> <li>- Testaaja kertoi, että hän meni usein väärin paikkoihin sen vuoksi, että hän kuvitteli sormen nostamisen peruvan reitin eikä vahvistavan sitä.</li> </ul>
Testihenkilö 4	<ul style="list-style-type: none"> <li>- Testaaja suoriutui hyvin kaikista reitin piirtämiseen liittyvistä tehtävistä</li> <li>- Hahmonvalinta-alueen tulisi kuitenkin olla suurempi.</li> <li>- Reitin peruutuspainikkeen kanssa tuli ongelmia, sillä se oli näkyvässä liian vähän aikaa.</li> </ul>
Testihenkilö 5	<ul style="list-style-type: none"> <li>- Testaajalla oli ongelmia hahmottaa piirtämismekanismia aluksi.</li> <li>- Kulmien läpi piirtäminen ei toiminut odotetusti.</li> <li>- Sormi estää näkemästä mihin reittiä piirtää.</li> </ul>

Neljä viidestä testihenkilöstä vastasi jälkikäteen lähetettyyn kyselyyn. Kyselyn monivalintatehtävät löytyvät liitteistä (liite 1). Kyselyn kysymyksissä koetettiin välttää johdattelevuutta kysymällä sama asia negatiivisesti ja positiivisesti asetelluilla kysymyksillä. Vastauksista laskettiin painotettu keskiarvo. Negatiivisten kysymysten vastaukset piti ottaa huomioon käänteisesti keskiarvoa laskiessa. Lopuksi laskettiin keskiarvo painotetuista keskiarvoista, joka oli 3.81. Maksimipistemäärä monivalintakysymyksissä oli 6. Keskiarvo ei kuitenkaan ole paras mahdollinen mittari näiden käytettävyystestituloksien arvioinnissa. Se antaa hyvän kuvan yleistyyväisyydestä, mutta ei kuvaa yksittäistapauksia. Näin ollen kokonaiskuvasta jäävät pois mahdolliset mielipiteenjakaajat, todella huonot ja hyvät tulokset. Vastausten jakautumisen voi nähdä liitteenä olevasta kyselytuloksesta. Keskiarvosta poikkeavien tuloksien tarkastelu erillisenä yleistuloksesta on hyödyllistä, sillä ne voivat osoittaa käytettävyydevirheen (Dumas & Redish 1993, 313).

Käytettävyystestissä kävi ilmi, että testihenkilöiden suurimmat ongelmat ohjaimen käytössä liittyivät kosketuksen tunnistukseen. Vaikka kerätyn kyselyn perusteella testihenkilöt olivat melko tyytyväisiä reitin piirtämiseen, kävi testeissä ilmi ongelmia, joiden korjaaminen parantaa järjestelmää merkittävästi. Suurin osa testihenkilöistä kohtasi ongelmia hahmon liikuttamisessa ja reitin piirtämisessä sen vuoksi, että se vaati tarkkaa osumista tietylle kentän alueelle. Hahmon valitseminen aiheutti ongelmia samasta syystä. Pelin 3D-kamera pahensi

ongelmia entisestään, sillä varsinainen kosketusalue on piilossa seinien takana. Kulmien yli piirtämistä ei myöskään tunnistettu riittävän tarkasti. Usea testihenkilö koetti oikaista kulman läpi piirtämällä seinän päältä useamman ruudun verran. Piirtojälfen tunnistuksen tarkkuus on parametri, jota täytyy hioa käytettävyydestien avulla sopivaksi. Tämän käytettävyydestin perusteella piirtäminen vaati käyttäjältä liikaa tarkkuutta. Laatikon liikuttamisessa ilmeni myös monia ongelmia. Suurin osa ongelmista johtui kosketuksen tunnistusalueen pienuudesta. Testihenkilöt eivät myöskään intuitiivisesti arvanneet, että heidän pitää koskettaa laatikkoa saadakseen hahmon tarttumaan siitä kiinni. Reitin peruminen oli haastavaa, sillä perumispainike ilmestyi ruudulle vain silloin kun kentälle oli piirretty reitti. Testin pelikentät olivat pieniä, joten piirretyt reitit olivat lyhyitä. Tämän vuoksi testihenkilöt eivät yksinkertaisesti ehtineet hahmottaa peruutuspainikkeen sijaintia ja painaa sitä liikkeen aikana. Suurin osa testihenkilöistä ei kyselyn perusteella kokenut pikakelauspainiketta hyödylliseksi. Pikakelaustoiminto oli suunniteltu pelaajan odotusajan vähentämiseksi isoissa pelikentissä. Pikakelaustoiminnon testaamiseen tarvittaisiin siis suurempia kenttiä. Näiden testien perusteella pikakelauspainikkeesta on vaikeaa sanoa muuta, kuin että sen sijainti pitäisi vaihtaa lähemmäksi toimintapainiketta, sillä testihenkilöillä tuntui olevan vaikeuksia käyttää painiketta myös sen poikkeavan sijainnin vuoksi.

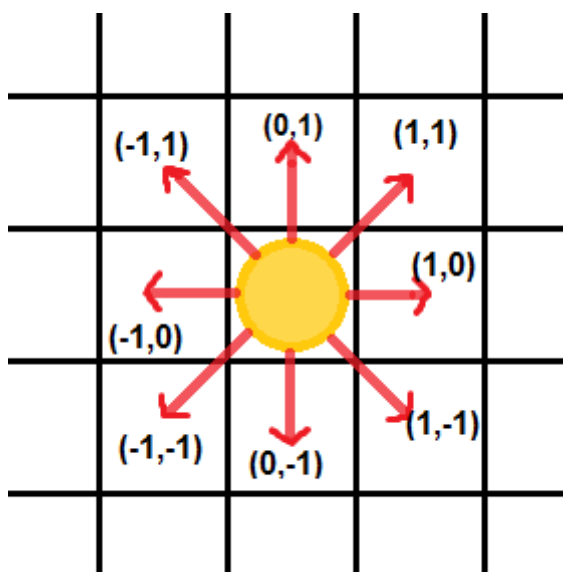
## **6.2 Parannustoimenpiteet käytettävyydestin pohjalta**

Testeissä kävi ilmi, että eniten ongelmia oli kosketuksen tunnistuksessa. Kosketus tunnistetaan liian pieneltä alueelta, ja piirtäessä pitää olla todella tarkka. Tämä ärsytti monia testihenkilöitä ja selvästi vaikutti heidän suhtautumiseensa negatiivisesti. Järjestelmään piti siis kehittää ominaisuus, joka laskee onko kosketuspisteen läheisyydessä mitään, mitä käyttäjä voi koskettaa. Kulmien yli piirtäminen aiheutti myös paljon hämmennystä. Piirtämisen helpottamiseksi päätettiin kehittää reitinlaskujärjestelmä, joka laskee reittejä viiden askeleen päähän. Jos käyttäjä piirtää kulman yli, viiva lasketaan pisteeseen, missä käyttäjän sormi palaa taas kuljettavalle reitille. Sormen päällä leijuvaa kursoria harkittiin myös ratkaisuksi sormen aiheuttamaan näköesteeseen, mutta se ei olisi välttämättä sopinut pelin linjaan. HUD:stä saatiin myös paljon palautetta. Pikakelauspainike oli ensimmäisen testin aikaan näytön yläosassa ja reitin perumispainike ilmestyi

pikakelauspainikkeen viereen siinä vaiheessa kun hahmo lähti liikkeelle. Pika-kelauspainike päätettiin siirtää toimintapainikkeen viereen, ja reitin perumis-painike päätettiin korvata toimintapainikkeella liikkeen aikana.

### 6.2.1 Kosketuksen tunnistusalueen laajennus

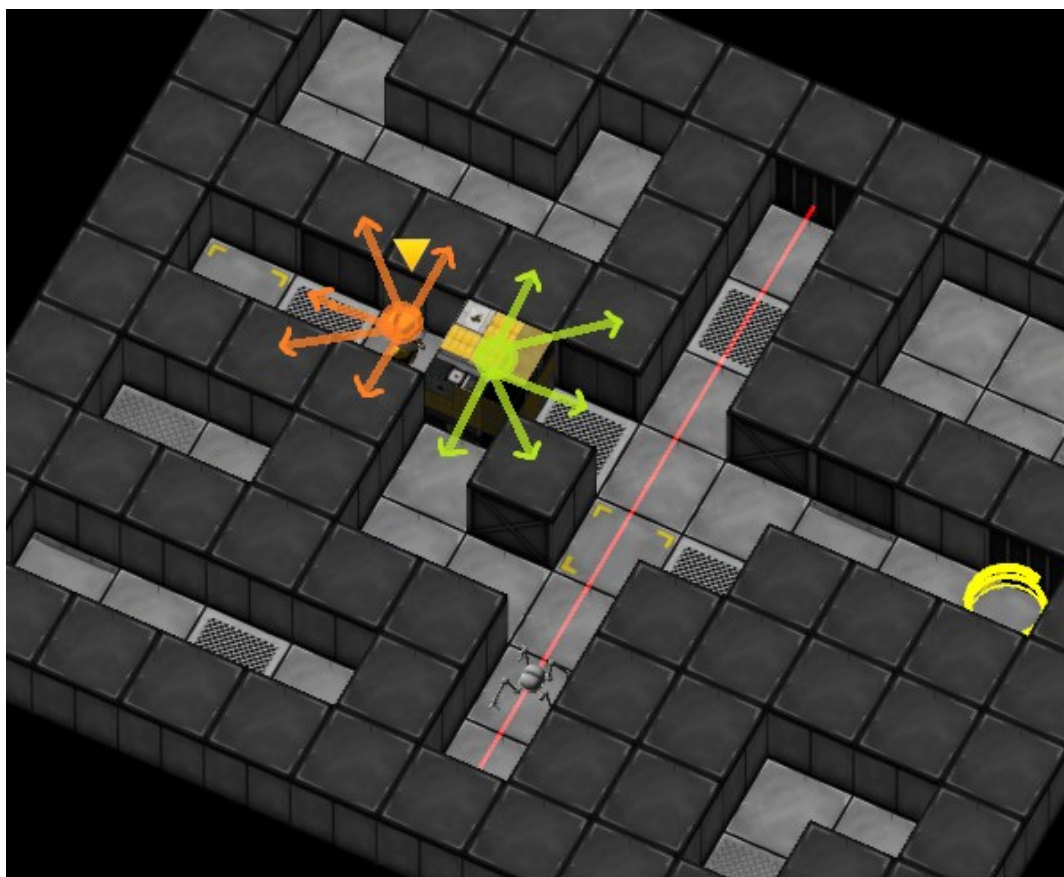
Parcelin pelilauta koostuu ruudukosta, jonka päälle pelilauta on rakennettu. Käyttäjän kosketus osuu aina yhteen pelilaudan ruutuun, joten kosketuksen tunnistusalueen laajentamiseksi kosketuspisteeseen päätettiin lisätä suuntavektoreita (kuva 5). Jos missään kosketuspisteestä ympärillä olevassa ruudussa on objekti, jota käyttäjä voi käyttää, käyttäjän katsotaan tarkoittaneen tätä objektia. Prioriteetin saa kuitenkin kosketuspiste. Kosketusalueetta ei laajenneta objektin ollessa kosketuspisteessä. Tehokkuuden vuoksi kosketusalueetta ei laajenneta, ellei kosketuspiste ole tyhjä. Kosketuspisteestä laajennus on pelissä käytössä silloin, kun pelaaja koskettaa ensimmäistä kertaa näyttöä ja silloin kun hän osuu seinään piirtäessään.



Kuva 5. Kosketusalueen laajennus pelilaudalla.

Kosketuksen osuessa jo valmiiksi oikeaan paikkaan, piste lisätään suoraan reit-tisuunnitelmaan. Kosketusalueen laajennuksen avulla hahmon valintaa ja vaih-toa, sekä reitin piirtämisen aloitusta helpotetaan. Tilanteessa, jossa kaksi hah-moa seisoo vierekkäin, kosketusalueen laajennus aiheuttaa ongelmia. Saattaa syntyä tilanne, jossa järjestelmä ei ole varma, kumpaa hahmoa käyttäjä tarkoittaa. Näissä tilanteissa hahmo ei vaihdu. Käyttäjän täytyy selventää valintaa

koskettamalla pistettä, joka on selkeästi valittavan hahmon puolella. Järjestelmän jatkokehitystä ajatellen on järkevää kehittää funktio, joka laskee kosketuspisteen etäisyyden hahmoon ja valitsee hahmon tämän perusteella. Laatikon vetäminen ja työntäminen vaatii myös kosketusalueen laajentamista. Aiemmin esitetty toteutus ei kuitenkaan toimi laatikkoa käsitellessä, joten ongelman ratkaisemiseksi kosketusalueen tunnistustoimintoa tuli muokata.



Kuva 6. Kosketusalueen laajennus laatikon ollessa kentällä

Vetäminen ja työntäminen tapahtuvat siten, että hahmon seisoessa laatikon vieressä, käyttäjä laskee sormensa laatikon päälle ja vetää sormeja siihen suuntaan, johon tahtoo laatikon liikkuvan. Tällöin hahmo ottaa kiinni laatikosta ja vetää tai työntää sitä haluttuun suuntaan. Edellinen ratkaisu ei toimi tässä tilanteessa sen takia, että laatikon ja hahmon kosketusalueet leikkaavat. Joko hahmo ei saa laatikosta saada kiinni, tai se jää kiinni laatikkoon. Ongelma ratkaistiin siten, että suoraan objektin vieressä olevat ruudut kuuluvat objektille, samoin diagonaalisesti taaksepäin olevat. Ruutu, jossa objekti on, kuulu aina objektille itselleen (kuva 6).

### 6.2.2 Reitin laskeminen

Kulmien läpi piirtämisen mahdollistamiseksi ohjainjärjestelmään tarvittiin rekursiivinen reitinlaskualgoritmi, joka laskee kaikki mahdolliset polut haluttuun pisteeseen ja valitsee niistä lyhyimmän. Puutietorakenteen huomattiin kuvaavan hyvin polkujen laskemista. Jokaisella reitin pisteellä on vanhempi – sitä ennen tullut piste – ellei piste ole reitin ensimmäinen piste. Jokaisella pisteellä on myös yksi tai useampia lapsia – sen jälkeen tulevia pisteitä – ellei piste ole polun viimeinen piste. Tietorakennetta kuvaamaan luotiin TreeElement-luokka, joka kuvaa yhtä solmua puutietorakenteessa. Sen luokkamuuttujina ovat lista solmun lapsista, tämänhetkisen solmun Vector2-arvo sekä solmun vanhempi (kuva 7).

```
public class TreeElement{  
    public List<TreeElement> children = new List<TreeElement>();  
    public Vector2 value;  
    public TreeElement parent;
```

Kuva 7. TreeElement-luokan muuttujat.

Puun rakentamista varten kirjoitettiin calculatePathsFrom-funktio (kuva 8). Varsinaista funktiota varten rekursion syvyys täytyy määrittää. Syvyyttä ei voida kuitenkaan määrittää rekursiivisen funktion sisällä, joten CalculatePathsFrom-funktiota kutsutaan overload-funktiolla. Niin kauan kuin syvyyteen viisi ei ole päästy, funktio laskee mahdolliset pisteet, joihin hahmo voi laillisesti liikkua ja lisää ne children-listaan. Tämän jälkeen funktiota kutsutaan rekursiivisesti children-listan alkioille, kunnes syvyys viisi on saavutettu. Kuvassa yhdeksän näkyy reitinlaskun periaate.

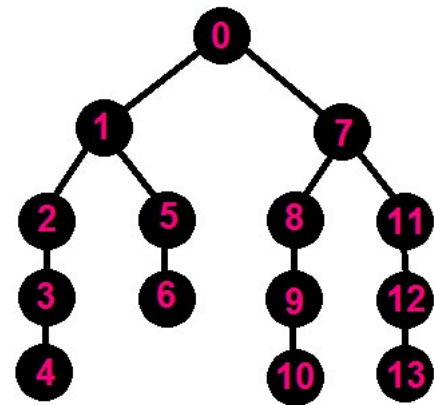
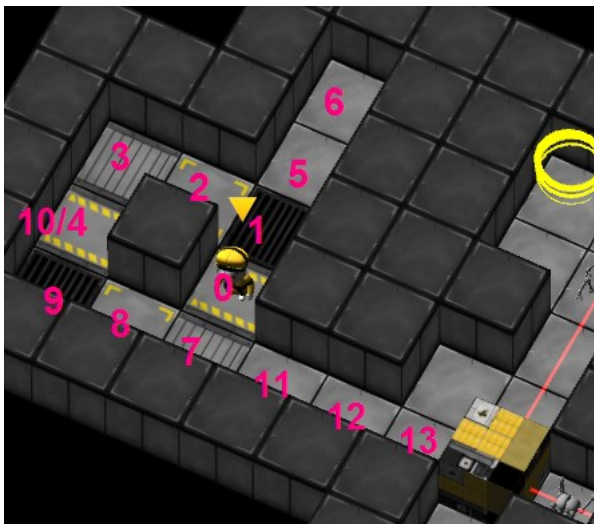
```

public void calculatePathsFrom(){
    calculatePathsFrom(1);
}

private void calculatePathsFrom(int depth){
    if(depth<5){
        foreach(Vector2 direction in directionsNext){
            Vector2 temp = value+direction;
            if(!LevelBoard.Instance.GetPiecesAt(temp).Any(x => x.IsSolid)){
                if(!GetOrderedAncestorValues(this).Any(x=>x == temp))
                    children.Add(new TreeElement(){value = temp, parent = this});
            }
        }
        foreach(TreeElement child in children){
            child.calculatePathsFrom(depth+1);
        }
    }
    else{
        return;
    }
}
}
}

```

Kuva 8. CalculatePathsFrom-funktiot.



Kuva 9. Reitin etsimisen havainnollistaminen.

GetLeaves-funktion (kuva 10) tarkoituksena on selvittää puutietorakenteen alimmat lapsettomat solmut, eli lehdet. Jos parametrina olevalla solmulla ei ole lapsia, palautetaan solmu itse. Muutoin solmun kaikki lapset käydään läpi rekursion avulla, kunnes saavutetaan lehti.

```

public static List<TreeElement> GetLeaves(TreeElement node){
    if(node.children.Count == 0){
        return new List<TreeElement>(){node};
    }else{
        var leaves = new List<TreeElement>();
        foreach(var child in node.children){
            leaves.AddRange(GetLeaves(child));
        }
        return leaves;
    }
}

```

Kuva 10. GetLeaves-funktio.

Reittien laskemiseen tarvitaan lehtien kaikki vanhemmat. Tätä varten kirjoitettiin GetOrderedAncestorValues-funktio (kuva 11). Jos parametrina olevan solmun vanhempi on olemassa, käydään rekursion avulla läpi kaikki vanhemmat siihen asti, että saavutetaan juurisolmu.

```

public static List<Vector2> GetOrderedAncestorValues(TreeElement node){
    if(node.parent != null){
        var x = GetOrderedAncestorValues(node.parent);
        x.Add(node.value);
        return x;
    }
    else{
        return new List<Vector2>(){node.value};
    }
}

```

Kuva 11. GetOrderedAncestorValues-funktio

Muualta ohjelmasta reitinlaskua kutsutaan CalculatePossiblePaths-funktiolla (kuva 12), jonka parametrina on reitinlaskun aloituspiste. Funktio palauttaa mahdolliset reitit listana Vector2-tyyppisistä listoista. Aluksi funktio luo juurisolmun, jonka arvo on sama kuin funktion parametrin arvo. Tämän jälkeen juurisolmusta lasketaan kaikki mahdolliset reitit, jonka jälkeen etsitään luodun puurakenteen alimmat lapsettomat solmut. Näistä muodostetaan kaikki mahdolliset reitit takaisin juurisolmuun. Mahdolliset reitit tallennetaan listaan listoista, jonka funktio palauttaa.



```

public static List<List<Vector2>> CalculatePossiblePaths(Vector2 startPoint){
    TreeElement root = new TreeElement(){value = startPoint};
    root.calculatePathsFrom();
    var allLeaves = TreeElement.GetLeaves(root);
    List<List<Vector2>> name = new List<List<Vector2>>();
    foreach(var leaf in allLeaves){
        name.Add(TreeElement.GetOrderedAncestorValues(leaf));
    }
    return name;
}
}

```

Kuva 12. CalculatePossiblePaths-funktio.

CalculatePossiblePaths-funktiota kutsutaan GetPathsTo-funktiolla (kuva 13), joka käy listan listoista läpi elementti kerrallaan vertaillen missä listassa haluttu piste tulee ensin esiin. Piste löytyttyä funktio palauttaa listan ennen pistettä olleista arvoista, joka lisätään hahmon liikkumislistaan. Jos pistettä ei löydy, mitään ei tapahdu.

```

List<Vector2> GetPathTo(Vector2 targetPoint){
    bool everythingIsOutOfBounds = false;
    AFewAhead = PointCalculation.CalculatePossiblePaths(boardPlan.LastAdded());
    int i = 0;
    while(!everythingIsOutOfBounds){
        everythingIsOutOfBounds = true;
        foreach(List<Vector2> plan in AFewAhead){
            if(i<plan.Count){
                everythingIsOutOfBounds = false;
                if(plan[i]==targetPoint){
                    return plan.GetRange(0,i);
                }
            }
        }
        i++;
    }
    return null;
}
}

```

Kuva 13. GetPathsTo-funktio

Reitinlaskemista voisi optimoida siten, että kaikkien mahdollisten reittien laskemisen sijaan laskettaisiin reitti lähtöpisteestä päätöspisteeseen. Tällä hetkellä reitti lasketaan viiden askeleen päähän kaikkiin mahdollisiin suuntiin. Peliohjaimen logiikan tehtävä on selvittää, mitkä poluista sisältävät halutun pisteen, ja mikä poluista on lyhyin reitti.

### 6.3 Toinen käytettävyydesti

Parannustoimien jälkeen järjestettiin toinen käytettävyydesti. Ohjaimen kehitysaika päättyi tähän käytettävyydestiin. Testin tarkoituksena oli selvittää, olivatko

ensimmäisen käytettävyydestin jälkeen tehdyt parannustoimenpiteet oikean suuntaisia ja mitä parannustoimenpiteitä voidaan tehdä jatkokehitystä ajatellen. Toinenkin käytettävyydesti tehtiin yhtäaikaaisesti ohjainjärjestelmälle ja Niko Nevalaisen (2014) suunnittelema graafiselle käyttöliittymälle. Testiä varten valittiin viisi 20–25-vuotiasta henkilöä, joilla on aikaisempaa kokemusta kosketusnäyttölaitteista. Osa testihenkilöistä oli osallistunut aikaisempaan käytettävyydestiin. Heidän kohdallaan tahdottiin selvittää, mitä mieltä he ovat peliin tehdyistä muutoksista. Testi toteutettiin ääneen ajatellen ja tekemällä havaintoja käyttäjän toiminnasta testitilanteessa. Lopuksi pidettiin loppuhaastattelu ja kerättiin kysely. Varsinaista testitarinaa ei tällä kertaa kirjoitettu, sillä edellisellä kerralla testihenkilöt hämmentyivät testitarinasta. Tultiin siihen tulokseen, että testitarina voidaan jättää pois, sillä tarkoituksena ei ole testata järjestelmän toimintaa tiettyssä tilanteessa, eikä pelillä muutenkaan ole tiettyä käyttötehtävää tai tilannetta. Testitarinan sijaan kehitettiin muutama testitehtävä, joilla testin etenemistä ohjattiin. Testi toteutettiin iPadilla, ja sitä kuvattiin kahdella kameralla edellisen käytettävyydestin tapaan. Myös ääni nauhoitettiin.

Pelin hahmojen kuvaukset ja erityiskyvyt kirjoitettiin paperille, johon testihenkilöitä pyydettiin tutustumaan ennen testin alkua. Peliä aikaisemmin pelaamattomille selitettiin myös lyhyesti pelin tavoite. Tämän jälkeen testihenkilöitä pyydettiin aloittamaan pelaaminen ensimmäisen maailman ensimmäisestä kentästä. Tabletin äänet oli tarkoituksella säädetty kovalle ennen testiä, ja testihenkilöä pyydettiin säätämään ääniä pienemmälle. Tarkoituksena oli selvittää, kaipaako käyttäjä äänivalikkoa peliin, vai käyttääkö hän tabletin omia äänenvoimakkuudensäätimiä. Tämän jälkeen testihenkilö sai pelata vapaasti muutaman kentän ja tutustua pelimekaniikkaan ja ohjaimeen. Vapaan pelaamisen jälkeen testihenkilö pyydettiin palaamaan takaisin kentän valintaan ja valitsemaan toisen maailman ensimmäinen kenttä, jossa esitellään uusi hahmo. Testihenkilöä pyydettiin pelaamaan muutama kenttä, jonka jälkeen häntä kehoitettiin pelaamaan vapaasti kenttiä ensimmäisestä nimeämättömästä maailmasta. Vapaapelaamisen jälkeen testi lopetettiin ja haastattelu aloitettiin. Taulukossa kaksi on esitelty testin tuloksia testihenkilöittäin.

Taulukko 2. Toisen käytettävyydestin tulokset testihenkilöittäin.

<b>Testin tulosten kuvaus</b>	
Testihenkilö 1	<ul style="list-style-type: none"> <li>- Testihenkilö ei aluksi hahmottanut toimintapainikkeen sijaintia vaan koetti erilaisia kosketuspohjaisia tapoja tehdä toiminta.</li> <li>- Testihenkilö oli aikaisemmin testaamassa ja sanoi, että kosketuksen tunnistus toimii paremmin.</li> <li>- Vetäminen ei onnistunut ihan joka kerralla, sillä testihenkilöllä oli vaikeuksia ymmärtää, että sormen tulee kulkea laatikon liikkeen mukana.</li> <li>- Kosketus hahmoa valitessa ei jostain syystä rekisteröidy niin kuin sen pitäisi.</li> <li>- Testihenkilö sanoi järjestelmän toimivan paremmin kuin edellisellä testauskerralla.</li> </ul>
Testihenkilö 2	<ul style="list-style-type: none"> <li>- Testihenkilölle sattui pieniä piirtovirheitä, joista osa oli kiinni sormen osumisesta väärään kohtaan ja osa tiedossa olevasta virheestä ohjaimessa (kun sormi osuu seinään silloin, kun hahmo voi mennä useita reittejä, on sattumanvaraista minkä hahmo valitsee, ellei lisää määrittelypisteitä tule myöhemmin).</li> <li>- Testihenkilön ensimmäinen refleksi reitin perumiseen oli tabletin näytön koskettaminen hahmon kohdalta.</li> <li>- Laatikon vetäminen sujui testihenkilön kohdalla hyvin, mutta muutaman kerran käyttäjä kosketti hahmoa, vaikka hänestä näytti, että hän kosketti laatikkoa. Ongelma johtui kameran kulmasta.</li> <li>- Reitti laskettiin muutaman kerran liian pitkälle, vaikka testihenkilö oli jo lopettanut sormen liikuttamisen.</li> <li>- Testihenkilön piirtämät reitit olivat melko lyhyitä.</li> <li>- Testihenkilön mielestä peli oli ihan hauska ja siinä oli sopivasti mietittävää. Liike ei hänen mielestään kulkenut tarpeeksi sulavasti, mutta haastattelussa kävi ilmi, että se johtuu lähinnä graafisista syistä (viiva häviää töksähdelin).</li> </ul>
Testihenkilö 3	<ul style="list-style-type: none"> <li>- Testihenkilöllä ei ollut lainkaan ongelmia reitin piirtämisen kanssa.</li> <li>- Hän myös koki, että on hyödyllistä, että hahmoja ei tarvitse erikseen kääntää kohdetta kohti kun tekee hahmon erityiskyvyn.</li> <li>- Testihenkilö koetti pyyhkäistä suuntaan, johon hän halusi kääntää hahmon.</li> <li>- Reitin piirtäminen hahmosta ei taaskaan onnistunut.</li> <li>- Laatikon vetäminen ja työntäminen onnistui ongelmitta.</li> <li>- Pikakelauspainike oli testihenkilön mielestä paremmassa paikassa, ja liikuminen oli helppoa.</li> </ul>
Testihenkilö 4	<ul style="list-style-type: none"> <li>- Testin alussa testihenkilölle sattui piirtovirhe, joka hämmensi testihenkilöä ja johti siihen, että testihenkilö ei aluksi uskaltanut piirtää pidempiä reittejä.</li> <li>- Hän veti myös kerran reitin suoraan laserin läpi, todennäköisesti sen takia, että sormi esti häntä näkemästä kenttää.</li> <li>- Laatikkoa liikuttaessaan testihenkilö koetti katsoa oman sormensa alta, missä hahmo menee.</li> <li>- Järjestelmä ei rekisteröinyt kosketusta hahmoa valitessa joka kerta.</li> <li>- Testihenkilön mielestä piirtäminen oli välillä haastavaa, sillä oli vaikea nähdä mistä kohti lähdetään liikkeelle.</li> </ul>

Jatkuu.

Taulukko 2. Toisen käytettävyydestin tulokset testihenkilöittäin.

Testin tulosten kuvaus	
Testihenkilö 5	<ul style="list-style-type: none"> <li>- Testihenkilö veti laatikkoa liian nopeasti, joka johti siihen, että kosketusta ei rekisteröity sormen mentyä muutaman ruudun laatikon edelle.</li> <li>- Testihenkilö mietti pitkään pikakelaus-painikkeen symbolia, epäselvä symboliikka?</li> <li>- Testihenkilö piirsi reitin laserin tielle, mutta ei käyttänyt pysäytystoimintoa.</li> <li>- Viiva lähti hahmosta vastakkaiseen suuntaan siitä, mihin testihenkilö veti. Virhe toistui useamman kerran testin aikana, mutta sitä ei enää testin jälkeen saatu toistettua.</li> <li>- Testihenkilölle jäi päällimmäisenä mieleen kohdattu outo virhe ja hän spekuloi sen syytä.</li> </ul>

Käytettävyydestin toteutus sujui ongelmitta ja materiaali oli vertailukelpoista edelliseen testiin nähden. Jälkikäteen kerätystä kyselystä laskettiin taas painotettu keskiarvo, josta huomattiin, että yleinen tyytyväisyys reitinlaskujärjestelmään on kasvanut (liite 3). Kysymyksiä oli uudelleenmuotoiltu selkeämmiksi, sekä itseään toistavat kysymykset poistettu. Ensimmäisen testin kohdalla niillä oli tarkoitus vähentää testin johdattelevuutta, mutta testihenkilöiden mielipiteet järjestelmästä vaihtuivat silti kysymyksen asettelun mukaan. Tämän testin kohdalla koettiin siis parhaaksi jättää negatiivisesti asetellut kysymykset pois. Yleistytyväisyys tuntuu olevan hyvällä tasolla, mutta muutamien kysymysten kohdalla vastaukset ovat hyvin jakautuneita. Reitin perumispainikkeeseen liittyvät kysymykset ovat tästä hyviä esimerkkejä.

Testissä ilmeni yllättävä virhe reitin piirtämisen aloittamisessa. Laajennettu tunnistusalue on käytössä myös hahmoa valitessa, mutta testin aikana ominaisuus ei toiminut odotetulla tavalla. Testissä ilmeni myös laatikon vetämisen eleen epäjohdonmukaisuus. Muualla ohjaimessa reittiä saa piirtää eteenpäin vapaasti ilman liikerajoituksia, mutta laatikkoa vetäessä sormi ei saa mennä laatikon edelle kun korkeintaan yhden ruudun verran. Jos sorme taas pitää laatikon kohdalla, on vaikea nähdä minne asti laatikko on vedetty. Suurin osa käyttäjistä ei käyttänyt pysähdyspainiketta odotetulla tavalla, vaan useammilla ensirefleksinä oli näpäyttää hahmoa jos reitin liikkuminen tahdottiin perua. Virheitä sattui useasti myös reittiä piirtäessä. Testihenkilön osuessa seinään kohdassa, jossa on mahdollista liikkua useampaan suuntaan, reitti lasketaan automaattisesti muutaman pisteen päähän. Ongelmia ilmeni jos käyttäjä ei enää aikonut sen jälkeen piirtää eteenpäin, sillä reitti piirtyi itseksensä liian pitkälle.

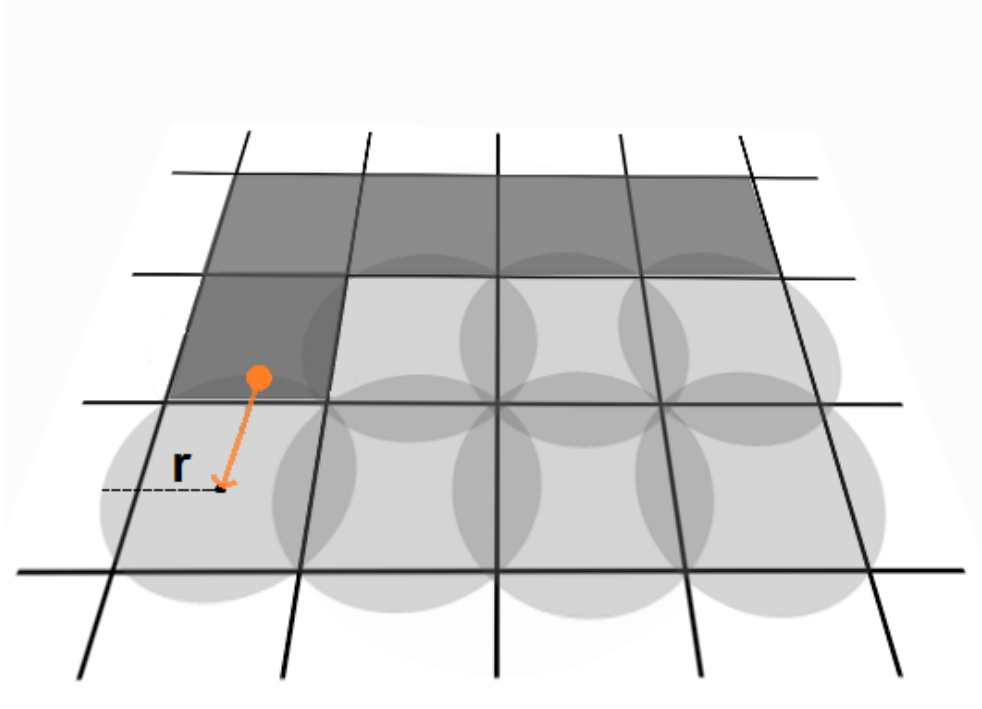
Muutoin testin tulokset olivat rohkaisevia ja osoittivat, että edellisen käytettävyydestin jälkeen tehdyt muutokset ovat olleet oikeansuuntaisia. Viimeisen testihenkilön kohdalla ohjelmassa oli virhe, joka aiheutti reitin piirtymisen vastakkaiseen suuntaan siitä, mihin käyttäjä liikutti sormeaan. Useista yrityksistä huolimatta virhettä ei pystytty toistamaan testin jälkeen. Tuotteen tulevien käytettävyydestien kohdalla kannattaa harkita vaihtoehtona kosketusnäyttölaitteen kosketuspistedatan tallentamista. Näin ongelmatilanteita voitaisiin tarkkailla paremmin.

#### **6.4 Parannusehdotukset toisen käytettävyydestin pohjalta**

Käytettävyydestien perusteella käyttäjien ensireaktio reitin perumiseen oli näytön näpäyttäminen hahmon kohdalla. On siis järkevää suunnitella komento uudelleen selkeyden vuoksi. Liikkeen peruminen hahmoa koskettamalla on helposti toteutettavissa ulottamalla kosketuksen tunnistusalueen laajennusjärjestelmä hahmoon myös liikkeen aikana. Laatikon vetämisen mekanismi aiheutti ongelmia, koska käyttäjän täytyy vetää sormea laatikon liikkeen mukaisesti. Tämä on epäjohdonmukaista järjestelmässä, jossa muu liikuttaminen tapahtuu viiveen kautta. Laatikkaa vetäessä on myös hankalaa nähdä, miten pitkälle on vedetty, sillä käyttäjän sormi peittää näkyvyyttä. Yrityksen pelisuunnittelija ehdotti ongelman korjaamiseksi, että liike tunnistetaan vain yhden akselin suhteen. Tällöin sormen liikkeellä toisen akselin suhteen ei ole väliä ja käyttäjä näkee selkeämmin, minne laatikko vedetään. Toinen vaihtoehto on implementoida virtuaalinen talutushihnajärjestelmä pidemmällä välimatkalla, siten että sormi kulkisi selkeästi laatikon edellä, ei sen kohdalla. Viivan ulkoasua tulisi myös muokata miellyttävämmäksi.

Kosketuksen tunnistusalueen laajennus on tällä hetkellä toteutettu lisäämällä suuntavektoreita kosketuspisteeseen. Vaihtoehtoinen on esitetty kuvassa 14. Tällä hetkellä kosketusalueen laajennusfunktio pyöristää sille annetut pisteet laudan pisteiksi. Tämä ei kerro miten lähelle ruutua kosketuspiste osui. Pyöristämisen sijaan voidaan laskea kosketuspisteen etäisyys lähimpien ruudukon ruutujen keskipisteisiin. Jokaisesta ruudukon ruudusta lähtee silloin ympyränmuotoinen tunnistusalue, jotka kentällä leikkaavat toisiaan. Kosketuksen osuessa ruudulle tarkistetaan, minkä ympyröiden säteen sisälle kosketuspiste osui.

Näiden ympyröiden keskipisteiden ja kosketuspisteen välinen etäisyys laskeaan, ja lyhyimmän etäisyyden päässä oleva ruutu tulkitaan kosketetuksi ruuduksi. Ympyröitä ei piirretä kentän ruuduille joissa on seinä, sillä käyttäjä ei voi piirtää seinän läpi. Tällöin seinälle osuvat kosketukset lukeutuvat johonkin seinän vieressä olevaan ruutuun osuneiksi. Ympyröitä voidaan myös painottaa muuttamalla säteen suuruutta. Esimerkiksi seinien vieressä olevien ruutujen sädettä voi kasvattaa tarpeen mukaan siten, että seinälle osunut kosketus tunnustetaan ruudulle tarkoitetuksi. Ruudukon ruutujen valitseminen etäisyyden perusteella korjaa ongelman reitinlaskemisessa. Ongelma syntyy, kun reitin piirtäminen lopetetaan kulman viereen ja viimeinen kosketuspiste on seinän päällä. Tällöin oletetaan, että käyttäjä tahtoo piirtää reitin kulman toiselle puolelle. Kosketuspisteen ja ruudun keskipisteen välimatkan laskemisen avulla ongelma saadaan helposti poistettua. Kosketuspiste tunnustetaan ruudulle, johon käyttäjä on jo piirtänyt, sillä siihen on lyhyin välimatka. Näin tarvetta laskea reitti kulman yli ei syntyisi.



Kuva 14. Kosketuspisteen laajennus kentällä. Tummanharmaalla merkityt ruudut ovat seiniä.

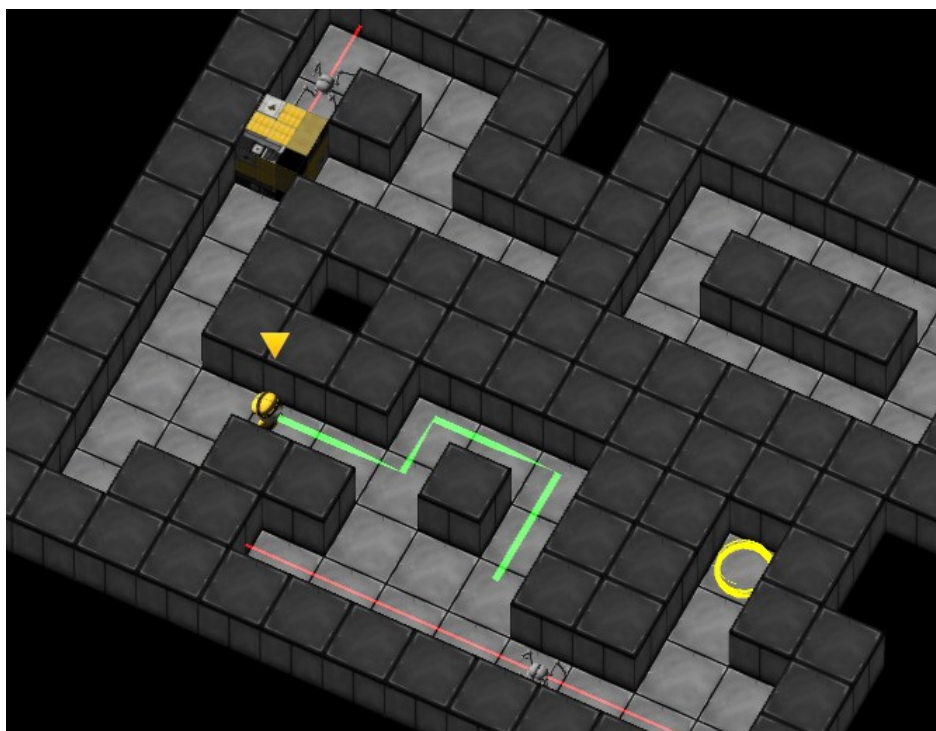
Ohjaimella voi tällä hetkellä liikuttaa vain yhtä hahmoa kerrallaan. Käyttäjä joutuu odottelemaan tai käyttämään pikakelauspainiketta kunnes reitti on kuljettu loppuun asti. Ohjaimen voi kehittää ominaisuuden, joka sallii usean hahmon liikuttamisen yhtä aikaa. Kun käyttäjä on saanut yhden hahmon reitin piirrettyä, hän voi odottamisen sijaan alkaa jo piirtää seuraavan hahmon reittiä. Tämä tekisi pelistä sulavamman ja mielenkiintoisemman käyttäjälle. Tässä luvussa esitettyjä parannusehdotuksia ei opinnäytetyön puitteissa ehditty toteuttamaan. Kehitysideoiden esittely on kuitenkin hyödyllistä järjestelmän jatkokehityksen vuoksi.

## 7 Pohdinta

Työn tarkoituksena oli suunnitella, kehittää ja testata peliohjain. Työssä onnistuttiin kokonaisuudessaan hyvin ja toiminnallisen vaatimusmäärittelyn tavoitteet saavutettiin. Ohjain on tällä hetkellä käytettävä osa peliä, ja se osoittautui käytettävyydesteissä riittävän helppokäyttöiseksi. Toimeksiantajayritys oli tyytyväinen ohjaimen kehitystyöhön, ja ohjain menee käyttöön osana peliprojektia. Ohjain kaipaa kuitenkin hienosäätöä ennen kuin se on valmis julkaistavaksi. Tärkeimmät korjattavat asiat käytettävyydestien perusteella ovat laatikon vetämisen mekanismin muuttaminen ja reitin perumismahdollisen selkeyttäminen.

Kosketusnäyttölaitteet tarjoavat mielenkiintoisen ja yhtäläisen julkaisualustan, jolle on mahdollista kehittää monenlaisia sovelluksia. Kosketusnäyttölaitteet ovat uudenlainen kehitysympäristö, jolle kehittäminen vaatii kosketusohjainten erityishuomiota käytettävyyden kannalta. Kosketusnäyttölaitteille kehittäminen on ajankohtaista, sillä kosketusnäyttölaitteet ovat yleistyneet viime vuosina. Lisäksi laitteiden väliset erot ovat pienentymässä ja laitteet ovat standardoitu-massa. Opinnäytetyöhön on kerätty yleisimpiä asioita, joita tulee ottaa huomioon kosketusnäyttölaitteille ohjelmistoja suunniteltaessa. Alalta alkaa löytyä tutkimustietoa, jota on hyvä hyödyntää kehityksessä. Kognitiiviseen psykologiaan liittyvä tieto on taas suurimmaksi osaksi melko vanhaa. Ihmiseen liittyvät tieteet kehittyvät hitaasti, kun taas teknologiaan liittyvä tieto muuttuu hyvin nopeasti. Uudelle teknologialle soveltuvia käyttötapoja on mielenkiintoista kehittää eteenpäin.

Normanin mallia ihmisen toiminnasta on hyödynnetty suunnittelussa siten, että pelin tilat on suunniteltu mahdollisimman yksiselitteisesti ja palaute selkeäksi. Tällöin käyttäjällä on vain yksi havainnollinen tapa edetä tavoitteeseensa. Komentojen suorittamiseen vaadittuja eleitä on vähennetty mahdollisuuksien mukaan, ja komennot on koetettu pitää mahdollisimman yksinkertaisina. Pelissä on visuaalisia elementtejä, joiden avulla käyttäjä voi päätellä, missä tilassa peli on. Palautteena reitin piirtämisestä käyttäjälle näytetään viiva, jota pitkin hahmo kulkee reitin valmistuttua (kuva 15). Reitistä voi helposti poistaa pisteitä piirtämällä takaisin päin, ja koko reitin piirtämistoiminnon voi perua. Näin ohjaimen kykyä palautua virheistä on koetettu parantaa. Ohjaimen antama palaute ei silti ole välttämättä tarpeeksi selkeää. Jos käyttäjä koskettaa näytöllä objektia, jonka kanssa ei voi olla vuorovaikutuksessa, ohjain ei anna palautetta. Myös käyttäjän piirtäessä reitin laserin ylitse, olisi hyvä osoittaa visuaalisesti, että reitti kulkee laserin yli. Tai jos reitti piirretään pisteeseen jonne hahmo ei voi mennä, järjestelmän olisi hyvä antaa siitä palautetta. Peliin voidaan esimerkiksi lisätä ominaisuus, jossa valittu hahmo osoitetaan pienellä hahmon kasvoja kuvaavalla ikonilla. Hahmon kasvojen ilme muuttuu sen perusteella, minne reittiä piirretään. Jos reitti piirretään ruutuun, jonne hahmo ei voi mennä, hahmo voisi kohautella olkiaan. Jos reitti taas piirretään laseriin, hahmo voisi näyttää kauhistuneelta.



Kuva 15. Hahmo liikkumassa piirrettyä reittiä pitkin.



Suunnittelun hiireen pohjaamisen sijaan on koetettu kehittää ohjain, joka hyödyntää kosketusnäyttöympäristön erityisominaisuuksia mahdollisimman hyvin. Kosketuksen kömpelyys on otettu huomioon laajentamalla kosketuksen tunnistusaluetta. Ei voida olettaa, että käyttäjä pystyy osumaan eksaktisti näytön pisteisiin. Myös käyttäjän sormi estää näkyvyyttä pelilaudalle. Pelin kamera vaikuttaa 3D-perspektiivin takia vaikeuttavasti reitin piirtämiseen. Se vaikeuttaa myös pelin tilan hahmottamista, sillä kameran aiheuttama kulma vääristää kosketuksen paikkaa tietyissä tilanteissa. Ongelman voi korjata toteuttamalla suunniteluvaiheessa ideoidun kameran, joka kuvaa pelin tapahtumia suoraan ylhäältä päin tai kehittämällä kursorin, joka leijuu juuri sormen päällä ja näin tarkentaa käyttäjän piirtämistä.

Ohjain on suunniteltu siten, että aloittelija pystyy pelaamaan sillä mahdollisimman nopeasti. Käyttäjää pitää kuitenkin avustaa jonkinlaisella tutoriaalilla, sillä ei voida olettaa kaikkien käyttäjien intuitiivisesti arvaavan, miten hahmoja liikutetaan tai miten liikkeen saa peruttua. Graafisilla GUI-elementeillä voi johdattaa käyttäjää seuraavaan askeleeseen komennon tekemisessä silloin, kun käyttäjä vielä opettelee ohjaimen käyttöä. Näin käyttäjälle esitellään käyttöliittymän ja ohjaimen tärkeimmät elementit pelaamisen yhteydessä. Ohjaimen elementit ovat suurimmaksi osaksi piilotettuja, mutta perustuvat suoraan manipulaation. Tällä suunnitteluratkaisulla on pyritty välttämään turhia käyttöliittymäelementtejä.

Käyttäjäkeskeisen suunnittelun periaatteita on käytetty ohjaimen kehitysvaiheessa. Vaatimusmäärittelyä on iteroitu yhdessä pelisuunnittelijan kanssa kehityksen edetessä. Pelisuunnittelija on ollut myös vahvasti mukana visionsa kanssa pelin suunnittelu- ja kehitysvaiheessa, joka johti siihen, että piirto-ohjain ei häiritse pelin mekaniikkaa, vaan toimii sen tukena. Käytettävyydestejä on suoritettu kehityssykylien välillä. Käytettävyydestit on toteutettu ääneen ajatellen ja havainnoimalla käyttäjää, sekä niiden jälkeen on toteutettu käyttäjähaastattelut ja kyselyt. Käytettävyydestien tuloksena on saatu lista korjattavista asioista, joka on priorisoitu ongelman vakavuuden perusteella. Prioriteettina oli korjata virheitä, jotka vaikuttivat pelin pelattavuuteen negatiivisesti ja varmistaa, että ohjaimen asettama haaste lisää pelaamisen mielekkyyttä, eikä vähennä sitä. Käytettävyydestit ovat olleet hyvä menetelmä ohjaimen käytettävyyden arvioi-

miseksi ja parantamiseksi. Kummastakin käytettävyydestä on saatu arvokasta palautetta, jonka perusteella on voitu tehdä korjaustoimenpiteitä. Testien kohdalla käyttäjän havainnointi ja ääneen ajattelu testimenetelminä ovat toimineet hyvin. Ääneen ajattelu oli joillekin testihenkilöille vaikeaa, varsinkin ongelmatilanteissa. Ääneen ajattelun tarkoituksena on saada selville ongelmatilanteet ja selvittää syyt niiden taustalla. Arviointia siis vaikeuttaa, että käyttäjät hiljenevät juuri ongelmatilanteen sattuessa. Kuitenkin käytettävyydestit ovat tarjonneet hyödyllisen mahdollisuuden saada käyttäjän mielipide kuuluviin kehitystyön yhteydessä. Suunnittelija sokeutuu helposti omien suunnitteluratkaisujensa puutteellisuudelle. Käytettävyydestit osoittavat ilmeiset puutteet käytettävyydessä hyvin nopeasti.

Ammatillisen kehitykseni kannalta kosketusnäyttölaitteille kehittäminen on ollut opettavaista. Opinnäytetyöni käsittelee laaja-alaisesti tuotekehitystä suunnittelusta testaamiseen. Kaikki vaiheet ovat opettaneet minulle uusia toimintatapoja. Tiimisuunnittelu ohjaimen suunnitteluvaiheessa oli erityisen opettava kokemus. Saimme kehitettyä paljon uudenlaisia ideoita, sillä suunnittelussa oli mukana monenlaisia näkemyksiä. Olen oppinut kommunikoimaan ja esittelemään ideoita paremmin. Opinnäytetyön toteutus tutustutti minut monimutkaisempiin ohjelmistokonsepteihin, joista on varmasti hyötyä urallani. Ohjaimen lähdekoodi koki monta ajatusvaihetta, mutta päätyi lopulta selkeään loppumuotoon. Ohjain jää yritykselle käyttöön ja osaksi peliä. Ohjelmointitaitoni ovat kehittyneet paljon: uusi teknologia tarjosi ohjelmistokehityksellisiä haasteita, joita oli mielenkiintoista ratkoa. Varsinaisten käytettävyydestien toteuttaminen oli myös kehittymiseni kannalta erityisen tärkeää. Pääsin tutustumaan käytettävyydestien hyötyihin ja hankaluuksiin, yleisimpiin testitapoihin sekä järjestämään omia käytettävyydestejä. Tietouteni käytettävyydestä ja sen testauksesta kasvoi paljon. On ollut mielenkiintoista tutustua käyttäjäkeskeisen suunnittelun periaatteisiin ja toimintatapoihin. Opinnäytetyöni teon aikana käyttäjänäkökulman tärkeys korostui ohjelmiston kehitysprosessia ajatellessa. Kokonaisuudessaan opinnäytetyö on herättänyt minussa kipinän suunnitella ohjelmistoja käyttäjälähtöisesti. Teknologia tulisi suunnitella ihmisen käytettäväksi ja intuitiiviset käyttöliittymät ovat tässä hyvä keino.

## Lähteet

- Ailisto, H. 2011. Ubiikkiteknologia. Teoksessa Oulasvirta, A. (toim.). 2011. Ihmisen ja tietokoneen vuorovaikutus. Helsinki: Gaudeamus Helsinki University Press Oy Yliopistokustannus, HYY yhtymä. 192–211.
- Au, W.J. 2012. Game Design Secrets. USA: Wiley.
- Dumas, J. S., Redish J. C. 1993. A Practical Guide to Usability Testing. UK: Intellect Books.
- Gazzard, A. 2013. The Psychology of Control and Video Games. Teoksessa Matthew Wysocki (toim.) Ctrl-Alt-Play : Essays on Control in Video Gaming. USA: McFarland. 121–132.
- Isokoski, P. 2011. Syöttölaitteet Teoksessa Oulasvirta, A. (toim.). 2011. Ihmisen ja tietokoneen vuorovaikutus. Helsinki: Gaudeamus Helsinki University Press Oy Yliopistokustannus, HYY yhtymä. 172–191.
- Kuutti, W. 2003. Käytettävyys, suunnittelu ja arviointi. Helsinki: Talentum.
- MacKenzie, S.I. 2013. Human-Computer Interaction – An Empirical Research Perspective. USA: Morgan Kaufmann, Elsevier.
- Nevalainen, N. 2014. Videopelit ja käytettävyys. Karelia-ammattikorkeakoulu. Tietotekniikan koulutusohjelma. Opinnäyte.
- Oulasvirta, A. 2011. Mitä on ihmisen ja tietokoneen vuorovaikutus? Teoksessa Oulasvirta, A. (toim.). 2011. Ihmisen ja tietokoneen vuorovaikutus. Helsinki: Gaudeamus Helsinki University Press Oy Yliopistokustannus, HYY yhtymä. 45–42.
- Pagulayan, R. J., Keeker, K., Fuller, T., Wixon, D., Romero, R.L. & Gunn, D.V. 2007. User-centered Design in Games. Studios User Research. Microsoft. 28.4.2014
- Saariluoma, P. 2011. Käyttäjä. Teoksessa Oulasvirta, A. (toim.). 2011. Ihmisen ja tietokoneen vuorovaikutus. Helsinki: Gaudeamus Helsinki University Press Oy Yliopistokustannus, HYY yhtymä. 45–61.
- Sinkkonen, I., Kuoppala, H., Parkkinen, J. & Vastamäki, R. 2002. Käytettävyyden psykologia. Helsinki: Edita Oyj.
- Toprac, P. 2013. The Psychology of Control and Video Games. Teoksessa Matthew Wysocki (toim.) Ctrl-Alt-Play : Essays on Control in Video Gaming. USA: McFarland. 21–33.
- Väänänen-Vainio-Mattila, K. 2011. Käytettävyys ja käyttäjakeskeinen suunnittelu Teoksessa Oulasvirta, A. (toim.). 2011. Ihmisen ja tietokoneen vuorovaikutus. Helsinki: Gaudeamus Helsinki University Press Oy Yliopistokustannus, HYY yhtymä. 102–126.
- Widgord, D. & Wixon, D. 2012. Brave NUI World: designing natural user interfaces for touch and gesture. Burlington, USA: Morgan Kaufmann Publishers, Elsevier.

## Ensimmäisen käytettävyydestin monivalintakyselyn tulokset

Negatiivisia kysymyksiä on kompensoitu. Nämä kohdat on merkitty tummalla.

	1	2	3	4	5	6	Vastaajat yhteensä –	Painotettu keskiarvo
Hahmon liikkuttaminen oli yksinkertaista	0,00 % 0	50,00 % 2	0,00 % 0	0,00 % 0	0,00 % 0	50,00 % 2	4	4
Tein useita virheitä reittiä piirtäessä	25,00 % 1	50,00 % 2	0,00 % 0	0,00 % 0	25,00 % 1	0,00 % 0	4	3,5
Hahmojen vaihtaminen oli helppoa	0,00 % 0	0,00 % 0	0,00 % 0	0,00 % 0	0,00 % 0	100,00 % 4	4	6
Reitin peruminen kesken piirtämisen oli yksinkertaista	25,00 % 1	0,00 % 0	50,00 % 2	25,00 % 1	0,00 % 0	0,00 % 0	4	2,75
Pikakelaus -ominaisuus tuntui hyödylliseltä	25,00 % 1	50,00 % 2	0,00 % 0	0,00 % 0	25,00 % 1	0,00 % 0	4	2,5
Reitin piirtäminen oli helppoa	0,00 % 0	0,00 % 0	25,00 % 1	25,00 % 1	0,00 % 0	50,00 % 2	4	5,5
Reitin perumispainikkeen sijainti oli hyvä	25,00 % 1	25,00 % 1	25,00 % 1	25,00 % 1	0,00 % 0	0,00 % 0	4	2,5
Action -painikkeet olivat oikeassa paikassa	0,00 % 0	0,00 % 0	0,00 % 0	0,00 % 0	50,00 % 2	50,00 % 2	4	5,5
Reitin piirtäminen oli hankalaa	50,00 % 2	0,00 % 0	25,00 % 1	0,00 % 0	0,00 % 0	25,00 % 1	4	3,25
Action -painikkeita tulisi muuttaa	0,00 % 0	25,00 % 1	25,00 % 1	50,00 % 2	0,00 % 0	0,00 % 0	4	2,75
Hahmon vaihtaminen aiheutti sekaannusta	50,00 % 2	50,00 % 2	0,00 % 0	0,00 % 0	0,00 % 0	0,00 % 0	4	4,5
Järjestelmä oli kokonaisuudessaan helppokäyttöinen	0,00 % 0	25,00 % 1	25,00 % 1	25,00 % 1	0,00 % 0	25,00 % 1	4	3,75
Olen tyytyväinen järjestelmään	0,00 % 0	50,00 % 2	25,00 % 1	0,00 % 0	25,00 % 1	0,00 % 0	4	3
							Kokonaiskeskiarvo	<b>3,81</b>

## Testitarina

Olet pitämässä taukoa työpaikallasi ja päätät pelata Parcelia iPadillasi.

- Aloita peli ja pelaa hetki
- Hetken päästä joku huomauttaa äänen voimakkuuden olevan liian kovalta. Ota äänet pois parhaaksi näkemälläsi tavalla.
- Jatkaessasi pelaamista huomaat, että kenttä jonka valitsit olikin väärä. Poistu takaisin kentänvalintaan ja valitse neljän kenttä.
- Pelaa hetki ja poistu päävalikkoon.
- Henkilö, joka aiemmin valitti äänenvoimakkuudesta, on poistunut huoneesta. Käännä äänenvoimakkuutta hieman kovemmalle.
- Äänenvoimakkuuden ollessa mieleiselläsi tasolla haluat jatkaa pelaamista. Valitse kenttä neljä uudestaan.
- Hetken päästä huomaat pelissä ongelmatilanteen. Aloita kenttä uudestaan.
- Huomiosi herpaantuu ja menet ruutuun, jossa peli päättyy tappioon. Eteenne haluamallasi tavalla.
- Pelaa kenttä läpi ja jatka seuraavaan kenttään.

## Toisen käytettävyydestin monivalintakyselyn tulokset

	1	2	3	4	5	6	Vastaajat yhteensä	Painotettu keskiarvo
Hahmon liikuttaminen oli yksinkertaista	0,00 % 0	0,00 % 0	0,00 % 0	40,00 % 2	20,00 % 1	40,00 % 2	5	5
Reitin piirtäminen oli miellyttävää	0,00 % 0	20,00 % 1	0,00 % 0	20,00 % 1	40,00 % 2	20,00 % 1	5	4,4
Reitin peruminen kesken piirtämisen oli yksinkertaista	20,00 % 1	0,00 % 0	0,00 % 0	20,00 % 1	40,00 % 2	20,00 % 1	5	4,2
Pikakelaus -ominaisuus tuntui hyödylliseltä	0,00 % 0	20,00 % 1	20,00 % 1	0,00 % 0	0,00 % 0	60,00 % 3	5	4,6
Reitin piirtäminen haluttuun sijaintiin toimi hyvin	0,00 % 0	0,00 % 0	20,00 % 1	40,00 % 2	20,00 % 1	20,00 % 1	5	4,4
Reitin perumispainikkeen sijainti oli hyvä	20,00 % 1	0,00 % 0	20,00 % 1	20,00 % 1	20,00 % 1	20,00 % 1	5	3,8
Action -painike olivat oikeassa paikassa	0,00 % 0	0,00 % 0	0,00 % 0	0,00 % 0	40,00 % 2	60,00 % 3	5	5,6
Laatikon vetäminen/työntäminen oli miellyttävää	0,00 % 0	20,00 % 1	0,00 % 0	40,00 % 2	20,00 % 1	20,00 % 1	5	4,4
Action -painike oli selkeä	0,00 % 0	0,00 % 0	0,00 % 0	0,00 % 0	20,00 % 1	80,00 % 4	5	5,8
Hahmon vaihtaminen tuntui yksinkertaiselta	0,00 % 0	0,00 % 0	0,00 % 0	20,00 % 1	0,00 % 0	80,00 % 4	5	5,6
Ohjainjärjestelmä oli kokonaisuudessaan helppokäyttöinen	0,00 % 0	0,00 % 0	0,00 % 0	40,00 % 2	20,00 % 1	40,00 % 2	5	5
Olen tyytyväinen ohjainjärjestelmään	0,00 % 0	0,00 % 0	20,00 % 1	20,00 % 1	40,00 % 2	20,00 % 1	5	4,6
							Kokonaiskeskiarvo	4,78