

Niklas Eurén

MOBIILIPELI

Tietojenkäsittelyn koulutusohjelma  
Ohjelmistosuunnittelun suuntautumisvaihtoehto  
2011

## MOBIILIPELI

Eurén, Niklas  
Satakunnan ammattikorkeakoulu  
Tietojenkäsittelyn koulutusohjelma  
Huhtikuu 2011  
Ohjaaja: Kyngäs, Jari  
Sivumäärä: 25

Asiasanat: mobiilipeli, peli, java me

---

Tämän opinnäytetyön tarkoituksena oli toteuttaa ajanvieteellinen mobiilipeli. Mobiilipeli tehtiin vain opinnäytetyötä varten.

Tavoitteena oli tutkia muita mobiilipelejä, mobiilipelien historiaa ja lopuksi saada aikaan ainutlaatuinen, mutta toteutukseltaan suhteellisen yksinkertainen peli. Tavoitteena oli myös se, että pelillä on uudelleenpeluuarvoa. Mobiilipeliin tulisi myös itse tehdyt grafiikat, äännet ja musiikit.

Opinnäytetyön tuloksena syntyi omalaatuinen mobiilipeli, jonka voi asentaa mobiililaitteeseen pelattavaksi.

Opinnäytetyö sisälsi pelisuunnittelua, ohjelmointia, äänitystä ja testausta.

## MOBILE GAME

Eurén, Niklas

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in Business Information Systems

April 2011

Supervisor: Kyngäs, Jari

Number of pages: 25

Keywords: mobile game, game, java me

---

The purpose of this thesis was to develop a recreational mobile game. The mobile game was made only for the thesis.

The aim was to explore other mobile games, mobile game history, and finally come up with unique, but relatively simple mobile game. The game had to have replay value. The aim was to do programming, graphics, sounds and music myself.

The outcome of the project was a unique mobile game which can be installed in a mobile device for playing.

The thesis contained game designing, programming and testing.

# SISÄLLYS

1	JOHDANTO .....	5
2	MOBIILIPELIEN HISTORIA.....	5
3	MOBIILIPELIN TOTEUTUS .....	7
3.1	Määrittely .....	7
3.2	Suunnittelu.....	9
3.2.1	Luokkasuunnittelu .....	10
3.2.2	Mobiililaitteiden asettamat rajoitukset.....	11
3.3	Ohjelmointi.....	13
4	GRAFIIKAT JA ÄÄNET.....	16
4.1	Alustavat grafiikat.....	16
4.2	Grafiikoiden viimeistely.....	16
4.3	Pelimusiikin ja äänien säveltäminen .....	17
5	VALMIS PELI.....	17
5.1	Tahmaajahai.....	17
5.2	Pelin puutteet .....	19
6	YHTEENVETO .....	21
	LÄHTEET .....	24

## 1 JOHDANTO

Tämän opinnäytetyön tekijän kiinnostus pelien tekemiseen alkoi noin 11 vuotta sitten, jolloin tuli tutuksi pelinteko-ohjelma Game Maker. Vähitellen syntyi yhä monimutkaisempia tietokonepelejä ja jossain vaiheessa pelien tekemisestä oli tullut harrastus. Pelejä tehtiin niin omaan käyttöön kuin myös vapaasti maailmanlaajuiseen jakoon. Pelintekijä hyötyi tästä saadessaan arvokasta kokemusta ja palautetta pelaajilta.

Mobiilipelien tekeminen tuli tekijälle tutuksi Satakunnan ammattikorkeakoulussa tietojenkäsittelyn koulutusohjelmassa Java ME – kurssilla. Tekijällä on myös kokemusta mobiilipelien pelaamisesta. Mobiilipeliksi kutsutaan peliä, jota voidaan pelata pienellä kannettavalla mobiililaitteella. Tässä opinnäytetyössä mobiililaitteeksi ei lueta käsikonsoleita. Mobiilipeli kulkee kätevästi mobiililaitteen, esimerkiksi matkapuhelimen, mukana ja on yleensä tarkoitettu ajanvietteeksi.

Tämän opinnäytetyön tekijän oli luontevaa lähteä tekemään opinnäytetyönä mobiilipeliä ottaen huomioon edellä kerrotun tekijän taustan.

## 2 MOBIILIPELIEN HISTORIA

Jotta voidaan paremmin ymmärtää opinnäytetyönä toteutettua mobiilipeliä, esitellään lyhyesti ensin videopelien historiaa.

Videopelien historia ulottuu 1940-luvun loppuun, jolloin liikkuvaa kuvaa näyttäviä kuvaputkia alettiin käyttää ohjustorjuntajärjestelmissä. Näiden järjestelmien pohjalta alkoi 1950-luvulla syntyä yksinkertaisia pelejä, ja 50-luvun lopulla ja 1960-luvulla videopelejä voitiin pelata jo suurtietokoneilla. Pelejä syntyi lisää nopeasti kasvavalla tahdilla ja pian niitä tehtiin jo pelihalleihin, konsoleille, PC:lle ja myöhemmin käsikonsoleille (Wikipedia/a 2010).

Käsi­konsoleilla pelattavia pelejä ei yleensä kutsuta mobiilipeleiksi, joten varsinaisten mobiilipelien historia alkaa vuodesta 1997. Silloin ensimmäinen yleiseen levitykseen päässyt mobiilipeli Snake näki päivänvalon Nokian matkapuhelimissa (Pelkonen 2005, 110; Wikipedia 2010/a). Tämä yksinkertainen matopeli sai todella paljon suosiota ja Nokian matkapuhelimet olivatkin tunnettuja juuri tuosta pelistä. Alun perin Snake oli kiinteästi asennettuna matkapuhelimiin, mutta sittemmin siitä ja sen uusista versioista on tullut erikseen ladattavia.

1990-luvun lopulla japanilaiset menestyivät I-Mode –verkossa pelattavien pelien avulla ja se synnytti villityksen WAP-pelejä kohtaan. I-Mode on japanilainen mobiili-internet. WAP-pelejä voitiin pelata käyttäen apuna matkapuhelimen WAP-selainta. Pelit olivat staattisia ja ne koostuivat vain kuvista, tekstistä, hyperlinkeistä ja tekstikentistä. Suurin osa niistä oli tekstiseikkailuja. Nokian 7100-sarjan matkapuhelimet olivat ensimmäisiä Euroopassa, jotka käyttivät tuota ominaisuutta hyväkseen. WAP-pelit saivat kuitenkin huonon maineen, koska niiden yhteysnopeus oli hidas. Tämä johti siihen, että ne eivät koskaan nousseet markkinoilla korkealle (Pelkonen 2005, 111; Costikyan 2002).

Mobiilipelit pääsivät massamarkkinoille vuonna 2001, kun esiteltiin ensimmäiset ladattavat mobiilipelit ja matkapuhelinten värinäytöt (Pelkonen 2005, 111). Pelimaailma mullistui ja pian kuluttajat olivat valmiita ostamaan mobiilipelejä laajentaakseen pelivalikoimaansa.

Nokia yritti luoda oman mobiilipelialustansa N-Gagen avulla 2003, mutta tämä yritys epäonnistui pääasiassa sen vuoksi että tuohon aikaan pelialustaa ei onnistuttu yhdistämään tarpeeksi hyvin varsinaiseen matkapuhelimeen. Laite oli kömpelö puhumiseen ja laitteistossa ilmeni myös joitain vikoja (Wikipedia 2010/b).

### 3 MOBIILIPELIN TOTEUTUS

Tämän opinnäytetyön tarkoituksena on toteuttaa ajanvietteellinen mobiilipeli. Mobiilipeli tehdään vain opinnäytetyötä varten.

Tavoitteena on tutustua muihin mobiilipeleihin ja mobiilipelien historiaan. Nämä tiedot ovat apuna mobiilipelin toteutuksessa. Tavoitteena on tehdä mobiilipelistä ainutlaatuinen, mutta toteutukseltaan yksinkertainen. Tavoitteeksi on myös asetettu mobiilipelin grafiikoiden, äänien sekä musiikin tekeminen.

#### 3.1 Määrittely

Määrittelemisen asettaa pelille rajat ja tarkentaa pelintekijälle syyn pelin tekemiseen. Perusteellinen määrittely myös erottaa harrastusmielessä tehdyt pelit liiketoimintaan suunnatuista peleistä. Liikemaailmassa pelifirmat odottavat saavansa pelinkehittäjiltään laadukkaita ja uudenaikaisia pelejä. Mobiililaitteet kehittyvät nopeasti, joten pelinkehittäjän on oltava tietoinen uudistuksista. Tavoitteena on usein olla ensimmäinen, eli kehitettävä uudenlainen pelikonsepti tai jotain vastaavaa, koska jos et itse keksi, joku muu keksii. (IGDA 2005, 31)

Syy voisi jossain tapauksessa olla myös mm. pelin tekeminen yritykselle ja sitä kautta palkan saaminen, pelin tekeminen suoraan myyntiin tai pelin tekeminen harrastusmielessä itselle tai pienelle kohdeyleisölle. Usein pelien tekeminen aloitetaan harrastusmielessä ja myöhemmin yritetään saada harrastuksesta ammattia. (The Game Development Wiki 2010)

Tämän mobiilipelin toteutuksessa lähdetään määrittelystä liikkeelle. Määritellään ensin pääpiirteittäin mitä pitää tehdä, millä ohjelmointikielellä, mille alustalle, miksi ja kenelle työ tehdään. Mobiilipeli tehdään vain opinnäytetyötä varten ja se syventää tekijän opinnoista saatuja tietoja.

Pelille on tärkeää asettaa rajat, koska usein on tiedettävä miten suuresta projektista on kyse ja mitä resursseja vaaditaan. On selvitettävä mahdolliset sudenkuopat, eli

mihin projekti voi kaatua. On päätettävä tarvittavan henkilöstön määrä, eli esim. tarvitaanko pelin tekemiseen erikseen ohjelmoija, suunnittelija, taiteilija ja muusikko/äänimies. Pitää määritellä kohdealustat, joille peli tullaan tekemään ja käytettävät ohjelmointikielet jotka usein määrittyvät juuri kohdealustan mukaan. Tärkein on kuitenkin vielä määrittelemättä, eli kohderyhmä. Varsinkin voittoa tavoittelevissa projekteissa on hyvin tärkeää määritellä minkä ikäiset pelaajat tulisivat pelaamaan peliä. Voidaan myös päätellä mitä pelaajat haluavat peliltä. (IGDA 2005, 31-32)

Tässä tapauksessa mobiilipelille asetetaan rajat, jotka tiedetään saavutettaviksi, mutta jotka eivät päästä tekijää liian helpolla. Voitettavissa olevat haasteet lisäävät motivaatiota pelin tekemiselle. Ideaalitapauksessa mobiilipelin tekemisessä ei olisi aikarajaa, tai aikaa olisi runsaasti, mutta koska kyseessä on opinnäytetyö, sen valmistumiselle annetaan aikaa puoli vuotta. Mobiilipelin tulee olla vähintään Snaken, Pac-Manin, Tetriksen ja Space Invadersin tasoinen yleisesti ottaen. Kyseiset pelit ovat ulkoasultaan yksinkertaisia, ja pelialue on mobiililaitteen ruudun kokoinen. Ne ovat kuitenkin koukuttavia pelejä, joiden uudelleenpeluuarvo on suuri.

Opinnäytetyönä tullaan tekemään ylhäältäpäin kuvattu omaperäinen 2D-peli, jossa kaikki tapahtuu pienellä pelialueella. Lisähaasteita tuo se, että ohjelmoinnin lisäksi opinnäytetyön tekijä tulee myös itse piirtämään grafiikat ja äänittämään musiikit ja äänitehosteet. Haasteita puolestaan vähentää se, että peliä varten ei välttämättä tarvitse kehittää mitään uutta toiminnallisuutta, vaan kaikki pelissä tapahtuva toiminnallisuus on jo aikaisemmin kehitetty ja dokumentoitu. Tähän kuuluu mm. pelaajan liikkuminen ja pelisilmukan (selitetään tarkemmin kappaleessa 3.3) koodaaminen, joihin löytyy runsaasti esimerkkimateriaalia internetistä ja kirjoista. Pelintekijän tehtäväksi jääkin pääasiassa materiaalin etsiminen, kokoaminen ja soveltaminen tekeillä olevaan projektiin. Se ei tee silti pelin tekemisestä helppoa, koska koodia pitää joka tapauksessa ymmärtää ja kirjoittaa itse.

Kohdealustana on Nokia E51 - matkapuhelin. Kyseinen matkapuhelin tukee Java ME:tä, ohjelmointikieltä jolla peli tullaan tekemään. Java ME valittiin, koska se oli tuttu opinnoista ja se soveltui parhaiten yksinkertaisen mobiilipelin tekemiseen. Peli



ei myöskään tule olemaan niin monimutkainen, että siihen tarvittaisiin tehokkaampaa ja suurempiin saavutuksiin pystyvää ohjelmointikieltä.

### 3.2 Suunnittelu

Moni ohjelmoija osaa tehdä pelin, mutta hyvän pelin tekemiseen tarvitaan muutakin kuin pelkkää ohjelmoimista. Hyvä peli syntyy hyvän suunnittelun kautta. Suunnitteluvaiheessa määrittelyvaiheen visiot heräävät eloon, ainakin paperilla. Paperi voikin olla yksi tärkeimpiä työvälineitä kun suunnitellaan pelejä. Paperille saa nopeasti ja kätevästi hahmoteltua pelimaailmaa ja kirjoitettua huomautuksia. Paperilla voi selvittää mm. matemaattisia kaavoja, kenttien rakenteita, isoja koodiryppäitä ja pelihahmojen ulkonäköä.

Suunnittelussa on hyvä tehdä aikataulusuunnitelma mobiilipelin toteutusta varten. Aikataulusuunnitelma sisältää tehtävät, niiden suoritusjärjestyksen ja tehtäville varatun ajan. Varsinaiselle pelimoottorille annettiin eniten aikaa, koska se on pohja jonka päälle aletaan koota pelin muita osia. Pelimoottorin on oltava helposti hallittava, jotta muiden osien lisäys onnistuu mahdollisimman sujuvasti. Pelimoottorista kerrotaan lisää luvussa 3.3. Muita aikataulusuunnitelmaan kuuluvia kohtia olivat mm. grafiikat ja tekoäly.

Pelintekijän tulisi laittaa itsensä sekä pelaajan että mahdollisen pelifirman (tai vastaavan) asemaan. Tämä auttaa ymmärtämään mitä peliltä oikeasti halutaan. Pelintekijällä voi olla nimittäin liian erilaiset visiot hittipelistä kuin esim. firmalla joka on hänet palkannut. Pelejä on paljon, ja jotenkin pitäisi erottua massasta jotta pelillä olisi kysyntää. Parhaiten erottuu massasta kun kehittää kokonaan uuden konseptin, mutta usein riittää kun kehittää jotain näkyvää uutta jo valmiiseen konseptiin. (IGDA 2005, 32)

Tässä vaiheessa opinnäytetyönä syntyvälle mobiilipelille kehitettiin idea. Mobiilipeli on ylhäältäpäin kuvattu, ja kaikki toiminta tapahtuu kiinteällä pelialueella, joka on matkapuhelimen ruudun kokoinen. Pientä pelialuetta on helpompi hallita ohjelmoijan näkökulmasta, varsinkin kun opinnäytetyön tekijällä ei ole paljon kokemusta

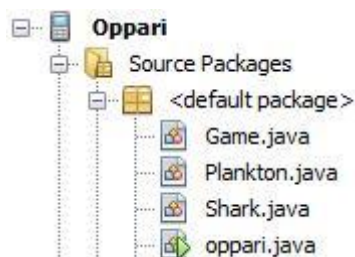
mobiiliohjelmoinnista. Mobiilipelin idea on omalaatuinen. Tarkoituksena oli kehittää idea, joka on erikoinen mutta samalla toimiva. Tapahtumat sijoittuvat merelle, jossa pelihahmon avulla kerätään planktonia ruuaksi ja uidaan haita pakoon. Plankton antaa voimia uimiseen, joten keräily ei pelkästään näy pisteiden kasvamisena. Planktonit liikkuvat mobiililaitteen ruudulla sattumanvaraisesti. Niitä on koko ajan olemassa kolme kappaletta, ja kun yksi syödään, uusi syntyy tilalle uuteen paikkaan. Vaikka pelialue on ruudun kokoinen, siinä ei silti ole seiniä, vaan ruudun reunat ovat yhteydessä toisiinsa. Tämä tarkoittaa sitä, että esim. jos plankton ui ruudun oikeasta laidasta läpi, se ilmestyy ruudun vasemmasta laidasta.

Haita on myös kolme, ja ne seuraavat pelaajaa välillä vaihtaen suuntaa. Hait myös vaihtelevat uintinopeuttaan. Tämä antaa pelaajalle mielikuvan itsenäisesti ajattelevista haista, jotka jahtaavat pelaajaa vain halutessaan. Tekoälyn tai näennäisen tekoälyn luominen on tässä mobiilipelissä olennaista, koska se luo peliin sattumanvaraisuutta ja sattumanvaraisuus luo uudelleenpeluuarvoa. Tekoäly tarkoittaa toiminnallisuutta, joka vaikuttaa järkevältä. Videopelien pelaajat ovat tottuneet näkemään vähintään alkeellista tekoälyä peleissä, joten esim. hait jotka suoraan seuraisivat pelaajan liikkeitä, eivät vaikuttaisi kovin mielenkiintoisilta. Kun hai osuu pelaajaan, pelaaja menettää voimia. Voimien loppuessa pelihahmo kuolee ja peli päättyy. Tämä voi siis tapahtua myös silloin, jos pelaaja jättää planktonia syömättä. Mitä kauemmin pelaaja pystyy pitämään pelihahmon elossa, sitä paremmat mahdollisuudet hänellä on päästä pisteille. Pistetaulukko tallentuu mobiililaitteen muistiin. Parhaiden pisteiden tavoittelu tuo myös uudelleenpeluuarvoa.

### 3.2.1 Luokkasuunnittelu

Pelin idean ja toiminnallisuuden selvittämisen jälkeen kehitettiin alustava suunnitelma pelissä käytettävistä luokista. Luokat luotiin heti alussa Netbeans-projektiin (Netbeansista tarkemmin luvussa 3.3). Tässä vaiheessa ne olivat vielä tyhjiä, mutta niihin pystyi luomisen ansiosta jo viittamaan koodissa. Opinnäytetyönä syntyvän mobiilipelin yksinkertaisesta rakenteesta johtuen luokkia ei ole paljon, ja niiden väliset suhteet olivatkin melkein heti selvillä. Tarkoituksena oli tehdä ohjelmoinnista mahdollisimman helppoa ja rakenteellisesti yksinkertaista

käyttämällä osittain apuna opinnäytetyön tekijän aikaisempia kokemuksia pelien tekemisestä. Opinnäytetyön tekijällä on laaja kokemus gml-ohjelmointikielen käytöstä Game Maker – ohjelmointiympäristössä, joten mobiilipelin tekemistä Netbeansin ja Java ME:n avulla peilattiin jatkuvasti Game Makerin ja gml:n käyttämiseen. Kuvassa 1 näkyy tässä mobiilipelissä käytetyt luokat.



Kuva 1. Opinnäytetyönä tehtävän mobiilipelin luokat

- oppari.java on luokka, joka näyttää mobiilipelin aloitusruudun ja sen jälkeen käynnistää varsinaisen pelin luomalla olion luokasta Game.java. Muuta tärkeää toiminnallisuutta tässä luokassa ei ole.
- Game.java on luokka, joka luo ja kontrolloi muita luokkia, luo pelisilmukan ja vastaa pelihahmon liikuttamisesta. Näiden lisäksi myös ruudun päivittäminen, eli mm. grafiikan näyttäminen oikealla paikalla oikeaan aikaan kuuluu tämän luokan tehtäviin.
- Shark.java on luokka, joka kontrolloi haita ja vastaa niiden tekoälystä.
- Plankton.java on luokka, joka kontrolloi planktoneita.

### 3.2.2 Mobiililaitteiden asettamat rajoitukset

Pelinkehittäjän näkökulmasta tietyt mobiililaitteiden asettamista rajoituksista nousevat muiden yläpuolelle ja näihin on hyvä kiinnittää huomiota mobiilipeliä suunnitellessa ja rakentaessa. Crooksin (2005, 11-13) mukaan näihin kuuluvat

#### 1. pieni näyttö

Mobiililaitteiden näyttö on pieni verrattuna tavallisten tietokoneiden näyttöihin. Tämä voi olla rajoittava tekijä varsinkin niille pelinkehittäjille, jotka ovat tottuneet kehittämään pelejä esim. pöytäkoneille. Pienellä näytöllä ei voi eikä kannatakaan esittää niin paljon tietoa kuin isommalla näytöllä,

joten grafiikoita on karsittava ja niiden suhteen on oltava luova. Tämän voi huomata esim. valikoissa, joissa ei saa olla liikaa valittavaa.

2. akun kesto

Näyttö kuluttaa akkua paljon. Suurin osa mobiililaitteiden käyttäjistä käyttää laitteitaan muuhun tarkoitukseen kuin pelaamiseen, jonka vuoksi ei ole suotavaa että pelit kuluttaisivat akun loppuun.

3. näppäimistö

Jotta mobiililaitteet olisivat mahdollisimman pieniä, suurin osa laitteiden valmistajista jättää tietokonemaisen näppäimistön pois. Tämän vuoksi pelaajaa ei kannata pakottaa syöttämään suuria tekstimääriä.

4. prosessointiteho

Mobiililaitteissa on luonnollisesti pienen kokonsa vuoksi pienemmät prosessorit kuin esim. pöytäkoneissa. Tämän vuoksi peliohjelmoijan olisi hyvä tehdä koodistaan mahdollisimman suorituskykyistä.

5. rajallinen muisti ja tallennuskapasiteetti

Useimmissa mobiililaitteissa ei ole kovinkaan paljon muistia ja tallennuskapasiteettia käytettävänä. Lisämuistia voidaan kuitenkin laitteisiin hankkia, mutta koska näin ei aina tehdä, on pelintekijän vastuulla optimoida pelit pieniksi ja vähän tallennuskapasiteettia kuluttaviksi.

6. multimedian käyttö ja kaiuttimet

Mobiilipelien kehittäjän on muistettava, että mobiililaitteita käytetään usein julkisissa tiloissa, joten peleissä on oltava mahdollisuus saada äänet pois päältä.

7. internetyhteys

Vaikka useat mobiililaitteet tarjoavatkin nettiyhteyden, sitä ei voi pelinkehittämisessä olettaa. Tämän lisäksi nettiyhteyttä ei välttämättä käytetä esim. hinnan takia vaikka se muuten olisi mahdollista.

Opinnäytetyön tekijällä on myös omakohtaista kokemusta mobiilipelien pelaamisesta. Pelikokemuksen myötä hänelle on tarkentunut kuva siitä, mitä mobiilipelien tekemisessä kannattaa ottaa huomioon. Yllä oleva lista on hyvä muistilista ja sen kohdat kannattaa pitää mielessä peliä suunniteltaessa.

Opinnäytetyön tekijä on mobiilipelejä pelatessaan huomannut, että varsinkin näppäimiin kannattaa kiinnittää huomiota. Hänellä itsellään on käytössä kosketusnäyttöinen matkapuhelin, jossa ei ole kontrollipainikkeita. On olemassa paljon pelejä, joissa ei pääse alkuvalikkoa pidemmälle, koska matkapuhelimella ei voi valita painikkeilla kyllä/ei. Useimmiten näissä kohdissa kysytään, että käytetäänkö pelissä ääniä vai ei. Jos siis haluaa mahdollisimman suuren kohderyhmän pelille, kannattaa ottaa huomioon tällaisetkin tilanteet.

### 3.3 Ohjelmointi

Peliohjelmoinnissa avainasemassa on pelisilmukka (game loop) (Wikipedia/c 2010). Se saa pelin etenemään riippumatta pelaajan antamista käskyistä tai niiden puuttumisesta. Kuvassa 2 näkyy, miltä pelisilmukka voi pseudokoodina (yksinkertaistettuna koodina) näyttää.

```
while(pelaaja ei poistu)
{
    tarkista mitä pelissä tapahtuu
    tarkista pelaajan antamat käskyt
    päivitä näyttö
}
```

Kuva 2. Pelisilmukka pseudokoodina

Pelisilmukassa voidaan mm. tarkistaa se, että osuuko pelihahmo viholliseen, haluaako pelaaja liikuttaa pelihahmoa ja mitä grafiikoita näytölle pitää piirtää. Kuvassa 3 näkyy opinnäytetyönä tehdyssä mobiilipelissä käytetty pelisilmukka.

```

public void run()
{
    //Pelilooppi
    while(true)
    {
        //Mitä pelissä tapahtuu
        verifyGameState();

        //Mitä käyttäjä tekee
        checkUserInput();

        //Päivitetään näyttö
        updateGameScreen(getGraphics());

        try
        {
            //Miten usein näyttöä päivitetään
            Thread.currentThread().sleep(30);
        }
        catch(Exception ex)
        {
            System.out.println("Virhe runissa: " + ex);
        }
    }
}
}

```

Kuva 3. Pelisilmukka kommentteineen

Kuvassa 3 on huomioitavaa myös kommenttien käyttö koodin joukossa. Kommentointi auttaa koodin ymmärtämisessä niin peliä rakentaessa, tarkasteltaessa koodia myöhemmin tai jos koodia lukee joku ulkopuolinen. Kuvasta näkyy myös ohjelmointityyli, joka on ohjelmoijakohtainen.

Koska tämä mobiilipeli tehdään Java ME:llä, ohjelmointitapana käytetään olio-ohjelmointia. Olio on ohjelmiston perusyksikkö, joka koostuu loogisesti yhteenkuuluvasta tiedosta ja toiminnallisuudesta. Oliot kommunikoivat keskenään lähettämällä ja vastaanottamalla viestejä. Olio-ohjelmoinnissa ohjelma muodostuu yhteistyössä toimivista olioista. Tämä ohjelmointitapa antaa ohjelmoijalle selkeän ja organisoidun tavan kontrolloida koodeja. On ensisijaisen tärkeää tietää mitä tapahtuu, missä ja miksi. Tämä on olennaisessa osassa virheiden paikannuksessa ja ongelmanratkonnassa, kuten myös luonnollisesti uuden koodin kirjoittamisessa. Jos

ei pysty hahmottamaan ohjelmointirakennetta, ei pysty hallitsemaan ohjelmoimista ja lopputuloksena on epäselvää koodia.

Pelin tekeminen ei välttämättä vaadi täydellisiä ohjelmointitaitoja, mutta tavallisesti koodia on luotava huomattavia määriä. Ohjelmointi onkin luontainen osa pelituotantoa. Kaikkein graafisimmatkin pelimoottorit vaativat loogisen rakenteen ja toimintamekaniikan taustalleen. (Manninen 2007, 47-48)

Tämän mobiilipelin ohjelmointia varten tarvittiin ohjelmointiympäristö ja ohjelmointikieli. Ohjelmointiympäristöksi valittiin NetBeans IDE ja ohjelmointikieleksi Java ME. NetBeans IDE on lähdekoodiltaan avoin ohjelmointiympäristö ohjelmistonkehittäjille. Se tarjoaa kaikki tarvittavat työkalut huipputasoisten työpöytäsovellusten, yritysohjelmistojen, web-sovellusten ja mobiilisovellusten luomiseen Javan (ja sen eri muotojen), C/C++:n, PHP:n, JavaScriptin, Groovyn tai Rubyn avulla (NetBeans.org 2010). NetBeans IDE avustaa ohjelmoinnissa paljon ja tarjoaa selkeän käyttöliittymän ohjelmistontekijän käyttöön. Kaikki luokat näkyvät selkeästi ja ohjelmointiympäristö ilmoittaa yleensä ymmärrettävästi koodauksessa tapahtuneista virheistä. Välillä tosin tulee vastaan ongelmia, joista on vaikea selviytyä epäselvien virheilmoitusten vuoksi. Mobiilipelin tekeminen aloitettiin NetBeans IDE:n versiolla 6.7 ja lopetettiin versiolla 6.9.1.

Java ME eli Java Platform, Micro Edition, on sulautettuja järjestelmiä (sis. mobiililaitteet) varten suunniteltu ohjelmointikieli (Wikipedia/d 2011). Ohjelmoinnissa käytettiin apuna koulussa saatuja oppeja varsinkin Java ME –kurssilta. Ohjelmointi eteni pääasiassa niin, että asetettiin tavoite joka päivän aikana piti toteuttaa ja sen jälkeen pyrittiin toteuttamaan se. Pelin toiminnallisuus ja ulkonäkö oli suurimmaksi osaksi jo suunniteltu, joten ohjelmointi oli tie jolla tavoitteeseen päästään. Suurimpiin ohjelmoinnin ongelmiin kuului pelimoottorin suunnitteleminen ja rakentaminen.

Pelimoottori on pelin taustalla vaikuttava voima, joka saa kaiken toimimaan. Tässä mobiilipelissä pelimoottori koostuu pääasiassa pelisilmukasta joka luo toiminnallisuutta koko ajan.

## 4 GRAFIIKAT JA ÄÄNET

### 4.1 Alustavat grafiikat

Peliä tehdessä kannattaa välttää ajan käyttämistä viimeistelyjen grafiikoiden tekemiseen, koska siitä harvoin on hyötyä ja vaatimukset grafiikoiden suhteen saattavat muuttua kesken projektin. Grafiikat saattavat inspiroida ohjelmoijaa, mutta niiden hiominen kannattaa toteuttaa vasta ohjelmointivaiheen päätyttyä. Heti ensimmäisten koodien synnyttyä on hyvä piirtää yksinkertaiset pelkistetyt grafiikat jotta koodia pystyy testaamaan käytännössä visuaalisesti.

Alustavien grafiikoiden tekemiseen käytetään Sib Icon Editoria siitä saatujen aikaisempien hyvien kokemusten vuoksi. Se tarjoaa hyvät työkalut pikseligrafiikan luomiseen, ja ainakaan alustavilta grafiikoilta ei enempää kannata vaatiakaan. Pikseligrafiikka sopii pelin tyyliin ja mobiililaitteen pieneen näyttöön. Opinnäytetyön tekijällä on myös aikaisempaa kokemusta pikseligrafiikan tekemisestä. Kyseinen graafinen tyyli keskittyy pienikokoisiin grafiikoihin, ja tällaiset grafiikat piirretään yleensä pikseli kerrallaan.

### 4.2 Grafiikoiden viimeistely

Alustavat grafiikat olivat jo visuaalisesti hyviä, mutta ne vaativat pientä hiomista. Tähän tekijä otti apuun Game Makerin sprite-editorin. Pääsyy ohjelman vaihtoon on se, että aikaisemmin käytetty Sib Icon Editor muuttui maksulliseksi eikä opinnäytetyön tekijällä ollut intressejä maksaa sen käyttämisestä. Game Makerin sprite-editori on yksinkertainen mutta toimiva pikseligrafiikkaa tuottava työkalu. Game Maker itsessään on työkalu tietokonepelien tekemiseen.

Grafiikoiden viimeistelyssä kiinnitettiin huomiota pääasiassa animaatioihin ja väreihin. Pelaajan hahmolle, haille ja planktonille tehtiin pienet animaatiot. Ne siis laitettiin liikehtimään kuvanvaihtojen avulla. Animaatiot saivat pelin elävänoloiseksi. Animaatiot toteutettiin yksinkertaisesti, eli käyttämällä jokaisessa animaation



ruudussa samaa kuvaa vähän muokattuna. Animaatiografiikoiden tekemisen jälkeen animaation toimiminen pelissä toteutettiin ohjelmoimalla. Tässä vaiheessa tehtiin myös pelin aloitusruutu, jossa on sarjakuvamainen kuva haista suu auki ja pelin nimi.

#### 4.3 Pelimusiikin ja äänien säveltäminen

Pelinkehittäjän on muistettava, että mobiililaitteita käytetään usein julkisissa tiloissa, joten peleissä on oltava mahdollisuus saada äänet pois päältä. Toinen vaihtoehto on käyttää peleissä mahdollisimman vähän ääniä. Äänet myös kuluttavat nopeasti akkua, joten yleinen muistisääntö on pitää mobiilipelin äänimaailma mahdollisimman yksinkertaisena. (Crooks II 2005, 12)

Opinnäytetyönä syntyvän mobiilipelin yksinkertaisuudesta johtuen varsinaiset äänitehosteet jätettiin kokonaan pois. Taustamusiikiksi äänitettiin akustisella kitaralla tekijän itse säveltämä lyhyt yksinkertainen blues-riffi, joka laitettiin ”looppaamaan”, eli alkamaan uudelleen alusta aina musiikin loputtua. Äänityksessä käytettiin korvamikrofonia ja äänitysohjelma Audacityä. Musiikin lyhyen keston ansiosta musiikkitiedostosta ei tullut kovin suurta, joten se ei vie liikaa tilaa mobiililaitteesta. Musiikilla ei pelissä ole tärkeää roolia.

## 5 VALMIS PELI

### 5.1 Tahmaajahai

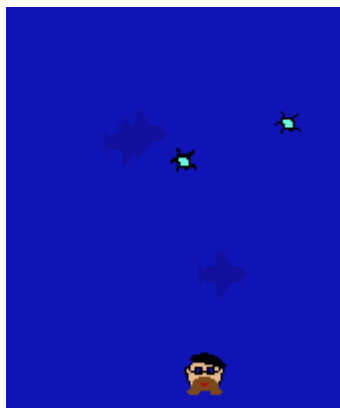
Opinnäytetyönä syntynyt mobiilipeli sai nimekseen Tahmaajahai. Se on ylhäältäpäin kuvattu humoristinen toimintapeli, jossa ideana on uida haita pakoon ja syödä planktonia. Pelialustana toimii Nokia E51 - matkapuhelin. Pelihahmoa ohjataan joko nuolinäppäimillä tai numeropainikkeilla.

Kuvassa 4 näkyy pelin aloitusruutu. Tahmaajahai nimenä viittaa tunnettuun elokuvaan Tappajahai.



Kuva 4. Pelin aloitusruutu

Kuvassa 5 näkyy kuvankaappaus pelistä. Hait näkyvät varjoina vedessä ja ne liikkuvat kohti pelihahmoa saadessaan vainun siitä. Vaaleansiniset pallukat ovat planktonia, ja ne liikkuvat sattumanvaraisesti pelialueella. Pelaaja voi syödä niitä liikkumalla niiden päälle. Peliruutu on pelkistetty, koska pienelle näytölle ei mahdu paljon grafiikkaa.



Kuva 5. Kuvankaappaus pelistä. Hait jahtaavat planktoninnälkäistä pelaajaa

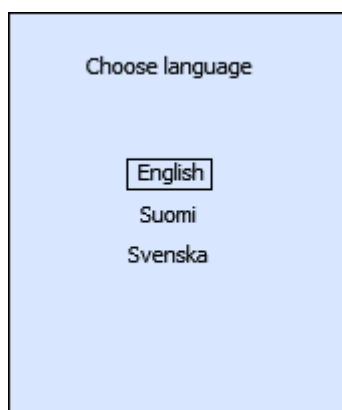
Tunnelman luominen pelissä onnistui hyvin. Pelin nopeatempoisuuden ansiosta pelaajalla ei ole aikaa jäädä miettimään seuraavaa siirtoaan, vaan hänen on liikuttava haita pakoon ja samaan aikaan jahdattava planktoneita. Haiden tekoäly onnistui myös odotetusti, ja niiden liikkuminen vaikuttaa oma-aloitteiselta. Pelissä ei ole pisteytysjärjestelmää, joten pelissä ei ole myöskään loppua. Pelaaja voi jahdata planktoneita ja paeta haita niin kauan kuin pystyy ja jaksaa.

## 5.2 Pelin puutteet

Pelin valmistumisen kanssa tuli ulkoisia ongelmia, joten siitä jäi pois pisteytysjärjestelmä, mutta peliä voi muuten pelata normaalisti. Pisteytysjärjestelmän puuttumisen takia pelin uudelleenpeluarvo ei ole kovin korkea joten siltä osin opinnäytetyön alussa asetetut tavoitteet eivät täysin toteutuneet. Äännet jäivät myös pois, mutta niiden pois jättämisellä ei ollut suurta merkitystä näin pienessä pelissä.

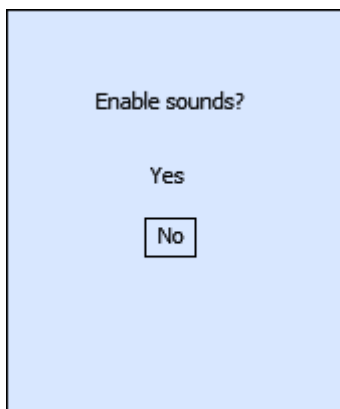
Vaikka kyseessä on pieni peli, pelissä olisi silti saanut olla alkukysymykset ja alkuvalikko. Nämä jäivät suunnitelmista pois, mutta jälkikäteen ajateltuna peli vaikuttaisi kaupallisemmalta ja sitä kautta valmiimmalta näiden kanssa.

Pelin nimi on suomenkielinen, mutta pelissä voisi olla silti mahdollisuus vaihtaa käytettävä kieli. Tämän pitäisi tapahtua heti pelin alussa, ensimmäisessä ruudussa, kuvan 6 mukaisesti.



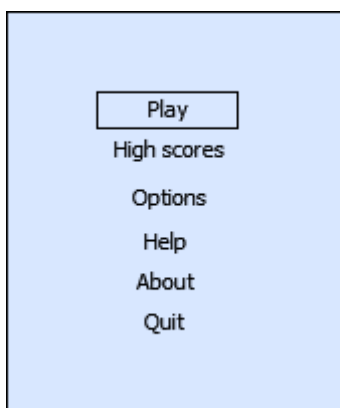
Kuva 6. Kielen valinta.

Koska englanti on yleisesti käytössä oleva kieli, myös pelissä se olisi voinut olla oletuskielenä. Tällaisessa pienessä pelissä eri kielivaihtoehtojen lisäys on suhteellisen helppoa, koska kovin paljon kirjoitettavaa ei ole. Jos peliin olisi tehty ääniä, alussa olisi voitu kysyä myös äänien käytöstä kuvan 7 mukaisesti.



Kuva 7. Äänet käytössä/pois käytöstä.

Äänien kannattaa olla oletusarvoisesti pois päältä, koska mobiilipelejä voidaan pelata tilanteissa joissa ääniä ei haluaisi kuulla. Kuvassa 8 näkyy, mitä vaihtoehtoja alkuvalikossa olisi voinut olla.



Kuva 8. Alkuvalikko.

Kohdasta *High scores* pelaaja voisi tarkastella parhaita tuloksia. Tulosikkunassa olisi näkyvissä esim. 10 parasta tulosta. *Options* – kohdasta pelaaja pääsisi hallitsemaan pelin asetuksia. Näihin voisi kuulua esim. vaikeustaso ja äänien käyttö. Jokaisessa pelissä pitäisi myös aina olla ohjeet, oli peli miten yksinkertainen tahansa. Ohjeet avautuisivat kohdasta *Help*. Ohjeen pitäisi ainakin selvittää, mitä pelissä pitäisi tehdä ja miten, eli idea ja kontrollit. Ohjeissa ei saa olla paljon tekstiä tai selattavaa, jotta pelaaja jaksaa lukea ne tarvittaessa. Kohdasta *About* pelaaja pääsisi näkemään pelin ja sen tekijän tietoja, esim. tekijänoikeudet, pelin versionumeron ja vaikkapa

sähköpostiosoitteen palautetta varten. Kohdasta *Play* pelaaja pääsisi pelaamaan ja kohdasta *Quit* pelaaja voisi poistua pelistä.

Valikkojen selailu olisi tehty mahdollisimman yksinkertaiseksi. Jos valikot olisivat olleet yllä olevien kuvien mukaisia, halutun kohdan voisi valita joko ylös/alas - painikkeilla, 2/8 – numeronäppäimillä tai liikuttamalla sormea/kynää näytöllä ylös tai alas. Oikean kohdan valinnan jälkeen eteenpäin pääsisi painamalla joko hyväksymispainiketta, 5 – numeronäppäintä tai painamalla sormella ruutua halutun vaihtoehdon kohdalta.

## 6 YHTEENVETO

Mobiilipelin tekeminen opinnäytetyönä oli minulle luonteva päätös, kun aloin toisena vuonna ensimmäisen kerran pohtimaan opinnäytetyöni aiheita. Se oli ensimmäinen aihe, joka tuli ja jäi mieleeni. Koska pelien tekeminen oli ollut harrastukseni kymmenen vuoden ajan, oli päätös helppo. Pelien tekeminen prosessina on siis suhteellisen tuttua ainakin pienimuotoisena. En ollut koskaan tehnyt mobiilipeliä, mutta silmissäni siinsi tulevaisuudennäkymä mobiilipelien kehittäjänä. Ajattelin syventää tietämystäni ja taitojani mobiilipelien kehittämisessä opinnäytetyöni kautta. Tavoitteena oli tehdä mahdollisimman omaperäinen mobiilipeli.

Jo aikaisessa vaiheessa päätin tehdä mobiilipelini Java ME:llä, koska olin saanut siihen alustavan koulutuksen Satakunnan ammattikorkeakoulussa. En ollut myöskään koskaan kokeillut tehdä mobiilipeliä millään muulla ohjelmointikielillä. Java ME oli looginen valinta myös sen vuoksi, koska oma matkapuhelimeni tukee sitä. Silti velvollisuuteni kaikelle avoimena IT-alan tulevana ammattilaisena oli selvittää muiden ohjelmointikielien mahdollisuudet, rajoitukset ja se, onko Java ME tosiaan paras vaihtoehto.

Luvussa 3.1 mainituista neljästä pelistä (Snake, Pac-man, Tetris ja Space Invaders) opinnäytetyönä tehty mobiilipeli on saanut eniten vaikutteita Snakesta ja Pac-Manista. Snaken vaikutteet näkyvät esteettömän pelialueen myötä ja Pac-Manista tutut pelaajaa jahtaavat haamut ovat muuttuneet tässä pelissä haiksi. Molemmista peleistä löytyvät kerättävät pallukat ovat mukana myös tässäkin pelissä, tosin planktonin muodossa.

Vaikutteiden ottamista muista peleistä ei voi välttää pelien tekemisessä, eikä siihen ole tarvettakaan. Pelejä sanan laajimmassa muodossa on kehitetty mahdollisesti niin kauan kuin ihmisiä on ollut, joten tietyt peruselementit näkyvät väkisinkin useimmissa nykyaikaisissa peleissä. Näihin kuuluvat mm. viholliset, jotka joko pyritään tuhoamaan tai joita paetaan (esim. shakki); pelin voittamisen kannalta arvokkaiden esineiden kerääminen (esim. Afrikan tähti), ja suorakulmion muotoinen pelialue, jota näkee useimmissa lautapeleissä ja videopeleissä.

Valmiissa pelissä olen kaikkein tyytyväisin pelin ideaan ja haiden tekoölyyn. Pelin idean oli tarkoitus olla erikoinen mutta toimiva, ja mielestäni se onnistui hyvin. Yleensä olen halunnut tehdä omalaatuisia pelejä, eikä opinnäytetyönä tehty mobiilipeli ollut poikkeus. Tekoölyssä olen tyytyväinen siihen, että haiden liikkuminen suuntautuu kohti pelaajaa, mutta sattumanvaraisesti, joten pelaaja ei koskaan tiedä miten hait tulevat seuraavaksi toimimaan. Tämä antaa pelaajalle mahdollisuuden päästä pakoon tukalastakin tilanteesta, mutta antaa haille myös mahdollisuuden saartaa pelaaja. Minulle on henkilökohtaisesti tärkeää, että jokainen pelikerta on erilainen, joten tähän panostaminen oli luonnollista.

Opinnäytetyön tekemiseen kului paljon enemmän aikaa kuin olin ajatellut, joten suunnitteluvaiheessa tekemäni aikataulus ei pitänyt. Matkan varrella oli paljon vaikeuksia, pääasiassa motivaation kanssa, mutta sain työn tehtyä. Jos opinnäytetyöllä olisi ollut tilaaja, kunnianhimoni ja motivaationi olisivat olleet todennäköisesti suurempia, ja mobiilipelistä olisi tullut kaikin puolin parempi. Pisteytysjärjestelmä olisi ollut paikallaan, grafiikat olisivat olleet vielä parempia ja kenties pelissä olisi ollut myös efektejä, kuten vedessä kelluvaa verta hain pureskellessa pelaajaa.

Opinnäytetyönä syntynyt mobiilipeli on vaatimaton, mutta tämä opinnäytetyö voi kuitenkin antaa mobiilipelin tekemisestä kiinnostuneelle lukijalle tietoa siitä, mitä mobiilipelin tekemisessä kannattaa ottaa huomioon ja mistä se koostuu. Olen silti tyytyväinen lopputulokseen, vaikka tavoitteet eivät täysin toteutuneetkaan.

## LÄHTEET

*Chauhan Mugdha 2005. Developing Java-Based Mobile Games* [verkkodokumentti]. [viitattu 25.11.2010]. Saatavissa:

<http://www.developer.com/ws/j2me/article.php/3502741/Developing-Java-Based-Mobile-Games.htm>

*Costikyan Greg 2002. WAP Game Design* [verkkodokumentti]. [viitattu 25.11.2010]. Saatavissa: <http://www.costik.com/presentations/>

*Crooks II Clayton E. 2005. Mobile Device Game Development*. Yhdysvallat: Charles River Media.

*Forum Nokia 2009. Why Choose Java For Mobile Game Development?* [verkkodokumentti]. [viitattu 25.11.2010]. Saatavissa:

[http://wiki.forum.nokia.com/index.php/Why\\_Choose\\_Java\\_For\\_Mobile\\_Game\\_Development%3F](http://wiki.forum.nokia.com/index.php/Why_Choose_Java_For_Mobile_Game_Development%3F)

*Game Programming Wiki 2010. How do I get Started* [verkkodokumentti]. [viitattu 25.11.2010]. Saatavissa: [http://gpwiki.org/index.php/How\\_do\\_I\\_get\\_Started](http://gpwiki.org/index.php/How_do_I_get_Started)

*IGDA 2005. 2005 Mobile Games White Paper* [verkkodokumentti]. [viitattu 25.11.2010]. Saatavissa:

[http://archives.igda.org/online/IGDA\\_Mobile\\_Whitepaper\\_2005.pdf](http://archives.igda.org/online/IGDA_Mobile_Whitepaper_2005.pdf)

*IGDAwiki 2010. Mobile Games* [verkkodokumentti]. [viitattu 25.11.2010]. Saatavissa: [http://wiki.igda.org/Mobile\\_Game\\_Development\\_SIG](http://wiki.igda.org/Mobile_Game_Development_SIG)

*Manninen, Tony 2007. Pelisuunnittelijan käsikirja: Ideasta eteenpäin*. Tallinna: Kustannus Oy Rajalla.

*NetBeans.org 2010. NetBeans IDE 6.9 Features* [verkkodokumentti]. [viitattu 25.11.2010]. Saatavissa: <http://netbeans.org/features/index.html>

*Pelkonen T. 2005. Mobile Games: An Emerging Content Business Area* [verkkodokumentti]. [viitattu 25.11.2010]. Saatavissa:

<http://www.springerlink.com/content/r450235712741g87/>

*Stack Overflow 2010. What Language To Learn For Developing Web/Mobile Games?* [verkkodokumentti]. [viitattu 25.11.2010]. Saatavissa:

<http://stackoverflow.com/questions/2403313/what-language-to-learn-for-developing-web-mobile-games>

*The Game Development Wiki 2010. How do I get Started* [verkkodokumentti]. [viitattu 25.11.2010]. Saatavissa:

[http://wiki.gamedev.net/index.php/How\\_do\\_I\\_get\\_Started](http://wiki.gamedev.net/index.php/How_do_I_get_Started)

*Wikipedia 2010/a. History of video games* [verkkodokumentti]. [viitattu 25.11.2010]. Saatavissa: [http://en.wikipedia.org/wiki/History\\_of\\_video\\_games](http://en.wikipedia.org/wiki/History_of_video_games)



*Wikipedia 2010/b. Mobile games* [verkkodokumentti]. [viitattu 25.11.2010]. Saatavissa: [http://en.wikipedia.org/wiki/Mobile\\_games](http://en.wikipedia.org/wiki/Mobile_games)

*Wikipedia 2010/c. Game programming* [verkkodokumentti]. [viitattu 25.11.2010]. Saatavissa: [http://en.wikipedia.org/wiki/Game\\_programming](http://en.wikipedia.org/wiki/Game_programming)

*Wikipedia 2011/d. Java Platform, Micro Edition* [verkkodokumentti]. [viitattu 19.2.2011]. Saatavissa: [http://en.wikipedia.org/wiki/Java\\_Platform,\\_Micro\\_Edition](http://en.wikipedia.org/wiki/Java_Platform,_Micro_Edition)