

Tallinna Ülikool  
Informaatika Instituut

# **Arvutimängude loomise protsess ja tarkvarad**

Seminaritöö

Autor: Leiko Luhamaa  
Juhendaja: Inga Petuhhov

Tallinn 2013

# Sisukord

Sisukord.....	2
Sissejuhatus .....	3
1. Arvutimäng.....	4
1.1 Lühiajalugu.....	4
1.2 Idee .....	5
2. Mängumootor .....	8
2.1 Mängumootori loomine .....	8
2.2 Olemasolevad mängumootorid.....	9
3. Tasemed.....	12
3.1 Kujundus.....	12
3.2 Varad .....	14
3.2.1 Modelleerimine.....	14
4. Lisavarad .....	16
4.1 Animatsioon .....	16
4.2 Heli .....	17
4.2.1 Muusika .....	17
4.2.2 Heliefektid .....	18
Kokkuvõte .....	19
Kasutatud kirjandus .....	20

## Sissejuhatus

Mängimisel on iga inimese elus üsna suur roll, sest just varajastest lapsepõlvemängudest õpime me esimesi probleemilahendusmeetodeid ning kujundame ka oma isiksust. Arvutimängudel on samasugune eesmärk. Need arendavad meid ja annavad meile ka samal ajal igavust peletavat tegevust.

Kuna arvutimängud ei ole mõeldud üksnes lastele ja noortele, vaid meelitavad üha vanemaid inimesi sellega tegelema, siis on loomulik, et ka mängude nõudlus kasvab üha suuremaks. Seepärast on vaja ka aina rohkem mänguloojaid, kes tahaksid ja oskaksid oma mõtteid mängudeks tegemiseks rakendada.

Käesoleva seminaritöö eesmärgiks on anda üldine ülevaade arvutimängude loomisest ja tuua näiteid selleks vajalikkudest tarkvaradest. Tööga tahetakse tekitada lugejas huvi videomängude loomise vastu ja anda neile teadmised edasiseks tegevuseks.

Seminaritöö on jagatud kolmeks suuremaks osaks. Esimeses osas räägib autor arvutimängudest ja nende ajaloost ning aitab kujundada ideid mängu loomiseks. Teises osas seletab autor lahti mängumootori tähenduse ja räägib olemasolevatest mängumootoritest. Kolmandas osas räägib autor arvutimängu taseme kujundamisest ja selle komponentidest.

Antud teema valiku ajendiks oli autori enda huvi arvutimängude ja nende loomise vastu ning soov jagada oma teadmisi ka teistega. Samuti eksisteerib selle teema kohta üsna vähe eestikeelset materjali.

# 1.Arvutimäng

Enne, kui saame üldse hakata tegelema arvutimängude planeerimise ja kujundamisega, tuleks kõigepealt mõista, mis reaalselt on arvutimäng. Tuntud mängu „*Civilization*“ autor Sid Meyeri sõnul on arvutimäng „seeria tähendusrikkaid valikuid“. Raph Koster, endine *Sony Online Entertainment*'i töötaja väidab: „Mängud on pärismaailma keeruliste probleemide multifilmiversioon“, „Mängud treenivad meid leidma alusmustreid“; „Lõbu on positiivne tagasiside, mida aju annab sulle asjade õppimise eest“. (Henno, 2011) Kõige lihtsamalt saab aga öelda, et mäng on vähemalt ühe mängijaga ja reeglitega tegevus, millel on olemas võidutingimus ning arvutimäng on mäng, mida mängitakse arvutiekraanil (Rogers, 2010).

„Lõbu“ andev mängude põhiatraktsioon on etteaimamatus, mis tekib mängija tegevusest. Mäng tuleb teha lõbusaks, kuid lõbu tekib vaid siis, kui mängus esineb üha uusi katsumusi, mis on korralikult tasakaalus nii et mäng ei oleks liiga lihtne ega ka liialt keeruline. (Henno, 2011)

## 1.1 Lühiajalugu

1947. aastal töötas Briti matemaatik Alan Turing välja teoreetilise arvuti male programmi. Tema kolleeg Ditrich Prinz kirjutas selle põhjal piiratud programmi, mis oli võimeline lahendama vaid „mate-in-two (matt-kahega)“ probleeme ehk olukordi, kus oli võimalik kahe käigu abil võita ning ei olnud võimeline mängima täis mängu. (Computer History Museum, n.d.)

1951. aastal tegi firma Ferranti valmis arvuti nimega NIMROD, mille ainus eesmärk oli mängida matemaatilist mängu „Nim“. (Goodeve, n.d.) Samal aastal tegi Aleksander S. Douglas valmis ka esimese arvutimängu „OXO“, mis kasutas digitaalset graafilist ekraani. „OXO“ on digitaalversioon mängust „*Noughts and Crosses*“ (trips-traps-trull). (Winter, n.d.)

1958. aastal tutvustati Brookhaveni Rahvusliku Laboratooriumi igaaastasel külalistepäeval William Higinbothami poolt loodud arvutimängu „*Tennis for Two*“ (Tennis kahele). Mäng töötas analoogarvutil ja kasutas ostsiloskoobi vektorekraani. (Brookhaven National Library, n.d.)

Läbi aastate 1957-1961 loodi Massachusettsi Tehnoloogiainstituudis asuvale eksperimentaalsele arvutile TX-0 hulk interaktiivseid graafilisi programme. Nende hulgas olid ka mängud „*Tic-Tac-Toe*“(Trips-Traps-Trull) ja „*Mouse in the Maze*“(Hiir labürindis). Mänge mängiti valguspliiatsi abil. (Graetz, 1981)

1962. aastal valmis mäng, mida peetakse esimeseks tõeliseks arvutimänguks, nimega „*Spacewar!*“. Mängiti kahekesi, juhtides kumbki ühte kosmoselaeva. Laeva sai keerata, liigutada ja sellega sai tulistada. Mängu mõtteks oli tabada teist mängijat enne, kui ise pihta saad. 11 aastat hiljem valmis esimene mäng, mida oli võimalik kõigil oma majapidamisse osta nimega „*Pong*“. (Juul, 2001)

Aastatel 1977-1993 domineerisid täielikult mängud, mis olid mõeldud ühele mängijale. Alles aastal 1993, kui valmis mäng nimega „*Doom*“, mis võimaldas mitmel mängijal samaaegselt ekraanil olla, hakkas *multiplayer* mängumoodus populaarsust koguma. Üleüldiselt on „*Doom*“, hoolimata kriitikast, olnud vägagi mõjukas ja pannud aluse esimese ja kolmanda persooni tulistamise mängude žanritele. (Juul, 2001)

## 1.2 Idee

Arvutimängu loomiseks peab kõigepealt olema olema hea mõte, mis ei hõlma ainult soovi teha mängu, vaid annab seletamisel üsna täpse ülevaate, milline meie mäng välja nägema hakkab. Mõtteid võib olla igasuguseid ning mõnikord on just kõige totramalt kõlavad ideed parimad, mis ei tähenda muidugi, et igast veidrast ideest hea mängu valmis saaks meisterdada, kuid päris kõiki mõtteid ka hüljata ei tohiks. Idee tekkimiseks on vaja inspiratsiooni, aga hea mõte võib tulla igal ajal ja kõikjal. Mõtteid saab ka ise mingil määral esile kutsuda näiteks järgneva viie tegevusega: (Rogers, 2010)

- 1. Loe midagi, mida sa tavaliselt ei loe.** Hulgaliselt on liikvel mängu, mis tunduvad kohati omavahel vägagi sarnased ning selle põhjus on lihtne. Paljudel arvutimängude tegijatel on sarnased huvid ja tegevused. Seepärast saab mängu teha huvitavamaks, kui arendada ennast valdkondades, millega igapäevaselt kokku ei puutu.
- 2. Jaluta või sõida autoga.** Kui inimese aju aktiivne osa tegeleb mingisuguse tuttava tegevusega, nagu käimine või autosõit, siis on inimese alateadvusel võimalik uitada ja hakata looma ühendusi, mida ta varem ei teinud. Selle protsessi käigus võib tekkida hulgaliselt ideid ka mängude jaoks.

3. **Mängi mõnda arvutimängu (soovitavalt halba).** Mängides halbu mängu, otsi üles, mis täpselt tehti halvasti ning ürita mõelda, mismoodi neid paremini teha.
4. **Osale vestlustes.** Oma mõtetest rääkimine ja teiste ideede kuulamine aitab luua uusi ideid, mis tavaliselt ületavad eelmisi. Nii aitad sa ka teistel oma mõtteid paika saada.
5. **Järgi oma kirge.** Ei või iial teada, millal sa saad oma lemmiktegevusi siduda mängudesse ja nii ei tundugi arvutimängu tegemine enam töö, vaid pigem lõbuna.

Hea mõte peaks kindlasti olema ka müüdav. Tõenäoliselt on vähe arvutimängu disainereid, kes teevad mängu ainult endale kasutamiseks. Ikka valmistatakse mäng, et seda teistega jagada ning võimalusel ka selle pealt veidike raha teenida. Arvutimängu edukaks levitamiseks rahva seas peab seepärast ka arvestama, mida mängijad tahavad. Tegelikkuses ei tea mängurid ka ise täpselt, mida nad heast mängust ootavad. Seepärast peabki mängu looja lisama mängu hulgaliselt detaile, mis võiksid mängurile huvi pakkuda. Selle töö juures on abistajaks teadmine, et mängurid tahavad ennast tunda mängu tegelase kaudu kellegina, kes nad reaalsuses ei ole. Olgu selleks siis keegi, kes on tark, tugev või lihtsalt milleski väga osav. Arvestama peaks ka sellega, millisele vanusegrupile mängu luuakse. Erinevate vanusegruppide huvid on üsnagi erinevad ning millegipärast pakuvad kindla vanusega inimestele alati huvi mängud, mis on mõeldud vanematele inimestele. (Rogers, 2010)

Mõtteid tekib tahes tahtmata, kuid mismoodi nendest siis ikkagi mäng saada. Ideede abil mõeldakse valmis lugu sellest, mis realselt toimuma hakkab. Loole peab mõtlema peategelase, keda juhib mängija. Tegelane võib olla vastavalt mängule kas visuaalne või mitte, kuid tema puudumisel ei ole mänguril rolli. Tegelasel on alati ka mingisugune nimetus ning see peaks kokku sobima mängu ja tegelase olemusega. Halb nimi nii mängul, kui ka tegelasel võib jätta kentsaka mulje ja teha mängu isegi ebameeldivaks. (Rogers, 2010)

Lisaks tegelasele on vaja sündmusi, mis hakkavad toimuma ning mida läbides tegelane loos edasi liigub. Lool on alati algus, keskosa ja lõpp. Loo algus on tegelase päritolu ja olemus. Sellega vastatakse, kes on meie tegelane ja mis temaga juhtus ning mismoodi sündmused mängu ehk keskosani jõudsid. Keskosa ongi mäng ise, mille jooksul mängur juhib tegelast ja üritab ületada erinevaid takistusi, et jõuda lõpuni. Lisaks on vaja lisada mängule lõpp, mis näitab edasisi sündmusi ja autasustab mängijat võidu tundega. (Rogers, 2010)

Peale loo ja tegelaste välja mõtlemist on kasulik kirjutada omale disaini dokument. See kujutab endast kogu mängu sisu ja toimumist filmistsenariumi kujul ning aitab saada paremat ettekujutust, milline mäng hakkab välja nägema. Kirjutamise ajal on võimalik eemaldada ka halbu mõtteid ja lisada midagi uut, kuid tuleks ka arvestada, et mängumaailm sobiks iseendaga kokku ning ei jää liiga palju vastamata küsimusi. Valmis dokument vähendab soovi programmeerimise käigus mängu muuta ning eemaldab võimaluse millegi unustamiseks. (LaMothe, 2002)

Mõnikord võib kirjutamise ajal tulla ette olukordi, kus mõtted ei taha enam tulla ning on tunne nagu ei olekski enam võimalik tööd edasi teha. Mõned mõtted, kuidas seda probleemi ületada: (Rogers, 2010)

1. **Kitsenda oma fookust.** Võimalik, et üritad teha liiga palju asju korraga. Jaga töö osad väiksemateks juppideks ning võta nad ette järk-järgult.
2. **Jaluta või tee trenni.** Inimese aju toitub verest, seepärast võib istudes jääda veri pidama vales asukohas. Liiguta ennast ja pane veri ringlema.
3. **Eemalda häirivad faktorid.** Mõnikord võib mõtte plokk olla tingitud muudest muredest, mille peale mõeldakse. Sellisel juhul tuleks enne need mured lahendada.
4. **Liigu heade osade juurde.** Mängudes on tihtilugu osi, mille kirjapanek ei ole üldsegi huvitav. Tee enne ära mõni põnev koht ja tule siis igavate osade juurde tagasi.
5. **Vaheta keskkonda.** Kodus võib olla liiga palju segajaid, nagu teler ja arvutimängud. Sellises olukorras tuleks minna kirjutama kohta, kus need puuduvad, nagu raamatukogu või välikohvik.

## 2. Mängumootor

Mängumootor ehk teisisõnu mänguraamistik on koodi kogum, mis moodustab töötava tarkvara ning seda kasutatakse mänguprojekti alusena. See koosneb mitmest väiksemast tarkvarast, mis võivad omada ka oma kasutajaliideseid. Mängumootorid on arvuti- ja videomängude tootjatele olulised, kuna head mängumootorit saab kasutada mitmetes erinevat tüüpi mänguprojektides. See laseb neil hoida aega kokku koodikirjutamise pealt ja lubab neil tegeleda mängu enda loomisega. (Sherrod, 2008)

### 2.1 Mängumootori loomine

On olemas suur hulk raamatuid, millest saab õppida mängumootori tegemist. Paljud neist nõuavad aga antud valdkonna üsna põhjalikke eelteadmisi. Mida täpselt teadma peab, sõltub küll vastavast raamatust kuid enamasti on nõuded üsnagi sarnased.

Põhiline nõue mängumootori loomisel on programmeerimiskeel. Paljud allikad kasutavad selleks keelt C++, kuna suur osa mängumootoreid põhineb sellel ja ka keele omadused on mängumootori tegemiseks sobivad aga reaalselt peaks ikka oskama seda programmeerimiskeelt, millega mängumootorit soovitakse teha. Seepärast on soovitatud õppida C++, et mõista selle alaseid raamatuid ning selle keele abil tehtud mängumootoreid ning kui on soovi teha oma mänguraamistikku muu programmeerimiskeelega, tuleks õppida ka selleks vastavat keelt. (Sherrod, 2008) Lisaks keelele võiksid olla tuttavad ka kõik muud programmid, mida kasutate mängumootori loomise juures. Mida täpselt vaja läheb, sõltub sellest, millist mängumootorit tehakse. Kindlasti peaks päris selge olema ka matemaatika, täpsemalt algebra, geomeetria ja trigonomeetria, mis aitab mõista 3D arvutigraafikat. (Hook, 1995)

Mängumootor tuleks kokku panna alamraamistike kaupa. Järgnevalt loend kõrgema astme alamraamistikest, millest mängumootor koosneb (Sherrod, 2008):

- **Renderdamissüsteem/Renderdamismootor** on mängumootori osa, mis loeb modelleerimisel tehtud faile ja loob nende alusel pildid, mida me näeme ekraanil. Olgu selleks siis objektid, maastik või tegelased mängus. (Slick, n.d.)
- **Füüsikaline süsteem/Füüsikamootor** on mängumootori osa, mis tegeleb suuruste, kauguste ja asukohtade arvutamisega ning see tuvastab ka põrkeid. Näiteks ütleb see



süsteem, milline kereosa hakkab kangelasel vigasaamisel veritsema ja kuidas ta seejärel liigub. (Bolton, 2006)

- **Sisendsüsteem**, nagu nimigi ütleb, haldab mänguri poolt sissetulevaid käsklusi.
- **Helisüsteem** haldab mängus olevaid helisid.
- **Võrgusüsteem/Võrgumootor** on vajalik ainult mootorites, millega luuakse võrgus mängimise võimalusega mängu. Haldab võrguühendusi, nende turvalisust jms. (Gillius's Programming, 2009)
- **Mäluhaldus programm** haldab ja koordineerib mälu võimalikult optimaalselt, et kiirendada mälutöötlusprotsessi ning sellega ka mängu ennast. (DeLoura, 2001)
- **Andmelaadimise programm** kontrollib tasemete, tegelaste ja objektide ettelaadimist. On mängu, kus on kasulikum laadida programmi sisse kogu olemasolev materjal, kuid mahukamatel mängudel laetakse mängu kvaliteedi ja kiiruse tagamiseks objekte kas tasemete või mõne muu süsteemi kaupa. (Leverton, n.d.)
- **Skriptimise süsteem** haldab koodifaile, mis juhivad mängu. Nendes saab kirja panna mängu kulgemise nii, et ei oleks vaja muuta mängumootori koodi. Näiteks, kui mängumootor ise võimaldab tegelase liigutamist paremale ja vasakule, siis skriptifailidesse pannakse koodi abil kirja tegelase võime hüpata või tulistada. Lisaks kodeeritakse skriptidesse ka mängu sündmused ja lugu. (Jones, 2006)

Hoolimata teadmisest, mis on mängumootor ja millest see koosneb, ei tähenda see veel, et on olemas oskus mänguraamistik ise kokku panna. Võimalus õppida on aga olemas ning appi tulevad meile selle juures mõned head raamatud. Näiteks võib tuua Allen Sherrodi „Game graphics programming“ ning David Eberly poolt kirjutatud raamatud „3D Game Engine Architecture“ ja „3D Game Engine Design“, millest esimene räägib pigem mängumootori teoreetilisest poolest ning teine õpetab mootori loomise praktilist osa. Võimalusel oleks ka kasulik mängude ja mängumootorite tegemist õppida kogenud programmeerijalt.

## 2.2 Olemasolevad mängumootorid

Mõeldes kõikidele mängudele, mis on loodud, siis on mõista, et meil peaks olema liikvel ka üsnagi suurel hulgal mängumootoreid, isegi kui ühe mootoriga tehakse mitmeid mängu. Pigem pööratakse ikka rohkem tähelepanu arvutimängudele, kui mootoritele, millel nad töötavad ja seetõttu ei puutu me igapäevaelus ka nendega eriti kokku. Järgnevalt tutvustatakse mõningaid mängumootoreid, nendele loodud mängude abil:

- **Jade** on Ubisofti poolt loodud mängumootor arvutimängule „*Beyond Good & Evil*“, mille peategelase järgi mootor ka omale nime sai. Mootor ise on üsnagi paindlik ning võimaldab erinevaid mängutsükleid ja detailset graafikat nii mängus kui ka animatsioonides. Samal mootril töötab ka näiteks „*Prince of Persia*“ triloogia. (MobyGames, n.d.)
- **Anvil**, tuntud ka nime all **Scimitar**, on samuti Ubisofti poolt loodud mängumootor, millel töötab mänguseeria „*Assassins Creed*“. (GamingExcellence, n.d.)
- **HPL** on mängumootor, mis on oma nime saanud õudusulmeirjanikult H.P. Lovecraft'ilt ning mille loojaks on väike Rootsi firma nimega Fictional Games. Mootori teisele versioonile, HPL2, on loodud populaarne õudusmäng „*Amnesia: The Dark Descent*“. (FictionalGames, n.d.)
- **Riot** on *Surreal Software*'i poolt loodud mootor, mis sisaldab taseme redaktorit ja modelleerijat. Mootoriga on loodud näiteks mängud „*The Suffering*“ ja „*The Lord of the Rings: The Fellowship of the Ring*“.

Muidugi on need vaid vähesed näited olemasolevatest mängumootoritest, kuid kahjuks pole ka siin loetletud mootoreid võimalik kõigil kasutada. Harva tahavad tootjad oma mootoreid enne nende aegumist lihtsalt kättesaadavaks teha ning isegi kui nad seda võimaldavad, ei tähenda see, et on võimalik mängumootorit tasuta kätte saada. Õnneks leidub aga ka tasuta kättesaadavaid mootoreid, mis sobivad alustavale mänguloojale harjutamiseks ideaalselt. Tutvustan järgnevalt neist mõningaid:

- **Blenderi** mängumootor on osa samanimelisest tarkvarapaketest, millega saab animeerida, renderdada ja tekstuurida 3D maailmaid. Mootoriga on võimalik luua arvutimängu ilma kodeerimata, tänu Blenderi loogika klotsidele. Blenderit on võimalik alla laadida ja kasutama õppida Blenderi kodulehel [www.blender.org](http://www.blender.org). (Bacone, 2012)
- **Build** on 2.5D tulistamismängude mängumootor, millega tehti algselt valmis mäng „*Duke Nukem 3D*“. Nüüdseks on seda täiustatud ja kasutatud ka rohkemate mängude loomiseks. Mootor on kättesaadav aadressil <http://advsys.net/ken/build.htm>. (Silverman, n.d.)
- **Cafu** on kõigile kättesaadav 3D graafika mootor ja mänguloomise komplekt, mille abil saab luua nii mängu, simulatsioone kui ka õpitarkvarasid kiirelt ja lihtsalt. Pakett on kodeeritud programmeerimiskeeles C++ ja kujundatud nii, et uute mängude

loomine oleks võimalikult lihtne. Tarkvara täiendatakse pidevalt ning mootor on kättesaadav aadressilt <http://www.cafu.de/>. (Fuchs, n.d.)

- **FIFE (*Flexible Isometric Free Engine*)** on mängul „*Fallout*“ põhinev vabavaraline mängumootor. Kodeerimine on võimalik nii programmeerimiskeeles C++ kui ka Python. Juhendid ja tarkvara asuvad aadressil <http://fifengine.net/>. (FIFE: Flexible Isometric Fallout-Like Engine, n.d.)
- **Gamvas** on Javascriptiga loodud 2D mängumootor. Mootoril on olemas rohkesti võimalusi tegelaste loomiseks ja kontrollimiseks. Osakeste süsteemiga on võimalik luua vihma, suitsu, tuld ja isegi sädemeid. Juhendid ja tarkvara on kättesaadavad aadressilt <http://gamvas.com/>. (Irrgang, n.d.)
- **Irrlicht** on vabavaraline kõrgetasemeline 3D mängumootor, kodeeritud C++ programmeerimiskeeles. Mootoril on olemas kõik uuemat tüüpi omadused, nagu müüdavatel 3D mängumootoritel. Veebist on mootorile võimalik leida kõiksugu täiustusi, kaasaarvatud lisasid teiste programmeerimiskeelte kasutamiseks. Mootor on kättesaadav tasuta aadressilt <http://irrlicht.sourceforge.net/>. (Welcome to the Irrlicht Engine, n.d.)
- **Jake2** on mängu „*Quake 2*“ mängumootori Java versioon ning selle kasutamiseks on vaja originaalmootori andmefaiile, mille kättesaamise juhend on koos tarkvara endaga olemas aadressil <http://bytonic.de/html/jake2.html>. (BytonicSoftware, n.d.)
- **Panda3D** on 3D mänguraamistik renderdamiseks ja mänguloomiseks programmeerimiskeeltega Python ja C++. Tarkvara on vabavaraline ja kättesaadav järgnevalt aadressilt <http://www.panda3d.org/>. (Carnegie Mellon University, n.d.)

Loetletud mänguraamistikud ei ole kaugeltki kõik, mis on vabavaraliselt kättesaadavad ning need ei pruugi olla ka parimad, kuid esimese mängu tegemiseks peaksid nad sobima. Edasi polegi muud, kui valida endale sobivaim mootor ja see alla laadida. Kindlasti võiks ka raamistiku kasutamisega teha lähemat tutvust ja alles siis hakata looma oma mängu, kuid see on juba igapähe enda otsustada.

### 3. Tasemed

Arvutimängu level ehk tase on mängu alamseksioon, milles mängija suhtleb maailmaga. Selleks, et mängu laadimine toimiks kõige tõhusamal viisil, tehakse seda tasemete kaupa ning mõned mängukonsoolid ja arvutid ei ole ka piisavalt võimsad, et uuemat tüüpi suuremahulisi mängu korruga laadida. Erinevate arvutimängude tasemed on üles ehitatud väga erinevalt, kuid siiski on neil olemas põhilised elemendid, millest tase koosneb.

(Castillo & Novak, 2008)

- **Toimumispaik** on ajastus ja koht, milles taseme tegevus aset leiab. See määrab ära, milliseid tegelasi ning objekte mängija võib kohata. Taseme toimumispaik peaks kindlasti sobima kokku ka mängumaailma toimumispaigaga. (Castillo & Novak, 2008)
- **Eesmärgid** on vajalikud, et mängumaailmas oleks midagi ka teha. Igas tasemes on etteantud ülesanded, mille täitmisel liigutakse mängus edasi. Eesmärgi olemuse määrab ära mängu liik ning mängija ülesandeks on leida strateegia, kuidas teatud eesmärki saavutada. (Castillo & Novak, 2008)
- **Riski-tulu** peab igas tasemes olema tasakaalustatud. Tasemes võib olla riske, kas suurte koletiste või ohtlike kohtade näol ning nende läbimisel saadakse mingi tasu. Tasakaal tasu ja riski vahel tähendab, et suurema riski puhul peaks olema ka suurem tasu ja vastupidi. Vastasel juhul võib mängija tunda end halvasti ning tal ei ole enam lõbus. (Castillo & Novak, 2008)
- **Mõistatused**, mida leidub paljudes mängudes, esitavad mängijale loogikal põhinevaid ülesandeid. Näiteks, mõnes tasemes tuleb liigutada objekte, et pääseda ligi taseme varjatud asukohtadele või tuleb leida peidetud vihjeid, et avada mõni uks. Ükskõik, millised on mõistatused, peavad mängijad ikka kasutama loogikat ja leidlikkust, et probleemi lahendada. (Castillo & Novak, 2008)

Hea taseme loomiseks peab kasutama kõiki tasemeloomise omadusi mõõdukalt ning neid üksteise suhtes tasakaalustama. Pole määratud struktuuri hea tasandi meisterdamiseks. Kõik sõltub katsetamisest ja uute mõtete rakendamisest. (Castillo & Novak, 2008)

#### 3.1 Kujundus

Taseme kujundamisel sõltub palju sellest, mis tüüpi mängu keegi omale teha soovib. Mängumaailmas jagunevad kõik mängud mõne žanri alla, kaasaarvatud need mida alles

tehakse. Mängu tüüp aga aitab kaudselt kindlaks määrata, milline loodava mängu struktuur hakkab välja nägema. Tasemekujundamise vaatenurgast jagatakse mängud seitsmesse kategooriasse. (Castillo & Novak, 2008)

- **RPG (*Role-Playing Games*) ehk rollimängud** on lool põhinevad mängud, kus iga uus keskkond viib lugu edasi. Sellist mängu kujundades jutustab disain ise lugu edasi. Mängud ise on küllaltki kauakestvad ja seetõttu on ka disainimine üsna pikk protsess (tasemed on mahukad). Tänu arvutite ja interneti kasvavale kiirusele kogub üha populaarsust rollimängude uuem versioon MMORPG (*Massively Multiplayer Online Role-Playing Games*). (Castillo & Novak, 2008)
- **Põnevusmängude (*Action games*)** põhiliseks elemendiks on põnevus, mistõttu mängudes ei keskenduta ka väga enam loole, vaid pigem märulilaadsele tulistamisele ja raiumisele. Selle mängužanri tasemete ülesehitus sõltub pigem alamžanrist ja seetõttu on tasemed erinevates mängudes ka väga erineva struktuuriga. (Castillo & Novak, 2008)
- **Liivakastimängudes (*Sandbox games*)** antakse mängijale vaba voli teha, mida ta soovib. Mängud toimuvad suurtel objektel, sündmusi ning võimalusi täis aladel, kus on ära määratud küll mängule üks kindel kulg, algusest lõpuni, kuid mängija saab igal hetkel sooritada ka kõrvalisi ülesandeid või niisama ümbrust uurida. (Castillo & Novak, 2008)
- **Lineaarsed mängud** on tuntud ka kui ettekodeeritud mängud, seetõttu, et neil on mängu kulg väga täpselt paika pandud ning puuduvad võimalused midagi väljaspool lugu teha. Lineaarsed mängud on tihti vägagi animatsioonirohked ja meenutavad seetõttu pigem interaktiivseid filme. Tasemed ise on mängu lihtsuse tõttu väga kvaliteetsed ning värvikirevad. (Castillo & Novak, 2008)
- **Strateegiamängudes** on põhiliseks tegevuseks ressursside kogumine ning seejärel nendega vägede loomine ja vallutamine. Seepärast peavad tasemed strateegiamängudes olema hästi läbimõeldud. Ressursid ei tohiks paikneda ainult ühes asukohas ning neile peaks olema võimalik ka ligi pääseda. Samal ajal peaks looma ka võimalusi kutsuda esile konkurentsi osapoolte vahel. Tasemed on piisavalt suured, et osalised kohe mängu algul üksteise vaateväljas ei oleks kuid, ei tohi olla ka liiga suured, et ära eksida ja vaenlast üldse mitte leida. (Castillo & Novak, 2008)
- **Simulatsioonimängude** põhimõte on luua tasemeid ja olukordi, mis oleksid võimalikult sarnased reaalsusele. Selliseid mängu luues võib tulla tihti ette olukordi,

kus kujundatakse reaalseid paiku maailmas. Ükskõik millega tegu on, simulatsioonides tehakse kõik võimalikult kõrge kvaliteediga. (Castillo & Novak, 2008)

- **Spordimängud** on küllaltki sarnased simulatsioonidele, sest mängu üritatakse teha võimalikult sarnased reaalsusele ning mängudes, kus kasutatakse areene või staadioneid ei ole vaja ka palju ettekujutusvõimet. Näiteks jalgpalliväljakutel on juba reeglitega paika pandud nõuded suuruse ja objektide asukohtade suhtes. Siiski on ka spordimänge, kus kujutlusvõime aitab luua uusi ja huvitavaid keskkondi mõne spordialaga tegelemiseks. (Castillo & Novak, 2008)

Alustades taseme kujundamist, oleks algselt vaja valmis teha plaan vastavast tasemest paberikandjal. Eriti vajalik on see olukorras, kus tööd ei tehta üksinda, vaid grupis ning on vaja teistele edasi anda oma mõtteid. Kolmemõõtmelises mängumaailmas kujundatakse plaanile enamasti taseme vaade ülevalt, kuid disaini tuleks mingil viisil sisse tuua ka kolmas mõõde, mis annab mängule realistlikuma mulje. (Castillo & Novak, 2008)

## **3.2 Varad**

Varadeks nimetatakse kõiki objekte, mis tuleb taseme kujundamisele eelnevalt mõne teise programmi abil valmis teha. Nende hulka kuuluvad objektid, tegelased, tekstuurid, helid, animatsioonid jms. Peale taseme ülesehituse valmis mõtlemist, tuleb hakata mõtlema varade peale, mida meie tase või tasemed sisaldavad. Selleks luuakse iga taseme kohta varade nimekiri, mida saab võimalusel anda edasi disainerite grupile. (Castillo & Novak, 2008)

Varade loomiseks on paljudel kättesaadavatel mängumootoritel kaasas selleks otstarbeks tarkvarad. Olukorras, kus vajalik vahend puudub, saab ise muretseda endale sobiva programmi, millega töö ära teha. Taseme loomiseks esmavajalikud varad luuakse modelleerimise ja tekstuurimisega (Castillo & Novak, 2008).

### **3.2.1 Modelleerimine**

Modelleerimine on mudelite loomise protsess, mida kasutatakse taseme täitmiseks objektidega. Kõik taseme objektid tuleb eraldi valmis modelleerida. Mudeleid on kasulik teha valmis alamkomponentide kaudu, sest vajadusel saab alamkomponentidest panna kokku uusi

objekte. (Castillo & Novak, 2008) Modelleerimist teostatakse eraldiseisvate tarkvaradega, millest mõnda järgnevalt tutvustan:

- **Maya** on üks populaarsemaid modelleerimis-, renderdamis- ja animeerimistarkvarasid. See sobib nii mängude, efektide, kui ka muude 3D disainiprojektide loomiseks ja kujundamiseks. Tarkvara ise on tasuline, kuid sellel on olemas tasuta proovi- ja õppeversioon tudengitele. (*Maya – 3D Animation Software*, n.d.)
- **Softimage** on kõrgetasemeline 3D-tegelase animatsiooni ja visuaalsete efektide tarkvara. Sellel on kaasas näoroboti tööriistad, millega saab teha kõrgekvaliteetseid näostruktuure. Tarkvarast on olemas tasuta proovipakett ja tasuta õppeversioon. (*Softimage – Visual Effects & 3D Game Development Software*, n.d.)
- **Wings 3D** on kergesti kasutatav kõrgema taseme modelleerija. Tarkvara pakub laias valikus modelleerimise tööriistu, kujundatava kasutajaliidese ning sisaldab valgustuse ja materjalide toetust. Tarkvara on tasuta kõigile kättesaadav aadressilt <http://www.wings3d.com/>. (Wings 3D, n.d.)

**Tekstuurimine** tähendab nii tekstuuride loomist, kui ka nende mudelitele omistamist. Vaadates objekti, saame me aru, kas see on konarlik, sile, märg, kuiv, terav või nüri. Tekstuurimise käigus tuleb kõikidele mängumaailmas asuvatele objektidele omistada pinnas. Kõik objektid on tehtud mingisugusest materjalist ja jutustavad meile mingi loo. Modelleerimistarkvarades on tavaliselt ka vahendid tekstuuride loomiseks. Üldiselt peaks olema võimalik luua tekstuure mistahes joonistusprogrammi abil. (Castillo & Novak, 2008)

## 4. Lisavarad

Selles peatükis tutvustan nüüd kahte varadegruppi, milleks on animatsioon ja heli ning mille eesmärgiks on teha mäng ilusaks ja kvaliteetseks. Neid õigesti kasutades aitab see kaasa põnevuse ja pinge loomisel, mis teeb mängu meeldejäavamaks ja huvitavamaks. Tegelikult on võimalik luua arvutimängu ka ilma neid varasid kasutamata, kuid tulemuseks oleks mäng, millel ei ole erilist süžeed ning mis ei ole ka silmale ilus vaadata.

### 4.1 Animatsioon

Animatsioon on liikuvate piltide loomise kunst. Meie räägime pigem arvutianimatsioonist, mis on sama asja edasiarendus ning mida tehakse arvutitega. Selleks, et luua liikumise illusiooni, näidatakse meile arvutiekraanil pilti ning seejärel asendatakse see kiiresti sama pildi koopiaga, mida on kergelt nihutatud. Selle võttega põhimõtteliselt saavutatakse liikumise illusioon telesaadetes ja filmides. (*Computer animation*, n.d.)

3D animeerimises ehitatakse (modelleeritakse) ekraanile objektid (mudelid), millele lisatakse külge virtuaalne luustik. 2D animeerimises kasutatakse erinevaid objekte ja töötluskihte, kas koos või ilma luustikuta. Animeerija saab siis objektide osi liigutada ja sellest liikumisest moodustatakse animatsioonilõigud. Töö lõpus tuleb kõik animatsioonid ka renderdada. (*Computer animation*, n.d.)

Arvutimängudes antakse animeerimise abil tegelastele võime liikumist näidata. Ilma animatsioonita oleks meil mängus jäigad ning kehalt liikumatud tegelased, kes müstiliselt vahetavad tasemel koordinaate. Animeerimist kasutatakse aga ka mängule videote loomiseks, et näidata sündmusi ja jutustada lugu piltlikult. Ka animeerimise tarkvarad on enamjaolt mängumootori pakettidega kaasas, kuid siiski tutvustan mõningaid animeerimise programme, millega oleks võimalik ka enda esimesele mängule animatsioone luua.

- **Animator HD** on varasemalt tuntud kui Animator DV, mis suutis varem töötatada vaid 2D tasemel. Nüüdseks on sellest välja töötatud kõrgema taseme animeerimistarkvara, millega saab luua 3D filme ja mida on võimalik kasutada ka filmivõtetel eeltöötlusvahendina. Tarkvara on võimalik proovida tasuta 30 päeva jooksul. (*AnimatorHD*, n.d.)



- **Anime Studio** on täiuslik animeerimise programm, millega saab luua nii 3D filme, kui ka joonisfilme ja animafilme. Sellega on lihtne luua isiklikke joonisfilmide taoliseid lühifilme, mida näidata ka teistele. Tarkvara ise on küll tasuline, kuid ka seda on võimalik testida 30 päeva tasuta. (*Anime Studi Pro – Overview*, n.d.)
- **Adobe After Effects**'i viimane versioon laseb kiirelt ja lihtsalt luua visuaalseid efekte ja liikuvat graafikat. Nagu teistelgi Adobe toodetel, on ka sellel hulgaliselt sisse ehitatud efekte ja lisasid, millega animeeriija tööd on üritatud lihtsustada. Ka seda toodet saab tasuta kasutada kahjuks ainult 30 päeva. (*Visual effects, motion graphics software – Adobe After Effects CS6*, n.d.)
- **Gmax** on kujundustarkvara **3ds Max** põhjal loodud tasuta kättesaadav 3D animeerimise programm, millel on olemas kõik kõrgema taseme funktsioonid animatsioonide ja animafilme loomiseks. Tarkvara on võimalik alla laadida aadressilt <http://www.turbosquid.com/gmax>, kus leiab ka programmile kasutusjuhendi. (*Description of Gmax 1.2*, n.d.)

## 4.2 Heli

Helide õige kasutamine aitab mängjatel süveneda rohkem mängu ja tekitab neile parema elamuse. Helidega aidatakse luua pilti keskkonnast, mida ekraanil ei näe, ning õigeaegsete helimuudatuste abil saab tekitada hetki, mis ehmatavad ja tekitavad põnevust. Mängudes jagunevad helid muusikaks ja heliefektideks ning hea mängu tegemiseks peab kasutama mõlemaid helisid osavalt. (Castillo & Novak, 2008)

### 4.2.1 Muusika

Muusika on mängus pidevalt mängiv heli, mis tekitab rütmi. Vanemates mängudes loodi muusikafail ja laeti see tasemesse, peale mida muusikat korrati, kuni taseme lõppemiseni. Uuemates mängudes aga kasutatakse muusikat, mis muutub, kui näiteks olukord läheb ohtlikumaks või midagi põnevat hakkab juhtuma. Selleks jälgitakse tegelase asukohta ja vajalikku kohta jõudmisel käivitatakse uusi heli osasid, mis meelitavad ka mängijat pingsamalt mängule kaasa elama. (Castillo & Novak, 2008)

Muusikat, mida mängus kasutatakse, teevad tootjad kas ise või lasevad palgatud muusikutel seda luua. Mängudisainer, kes alles õpib mängu tegema, kasutab selleks tõenäoliselt

internetist alla laetud muusikat, mis on mõne helitöötlusprogrammiga tundmatuseni muudetud. Seepärast jääb muusika loomine ja töötlemine siinkohal igäühe oma vastutusele.

#### **4.2.2 Heliefektid**

Heliefektid on kõik mängus tehtavad helid. Mängumaailmas on helid ühed põhilised mõjutajad, mis aitavad süveneda mängumaailma reaalsusesse. Mida rohkem on heliefekte, ja mida kvaliteetsemad need on, seda usutavam on ka mängumaailm. Isegi samadel objektidel, millel on erinev materjal, võib olla erinevaid heliefekte. Näiteks jalasammud kõlavad erinevalt, kui käia kivitaseandil või puidust põrandal. Heliefektidega saab ka luua illusiooni objektidest, näiteks lisades mängu linnulaul, tekib ka ilma linde nägemata ettekujutus, et nad kusagil ikka ringi liiguvad. (Castillo & Novak, 2008)

## Kokkuvõte

Käesoleva seminaritöö eesmärgiks oli anda üldine ülevaade arvutimängude loomisest ja tekitada sellega lugejas huvi antud teema vastu. Selleks kogus autor materjali võimalikult paljudest allikatest, mis rääkisid arvutimängude loomisest pigem teoorias, kui praktikas.

Töö alguses tutvustati arvutimängude ajalugu ning seletati lahti arvutimängu mõiste. Samuti anti nõu, kuidas kujundada mängu loomise ideed ja selle põhjal teha valmis ka disainiplaan.

Autor andis töös ülevaate mängumootoritest, nende loomisest ja tõi esile ka mõned juba loodud mängumootorid. Võimalikult selgeks üritati teha piir, kuskohast algab mäng ja lõppeb mängumootor, kuna tihti jääb just see paljudele segaseks.

Seletati lahti tasemekujunduse mõiste ning üritati võimalikult täpselt rääkida, mida ja kuidas on vaja teha, et oleks võimalik teha mängule kvaliteetseid tasemeid ning lõppkokkuvõttes ka kvaliteetset mängu. Toodi ka näiteid modelleerimis- ja animeerimistarkvaradest.

Töö edasiarendusena näeb autor võimalust saadud teadmiste abil hakata looma enda arvutimängu. Mängu tegemisel võiks lähtuda antud tööst ja selle juhenditest. Samuti võiks mängu tegemiseks kasutada selles töös näidetena toodud tarkvarasid.

## Kasutatud kirjandus

*AnimatorHD*(s.d.)

<http://animatordv.com/> (viimati külastatud 26.02.2013)

*Anime Studi Pro – Overview*(n.d.)

<http://anime.smithmicro.com/anime-pro.html> (viimati külastatud 26.02.2013)

Bacone, V.K. *Blender Engine:Characters*(November 2012)

<http://www.packtpub.com/article/blender-engine-characters> (viimati külastatud 24.02.2013)

Bolton,D. *What is a Physics Engine?*(22.10.2006).

<http://cplus.about.com/b/2006/10/22/what-is-a-physics-engine.htm>

(viimati külastatud 21.02.2013)

Brookhaven National Laboratory. *The First Video Game?*(n.d.).

<http://www.bnl.gov/bnlweb/history/higinbotham.asp> (viimati külastatud 19.02.2013)

BytonicSoftware. *jake2*(n.d.)

<http://bytonic.de/html/jake2.html> (viimati külastatud 24.02.2013)

Carnegie Mellon University. *Welcome to Panda3D*(n.d.)

<http://www.panda3d.org/> (viimati külastatud 24.02.2013)

Castillo, T., & Novak, J.(2008). *Game Development Essentials:Game Level Design*. Clifton Park USA: Delmar.

*Computer animation*(n.d.)

[http://www.sciencedaily.com/articles/c/computer\\_animation.htm](http://www.sciencedaily.com/articles/c/computer_animation.htm)

(viimati külastatud 26.02.2013)

Computer History Museum. *Opening Moves:Origins of Computer Chess*(n.d.).

<http://www.computerhistory.org/chess/main.php?sec=thm-42b86c2029762&sel=thm-42b86c4252f72> (viimati külastatud 16.02.2013)

DeLoura, M.A.(2001). *Game Programming: Gems 2*. Boston USA:Charles River Media.

*Description of Gmax 1.2*(n.d.)

<http://games.brothersoft.com/gmax.html> (viimati külastatud 26.02.2013)

FictionalGames. *About Fictional Games*(n.d.)

<http://www.frictionalgames.com/site/about> (viimati külastatud 24.02.2013)

*FIFE: Flexible Isometric Fallout-like Engine*(n.d.)

<http://classes.soe.ucsc.edu/cmeps164/Spring09/ass1/abrozek/FIFE.htm> (viimati külastatud 24.02.2013)

Fuchs, C. *Cafu 3d Game and Graphics Engine*(n.d.)

<http://www.cafu.de/> (viimati külastatud 24.02.2013)

GamingExcellence. *Exclusive:Assassins Creed Q&A – The Scimitar engine*(25.10.2007)

<http://www.gamingexcellence.com/features/exclusive-assassins-creed-q-and-a-the-scimitar-engine> (viimati külastatud 24.02.2013)

Gillius's Programming. *GNE-Game Networking Engine*(9.03.2009).

<http://www.gillius.org/gne/> (viimati külastatud 21.02.2013)

Goodeve,P. *NIMROD*(n.d.).

<http://www.goodeveca.net/nimrod/> (viimati külastatud 16.02.2013)

Graetz,J.M.(August 1981). *The origin of Spacewar*.Creative Computing.

<http://www.wheels.org/spacewar/creative/SpacewarOrigin.html>

(viimati külastatud 19.02.2013)

Hook, B.(1995). *Building a 3D game engine in C++*. Chichester UK:Wiley.

Henno,J.(2011). *Game programming coursebook*. Tallinn: Tallinn University of Technology Press.

Irrgang, H. *Home:Gamvas – the html5 canvas game engine*(n.d.)

<http://gamvas.com/> (viimati külastatud 24.02.2013)

Jones, S.(4.10.2006). *Game Scripting 101*

<http://www.gameengineer.net/downloads/tutorials/AngelScript/Game%20Scripting%20101.pdf> (viimati külastatud 23.02.2013)

Juul,J. *A clash between Game and Narrative: A history of the Computer game*(17.04.2001).

<http://www.jesperjuul.net/thesis/2-historyofthecomputergame.html>

(viimati külastatud 19.02.2013)

LaMothe, A.(2002). *Windows Game Programming For Dummies*. Chichester UK:Wiley.

Leverton, M. *Datafile routines*(n.d.).

<https://www.allegro.cc/manual/4/api/datafile-routines/> (viimati külastatud 23.02.2013)

*Maya – 3D animation Software* (n.d.)

<http://usa.autodesk.com/maya/> (viimati külastatud 25.02.2013)

MobyGames. *Game Engine:JADE*(n.d.)

<http://www.mobygames.com/game-group/game-engine-jade> (viimati külastatud 24.02.2013)

Rogers,S.(2010). *Level up! : the guide to great video game design*. Chichester UK: Wiley.

Sherrod, A.(2008). *Game graphics programming*. Boston USA:Charles River Media.

Silverman, K. *Ken Silverman's Build Engine*(n.d.)

<http://advsys.net/ken/build.htm> (viimati külastatud 24.02.2013)

Slick, J. *What is rendering? Finalizing the 3D image*(n.d.).

<http://3d.about.com/od/3d-101-The-Basics/a/Rendering-Finalizing-The-3d-Image.htm>

(viimati külastatud 21.02.2013)

*Softimage – Visual Effects & 3D Game Development Software* (n.d.)

<http://usa.autodesk.com/adsk/servlet/pc/index?id=13571168&siteID=123112>

(viimati külastatud 25.02.2013)

*Visual effects, motion graphics software – Adobe After Effects CS6*(n.d.)

<http://www.adobe.com/ee/products/aftereffects.html> (viimati külastatud 26.02.2013)

*Welcome to the Irrlicht Engine*(n.d.)

<http://irrlicht.sourceforge.net/> (viimati külastatud 24.02.2013)

*Wings 3D. A polygon modeler*(n.d.)

<http://www.wings3d.com/> (viimati külastatud 25.02.2013)

*Winter,D. Pong-Story:A.S.Douglas' 1952 Noughts and Crosses game*(n.d.).

<http://www.pong-story.com/1952.htm> (viimati külastatud 19.02.2013)