

TARTU ÜLIKOOL  
MATEMAATIKA-INFORMAATIKATEADUSKOND  
Arvutiteaduse instituut  
Informaatika eriala

**Dmitri Nikulin**  
**Interaktiivne mäng programmeerimise õppimiseks**  
Bakalaureusetöö (6 EAP)

Juhendaja: Helle Hein

Autor: Dmitri Nikulin.....“.....“ juuni 2011

Juhendaja: Helle Hein.....“.....“ juuni 2011

Lubada kaitsmisele

Professor: .....“.....“ juuni 2011

TARTU 2011

## Sisukord

Sissejuhatus .....	4
1. Mängu loomise protsess .....	6
2. Programmi nõuded .....	7
2.1. Üldinfo .....	7
2.2. Mittefunktsionaalsed nõuded.....	7
2.2.1. Kasutatavus.....	7
2.2.2. Jõudlus .....	7
2.2.3. Käideldavus .....	7
2.2.4. Tugi.....	7
2.2.5. Paigaldus.....	8
2.3. Funktsionaalsed nõuded .....	8
2.3.1. Tehingud .....	8
3. Kasutatud tehnoloogiad.....	9
4. Süsteemi üldine arhitektuur.....	10
4.1. Kasutusloo diagramm .....	10
4.2. Klassidiagramm .....	12
5. Kasutajaliides .....	14
5.1. Tekstid, ikoonid ja graafika.....	14
5.2. Ekraanipaigutuse teemad.....	14
5.3. Sisend ja väljundseadmed.....	14
5.4. Programmi kasutamise õppimine .....	14
6. Kasutusjuhend .....	15
6.1. Programmi instaleerimine.....	15
6.1.1. Failide kopeerimine CD-lt.....	15
6.1.2. Vajaliku tarkvara instaleerimine.....	15
6.1.3. Programmi käivitamine .....	15
6.2. Programmi kasutamine .....	16
6.2.1. Mäng .....	16
6.2.1.1. Eesmärk .....	16

6.2.1.2. Mängu kirjeldus.....	17
6.2.1.3. Käsud.....	18
6.2.1.4. Nupud .....	19
6.2.1.5. Mänguväljakud .....	20
6.2.1.6. Edukas läbimine .....	20
6.2.2. Teooria.....	20
6.2.3. Juhend .....	21
7. Piirangud ning tekkinud probleemid .....	22
8. Edasiarendamise võimalused .....	22
Kokkuvõte .....	23
Abstract .....	24
Kasutatud kirjandus.....	25

## Sissejuhatus

Mis on programmeerimine? See on protsess ja uute programmide loomise kunst, kasutades erinevaid programmeerimiskeeli. Programmeerimine ühendab endas loogika, teaduse, matemaatika ja tehnika elemente. Võiks öelda, et programmide loomine koosneb 90% matemaatikast, 9% loogikast ning ülejäänud osas koodi kirjutamisest [1-2].

Autoril polnud ülikooli astudes programmeerimise kogemust ja alguses oli väga raske süveneda informaatika erialasse, sest tavaliselt õppejõud arvab, et üliõpilasel on baas- ja algteadmised juba olemas. See on üks põhjustest, miks käesolev bakalaureusetöö teema sai valitud.

Käesoleval bakalaureusetööl on kaks eesmärki. Esiteks luuakse platvorm, kus saab õppida algteadmisi sellises valdkonnas nagu programmeerimine. Põhilised teemad, mis tulevad vaatluse alla, on järgmised:

- funktsioonid;
- tsüklid.

Teiseks eesmärgiks on aidata üliõpilastel aru saada, kas nad on õigesti toiminud, valides programmeerimist oma erialaks või mitte.

Programmeerimise õpetamiseks on veebis süsteem Light-Bot, kuid programmi lähtekoodi pole avatud. Programmeerimist tuli alustada tühjalt kohalt. Käesoleva töö ülesandeks oli teha analoogiline programm, kuid likvideerida esialgse programmi mõned vead. Loodud programmi eelisteks on see, et tank ei alusta käike oma algseisust vaid jätkab viimasest kohast.

Väga oluline on muuta õppimine maksimaalselt huvitavaks ja seetõttu on vaja, et õpetamise platvorm oleks kaasakiskuv ja interaktiivne, kuid peab arvestama ka sellega, et mäng peab asuma veebis. Kõige loogilisem lahendus antud probleemile on kasutada keskkonda Adobe Flash ehk lihtsalt Flash.

Flash fail võib endas sisaldada nii vektor- kui ka rastergraafikat. Vaatamata kogu animatsiooni keerukusele saame hoida faili suurust väiksena. See väga tähtis, sest kui veebirakenduse fail oleks liiga suur, siis brauserid ei saaks seda faili laadida [3].

Töoga on kaasas CD, millel asuvad:

- kasutusjuhend;
- tarkvara lähtekood;

- kompileeritud kood;
- html fail veebirakenduse kasutamiseks;
- bakalaureusetöö kirjalik osa.

## 1. Mängu loomise protsess

Mängu loomise protsess on väga loomulik töö. Selleks, et luua ükskõik millist mängu, peab olema idee. Järgmiseks sammuks on vaja organiseerida mängu kontseptsioon, et paremini aru saada eesmärgist ja põhimõtetest. On vaja üritada kirjeldada oma tulevast mängu ühe lausega.

Kui kõik eelmised sammud on tehtud, siis nüüd on olemas baas, millest lähtuda. On vaja kirjeldada lühidalt mängu tähtsamad osad. Oluline on mitte minna detailidesse. Detailid tekivad arendamise protsessis, kuid tuleb tähele panna, et mängu kontseptsiooni saab muuta, aga idee peab jääma samaks, muidu mäng ei saa kunagi valmis [4].

Peale idee formuleerimist tuleb luua mängu disain. See annab parema arusaamise sellest mida on vaja realiseerida.

Viimaseks sammuks on programmeerimiskeele valimine ja mängu kirjutamine. Tuleb alustada näidete uurimisest. Internetis on palju avalikke õppematerjale, kus selgitatakse põhjalikult mängu loomise protsessi.

## **2. Programmi nõuded**

Nõuded jagunevad funktsionaalseteks ja mittefunktsionaalseteks. Funktsionaalsed nõuded kirjeldavad tööd, mida antud süsteem peab tegema. Mittefunktsionaalsed nõuded pigem kirjeldavad, milline süsteem peaks olema [5-6].

### **2.1. Üldinfo**

Loodav programm on mäng, mille abil on üliõpilastel võimalus programmeerimise algtõdesid õppida ja arendada.

### **2.2. Mittefunktsionaalsed nõuded**

#### **2.2.1. Kasutatavus**

- Kõik elemendid peavad olema hästi läbi mõeldud. Kõik liigsed objektid, millel ei ole määratud kindlat funktsiooni, muudavad programmi keerulisemaks. Programmis peavad domineerima pildid, mitte puhas tekst.
- Programm peab olema võimalikult intuitiivne – ei ole vaja lugeda manuaali, et saada aru mida peab tegema.
- Kui mängu reeglid jäävad intuitiivselt arusaamatuks, siis rakendus peab omama eestikeelset abiinfot. Nuppude vajutamisel peab olema lühike tegevuse seletus.
- Rakendusel peab olema kasutajasõbralik ja mugav kasutajaliides.

#### **2.2.2. Jõudlus**

- Viivistusi ei tohiks olla.
- CPU kasutus peab olema väike, sest tegemist on veebirakendusega.

#### **2.2.3. Käideldavus**

- Rakendus peab olema võimeline töötama võrgus ja lokaalses süsteemis.
- Rakendus ei tohiks kasutaja arvutilt nõuda väga suurt ressursi
- Programm peab toimima kiiresti. Inimesed on kannatamatud ning kui klikkide järel ei hakka midagi toimuma või kui mingi funktsiooni täitmine võtab liiga palju aega, siis huvi liidese vastu lihtsalt kaob.

#### **2.2.4. Tugi**

- Rakendus peab olema programmeeritud programmeerimiskeeles ActionScript.
- Rakenduse dokumentatsioon peab vastama UML standardile
- Rakendus peab töötama platvormidel Windows XP, Windows Vista, Windows 7, Mac OS, Linux.

#### **2.2.5. Paigaldatavus**

Rakendus peab olema kergesti paigaldatav ja vajadusel kiiresti uuendatav.

### **2.3. Funktsionaalsed nõuded**

#### **2.3.1. Käideldavus**

- Tanki liigutamine.
- Käikude lisamine.
- Mänguväljaku muutmine.
- Algseisu vaatamine.
- Käikude kustutamine.
- Õppematerjalide lehitsemine.
- Kasutusjuhendi vaatamine.



### 3. Kasutatud tehnoloogiad

Programmi koostamiseks kasutatakse tarkvara ActionScript 3.0 ja Adobe Flash CS5. Viimane on võimas programm, mille abil saab luua erinevaid multimeedia "faile" - mängu, animatsioone, veebilehti ja rakendusi mobiilidele. Kasutajal on Flashi nägemiseks tarvis installeerida Adobe Flash Player, kuid tavaliselt on see juba tehtud.

Esmakordselt esitleti keskkonda Flash aastal 1996 ning ta saavutas kohe suure populaarsuse animatsioonide tegemisel, veebilehtede interaktiivsemaks muutmisel jne. Adobe Flash'is salvestatakse failid FLA formaati, kuid esitlemiseks veebis või arvutis konverteeritakse see SWF formaati [7].

Keskkonnaga Flash kaasasolev skriptimiskeel Actionscript kujutab endast baitkoodiks kompileeritavat keelt, mis võimaldab võimsaid mängu kiiresti programmeerida. Oma süntaksilt on ta sarnane keelega JavaScript [3].

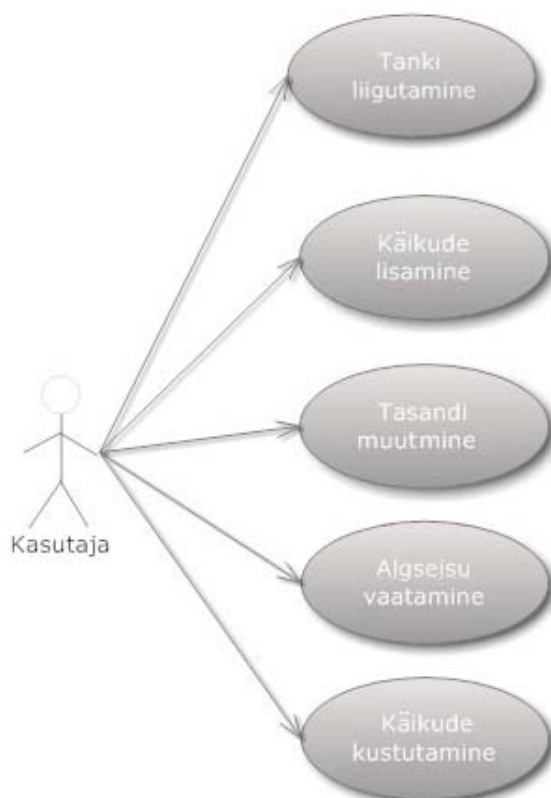
Lisaks kasutati kodulehekülje tegemiseks CSS, HTML, PHP. Kõikide failide üleslaadimiseks kasutati *FileZilla Client* programmi.

## 4. Süsteemi üldine arhitektuur

Arhitektuuri kirjeldamiseks kasutatakse kahte UML diagrammi. Esimesel kasutusloo diagrammil näidatakse kasutajate võimalusi mängus. Teisel on klassidiagramm, mis annab üldise arusaamise koodi struktuurist.

### 4.1. Kasutusloo diagramm

Joonisel 1 on esitatud programmi kasutaja võimalikud tegevused [8].



Joonis 1. Kasutuslugude diagramm.

Tabelis 1 esitatakse kasutuslugude detailsem kirjeldus.

Tabel 1. Kasutuslugude detailne kirjeldus

Kasutuslugu 1	Tanki liigumine.	
Kontekst	Süsteem, mis võimaldab objekti(tanki) positsiooni muuta.	
Tase	Põhiülesanne.	
Põhitegelane	Kasutaja.	
KIRJELDUS	Samm	Tegevus

	1	Kasutaja võtab alguskäsu.
	2	Kasutaja paneb alguskäsu oranžile kastikestele.
	3	Kasutaja võtab liikumise käsu(d).
	4	Kasutaja paneb liikumise käsu oranžile kastikesele.
	5	Kasutaja klikib nupul "Algus".
Tulemus veasituatsiooni korral	Tanki liikumist ei toimu.	

Kasutuslugu 2	Käikude lisamine.	
Kontekst	Süsteem, mis võimaldab lisada programmi käike.	
Tase	Põhiülesanne.	
Põhitegelane	Kasutaja.	
KIRJELDUS	Samm	Tegevus
	1	Kasutaja võtab suvalise käsu.
	2	Kasutaja paneb käsu oranžile kastikestele.
Tulemus veasituatsiooni korral	Ei toimu käikude lisamist.	

Kasutuslugu 3	Mängulaua muutmine.	
Kontekst	Süsteem, mis võimaldab mängulauda suurendada või vähendada.	
Tase	Põhiülesanne.	
Põhitegelane	Kasutaja.	
KIRJELDUS	Samm	Tegevus
	1	Kasutaja klikib nupul "pluss" või "miinus".
Tulemus veasituatsiooni korral	Mängulaua muutumist ei toimu.	

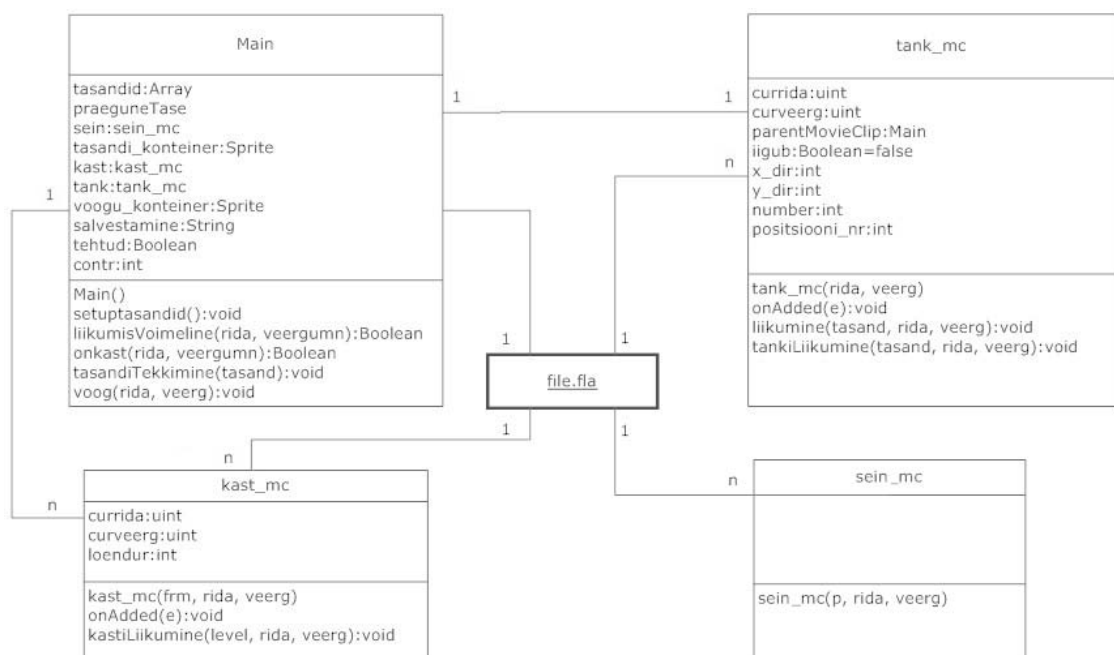
Kasutuslugu 4	Algseisu vaatamine.	
Kontekst	Süsteem, mis võimaldab kasutajatel tanki algseisu vaadata.	
Tase	Põhiülesanne.	

Põhitegelane	Kasutaja.	
KIRJELDUS	Samm	Tegevus
	1	Kasutaja klikib nupul “Algseis”.
Tulemus veasituatsiooni korral	Ei toimu algseisu vaatamist.	

Kasutuslugu 5	Käikude kustutamine.	
Kontekst	Süsteem, mis võimaldab kõik käigud kustutada ja alustada mängu uuesti.	
Tase	Põhiülesanne.	
Põhitegelane	Kasutaja.	
KIRJELDUS	Samm	Tegevus
	1	Kasutaja klikib nupul „Uuesti“.
Tulemus veasituatsiooni korral	Käikude lisamist ei toimu.	

## 4.2. Klassidiagramm

Programmi klassidiagramm on esitatud Joonisel 2 [9-10].



Joonis 2. Klassidiagramm.

Fail `fail fla` on Flash fail, mis reguleerib kogu programmi tööd. Siin asub mängu funktsionaalsus.

Klass `Main` joonistab kogu mängu osa. Antud klass kontrollib ja muudab mänguväljakut.

Klass `tank_mc` genereerib tanki ja viib selle vajalikule positsioonile, lisaks on võimalus tanki positsiooni muuta ja kontrollida kas ta saab liikuda või mitte.

Klass `kast_mc` genereerib kasti ja annab talle vajaliku positsiooni, lisaks on võimalus kasti positsiooni muuta. Kui kast liigutati risti peale, siis ta muutub oma värvi.

Klass `sein_mc` genereerib seinad, liikumise teed, roheline risti, millele on vaja kasti liigutada ja määrab nende positsiooni.

## **5. Kasutajaliides**

### **5.1. Tekstid, ikoonid ja graafika**

Kõik tekstid nuppudel on kirjutatud stiilis *Copperplate Gothic Light*, suurem osa nendest suurte tähtedega, kasutatud on erinevaid ikoone ja graafilisi objekte, mis täidavad nuppude funktsioone. Taust on valitud ergutava tooniga, et kasutajal tekiks huvi edasi mängida [12].

### **5.2. Ekraanipaigutuse teemad**

Menüü on esitatud piltide-nuppude valikuga ning suurem osa ruumist on ära kasutatud. Objekti paigutus muutub vastavalt ekraani suuruse muutumisele.

### **5.3. Sisend ja väljundseadmed**

Hiirt kasutatakse valiku tegemisel, väljundseadmeks on ekraan.

### **5.4. Programmi kasutamise õppimine**

Kui midagi jääb arusaamatuks, siis on kättesaadav ka programmi kirjeldus. Lisaks on mõnedel nuppudel ka nende tegevuste lühikirjeldus.

## **6. Kasutusjuhend**

### **6.1. Programmi installeerimine**

#### **6.1.1. Failide kopeerimine CD-lt**

Ainuke nõue programmi käivitamiseks on Flash mängija olemasolu ja interneti ühendus. Soovitav on kogu CD sisu kopeerida kohalikku kataloogi, kuna programmi uuesti kompileerimine ja Adobe Flash installeerimine nõuavad kirjutamise õiguseid.

#### **6.1.2. Vajaliku tarkvara installeerimine**

Kõigepealt on vaja veenduda, et arvutisse on installeeritud Flash mängija. Kui ei ole, siis võib käivitada Flash installeerimist kataloogist, kuhu on kogu CD informatsioon koperitud (kasutada faili `install_flash_player.exe`). Teiseks võimaluseks on alla laadida viimane versioon (hetkel on selleks Adobe Flash Player 10.3.181.26) aadressilt: <http://get.adobe.com/flashplayer/>.

#### **6.1.3. Programmi käivitamine**

Kui Flash mängija on juba olemas ja installeeritud, siis programmi käivitamiseks on kolm võimalust.

1. Klikkida kaks korda failil `file.swf` ja programm ongi käivitatud. Tulemuseks peab saama rakenduse, nagu on näha Joonisel 3.
2. Küllastada programmi kodulehekülge, mis asub aadressil: <http://suncap.zzl.org/bakalaureusetoo/>. Valida menüüs nupp „Mäng / Õppimine“. Tulemuseks peab ilmuma ekraanile sama rakendus nagu Joonisel 3.
3. Kasutada lähtekoodi kompileerimist. Arvutis peab olema kas Adobe Flash CS4 või Adobe Flash CS5. Avada fail `file fla` ühega nendest programmidest ja valida klahvikombinatsioon „Cntr + Enter“. Pidada silmas, et kõik ActionScript (laiend `.as`) ja Flash (laiend `.fla`) failid peavad asuma samas kataloogis. Tulemuseks peab ilmuma rakendus nagu Joonisel 3.



Joonis 3. Pilt töötavast rakendusest.

## 6.2. Programmi kasutamine

Programm jaguneb kolmeks osaks:

1. Mäng – programmi peamine osa. Siin on võimalik oma teadmisi rakendada.
2. Teooria – osa, kus saab õppida põhimõisteid teemadest tsüklid ja funktsioonid. Vaatleme ainult kahte teemat, sest mäng ise õpetab ainult nende teemade kasutamist.
3. Juhend – kirjeldatakse mängu reeglid ja põhimõtted.

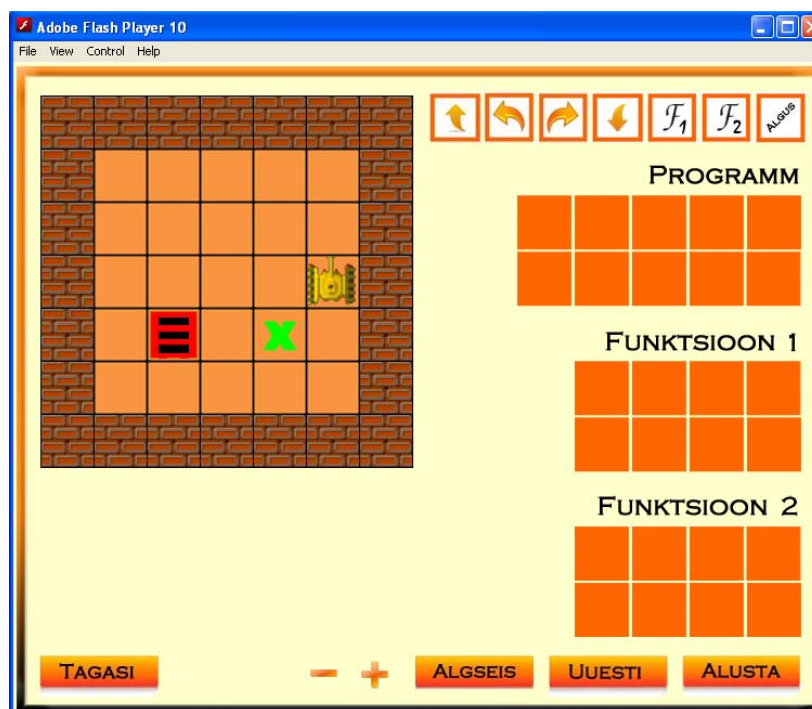
### 6.2.1. Mäng

#### 6.2.1.1. Eesmärk

Kasutaja mängib kollase tankiga. Tanki võib erinevate käskudega liigutada edasi, tagasi või teha vasak- või parempööret. Mängu eesmärgiks on liigutada punast kasti rohelisele ristile. Kui kasutaja selle saavutas, siis muutub kasti värv roheliseks. Kasti liigutamise võimalus on ainult tankil. Mänguväljak on sooritatud ainult siis, kui kõikide kastide värv on roheline.

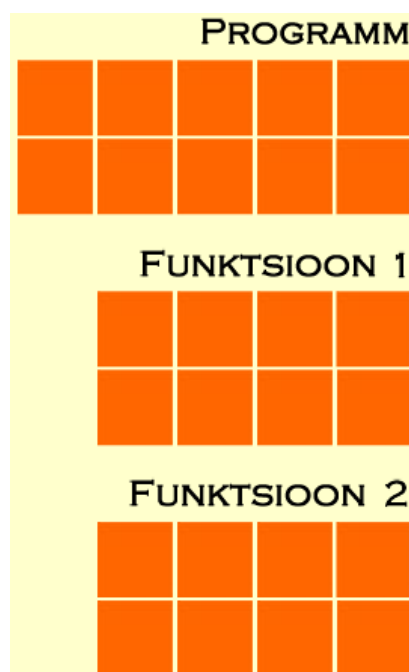


## 6.2.1.2. Mängu kirjeldus



Joonis 4. Mängu aken.

Joonisel 4 on näha mängu aken ühe mänguväljakuga. Tanki liigutamiseks on vaja anda talle käsklusi. Selleks tuleb neid võtta hiirega ja panna oranžidele kastikestele. See on mälu analoog. Iga kastike kujutab endast ühte mälupesad.



Joonis 5. Mälupesad.

“Programmi” pesad omavad tähtsamat rolli, kui funktsiooni pesad, sest kogu protsess alustab tööd just nendest. Alles siis kui “programmi” pesadel leidub funktsiooni käske, hakatakse ka funktsioonipesasid vaatlema.

### 6.2.1.3. Käsud

Edasi vaatleme igat käsku eraldi. Mängus on kolme tüüpi käske:

- Liikumise käsud
- Funktsiooni käsud
- Alguse käsk

#### Liikumise käsud

Liikumise käskudega on võimalik tanki asukohta muuta. Käivitamiseks on vaja võtta liikumise käsk ja panna see ühele oranžile kastikesele. Seda käsuliiki võib panna igale mälupevale vaatamata sellele, kas see on funktsiooni või programmi pesa.

#### Funktsiooni käsud

Need käsud viitavad sellele, et programmil on vaja sooritada kõik funktsioonis asuvad käsud. Funktsiooni saab käivitada, kui võtta funktsiooni käsk ja panna ühele oranžile kastikesele. Funktsiooni käsku ( $F_1$ ,  $F_2$ ) ei saa panna funktsioonide kastikestele (Joonis 5). Peale funktsiooni käsu lisamist ilmub paneel (Joonis 6), kus saab valida korduste arvu, st. mitu korda järjest funktsioon hakkab oma kärke kordama. Rohelise nupuga korduste arv suureneb, kuid punasega väheneb. Vaikimisi korduste arv võrdub ühega. Sellel etapil kasutaja ei saa tanki liigutada, tanki liigutamine on võimalik ainult pärast tulemuse salvestamist, mida tehakse nupuga “Lõpp”. Lõpuks ilmub korduste arvu teade.



Joonis 6. Funktsiooni korduste arvu paneel.

## Alguse käsud





On ainult üks alguse käsk, mis näitab programmi töö alustamist. Tuleb pidada meeles, et ilma selle käsuta rakendus ei hakka tööle. Alguse käsk tuleb panna programmi esimesele mälupeale. Näide kõikidest käskudest ja nende liikidest on toodud Joonisel 7.

	Tank teeb ühe sammu ettepoole. Liikumise käsk.
	Tank pöörab vasakule. Liikumise käsk.
	Tank pöörab paremale. Liikumise käsk.
	Tank teeb ühe sammu tagasi. Liikumise käsk.
	Esimese funktsiooni väljakutsumine. Funktsiooni käsk.
	Teise funktsiooni väljakutsumine. Funktsiooni käsk.
	Määratakse programmis algust. Peab panema esimesse kastikesse. Alguse käsk.

Joonis 7. Käskude kirjeldus.

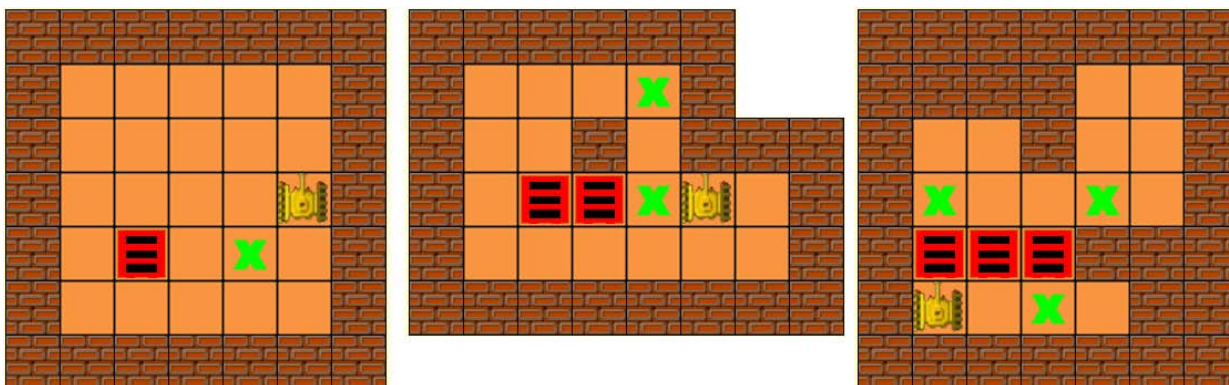
### 6.2.1.4. Nupud

Selleks, et midagi hakkaks programmis toimuma, peab kasutama nuppe. Nende kirjeldus on toodud Joonisel 8.

	Programm alustab oma tööd.
	Kusutakse kõik objektid. Tank viiakse algpositsiooni
	Näidatakse tanki algseis. Midagi ei kustuta ja ei muutu.
	Mänguväljaku muutmine. Pluss suurendab mänguväljakut, miinus vähendab.

Joonis 8. Nuppude kirjeldused.

### 6.2.1.5. Mänguväljakud



Joonis 9. Mänguväljakud.

Hetkel on mängus on kolm erinevat mänguväljakut (Joonis 9). Mängu alustatakse kõige lihtsamast väljakust, et kasutaja saaks paremini aru programmi loogikast ja struktuurist. Soovi korral on võimalik mänguväljakuid vahetada.

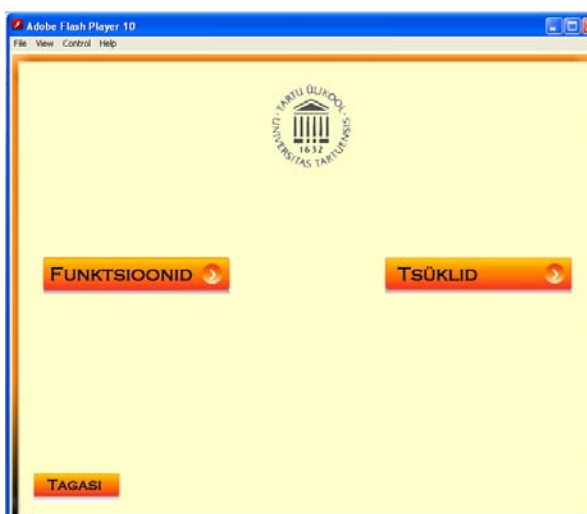
### 6.2.1.6. Edukas läbimine

Kui kasutaja on edukalt esimese taseme (mänguväljaku) läbinud, siis ilmub ekraanile teade “Palju õnne!

Nüüd on aeg midagi raskemat lahendada.” Järgnevatel mänguväljakutel sellist teadet ei ilmu.

### 6.2.2. Teooria

Teooria osa jaguneb kaheks: funktsioonid ja tsükliid (Joonis 10). Vajutades ühele kahest nupust, on näha teksti, kus selgitatakse, mis on funktsioon ja tsükkel. Kogu õppimine toimub programmeerimiskeele Python baasil.



Joonis 10. Teooria aken.

### **6.2.3. Juhend**

Kasutajal on võimalus vaadata kasutusjuhendit, kuid selleks peab olema internetiühendus, sest vastava nupu klikkimisel avaneb programmi kodulehekülg.

## **7. Piirangud ning tekkinud probleemid**

Antud programm on piiratud võimalustega. Peamine probleem seisneb selles, et puudub käskude kastikeste eemaldamise võimalus. Seda probleemi saab lahendada, kui vajutada nupule “Uuesti”, kuid see on ebamugav. Teine tähtis probleem on seotud funktsioonidega. Mängus saab muuta funktsiooni korduste arvu, kuid oleks tore, kui igale funktsioonile saaks seda arvu eraldi märkida. Puuduseks saab märkida ka väikest mänguväljakute arvu. Kasutajale oleks igav seda mängu mitu korda mängida. Mänguväljakute arv peaks olema vähemalt 15.

## **8. Edasiarendamise võimalused**

Edasiarendamise võimalusi on palju. Esiteks tuleks alustada ülaltoodud probleemide lahendamise ja edasi laiendada programmi. Näiteks oleks tore, kui saaks õppida mitte ainult funktsioone ja tsükleid, vaid ka muutujaid ja massiive. Programmi kodulehekülge võib teha ka atraktiivsemaks, kasutades erinevad veebitehnoloogiad.

## **Kokkuvõte**

Käesoleva bakalaureusetöö eesmärgiks oli luua programm, mis annab kasutajale võimaluse õppida programmeerimise mõningaid oskusi. Õppimise protsess peab olema huvitav ja võimalikult lihtne. Programmi kasutajateks võiksid olla esimese kursuse üliõpilased, kes omandavad algteadmisi programmeerimises. Koostatud programm on interaktiivne ja intuitiivne ning probleemi tekkimisel on kasutajal võimalus vaadata kasutusjuhendit.

Lõputöö tekstiline osa esitab programmi analüüsi, kirjeldades erinevaid nõudeid, struktuuri ja ehitust. Olemas on ka põhjalik paigaldamisjuhend. Süsteem ise on arendatud keeles ActionScript Adobe Flash raamistikus. Kõikide failide üleslaadimiseks kasutati tarkvara FileZilla Client.

# **The interactive game to learn programming fundamentals**

Bachelor Thesis

Dmitri Nikulin

## **Summary**

The aim of the present bachelor thesis is to create an interactive system to help students in learning some programming fundamental skills.

The game rules are very simple. The player can move a yellow tank with different commands and the purpose is to push a red box on green cross with the tank. After the goal has been achieved color of the box will change to green. The game has three playgrounds with different level of difficulty.

The textual part of the bachelor thesis is an analysis of the created system. Generally this document can be divided into seven parts – introduction, requirements, system overview, user guide, used technologies, system unsolved problems and finally future plans.

In introduction a short description of the system is presented. The second chapter describes system functional and nonfunctional requirements. The third chapter gives a code structure overview and describes basic use cases of the game and the user guide. The unsolved problems of the new system and the future work are described in the last chapter.

The new system is ready for installation and using.



## Kasutatud kirjandus

1. Programmeerimine, Wikipedia, <http://et.wikipedia.org/wiki/Programmeerimine> (viimati vaadatud 27.04.2011).
2. Programeerimine, Wikipedia, [http://wiki.hinnavaatlus.ee/index.php/Sissejuhatus\\_programmeerimisse](http://wiki.hinnavaatlus.ee/index.php/Sissejuhatus_programmeerimisse) (viimati vaadatud 27.04.2011).
3. ActionScript, Wikipedia, <http://en.wikipedia.org/wiki/ActionScript>(viimati vaadatud 27.04.2011).
4. Mängu loomise protses, <http://blog.moonmana.com/page/sozdanie-flesh-igry-s-chego-nachat> (viimati vaadatud 20.06.2011).
5. Wikipedia, Functional requirement [http://en.wikipedia.org/wiki/Functional\\_requirement](http://en.wikipedia.org/wiki/Functional_requirement) (viimati vaadatud 15.06.2011).
6. Wikipedia, Non-functional requirement [http://en.wikipedia.org/wiki/Non-functional\\_requirement](http://en.wikipedia.org/wiki/Non-functional_requirement) (viimati vaadatud 15.06.2011) .
7. Flash, Wikipedia, [http://en.wikipedia.org/wiki/Adobe\\_Flash](http://en.wikipedia.org/wiki/Adobe_Flash) (viimati vaadatud 27.04.2011).
8. Use case diagram, <http://www.ts.mah.se/RUP/wyliecollegeexample/courseregistrationproject/artifacts/analysisdesign/sadoc.htm#Toc> (viimati vaadatud 20.06.2011).
9. Class diagram, Wikipedia, [http://en.wikipedia.org/wiki/Class\\_diagram](http://en.wikipedia.org/wiki/Class_diagram)(viimati vaadatud 20.06.2011).
10. Class diagram, Robert C. Martin, <http://www.objectmentor.com/resources/articles/umlClassDiagrams.pdf> (viimati vaadatud 20.06.2011).
11. Kasutajaliidese kavandamine Loeng 7, E. Leego, [http://courses.cs.ut.ee/2009/ui/uploads/Main/MTAT.03.032\\_Loeng\\_7\\_030409\\_Leego.pdf](http://courses.cs.ut.ee/2009/ui/uploads/Main/MTAT.03.032_Loeng_7_030409_Leego.pdf) (viimati vaadatud 6.06.2011).
12. Flash. <http://metshein.com/content/view/16/26/> (viimati vaadatud 27.04.2011).
13. Flash.<http://mdesign.ru/publications/programming/41213c428b8e8?start> (viimati vaadatud 27.04.2011).
14. Flash AS3 Programming, Adobe Corp. (viimati vaadatud 15.06.2011).
15. Real Word Flash Game Development, Chris Griffith (viimati vaadatud 15.06.2011).
16. ActionScript 3.0 for Flash 2009, Colin Mooock (viimati vaadatud 15.06.2011).

17. ActionScript 3.0 Сборник рецептов 2008, Джон Лотт, Деррон Шалл, Кейт Питерс.  
(viimati vaadatud 15.06.2011).
18. Viidete loomine. <http://flash-master.net/archives/153> (viimati vaadatud 15.06.2011).
19. How to Cheat in Adobe Flash CS5: The Art of Design and Animation, 2010, Chris Georgeries (viimati vaadatud 15.06.2011).
20. *Flash*'i presentatsioon. [http://easyflash.org/flashlearn/yourflashwork/876-prezentaciya-  
tehnologiya-flash.html](http://easyflash.org/flashlearn/yourflashwork/876-prezentaciya-<br/>tehnologiya-flash.html) (viimati vaadatud 15.06.2011).
21. XML+AS3, autor on kasutaja ily294, [http://easyflash.org/flashlearn/yourflashwork/878-  
flash-vizitka-xmlas3.html](http://easyflash.org/flashlearn/yourflashwork/878-<br/>flash-vizitka-xmlas3.html) (viimati vaadatud 12.06.2011).
22. Create a flash game like security tutorial, Emanuele feronato,  
<http://www.emanueleferonato.com/2007/07/27/create-a-flash-game-like-security-part-3/>  
(viimati vaadatud 27.04.2011).
23. Lihtsa mängu tekitamine, kasutaja mr.developer  
[http://easyflash.org/flashlearn/flashvideotutorials/1647-seriya-video-urokov-po-  
sozdaniyu-prostoj-igry-ot-nachala-i-do-konca-svoimi-rukami.html](http://easyflash.org/flashlearn/flashvideotutorials/1647-seriya-video-urokov-po-<br/>sozdaniyu-prostoj-igry-ot-nachala-i-do-konca-svoimi-rukami.html) (viimati vaadatud  
12.06.2011).
24. Veebivorm, <http://ruseller.com/lessons.php?rub=29&id=283> (viimati vaadatud  
18.06.2011).
25. Veebi vorm, <http://ruseller.com/lessons.php?rub=29&id=283> (viimati vaadatud  
18.06.2011).
26. Dark, Side, Fade, Lee Brimelow, <http://www.gotoandlearn.com/> (viimati vaadatud  
10.06.2011).