

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Joosep Tiismus 134552

ADOBE AFTER EFFECTSI TÖÖKIIRUSE TÕSTMISE LAHENDUS

Bakalaureusetöö

Juhendaja: Tarvo Treier
Magister MSc

Tallinn 2018

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Joosep Tiismus

30.12.2017

Annotatsioon

Käesoleva bakalaureusetöö eesmärgiks on kiirendada Adobe After Effectsi monteerimise efektiivsust lubades kasutajal salvestada enda animatsioonid, et neid hiljem taaskasutada.

Töö alguses analüüsitakse erinevaid eksisteerivaid monteerimiseks mõeldud programme, kuidas nemad võtmekaadrite ja animatsiooniga hakkama saavad ning millised pistikprogrammid või skriptid aitavad töökiirust parandada võrreldes sisse ehitatud funktsionaalsusega.

Antud lõputöö käigus valmib skript, mis suudab võtmekaadreid salvestada ja taaskasutada nii, et see on kasutajale võimalikult mugav ja efektiivne.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 29 leheküljel, 8 peatükki, 15 joonist, 2 tabelit.

Abstract

Solution for increasing workflow in Adobe After Effects

The objective of this bachelor thesis is to increase workflow in the video editing software, Adobe After Effects, by allowing user to save custom animations and reuse them at a later date, thus saving time on creating them from scratch.

The author analyses different existing editing software to conclude if any of them already offer a solution for saving and reusing animations, that is simple and fast for the user but also complex enough for professionals to do their work in. After determining that After Effects is the way to go, different existing plugins and scripts are being compared to see how they help with the current problem and also portraying their downsides, as to why they aren't suitable enough.

The result of this thesis is a script which can save user generated keyframes and reuse them later on. It should be faster than any of the existing methods, yet simple enough for any user to handle. To conclude how well the new script holds up, a simple test will be run between it and different existing methods to see how well it compares.

The thesis is in Estonian and contains 29 pages of text, 8 chapters, 15 figures, 2 tables.

Lühendite ja mõistete sõnastik

AE	Adobe After Effects
Mac	Apple poolt loodud arvutid
ExtendScript	Javascriptil põhinev programmeerimiskeel, mida kasutatakse Adobe programmide skriptimiseks
JavaScript	Objektorienteeritud programmeerimiskeel, mida kasutatakse peamiselt veebilehtede skriptimiseks
Pistikprogramm	Plugin
Kiirklahv	Shortcut
FX Console	After Effectsi pistikprogramm efektide lisamiseks klippidele
KF Console	Lõputöö käigus loodav skript
SDK	Tarkvaraarenduse tööriistade kogumik, mille abil luuakse tarkvara kindla tarkvaraplatvormi jaoks

Sisukord

1 Sissejuhatus	10
1.1 Probleem.....	10
1.2 Eesmärk	10
1.3 Alameesmärgid.....	11
1.4 Töö ülesehitus.....	11
2 Monteerimine	12
2.1 Mis on monteerimisprogramm?.....	12
2.2 Kuidas võtmekaadreid kasutada	12
2.3 Probleem ja lahendus.....	13
3 Erinevad monteerimisprogrammid	15
3.1 Enimkasutatud monteerimisprogrammid	15
3.2 Programmide võimekus.....	16
3.2.1 Monteerimisprogrammid.....	16
3.2.2 Kompositsiooni loomise programmid	17
3.2.3 After Effects on parim valik animatsioonide loomiseks	17
4 Eksisteerivad lahendused.....	19
4.1 Pistikprogrammid	19
4.1.1 Pistikprogrammide plussid	22
4.1.2 Pistikprogrammide miinused.....	22
4.2 Sisseehitatud võimalused.....	23
5 Kas võtmekaadrite taaskasutamise vastu on vajadus?.....	25
5.1 FX Console	25
5.2 KF Console	26
5.3 Kaasmonteerijad	26
6 Lahenduse arendus	28
6.1 Arendamisel kasutatud tehnoloogiad ja võimalused	28
6.1.1 Pistikprogramm	28
6.1.2 HTML5 paneel	28
6.1.3 Skript	29

6.2	Arendamisel kasutatud tööriistad	29
6.3	Versioonid	29
6.3.1	Versioon 1	30
6.3.2	Versioon 2	31
6.3.3	Versioon 3	32
7	Valminud rakendus.....	33
7.1	Esmakordne ülesseadmine.....	33
7.2	Rakenduse kasutamine	34
7.3	Ajaline võrdlus teiste rakendustega.....	35
7.3.1	Manuaalselt animatsioonide tegemine	36
7.3.2	Eelloodud pistikprogrammi kasutamine.....	37
7.3.3	<i>KF Console</i>	37
7.3.4	Kokkuvõte	38
8	Kokkuvõte	39
	Kasutatud material.....	40
	Lisa 1 – Koodi asukoht.....	42

Jooniste loetelu

Joonis 1 Näide: Positsiooni võtmekaadrid ajateljel: (a) ajatelje alguses, (b) ajatelje keskel, (c) ajatelje lõpus.	13
Joonis 2. Adobe Premiere ajatelg erinevate klippidega.....	16
Joonis 3. Adobe After Effectsi ajatelg erinevate klippidega	17
Joonis 4. <i>Mister Horse</i> pistikprogramm	20
Joonis 5. <i>Motion 2</i> pistikprogramm.....	20
Joonis 6. <i>Layermonkey</i> pistikprogramm.....	21
Joonis 7. <i>Keyframe Presets</i> pistikprogramm	22
Joonis 8. Adobe After Effectsi tööaken erinevate moodulitega (10 nähtavat akent ja 5 vahekaartidena).....	23
Joonis 9. FX Console aken otsinguaknaga ja kiirklahvidega	25
Joonis 10. Lahenduse esialgne kuju	30
Joonis 11. Võtmekaadrite salvestamise funktsioon.....	31
Joonis 12. Võtmekaadri klipile lisamise funktsioon.....	31
Joonis 13. <i>KF Console</i> lõppkuju	33
Joonis 14. <i>KF Console</i> tegevusskeem	35
Joonis 15. Animatsiooni lisamise etapid erinevate meetoditega	36

Tabelite loetelu

Tabel 1. Erinevate animatsioonide loomise ajakulu.....	14
Tabel 2. Meetodite võrdlus.....	38

1 Sissejuhatus

Videotöötluste maailmas on palju erinevaid programme millega videosi monteerida, kuid professionaalsel tasemel, inimeste jaoks kes igapäevaselt sellega tegelevad, on neid ainult käputäis ning isegi need programmid ei ole perfektsed ja jätavad teatud valdkondades soovida.

1.1 Probleem

Peamine probleem videode töötlemise juures on aeg. Kõik tegevused nõuavad aega ja paljud on sellised mida saaks automatiseerida, sest need kipuvad ennast kordama. Üks sellistest tegevustest on animatsioonide loomine ja kasutamine. Nende loomine võtmekaadreid kasutades võib võtta paar sekundit kuni pool minutit, oleneb animatsiooni keerukusest, kuid see on piisav, et töökiirust häirida ja päeva lõpuks võib olla kaotsi läinud selle arvelt juba pool tundi.

Peamine motivatsioon lahenduse leidmiseks tuli autoril, kes tegeleb videode monteerimisega igapäevaselt. Ta avastas, et kuigi pidevalt on vaja taaskasutada samu animatsioone, puuduvad selle jaoks efektiivsed meetodid ja küsides kaasmonteerijate käest lahendusi, leidis et ei ole üksi selle probleemiga.

Adobe After Effects on ainult üks programmidest, mis on mõeldud monteerimiseks, kuid peamine rõhk on lahenduse leidmine sellele programmile, sest kuigi võtmekaadreid on võimalik teistes programmides luua, siis sama tulemuse saavutamine on keerukam ja kohati võimatu.

1.2 Eesmärk

Lõputöö eesmärk on luua Adobe After Effectsile lahendus, mis kiirendab animatsioonide taaskasutamist klippidel ja on teistest olemasolevatest lahendustest kõige efektiivsem.

1.3 Alameesmärgid

Arenduskäik koosneb erinevate eksisteerivate lahenduste uurimisest. Võrreldakse teisi monteerimiseks mõeldud programme ja kuidas nemad animatsioonidega toime tulevad, olemasolevaid pistikprogramme ning teisi skripte. Lõpuks luuakse lahendus, mis jätab ära teiste lahenduste negatiivsed küljed ja kiirendab kogu animatsioonide rakendamise protsessi.

1.4 Töö ülesehitus

Töö teises peatükis analüüsib autor, mis on monteerimine ja kuidas võtmekaadreid kasutada, et anda lugejale parem ülevaade miks sellist lahendust on vaja ja kuidas animatsioonide loomine töötab. Töö kolmandas peatükis võrreldakse erinevaid monteerimisprogramme ja nende võimekust, et valida kõige mõistlikum programm animatsioonide loomiseks. Neljandas peatükis analüüsitakse After Effectsi jaoks eksisteerivaid- ja sisse ehitatud lahendusi, et enda lahendus nende põhjal arendada. Viiendas peatükis uuritakse, kas loodava skripti jaoks on vajadus ja mida teised monteerijad pakutud lahendusest arvavad. Kuuendas peatükis räägitakse arenduskäigust ja seitsmendas peatükis uuritakse valminud lahendust ning võrreldakse seda teiste eksisteerivate lahendustega, et määrata selle efektiivsus.

2 Monteerimine

Antud peatükis autor selgitab, mis on monteerimisprogramm, mis on võtmekaadrid, kuidas nendega animatsioone luua ja peamised probleemid, mis kaasnevad nendega töötamisel. Selle eesmärk on anda parem ülevaate lugejale miks sellist lahendust on vaja, mida soovitakse saavutada ja kui keeruline on animatsioonide loomine.

2.1 Mis on monteerimisprogramm?

Tänapäeval filmid ja isegi lihtsamad videod koosnevad erinevatest videoklippidest. Klippe on võimalik kokku panna erinevate programmidega, mis varieeruvad kasutusvajaduse poolest. Kõige lihtsamaid on võimalik enda telefonis kasutada, mille peamine funktsioon on ainult klippide liitmine, kuid keerulisematega on võimalik lisada olemasoleva klipi peale teksti või pilte, kombineerida erinevaid klippe või luua midagi täiesti uut.

Teksti ja illustratsioonide lisamine ei ole keeruline tegevus, kuid staatilise pildi ekraanile panemine ei ole kõige optimaalsem ja võib vaatajat isegi häirida. Selleks animeeritakse illustratsioone, et juhtida paremini vaataja pilku, tõsta esile eeltoodud elemendid ja muuta nende ilmumine inimeste jaoks sujuvamaks. Animatsioonide loomist võimaldavad enamuse modernseid monteerimisprogramme ja need saavutatakse võtmekaadritega [1].

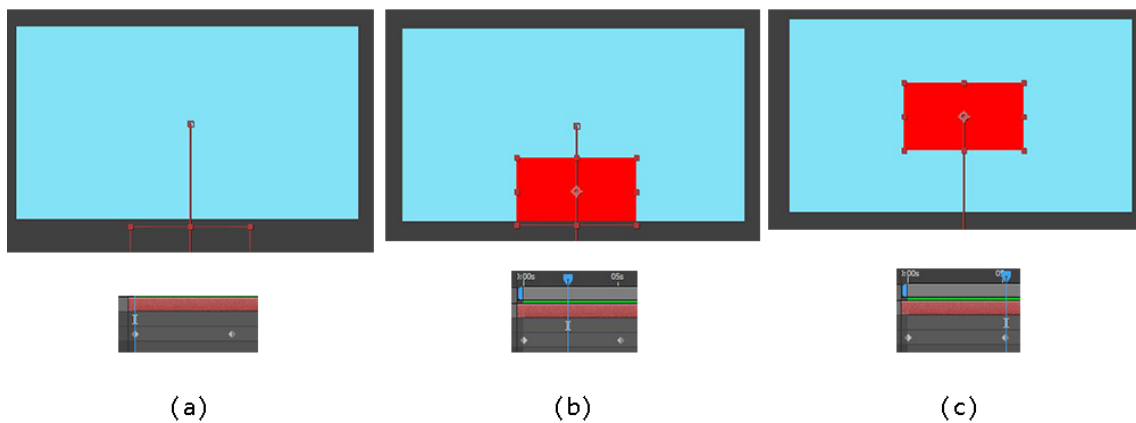
2.2 Kuidas võtmekaadreid kasutada

Igal visuaalsel elemendil, näiteks pildil või tekstil, on erinevad parameetrid – positsioon, suurus, läbipaistvus, rotatsioon. Need on põhitalad, millele animeerimine keskendub.

Praktiliselt on võtmekaadrite kasutamine lihtne ja otsekohene. Kasutaja valib ajateljel alguspunkti ning sätib illustratsiooni, mida soovib kuvada, esialgsesse seisundisse. Selle järel ta annab programmile teada, et soovib klipi parameetrid meelde jätta ning ajajoonele ilmuvad võtmekaadrid iga muudetud parameetri kohta. Järgmisena liigub kasutaja ajateljel lõpp-punkti ja muudab illustratsiooni seisundit selliseks milliseks soovib. Enam

ei ole programmile vaja öelda, et ta jätaks meelde parameetrid, sest ta salvestab ise automaatselt iga muudatuse, mis on erinev esialgsete salvestatud parameetritega. Sellega on lihtne animatsioon loodud. Ajajoonel liikudes on võimalik näha, kuidas ta ühest seisundist teise liigub.

Joonisel 1 on näha lihtsat animatsiooni, kus pilt liigub ühest positsioonist teise. Selle jaoks on loodud kaks võtmekaadrit, millest esimene hoiab pildi positsiooni, kui ta on kaadrist väljas ja teine on salvestanud pildi positsiooni kaadri keskel.



Joonis 1 Näide: Positsiooni võtmekaadrid ajateljel: (a) ajatelje alguses, (b) ajatelje keskel, (c) ajatelje lõpus.

Keerulisemate animatsioonide puhul luuakse vahepealsed sammud, mis ütlevad illustatsioonile, kas ta peab kiirust lisama, aeglustama või kõikvõimalike muid asju tegema, mis on vajalik parima tulemuse saavutamiseks.

2.3 Probleem ja lahendus

Võtmekaadrid on vajalikud igal pool ja neid tuleb pea, et igapäevaselt kasutada, kuid paljudel juhtudel kasutatakse samu või vähemalt sarnaseid võtmekaadreid, mida on kunagi eelnevalt kasutatud. Otseselt ei võta lihtsamate animatsioonide loomine palju aega, ainult paar kuni paarteist sekundit, kuid lõpptulemusena ajakulu akumulereub ja nädalas võib hakkab selle peale tunde kaduma. Rääkimata, et iga kord, kui on vaja sama animatsiooni uuesti kasutada, paneb see töötamisele vaimse pausi, sest on vaja sooritada tegevus, mis on ajju juba tuubitud. Animatsioonide taaskasutamine ja lisamine peaks aga juhtuma mõttekiirusel, ilma monteerimisest kõrvale kaldumata.

See ei ole probleem, mida saab lahendada, sest animatsioonid vajavad pidevalt muutmist ja olemasolevad lahendused jäävad puudulikuks, kuid see on protsess mida on võimalik kiirendada. Lubades kasutajal salvestada enda võtmekaadrid, et neid hiljem teiste animatsioonide peal kasutada, välistaks vajaduse neid iga kord uuesti luua. Esmakordsel kasutamisel on vaja aega investeerida, et animatsioon tekitada, kuid peale ühte korda ei ole vaja enam samu võtmekaadreid kunagi luua. Soovides animatsiooni muuta, võib eelnevad võtmekaadrid välja kutsuda, muudatused sisse viia ja uus versioon salvestada, samuti, ilma kõike uuesti tegemata.

Demonstreerimiseks, kui palju aega võib võtta erinevate animatsioonide loomine, valmistas autor teatud animatsioonid ja salvestas tulemused tabelisse 1. Ajaline kulu varieerub iga kasutaja kohta, kuid eesmärk oli anda üldine vaade, kuidas animatsioonide loomise ajakulu sõltub võtmekaadrite arvust ja kuigi paar sekundit ei ole tohutu aeg mida kulutada, et animatsioon iga kord uuesti luua, siis keerulisematel juhtudel, mis võtavad üle poole minuti, ei ole see enam mõistlik.

Animatsioon	Lihtne haihtumine ekraanilt	Sujuv positsioonivahetus	Animatsioon, mis kukub ja põrkab vastu maad
Võtmekaadrite arv	2	4	35
Ajakulu	3 sekundit	10 sekundit	1 minut

Tabel 1. Erinevate animatsioonide loomise ajakulu

3 Erinevad monteerimisprogrammid

Tänapäeval on monteerimisprogramme lõpmatult palju ja need varieeruvad kasutusala, hinna, efektiivsuse ja paljude muude tegurite poolest. On olemas lihtsamaid, millega iga inimene saaks hakkama koheselt, kuid mis piirduvad väga vähese funktsionaalsusega ja keerulisemaid, mis võivad võtta nädalaid, et selgeks õppida ning on mõeldud professionaalseks kasutuseks, mitte hobikorras.

Enamus programme toetavad võtmekaadrite loomist ja kasutamist. Seega on võimalik, et äkki ühel, populaarsematest programmidest, on juba leitud parem lahendus animatsioonide taaskasutamisele ning lisaks uuritakse selles peatükis, mis mõte on üldse After Effectsi peal monteerida, kui on olemas teised variandid.

3.1 Enimkasutatud monteerimisprogrammid

Tõestamiseks, miks After Effects on kõige mõistlikum programm animatsioonide loomiseks, tuleb kõigepealt uurida, kas eksisteerib paremaid videode töötlemiseks loodud programme ja kuidas nemad animatsioonidega tegelevad. Selleks otsisin internetist viis kõige populaarsemat konkurenti [2], [3], [4].

- HitFilm Pro
- Adobe Premiere
- Sony Vegas Pro
- Final Cut Pro
- Cyberlink PowerDirector

Konkurendid on valitud nende ligipääsetavuse alusel, kas nendes on võimalik tegeleda animatsioonidega ja luua enda võtmekaadreid ning kas need on piisavalt robustsed funktsionaalsuse poolest, et rahuldada ka professionaale.

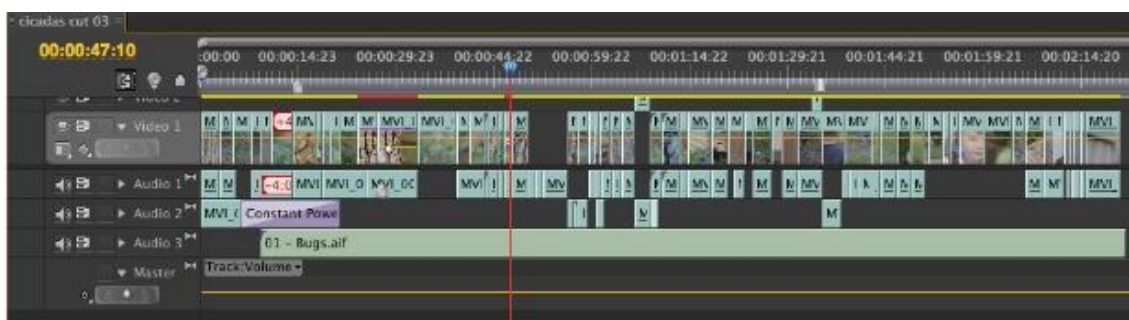
3.2 Programmide võimekus

Monteerimiseks ei ole palju vaja ja enamused programmid saavad hakkama lihtsate ülesannetega, kuid antud töö sihtgrupiks on inimesed, kelle jaoks monteerimine on töö või igapäevane tegevus. Inimesed kes pidevalt loovad videosid ja iga kokkuhoitud sekund loeb.

Eelnimetatud programmid on võimalik jagada kahte gruppi. Monteerimisprogrammid ja kompositsiooni loomise programmid.

3.2.1 Monteerimisprogrammid

Monteerimisprogrammide all mõeldakse neid, mille põhiline eesmärk on klippide paika seadmine. Nad on loodud eesmärgiga luua film või video ja sisaldavad peamist funktsionaalsust, mis on tähtis selle saavutamiseks nagu eelloodud üleminekud, heli parandamine või värvide tasakaalustamine. Klippe saab asetada ajateljele üksteise kõrvale (Joonis 2), kuid ka üksteise kohale, mis tähendab et video võib koosneda sadadest erinevatest klippidest aga kasutajal on seda lihtne hallata.



Joonis 2. Adobe Premiere ajatelg erinevate klippidega

Selliste programmide alla kuuluvad eelnimetatutest Cyberlink Powerdirector, Sony Vegas Pro, Final Cut Pro ja Adobe Premiere. Kõikidel on oma negatiivsed ja positiivsed pooled, kuid üleüldises kasutuses on sarnased. Ainuke, milline neist ei ole sobilik, on Final Cut Pro, sest seda on võimalik ainult Mac arvutite peal kasutada, samas kui teised toetavad nii Windowsit kui ka Maci.

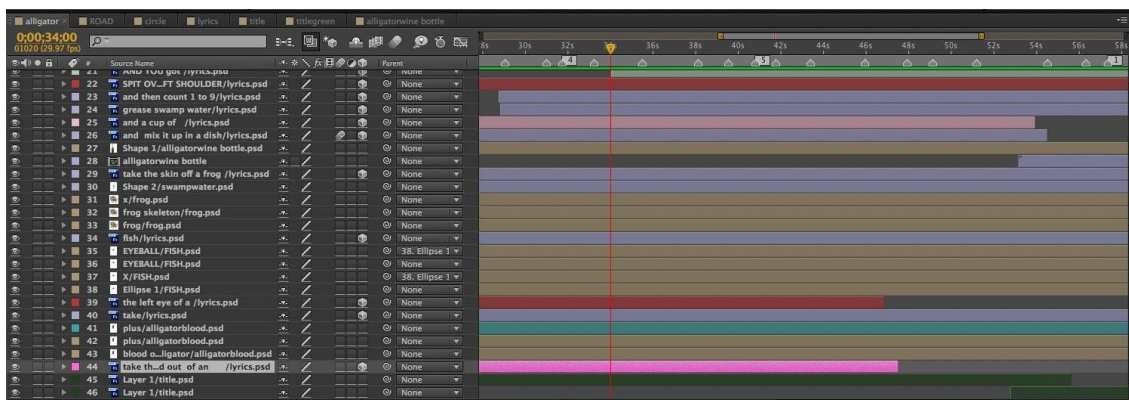
Kuigi need saavad hakkama videode kleppimisega ja animatsioonide loomine on võimalik luua, siis nende loomine ei ole piisavalt täiuslik. Valmistatud animatsioonid näevad välja algelised, sest võtmekaadrite loomine ja haldamine ei ole nende programmide eesmärk. See on

kõige selgemini arusaadav Premiere Pro'ga, sest see on loodud sama firma poolt kui After Effects ja need kaks programmi eksisteerivadki eraldi, sest täidavad erinevaid eesmärke [5].

3.2.2 Kompositsiooni loomise programmid

Teine peamine grupp on kompositsiooni loomise programmid, mille alla kuuluvad Hitfilm Pro ja After Effects.

Mis eristab neid monteerimisprogrammidest on see, et klippe ei ole võimalik paigutada üksteise taha vaid iga klipp on eraldi real (Joonis 3). See teeb projekti haldamise, kus on sadu erinevaid klippe, keerulisemaks, kuid nende programmide peamine eesmärk ei ole panna kokku täispikk film, vaid hoopis rakendada efekte eelnevalt välja valitud klippide peal, luua animatsioone ja tegeleda keeruliste protsessidega mida tavalised monteerimisprogrammid ei võimalda.



Joonis 3. Adobe After Effectsi ajatelg erinevate klippidega

Funktsionaalsuse poolest on After Effects ja Hitfilm Pro sarnased, kuid After Effectsi eelis on ühilduvuses teiste Adobe programmidega, mis tähendab, et ta võib üle kanda Adobe Premiere projekte, et neid ise kasutada, salvestades klippide asukohad ajateljel. Lisaks sellele on After Effectsil suurem kasutajabaas, mis tähendab, et sellele on loodud rohkem pistikprogramme, paremad juhendid internetis ning rahvarohked foorumid, kes aitavad probleemide tekkimisel.

3.2.3 After Effects on parim valik animatsioonide loomiseks

Kokkuvõtteks, miks After Effects sai valitud, seisneb selles, mida kasutajad soovivad saavutada. Tegeledes animatsioonide loomise ja kasutamisega, on võimalik neid

enamustest monteerimisprogrammides luua, kuid mitte professionaalsel tasemel. Selleks on targem kasutada spetsialiseerunud programme, mis on mõeldud nende haldamiseks ja nende hulgast tõuseb esile kõige rohkem After Effects. See on loodud, et võtmekaadrite ja teiste efektidega tegeleda ning kuigi klippide liitmine on kohmakas, siis see ei ole antud programmi eesmärk. Lisaks on see algajale animatsiooni loojale mugavam, sest After Effectsi kohta on kõige rohkem dokumentatsiooni, et programmi lihtsamini mõista.

4 Eksisteerivad lahendused

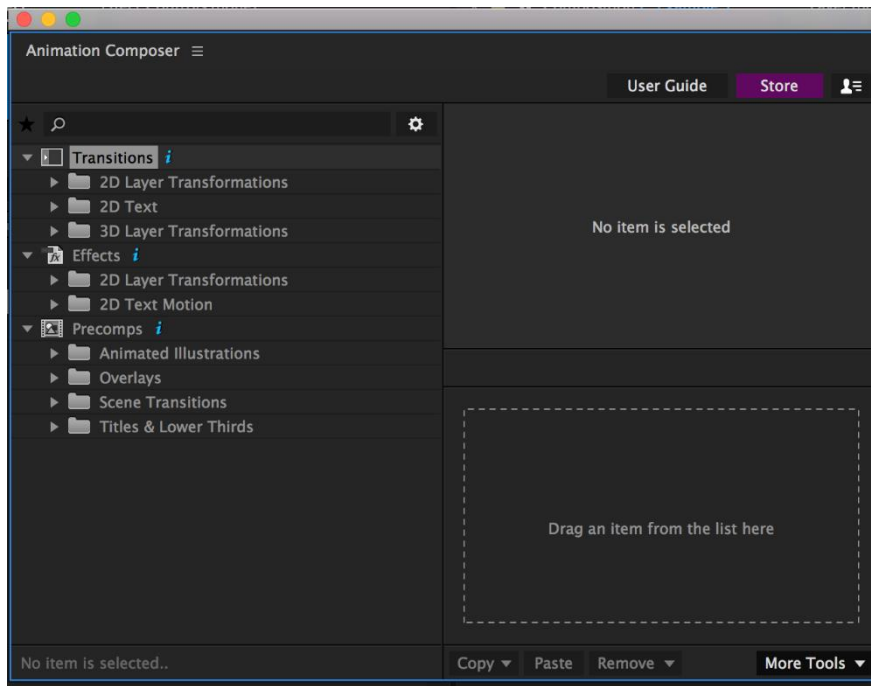
Teades, et pühendume Adobe After Effectsile, saame uurida, mis lahendused hetkel eksisteerivad sellele programmile, et kiirendada animatsioonide loomist ja taaskasutamist. Selles peatükis analüüsitakse erinevaid internetist leitud pistikprogramme, skripte ja sisseehitatud võimalusi, mida After Effects ise pakub, et teada kas sarnane lahendus on juba olemas või inspiratsiooniks, mida töö käigus arendatud lahendus peaks sisaldama.

4.1 Pistikprogrammid

Järgnevalt nimetatud on mõned tuntumad pistikprogrammid, mida saab endale soetada kas tasuta või raha eest. Alljärgnevad programmid on valitud nende võimekuse poolest võtmekaadritega tegeleda.

- Mister Horse [6]

Programm, millel on eelnevalt loodud animatsioonid. Neid animatsioone saab klippidele lisada, kuid ei ole võimalik animatsioone muuta või eemaldada. Pistikprogramm on tasuline, kuid pakutakse tasuta versiooni vähemate animatsioonidega. Joonisel 4 on võimalik näha kasutajaliidest, mis koosneb kaustadesüsteemist, hoides kõiki animatsioone, eelvaate aknast, mis kuvab valitud animatsiooni kasutajale enne klipile lisamist ja lisamise aknast, kuhu animatsioonid tirida, et nad klipile lisataks.



Joonis 4. *Mister Horse* pistikprogramm

- Motion 2 [7]

Pistikprogramm, joonis 5, mis automatiseerib võtmekaadrite kasutamise ja aitab dünaamilise liikumisega ehk muudab animatsioonid rohkem realistlikumaks ja sujuvamaks, lisades talle vahepealseid võtmekaadreid.



Joonis 5. *Motion 2* pistikprogramm

- LayerMonkey [8]

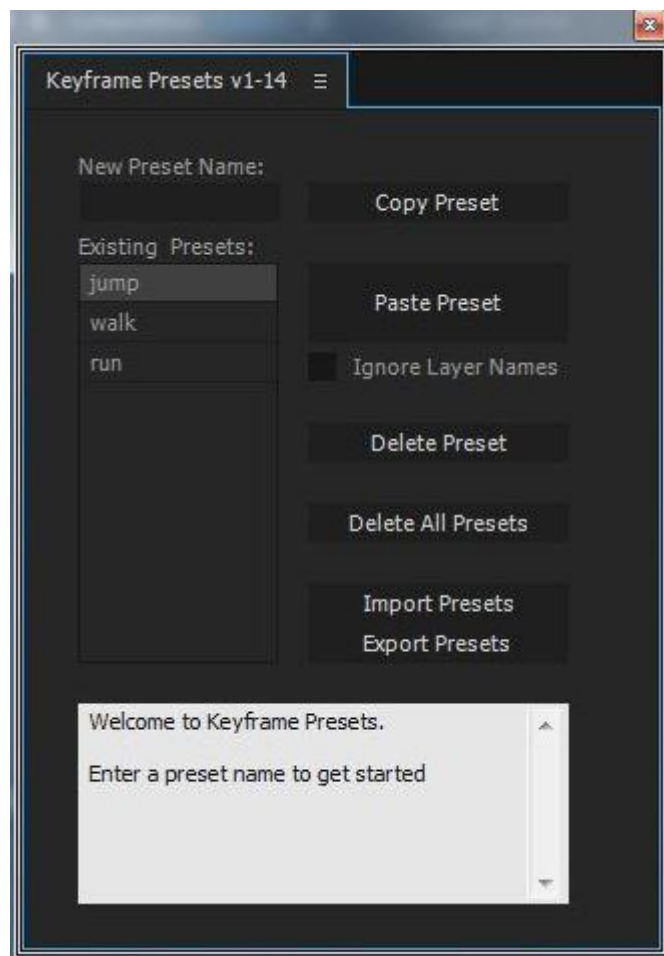
Skript, mis laseb kasutajal kompositsiooni paika panna ja animeerib selle ise vastavalt soovitud algoritmidele. Joonis 6 peal on võimalik näha, et kõige esimene samm on algoritmi valik, mis määrab kuidas animatsioonid luuakse ja ülejäänud valikud eksisteerivad, et animatsiooni paremini kohandada.



Joonis 6. Layermonkey pistikprogramm

- Keyframe Presets [9]

Pistikprogramm, mille eesmärk on võtmekaadreid salvestada ja taaskasutada. See on kõige sarnasem soovitud lahendusele, kuid mitte piisavalt efektiivne ja liiga aega nõudev, et kasutada. Joonis 7 pealt on näha, et see sisaldab samu funktsioone, mis soovitud lahendus, kuid ka palju rohkematki, mis muudab pistikprogrammi kohmakamaks ja raskemaks hallata.



Joonis 7. *Keyframe Presets* pistikprogramm

4.1.1 Pistikprogrammide plussid

Pistikprogrammidel on mitu head külge. Enamjaolt on animatsioonid eelnevalt loodud, nii et kasutaja ei pea ise neid välja mõtlema ja saab hakata koheselt neid klippide peal rakendama. Need animatsioonid näevad head välja, sest on loodud professionaalide poolt ja ei pea muretsema enda disaini oskuste pärast. Eelnevalt toodud näidetest, Mister Horse seda teebki.

Lisaks sisaldavad mõned pistikprogrammid eelvaate aknaid, et kasutaja näeks enne klipile lisamist, missugune animatsioon välja paistab. See hoiab aega kokku, sest kaotab vajaduse ise kõik animatsioonid enne läbi proovida, et teada millised nad välja näevad.

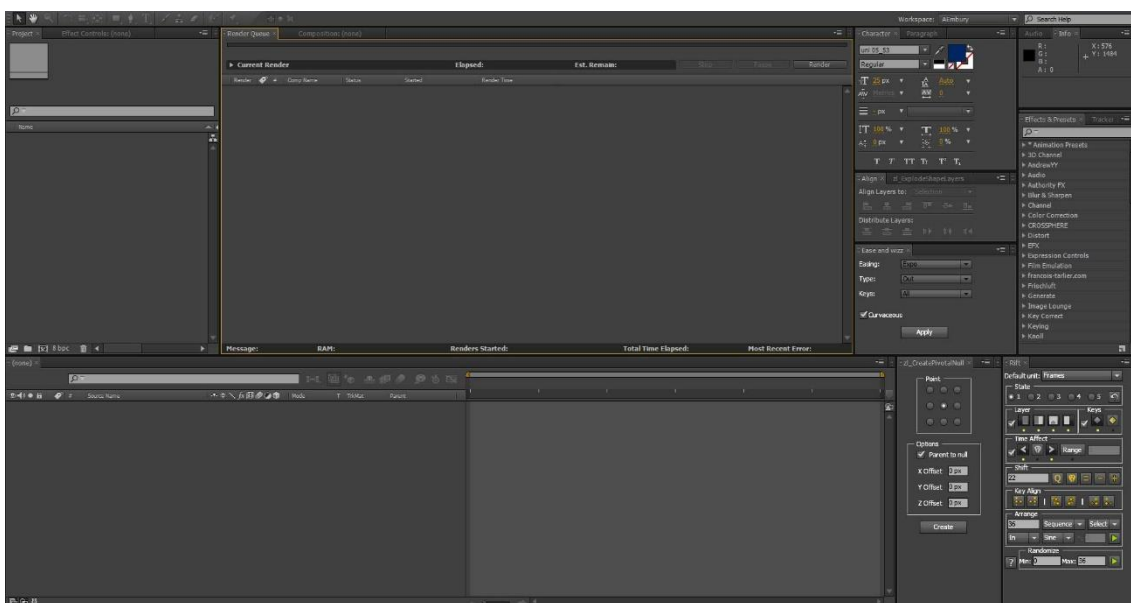
4.1.2 Pistikprogrammide miinused

Peamised probleemid, mis pistikprogrammidel esinevad on, et ei saa kohandatud animatsioone luua. Teatud hetkest alates ei ole võimalik enam samu animatsioone

kasutada, sest nad muutuvad igavaks, uued kliendid soovivad unikaalseid animatsioone või need ei sobi video stiiliga piisavalt hästi kokku. Selle jaoks on kaks lahendust, kas ise uusi animatsioone hakata looma, pistikprogrammi väliselt või leida uus pistikprogramm, värskete animatsioonidega. Sellega aga kaasnevad omaette probleemid.

Pistikprogrammid võtavad ekraaniruumi, mis on täis erinevaid aknaid, mõeldud kliipi töötlemiseks ja efektide lisamiseks. Juba üks ekstra pistikprogramm sööb ruumi, mida tähtsamad aknad kasutavad. Juhul kui on kasutuses rohkem, kui üks animatsiooni loomise programmidest, siis hakkab ruumi puudus tõsiseks probleemiks muutuma. Rääkimata, et need ei ole odavad ja iga projekti jaoks uue ostmine ei ole mõistlik.

Joonis 8 peal on võimalik näha, milline standardne After Effectsi töökeskkond välja paistab. Aknaid on võimalik üksteise taha lisada vahekaartidena, et ruumi kokku hoida, kuid sellega kaasneb üks lisa samm, kui soovitakse neid esile tõsta.



Joonis 8. Adobe After Effectsi tööaken erinevate moodulitega (10 nähtavat akent ja 5 vahekaartidena)

4.2 Sisseehitatud võimalused

After Effectsile on loodud sisseehitatud võimalus animatsioonide salvestamiseks manuaalselt [10], kuid terve selle protsessi läbimine on tülikas ja aeganõudev ning salvestades palju erinevaid animatsioone, muutub nende haldamine keerukaks.

Kõigepealt on vaja võtmekaadrid luua ja siis parema hiireklõpsuga vajutada, et soovitakse salvestada. Peale seda lisatakse salvestatud animatsioon efektide paneeli, eraldi kausta.

Keerulisemate animatsioonide puhul on selle protsessi läbimine mõttekas, kuid lihtsamate võtmekaadrite loomisel on neid kiirem käsitsi nullist teha, kui hakata salvestatud animatsiooni eraldi aknast otsima ja klipil kasutama.

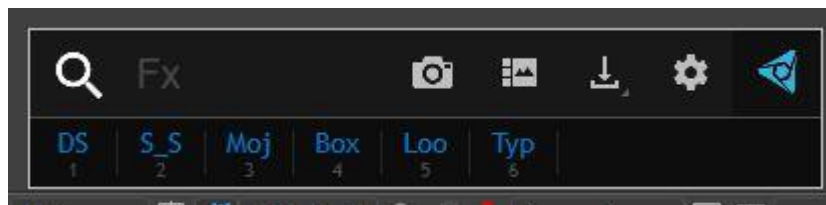
Teine sisseehitatud võimalus on kasutada efektipõhiseid animatsioone, mis vajavad natukene manuaalset tööd, kuid ei vaja animatsiooni loomise oskust [11]. Need on leitavad efektide paneelist ja saab kasutada klippide peal sama moodi nagu tavalisi efekte. Probleem nendega on, et nad ei näe professionaalsed välja ja tekitavad klipile ainult ülemineku, kuid ei kontrolli ühtegi klipi parameetrit võtmekaadritega.

5 Kas võtmekaadrite taaskasutamise vastu on vajadus?

Hetkel ei ole sellist töökiirust tõstvat rakendust olemas After Effectsile. Eksisteerivad pistikprogrammid, mis kasutavad eeltehtud animatsioone ja programm ise pakub võtmekaadrite salvestamise funktsionaalsust, kuid kui olemasolevad meetodid on nii ebaefektiivsed, siis miks ei ole midagi sarnast loodud ja kas on üldse soovi selle vastu?

5.1 FX Console

2016 tuli välja tasuta pistikprogramm After Effectsile, nimega *FX Console* [12].



Joonis 9. FX Console aken otsinguaknaga ja kiirklahvidega

Pistikprogramm oli sisu poolest lihtne. Kasutaja valib klippe, mille peale soovib efekti lisada. Vajutades klaviatuuril *ctrl + space*, tekitab see kiirklahvide kombinatsioon klippe kohale väikese akna otsingukastiga, kuhu on võimalik kirjutada soovitud efekti nimi ja programm kuvab kõik efektid, mis sellele nimele vastavad.

Efektide lisamine on sisseehitatud funktsioon ja ei vaja pistikprogramme. Sama funktsionaalsus on saavutatav efektide paneeliga, kus on isegi olemas enda otsingukast ja lisaks kõik eksisteerivad efektid väljakuvatud kujul. Kohati on *FX Console* isegi halb, eriti algajale, sest selle kasutamine eeldab efektide nimede peast teadmist. Positiivne külg tuleb aga sellest, et ta suurendab töökiirust märgatavalt.

Vanasti, et efekt klipile lisada, oli vaja see kõigepealt üles otsida, kas efektide paneelist või rippmenüüst. Kogu protsess võis võtta 5 - 15 sekundit, olenevalt kui hästi kasutaja menüüde süsteemi teab ning palju efekte on alla laetud. Nüüd aga, *FX Console* abiga, on efektide lisamine protsess, mille saab sooritada mõttekiirusel. Teades, eelnevalt efekti nime, võtab selle lisamine alla sekundi, sest ei pea hiirt kasutama ja piisab ainult paarist nupu kombinatsioonist.

Pistikprogramm ei saavuta mitte midagi uut, kuid ta suurendab üleüldist töökiirust. Enne seda aga ei eksisteerinud midagi sarnast ja keegi ei teadnud seda küsidagi.

5.2 KF Console

FX Console eesmärk oli efektide lisamine klippidele. Selle töö käigus valminud skripti, *KF Console*, eesmärk on aga animatsioonide lisamine klippidele. See aitab enamjaolt inimesi, kes tegelevad reklaamidega, graafilise disaini, muusikavideotega, või muud sorti tööga, mis vajavad eraldiseisva teksti või piltide liikuma panemist.

KF Console ei lahenda kõiki probleeme ning selle efektiivsus sõltub inimesest endast, sest alguses ei ole ühtegi eelnevalt programmeeritud animatsiooni ja kasutaja peab ise nullist neid sinna lisama. See aga omakorda eeldab, et kasutajal on piisavad teadmised animatsioonide loomiseks, sest tulemuse kvaliteet sõltub oskustest.

5.3 Kaasmonteerijad

Lõputöö eesmärk oli luua lahendus mille vastu autoril oli kindel vajadus, kuid oluline on teada, mida teised monteerijad sellest arvavad. Kas neil on sarnased probleemid või on leidnud ise mingeid lahendusi? Selle jaoks sai küsitud kolme monteerija käest, kes igapäevaselt videode monteerimisega tegelevad ja pühenduvad peamiselt animatsioonide loomisele, kuidas nad võtmekaadrite loomise ja kasutamisega toime tulevad.

Küsitud sai, kas nad kasutavad animatsioonide loomiseks pistikprogramme või teevad seda käsitsi. Vastuseks tuli enamusest, et pea aegu mitte keegi ei kasuta pistikprogramme ja kõik tehakse manuaalselt. Peamiselt, sest pistikprogrammid on kas hinnaklassist väljaspool või liiga piiravad. Monteerijad soovivad võimu selle üle, mis nad loovad ning eelnevalt ette tehtud animatsioonid ei näe välja piisavalt professionaalsed, sest need on ülekasutatud, ehk vaatajad võivad hakata märkama, et neid animatsioone kasutatakse teiste inimeste videodes ja video enda terviklikkus kannatab või etteantud animatsioonid ei rahulda stiilseid vajadusi.

Kasutatavad lahendused võtmekaadrite taaskasutamise jaoks olid algelised. Kui tekkis vajadus luua sarnane animatsioon, mis juba varem eksisteeris, siis kas tehakse see nullist valmis või kui on tegemist projektiga, kus neid võtmekaadreid eelnevalt mingi klipi peal sai kasutatud, siis nad kopeeritakse ja kleebitakse uuele klipile.

Monteerijad väitsid, et kogu protsess on aeganõudev ja vajaks kindlasti automatiseerimist, kuid ei ole piisavalt pakiline, et oleks ise viitsitud uusi lahendusi otsida. *KF Console* vastu aga näidati huvi.

6 Lahenduse arendus

Selles peatükis võtab autor läbi erinevad meetodid, kuidas arendada After Effectsi jaoks pistikprogramme ja annab ülevaate enda programmi arenduskäigust.

6.1 Arendamisel kasutatud tehnoloogiad ja võimalused

Alustades programmeerimist, oli enamjaolt teada, milline lõplik rakendus peaks välja nägema, kuid mitte, kuidas see on saavutatav. Lugeses Adobe enda dokumentatsiooni rakenduste arendamise kohta [13], on 3 erinevat lähenemisviisi soovitud tulemuse saavutamiseks, mis kõik on erinevad, omaette plusside ja miinustega.

6.1.1 Pistikprogramm

Esimene ja kõige otsesem võimalus on luua pistikprogramm kasutades Adobe enda SDKd (Standard Development Kit). Selleks on neil kodulehel link, kus seda saab tasuta alla laadida ning minimaalne dokumentatsioon, et arendamist alustada [14]. Koodi kirjutamine käib C++ keeles ja vajab Microsoft Visual Studiot, mis on programmeerimiskeskond, et Adobe pistikprogramme kirjutada ja jooksutada.

Peale SDK juhendi läbi lugemist oli arusaadav, et algaja jaoks, kes ei ole palju kokku puutunud C++ keelega, ei ole info piisav ja internetist seda juurde otsides, ei olnud samuti palju kohti, kus saaks algaja pihta hakata. Pooled näiteprogrammid, mis SDKga kaasa tulid, ei käivitunud ja andsid veateateid, sest need olid loodud aegunud After Effects versiooni peal ja koodi ei oldud uuendatud, et töötaks programmi 2018 versiooniga. Terve protsess oli keerukas ja lugeses infot foorumitest, tuli esile, et niimoodi pistikprogrammide loomine oli ka aegunud meetod ning eksisteerib lihtsamaid lahendusi, mis võtavad vähem aega ja on modernsemad.

6.1.2 HTML5 paneel

Teine võimalus oli luua lahendus HTML5 paneelina, mis tähendab et programmeerimine toimuks kasutades HTMLi (Hypertext Markup Language), CSSi (Cascading Style Sheets) ja Javascripti. See oleks sarnane veebilehe loomisele.

Miinused, mis selle proovimisega koheselt selgeks sai oli, et dokumentatsioon on pea aegu olematu [15]. HTML5 paneelid on äsja kasutusele võetud viimase 3 After Effectsi versiooni käigus ja täieliku ametliku dokumentatsiooni ei ole jõutud luua. Rääkimata sellest, et luua rakendus, mida igäüks saab kasutada, peab tegelema autorisatsiooni probleemidega [16], sest pistikprogramm on ühendatud internetiga ja pahatahtlike programmide loomine oleks lihtne. See aga tähendab, et tavakasutaja ei saa seda hakata prototüübi vormis testimagi.

6.1.3 Skript

Viimane meetod, mida valdav enamus isehakanaud After Effectsi rakenduste loojatest kasutab, on skriptide loomine. Skriptid kirjutatakse Extendscriptis, mis põhineb Javascriptile ja selle jooksumisest piisab, kui see ainult õigesse lahtrisse tõsta.

Üks peamistest muredest oli algselt, et puudus info skriptimise võimekusest ja kas soovitud funktsionaalsus on selle lahendusega saavutatav. Autori kokkupuude skriptide loomisega on olnud minimaalne ja puudus teadmine, kas skriptiga on üldse võimalik luua mooduli, mida annab paigaldada After Effectsi aknasse, mis suudab küsida infot teistest moodulitest, selle salvestada ja hiljem rakendada klippide peal [17]. Lõpuks sai otsustatud selle kasuks lahenduse arendamise puhul, sest selle kohta eksisteeris kõige rohkem abimaterjali ja ExtendScript ei olnud keerukas selgeks saada.

6.2 Arendamisel kasutatud tööriistad

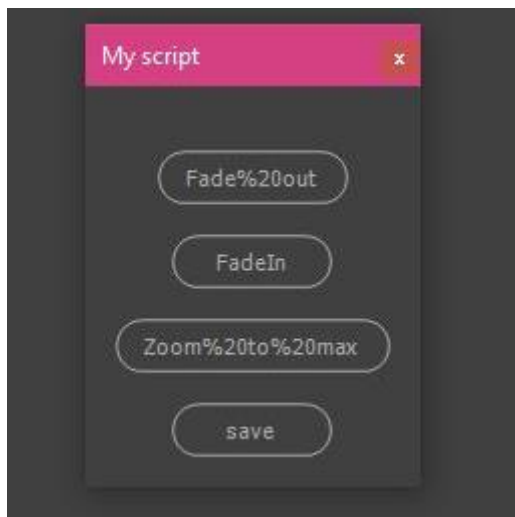
Lahenduse arendamiseks kasutati Adobe sisseehitatud tekstiredaktorit [18], *The ExtendScript Toolkiti*, mis aitab luua, muuta ja siluda eksisteerivaid skripte. See ei ole mõeldud ainult After Effectsi jaoks, vaid kasutatakse samuti teistes Adobe programmides. Nagu näiteks Adobe Photoshopis või Adobe Illustratoris. Skriptid kirjutatakse ExtendScript keeles, mis on tuletatud Javascriptist.

6.3 Versioonid

Lahendus sai arendatud etapiviisiliselt. Kõigepealt katsetades, kas on võimalik saada kätte peamine funktsionaalsus ja siis lisades täiendusi, mis aitasid kaasa efektiivsusele.

6.3.1 Versioon 1

Esialgne plaan oli luua aken, mida annab paigaldada After Effectsis teiste akende kõrvale nii, et see on pidevalt ees ja sellele on kohene ligipääs olemas. Peamised funktsioonid, mis oli vaja tööle saada oli võtmekaadrite salvestamine ja salvestatud animatsioonide lisamine klippidele. Joonis 10 pealt on võimalik näha kõige esialgsem kuju, mis sisaldas ainult salvestamise nuppu ja animatsioonide lisamise nuppe.



Joonis 10. Lahenduse esialgne kuju

Üks esimestest funktsionaalsustest, mis oli vaja tööle saada, oli võtmekaadrite salvestamine, sest see vajab erinevate parameetrite salvestamist teksti kujule ja sellest võis järeldada, kui võimekad on skriptid. Peale pikaajalist otsimist ei olnud võimalik leida elegantset viisi selle sooritamiseks, kuid see oli saavutatav sisseehitatud funktsiooniga mida on võimalik näha joonis 11 pealt.

Iga kord kui vajutatakse salvestamise nuppu, võtmekaadrid salvestatakse etteantud lahtrisse iseseisva failina.

```
saveButton.onClick = function()
{
    app.executeCommand(3075);
}
```

Joonis 11. Võtmekaadrite salvestamise funktsioon

Lahenduse akna avamisel käib ta läbi lahtri, kus salvestatud animatsioonid paiknevad ja loob iga faili kohta ühe nupu. Faili nime järgi saab nupp enda nime ning selle peale vajutades, *KF Console* rakendab seda eelvalitud klipile peal. Joonis 12 pealt on võimalik näha koodijuppi, mis läheb tööle nupule vajutamiseks.

```
function applyMyPreset()
{
    mainLayer.applyPreset(files[this.name]);
    mainWindow.close();
}
```

Joonis 12. Võtmekaadri klipile lisamise funktsioon

6.3.2 Versioon 2

Saavutades peamise funktsionaalsuse esimese versiooniga ja teades, et lahendus on tehtav skriptimisega, oli võimalik mõelda paranduste ja muu funktsionaalsuse peale, mis suurendab töökiirust veel rohkem. Sellega jõuti järeldusele, et eraldiseisev aken, mis on pidevalt ees, ei tule kasutajale kasuks, vaid pigem hakkab segama. Praktilise poole pealt ei tule see isegi kiirem kui sisseehitatud efektide aknast salvestatud animatsioonide väljakutsumine. Kõige mõistlikum lahendus oli luua aken, mis avaneb ainult siis, kui seda on vaja kasutada, sarnaselt olemasolevale pistikprogrammile, *FX Console*.

Lisades funktsionaalsust juurde, oli tähtis et seda liiga palju ei tekiks, sest muidu on risk, et need muutuvad pigem segavaks kui kasulikuks. Eesmärk on lahendust minimalistlikuna hoida. Peamised asjad mis skriptil veel vaja oli, oli loodud võtmekaadrite kustutamise nupp ja nime muutmise nupp, juhul kui soovitakse animatsioone ümber nimetada või täielikult kaotada, et nad rakenduse aknas ruumi ei raiskaks.

Töökiiruse omakorda tõstmiseks, sai lisatud veel kiirklahvi võimalus. See tähendab, et igale animatsioonile on võimalik oma enda kiirklahvi lisada ja rakenduse akna avamisel,

vajutades eelnevalt määratud klahvi klaviatuuril, lisatakse animatsioon koheselt klipile, kahandades terve protsessi ajakulu vähemaks kui üks sekund.

Kiirklahvid on lahenduse üks tähtsamaid elemente, sest need võimaldavad animatsioone lisada välkkiiruselt. Kiirklahvide lisamine aga ei tohi olla keerukas või liiga aega nõudev tegevus, sest animatsioonid vahetuvad pidevalt ja vastavalt nendele soovitakse kiirklahve muuta. Selle tegevuse jaoks on eraldiseisev nupp, millele vajutades avaneb uus aken, kus vajutades nupule klaviatuuril, köidetakse see animatsiooniga. Valitud kiirklahv salvestatakse eraldi tekstifaili koos animatsiooni nimega, mis käitub andmebaasina kõikide kiirklahvide kombinatsioonide jaoks.

Eelnevaga seoses oli vajalik animatsiooni kustutamise ja ümber nimetamise funktsioonid uuendada, et nad arvestaksid andmebaasi olemasolu ja teeksid seal vastavaid muudatusi.

6.3.3 Versioon 3

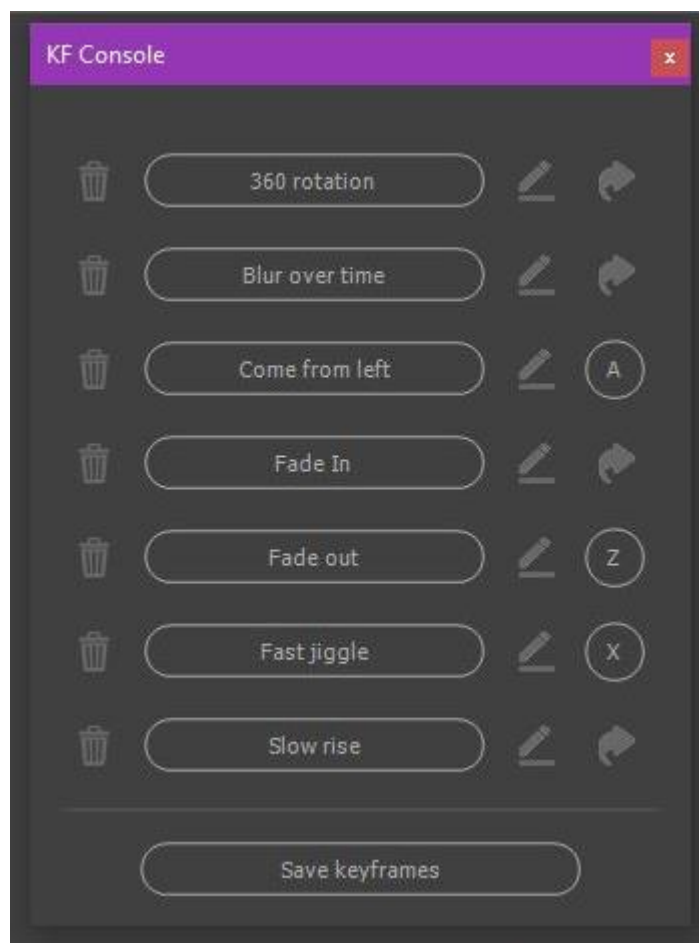
Peale vajaliku funktsionaalsuse saavutamist oli vaja tegeleda disainiga, sest After Effectsi enda vaikimisi kujundus on ruumi raiskav ja kehv kasutajale. Info selle kohta oli aga minimaalne ja paljud foorumid viitasid veebilehtedele ja allikatele, mis ei ole hetkel enam kättesaadavad. Õnneks oli loodud üks mitteametlik dokument, kus paiknes kõik vajalik, et teha lahendus kasutajasõbralikumaks [19].

Ruumi kokkuhoiu ja selgema kasutajaliidese loomiseks sai teatud nupud asendatud ikoonidega, selle asemel et nad kuvaks välja mida teevad. Extendscriptil on sisseehitatud ikoonid, kuid need olid puudulikud ja sobivad pildid tuli ise luua ja lisada skripti kausta.

Lõpptulemus on nähtav joonis 7 pealt, kus kõige vasakpoolsem nupp on animatsiooni kustutamiseks. Keskmise nupu peale vajutamine rakendab animatsiooni klipil. Sellest parempoolsem laseb animatsiooni ümber nimetada ja kõige parempoolsem nupp on kiirklahvide jaoks, mis kuvab välja olemasoleva kiirklahvi, kui see on lisatud. Vastasel juhul näitab ikooni, kuni kiirklahv lisatakse.

7 Valminud rakendus

Valminud lahendus, joonis 13, on lihtne kasutada ja vajab umbes 5 minutit ülesseadmiseks. Siinses peatükis läbime esimesed sammud, mida tavakasutaja peab tegema, et skript valmis seada ning võrdleme seda teiste olemasolevate lahendustega, et näha, kui palju see monteerimisprotsessi tegelikult kiirendab.



Joonis 13. *KF Console* lõppkuju

7.1 Esmakordne ülesseadmine

Lahendus on loodud skripti failina, mis tähendab, et see ei vaja installimist. Tööle saamiseks peab skripti faili tõstma õigesse kausta mis asub kaustas:

(Windows)<drive>\Program Files\Adobe\Adobe After Effects CC <version>\Support Files\Scripts

Ainukene probleem, mida ei saa lahendada lihtsalt on see, et After Effects ei lase kiirklahve programmisel lisada või muuta. Lahenduse efektiivsus aga sõltub suuremahuliselt sellest, kui kiiresti on seda võimalik välja kutsuda ja selleks on vaja anda talle kiirklahv. Selle lisamiseks on vaja minna kausta:

(Windows)<drive>\Users\<username>\AppData\Roaming\Adobe\After Effects\<version>

Otsida fail, mille pealkirjas on sõne *shortcuts* ja enda vabalt valitud tekstiredaktoriga fail avada, otsida välja rida *ExecuteScriptMenuItem01* ning enda nupu kombinatsioon sinna kirjutada [20].

7.2 Rakenduse kasutamine

Olles loonud võtmekaadrid ja soovides need salvestada, tuleb hiirega nendest üle tõmmata, avada *KF Console* ja vajutada nuppu *save keyframes*. Seejärel saab anda animatsioonile nime ning see salvestatakse arvutisse.

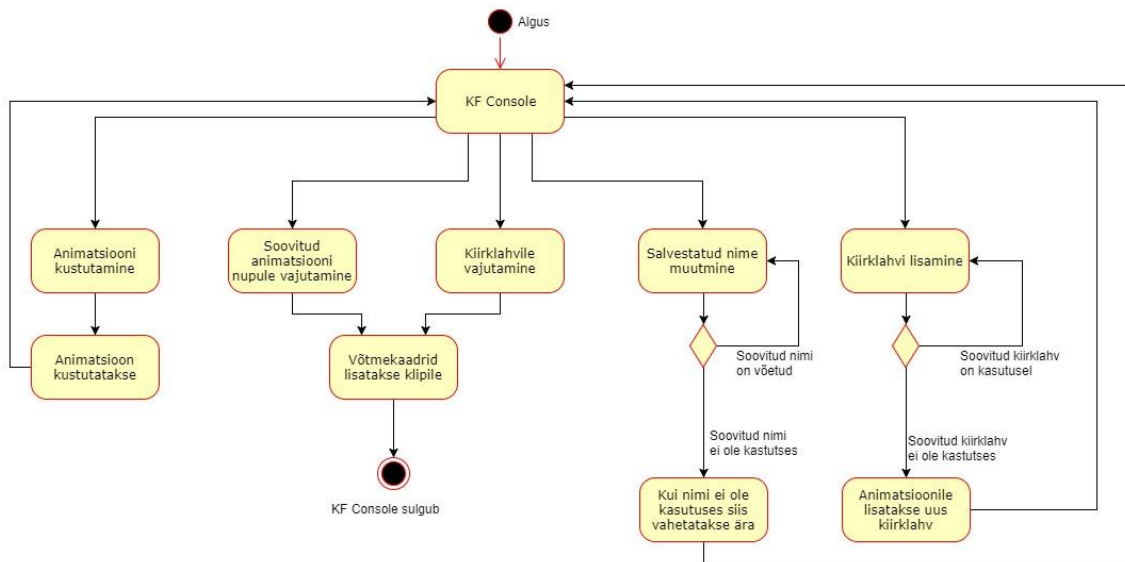
Rakendus aga ei piirdu ainult ühe paari võtmekaadrite salvestamisest vaid see võib korraga salvestada mitu erinevat parameetrit. Näiteks lisaks animatsioonile, kus pilt ainult liigub, võib salvestada suuruse parameetrid, nii et pilt liigub ja vahetab suurust sama animatsiooni jooksul. Lisaks sellele, kui on klipile lisatud efekte võtmekaadritega, siis ta jätab ka need meelde ja hiljem lisab väljavalitud klipile kõik parameetrid koos efektiga.

Et rakendada salvestatud animatsioone klipil, on kaks erinevat võimalust. Esimene on nupule vajutamisega ja teine on kasutades kiirklahvi. Animatsioonide puhul, mida kasutatakse tihedamalt, on soovitatud siduda kiirklahviga, sest see on kiirem, kui hiirega nupu välja valimine, et sellel vajutada.

Peale soovitud animatsiooni nupule vajutamist ilmuvad soovitud võtmekaadrid klipil nagu tavaliselt ning neid on võimalik edasi kujundada või ajateljel liigutada ja *KF Console* aken sulgeb iseennast.

Animatsiooni lisamine ei piirdu aga ühe klipiga. Kui on valitud rohkem, kui üks klipp, siis *KF Console* rakendab väljavalitud animatsiooni kõikide klippide peal korraga.

Kõik tegevused ja nende tulemused alates *KF console* väljakutsumisest, on nähtavad joonis 14 peal.

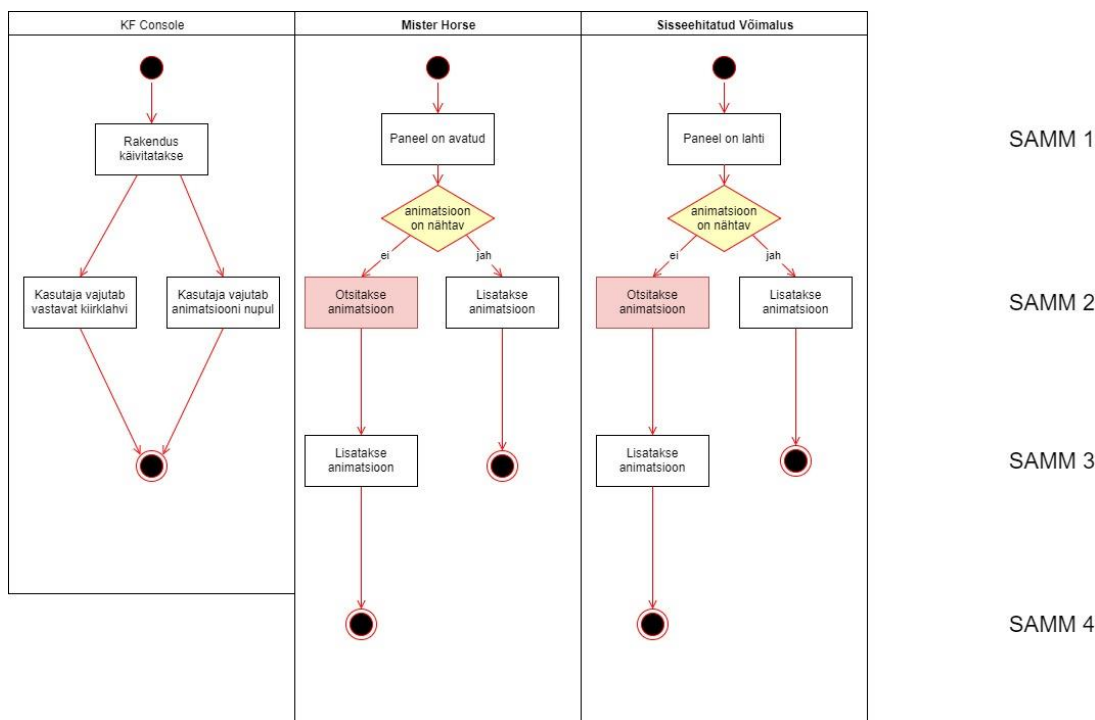


Joonis 14. KF Console tegevuskeem

7.3 Ajaline võrdlus teiste rakendustega

Mis eristab olemasolevaid lahendusi *KF Consolest* on see, et lahendustel on etteantud animatsioonid ja need kaotavad unikaalsuse võlu. Lisaks võtavad nad väärtuslikku ruumi ekraanil mida on võimalik paremini kasutada.

KF Consolega on võimalus enda loomingulisus salvestada ja seda taaskasutada ilma ruumi raiskamata ja saavutades selle kiiremini kui ühegi teise pistikprogrammiga. Kiirus tuleneb sellest, et on vähem samme mida peab läbima animatsiooni lisamiseks. Joonis 15 pealt on võimalik näha, kuidas loodud lahendust võrreldakse *Mister Horse* pistikprogrammiga ja sisseehitatud lahendusega. Kõige optimaalsemal juhul on animatsioonide lisamine sama kiire igal lahendusel, kuid enamustel juhtudest tuleb üks samm juurde, mis on õige animatsiooni otsimine. See on kõige aeganõudvam etapp, sest vajab animatsioonide otsimist kaustade struktuurist ja tihtipeale tuleb seda korrata erinevate animatsioonide otsimise puhul.



Joonis 15. Animatsiooni lisamise etapid erinevate meetoditega

Demonstreerimaks, kuidas *KF Console* võrdleb *Mister Horse* pistikprogrammiga ja sisseehitatud funktsionaalsusega ajaliselt, tegi autor iga lahendusega täpselt sama katse, mis oli 3 pildi animeerima panek.

7.3.1 Manuaalselt animatsioonide tegemine

Tehes manuaalselt animatsioonid nullist, läks umbes 21 sekundit. Sinna alla kuulub esialgse animatsiooni loomine, selle kopeerimine ja kleepimine järgmisele kahele klipile. Demonstratsioonis sai kasutatud kiirklahve, et parameetreid kuvada ja animatsiooni hõlpsamini luua.

Hea külg manuaalse animatsiooni tegemisel on see, et säilib kontroll kõige üle. Igat pisemat liigutust on võimalik hiljem muuta.

Negatiivne pool seisneb selles, et see on ajamahukas. Hetkel on demonstratsiooniks toodud üks lihtsamatest animatsioonidest, kuid kui soovitakse muuta lisaks positsioonile selle suurust või tehes liikumist natukenegi keerulisemaks, võtab loomine kordades rohkem aega. Võrreldes teise kahe lahendusega oli käsitsi tegemine juba praegugi liiga ajakulukas.

7.3.2 Eelloodud pistikprogrammi kasutamine

Demonstratsiooniks, kuidas pistikprogramid aitavad animatsioonidega, sai *valitud Mister Horse*, mida autor kasutab pea-aegu igapäevaselt ja on tuttav selle võimekusega. Animatsioonide lisamine võttis aega umbes 7 sekundit ja kasutamine seisneb selles, et valitakse soovitud klipid ning eraldi aknas tõmmatakse animatsioon õigesse lahtrisse, mis lisab võtmekaadrid automaatselt klippidele.

Hea külg on selles, et animatsioonid on eelnevalt loodud. Need on professionaalsel tasemel nii, et kui kasutaja ei oska animeerida siis tulemust on ikkagi meeldiv vaadata. Lisaks sellele on animatsioonide lisamine kiire, neid võib lisada mitmele klipile korraga ja nad on dünaamilised, ehk muudavad enda pikkust ise, vastavalt klipi algusele või lõpule.

Negatiivne pool seisneb aga selles, et see rakendus võtab ekraanil tohutult ruumi, sest see sisaldab endas mitmeid aknaid, mis on mõeldud animatsiooni kuvamiseks ja kaustade struktuuri hoidmiseks. Lisaks peale pikemat kasutust hakkas see muutuma limiteerivaks, sest uusi animatsioone ei ole võimalik lisada ja samade kasutamine ei näe peale tükk aega enam hea välja. Nagu näha demonstratsiooni puhul siis ei olnud võimalik saavutada sama animatsioon, mis teise kahe lahendusega, sest selline variant üldse puudus. loodud animatsioon on sarnane, kuid see ei ole üks-ühele sama.

7.3.3 *KF Console*

Kasutades töö käigus loodud skripti võtab animatsiooni lisamine 5 sekundit. Seda juhul, kui on kiirklahv lisatud soovitud animatsioonile, et seda kiirelt välja kutsuda ja animatsioon ise on eelnevalt salvestatud. Arvestades esmakordset animatsiooni loomist, siis oleks ajakulu suurem olnud, kuid demonstratsiooni eesmärk oli ainult animatsioonide lisamine.



Positiivne külg seisneb selles, et animatsioonide rakendamine toimub välkkiirelt, eriti kui on kiirklahv animatsioonile seatud.

Probleem aga seisneb kasutajas, sest tema peab esialgselt animatsiooni ikkagi looma ning maksimaalse tulemuse saavutamiseks kiirklahvi lisama. Lisaks sellele, animatsiooni kvaliteet sõltub kasutajast, sest midagi ei ole ette antud nagu ostetavate pistikprogrammide puhul.

7.3.4 Kokkuvõte

Etteantud demonstratsioon oli ainult väga algne näide, et saaks üldisest mõttest aru. Kui animatsioonid oleksid keerulisemad siis manuaalselt tegemine võtaks palju rohkem aega. Kõik sõltub klippide arvukusest ja animatsiooni keerukusest. Saavutatud tulemus oli aga positiivne, sest *KF Console* oli kõige kiirem lahendus. Kokkuvõtvat tulemust on võimalik näha tabelist 2.

<https://www.youtube.com/watch?v=BNOLXOgBpKw>

Meetod			KF Console
Hind	Tasuta	Tasuta – 360€	Tasuta
Kiirus	21 sekundit	7 sekundit	5 sekundit
Lihtsus	Keeruline	Lihtne	Keskpärase
Eeltöö	Ainult eeltöö ongi	Puudub	Ühekordne

Tabel 2. Meetodite võrdlus

8 Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli luua lahendus mis kiirendab töökiirust After Effectsis, lubades kasutajal taaskasutada eelnevalt loodud võtmekaadreid.

Peamine probleem seisnes selles, et võtmekaadrite loomine oli ajamahukas, olemasolevad lahendused limiteerivad ja animatsioonide loomine on protsess, mis kipub korduma enamuste projektide puhul ning vajab automatiseerimist.

Bakalaureusetöö käigus sai autor paremad teadmised, kuidas pistikprogramme valmistada ja mis teeb kasutajaliidese efektiivsemaks. *KF Console* oli rahuldav, sest võrreldes eksisteerivate lahendustega, oli see kiirem. See aga ei tähenda, et sellel ei esine probleeme. Kõik vead ja puudused tulevad ajaga, sest lühiaegne testimine ei ole piisav limitatsioonide väljaselgitamiseks. Üks probleemidest, mis võib tekkida, on see, kui lisatakse liiga palju animatsioone. Hetkel võib lisada 40 animatsiooni enne kui akna ruum otsa saab.

Töö eesmärk sai täidetud. Valmis lahendus Adobe After Effectsile, mis kiirendab võtmekaadritega töötamise aega ja on efektiivsem, kui teised eksisteerivad lahendused.

Kasutatud material

- [1] „Blender Wiki,“ [Võrgumaterjal]. Available: <https://wiki.blender.org/index.php/Doc:ET/2.6/Manual/Animation/Keyframes>. [Kasutatud 2 January 2018].
- [2] "Adorama Learning Center," [Online]. Available: <https://www.adorama.com/alc/8853/article/4-must-have-motion-graphic-design-applications>. [Accessed 12 December 2017].
- [3] „Pluralsight,“ [Võrgumaterjal]. Available: <https://www.pluralsight.com/blog/creative-professional/software-need-know-motion-graphics>. [Kasutatud 12 December 2017].
- [4] „PCMag,“ [Võrgumaterjal]. Available: <https://www.pcmag.com/article2/0,2817,2397215,00.asp>. [Kasutatud 12 December 2017].
- [5] D. O. P. M. A. E. Jarle Leirpoll, „Motion Graphics Inside Premier Pro,“ %1 *The Cool Stuff in Premiere Pro*, 2017, pp. 557-675.
- [6] „Mister Horse,“ [Võrgumaterjal]. Available: <https://misterhorse.tv/>. [Kasutatud 13 September 2017].
- [7] „Motion,“ [Võrgumaterjal]. Available: <https://www.mtmograph.com/motion/>. [Kasutatud 13 September 2017].
- [8] „LayerMonkey,“ [Võrgumaterjal]. Available: <https://aescrpts.com/layermonkey/>. [Kasutatud 13 September 2017].
- [9] „Keyframe Presets,“ [Võrgumaterjal]. Available: <https://aescrpts.com/keyframe-presets/>. [Kasutatud 2 January 2018].
- [10] „Rocketstock,“ [Võrgumaterjal]. Available: <https://www.rocketstock.com/blog/how-to-save-your-after-effects-presets/>. [Kasutatud 2 January 2018].
- [11] B. G. Lisa Fridsma, „Creating a Basic Animation Using Effects and Presets,“ %1 *Adobe After Effects CC Classroom in a Book*, 2017, pp. 53-79.
- [12] „FX Console,“ [Võrgumaterjal]. Available: <https://www.videocopilot.net/blog/2016/10/new-workflow-plugin-in-fx-console-is-now-available/>. [Kasutatud 15 November 2017].
- [13] „Adobe,“ [Võrgumaterjal]. Available: <https://www.adobe.io/apis/creativecloud/aftereffects.html>. [Kasutatud 25 September 2017].
- [14] „Adobe SDK,“ [Võrgumaterjal]. Available: <https://console.adobe.io/downloads/ae>. [Kasutatud 4 January 2018].
- [15] „HTML 5 panel guide,“ [Võrgumaterjal]. Available: https://d1g4ig3mxc5xed.cloudfront.net/static/installers/ae/cep/cc_all/doc/documentation.pdf. [Kasutatud 1 December 2017].

- [16] „HTML5 panels - Getting Started,“ [Võrgumaterjal]. Available:
] <http://victorianece.com/2015/12/getting-started-with-html5-cep-panels-for-after-effects/> . [Kasutatud 1 December 2017].
- [17] „Dockable UI panel,“ [Võrgumaterjal]. Available:
] <http://aenhancers.com/viewtopic.php?t=2263>. [Kasutatud 1 December 2017].
- [18] „Adobe,“ [Võrgumaterjal]. Available:
<http://www.adobe.com/devnet/scripting/estk.html>. [Kasutatud 28 October 2017].
- [19] P. Kahrel, „ScriptUI,“ 2015. [Võrgumaterjal]. Available: <http://fotozcech.cz/wp-content/uploads/2015/11/scriptui-2-8.pdf>. [Kasutatud 20 December 2017].
- [20] „AEScripts,“ [Võrgumaterjal]. Available:
] <https://aescrpts.com/knowledgebase/index/view/faq/use-keyboard-shortcut/>.
[Kasutatud 3 January 2018].
- [21] „Red Giant,“ [Võrgumaterjal]. Available: <https://www.redgiant.com/universe/>.
] [Kasutatud 13 September 2017].

Lisa 1 – Koodi asukoht

<https://github.com/MuffinDude/KF-Console>