

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Helina Kruuk 179546IAIB

**ÕIGUSAKTI VISUALISEERIMINE
KOLMEMÕÕTMELISE
PUUSTRUKTUURINA**

bakalaureusetöö

Juhendaja: Ermo Täks
PhD

Tallinn 2020

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: Helina Kruuk

18.05.2020

Annotatsioon

Käesoleva bakalaureusetöö eesmärk on luua veebirakendus, mis võimaldab visualiseerida õigusakti tervikteksti ning terviktekstis tehtud muudatusi kolmemõõtmelise puustruktuurina.

Töö esimeses osas kirjeldatakse andmete visualiseerimise põhimõtet ning kuidas on seda rakendatud õigusaktide visualiseerimiseks. Järgnevas osas antakse ülevaade töös kasutatud tehnoloogiatest ning nende valiku põhjustest. Kahes viimases osas kirjeldatakse veebirakenduse valmimise protsessi, valideerimist ning võimalikke edasiarendusi.

Töö tulemusena valmis veebirakendus, millega saab visualiseerida õigusakti.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 25 leheküljel, 6 peatükki, 13 joonist, 1 tabelit.

Abstract

Legal act visualization as a three-dimensional tree

In the Republic of Estonia, the central database of legislation is the information system called the Riigi Teataja. The comparison mode in the Riigi Teataja is inconvenient and time-consuming. As well changes to an already voluminous legislation can easily be overlooked. These are the main reasons that made the author create something that could make the process of looking the Estonian database of legislation easier and more understandable.

The aim of this thesis is to create a web application that allows the user to visualize complete texts of legal acts and changes made to them as a three-dimensional tree.

The work is divided into four parts. The first part of the work describes the data visualization principle and how it has been implemented to visualize legislation. The second part provides an overview of the technologies used in the work and the reasons for their selection. The last two parts focus on the process of completing the web application its validation process and also possible improvements that could be made further on.

As a result of this thesis a web application that can visualize legislation was completed.

The thesis is in Estonian and contains 25 pages of text, 6 chapters, 13 figures, 1 tables.

Lühendite ja mõistete sõnastik

<i>canvas</i>	HTML element, kuhu renderdakse 2D ja 3D graafika
<i>Fragement shader</i>	GLSL-i koodi osa pikslite värvimiseks.
GLSL	<i>OpenGL Shading Language</i> Varjutamiskeel, mis ühendab omavahel tippude ja fragmentide töötlemise.
Google Trends	Veebisait, mis analüüsib populaarsemate otsingupäringute populaarsust Google otsingus.
GPU	<i>Graphics Processing Unit</i> Graafikakaardil paiknev mikroprotsessor, mis on projekteeritud spetsiaalselt graafikainformatsiooni töötlemiseks ja kuvamiseks
HTML	<i>Hypertext Markup Language</i> Märgistuskeel veebilehtede märgendamiseks
<i>hub and spoke</i>	Diagramm, mille tsentraalne üksus on ümbritsetud teiste üksustega.
IEWebGL	<i>WebGL for Internet Explorer</i> Pistikprogramm, mis lisab Internet Exploreri WebGL-i toe
NSM	<i>Normal Simplex Models</i> Tööriistade ja rakenduste komplekt, mida Normal-Simplex projekti raames arendatakse
<i>stackoverflow</i>	Küsimuste ja vastuste veebileht programmeerijatele.
<i>Vertex shader</i>	GLSL-i koodi osa punktide loomiseks.
WebGL	Veebigraafika raamistik.
XML	<i>Extensible Markup Language</i> Märgistuskeel dokumendi struktuuri loomiseks

Sisukord

1	Sissejuhatus	10
1.1	Taust ja probleem.....	10
1.2	Eesmärk ja oodatav tulemus	11
1.3	Ülevaade tööst.....	11
2	Andmete visualiseerimine	12
2.1	Kasutusvaldkonnad	13
2.2	Õigusaktide visualiseerimine	13
2.2.1	Puukaart	14
2.2.2	Puustruktuur	15
2.2.3	Jõuga suunatud graaf	15
2.2.4	Sõnapilv	16
3	Kasutatud tehnoloogiad	17
3.1	WebGL.....	17
3.2	Three.js	17
3.2.1	Three.js valiku põhjendus.....	18
3.3	JavaScript.....	19
3.4	HTML	19
4	Veebirakenduse arendusprotsess	20
4.1	Rakenduse analüüs.....	20
4.1.1	Rakenduse nõuded.....	20
4.2	Andmed.....	21
4.2.1	Riigi Teataja võrguväljaanne.....	21
4.2.2	Andmete lugemine.....	21
4.3	Graafika kuvamine.....	22
4.4	Õigusakti puustruktuuri ehitus.....	22
4.4.1	Paragrahv ilma lõigete ja punktideta	23
4.4.2	Paragrahv ainult lõigetega	23
4.4.3	Paragrahv lõigete ja punktidega	24

4.4.4	Paragrahv ainult punktidega	24
4.4.5	Paragrahvid ülaindeksitega.....	25
4.4.6	Terviklik õigusakt veebirakenduses kujutatud puustruktuurina.....	25
4.5	Puustruktuuri loomine.....	26
4.5.1	Tüve loomine.....	26
4.5.2	Õigusakti paragrahvide joonestamine	26
4.5.3	Sihivektori leidmine	28
4.5.4	Õigusakti lõikude ja punktide joonestamine	28
4.6	Õigusakti redaktsioonides tehtud muudatused	29
4.6.1	Muudatuste leidmine	29
4.6.2	Ajaskaala loomine	29
4.6.3	Muudatuste esitamine	30
4.7	Veebirakenduse vaated	31
5	Valideerimine ja võimalikud edasiarendused.....	32
5.1	Veebirakenduse valideerimine.....	32
5.2	Võimalikud edasiarendused.....	33
5.2.1	Õigusakti tekstilise sisu kuvamine	33
5.2.2	Kõik vaated ühel kujul.....	33
5.2.3	Õigusaktide hulga laiendamine	33
6	Kokkuvõte	34
	Kasutatud kirjandus	35

Jooniste loetelu

Joonis 1. Portugali Vabariigi põhiseadus puukaardina [8].	14
Joonis 2. Ameerika Ühendriikide põhiseadus puustruktuurina [11].	15
Joonis 3. Ameerika Ühendriikide autoriõiguste koodeks jõuga suunatud graafina [11].	16
Joonis 4. Töölepingu seadus § 3	23
Joonis 5. Töölepingu seadus § 1 lõiked 1 - 5	23
Joonis 6. Töölepingu seadus § 24 lõige 1 punktid 1- 4 ja lõiked 2, 3 ja 4	24
Joonis 7. Töölepingu seadus § 19 punktid 1-6	24
Joonis 8. Töölepingu seadus § 137 ja § 137 ¹	25
Joonis 9. Töölepingu seadus	25
Joonis 10. Siglõigu lõpp-punktide pööramine	27
Joonis 11. Programmikood sirglõigule punktide leidmiseks	28
Joonis 12. Muudatuste ajaskaala	30
Joonis 13. Töölepingu seaduses tehtud muudatused	31

Tabelite loetelu

Tabel 1. Three.js ja P5.js raamistiku võrdlus	18
---	----

1 Sissejuhatus

Õigus kui üldise iseloomuga käitumisnormide kogum, mõjutab ühiskonna igat taset, seades ühiskonna-liikmetele kindlad õigused ja kohustused. Käitumisnormide süsteem rajatakse kindlate põhimõtete järgi ning koondatakse süstemaatiliselt õigusharudesse ja aktidesse. Õigusnormid vormistatakse kirjalikult ja koondatakse kindlatele nõuetele vastavasse dokumenti ehk õigusakti, mille abil reguleeritakse riigi ja kodanike omavahelisi suhteid ning korraldatakse kodanike ühiselu.

Tänapäeva ühiskond on pidevas arengus ning sellega suureneb vajadus mitmesuguste suhete õigusliku reguleerimise järele. Ühiskonna arengu käigus käitumisreeglid muutuvad ning sellega muutub ka õigusaktide sisu. Õigusakti lisatakse ridu juurde, eemaldatakse ning täiendatakse neid. Mida aeg edasi seda mahukamaks ja keerulisemaks sellised dokumendid lähevad. Samuti tuleb säilitada muudatused õigusaktide sisus ja nendega peab olema võimalik tagantjärgi tutvuda. Teada on, et inimese aju töötleb visuaalset teavet mitu korda kiiremini kui teksti. Siin kohal tuleb appi visualiseerimine, mis võimaldab lihtsustada õigusaktide sisu mõistmist ning visualiseerida õigusaktis tehtud muudatusi.

1.1 Taust ja probleem

Eesti Vabariigis on õigusaktide keskseks andmebaasiks Riigi Teataja infosüsteem. Riigi Teataja on ametlik võrguväljaanne, kus avaldatud õigusaktid saavad õigusjõu. Alates 2010. aasta 1. juunist antakse Riigi Teatajat välja ainult internetis ilmuva võrguväljaandena [1]. Infosüsteemis on kättesaadav enamik akte alates 1990. aastatest. Terviktekstid on ametlikud ning varustatud digitaalse templiga, mis tagab akti muutmatuse pärast avaldamist. Iga muudatuse korral koostatakse muudatusi sisaldav ajakohane terviktekst ning see avaldatakse koos muutva aktiga. Tervikteksti iga redaktsioon on seotud eelneva ja sellele järgneva redaktsiooniga. Ühe ja sama õigusakti raames on võimalik kahte redaktsiooni omavahel võrrelda, nii et oleks kuvatud, millised muudatused on õigusaktis tehtud.

Redaktsioonide omavaheline võrdlus on aga ebamugav, aeganõudev ning mahukate õigusaktide korral võib jääda muudatused kahe silma vahele. Aastatega on osade õigusaktide muudatuste arv nii suur, et puhtalt redaktsioone võrreldes, ei ole võimalik kindlaks teha, kuidas on mõni paragrahv redaktsioonide kaupa muutunud. Näiteks võlaõigusseadust ja karistusseadustikku on alates vastu võtmisest muudetud vastavalt 42 ning 92 korda. Samuti ei anna terviktekst kiiret ja selget ülevaadet õigusakti keerukusest (paragrahvide, lõigete, punktide hulgast).

1.2 Eesmärk ja oodatav tulemus

Käesoleva bakalaureusetöö eesmärk on luua veebirakendus, mis võimaldab visualiseerida õigusakti tervikteksti ning terviktekstis tehtud muudatusi. Terviktekst visualiseeritakse kolmemõõtmelise puustruktuurina, mis kujutatakse kuusena. Kuuse oksad moodustuvad õigusakti paragrahvide, lõigetest ning punktidest ja kujutatakse erinevate värvidega. Lähemalt kirjeldatakse puustruktuuri ülesehitust punktis 4.4. Veebirakendus peab võimaldama kasutajal valida erinevate õigusaktide vahel ning kuvama kasutajale kehtiva redaktsiooni puustruktuuri ning õigusaktis tehtud muudatuste puustruktuuri.

1.3 Ülevaade tööst

Teine peatükk annab ülevaate visualiseerimise olemusest, tuntumatest kasutusvaldkondadest ning kuidas on visualiseerimist rakendatud õigusaktide kujutamiseks. Kolmas peatükk annab ülevaate kasutatud tehnoloogiatest ja nende valiku põhjendustest. Neljas peatükk keskendub veebirakenduse valmimise protsessile ning viiendas peatükis valideeritakse valminud veebirakendust, tuues sealjuures esile võimalikke edasiarendusi. Kokkuvõttes analüüsitakse lühidalt tööd.

2 Andmete visualiseerimine

Andmete visualiseerimine on andmete esitamine pildilises või graafilises vormingus. Mida mahukamad ja keerulisemad on andmed, seda olulisemaks muutub nende visuaalne analüüs ja esitamine. Tänapäeval on andmete visualiseerimisest saanud kiiresti arenev valdkond, mis on segu teadusest ning kunstist. Visualiseerimine peab ennekõike andmeid täpselt edastama, ilma neid eksitamata ega moonutamata. Hea visuaalne pilt või graafika ütleb rohkem kui 1000 sõna. [2]

Andmete visualiseerimise eelised:

- Informatsiooni arusaadavus – graafiline esitus võimaldab visualiseerida suurt hulka andmeid muutes need arusaadavaks ja paremini mõistetavaks [3]. Inimaju haarab pilti tekstist oluliselt kiiremini.
- Seoste loomine – Puhtalt numbrite jada või mahukas tabel ei ole kergesti mõistetav. Visualiseerimine aitab kujutada seoseid andmete ning nendega tehtavate operatsioonide ja tulemuste vahel [4].
- Mustrite tuvastamine – visualiseerimine võimaldab tuvastada andmetes esinevaid mustreid ja trende, mis algandmetest on raskesti välja loetavad [3]. Mustreid saab visualiseerida väga mitmetasandiliselt: läbi värvitoonide, tekstifontide ja piltide.
- Interaktiivsus – Visualiseerimine võimaldab esitada andmeid interaktiivsel viisil. Visuaalid kasutavad dünaamilise diagramme, värve ning kujundeid, mida vastavalt kasutaja sisendile muudetakse. See võimaldab kasutajatel andmeid uurida, nendega manipuleerida ja suhelda. [4]

2.1 Kasutusvaldkonnad

Andmete visualiseerimist kasutatakse väga erinevates eluvaldkondades.

Mõned andmete visualiseerimise kasutusvaldkonnad:

- Turundus – reklaamide populaarsus, liiklus veebilehtedel.
- Müük – müügitulemused, populaarsed tooted, müügitrendid.
- Finants – sissetulekud ja väljaminekud, puhaskasum.
- Logistika – saatmis- ja kohaletoimetamisajad, lao maht, saadetiste arv, tarneteekonnad.
- Tervishoid – haiguste levimine, rahvastiku tervisenäitajad.

2.2 Õigusaktide visualiseerimine

Seadused, määrused, normid või üldised juriidilised dokumendid reguleerivad suurt osa inimeste igapäevasest elust. Need määravad piirid äriprotsessidele, inimestevahelisele suhtlusele ning panevad aluse riigi toimimisele. Kuid seadusi ja juriidilisi tekste on peaaegu alati raske mõista. Teistest valdkondadest, nagu finants, logistika või müük, on juba teada, et visualiseerimine aitab keerukaid aspekte paremini mõista. [5]

Visualiseerimist õigusvaldkonnas aga kohtab harva, vaatamata visualiseerimistehnikate suurele potentsiaalile. Peamised visualiseerimise meetodid keskenduvad juriidiliste teadmiste õpetamiseks ning on mõeldud juristidele, advokaatidele ning juriidilistele nõustajatele mitte lõppkasutajale, kes vajab juriidilist tuge. Info- ja kommunikatsioonitehnoloogiaid (IKT) kasutatakse juriidilises valdkonnas veel vähe, kuid sellegipoolest on mõned näited, mis näitavad IKT tehnoloogiate potentsiaali antud valdkonnas. [5]

Järgnevates alapeatükkides tuuakse võimalikud lahendused, kuidas on visualiseeritud õigusakti. Töö pikkuse huvides keskendutakse lahendustele, mis visualiseerivad ühte õigusakti mitte erinevate õigusaktide vahelisi seoseid.

2.2.1 Puukaart

Puukaart on visuaalne lähenemisviis kirjeldamaks hierahilisi struktuure. Tavaliselt on andmed kuvatud riskülikute kogumine, kus igat rühma kirjeldava risküliku pindala on proportsionaalne sellega kui suure osa moodustab see rühm kogu andmete hulgast. Kasutades värvilahendusi ning interaktiivsust on võimalik visualiseerida mitmeid erinevaid tasemeid. [6]

Juriidilistel dokumentidel võib olla märkimisväärne arv lehti ning erinevaid hierarhitasandeid (peatükid, paragrahvid, lõiked). Sellise sisu analüüsimine on suur väljakutse. Kasutades puukaart diagrammi saab mahukaid õigusakte visualiseerida ühel lehel. Selline lähenemisviis võimaldab alustada õigusakti tervikvaatest ning liikuda erinevate hierarhiatasemeid pidi sügavamale. [7]

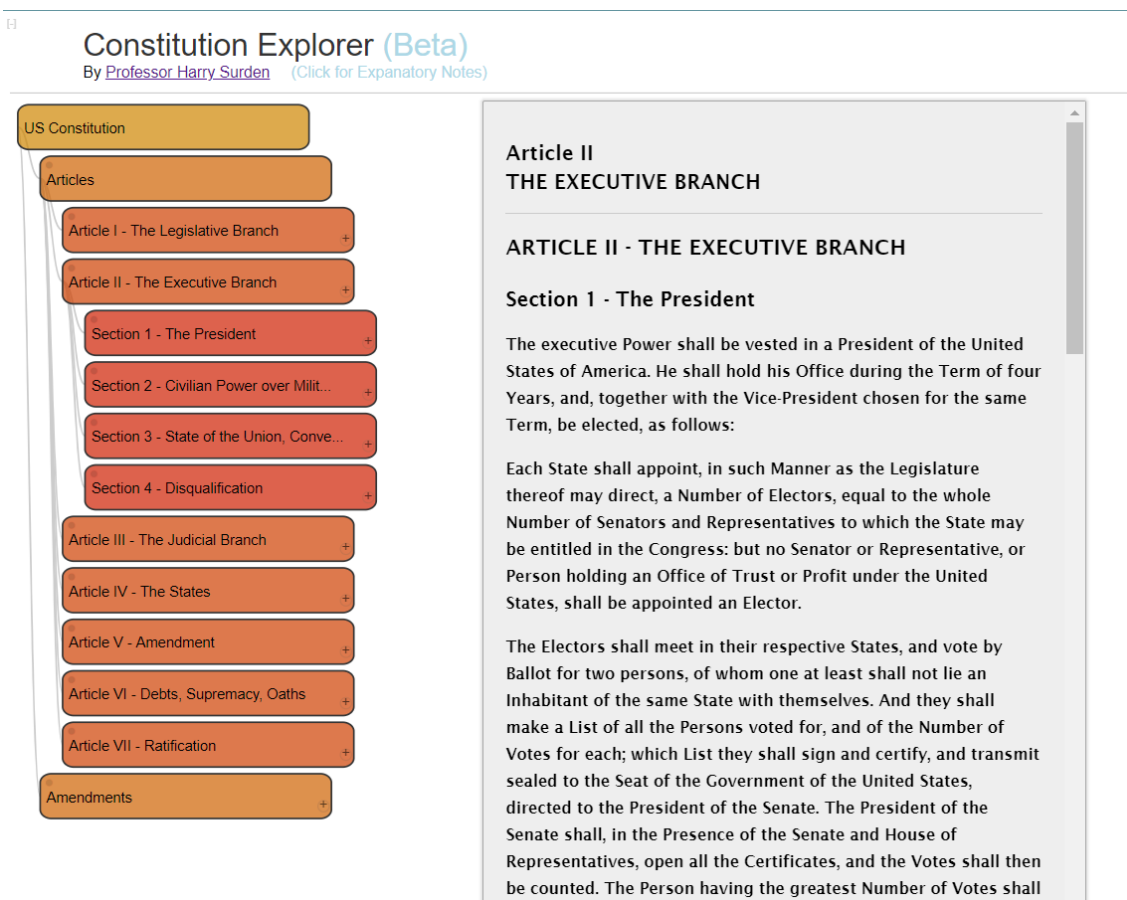
Norma-Simplex on projekt, mis tegeleb määruste ja seaduste tekstide parendamise ja hindamisega. Projekti peamised ülesanded on seotud dokumentide modelleerimisega. *NSM* on tööriistade ja rakenduste komplekt, mida Normal-Simplex projekti raames arendatakse. Antud tööriistade alla kuulub NSM Viz, millega saab vaadata projekti raames valminud visualisatsioone. Visualisatsioonides on kasutatud Portugali Vabariigi juriidilisi tekste (Joonis 1). [8]



Joonis 1. Portugali Vabariigi põhiseadus puukaardina [8].

2.2.2 Puustruktuur

Puu on hierarhiline andmestruktuur, mis salvestab teabe hierarhilises stiilis. Puu koosneb tippudest, mis on omavahel ühendatud servadega [9]. Kasutades õigusakti XML (*Extensible Markup Language*) vormingus, saab õigusakti visualiseerida puustruktuurina. XML-dokumendil on puustruktuur, kus juurelement asub ülaosas ja alamelemendid on ühendatud juurelementidega [10]. Kolorado ülikooli õigusteaduskonna dotsent, Harry Surden arendas rakenduse, mis visualiseerib Ameerika Ühendriikide põhiseadust puustruktuurina (Joonis 2) [11].



Joonis 2. Ameerika Ühendriikide põhiseadus puustruktuurina [11].

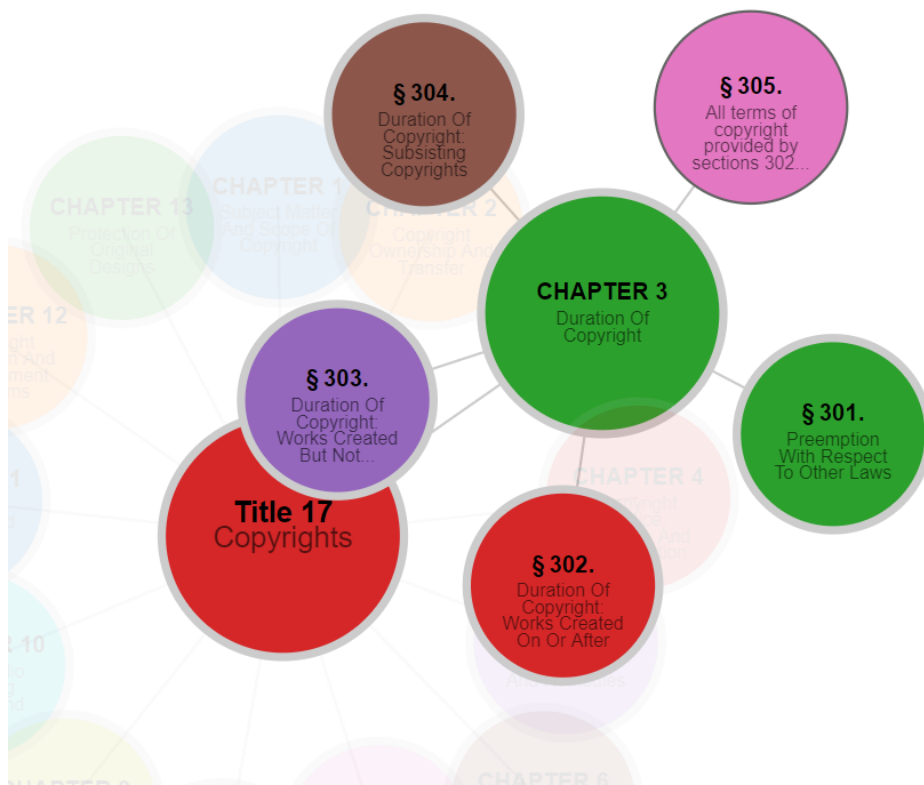
2.2.3 Jõuga suunatud graaf

Jõuga suunatud graafid võimaldavad visualiseerida graafi esteetiliselt meeldival viisil. Nende eesmärk on paigutada graafi sõlmed nii, et kõik servad oleksid enam vähem võrdse pikkusega ja seal on võimalikult vähe omavahel ristuvaid servi. Füüsiliste objektide näiteks gravitatsioonile reageerivate molekulide või interaktsioonide modelleerimiseks kasutatakse sageli jõuga suunatud graafe. [12] Kolorado ülikooli

õigusteaduskonna dotsent, Harry Surden arendas rakenduse, mis visualiseerib Ühendriigi autoriõiguste ja patendi koodeksit. Visualiseerimiseks kasutatakse jõuga suunatud graafi ning koodeksi hierarhi kuvatakse kasutades *hub and spoke* paigutust (vt Joonis 3) [11].

US Code Explorer 2: Force Directed Graph (Beta)

By [Professor Harry Surden](#) ([Click for Explanatory Notes](#))



Joonis 3. Ameerika Ühendriikide autoriõiguste koodeks jõuga suunatud graafina [11].

2.2.4 Sõnapilv

Sõnapilv on eri suuruses sõnade kogum või klaster. Sõnapilv pakub visuaalset pilti tekstis kasutatud sõnavarast. Kõige populaarsem lähenemisviis on tähesuuruste muutmise, kus kõige enam esinenud sõnad visualiseeritakse suuremalt kui vähem esindatud sõnad. See annab kiire ülevaate, dokumendis käsitlevatest teemadest ning millised sõna on kõige silmapaistvamad. [7]

3 Kasutatud tehnoloogiad

Käesolevas peatükis kirjeldatakse veebirakenduse arendamisel kasutatud tehnoloogiad ning nende valiku põhjused.

3.1 WebGL

WebGL on veebigraafika raamistik interaktiivse 2D ja 3D graafika renderdamiseks. Graafika renderdakse HTML-i elemendi *canvas* sisse. WebGL jookseb GPU (*Graphics Processing Unit*) peal, millele tuleb ette anda kaks funktsiooni: tipu vari ehk *vertex shader* ja fragmenti vari ehk *fragment shader*. Esimese funktsiooni ülesandeks on arvutada tipu positsioon ning teise ülesandeks on arvutada iga piksli värvus. Mõlemad funktsioonid on kirjutatud C/C++ laadses programmeerimiskeeles GLSL (*OpenGL Shading Language*), mis on mõeldud erinevate graafika rasterdamise objektide arvutamiseks. [13]

Autor valis selle tehnoloogia oma praktilise osa realiseerimiseks, sest WebGL võimaldab renderdada 3D graafikat ilma pistikprogramme kasutamata. Kuna puhas WebGL on suure õppimiskõveraga, siis soovis autor kasutada mõnda raamistikku, mis kasutab WebGL-i. Samuti toetavad WebGL-i kõik kaasaegsemad veebibrauserid: Firefox 4+, Google Chrome 9+, Opera 12+, Safari 5.1+, Internet Explorer 11+ ja Microsoft Edge build 10240+. WebGL-i toetavad ka erinevad mobiiliplatvormid.

3.2 Three.js

Three.js on Javascripti raamistik, mis pakub reaajas 3D graafikat ning on mõeldud kasutamiseks Javascripti keskkonnas. Three.js kasutab WebGL-i 3D objektide joonistamiseks, muutes protsessi lihtsamaks ja kiiremaks. Three.js töötab kõikides kaasaegsetes veebibrauserites, välja arvatud Internet Exploreri vanemad versioonid. Internet Exploreri vanemate versioonide jaoks on olemas pistikprogramm IEWebGL, mis lisab veebibrauserisse WebGL-i toe. [14]

3.2.1 Three.js valiku põhjendus

Autor katsetas kolmemõõtmeliste puustruktuuride loomiseks kahte erinevat Javascripti raamistikku - Three.js ja p5.js. P5.js on kirjutatud javascriptis, kuid põhineb programmeerimiskeelel Processing. Processing on visuaalse kunsti kodeerimise tööriist. P5.js võimaldab luua kiiresti ja lihtsalt erinevaid graafilisi prototüüpe. [15]

Lõpliku tehnoloogia valiku tegemiseks võrdles autor mõlema raamistiku dokumentatsiooni, näidiskendusi, videoõpetusi ning kui aktiivne on see *stackoverflow*-s ning *Google Trend*-ides. Õppimiskõver kirjeldab autori subjektiivset arvamust kui raske on ühte või teist raamistikku kasutada.

Võrdluse tulemusena valmis tabel (vt Tabel 1), mis kirjeldab tehnoloogiate võrdlemise tulemusi. Võrdlusparameetrite hindamiseks kasutas autor 2-palli süsteemi, kus kaks palli sai raamistik, mille parameeter ületas teise oma. Õppimiskõvera eest sai 2 palli raamistik, mille õppimiskõver oli kõige väiksem ehk seda oleks olnud autori arvates kergem praktilise osa loomiseks kasutada.

Tabel 1. Three.js ja P5.js raamistiku võrdlus

Parameeter	Three.js	P5.js
Dokumentatsioon	2	1
Näidiskendused	2	1
Videoõpetused	1	2
Google Trends	2	1
Stackoverflow	2	1
Õppimiskõver	1	2
Punktisumma	10	8

Autori arvates oli three.js-i dokumentatsioon põhjalikum, sest praktiliseks tööks vajalikud osad olid väga detailselt kirjeldatud. Näidiskendusi on Three.js-is üle 250, mis on kõik leitavad nende Github-i repositooriumis. Videoõpetused olid, aga autori arvates paremad p5.js-il, sest õpetused olid pikad ning nende järgi sai harjutada koodi kirjutamist. Google Trendide aktuaalsuse tabel näitab, et three.js on viimase viie aasta jooksul olnud populaasem valik võrreldes p5.js-iga. Ka *Stackoverflow*-s on three.js-i mainitud rohkemates küsimustes kui p5.js, vastavalt 16029 vs 1531 küsimust. Autorile

tundus Three.js-i õppimiskõver suurem, sest ülesande lahendamise võimalusi kui ka näidisrakendusi on rohkem.

3.3 JavaScript

Javascript on veebi programmeerimiskeel, mis on kasutusel enamus veebilehtedel. Tegemist on dünaamilise ning interpreteeritava kõrgkeelega, mis sobib objektorienteeritud ja funktsionaalsete programmeerimistiilidega. Keel võimaldab animeerida ja dünaamiliselt uuendada sisu ning kontrollida multimeediat. Kõige populaarsem on kliendipoolne JavaScript, kuid seda kasutatakse ka serveri pool. [16]

Esikomponendi loomiseks kasutab autor puhas javascripti. Arvestades three.js-i suurt õppimiskõverat ei soovinud autor lisada juurde keerukamaid esikomponendi raamistikke.

3.4 HTML

HTML on hüpertexti märgistuskeel veebilehekülje sisu struktureerimiseks. HTML jaotub elementideks, mis kirjeldavad, kuidas tuleks sisu veebilehel kuvada. HTML-i kasutatakse erinevates veebirakendustes nii mobiilis kui arvutis. Uusim versioon on HTML5. [17]

4 Veebirakenduse arendusprotsess

Käesoleva bakalaureusetöö eesmärk on luua veebirakendus, mis võimaldab visualiseerida õigusakti tervikteksti ning terviktekstis tehtud muudatusi. Rakenduses tuli rõhku panna ka visuaalsele poolele ja kasutajamugavusele. Veebirakenduse arenduse saab jaotada konkreetseteks etappideks, mis on vaja läbida.

4.1 Rakenduse analüüs

Rakenduse arendus algas esmalt analüüsi protsessiga, kus uuriti põhjalikult juba kasutuselolevaid õigusaktide visualiseerimise lahendusi. Kaardistati ära nende plussid ja miinused ning pandi kirja nõuded rakendusele.

4.1.1 Rakenduse nõuded

Funktsionaalsed nõuded, mis peavad olema loodaval veebirakendusel.

1. Kasutaja saab valida õigusakti, millest soovib genereerida puustruktuuri.
2. Kasutajale kuvatakse õigusakti puustruktuur kui kuusk.
3. Kasutaja saab vaadata õigusakti metaandmeid (jõustumise kuupäev, vastuvõtmise kuupäev jne).
4. Kasutaja saab kaameraga sisse ning välja suumida.
5. Kasutaja saab kaamerat pöörata ja liigutada.
6. Kasutaja saab lähtestada kaamera positsiooni.
7. Kasutaja saab infot puustruktuuri värvide kohta.
8. Kasutaja saab vaadata õigusakti muudatuste puustruktuuri.
9. Kasutaja saab õigusakti muudatuste puustruktuuril näha kõiki kehtinud redaktsioone.
10. Kasutaja saab vaadata õigusakti muudatuste puustruktuuril ajaskaalat.
11. Kasutaja saab vaadata õigusakti muudatuste ajaskaalal aastaarve esimeset redaktsioonist kuni tänapäevani.

Lisaks funktsionaalsetele nõuetele peavad kehtima ka järgmised mittefunktsionaalsed nõuded:

1. Veebirakenduse kasutajaliides on lihtne ja mugav.
2. Veebirakendus peab olema hea disainiga.
3. Rakendus on eesti keeles.

4.2 Andmed

Järgnevas peatükis kirjeldatakse lähemalt, kust pärinevad veebirakenduses kasutusel olevad andmed ja milliseid andmeid kasutatakse veebirakenduse poolt.

4.2.1 Riigi Teataja võrguväljaanne

Riigi Teataja on Eesti Vabariigi ametlik võrguväljaanne, kus avaldatakse õigusaktid ja muud dokumendid, mis on loetletud Riigi Teataja seaduses §-s 2. Võrguväljaandes avaldatakse vastuvõetud kujul algtekste ja terviktekste. Terviktekst kajastab kõiki aktis tehtud muudatusi. Alates 1. juunist 2002, mil elektrooniline Riigi Teataja kasutusele võeti, on Riigi Teatajas olemas kõikide seaduste ja üleriigiliste määruste kõik terviktekstide ajas kehtinud redaktsioonid. [1]

Antud bakalaureusetöös kasutatakse andmetena Riigi Teataja võrguväljaandest pärinevaid seaduste terviktekste, mis on rakenduses kasutusel XML vormingus. Kehtivat tervikteksti redaktsiooni kasutatakse tervikteksti visualiseerimiseks puustruktuurina ning vanemaid redaktsioone võrreldakse omavahel, et leida täpsed muudatused terviktekstides. Leitud muudatused visualiseeritakse samuti eraldi puustruktuurina.

4.2.2 Andmete lugemine

Andmete visualiseerimiseks tuli kõigepealt XML-ist välja lugeda rakendusele vajalikud andmed. Õigusakti XML-ist loetakse välja metaandmed, paragrahvid, lõiked ning punktid.

Veebirakenduses kasutatavad õigusakti metaandmed on:

- Akti väljaandja
- Akti liik
- Teksti liik
- Vastuvõtmise kuupäev
- Jõustumise kuupäev
- Redaktsiooni jõustumise kuupäev
- Redaktsiooni kehtivuse lõppkuupäev

Õigusakti paragrahvid salvestatakse listina, kus iga listi element on paragrahvi objekt. Igal paragrahvi objektil on paragrahvi number, kehtivuse tõeväärtus, paragrahvi pealkiri, lõiked ning punktid. Mõnedel paragrahvidel on ka sisutekst. Lõiked ning punktid on sarnaselt paragrahvidele salvestatud objektina, kus parameetriteks on number, kehtivuse tõeväärtus ja sisutekst. Kui lõikel on punktid siis need on selle all välja toodud.

4.3 Graafika kuvamine

Veebirakenduses 3D graafika kuvamiseks on vaja kolme objekti: stseeni, kaamerat ja renderdajat. Three.js renderdaja võtab sisendiks kaamera ja stseeni ning renderdab 3D graafika HTML elemendi *canvas* sisse. Stseenile lisatakse puustruktuuri objekt. Stseene on praktilises osas kaks: esimene sisaldab kehtiva tervikteksti puustruktuuri ning teine terviktekstis tehtud muudatuste puustruktuuri.

4.4 Õigusakti puustruktuuri ehitus

Käesolevas peatükis kirjeldatakse õigusakti puustruktuuri ehitust. Ehituse kirjeldamiseks kasutatakse kehtivat Töölepingu seaduse tervikteksti [18]. Erinevate osade eristamiseks kasutatakse praktilises osas värve. Paragrahvid on pruuni värvi, lõiked rohelised ning punktid kollased.

4.4.1 Paragrahv ilma lõigete ja punktideta

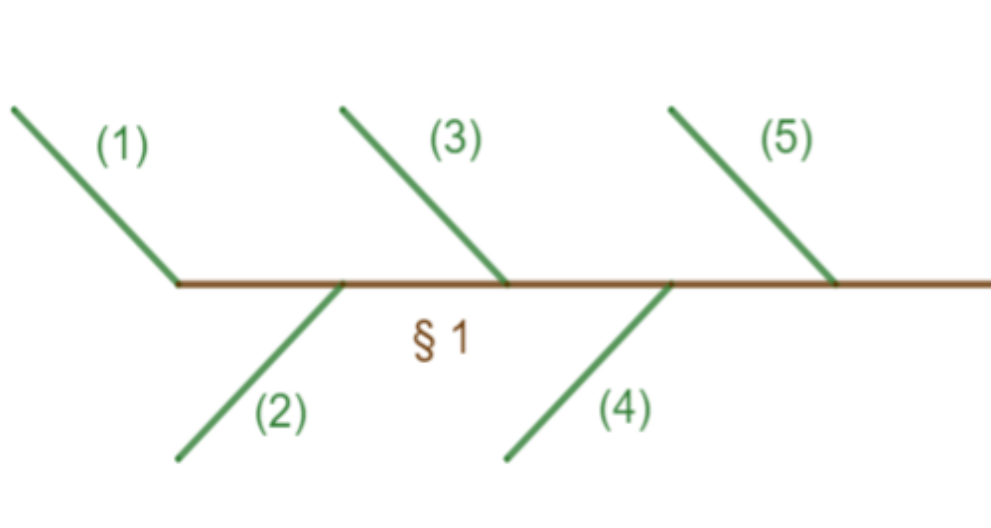
Paragrahvid, millel puuduvad lõiked ja punktid visualiseeritakse ühe sirglõiguna kahe punkti vahel (Joonis 4). Kehtiv paragrahv on pruuni värvi. Kehtetu või tekstist välja jäetud paragrahv on punast värvi.



Joonis 4. Töölepingu seadus § 3

4.4.2 Paragrahv ainult lõigetega

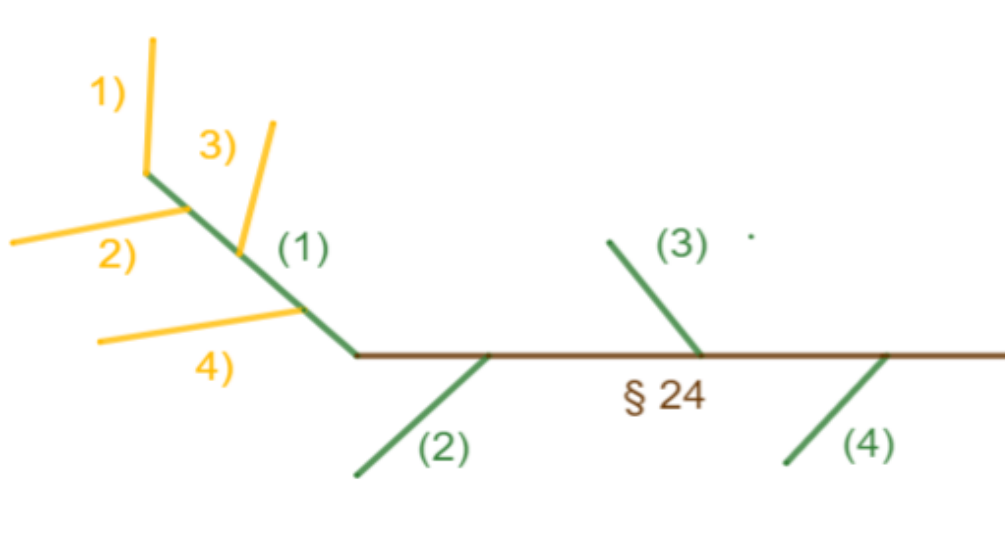
Paragrahvi sirglõik jaotatakse osadeks. Igast osast väljub õigusakti lõike sirglõik. Lõige number 1 paikneb alati sirglõigu otsas (Joonis 5).



Joonis 5. Töölepingu seadus § 1 lõiked 1 - 5

4.4.3 Paragrahv lõigete ja punktidega

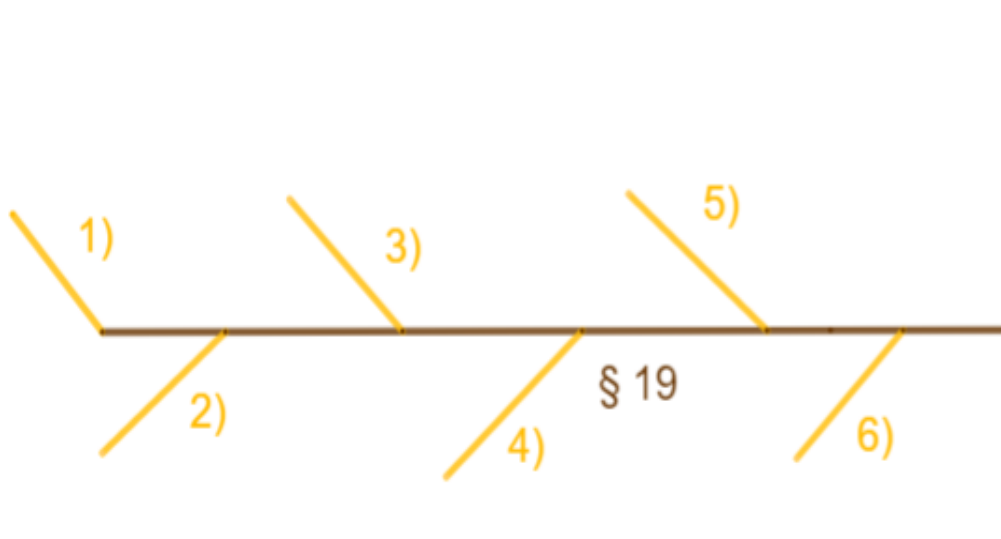
Paragrahvi sirglõik jaotatakse osadeks. Igast osast väljub õigusakti lõike sirglõik. Kui õigusakti lõikel on punktid, siis jaotatakse lõike sirglõik osadeks. Igast lõike sirglõigust väljub punkti sirglõik (Joonis 6).



Joonis 6. Töölepingu seadus § 24 lõige 1 punktid 1- 4 ja lõiked 2, 3 ja 4

4.4.4 Paragrahv ainult punktidega

Paragrahvi sirglõik jaotatakse osadeks. Igast osast väljub õigusakti punkti sirglõik, tähistatud kollaselt (Joonis 7).



Joonis 7. Töölepingu seadus § 19 punktid 1-6

4.4.5 Paragrahvid ülaindeksitega

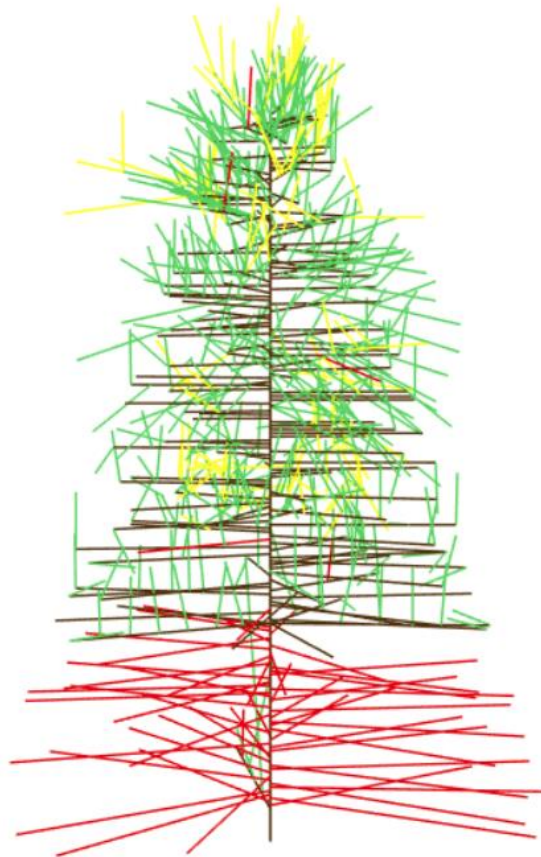
Ülaindeksitega määratud paragrahvid väljuvad ühest tüvepunktist kindlal kõrgusel (Joonis 8).



Joonis 8. Töölepingu seadus § 137 ja § 137¹

4.4.6 Terviklik õigusakt veebirakenduses kujutatud puustruktuurina

Töölepingu seadus oma paragrahvide, lõigete ja punktidega kujutatud puustruktuurina (Joonis 9).



Joonis 9. Töölepingu seadus

4.5 Puustruktuuri loomine

4.5.1 Tüve loomine

Puustruktuuri joonestamist alustatakse tüvest. Tüve pikkuseks on määratud 2100 ühikut, mööda Z - telge ülesse. Tüvi koosneb punktidest, millel on xyz koordinaadid. Tüve punktidel muutub vaid z koordinaat ning x ja y koordinaat on 0. Punkti loomiseks kasutatakse three.js klassi Vector3. Vector3 kirjeldab numbrite kolmikut, mida saab kasutada järgnevalt: punkti kirjeldamiseks 3D ruumis, pikkuse ja suuna kirjeldamiseks 3D ruumis või suvalise kolme numbri kirjeldamiseks [19]. Tüve punktide arv sõltub õigusakti paragrahvide numeratsioonist. Punktide vahemaa leidmiseks jagatakse tüve pikkus paragrahvide arvuga. Paragrahvide arv on võetud arvestamata ülaindeksiga paragrahve. Näiteks § 1¹ ei loeta eraldi ühikuks tüvepunktide loomisel.

Lõpliku tüve moodustamiseks ühendatakse tüvepunktid omavahel, kasutades three.js klassi Line. Line võtab sisendiks värvi, alguspunkti ja lõpp-punkti ning joonestab nende vahele pidevjoone. Algus ja lõpp-punkt peavad olema Vector3 objektid. Tüvi on pruuni värvi.

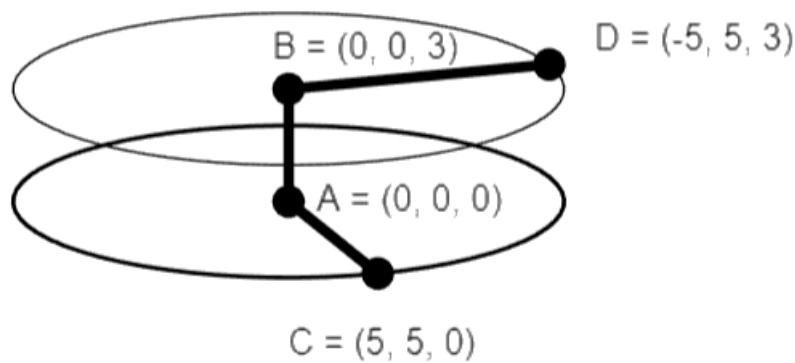
Tüve punktid salvestatakse listina ning kasutatakse edaspidi paragrahvide joonestamiseks.

4.5.2 Õigusakti paragrahvide joonestamine

Paragrahvide joonestamise eelduseks on tüvepunktide olemasolu. Igast tüvepunktist joonestatakse üks või enam paragrahvi. Kehtivad paragrahvid on pruuni värvi. Paragrahvide joonestamist alustatakse kõige suurema numbriga paragrahvist. Kehtetud või terviktekstist välja jäetud paragrahvid on punast värvi. Paragrahv on sirglõik, mille alguspunkt on tüvepunkt.

Paragrahve joonestatakse mööda z-telge alt ülesse ning § 1 on alati puu kõige ülemine sirglõik. Kõige esimesena joonestatava paragrahvi x ja y koordinaadid antakse programmis ette, mis määrab kõige alumise sirglõigu pikkuse. Z koordinaat on võrdne tüvepunkti z koordinaadiga. Iga järgneva paragrahvi sirglõik, mis tüvepunktist joonestatakse on proportsionaalselt lühem kui eelmine.

Paragrahvide sirglõikude lõpp-punktid paiknevad ümber tüve (vt Joonis 10). Punktid A ja B on tüvepunktid ning C ja D lõpp-punktid. Punkti D on punkti C suhtes pööratud 90 kraadi võrra päripäeva. Punktide pööramiseks kasutatakse three.js klassi Quaternion. Quaternionil on meetod setFromAxisAngle(), mis võtab parameetritena telje ühikvektori ja nurga radiaanides ning vastavalt antud väärtustele muudab lõpp-punkti xyz koordinaate. Pööratava nurga leidmiseks kasutatakse meetodit random(), mis genereerib suvalise arvu vahemikus [0, 360]. Nurk kraadides teisendatakse radiaanidesse.



Joonis 10. Sirglõigu lõpp-punktide pööramine

Paragrahvide joonestamisel võib esineda olukord, kus ühest tüvepunktist tuleb joonestada mitu paragrahvi sirglõiku, näiteks olukorras, kus terviktekstis on § 3 ja § 3¹. Sellisel juhul kasutatakse valemit:

$$pööratav_nurk = 360^\circ \div (paragrahvi_indeksite_arv + 1)$$

Kahe paragrahvi joonestamisel ühest tüvepunktist oleks esimese paragrahvi nurk 0° ning teise nurk 180°.

4.5.3 Sihivektori leidmine

Sihivektoriks nimetatakse sirge suvalise kahe erineva punkti poolt määratud vektorit [20]. Olgu sirge s alguspunkt $A(x_1, y_1, z_1)$ ja lõpp-punkt $B(x_2, y_2, z_2)$, siis sirge sihivektor on:

$$\vec{s} = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$$

Sihivektori pikkus võimaldab leida sirgel asetsevaid punkte.

4.5.4 Õigusakti lõikude ja punktide joonestamine

Õigusakti lõikude või punktide joonestamiseks tuleb leida sirglõigule punktid. Punktide leidmiseks kasutatakse sihivektorit ja tema pikkust. Sihivektori x , y ja z koordinaadid jagatakse läbi soovitud punktide arvuga. Saadud uued x , y ja z koordinaadid võimaldavad välja arvutada punktide koordinaadid. Programmikood sirglõigule punktide leidmiseks:

```
function calculateLinePoints(startVector, endVector, pointsNeeded) {
  let directionVector = calculateDirectionVector(startVector,
endVector);

  let lineStepX = directionVector.x / pointsNeeded;
  let lineStepY = directionVector.y / pointsNeeded;
  let lineStepZ = directionVector.z / pointsNeeded;

  let linePoints = [];

  for (let i = 1; i <= pointsNeeded; i += 1) {
    linePoints.push(new THREE.Vector3(
      startVector.x + lineStepX * i,
      startVector.y + lineStepY * i,
      startVector.z + lineStepZ*i))
  }

  return linePoints;
}
```

Joonis 11. Programmikood sirglõigule punktide leidmiseks.

Õigusakti lõik joonestatakse sirglõiguna, mille alguspunktiks on paragrahvi sirglõigu üks punkt. Lõikude lõpp-punktid paiknevad ümber paragrahvi sirglõigu. Lõpp-punkti pööramiseks kasutatakse `three.js` klassi `Quaternioni`, millele antakse paragrahvi sirglõigu normaalvektor ning nurk radiaanides. Kehtiv lõik kujutatakse rohelise värviga. Kehtetu lõik või terviktekstist välja jäetud lõik kujutatakse punasega.

Õigusakti punkt joonestatakse sirglõiguna, mille alguspunktiks on paragrahvi sirglõigu üks punkt või lõike sirglõigu üks punkt. Õigusaktide punktide lõpp-punktid paiknevad

üumber paragrahvi sirglõigu (vt joonis 7) või üumber lõike sirglõigu (vt joonis 6). Kehtiv punkt kujutatakse kollase värviga. Kehtetu või terviktekstist välja jäetud punkt kujutatakse punasega.

4.6 Õigusakti redaktsioonides tehtud muudatused

Redaktsioonide võrdlusel leitakse õigusaktist tehtud muudatused. Muudatused kuvatakse samuti puustruktuurina. Käesolevas peatükis kirjeldatakse muudatuste puustruktuuri loomist.

4.6.1 Muudatuste leidmine

Õigusaktist tehtud muudatused leitakse kahe redaktsiooni võrdluse teel. Korraga võrreldakse eelmist ning järgmist redaktsiooni ning tulemusena saadakse, millised õigusakti paragrahvid, lõiked või punktid on muutunud. Õigusaktist otsitavad muudatused on:

- Paragrahvi pealkirja muutmine
- Paragrahvi, lõike või punkti sisuteksti muutmine
- Paragrahvi, lõike või punkti kehtivuse muutmine
- Paragrahvi, lõike või punktide juurde lisamine

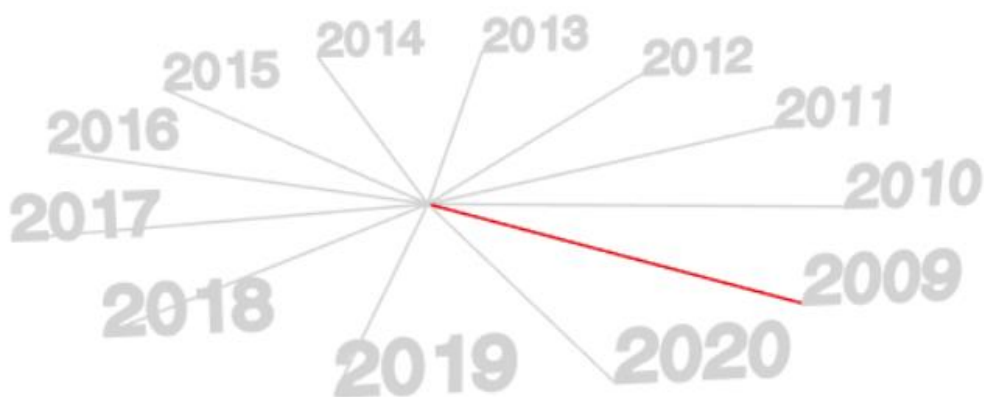
Leitud muudatused kuvatakse ajaskaalal.

4.6.2 Ajaskaala loomine

Ajaskaala on 360 kraadine ring, mis jagatakse aastate kaupa sektoriteks (vt Joonis 12). Kõige esimene sektor kujutab aastat, millal õigusakt jõustus ning iga järgnev sektor on aasta edasi. Kõige viimane sektor kujutab hetkel olevat aastat. Igal redaktsioonil on oma jõustumise kuupäev ning kindel koht ajaskaalal. Vanema ja uuema redaktsioonide võrdlemisel leitud muudatused esitatakse uuema jõustumise kuupäeva järgi ajaskaalal.

Ajaskaala joonestamiseks leitakse aasta nurk, milleks kasutatakse valemit:

$$aasta_nurk = 360^{\circ} \div (praegune_aasta - esimese_redaktsiooni_aasta + 1)$$



Joonis 12. Muudatuste ajaskaala

4.6.3 Muudatuste esitamine

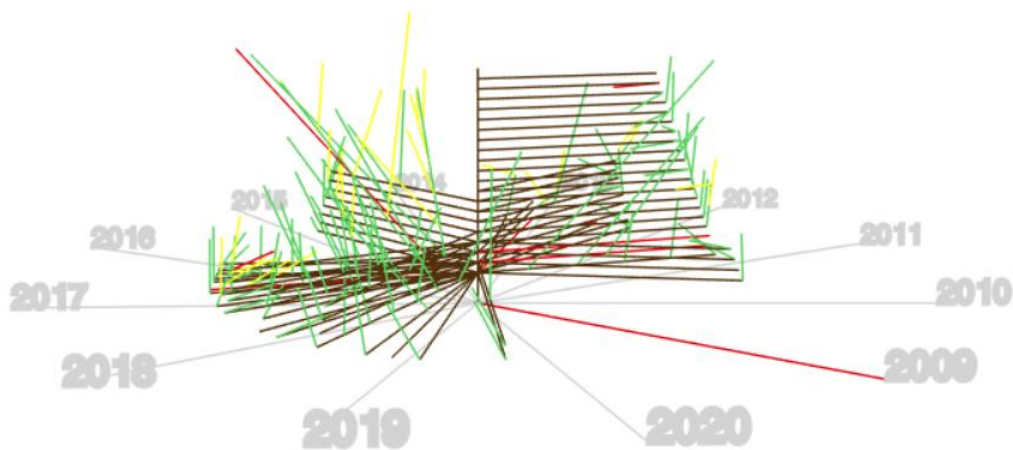
Käesolevas peatükis selgitatakse vaadeldava redaktsiooni sektori nurga leidmist. Sektori nurga leidmiseks kasutatakse eelmise punktis toodud aasta nurga valemit ja kuu nurga leidmise valemit. Kuu nurga leidmise valem on:

$$kuu_nurk = aasta_nurk \div 12$$

Muudatuste joonestamiseks vajaliku nurga leidmiseks võetakse nurga suurus 0° alati esimese redaktsiooni jõustumise aasta jaanuari kuu järgi ning leitakse sektori nurk kasutades järgnevat valemit:

$$\begin{aligned}
 sektori_nurk = & (vaadeldava_redaktsiooni_aasta \\
 & - esimese_redaktsiooni_aasta) * aasta_nurk \\
 & + (vaadeldava_redaktsiooni_kuu - 1) * kuu_nurk
 \end{aligned}$$

Teades sektori nurka, redaktsioonide võrdlusel saadud muudatuste arvu ning nende struktuuri saab esitada muudatusi puustruktuurina (vt joonis 12).



Joonis 13. Töölepingu seaduses tehtud muudatused

4.7 Veebirakenduse vaated

Veebirakendus koosneb kolmest vaatest: esilehest, õigusakti kehtiva redaktsiooni vaatest ja õigusaktis tehtud muudatuste vaatest.

Esilehel kuvatakse kasutajale sissejuhatav tekst ning valiku vorm. Valiku vorm sisaldab viite erinevat õigusakti, millest saab genereerida õigusakti puustruktuure. Peale valiku tegemist suunatakse kasutaja kehtiva redaktsiooni vaatesse.

Kehtiva redaktsiooni vaade koosneb metaandemetest, infoväljadest ning kehtiva redaktsiooni puustruktuurist. Metaandmetes kuvatakse 4.2.2 kirjeldatud andmed ning õigusakti pealkiri. Infoväljadel kirjeldatakse lühidalt puustruktuuri ehitust ning kasutatud värvide tähendust. Kehtiva redaktsiooni puustruktuuri saab kasutaja liigutada, pöörata ja sisse/välja suumida. Lehel üleval paremas nurgas on nupp „Ajalugu“, mis viib kasutaja edasi õigusaktis tehtud muudatuste vaatesse.

Õigusaktis tehtud muudatuste vaade sisaldab puustruktuuri ning infovälja. Infoväli kirjeldab lühidalt, kuidas on leitud muudatused õigusaktis, mida kujutab ajaskaala ning kuidas kujutatakse muudatused puustruktuuris. Puustruktuuri saab kasutaja liigutada, pöörata ning sisse/välja suumida.

5 Valideerimine ja võimalikud edasiarendused

Käesolevas peatükis antakse ülevaade valideerimise tulemustest ja veebirakenduse võimalikest edasiarendustest.

5.1 Veebirakenduse valideerimine

Käesoleva bakalaureusetöö praktilises osas valminud veebirakendust anti testida kahel isikul, kelle tööülesanded nõuavad õigusaktide tundmist ning nende käsitlemist. Üks testija oli ametilt tegutsev jurist ning teine oli õigusteenuse juht. Tagasisidena toodi, et õigusaktide visualiseerimise idee on väga huvitav ning nähakse potentsiaali sellise rakenduse edasi arendamiseks.

Mõlemad testijad tõid välja, et sellisest visualiseeringust saaksid suurt kasu lõigata õigusloomega tegelevad juristid, kes saaksid vaadata normitehnilise poole pealt, milline valdkond on kui täpselt reguleeritud. Mida rohkem on paragrahve, seda detailsemalt on valdkond reguleeritud. Kuna valdkonna reguleeritavust käsitletakse koolkonniti erinevalt, saaksid siinkohal analüütikud või ka magistrandid tõestada hüpoteesi, kas detailsem seadusandlus on tõhusam kui mitte nii detailne või kas paragrahvide rohkus tähendab seadusandluse paremust.

Rakenduse funktsionaalsuse poolelt toodi välja, et võiks olla näha, millist paragrahvi on redaktsioonide vastuvõtmisest kõige rohkem muudetud. Selline võimalus näitaks seadusloojatele juba koheselt kitsaskohad kätte. Samuti toodi välja, et kuusepuu okstele vajutades võiks olla kuvatud õigusakti sisu.

Visuaalse poole pealt toodi välja, et üldine visualiseering kuusepuuna on meeldiv. Puustruktuurilt on kompaktselt näha, millised õigusaktid on mahukamad ja milliseid õigusakte on kõige rohkem muudetud. Mainita ka seda, et lõbusa vahepalana saaksid huvilised katsetada, milline seadus kõige rohkem kuusepuule sarnaneb.

5.2 Võimalikud edasiarendused

5.2.1 Õigusakti tekstilise sisu kuvamine

Kuvada õigusakti sisu veebirakenduses. Suunates hiirekursari suvalisele sirglõigule või sellele vajutades, kuvatakse valitud sirglõigu tekstiline sisu. Sisuks oleks paragrahvi, lõike või punkti pealkiri, tähis ja sisutekst. Muudatuste puustruktuuri korral kuvada lisaks sisutekst enne muutmist ja pärast muutmist.

5.2.2 Kõik vaated ühel kujul

Ühildada kehtiva redaktsooni puustruktuur ning muudatuste puustruktuur ühte vaatesse. Ühises vaates oleks kuvatud ajaskaala, õigusakti algtekst ning muudetud seaduse osad oleksid markeeritud. Markeeritud osi saaks täpsemalt vaadata teises vaates. Nõuab põhjalikku analüüsi.

5.2.3 Õigusaktide hulga laiendamine

Suurendada õigusaktide hulka luuas rakendusele juurde andmebaas, kus hoitakse kõikide Eesti Vabariigi seaduste terviktekste ning nende kehtivaid ja kehtetuid redaktsioone.

6 Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli luua veebirakendus, mis võimaldab visualiseerida õigusakti tervikteksti ning terviktekstis tehtud muudatusi. Nagu tööst selgus, on õigusaktidest peaaegu alati raske aru saada, kuid visualiseerimine võimaldab keerukaid aspekte paremini mõista.

Töös anti ülevaade visualiseerimise olemusest ning kuidas seda õigusaktidega rakendada. Eesmärgi saavutamiseks analüüsiti esmalt olemasolevaid õigusaktide visualiseerimise lahendusi, lisaks keskenduti erinevatele tehnoloogiatele, mida on kasutatud eesmärgi saavutamiseks. Samuti on autor kirjeldanud arendusprotsessi, milles selgitatakse täpsemalt nii projekti analüüsi, puustruktuuri ehitust ja arendusprotsessi ning millest koosneb rakenduse esikomponent.

Töö tulemusena valmis veebirakenduse prototüüp, mis võimaldab visualiseerida Riigi Teataja võrguväljaandest pärinevaid õigusaktide terviktekste ning terviktekstis tehtud muudatusi kolmemõõtmelise puustruktuurina. Valminud veebirakendus realiseerib analüüsis püstitatud funktsionaalseid ja mittefunktsionaalseid nõudeid.

Valminud veebirakenduse jaoks koostati tagasiside küsitlus isikutega, kelle tööülesanded nõuavad õigusaktide tundmist ja nende käsitlemist. Tagasisidest selgus, et õigusakti visualiseering oleks vajalik õigusloomega tegelevatele juristidele, kes saaksid seda kasutada analüüsi eesmärgil.

Käesoleva bakalaureusetöö edasiarendusena saaks lisada õigusakti tekstilise sisu kuvamise, mis annab hiireliigutusega algse ettekujutuse õigusakti sisust. Seaduse loojatele oleks hea täiendusena paragrahvi muutumiste arv, mis näitaks seaduse kitsaskohti. Lisafunktsioonide võimalustest saab laiemat ülevaadet tagasisides ja peatükis 5.

Lõputöö andis autorile suurel määral uusi programmeerimisoskuseid, kuna graafika programmeerimisega polnud autor enne kokku puutunud.

Kasutatud kirjandus

- [1] Justiitsministeerium, “Riigi Teataja võrguväljaandest,” [Online]. Available: <https://www.riigiteataja.ee/abiLeht.html?id=1>. [Accessed 1 May 2020].
- [2] C. O. Wilke, “Fundamentals of Data Visualization,” O’Reilly Media, Inc., April 2019.
- [3] M. Matias, “Five advantages of data visualization,” 9 October 2018. [Online]. Available: <https://www.netquest.com/blog/en/five-advantages-of-data-visualization>. [Accessed 30 April 2020].
- [4] M. Chatterjee, “Introduction to Data Visualisation- Why is it Important?,” 24 September 2019. [Online]. Available: <https://www.mygreatlearning.com/blog/introduction-to-data-visualisation-why-is-it-important/>. [Accessed 30 April 2020].
- [5] B. Dirk and N. Kawa, “Visualizing Law - A Norm-Graph Visualization Approach based on Semantic Legal Data,” in *The 4th International Conference of the Virtual and Augmented Reality in Education*, Budapest, 2018.
- [6] Y. Holtz, “Treemap,” [Online]. Available: <https://www.data-to-viz.com/graph/treemap.html>. [Accessed 2 May 2020].
- [7] C. Nuno and B. Luís, “Transforming Legal Documents for Visualization and Analysis,” in *11th International Conference on Theory and Practice of Electronic Governance (ICEGOV2018)*, New York, 2018.
- [8] L.S.Barbosa, J.N.Oliveira, J.J.Almeida, Á.I.Sanromán and N.R.Carvalho., “Norma Simplex Models,” [Online]. Available: <http://nsm.nrc.pt/index.html>. [Accessed 2 May 2020].
- [9] “Trees in Data Structure,” [Online]. Available: <https://www.tutorialride.com/data-structures/trees-in-data-structure.htm>. [Accessed 2 May 2020].
- [10] T. Myer, “A Really, Really, Really Good Introduction to XML,” 24 August 2005. [Online]. Available: <https://www.sitepoint.com/really-good-introduction-xml/>. [Accessed 2 May 2020].
- [11] H. Surden, “Data Visualization Projects,” [Online]. Available: <http://www.harrysurden.com/wordpress/projects>. [Accessed 2 May 2020].
- [12] Taihua Xu; Jie Yang; Guanglei Gou, „A Force-Directed Algorithm for Drawing Directed Graphs Symmetrically,“ *Mathematical Problems in Engineering*, kd. 2018, nr 1, pp. 1-5, 19 Nov 2018.
- [13] “WebGL Fundamentals,” [Online]. Available: <https://webglfundamentals.org/webgl/lessons/webgl-fundamentals.html>. [Accessed 29 April 2020].
- [14] “Three.js Fundamentals,” [Online]. Available: <https://threejsfundamentals.org/threejs/lessons/threejs-fundamentals.html>. [Accessed 30 April 2020].
- [15] A. Parrish, “First steps with p5.js,” 2019. [Online]. Available: <https://creative-coding.decontextualize.com/first-steps/>. [Accessed 5 May 2020].
- [16] D. Flanagan, “JavaScript: The Definitive Guide, 7th Edition,” O’Reilly Media, Inc, June 2020.

- [17] D. G., "What is HTML? The Basics of Hypertext Markup Language Explained," 2019. [Online]. Available: <https://www.hostinger.com/tutorials/what-is-html>. [Accessed 29 April 2020].
- [18] "Töölepingu seadus (lühend - TLS)," [Online]. Available: <https://www.riigiteataja.ee/akt/122122012030?leiaKehtiv>. [Accessed 10 May 2020].
- [19] "Vector3," [Online]. Available: <https://threejs.org/docs/#api/en/math/Vector3>. [Accessed 10 May 2020].
- [20] M. Abel, "Kursuses "Algebra ja geomeetria" kasutatud slaidid," [Online]. Available: <http://math.ut.ee/pmi/kursused/ag/SirgeVorrandid.pdf>. [Accessed 10 May 2020].