

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Blaž Drole

**Programski modul za 3-dimenzionalno (3D) vizualizacijo tkanin**

DIPLOMSKO DELO  
VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE  
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Mira Trebar

Ljubljana, 2012

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.



Št. naloge: 00340/2012

Datum: 04.09.2012

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **BLAŽ DROLE**

Naslov: **PROGRAMSKI MODUL ZA 3-DIMENZIONALNO (3D) VIZUALIZACIJO  
TKANIN**

**PROGRAM MODULE FOR 3-DIMENSIONAL (3D) VISUALIZATION OF  
FABRIC**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Računalniška vizualizacija je ena od pomembnih prednosti pri oblikovanju tkanin v procesu proizvodnje blaga v tekstilni industriji. V ta namen se vse pogosteje uporablja 3-dimenzionalna (3D) predstavitev vzorcev, ki omogoča prikaz realnih oblik niti v tkanini pred začetkom proizvodnje. Kandidat naj v diplomski nalogi preuči potrebna orodja in spletne tehnologije, ki omogočajo prikaz vzorcev tkanin v brskalnikih in delujejo v operacijskem sistemu Linux. Na osnovi zahtev obstoječega programa za načrtovanje ArahWeave naj izdelava modul za 3D prikaz poteka niti v preji. Programska rešitev naj bo preverjena v najpogosteje uporabljenih brskalnikih, kot so Firefox, Opera in Chrome.

Mentor:

  
doc. dr. Mira Trebar



Dekan:

  
prof. dr. Nikolaj Zimic

**IZJAVA O AVTORSTVU**  
**diplomskega dela**

Spodaj podpisani Blaž Drole,  
z vpisno številko 63980027,

sem avtor diplomskega dela z naslovom:

Programski modul za 3-dimenzionalno (3D) vizualizacijo tkanin

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Mire Trebar,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 10. 12. 2012

Podpis avtorja:

Docentu dr. Miri Trebar se iskreno zahvaljujem za mentorstvo, pomoč in strokovne nasvete pri diplomskem delu.

Hvala mag. Dušanu Petercu za pomoč, strokovne nasvete in podporo pri izdelavi programa in tudi diplomskega dela.

Za potrpežljivost in podporo pri študiju se zahvaljujem tudi svojim staršem.

# Kazalo

<b>Povzetek.....</b>	<b>1</b>
<b>Abstract.....</b>	<b>2</b>
<b>1 Uvod.....</b>	<b>3</b>
<b>2 Računalniška predstavitev tkanin.....</b>	<b>5</b>
2.1 Dizajn.....	5
2.2 Simulacija.....	6
<b>3 3D grafika – orodja.....</b>	<b>9</b>
3.1 Strojna in programska oprema.....	9
3.2 Brskalniki.....	14
<b>4 ArahWeave3D program.....</b>	<b>17</b>
4.1 Izris poteka niti v 3D.....	17
4.2 Vhodni podatki.....	23
4.3 Postopek izdelave modela.....	25
4.4 Generacija niti.....	29
4.5 Izris v prostoru.....	39
4.6 Interakcija uporabnika z modelom tkannine.....	41
<b>5 Analiza rezultatov- testiranje.....</b>	<b>43</b>
<b>6 Sklepne ugotovitve.....</b>	<b>47</b>
<b>7 Viri.....</b>	<b>49</b>

# Seznam uporabljenih kratic in simbolov

3D	3-dimensional (3-dimenzionalno)
API	Application Programming Interface (Programski vmesnik)
CAD	Computer Aided Design (Računalniško Podprti Dizajn)
CAM	Computer Aided Manufacturing (Računalniško Podprta Proizvodnja)
DOM	Document Object Model (Objektni model dokumenta)
FPS	Frames Per Second (Okvirov na sekundo)
GLSL	OpenGL Shading Language (Jezik senčnikov v OpenGL)
HTML5	HyperText Markup Language 5 (Protokol za prenos hiperteksta)
WebGL	Web Graphics Library (Knjižnjica za spletno grafiko)
OpenGL ES	OpenGL for Embedded Systems (OpenGL za vgradne sisteme)
SQL	Structured Query Language (Strukturiran jezik poizvedb)
XML	eXtensible Markup Language (Razširljivi označevalni jezik)

# Povzetek

Razvoj novih vzorcev v tekstilni industriji je delo, ki zahteva veliko mero natančnosti in vizualizacije končnega vzorca. Čeprav se za razvoj že vrsto let uporabljajo računalniki, so bili tehnološko omejeni z možnostmi vizualizacije končnega rezultata na dvodimenzionalen prikaz. Z napredkom računalniške tehnologije, še posebej grafičnih kartic, pa je mogoče tudi na osebnih računalnikih prikazati zapletene vezave tkanin v trodimenzionalnem (3D) prostoru. To olajša vizualizacijo končnega rezultata na način, ki je razumljiv tudi običajnim ljudem. Cilj diplomske naloge je preveriti stanje na trgu prikaza 3D grafike za potrebe tekstilne industrije in nato v sodelovanju s podjetjem Arahne d. o. o. razviti modul za 3D vizualizacijo tkanine v operacijskem sistemu Linux. Modul je del programskega paketa za razvoj tkanine ArahWeave, ki uporablja spletni brskalnik za prikaz modela tkanine. Za prikaz 3D grafike v spletnem brskalniku se uporablja standard WebGL. Ob klicu modula iz glavnega programa se ustvarijo parametri izbrane tkanine, ki tvorijo vhodne podatke za naš modul. Za izdelavo modela je uporabljen programski vmesnik Three.js. Glede na vhodne podatke je potrebno izračunati potek vsake niti v prostoru, ji prirediti zahtevane lastnosti debeline in barve, ter jo postaviti na pravilno pozicijo v prostoru glede na začetek tkanine. Tkanino sestavljajo niti osnove in njim prečno postavljene niti votka. Ko je izrisana celotna tkanina, je potrebno zagotoviti interakcijo upravnika s 3D modelom tkanine v prostoru. Omogočiti moramo rotacijo modela, zamik vstran in povečavo-pomanjšavo velikosti prikazanega vzorca. Potrebno je zagotoviti tudi obdelavo prikaza modela, kot naprimer odstranitev niti, spreminjanje prosojnosti in možnost prikaza niti v žičnatem modelu. Upoštevana je primerjava različnih brskalnikov, ki jih lahko uporabimo za zagon modula v operacijskem sistemu Linux.

Ključne besede: 3D, WebGL, tkanina, niti osnove, niti votka

# Abstract

Development of new designs in textile industry is a work, that requires a great deal of precision and the need for end design visualisation. Although computers are being used for many years now, we were technologically limited when visualising the final result to only two dimensional display. With the advances in computer technology, especially graphic cards, we are now able to display complex weave in three dimensional (3D) space even on personal computers. This greatly simplifies the visualisation of end result in a way that is understandable to regular people. The goal of this assignment is to verify the current state of 3D graphic visualisation available for the need of textile industry and then with the cooperation with Arahne d.o.o. develop a module for 3D visualisation of weave that works on Linux operating system. The module is a part of a program package for development of weave called ArahWeave and uses a web browser to display the model of the weave. To display a 3D model in the browser we use WebGL standard. When the module is called from the main program, the parameters of selected weave are created. This data serve as the input parameters for our module. Program interface used to develop our module is Three.js. Based on our input parameters we have to compute the path of each thread in space, apply required properties of width and color, and position the thread on the correct place in the space relative to the start of the weave. The weave is made of warp threads and their transverse weft threads. When the whole weave is displayed we must assure user interaction with 3D model, such as thread removal, transparency adjust and ability to display a wireframe of the model. Included is a comparison of various browsers, that we can use to run our module, in Linux operating system.

Key words: 3D, WebGL, weave, warp threads, weft threads

# Poglavje 1

## Uvod

Moderni tekstilni izdelki imajo zelo raznolike vzorce, ki se dandanes razvijajo z uporabo programov. Ti programi so v preteklosti večinoma omogočali samo 2-dimenzionalno predstavitev na zaslonu računalnika. Z naprednim razvojem programskih jezikov in orodij se je izkazalo, da je že pri načrtovanju vzorcev možno zagotoviti 3-dimenzionalen grafični prikaz. Pri čemer se da uporabiti tudi programske rešitve, kot so WebGL in JavaScript, ki jih lahko uporabimo v različnih operacijskih sistemih. To nam zagotavlja obsežen sprejeto prikaza, ne samo za potrebe razvoja, temveč tudi predstavitev tkanine stranki še pred samim tkanjem. Poleg obstoječih rešitev za prikaz 3D tekstila, ki so na voljo na trgu, obstajajo tudi druge tehnologije, ki smo jih uporabili pri razvoju našega programa. Ena od zahtev je bila uporaba odprtih standardov in tehnologij, kjer je to mogoče. Pričakovana je čim širša sistemska podpora in tekoče delovanje na operacijskem sistemu Linux. Za izpolnitev teh pogojev je idealna rešitev razvoj v HTML, natančneje s pomočjo JavaScripta, v enem od odprtih programskih vmesnikih za WebGL. Dodatna prednost je tudi možnost predstavitve že izdelanih tkanin na spletnih straneh, brez potrebe po nakupu ali inštalaciji dodatnih programov.

Za razvoj 3D aplikacij v HTML s pomočjo JavaScripta je na voljo več rešitev. Vse za prikaz 3D grafike uporabljajo standard WebGL, ki temelji na poenostavljenem standardu za prikaz 3D grafike OpenGL in je enakovreden implementaciji za mobilne naprave OpenGL ES. Torej ni omejen samo na splet, temveč se lahko uporablja tudi na različnih telefonih in tablicah. Izbran je bil programski vmesnik Three.js, ki je eden najbolj uporabljenih višjenivojskih programskih vmesnikov za 3D rešitve na spletu za delo v WebGLju. Namen višjenivojskega programskega vmesnika je zmanjšanje kompleksnosti in hitrejši razvoj aplikacije. Dane imamo vhodne podatke, zgenerirane v glavnem programu ob klicu modula za 3D prikaz. Potrebno je pravilno interpretirati vhodne podatke in jih pretvoriti v primerne parametre za uporabo v 3D prostoru. V Three.js je potrebno zgraditi 3D model niti, glede na podane lastnosti, ji določiti primerne lastnosti barve, ter pravičen potek v prostoru glede na preostale

niti.

Po izrisu celotnega modela tkanine v prostoru, je potrebno zagotoviti interakcijo uporabnika z modelom. Zaželjena je tako manipulacija nad samo tkanino, s spreminjanjem nekaterih specifičnih lastnosti niti, kot tudi navigacija v prostoru v vseh smereh (rotacija okoli osi, zamik vstran in povečava ter pomanjšava modela). Sledi analiza kako je naš 3D model tkanine primerljiv z realno tkanino in opis dobrih in slabih lastnosti naše implementacije. Za zaključek so testirani vsi brskalniki, ki podpirajo WebGL v operacijskem sistemu Linux. Opisano je, kako njihove specifične implementacije standarda WebGL vplivajo na kakovost in hitrost izrisa tkanine.

## Poglavje 2

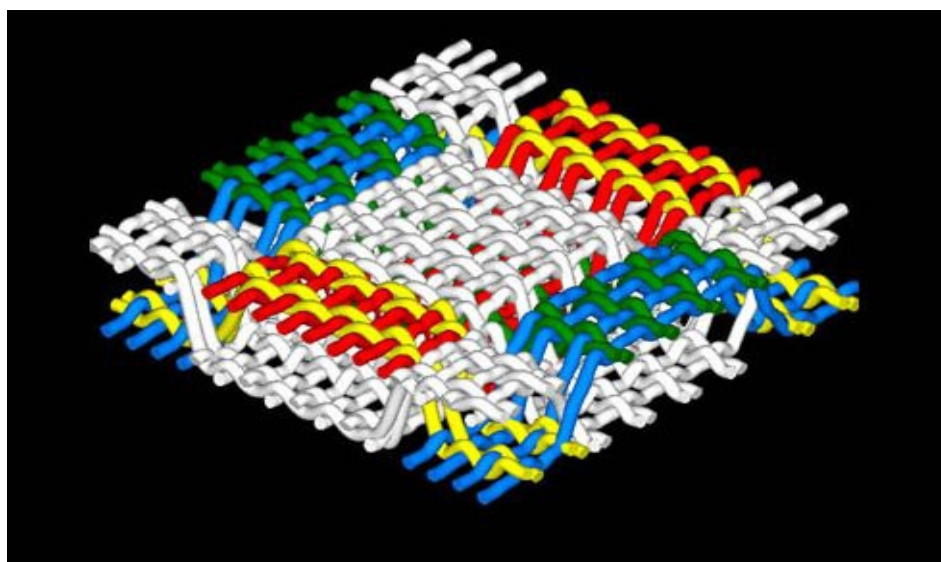
### Računalniška predstavitev tkanin

Računalniška predstavitev tkanin se osredotoča predvsem na dva sklopa. Eden je sam dizajn tkanin, pri katerem nam je prikaz ustvarjenega v 3D prostoru v veliko pomoč pri načrtovanju. Drugi sklop je simulacija lastnosti tkanin, kot so prožnost raztegljivost in čimbolj realno obnašanje tkanine glede na zunanje vplive.

#### 2.1 Dizajn

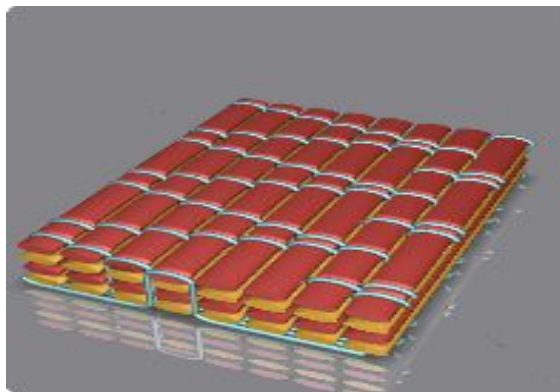
Pri načrtovanju tkanin je 3D pogled v veliko pomoč zaradi boljše predstave o izgledu končnega modela tkanine. Pri pregledu obstoječega stanja na trgu programov za dizajniranje tkanin izstopata dve podjetji:

Podjetje YX-ENDIS nudi tridimenzionalno predstavitev tkanine kot enega od možnih korakov v postopku izdelave tkanine [11]. Rezultat je viden na sliki 1.



Slika 1: Primer 3D simulacije podjetja YX-ENDIS.

Bolj napredno rešitev, ki je prikazana na sliki 2, ponuja podjetje ScotWeave, ki ima v svojem prodajnem sklopu program Technical Weaver specializiran za prikaz najrazličnejših tipov vezav s širokim spektrom uporabnosti, tako v tekstilni industriji, kot tudi v medicini in drugih panogah [16].



Slika 2: ScotWeave Technical Weaver program za prikaz 3D tkanine.

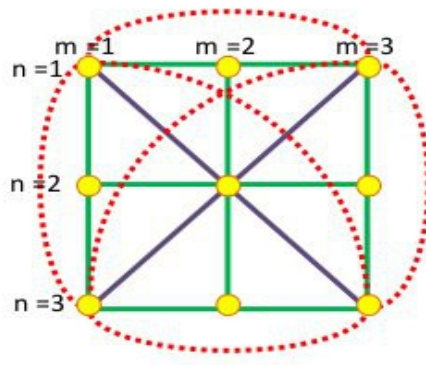
Vse rešitve imajo skupno uporabo grafičnih tehnologij (DirectX, OpenGL) za predstavitev 3D slike tkanine. Ne glede na to ali so realizirane kot samostojni program (ScotWeave), ali del večjega paketa (YX-Weave), imajo omejeno prenosljivost med različnimi sistemi. Obenem specializiranje na listne vezave (enostavni vzorci z veliko stopnjo ponovljivosti), nakazuje na pomanjkanje sposobnosti za prikaz zahtevnih vzorcev tkanine velikih dimenzij.

## 2.2 Simulacija

Simulacija tkanin se loči glede na prikaz tkanin za potrebe industrije (modna, avtomobilska, ...) in na grafične prezentacije, kjer je pomembno čim bolj realno obnašanje tekstila, za zagotovitev čim bolj realne simulacije objekta v prostoru (naprimer obleke v video igri). Za enostavno simulacijo obnašanja tekstila je kot najbolj enostaven model v uporabi masovno-vzmetni model. Ta definira tkanino kot sistem medsebojno povezanih točk z določeno maso. Točke so povezane med seboj z vzmetmi [21].

Na sliki 3 je razvidno, kako so točke z maso medseboj povezane z vzmetmi. Vsaka vzmet ima določen faktor blaženja in maksimalno dolžino raztega. Točke so med seboj povezane s tremi tipi vzmeti, ki skrbijo za predstavitev vplivov različnih sil na tkanino. To so strukturne, križne (angl. shear) in upogibalne. Na sliki 3 zelena barva predstavlja strukturne, črna križne

in rdeča točkasta upogibalne vzmeti.



Slika 3: Prikaz točk povezanih z vzmetmi pri masno-vzmetnemu modelu.

Za izračun sil med točkami lahko uporabimo kar Newtonov zakon gibanja, ki definira silo, kot produkt mase in pospeška. Sile, ki delujejo na posamezno točko se delijo na notranje (razteg, strižna sila in upogib), ter zunanje (veter in gravitacija). Potrebno je rešiti problem, kako čimbolj učinkovito preračunati sile, ki delujejo na masne točke [17].

Poleg simuliranja čim bolj realnega obnašanja tkanine pod vplivom različnih sil, je poseben iziv simulacija detekcije trkov z drugimi objekti v prostoru. To je še posebno pomembno v modni industriji, kjer se uporabljajo simulacije za prikaz interakcije tkanine na modelu človeka.



# Poglavje 3

## 3D grafika – orodja

Predstavljena so orodja uporabljena pri razvoju programa, uporabljena strojna in programska oprema, spletni standardi in razlike med brskalniki, ki jih lahko uporabimo za izvajanje programa.

### 3.1 Strojna in programska oprema

#### **Računalniški sistem**

Razvoj v podjetju Arahne d.o.o. je v potekal na 27 inčnem računalniku iMac, z operacijskim sistemom OSX 2.6.8 Snow Leopard, procesorjem Intel core i5 2500, grafično kartico AMD Radeon HD4850 ter 4GB systemskega spomina. Kot programerski urejevalnik je bil uporabljen TextWrangler podjetja Bare Bones Software. Za nadaljni razvoj je bil uporabljen računalnik z AMD Athlon 64 X2 procesorjem, 2 GB systemskega spomina, z grafično kartico AMD Radeon HD 6850, ki vsebuje 1GB grafičnega spomina na operacijskem sistemu Ubuntu Linux 12.04. Uporabljen je bil odprtokodni urejevalnik gedit. Uporabljen je bil tudi prenosnik ASUS A6Km s procesorjem AMD Turion 64 in 1 GB systemskega spomina, ki vsebuje grafično kartico Nvidia GeForce Go 6300, ter Notepad++ kot programerski urejevalnik. Operacijski sistem na prenosniku je Windows 7. Za testiranje programa so bili uporabljeni sledeči brskalniki: Mozilla Firefox (verzije 10-15), Google Chrome, Opera (verzija 12.11) ter Apple Safari 5.0. Internet Explorer ne podpira WebGL-a zato ni bil uporabljen. Safari 5.0 privzeto ne podpira WebGL, zato je potrebno vključiti podporo v sistemskem menuju. Enako velja za Opero.

#### **Standardi za prikaz 3D grafike na spletu**

**VRML** omogoča 3D predstavo interaktivne vektorske grafike. Za njegovo delovanje v brskalniku je potrebna inštalacija posebnega vtičnika in zato ni primeren za splošno uporabo. **X3D** je naslednik VRML standarda z nekaterimi izboljšavami. Ravno tako zahteva za svoje

delovanje poseben vtičnik.

**X3DOM** je predlagan model sintakse in implementacija skriptne knjižnice. Cilj je integracija brez vtičnika, z uporabo WebGL in JavaScript-a v HTML5. Deluje v vseh brskalnikih, ki imajo podporo za WebGL. Omogoča integracijo in manipulacijo X3D scen, ki se izrisujejo z WebGL. Je odprto-kodni okvir, ki omogoča definiranje 3D scene in njeno obnašanje, brez potrebe po uporabi nizkonivojskega JavaScript- a ali GLSL kodiranja.

**HTML5** je najnovejša implementacija spletnega standarda HTML, ki omogoča 3D prezentacijo na elementu platna (angl. canvas). Funkcije, ki jih nudi so: Risanje (trikotniki, krivulje, črte, bezier in kvadratne funkcije), efekti (senčenje, linearni in radialni gradienti, alpha transparentnost, komponiranje), transformacije (skaliranje, rotacije, tranzicije, matrika transformacij). Za risanje se uporablja JavaScript. Canvas element omogoča vse osnovne operacije za realizacijo bolj kompleksnih modelov.

Novosti, ki jih prinaša so:

- novi elementi (<video>, <audio>, <section>, <header>, <footer>, <aside>, ...),
- novi atributi,
- canvas element za 2D in 3D izris,
- predvajanje video in audio vsebin,
- podpora „offline“ aplikacij,
- lokalni prostor za shranjevanje podatkov,
- lokalna SQL podatkovna baza,
- editiranje dokumentov,
- drag-and-drop,
- upravljanje z zgodovino brskanja,
- MIME tipi in registracija rokovanja s protokoli,
- mikro podatki (microdata),
- Web storage (shranjevanje podatkov lokalno, podobno kot piškoti vendar z večjo prostorsko kapaciteto in izboljšanim programskim vmesnikom).

Novost v HTML 5 je tudi podpora za CSS verzije 3, ki omogoča nove izbire, nove lastnosti, animacije, 2D in 3D transformacije, zaobljene robove, efekte senc in možnost prenosa pisave (angl. fonts). Uradno standard še ni dokončan [9].

**JavaScript** je popularen skriptni jezik podprt v vseh spletnih brskalnikih in ostalih spletnih

orodjih [7]. Je lahek (v smislu porabe virov), interpretni, objektno orientiran jezik s funkcijami prvega reda. Omogoča uporabo interaktivnih funkcij na spletnih straneh, ki so drugače statična. Namen jezika je v manipulaciji strani brez potrebe po osvežitvi celotne spletne strani. Uporabljen je za interaktivne elemente spletnih strani, kot so navigacijski gumbi in razni menuji. Ob tem se uporablja tudi za identifikacijo strani za potrebe raznih analitičnih strežnikov, ki zajemajo statistične podatke o prometu. Razvit je bil iz Netscapovega LiveScript jezika. Nima nič skupnega z programskim jezikom Java. Koda se pred izvajanjem ne prevede v strojni jezik, temveč se program v JavaScriptu izvršuje sproti vrstico za vrstico. Standard JavaScripta je poznan kot ECMAScript. Od leta 2012 vsi moderni spletni brskalniki polno podpirajo ECMAScript 5.1, v pripravi je že novejši standard verzije 6 [20].

Podpira objektno, imperativno in funkcionalne načine programiranja. Je del brskalnika in se primarno uporablja za izdelavo izboljšanih uporabniških vmesnikov in dinamičnih spletnih strani. Velik vpliv na sintakso jezika je imel programski jezik C, pa tudi Java, Perl in Python [13]. Način poimenovanja je izbran iz programskega jezika Java. Ključna načela oblikovanja so povzeta po Self in Scheme programskima jezikoma [5].

**WebGL** [6] je spletni standard osnovan na OpenGL ES in zagotavlja API za 3D grafiko. Uporablja OpenGL jezik za programiranje senčnikov GLSL in HTML5 canvas element za prikaz v brskalniku. Dostopen je preko Documents Object Model (DOM) vmesnika [14]. Kot del JavaScripta je na voljo avtomatsko upravljanje s spominom, za kar skrbi brskalnik. Cilj WebGL je strojna pohitritev 3D grafike, brez uporabe posebnih vtičnikov s podporo v vseh brskalnikih ne glede na uporabljeno platformo. Standard odlikuje široka podpora (Firefox 4.0+, Chrome 9+, Safari 5.0+, Opera 12+, razen za Internet Explorer. Internet Explorer se da nadgraditi s Chrome Frame ali IEWebGL). Med mobilnimi brskalniki so podprti: Nokia N900, BlackBerry Playbook, Firefox for mobile in Opera mobile 12 na Androidu. Uporablja strojno pospešitev in GPU za renderiranje scene. Podprt je tudi na mobilnih platformah (Android, BlackBerry Playbook, Firefox for mobile, Opera Mobile 12).

WebGL je v osnovi eden od načinov izrisa na `<canvas>` elementu, kar mu omogoča enostavno kombiniranje s HTML in ostalimi elementi spletnih strani, ki so lahko nanizane nad ali pod 3D vsebinami. V primerjavi z OpenGL ne podpira: 3D tekstur, računanja s plavajočo vejico z dvojno natančnostjo, vertex atribut 0 ima spremenjeno funkcionalnost, omejeno NPOT (Non-power of two) teksturno podporo [7]. To so teksture, katerih velikost ni

potenca števila 2 (2, 4, 8, 16,...), ampak jo lahko poljubno določimo.

Prednosti WebGL:

- API baziran na poznanih in splošno sprejetih 3D grafičnih standardih,
- Cross-browser in medplatformska kompatibilnost,
- Integracija s HTML vsebinami, med drugimi nivojsko komponiranje (layered compositing), interakcija z ostalimi HTML elementi in standardni HTML mehanizmi za ravnanje z dogodki (event handling),
- strojna pospešitev 3D v okolju brskalnika,
- skriptno okolje, ki omogoča enostavno izdelavo 3D grafičnih prototipov,
- ni potrebno prevajanje pred izvajanjem in vizualizacijo grafike..

WebGL je nizkonivojski API, kar pomeni, da programiranje ni enostavno. Jezik OpenGL za programiranje senčnikov je sam zase celo programabilno okolje. Zato tudi realizacija enostavnih stvari zahteva kar precej kode. Potrebno je naložiti, prevesti in povezati senčnike, postaviti spremenljivke, ki jih je potrebno podati senčnikom in preračunati matrično matematiko za animacijo oblik. Osnove potrebne za delo z WebGL jezikom so:

- GLSL, jezik uporabljen za programiranje senčnikov v OpenGL in WebGL,
- računanje z matrikami za potrebe transformacij,
- točkovni medpomnilniki potrebni za podatke o poziciji točk, normal, barv in tekstur.

Za razvoj standarda skrbi Khronos Group [6].

**Three.js** je visoko-nivojska knjižnica za prikaz 3D grafike v HTML5 `<canvas>` elementu, SVG ali WebGL načinu. Pisana je v skriptnem jeziku Javascript in je trenutno (november 2012) ena najbolj priljubljenih rešitev za delo s 3D grafiko na spletu [10]. Razvija jo več kot 30 razvijalcev. Sodi med najbolj popolne WebGL okvire za razvoj 3D grafike [18]. Pokriva matrike, scene, mrežne modele, materiale, senčenje, meglo... Po besedah avtorja projekta je cilj Three.js razvoj lahke (angl. lightweight) 3D knjižnice z nizkim nivojem kompleksnosti [4].

Lastnosti:

- knjižnica omogoča `<canvas>`, `<svg>` in WebGL izrisovalnike,
- efekti ki jih podpira so: stereoskopska slika, efekt prekržanih oči, stereo projekcija,...
- scene: dodajanje in odstranjevanje objektov in megle v realnem času,
- Tipi kamer: perspektivna in ortogonalna (vpliva na način pogleda),

- kontrolerji (način premikanja po prostoru): kroglično (angl. trackball), FPS, po določeni poti,...
- tipi luči: ambientna, usmerjena, točkovna,
- sence: sposobnost objektov za prejemanje in ustvarjanje sence,
- materiali: Lambert, Phong,... Vsi podpirajo teksture, gladko senčenje,
- senčniki: podpora vseh WebGL možnosti senčenja,
- podpora za postprocesiranje,
- podprti objekti: mreža, delci, linije, trakovi, vse s spremenljivimi nivoji detajlov,
- geometrije: ploskev, kocka, sfera, torus, 3D tekst in več,
- modifikatorji objektov: struženje, izbočenje in tubularno,
- nalagalniki: binarni, slikovni, JSON in scenski,
- celoten sklop časovnih in 3D matematičnih funkcij, vključno z: matrično, UV,
- izvozniki za kreiranje Three.js kompatibilnih JSON datotek v vseh priljubljenih programih za 3D grafiko (Blender, CTM, FBX, 3D Max),
- podpora: API dokumentacija je v izgradnji, javni forum in wiki,
- več kot 150 primerov uporabne kode vključno s 3D tekston, modeli, teksture, zvok in ostali podporni dokumenti.

Knjižnica Three.js je bila izbrana za razvoj programa kot najbolj racionalna rešitev. Omogoča hiter razvoj aplikacij in zadovoljive performanse za dosego pričakovanih ciljev od modula za programski paket ArahWeave. Najpomembneje pa je, da omogoča predstavo 3D tkanine s spletnimi tehnologijami [19].

**DAT.gui** je knjižnica za enostavno vzpostavitev uporabniškega vmesnika za spreminjanje vrednosti spremenljivk v 3D okolju WebGL. Nudi enostavno interakcijo z različnimi opcijami vmesnika. Nudi nam enostavno izbiro z uprabo true in false parametrov, kot določitev poljubne vrednosti drsnika ter zagon poljubne procedure ob kliku na gumb. Lahko se dinamično odzive glede na spremembe v programu, ki vplivajo na različne izbirnike vmesnika [3].

## 3.2 Brskalniki

Pomemben faktor pri izvedbi programa je uporabljen brskalnik, od katerega se pričakuje dobra podpora spletnih standardov, hitra izvedba programa in majhna poraba systemskega spomina. Preiskusili smo naslednje brskalnike: Mozilla Firefox, Google Chrome, Safari, Internet Explorer in Opera [2].

**Firefox:** Podpira WebGL od verzije 4 naprej, kasnejše verzije imajo tudi podporo za „Cross-Origin Resource Sharing“, ki prinaša varnostne izboljšave pri izvajanju na lokalnem sistemu. To so popravki pri nalaganju lokalnih datotek, kot na primer slike za teksture, ki bi se jih lahko uporabilo za zlonameren prevzem kontrole nad brskalnikom. Značilno za Firefox je velika poraba systemskega spomina in neodzivnost brskalnika v fazi inicializacije in izrisa WebGL scene. To pomeni, da je uporabnik primoran potrpeti, dokler WebGL program ni končal s svojim delom, preden lahko nadaljuje z delom, kar ima za posledico zmanjšano multioperabilnost v sistemu. Ima drugačen način prepoznavanja uporabe kolesčka pri miški. Ne upošteva podane barve ozadja. Na Linux operacijskem sistemu slabša hitrost izvajanja programa (nižji fps), v primerjavi z Chrome na istem sistemu, kar je verjetno posledica slabše implementacije WebGL izrisovalnika oziroma neoptimalnega izkoristka grafičnih gonilnikov.

**Chrome:** podpira WebGL od verzije 9 naprej in od verzije 13 naprej tudi „Cross-Origin Resource Sharing“, kar nam onemogoča nalaganje podanih slikovnih datotek s teksturami. To omejitve se da zaobiti z uporabo ukaza `-allow-file-access-from-files` pri zagonu brskalnika. Je več niten, kar omogoča odzivnost ostalih odprtih spletnih strani tudi med fazo inicializacije in izrisa WebGL scene. Na sistemu Linux je blokirano izvajanje na AMD grafičnih karticah, kar se da zaobiti z uporabo ukaza `--enable-webgl -ignore-gpu-blacklist` pri zagonu brskalnika. Boljši izkoristek sposobnosti grafične kartice nam omogoča večjo hitrost izvajanja (angl. fps – slik na sekundo) v primerjavi z Firefox in Opero.

**Safari:** Podpora za WebGL je privzeto izključena, potrebno jo je vključiti v meniju za razvijalce. Z aktiviranjem menija za razvijalce je potrebno v nastavitvah izbrati menu Napredno in tam odključati opcijo »Show Developer menu in menu bar«. To velja za verzijo pri Mac Os X, verzija za Windows WebGL-a ne podpira. Ne deluje na operacijskem sistemu Linux. Bil je uporabljen za testiranje na iMac računalniku pri razvoju v podjetju.

**Internet Explorer** ne deluje na Linux operacijskem sistemu. Tudi v Windows OS ne podpira

WebGL standarda. Kot razlog podajajo varnostne pomankljivosti, ki pa so jih ostali brskalniki uspeli rešiti. Obstajajo rešitve ki nam omogočijo podporo za WebGL v internet Explorerju. To so uporaba vtičnika Instant Reality, Flash-a ali Google Frame-a.

**Opera** podpira WebGL strojno pospeševanje od verzije 12 naprej. Podporo za strojno pospešitev je potrebno aktivirati z nastavitvijo dveh spremenljivk *opera:config#Enable Hardware Acceleration* in *opera:config#Enable WebGL* na vrednost 1 v konfiguraciji brskalnika [8].



## Poglavje 4

### ArahWeave3D program

Predstavljen je problem, ki vključuje predstavitev programa ArahWeave in način, kako pridobimo podatke, ki so potrebni za izris modela tkanine v 3D prostoru. Nato so podrobneje predstavljeni naši vhodni podatki in opisani koraki, ki so potrebni za prikaz tkanine v 3D prostoru. Najprej moramo definirati osnovne lastnosti scene (3D prostor), določiti parametre pogleda (kamera, osvetlitev) in funkcije za izrisovanje objektov v naši sceni (3D prostoru). Nato je opisana izgradnja objekta, ki nam predstavlja nit, definicija (generacija) pripadajočega materiala, izgradnja pomožnih objektov (ravnilo, prikaz lastnosti barv in vmesnik dat.gui), ter izris v 3D prostoru.

#### 4.1 Izris poteka niti v 3D

V nadaljevanju je opisana rešitev prikaza poteka niti v tkanini v 3D prostoru, na čim bolj realističen način. Za podane vhodne podatke, je potrebno izračunati potek niti v prostoru v vseh točkah preseka z sekajočimi nitmi. To je narejeno za vsako nit posebej, ob upoštevanju danih nivojev v vhodnih podatkih. Da prikažemo objekt v 3D prostoru, moramo v izbrani knjižnici za prikaz v 3D prostoru Three.js izpolniti naslednje korake:

- Definirati moramo sceno, ki nam predstavlja 3D prostor.
- V sceno vstavimo eno ali več luči, ki so potrebne za osvetlitev objektov v 3D prostoru.
- Nato lahko v prostor postavimo poljubno število objektov.

Za objekt je potrebno definirati dve lastnosti: geometrijo in material. Geometrija je sestavljena iz množice med seboj povezanih točk in določa obliko objekta. Tri ali štiri med seboj povezane točke tvorijo lice, na katero se prilepi material. Za pravilen prikaz materiala je potrebno izračunati vektorje normale za vsako lice, kar zagotovi pravilno osvetlitev materiala glede na uporabljeno pozicijo luči v prostoru. Volumen niti je sestavljen iz  $n$  krožnih presekov, sestavljenih iz 2 ali več točk, povezanih med seboj. Velikost preseka je

definirana z debelino niti. Za presek niti na določeni poziciji je potrebno izračunati višino glede na vrednost 0 na osi Y. Višina je lahko pozitivna ali negativna, kar je odvisno od nivoja segmenta glede na sekajoče niti. V primeru, da je trenutni nivo enak predhodnemu, se lahko izračun višine izpusti, ampak se samo podaljša dolžina segmenta dokler se nivo ne spremeni. To zagotovi minimalno število točk za prikaz geometrije niti in s tem zmanjša kompleksnost objekta. Kadar nastopi prekinitev niti v tkanini, je potrebno to upoštevati in zagotoviti pravilno prekinitev in nadaljevanje v primeru da je prekinitev začasna. V primeru franže (podaljšana nit brez podanih nivojev) na začetku niti, je potrebno to upoštevati in podaljšati nit na začetku objekta. Zagotoviti je potrebno primeren material, ki je sposoben zadovoljivo prikazati vse lastnosti niti, ki so podane v vhodnih podatkih. Izbran material mora biti sposoben prikaza pravilne barve glede na lokalno osvetlitev, transparentco, lesk, bleščečnost, in biti sposoben prikaza teksture za nit, če je le-ta podana. Združena geometrija in material tvorita objekt, ki ga je potrebno umestiti v 3D prostor. Za vsako nit je podana pozicija, ki jo je potrebno upoštevati pri pozicioniranju niti v naši sceni.

Ko je prikazana celotna tkanina, je potrebno poskrbeti za pravilno pozicioniranje glede na kamero, kar nam zagotovi čimboljši pregled nad tkanino. To dosežemo s premikom tkanine v prostoru in rotacijo za poljuben kot. Potrebno je zagotoviti tudi pravilen izris pomožnih podatkov, ki nam olajšajo predstavo o lastnostih tkanine. To so ravnilo in podatki o uporabljenih barvah niti. Za ravnilo je zelo pomembna pravilna dolžina, saj v nasprotnem primeru dobimo napačno predstavo o velikosti tkanine. Podatki o barvi morajo biti pregledni in reprezentančni.

Po izrisu vseh elementov v sceno, je potrebno zagotoviti interakcijo z objekti v prostoru. Za to je potrebno zagotoviti prosto premikanje po prostoru in interakcijo s tkanino. Za premik po prostoru uporabimo miško, ki nam omogoča rotacijo tkanine in vseh spremljevalnih objektov, premik v stran in povečavo ter pomanjšavo pogleda na tkanino (angl. Zoom). Za manipulacijo tkanine skrbi vmesnik, ki nam omogoča:

- Odstranitev in prikaz niti,
- prikaz mrežnega modela niti,
- spreminjanje transparentnosti v korakih,
- prikaz ali odstranitev ravnila,
- prikaz ali odstranitev podatkov o barvah niti oziroma uporabljenih teksturah.

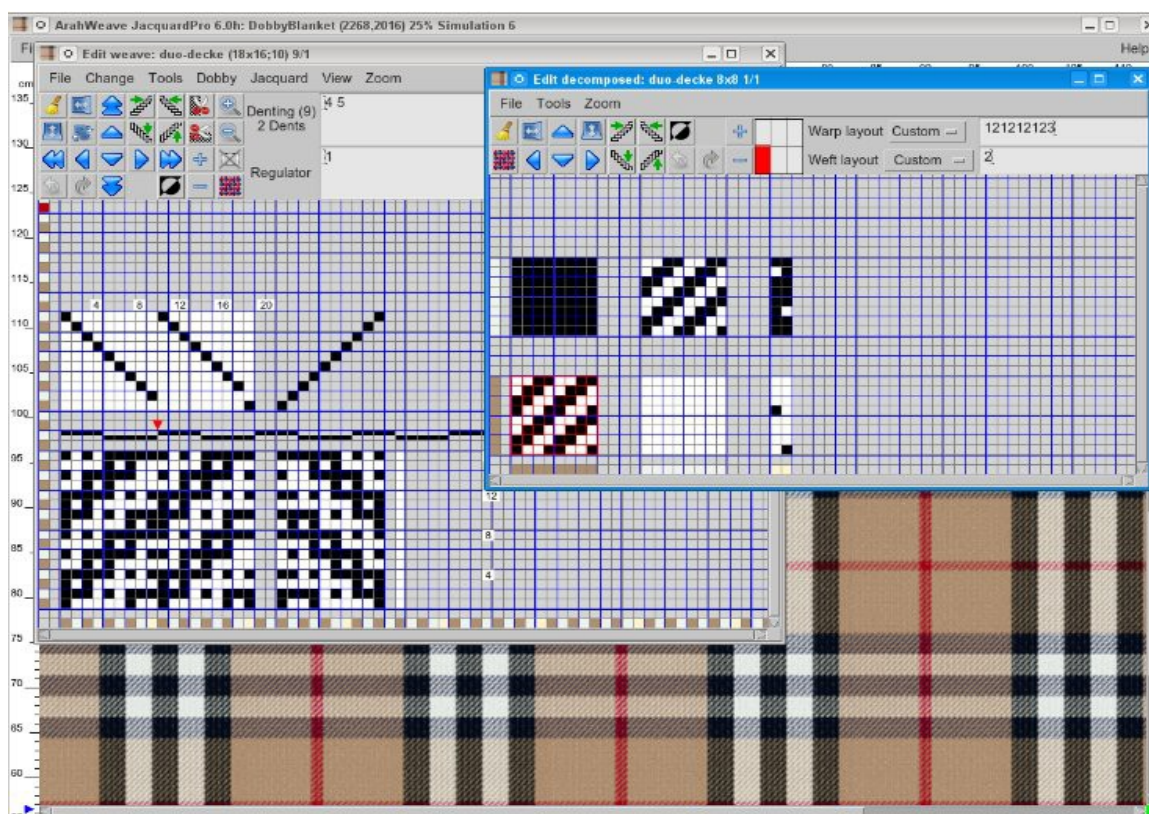
Zagotoviti je potrebno prikaz maksimalno velike tkanine z zadovoljivo kakovostjo prokaza.

### **Program ArahWeave**

Podjetje Arahne d.o.o. je bilo ustanovljeno leta 1992 [15]. Pravno organizacijsko je organizirano kot družba z omejeno odgovornostjo (d.o.o.) v zasebni lasti. Po velikosti sodi med mikro podjetja s tremi zaposlenimi. Osnovna dejavnost je načrtovanje in razvoj računalniške opreme. Specializirano je za razvoj CAD/CAM (Computer Aided Design / Computer Aided Manufacturing - Računalniško Podprti Dizajn / Računalniško Podprta Proizvodnja) sistemov za tkane tekstilije. Glavne stranke so tkalnice, ki se ukvarjajo s proizvodnjo vseh tipov tkalskih proizvodov, kot so: obleke, kravate, šali, etikete, rjuhe, prti, odeje, brisače in tudi razni tekstili na osnovi plastike. Med njihove stranke spadajo tudi dizajnerski studii, pretvorniki, univerze in ročni tkalci [1]. ArahWeave je CAD/CAM program za razvoj tekstila, katerega izgled je prikazan na sliki 4.

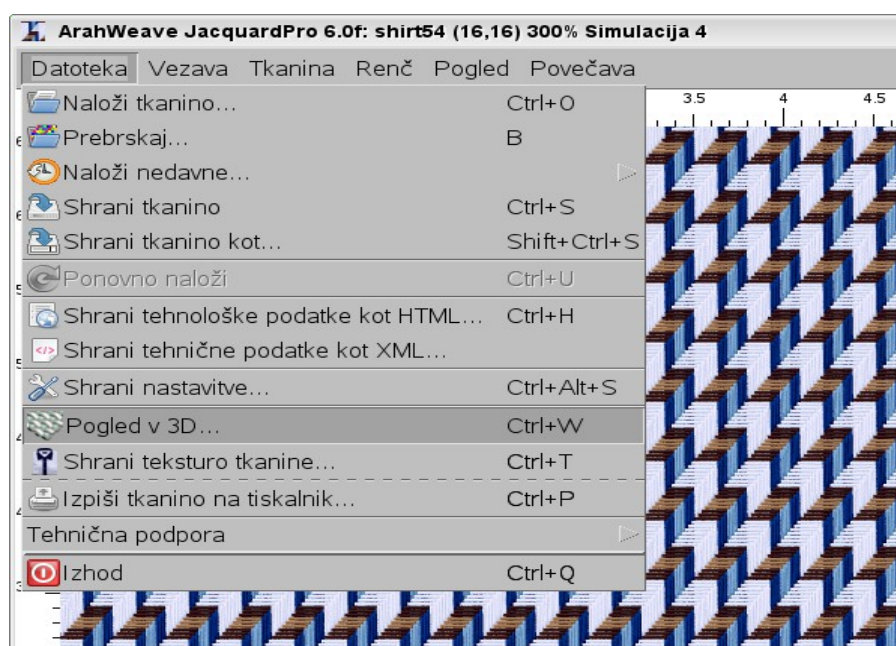
Podpira naslednje funkcije:

- razvoj listnih in žakarskih tkanin,
- vključena je baza več kot 35000 tkanin,
- omogoča urejanje vezave tkanine,
- spreminjanje nitnega vzorca v tkanini,
- editiranje niti v tkanini,
- spreminjanje barv niti,
- spreminjanje gostote tkanine,
- podpira ločeno urejanje nivojev v večslojnih tkaninah,
- generator nitnega vzorca omogoča izdelavo kompleksnih vzorcev niti v tkanini,
- risanje nitnega vzorca direktno v simulaciji tkanine,
- izvoz tehničnih podatkov o tkanini v html in xml formatih,
- preračun cene tkanine,
- realistična simulacijat tkanin v realni velikosti,
- pogled preseka tkanine,
- podpora ekskluzivnih tehnik kot so fil coupe, dodani votki, pretisk, ikat in žakarsko barvno senčenje,
- transformacija žakarskih kart v slikovne datoteke,
- podpora za Dropbox sinhronizacijo datotek.



Slika 4: Izgled programa ArahWeave.

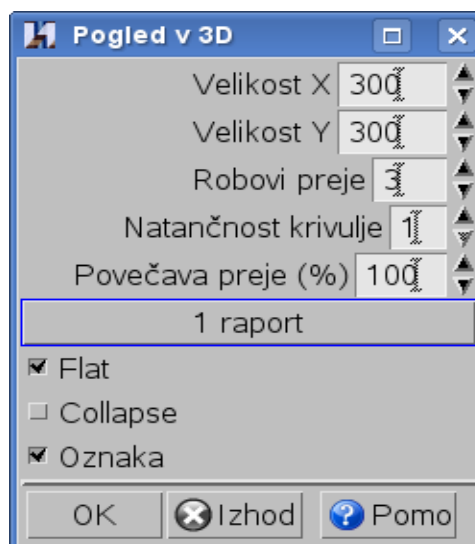
Deluje v operacijskem sistemu Linux. Uporabljena distribucija je SUSE Linux. Program direktno podpira večino statev (Stabuli, Grosse, Bonas, Smit-Textile, Vamatex, Dornier, ...) ter omogoča prenos žakarskih, listnih, vdevalnih in snovalnih podatkov. Vse tehnične podatke lahko izvozimo v HTML ali XML obliki. Možnost dvosmerne izmenjave podatkov z AOP centrom, preko XML standardov. Direktna povezava na statve preko mreže. Program vključuje bazo barv 'PANTONE for fashion and home', ter srednjeevropski standard za predstavitev barv 'RAL' [12]. Program vključuje bazo 35000 unikatnih vezav. Izbirnik za klic modula ArahWeave3D iz menija programa ArahWeave je prikazan na sliki 5.



Slika 5: Program ArahWeave z opcijo za klic modula za 3D pogled ArahWeave3D.

### Vmesnik z opcijami programa ArahWeave3D

Ob izbiri opcije v programu ArahWeave za pogled v 3D se nam prikaže vmesnik podan na sliki 6.



Slika 6: Opcije pri klicu programa ArahWeave3D

Izberemo lahko med naslednjimi lastnosti prikaza v programu ArahWeave3D. Opcija Velikost X nam poda velikost tkanine po osnovi (število niti osnove na osi X). Opcija

Velikost Y pa nam poda velikost tkanine po votku (število niti votka na osi Y). Izberemo število robov tkanine podanih v opciji »Robovi tkanine«, kar nam poda natančnost izrisa preseka niti. Večje kot je število robov tkanine, bolj zaobljena je nit in temu primerno bolj realistična. Od izbire natančnosti prikaza je odvisna maksimalna velikost prikazane tkanine. Pri večjem številu robov tkanine je potrebno več systemskega spomina in tudi čas prikaza se ustrezno podaljša. Vrednosti večje od 16 ne dajejo več vidnih izboljšav. Pri izbiri prikaza robov tkanine 1 se uporabi poenostavljen model prikaza niti. V tem primeru so niti prikazane kot daljice brez 3D volumna. To nam omogoča prikaz precej večjih tkanin z namenom pogleda celotne tkanine. Natančnost krivulje določa število korakov v prehodih med nivoji. Pri vrednosti opcije natančnost krivulje 1 krivulje praktično ni in so prehodi med različnimi nivoji direktni. Večja kot je natančnost krivulje, bolj tekoč je prehod med različnimi nivoji. Vrednosti nad 5 nam ne dajo več vidne izboljšave. Večje vrednosti zahtevajo daljši čas generacije tkanine in porabijo več systemskega spomina, ter direktno vplivajo na maksimalno možno velikost prikazane tkanine. Povečava tkanine nam določa kolikšna velikost tkanine je prikazana v prostoru. Raport avtomatsko nastavi število izvoženih niti v tkanini na eno samo ponovitev vzorca. Opcija izbire Flat poda način prikaza tkanine. Ločuje med prikazom tkanine v normalni obliki kot je v realnosti in prikazom poudarjenih nivojev. Če opcija Flat ni izbrana so nivoji v prikazani tkanini namenoma pretirani. Razlog za to je boljša predstava vezave tkanine in enostaven pregled nad načinom poteka niti v tkanini. Bolj pregledno je katere niti potekajo, po katerem nivoju, oziroma so vezane bolj ohlapno na tkanini. Opcija Collapse preprečuje dvigovanje niti ki dlje flotirajo (gredo zgoraj ali spodaj brez prevezave). Tako da sta samo dva nivoja.

Izbirnik Oznaka nam določa, ali bodo ob izrisu zraven modela tkanine prikazane tudi oznake barv niti z vsemi podanimi podatki. Ti podatki nam povejo debelino niti, njeno barvo podano po mednarodnem standardu in ime barve. To velja v primeru da so vsi podatki na voljo, v nasprotnem primeru se izrišejo samo podatki, ki so podani. To je pomembno zaradi različnih tehnologij zaslonov. Računalniški monitorji na splošno niso sposobni prikaza realnih barv. To je še posebej razvidno na cenejših inačicah in pogosto na zaslonih uporabljenih v prenosnikih.

Po izbiri opcij za prikaz se ob kliku na gumb OK zgenerira datoteka z vhodnimi podatki za program ArahWeave3D. Datoteka je tekstovnega tipa v unicode formatu, v njej so definirane spremenljivke, ki jih lahko JavaScript direktno naslavlja, torej ni potrebne dodatne obdelave

vhodnih podatkov. Nato se zažene izbrani brskalnik (ponavadi Firefox ali Chrome) s programom ArahWeave3D, ki, z upoštevanjem vhodnih podatkov, prikaže tkanino v 3D prostoru.

## 4.2 Vhodni podatki

Vmesnik za vnos vhodnih podatkov je osnovan na tekstovni datoteki, izvoženi iz programa ArahWeave. Datoteka je vključena v HTML kot skripta, kar nam omogoča sklic podanih parametrov kot normalnih spremenljivk v Javascript-u. Sestavljen je iz več segmentov, ki opisujejo tkanino v celoti.

### Lastnosti tkanine

Datoteka z vhodnimi podatki vsebuje naslednje parametre:

- Število niti osnove in votka,
- Število točk, ki generirajo presek niti (od 2 naprej), več niti omogoča boljše predstavitev niti na račun velikosti generirane tkanine,
- Število korakov v krivulji tranzicije med nivoji (od 1 naprej), več kot je korakov bolj tekoče delujejo tranzicije med nivoji, vendar zahtevajo več virov in s tem vplivajo na maksimalno velikost prikaza tkanine,
- Barva ozadja,
- Višina nivojev v več-nivojski tkanini, kjer je višina podana kot fiksno število, ki se kasneje pomnoži z zahtevanim nivojem, da dobimo iskano višino,
- Ime tkanine,
- Opcija za izris ravnila (da/ne),
- Izris podatkov o barvah in gostoti tkanine (da/ne),
- Podan tekst za izpis na vmesniku,
- Tabela nivojev vezave za osnovo in votek (2 tabeli),
- Niz uporabljenih niti v vezavi.

### Tabela nivojev

Tabela nivojev določa nivoje za vsako nit v tkanini. Podani so s črkami, kjer male črke (a, b,

c, ...) podajajo nivoje v globino, velike črke (A, B, C...) pa nivoje v višino. Osnovni nivo je 0 po osi Y. Višina posameznega nivoje je večkratnik vrednosti podane v spremenljivki *layerOffsetYmm*.

Dodana je tudi logika za primer, ko v tabeli nivojev nastopa vrednost 0, saj se v tem primeru na podanem mestu nit v tkanini ne nahaja. To velja za primere tkanine, kjer so določene niti prisotne samo v določenih odsekih vezave. Take niti se pogosto uporabljajo v zavesah.

Poseben primer, ki ga je potrebno upoštevati, je kadar je na začetku vezave dodana franža. Franža je del tkanine, kjer ni podanih nivojev, temveč je nit za določeno dolžino podaljšana. Ponavadi je prisotna na začetku ali na koncu tkanine. Da je franža prisotna, nam pove spremenljivka *fringeRuler*. Če je njena vrednost true, je franža prisotna, v nasprotnem primeru je ni. Dolžino franže dobimo tako, da v tabeli *yarnOffsetCmX* in *yarnOffsetCmY* preberemo prvo vrednost. Ta se v tkanini, kjer franža ni prisotna ponavadi začne pri 0. V nasprotnem primeru nam prva vrednost določa dolžino franže na začetku tkanine. Pojavi se lahko na začetku ali koncu tkanine. V programu ArahWeave3D je lahko le na začetku tkanine. Potrebno je zagotoviti pravilno branje tabele podanih pozicij niti z upoštevanjem dejstva, da je pred prvo pozicijo niti franža. Upoštevati je potrebno dolžino podano v tabeli in jo dodati pri vsakem izračunu dolžine segmentov niti. Potrebno je tudi zamakniti začetek ravnila, da se začne na poziciji prve niti in ne na začetku tkanine kjer je franža, ki se ne prišteje k dolžini tkanine.

### **Lastnosti posameznih niti**

Za vsako nit je podana njena absolutna lokacija kje se nahaja v prostoru, glede na 0 (startno) pozicijo v tabeli *yarnOffsetCmX* za niti osnove in v tabeli *yarnOffsetCmY* za niti votka. Tip niti nam določa katera nit je uporabljena na katerem mestu v tkanini. Preberemo v tabeli *yarnSequenceX* za niti osnove in v tabeli *yarnSequenceY* za niti votka. Tip niti je zaradi manjše porabe prostora podan kot en sam dolg string znakov kjer vrednost »a« v *yarnSequenceX* predstavlja prvo nit v tabelah *yarnColorsX*, *yarnDiameterMmX*, *yarnTransparencyX*, *yarnLusterX*, *yarnTextureX*, *yarnTextureLenghtMmX*, *yarnTextureMultiplyX*, vrednost »b« drugo in tako naprej. Pozicija v tabeli nam pove na katero nit se podatki nanašajo. Torej vrednost na 16 poziciji v tabeli nam pove kakšnega tipa nit je 17 nit v tkanini. Ta razlika nastane zato, ker se indeksi tabele začnejo pri 0. Vsaka nit ima definirano barvo v RGB formatu, debelino v milimetrih, transprenci, luster (bleščečnost), ter v nekaterih

primerih tudi teksturo. V primeru, da ima nit teksturo, potrebujemo poleg podane datoteke s teksturo (slikovna datoteka s končnico .JPG ali .PNG) in poti do te datoteke (običajno je v istem direktoriju kot program, tako da je potrebno le ime slikovne datoteke, ki določa teksturo), še dolžino podane teksture (spremenljivka *yarnTextureLenghtMmX*) in faktor multiplikacije (spremenljivka *yarnTextureMultiplyX*), za zagotovitev pravilnega prikaza teksture po celotni dolžini niti. To je potrebno zato, ker so nekatere teksture ponovljive in večbarvne, kar moramo upoštevati za čimbolj realističen prikaz niti.

### 4.3 Postopek izdelave modela

Predstavljen je način vzpostavitve in definicija lastnosti našega 3D prostora, potek izračuna nivojev v vezavi za vsako nit in določitev pozicije v prostoru na kateri se mora nit nahajati.

#### Inicializacija scene

Da lahko prikažemo izdelan model tkanine v prostoru, je v Three.js potrebno definirati sceno. Scena je definirana s klicem *THREE.Scene()*. Za pogled v sceno je treba določiti tip kamere. Ta nam določa način pogleda v 3D prostoru. Uporabljena je perspektivna projekcija s 50° začetnim vidnim kotom, prikazom od minimalno 5 do maksimalno 10000 enot. Kamero nato premaknemo iz izhodišča, kamor se postavi ob definiciji, na izračunano točko na osi Z podano v spremenljivki *cameraPositionZ*. Vrednost spremenljivke *cameraPositionZ* nam da izračun maksimalne dolžine tkanine po osnovi in votku, kjer se vzame večja vrednost, da se zagotovi prikaz tkanine v celotnem obsegu. V primeru zelo majhnih tkanin, ko je izračunana spremenljivka *cameraPositionZ* manjša od 10, je ta ponastavljena na minimalno vrednosti 10.

Izrisovalnik se definira s klicem *THREE.WebGLRenderer*. Uporabljeni so naslednji parametri: *clearColor: 0xffffff*, *clearAlpha: 1*, *antialias: true*, *maxLights: 8*. Ti nam določijo barvo, ki je uporabljena za izbris scene, v našem primeru je bela, ali se resetira Alpha kanal, privzeta je uporaba glajenja robov in maksimalno število luči v sceni je postavljeno na 8. Določi se barva ozadja, ki je podana v vhodnih podatkih.

Določiti moramo še velikost okna prikaza, ki je skoraj celotna velikost trenutne strani v brskalniku. Izrisovalnik nato postavimo v telo HTML dokumenta. Da je model v sceni viden,

je potrebno definirati še eno ali več luči. Definiranih imamo 6 luči, ki osvetlujejo model z vseh strani (levo, desno, zgoraj, spodaj, spredaj in zadaj). Vse so bele barve in so pomaknjene za 1000 enot iz koordinatnega izhodišča. Toliko luči je potrebnih zato, da je ob rotaciji tkanine v prostoru ta vidna v vseh pozicijah. Moč luči je potrebno pazljivo izbrati, saj je v primeru prevelike osvetljenosti prikaz barv napačen. Barve delujejo preosvetljene in so nerealne. V primeru premajhne moči osvetlitve pa je težko videti lastnosti tkanine. Problem nam predstavlja tudi prekrivanje luči, še posebej stranske v interakciji z lučjo spredaj ali zadaj, saj se med seboj ojačujejo.

Dodamo poslušalnike (angl. *EventListener*, posebna logika, ki se odzove na določen dogodek v sceni, na primer klik gumba miške), za dogodke. Služijo pri interakciji z miško in v primeru spremembe velikosti okna brskalnika. Pri miški spremljamo pritisk gumbov, ki nam povedo, kdaj hoče uporabnik spremeniti pogled na tkanino v prostoru (sceni). Za premikanje v prostoru je upravljen ukaz *THREE.TrackballControls()*, kjer je levi gumb rotacija, desni zamik v prostoru, kolesček ali sredinski gumb spremeni stopnjo povečave. Pri uporabi kolesčka se spremeni kot opazovanja kamere v prostoru. Pri uporabi srednjega gumba (ponavadi gumb kolesčka) pa se približujemo ali oddaljujemo od koordinatnega izhodišča.

Za izris skrbi funkcija *animate()*, ki uporablja klic *requestAnimationFrame()*. Ta nam določi okno 60 krat na sekundo in pove kdaj lahko izrišemo spremembe v sceni. Funkcija *requestAnimationFrame()* je priporočena kot standard za izrisovanje in je podprta v vseh brskalnikih. Funkcija *animate()* se izvede samo kadar je nastala sprememba v naši sceni in je potrebno izrisati popravek, v nasprotnem primeru se ohrani na zaslonu prejšnje stanje. To je potrebno zaradi zahteve po optimizaciji sistemskih virov.

### **Izračun nivojev v vezavi**

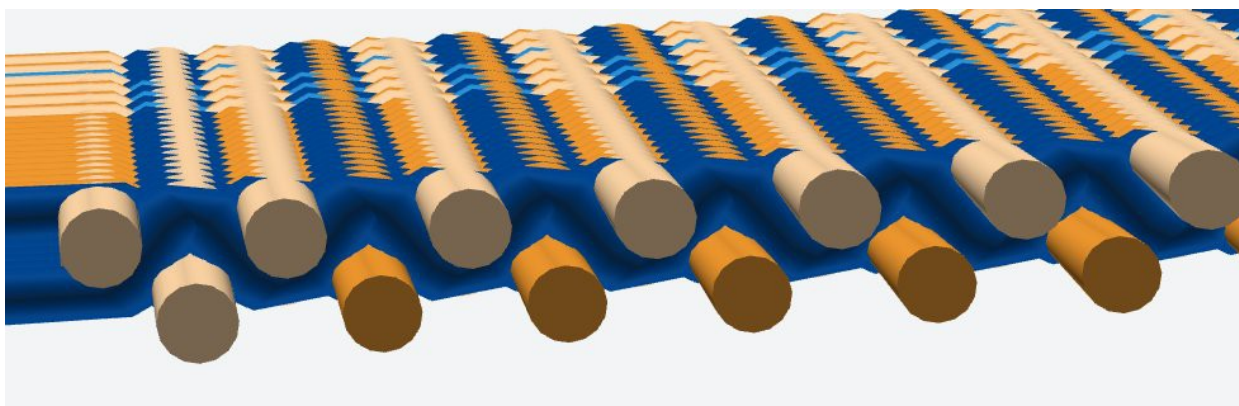
Vsaka vrstica v tabelah *yarnWeaveX* in *yarnWeaveY* predstavlja svojo nit. Nivoji niti so podani v obliki stringa, kjer vsaka črka v stringu predstavlja en nivo niti v tkanini. Najprej preverimo vrednost za trenutno vrstico v tabeli *yarnSequenceX*. To dobimo tako, da pogledamo pozicijo v tabeli *yarnWeaveX* in istoležno pozicijo v tabeli *yarnSequenceX*, ki nam določa tip niti ki jo trenutno gradimo. Če je ta vrednost 0 potem na tej poziciji ni niti in lahko preskočimo logiko za izračun nivojev ter gremo na naslednjo vrstico v tabeli *yarnWeaveX*. Za vsak posamezen vozelj v tkanini, ki je podan v vrsticah *yarnWeaveX*, je potrebno izračunati točno pozicijo v prostoru.

Podatki so podani v obliki string-a kjer male črke (a, b, c...) predstavljajo nizke nivoje, velike črke (A, B, C...) pa visoke nivoje. Od ASCII vrednosti za dan nivo se odšteje določeno število (96 za nizke in 64 za visoke nivoje), da dobimo vrednosti nivojev vezave od 1 naprej v pozitivni smeri za visoke nivoje ter vrednosti od -1 v negativni smeri za nizke nivoje. Dobljena vrednost se pomnoži z vrednostjo dano v spremenljivki *layerOffsetYmm*. Spremenljivka *layerOffsetYmm* nam pove kolikšna je razlika med posamezni nivoji podana v milimetrih. Ko jo pomnožimo z dobljeno vrednostjo nivoja, dobimo višino trenutnega nivoja v 3D prostoru po Y osi. Pri izračunu nivojev moramo v primeru prehoda med različnimi nivoji dodati vmesne korake, ki zagotovijo tekoč potek niti v prostoru. Več kot je vmesnih korakov bolj vizualno tekoč je izgled prehoda med nivoji v tkanini. Število vmesnih korakov nam poda vrednost spremenljivke *curvePrecision*. Njene vrednosti so lahko od 1 (ni vmesnih korakov med nivoji) naprej, vendar pri vrednostih nad 6 ni več vidne razlike v kakovosti. V primeru več zaporednih enakih nivojev, si je potrebno zapolniti le prvega. V tem primeru združimo segmente v eden dolg samostojen segment tako, da podaljšamo vrednost spremenljivke *lenCon*, ki je začasna spremenljivka zadolžena za pomnenje dolžine posameznega segmenta niti. Izračunani nivoji se shranijo v začasno tabelo *tempArray*, ki hrani potek nivojev za vse niti za izris.

Pri generaciji nivojev uporabimo praktično optimizacijo, ki temelji na dejstvu, da je pogosto v tkaninah potek različnih niti enak. Torej imajo enak potek nivojev v vezavi. To velja še posebno pri enostavnih tkaninah, kjer se v bistvu način prepletanja ponavlja. Kadar je prvi element v vrstici tabele *yarnWeave* označen z »=« nam to pove, da je geometrija niti v tej vrstici ponovitev ene od prejšnjih. Katera nit je ponovljena nam poda cifra, ki je podana za enačajem. Ker imamo nivoje za to nit že izračunane, ti niso podani v tabeli *yarnWeaveX*. Zapomniti si moramo vse niti, ki so ponovljive, da jih lahko po generaciji geometrije pravilno izrišemo. Zato za vsako nit pred izračunom nivojev pregledamo celotno tabelo *yarnWeaveX*, od trenutne pozicije naprej, ali obstaja kakšna nit z enako geometrijo. Vrednosti so podane z =##, kjer ## določi vrstico kjer se nahaja originalna nit. Če je ta vrednost enaka trenutni poziciji v tabeli smo dobili ponovljivo nit. Če se vrstica ne začne z »=«, lahko sklepamo, da vrstica vsebuje normalno definirane nivoje za nit. V primeru da dobimo eno ali več ponovljivih niti, si jih zapomnemo v začasni tabeli *multiple*. V tej tabeli imamo torej shranjene pozicije, kje v tkanini se nahajajo niti z istim zaporedjem nivojev. Tabelo *multiple* hranimo do konca risanja vseh ponovljivih niti v prostoru, nato jo lahko izbrišemo.

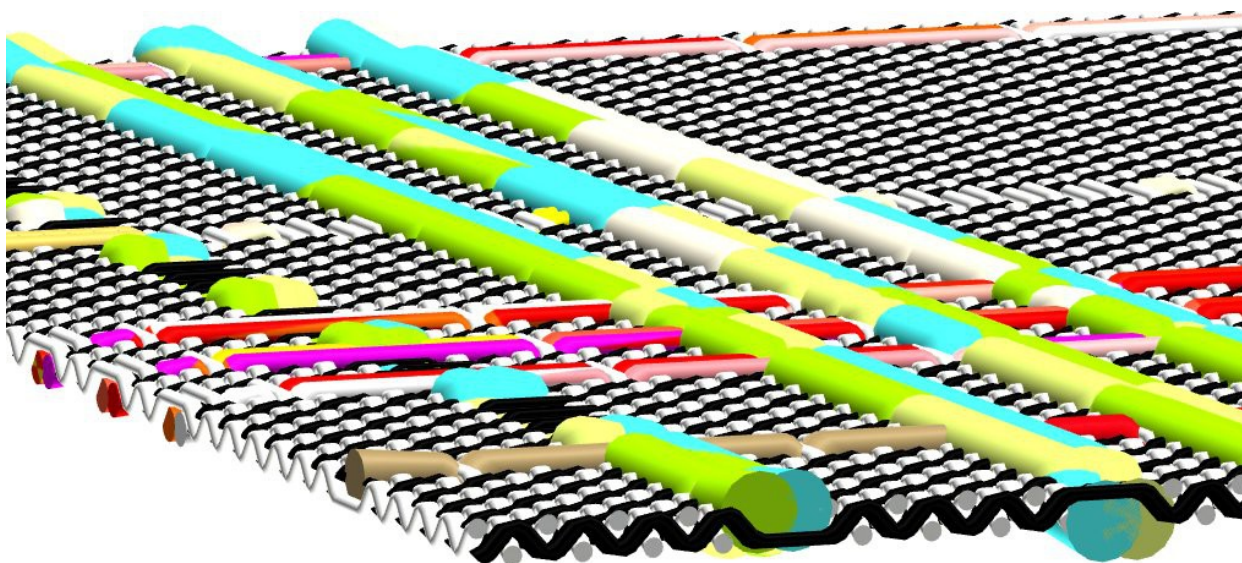
### Pozicija niti v prostoru

V tabelah *yarnOffsetCmX* in *yarnOffsetCmY* imamo podano absolutno pozicijo za vsako nit podano kot oddaljenosti od začetne pozicije 0,0,0 na X,Y,Z osi. Pozicije so predhodno izračunane v programu ArahWeave in so podane preko vmesnika. V tabeli *yarnOffsetCmX* so vrednosti podane na 6 decimalnih mest natančno. To je potrebno zaradi natančnosti prikaza in dejstva, da so nekatere niti lahko zelo tanke. Tkanina se generira od začetne pozicije v točki (X=0, Y=0, Z=0) navzven. Osnova se gradi po X osi od 0 v pozitivni smeri. Votek se gradi po osi Z od 0 v negativni smeri. Tako dobimo tkanino v ravnini XZ. V bistvu je vsaka nit zgenerirana na enak način ne glede na to kam bo v tkanini postavljena. Ko je nit ustvarjena, pogledamo v tabeli za pozicijo kam v prostoru spada in jo premaknemo na zahtevano mesto. Niti votka moramo prej še ustrezno zarotirati za  $90^\circ$  po osi Z, da so pravilno orientirane. Ko je zgenerirana celotna tkanina, se le-ta zasuka v prostoru za  $\sim 90^\circ$ , tako da je postavljena direktno pred kamero in jo vidimo v celoti. Razlog da je zasuk malo manjši od  $90^\circ$  je v boljšem pregledu tkanine, v primeru da ni ravno pravokotna glede na opazovalca, kamero, in s tem dobi boljši 3D efekt. Obenem se vsa tkanina premakne za polovico dolžine osnove po Y osi in polovico dolžine votka po X osi v negativni smeri iz središča. To je potrebno za pravilno rotacijo v prostoru. Ker se rotacija vedno izvede glede na izhodiščno os, deluje po premiku bolj uravnoteženo, ker tkanina rotira glede na njeno masno središče in ne okoli spodnjega levega vogala. Primer prikaza enostavne tkanine vidimo na sliki 7.



Slika 7: Primer pozicioniranja niti v prostoru, enakomeren razmak.

Ker program ne spreminja pozicije niti, je možno da se niti, ki so postavljene preblizu skupaj deloma prekrivajo. To se dogaja v primerih tkanin z zelo veliko gostoto in včasih v primeru da je v tkanini prisotna zelo debela nit. V resnični tkanini bi se te niti deformirale oziroma stisnile glede na razpoložljivi prostor. Program te deformacije ne simulira zato je v tem primeru vidno prekrivanje niti, kot je to razvidno na sliki 8, kjer se v tkanino vstavljene debelejše niti deloma prekrivajo.



Slika 8: Pozicioniranje v prostoru pri različnih debelinah niti.

## 4.4 Generacija niti

### Geometrija

Za generacijo geometrije sta uporabljeni funkciji *makeGeometryX*, za niti osnove in *makeGeometryY*, za niti votka. Potrebujemo naslednje podatke:

- debelino niti,
- širino niti,
- število točk, ki tvorijo segment določa parameter *yarnQ*,
- število segmentov niti,
- višina nivoja za vsak segment niti, podano v tabeli *tempArray*,
- dolžina posameznega segmenta v niti, podana v tabeli *tempArrayLen*,

- kot rotacije segmenta na dani poziciji, podano v tabeli angle,
- oblika preseka niti.

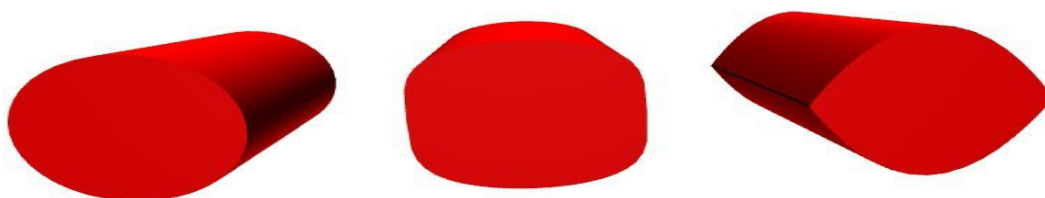
Da dobimo model niti kličemo funkcijo *THREE.YarnGeometry()*, ki ustvari model niti po naslednjem postopku. Glede na vhodne podatke o širini, višini, obliki in številu točk, ki tvorijo vsak segment, najprej izračunamo koordinate točk za en segment in si zapomnemo njihove koordinate po X in Y osi v posebni začasni tabeli. Te koordinate se kasneje uporabljajo za generacijo preostalih segmentov s primernim zamikom po Z osi. Ta korak nam prihrani računanje točk za obliko preseka za vsak nadaljni segment.

Za izračun oblike preseka posameznega segmenta uporabimo formuli:

$$C(t)_x = \frac{w}{2} \cos(2\pi t) \quad 0 \leq t \leq 1 \quad (1)$$

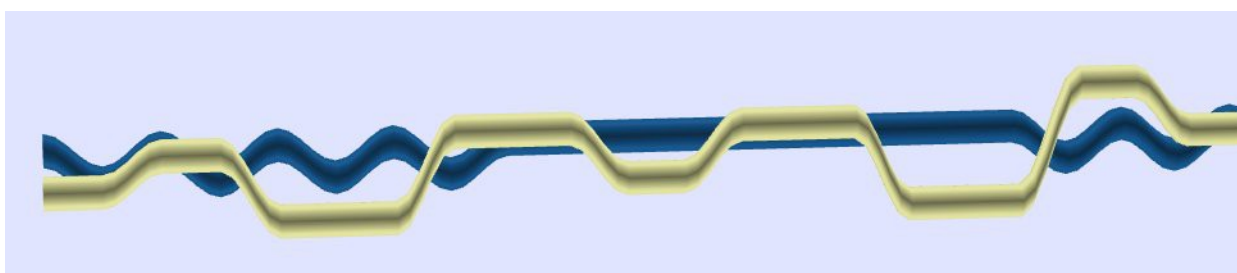
$$C(t)_y = \begin{cases} \frac{h}{2}(\sin(2\pi t))^n & \text{if } 0 \leq t \leq 0.5 \\ -\frac{h}{2}(-\sin(2\pi t))^n & \text{if } 0.5 \leq t \leq 1 \end{cases} \quad (2)$$

Enačba opisuje izračun potenčne elipse [9]. Kjer je  $w$  širina preseka niti in  $h$  predstavlja višino preseka niti. Oblika preseka niti je odvisna od potence števila  $n$ . Pri  $n = 1$  je presek okrogel. Pri  $n \leq 1$  ima presek obliko zaobljenega pravokotnika in pri  $n = 2$  ima presek obliko elipse z ostrim robom. Razlog za izbiro potenčne elipse, za prikaz preseka v nasprotju z navadnim krogom, je v možnosti variacije višine in širine niti po potrebi, glede na deformacije v tkanini. Ta opcija sicer zaenkrat ni realizirana v programu temveč se v praksi uporablja krožna oblika predstavitve preseka niti. Na sliki 9 so vidne možne oblike preseka niti pri enaki višini in širini glede na različno potenco  $n$ .

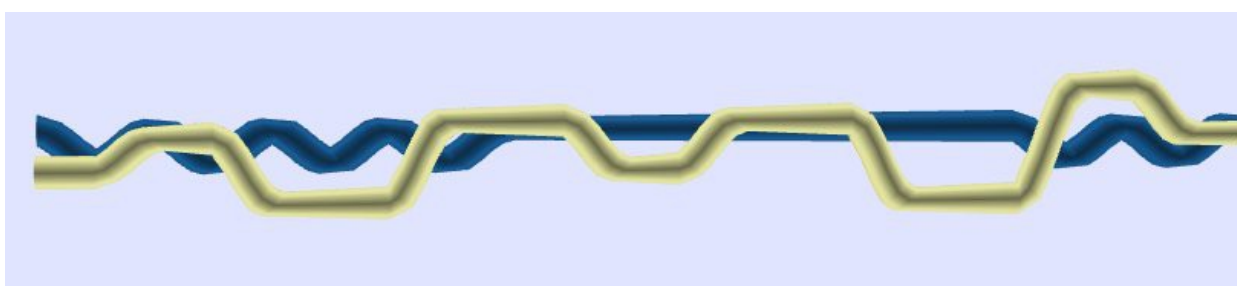


Slika 9: Različne možnosti oblike preseka niti glede na potenco  $n$ . (levo  $n = 1$ , v sredini  $n = 0.5$  in desno  $n = 2$ ).

Na obliko preseka pa poleg potence vpliva tudi izbrana kakovost modela, ki je podan v vhodnih podatkih s spremenljivko *numYarnEdges*. Če je *numYarnEdges* enak 1 je uporabljen alternativni način prikaza z daljicami, kjer niti nimajo pravega volumna. Pri vrednosti 2 je presek niti sestavljen iz samo dveh točk med seboj oddaljenih za širino  $\omega$ . V tem primeru imajo niti obliko trakov. Pri vrednosti 3 ima presek obliko trikotnika, pri vrednosti 4 pa kvadra. Razen pri vrednosti 1, ko so možnosti materiala omejene, so niti v ostalih primerih sposobne prikazati zahtevane lastnosti niti. Torej izgleda tkanina na daleč barvno pravilno. Pri računanju pozicije segmentov je potrebno prebrati pozicijo točk iz začasne tabele in jih primerno popraviti po višini glede na podan nivo segmenta po osi Y. Ta podatek je podan v tabeli *tempArray*. In po poziciji na osi Z, ki nam da oddaljenost trenutnega segmenta od začetka niti. Za vsak segment se nato preračuna rotacija preseka glede na os Y. Ta rotacija nam zagotavlja, da nit pri prehodih med različnimi nivoji ohranja debelino, kar opazno izboljša realizem pri tranzicijah med nivoji. To je posebno opazno pri velikih prehodih iz visokega na nizki nivo, kot je razvidno na slikah 10 in 11.



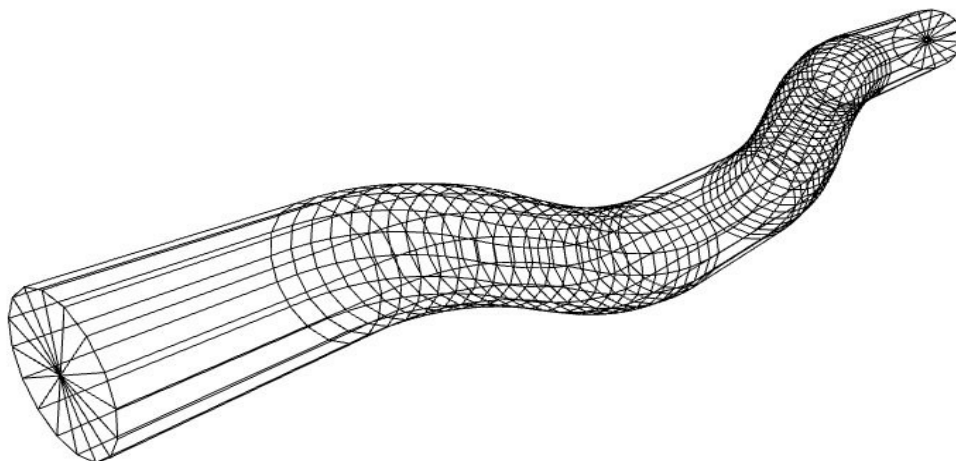
Slika 10: Primer niti brez izračuna kota vmesnega segmenta pri prehodu med nivoji.



Slika 11: Potek niti z izračunom kota vmesnega segmenta pri tranziciji med nivoji.

Dolžina tabele *tempArray* nam da število segmentov, ki tvorijo nit. Ko pridemo do konca tabele *tempArray*, imamo nit zgrajeno, potrebno je še zapolniti oba konca. To se doseže tako,

da na sredini prvega segmenta definiramo točko, ki jo povežemo z vsemi točkami, ki tvorijo segment. Tako iz krožnice dobimo površino kroga. Enako storimo na koncu niti. Pri generaciji točke na sredini zadnjega segmenta je potrebno upoštevati pravilno razdaljo od izhodišča po Z osi. To je potrebno zato, da je točka postavljena na pravo mesto in je konec niti pravilno zapolnjen. Primer žičnatega modela niti vidimo na sliki 12.



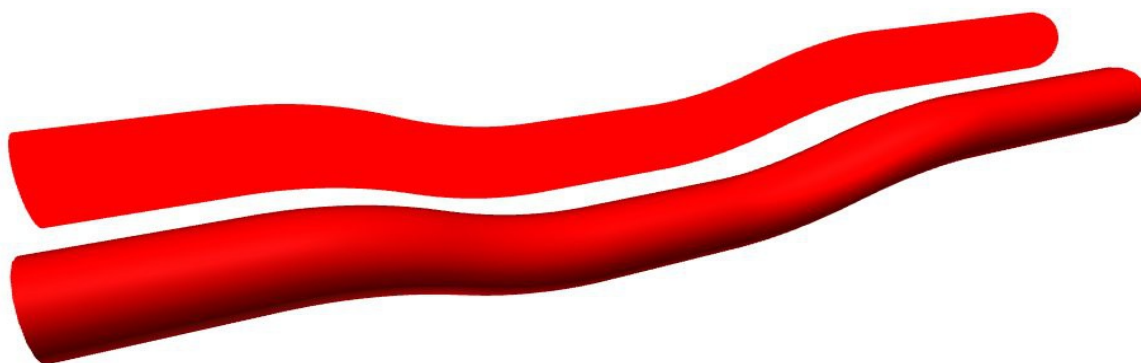
Slika 12: Žičnati model niti.

Ko imamo izračunane vse točke modela niti, moramo za pravilen prikaz materiala izračunati normale za lica vseh pravokotnikov, ki tvorijo ovoj niti. Samo v primeru pravilnega izračuna lic bodo niti ob osvetlitvi pravilno prikazane. To dobimo tako, da vzamemo izračunane normale v vseh štirih točkah, ki tvorijo pravokotnik in iz njih izračunamo normalo lica, ki jo zapišemo v objekt. Na koncu je potrebno izračunati normale za oba konca niti. V nasprotju s preostalo površino niti so lica obeh končnih površin sestavljena iz trikotnikov. Zato za izračun lic uporabimo trikotnike med dvema točkama na obodu segmenta in kot tretjo točko uporabimo sredinsko točko. To storimo za vse pare točk na obodu segmenta dokler nimamo pokrite celotne površine. Paziti moramo na pravilno usmeritev normale, da dobimo pravilen prikaz materiala pod osvetlitvijo. V primeru napačnega predznaka se lahko zgodi, da material ne bo pravilno prikazan.

### Material

Za generacijo materiala se kličeta funkciji *makeMaterialX* za osnovo in *makeMaterialY* za vtek. Material da posamezni niti videz in določa njeno barvo oziroma teksturo,

transparenco, odsevnost in bleščečnost. Izbiran material definira možnosti prikaza različnih lastnosti niti v tkanini. Za prikaz različnih materialov nam Three.js daje na razpolago več možnosti. Najbolj osnovni material, ki nam je na voljo je *MeshBasicMaterial*. Le-ta lahko prikaže samo barvo podano v RGB formatu, pri čemer je prikaz barv neodvisen od osvetlitve objekta. Naprednejši material ki je na voljo, je *MeshLambertMaterial*. Ta material lahko poleg vseh možnosti, ki jih omogoča *MeshBasicMaterial* podpira tudi transparencu, emisijo svetlobe (objekt je lahko vir osvetlitve), ambientno osvetlitev in reflektivnost (stopnja odboja svetlobe od objekta). Naslednji material na voljo je *MeshPhongMaterial*, ki je tudi naš izbran način prikaza, ker izpolnjuje vse zahtevane pogoje za prikaz materiala barve in teksture niti. Poleg vseh parametrov, ki jih podpira *MeshLambertMaterial* material, ima na voljo še Poudarjeno osvetlitev v ozkem snopu svetlobe (angl. Specular highligh), podporo za določitev sijaja materiala, simulacija izboklin in gub (angl. bumpMap) in prikaz z osvetlitvijo za lažen prikaz izboklin in gub brez uporabe dodatne geometrije (angl. normalMap). Ta material je bil izbran ravno zaradi sposobnosti osvetlitve v ozkem snopu svetlobe (specular highligh), ki nam omogoča simulacijo bleščečih niti. Na sliki 13 vidimo razliko v prikazu niti pri različnih materialih.



Slika 13: Razlika prikaza materialov pri na osvetlitvi od zgoraj, kjer je spredaj prikazan »*MeshPhongMaterial*«, zadaj pa osnoven »*MeshBasicMaterial*«.

Vsi materiali imajo skupno lastnost definicijo barve, opcijo prikaza v žičnatem modelu (angl. wireframe), izbira načina senčenja (angl. Shading) in definirano, ali na material vpliva uporaba megle v sceni (angl. fog). Pri nitih s teksturo je potrebno dodatno poznati dolžino

nit in dolžino podane teksture ter faktor multiplikacije, ki je potreben za pravilen prikaz teksture po celotni dolžini niti. Faktor multiplikacije je določen v vhodnih podatkih. Obenem je zaradi večjega realizma začetek vsake teksture zamaknjen za naključno vrednost. S tem se izognemo primeru, da bi izgledale vse teksture na nitih enako, kar se v realnosti v tkanini ne dogaja.

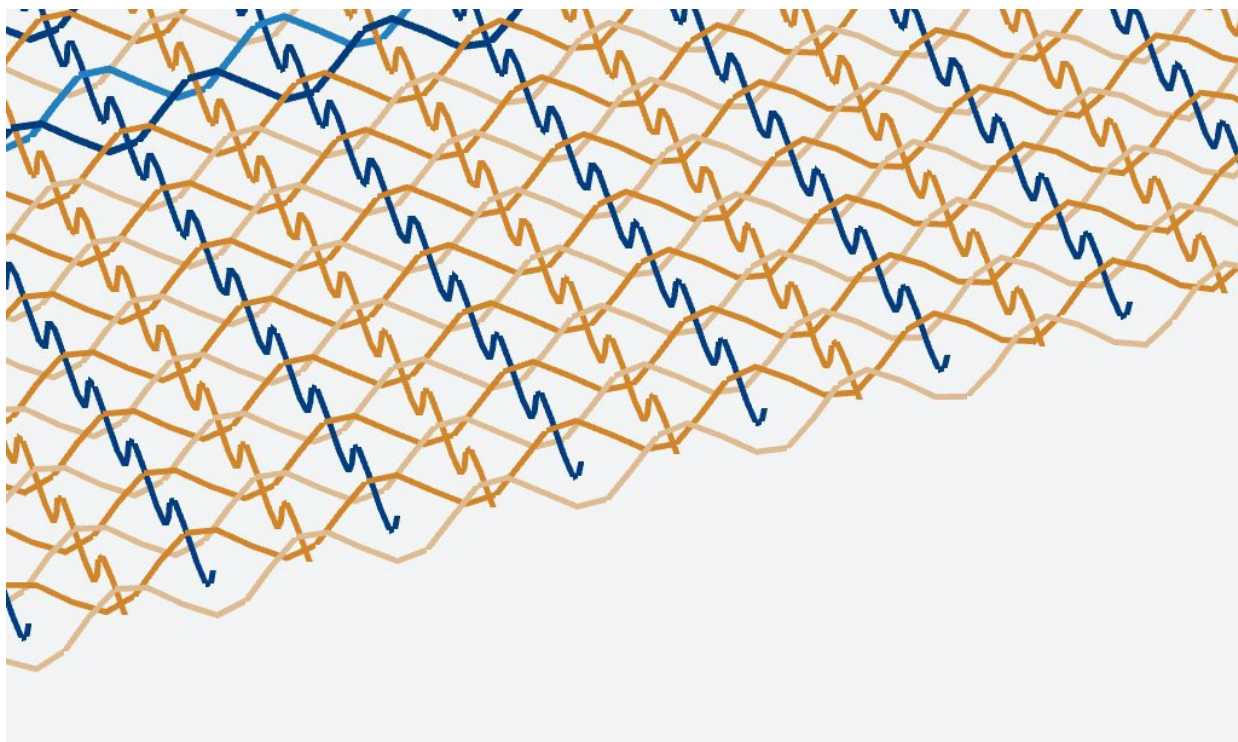
Niti brez teksture imajo definirano barvo, ki je podana v RGB formatu. Nivo za vsak odtenek rdeče, zelene in modre je določen v rangi od 0 do 255, oziroma od 00 do FF heksadecimalno za vsako barvo. V prikazih je rdeča barva podana kot FF0000, zelena kot 00FF00 in modra kot 0000FF. Za vsako nit lahko tudi določimo različne stopnje transparence in bleščečnost (»luster«). Na sliki 14 imamo primer niti s teksturo.



Slika 14: Nit s prilepljeno teksturo.

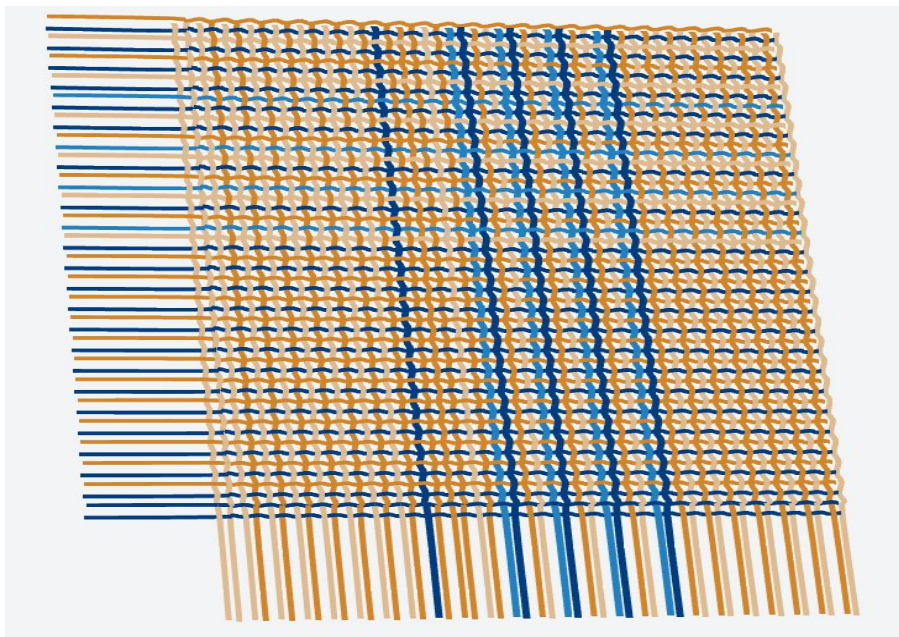
### **Alternativni (poenostavljen) način prikaza**

Poleg pravega 3D prikaza imamo kot opcijo na voljo tudi poenostavljen način prikaza modela tkanine v 3D prostoru. V tem načinu niti nimajo volumna, temveč so prikazane kot serija daljic med podanimi nivoji. Daljice v WebGL nimajo volumna, vendar so v oddaljenem pogledu sposobne prikazati zadovoljiv približek modela tkanine, z nekaterimi omejitvami. Prednost tega načina je možnost prikaza tkanine večjih dimenzij, ki jih standardni način prikaza ne zmore prikazati. Na sliki 15 vidimo primer prikaza niti v 3D prostoru z uporabo zlepk (angl. Spline) daljic.



Slika 15: Tkanina prikazana z uporabo daljic in zlepka (angl. Spline).

Za generacijo modela potrebujemo izračunane nivoje za niti v vsakem presečišču, kjer nit seka eno od drugih niti in s tem spremeni pozicijo. Nivoji so podani v tabelah *yarnWeaveX* in *yarnWeaveY* za vsako nit. Tabela nivojev se uporabi kot vhodni podatek za generacijo zlepka (angl. Spline). Zlepke zgenerira dodatne vmesne točke tranzicije med nivoji, pri čemer je cilj čim bolj tekoč prehod med nivoji. Za rezultat dobimo tekoč potek niti v prostoru. Koliko je nit tekoča, brez stopnic, je odvisno od izbire števila korakov med vsakim nivojem. To vrednost izberemo v vmesniku z opcijami programa, ko kličemo program ArahWeave3D pri izbiri natančnosti krivulje. Če je izbrana natančnost krivulje 1, vmesnih korakov ne bo in bo izrisana nit zelo žagasta, brez dodanih vmesnih točk. Prehodi med nivoji ne bodo imeli oblike krivulje. Natančnost krivulje nad 6 ni priporočljiva, saj se višje vrednosti ne poznajo pri prikazu, temveč le po nepotrebnem povečajo kompleksnost računanja. Na sliki 16 lahko vidimo prikaz cele tkanine z alternativnim načinom izrisa z uporabo zlepka daljic.



Slika 16: Prikaz tkanine z alternativnim načinom izrisa z uporabo daljic.

Prikaz z daljicami ima več pomankljivosti, ki omejujejo njegovo uporabnost kot način za predstavitev tkanine. Primeren je za hiter predogled in za prikaz zelo velikih tkanin, ki jih standarden model ni zmožen prikazati.

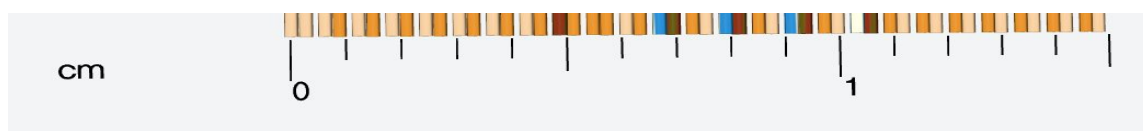
Ena od pomankljivosti je omejena podpora za prikaz materiala. Na voljo je le osnoven material (*LineBasicMaterial*), kar pomeni, da lahko prikažemo samo RGB barve, brez dodatnih izboljšav prikaza. Tako osvetlitev v prostoru ne vpliva na prikaz materiala, kot to omogočata bolj napredna načina prikaza materiala kot sta *LambertMaterial* in *PhongMaterial*. Problem je tudi pomanjkanje podpore za prikaz tekstur. V tem primeru se namesto texture uporabi osnovna barva, ki je vedno definirana za vsako nit v tabeli *yarnColorsX* za niti osnove in v tabeli *yarnColorsY* za niti votka. Ravno tako ne podpira določitve bleščečnosti, ki pride v poštev v primeru uporabe svetlečih niti in prozornosti. Drug problem, ki se pojavi sicer poredko, je nepravilen potek niti med nivoji. Včasih se algoritem zleпка za generacijo vmesnih točk med nivoji zmoti in za rezultat dobimo čudne vozle v poteku niti v tkanini. To se ponavadi pojavi v primerih, ko je v tkanini daljši premor med nasprotnimi nitmi. Kar očitno zmede algoritem za generacijo zleпка, v napačen izračun. Primer napačnega poteka je viden na sliki 17, kjer je potek niti na levi strani preveč ošpičen. Pravilen potek je razviden na desni strani slike, ko je prehod med nivoji bolj tekoč.



Slika 17: Primer napačnega poteka niti.

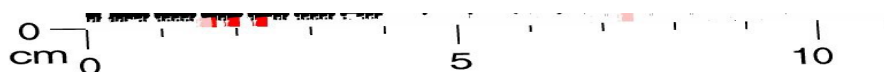
### Ravnilo

Ravnilo je namenjeno lažji predstavi realne velikosti tkanine v prostoru. Ustvari se po celotni dolžini osnove in votka. Za generacijo ravnila preberemo dolžino tkanine za niti osnove iz zadnje vrednosti tabele *yarnOffsetCmX*, ter dolžino tkanine za niti votka iz tabele *yarnOffsetCmY*. Glede na dolžino stranice imamo dva načina prikaza ravnila. Pri tkaninah krajših od 10 cm se izriše ravnilo z milimeterskimi razdaljami in vrednosti se izpišejo na vsak centimeter. Primer centimeterskega ravnila je na sliki 18.



Slika 18: Centimetersko ravnilo pri tkaninah krajših od 10 cm.

V nasprotnem primeru izrišemo segmente ravnila na vsak centimeter razdalje, pri čemer se izrišejo cifre na vsakih 5 centimetrov (torej 5, 10, 15), kot je razvidno iz slike 19.



Slika 19: Centimetersko ravnilo pri velikih tkaninah.

Poleg meterskega je podprto tudi ravnilo za palec. Pri katerem se izrišejo vrednosti za vsak palec. Uporabljena pretvorba je 1 palec = 2,54 cm.

### Podatki o barvi oziroma teksturi niti

Kot opcija pri izrisu tkanine je v programu podana možnost izris podatkov o uporabljenih materialih za vsako nit. Izriše se pravokotnik, ki ima prirejen material posamezne niti za lažjo vizualizacijo barve oziroma teksture. Zraven se izpiše debelina niti podana v Nm (število metrov preje za 1 gram) [22], ime barve in njena specifikacija po standardu. Nad podatki o barvi je podana gostota tkanine po osnovi oziroma votku. Primer izrisa podatkov o uporabljeni barvi v tkanini je prikazan na sliki 20.

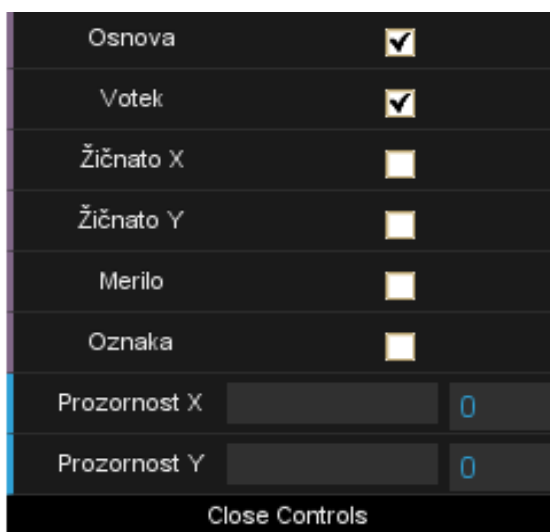
Density Warp 16/cm	
	2/20 Nm 120 S
	2/70 Nm 300 S
	2/8 Nm 120 S
	2/4 Nm 120 S
	2/4 Nm 120 S
	2/12 Nm 120 S
	2/12 Nm 120 S
	2/12 Nm 120 S
	2/12 Nm 120 S
	2/12 Nm 120 S

Slika 20: Podatki o gostoti tkanine, debelini niti, uporabljenih barvah in teksturah.

V primeru, da kateri od podatkov manjka, se izpusti pri izrisu. Ti podatki dajo uporabniku boljši pregled nad uporabljenimi nitmi in njihovimi lastnostmi in olajšajo razumevanje lastnosti tkanine.

### Uporabniški vmesnik DAT.gui

Vmesnik DAT.gui [3], ki je prikazan na sliki 21, omogoča manipulacijo nad modelom tkanine. Izbran je bil zaradi enostavne implementacije potrebnih opcij za manipulacijo tkanine. Na izbiro imamo možnosti odstranitve prikaza osnovnih ali votkovnih niti iz scene in prikaz osnovnih ali votkovnih niti v žičnatem modelu. To dosežemo tako, da odkljukamo izbrano opcijo v vmesniku. Sprememba je takoj vidna v sceni. Določimo lahko prozornost za niti osnove ali votka posebej, vendar le za vse niti hkrati. Prikažemo ali odstranimo lahko tudi ravnilo ter podatke o barvah niti v tkanini. Vsi ti podatki se zgenerirajo tudi v primeru, da niso bili izbrani za takojšnji prikaz ob vzpostavitvi tkanine. Zato so že pripravljene za takojšnji prikaz brez potrebnega čakanja za njihovo generacijo v primeru, da jih želimo prikazati naknadno.



Slika 21: Vmesnik DAT.gui z opcijami.

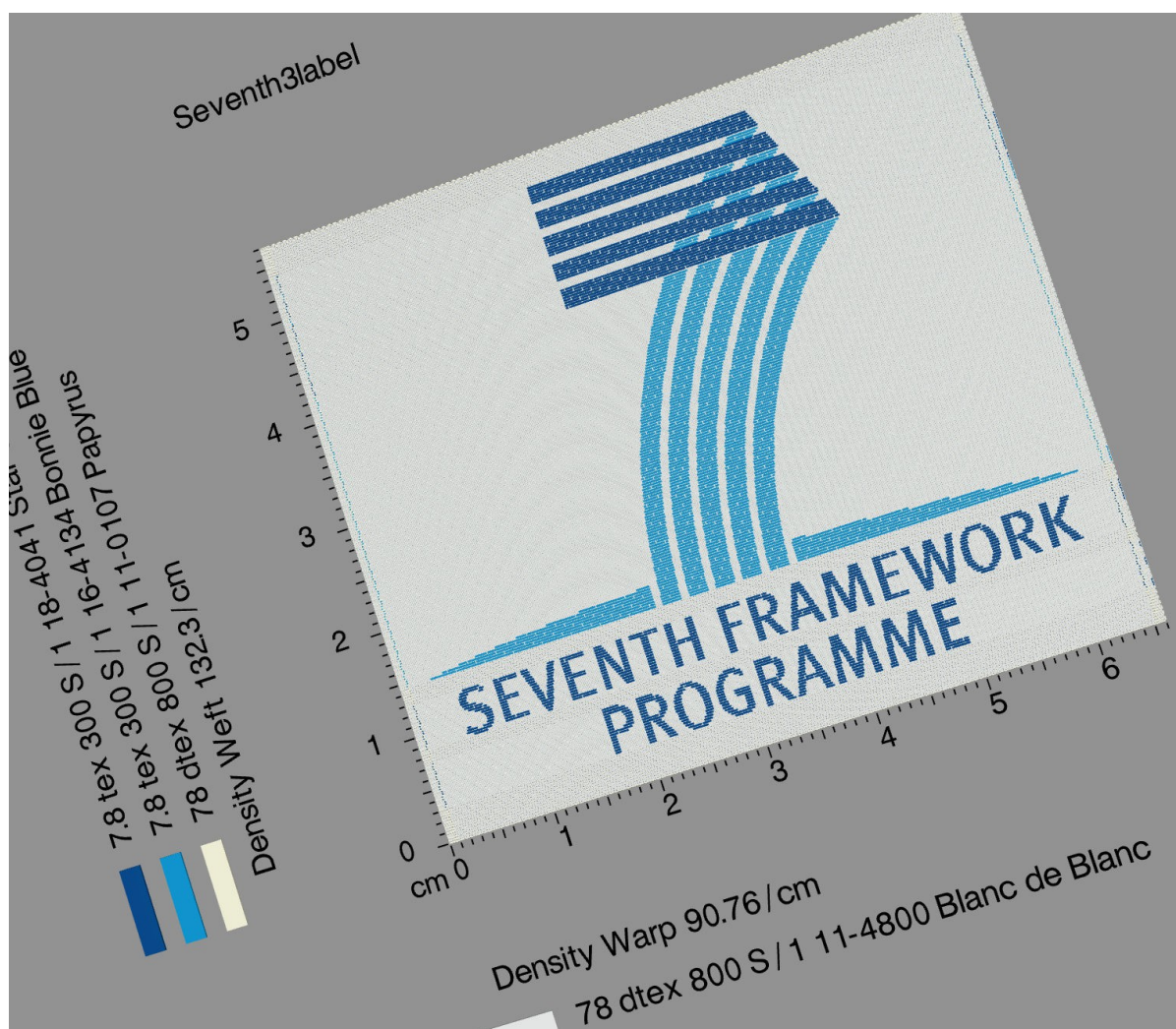
Pri alternativnemu načinu prikaza tkanine z daljicami, je vmesnik podoben. Odstranjeni sta možnosti prikaza tkanine v žičnatem modelu in prozornost niti. Razlog je v tem, da material uporabljen v alternativnem načinu prikaza z daljicami ne podpira obeh možnosti.

## 4.5 Izris v prostoru

Ko je objekt, ki predstavlja nit, ustvarjen s klicem funkcije *YarnGeometry()*, se shrani v posebno tabelo *groupX* za niti osnove in *groupY* za niti votka. Tabeli sta tipa *THREE.Object3D()* in vsebujeta vse niti v tkanini. Razlog za to rešitev je v hitrosti izrisa. Po naši rešitvi se vse niti osnove in vse niti votka izrišejo naenkrat, namesto da bi izrisovali vsako posebej, kar bi bilo časovno bolj potratno. Drugi razlog, je v opciji v vmesniku odstranitve niti iz scene. Pri tem se niti fizično izbrišejo iz scene. Da tkanino lahko ponovno prikažemo, moramo imeti vse niti vnaprej nekje shranjene. Lažje in predvsem hitreje je odstraniti in ponovno prikazati vse niti naenkrat kot pa vsako posebej. Ker je generacija geometrije niti časovno najbolj potraten del programa, se uporablja enostavna optimizacija za pohitritev izrisa. Ob vsaki generaciji gometrije niti se preveri tabela nivojev od trenutne pozicije naprej z namenom izločiti niti z enakim potekom nivojev. Za izris le-teh se porabi isti model, če ustrezajo tudi ostali parametri.

Da lahko ponovno uporabimo ustvarjeno geometrijo za izris niti podane v tabeli *multiple* (tabela hrani niti z enakim potekom nivojev), mora debelina ponovljive niti ustrezati originalni niti. V nasprotnem primeru moramo ustvariti novo geometrijo s primerno debelino niti. Ker se lahko v tkanini pojavljajo različne debeline niti, si ob izrisu ponovljivih niti pomagamo s shranitvijo že generiranih modelov z različno debelino v posebno tabelo. Pri izrisu ponovljivih niti tako najprej pogledamo v začasno tabelo, če že imamo na voljo model s primerno debelino niti. Če je ta na voljo, ga uporabimo, v nasprotnem primeru smo primorani ustvariti nov model niti. Ta optimizacija nam v nekaterih primerih občutno poveča hitrost prikaza tkanine. Opazna je tudi dosti manjša poraba pomnilnika.

Najprej se zgenerirajo ter izrišejo niti votka zatem še niti osnove. V bistvu je vrstni red nepomemben, tako je pač nanoslo tekom razvoja programa. V primeru, da je vrednost spremenljivke *drawRuler* enaka *true* se izriše ravnilo. Vrednost spremenljivke *metricRuler* nam pove ali zgradimo in prikažemo metrsko ali ravnilo za palec. Ravnilo se sicer zgenerira v vsakem primeru, objekt se shrani v spremenljivki, zato da lahko kadarkoli vključimo prikaz v vmesniku *DAT.gui*. Podatki o uporabljenih barvah in ali teksturah so prikazani ob prikazu tkanine, če je spremenljivka *yarnColorNames* enaka *true*. Tako kot ravnilo se podatki zgenerirajo vsakič ob zagonu programa in se shranijo v spremenljivki, da jih lahko kadarkoli po želji izrišemo v sceni ob izbiri opcije v vmesniku *DAT.gui*. Nazadnje se izriše grafični vmesnik *DAT.gui* z vsemi opcijami, ki nam jih omogoča za manipulacijo nad modelom tkanine. Na sliki 22 je prikazana tkanina z ravnilom in podatki o barvi niti.



Slika 22: Izrisana tkanina z ravnilom in podatki o barvah.

## 4.6 Interakcija uporabnika z modelom tkannine

Za interakcijo s programom se uporablja miška. Na voljo so naslednje opcije:

- Ob držanju levega gumba in premiku miške se vrtila kamera (angl. Rotation) okoli centra scene, kar nam da pogled na tkanino iz vseh smeri  $360^\circ$  okoli središča.
- Ob držanju desnega gumba in premiku miške se pogled zamakne v stran v smeri premika (angl. Shuffle). To je uporabno pri povečani tkanini (angl. Zoom in), ko želimo natančneje pregledati vezavo in je prikazan samo izsek celotne tkanine. Omogoča nam premikanje po površini tkanine v smeri premika miške, pri čemer se

kot pogleda ne spreminja.

- S kolesčkom na miški spreminjamo vidni kot pogleda kamere v prostor. Ob rotaciji kolesčka naprej se vidni kot manjša in s tem povečamo pogled na izbrani omejeni del tkanine (angl. Zoom in). Ob rotaciji kolesčka nazaj se vidni kot poveča (angl. Zoom out), kar nam omogoča pogled na večji del tkanine naenkrat. Druga možnost spremembe kota pogleda je ob držanju srednjega miškega gumba (če kliknemo kolesček na miški ta deluje kot tretji (srednji) gumb). Pri držanju srednjega miškega gumba in premiku naprej se kamera premakne bližje proti sredini scene glede na središče trenutnega pogleda v prostoru, kar nam približa objekt, ob premiku nazaj pa se kamera oddalji od objekta in s tem prikaže večji del tkanine.

Z izbiro opcij v grafičnem vmesniku DAT.gui lahko spremenimo lastnosti tkanine za niti osnove ali votka posebej. Možnosti, ki jih imamo na voljo so prikaz niti v žičnatem modelu, spreminjanje transparentnosti, odstranitev vseh niti osnove ali vseh niti votka in njihov ponoven prikaz v sceni, prikaz ravnila ter prikaz oziroma odstranitev podatkov o uporabljenih barvah in teksturah.

## Poglavje 5

### Analiza rezultatov- testiranje

Pri izdelavi programa nas zanimata predvsem dva vidika. Prvi je kakovost prikazanega modela v primerjavi z realno tkanino. Drugi pa je poraba sistemskih virov potrebnih za prikaz preje.

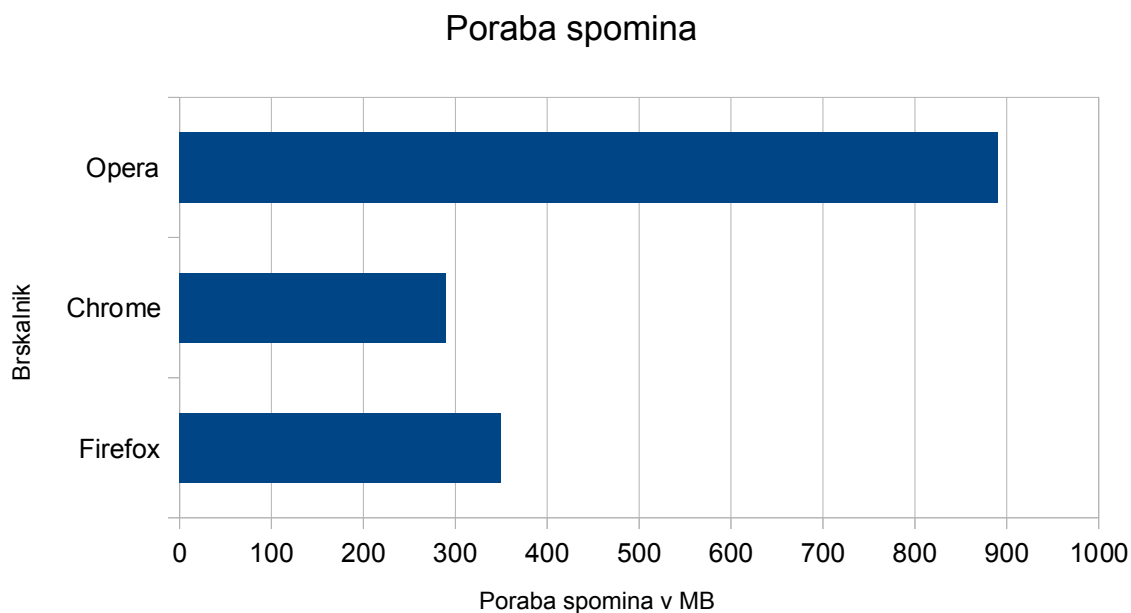
#### **Analiza kakovosti modela**

Model prikazane tkanine je v večini primerov zelo podoben realni tkanini. To velja še posebno za prikaz tkanine z enostavno vezavo. Te lahko prikažemo v velikih velikostih (2000x2000 niti in več), s kratkim časom prikaza in majhno porabo sistemskega spomina. Medtem ko je za kompleksne tkanine model zadovoljiv, so pri natančnejši analizi vidne nekatere pomankljivosti. Posebno pri tkaninah ki vsebujejo različno debele niti in pa tudi pri tkaninah z variabilno gostoto, kjer so niti zelo stisnjene, je včasih, pri pogledu od blizu, vidno delno preisavanje niti, ena čez drugo. To je posledica namernega neupoštevanja deformacij niti ob njihovi interakciji, z namenom zmanjšanja kompleksnosti pri izračunu geometrije niti. Ravno tako v primerih, ko bi se v realni tkanini, niti postavljene zelo blizu ena drugi, te korigirale z rahlim zamikom vstran, kjer je to prostorsko mogoče, v našem programu zaradi upoštevanja absolutne pozicije niti v prostoru to ni upoštevano. Kljub tej pomankljivosti je končni rezultat predstavitve tkanine v 3D prostoru zadovoljiv približek za predstavitev končne slike izgleda tkanine.

Problem se pojavi pri maksimalni možni velikosti prikaza pri žakarskih vzorcih, ko program včasih ni sposoben prikazati dovolj velikega izseka tkanine, da bi lahko dobili primerno predstavo o obliki vzorca. Razlog za to pomankljivost je v kompleksnosti niti, ki sestavljajo žakarske tkanine. Te niti se v veliki meri razlikujejo ena od druge, tako da ne moremo uporabiti naše optimizacije, ki deluje le za identične niti z istimi nivoji. Posledično je maksimalno število niti, ki jih lahko prikažemo v tej tkanini relativno majhno (cirka 500x500), kar je pri žakarskih tkaninah, ki so sestavljene iz velikega števila niti, dostikrat premalo, da bi lahko prikazali celoten vzorec naenkrat.

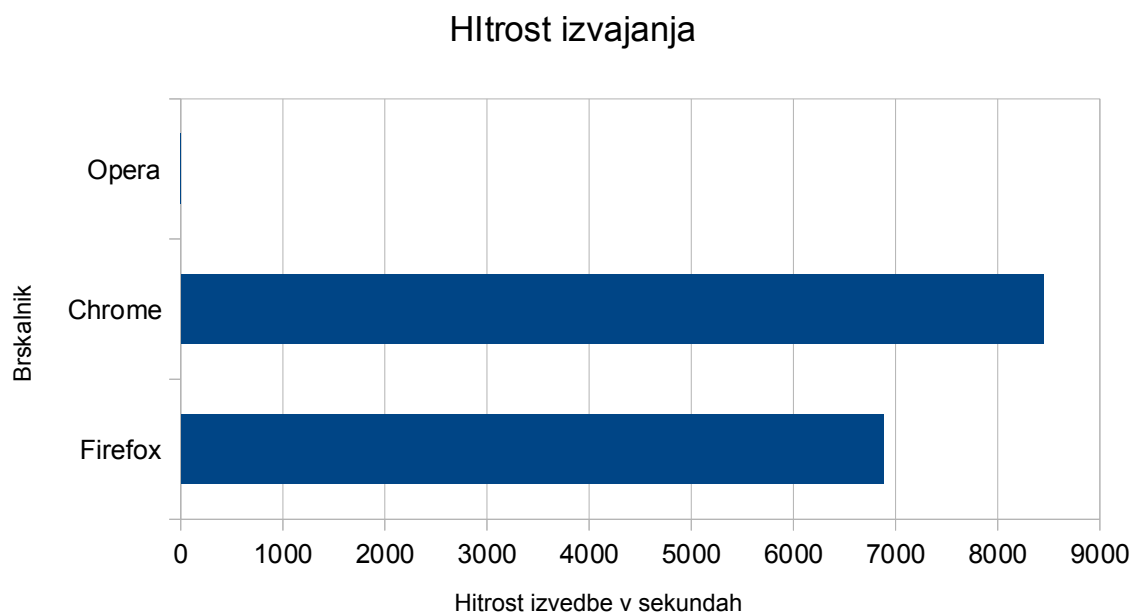
### Test zmogljivosti programa v različnih brskalnikih

Testirani so bili brskalniki s podporo za Linux. Testi so izvedeni v Ubuntu Linux verzije 12.04. V ta namen je bila uporabljena tkanina mnogih prej (preja – predivo počesano in združeno v niti) velikosti 400x400 niti. Za testiranje porabe spomina za Firefox in Opera je uporabljen sistemski nadzornik. Za Chrome pa je uporabljen interni Task Manager, ker je brskalnik večniten in lažje sledi porabi izbranega zavihtka. Zabeležena je poraba spomina brskalnika pred in po izvedbi programa in iz tega izračunana poraba našega programa. Program je zagnan 10 krat v vsakem brskalniku in iz dobljenih podatkov je izračunana povprečna vrednost. Slika 23 nam podaja povprečno količino porabljenega sistema spomina za vsak brskalnik, ki smo ga testirali.



Slika 23: Poraba sistema spomina v brskalnikih.

Čas potreben za izvedbo programa dobimo z izvedbo merilnika v programu samem, tako da si zapomnimo čas ob zagonu in koncu programa, ter nato končnemu času odštejemo začetnega. Dobljene povprečne vrednosti imamo podane na sliki 24. Vrednost za Opero manjka, ker merilnik ni deloval pravilno in zato nimamo verodostojnih rezultatov.



Slika 24: Potreben čas za izvedbo programa.

Opera (verzija 12.02) porabi največ systemskega spomina od vseh testiranih brskalnikov. Čas izvajanja manjka, ker se izpisi v konzolo iz neznanega razloga ne izvajajo pravilno je pa po hitrosti izrisa počasnejša od Chroma, vendar hitrejša od Firefox-a.

Chrome (verzija 20.0.1132.47) ima relativno majhno porabo spomina, ne prikaže tkanine v velikosti kot jih lahko Firefox. Realno je najhitrejši od testiranih brskalnikov, čeprav to po meritvah ni razvidno. Razlog za to je v dejstvu, da je izmerjena vrednost čas od zagona programa, do izvršitve procedure *render()*. Čas od klica procedure *render()* pa do izrisa tkanine na zaslonu pa je različen za vsak brskalnik in je najhitrejši v Chrome.

Firefox (verzija 16.02) lahko prikaže tkanine večjih dimenzij ob dosti večji porabi spomina. Neodziven med gradnjo modela tkanine. Ne uporablja glajenja robov (angl. Antialiasing). Opažena razlika v delovanju brskalnikov, je tudi hitrost sprostitve porabljenega spomina ob zaprtju zavihka s programom. Opera in Chrome to storita takoj, Firefox pa šele čez nekaj časa.

Safari (5.0) je bil uporabljen tekom razvoja na iMac računalniku vendar ni bil stestiran, ker ne obstaja verzija za Linux.

Internet Explorer je izpuščen zato ker ne podpira WebGL in ga ni možno uporabiti na Linux OS.



## Poglavje 6

### Sklepne ugotovitve

Predstavitev tkanine v 3D prostoru, ki nam jo nudi program ArahWeave3D, nam da dobro predstavijo kakšna bo tkanina v resničnosti. To velja še posebej za tkanine z enostavnim vzorcem. Prikaz lastnosti niti (debelina niti, barve, bleščечost) je praktično enaka realnim nitim. Prikaz gostote tkanine in oblika vzorca v naši simulaciji sta primerljivi z realno tkanino. Pri pogledu nad celotno tkanino dobimo dobro idejo kako bo ta izgledala ko bo stkana. Pri pogledu iz bližine na sam potek vezave tkanine je dobro razviden potek niti v tkanini po različnih nivojih, kar dizajnerju pomaga pri vizualizaciji točnega poteka posamezne niti v tkanini. Obenem pa omogoča tudi laiku boljšo predstavilo zgradbe tkanine in ocenitev končnega rezultata.

Dodatna prednost v primerjavi s konkurenčnimi rešitvami, je možnost enostavne integracije modula za prikaz 3D tkanine, v spletno stran stranke. Pri čemer naš program deluje popolnoma neodvisno od glavnega programa ArahWeave. V tem načinu naš modul deluje kot samostojni program. Potrebujemo samo datoteko z vhodnimi podatki, ki jo enkrat ustvarimo v glavnem programu ArahWeave, in jo nato uporabimo pri vsakem zagonu našega programa. To nam nudi večjo fleksibilnost in prenosljivost med različnimi operacijskimi sistemi. Čeprav je program ArahWeave omejen za delo na Linux operacijskem sistemu, lahko v njem izdelano tkanino prikažemo kjer nam to ustreza (Windows, MacOS, internet), neodvisno od glavnega programa.

Glavna pomankljivost je maksimalna velikost prikazane tkanine. Omejuje jo velikost razpoložljivega pomnilnika in hitrost generiranja geometrije niti. Ponovna uporaba že izdelane geometrije niti v tkaninah, ki vsebujejo več niti z identičnimi nivoji poteka, nam sicer omogoča v teh primerih prikaz znatno večje velikosti tkanine. Problem je tudi v dejstvu, da ne moremo vnaprej vedeti, ali bo izbrana tkanina lahko prikazana pri izbranih parametrih kakovosti modela. Zato je včasih potrebno nekaj testiranja nastavitvev vhodnih podatkov preden dobimo optimalne vrednosti za prikaz posamezne tkanine. V primeru, da v tkanini nimamo ponovljivih niti z enako geometrijo lahko prikažemo do 200000 (približno

500x500) niti v prostoru, v nasprotnem primeru (najbolj enostavna vezava ima samo 2 različni niti) pa tudi 10 krat več.

Pomemben faktor pri zmogljivostih je omejitev JavaScripta, da lahko koristi samo eno procesorsko jedro naenkrat. Pogosto pri velikih tkaninah program naleti na »time out« brskalnika preden je dokončal s svojim delom. V tem primeru je potrebno zmanjšati velikost tkanine. Poraba sistemskega spomina je ravno tako omejitev, saj tako Firefox in Opera hitro zasičita prosti spomin, tako da ni nenavadno da pri veliki in kompleksni tkanini porabi tudi po več gigabajtov, odvisno od maksimalnega spomina, ki je na voljo v računalniku.

Kljub tem omejitvam pa naš program zadovoljivo izpolnjuje zastavljene cilje o kakovosti prikaza 3D modela tkanine in enostavnosti integracije na spletne strani, ter prenosljivosti v različnih operacijskih sistemih.

# Viri

- [1] (2012) An Evaluation on Woven Cloth Rendering Techniques. Dostopno na:  
[http://www.ivs.auckland.ac.nz/ivcnz2011\\_temp/uploads/1445/3-IVCNZ\\_11\\_WallaceYuen.pdf](http://www.ivs.auckland.ac.nz/ivcnz2011_temp/uploads/1445/3-IVCNZ_11_WallaceYuen.pdf)
- [2] (2012) Browser Support. Dostopno na:  
[http://www.x3dom.org/?page\\_id=9](http://www.x3dom.org/?page_id=9)
- [3] (2012) Dat-gui, a lightweight controller library for JavaScript. Dostopno na:  
<http://code.google.com/p/dat-gui/>.
- [4] (2012) DobbyWeaving 3D Yarn Crosses. Dostopno na:  
[http://www.yxendis.com/Site\\_UK/DobbyF3.htm#weaving\\_software](http://www.yxendis.com/Site_UK/DobbyF3.htm#weaving_software)
- [5] (2012) Efficient JavaScript. Dostopno na:  
<http://dev.opera.com/articles/view/efficient-javascript/>.
- [6] (2012) Geometric and Mechanical Modelling of Textiles Dostopno na :  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.131.5234&rep=rep1&type=pdf>
- [7] (2012) Getting Started – WebGL public Wiki. Dostopno na:  
[http://www.khronos.org/webgl/wiki/Getting\\_Started](http://www.khronos.org/webgl/wiki/Getting_Started)
- [8] (2012) How to Enable Hardware Acceleration and WebGL in Opera 12. Dostopno na:  
<http://techdows.com/2012/06/turn-on-hardware-acceleration-and-webgl-in-opera-12.html>
- [9] (2012) HTML5 differences from HTML4. Dostopno na:  
<http://www.w3.org/TR/html5-diff/>
- [10] (2012) KadrmasConcepts Blog. Dostopno na:  
<http://www.kadrmasconcepts.com/blog/2011/11/06/subdivision-surfaces-with-three-js/>.
- [11] (2012) Mrdoob/three.js-GitHub. Dostopno na:  
<https://github.com/mrdoob/three.js/>.
- [12] (2012) Mozilla Developers Network. Dostopno na:  
<https://developer.mozilla.org/en-US/docs/JavaScript>
- [13] RAL Color Chart. Dostopno na:  
<http://www.ralcolor.com/>
- [14] (2012) Spletna stran podjetja ARAHNE d.o.o.. Dostopno na:  
<http://www.arahne.si/>.

[15] (2012) SloExport. Dostopno na:

<http://www.sloexport.si/Card.asp?ms=5665388>.

[16] (2012) ScotWeave technical Weaver brochure. Dostopno na:

[http://www.scotweave.com/media/resources/technical\\_weaver.pdf](http://www.scotweave.com/media/resources/technical_weaver.pdf)

[17] (2012) Speeding up Cloth Simulation by Linearizing the Bending Function of the Physical Mass-Spring Model. Dostopno na: [http://www.vis.uky.edu/3dimpvt/assets/papers/17th%20pm/2011\\_3DIMPVT\\_ClothSimulation.pdf](http://www.vis.uky.edu/3dimpvt/assets/papers/17th%20pm/2011_3DIMPVT_ClothSimulation.pdf)

[18] (2012) THREE.js Referance. Dostopno na:

<http://threejs.org/io/s/>.

[19] (2012) WebGL Specifications. Dostopno na:

<https://www.khronos.org/registry/webgl/specs/1.0/>.

[20] (2012) What Is JavaScript. Dostopno na:

[http://what-is-what.com/what\\_is/javascript.html](http://what-is-what.com/what_is/javascript.html)

[21] (2012) WebGL OpenGL ES 2.0 for the Web. Dostopno na:

<http://www.khronos.org/webgl/>

[22] (2012) Yarn numbering system and conversions. Dostopno na:

<http://www.swicofil.com/companyinfo/manualyarnnumbering.html>