

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
BIOINFORMATIKOS SPECIALYBĖ

Bakalauro baigiamasis darbas

**3D modeliavimas ir modelių pritaikymas bioinformatikos
svetainės straipsniams**

3D modeling and model implementation to bioinformatics website articles

Atliko: Bioinformatikos 4 kurso studentė

Meda Škimelytė (parašas)

Darbo vadovas:

Lekt. Irus Grinis (parašas)

Vilnius
2022

Santrauka

Bioinformatikos svetainėje yra pateikiami straipsniai įvairiomis bioinformatikai aktualiomis temomis, iliustruojami piešiniais ir schemomis. Svetainės turinio pajvairinimui buvo iškelta idėja kurti interaktyvius 3D modelius, leidžiančius suteikti skaitytojams kitokį informacijos gavimo būdą. Buvo parašyti septyni straipsniai mikrobiologijos bei molekulinės biologijos temomis, iš kurių penkiems sukurti kokia nors prasme interaktyvūs projektai, papildantys straipsniuose pateikiamą informaciją, o viename iš straipsnių pateikiama iliustracija, sukurta komponuojant 3D modelius.

Raktiniai žodžiai: Svetainė, turinio valdymo sistemos, WebGL, 3D modeliavimas, mikrobiologija

Summary

The current bioinformatics website features articles on a variety of bioinformatics topics, all illustrated with images or diagrams. To bring some diversity into content, an idea was put forward, suggesting that interactive 3D models could be integrated, allowing readers a different way to learn new information. Seven articles have been written on the topics of microbiology and molecular biology, five of which have interactive projects that complement the articles, and one which contains an illustration that was generated using 3D models.

Keywords: Website, content management systems, WebGL, 3D modeling, microbiology

Turinys

Santrauka	1
Summary	2
Įvadas	4
1. Bioinformatikos svetainė	6
1.1. Bioinformatikos svetainės tvarkymas	6
2. 3D modeliavimas	8
2.1. Modeliavimo principai	8
2.2. Blender programinė įranga	8
2.2.1. Pagrindiniai terminai	8
2.2.2. Blender režimai	10
2.2.3. Modifikatoriai	10
2.2.4. Svarbios funkcijos	11
2.2.5. Cycles vaizdavimo variklis	12
2.2.5.1. Cycles projektas	13
2.2.6. Procedūrinis modeliavimas	14
2.2.6.1. Procedūrinio modeliavimo pavyzdys	14
3. 3D modeliai internetinėje erdvėje	15
3.1. GitHub	15
3.2. Objektų vizualizavimas su WebGL	15
3.2.1. Three.js	15
3.2.2. Unity WebGL	16
3.2.2.1. Unity WebGL projektų kūrimas	17
3.2.2.2. Unity projektų naudojimas svetainėje	17
4. Straipsniai	19
4.1. Prokariotinė ląstelė	19
4.2. Mikrobiologijos įvadas	20
4.3. Ląstelių tipai ir organelės	21
4.4. Prokariotai. Bakterijos ir archėjos	21
4.5. Prokariotai. Bakterijų klasifikacija	23
4.6. Eukariotai. Gyvūninė ir augalinė ląstelės	24
4.7. DNR ir RNR	25
4.8. Virusai. Bakteriofagai	26
Rezultatai ir išvados	28
Literatūra	29
Priedas nr.1	31
Priedas nr.2	32
Priedas nr.3	33

Įvadas

Prasidėjus COVID-19 pandemijai, viena aktualiausių temų visuomenėje buvo nuotolinis mokymasis. Kaip žinoma, pradinis nuotolinio mokymosi etapas mokyklose nebuvo sklandus, nors interneto technologijos gali būti puikus įrankis pasiekti įvairių auditorijų, suteikti priemones žinių įgijimui, savęs tobulinimui ir išbandymui. Todėl autoriui svarbus bioinformatikos svetainės kūrimas ir jos tobulinimas, funkcijų plėtimas. Svetainėje pateikiama informacija apie bioinformatikos mokslo sritį, o studentų komanda parengusi daug įvairių straipsnių bei užduočių, padedančių geriau suvokti bioinformatikos prasmę ir vertę. Pradėjus dirbti su svetaine, kuri yra adresu <https://www.bioinformatika.lt/>, buvo pateikiami straipsniai, kuriuose gautos žinios gali būti patikrinamos naudojantis elektroninių mokymų erdve. Ši svetainė taip pat gali būti vertinga visai bioinformatikų bendruomenei – šiuo metu Lietuvoje nėra plačiai žinomos erdvės, kur bioinformatikos specialistai, įvairaus amžiaus bei sričių žmonės galėtų bendradarbiauti, kelti įvairias diskusijas, klausti ir gauti atsakymus. Verta paminėti, kad bioinformatikos svetainė originaliai sukurta tik lietuvių kalba. Autorius apsiėmė svetainės kūrimu tiek lietuvių, tiek anglų kalbomis, norint padaryti svetainę pasiekiamesnę ne tik lietuvių bendruomenei, bet ir kitų šalių lankytojams.

Pasirinkta darbo kryptis – kitoks, kūrybingas sprendimo būdas straipsniams svetainėje pateikti. Tam autorius naudoja 3D modeliavimą kaip vizualinę mokymosi priemonę, akompanuojančią straipsnį. Nėra reikšmingų įrodymų, kurie nurodytų, kad 3D modeliai padeda geriau išmokti ar suvokti tam tikrą temą, tačiau buvo vykdomas tyrimas, kuriame stebimas poveikis rezidentų mokymuisi apie Fallot tetralogiją¹, kur 35 studentai buvo atsitiktinai suskirstyti į dvi tiriamąsias grupes pagal gaunamą Fallot tetralogijos iliustraciją. Viena grupė mokėsi naudodami 2D iliustraciją, kita – fizinį 3D modelį. Visa kita informacija bei mokymo metodai buvo identiški. Tyrimo rezultatuose buvo matomas didesnis pasitenkinimas mokantis naudojant 3D modelį, palyginus su 2D nuotraukomis, be to, naudojantys 3D modelį jautėsi labiau užtikrinti savo gebėjimu gydyti pacientus. Kita vertus, skirtumas tarp žinių nebuvo statistiškai reikšmingas ($p > 0.001$). [LH17] Panašus tyrimas buvo vykdomas norint rasti alternatyvą gamtos istorijos muziejams. 3D modeliai buvo kaip kita priemonė mokymosi procesui esant gamtos istorijos muziejų trūkumui, todėl tyrimas buvo skirtas išsiaiškinti, ar 3D modelių naudojimas gali padėti zoologijos motyvacijai ir mokymuisi. Šiame tyrime buvo trys grupės: kontrolinė, kuri gavo tik tradicines pamokas, praktinė, gavusi tiek tradicines pamokas, tiek praktinius užsiėmimus su biologine medžiaga, ir 3D modelio, gavusi tradicines pamokas bei praktinius užsiėmimus su 3D varliagyvių rūšių modeliais. Vertinama buvo naudojant klausimynus prieš mokymąsi bei baigus kursą. Taip pat atliktas dvipusis ANOVA² testas tarp eksperimentinių grupių. Kitaip nei Fallot tetralogijos tyrime, 3D modelio bei praktikos grupės parodė reikšmingą ($p < 0.001$) mokymosi efektyvumo padidėjimą palyginus su kontroline grupe, kas parodo 3D modelių svarbą mokant zoologijos. [CAB⁺21] Nors autoriaus

¹angl. *Tetralogy of Fallot* – reta įgimta širdies liga, keturių širdies anatominių defektų derinys

²angl. *Analysis of Variance* – statistinis testas, remiantis hipotezių tyrimais, kuris tiria dviejų veiksnių įtaką priklausomam kintamajam. [Mac18]

3D modeliai nėra fiziniai, tikimasi, kad tai svetainės skaitytojams leis geriau pasisavinti pateikiamą informaciją, suteiks didesnę pasitenkinimą skaitytojams ir pajėvins svetainės turinį.

Kuriant 3D modelius buvo svarbūs ir kiti aspektai – naujausia karta daug naudoja kompiuterinėmis technologijomis, žaidžia kompiuterinius žaidimus. Kuriant interaktyvius modelius galima būtų labiau sutelkti moksleivių dėmesį į informacijos gavimą, tikimasi suteikti vizualinio mokymosi. Be to, 3D modeliavimas yra labai populiarus būdas parodyti potencialiems pirkėjams parduodamo produkto funkcijas. Šią technologiją galima naudoti įvairiose pramonės šakose – inžinieriai, dizaineriai, pastatų savininkai, rangovai ir daugelis kitų naudoja šią technologiją norėdami pateikti savo pasiūlymus. [Dyv21]

Darbo tikslas yra naudojantis Blender 3D modelių kūrimo programa sukurti interaktyvius biologinius ir molekulinus modelius, kuriuos vėliau galima pateikti svetainėje kaip būdą vizualiai suprasti ir išmokti straipsniuose pateikiamą informaciją. Šiam tikslui pasiekti užsibrėžti keli žingsniai:

1. temos bei modeliavimo subjektų analizė, vizualios ir tekstinės informacijos paieška;
2. 3D objekto kūrimas, modelio eksportavimas;
3. 3D modelių komponavimas ir programavimas kuriant interaktyvų projektą;
4. projekto straipsniui tinkamos informacijos surinkimas, tvarkymas, rašymas anglų ir lietuvių kalbomis;
5. straipsnio pateikimas ir 3D projekto įterpimas svetainėje.

Integravimui į svetainę pasitelkiama WebGL aplikacijų programavimo sąsaja (angl. *API*), kuriama Unity žaidimų varikliu.

1. Bioinformatikos svetainė

Bioinformatikos svetainės, esančios adresu <https://www.bioinformatika.lt/>, projektas buvo pradėtas 2019 metais. Svetainė sukurta naudojantis WordPress turinio valdymo sistema. WordPress naudojimas svetainės turinio pateikimui yra puikus pasirinkimas. Teiginiui patvirtinti galima išskirti kelis teiginius:

1. WordPress yra labiausiai paplitusi turinio valdymo sistema.
 - (a) Dėl šios priežasties yra daugiau informacijos, kaip naudotis šia turinio valdymo sistema.
 - (b) WordPress turi vieną didžiausių bendruomenių tarp turinio valdymo sistemų – galima lengviau gauti atsakymus į iškilusius klausimus. Taip pat galima atrasti daug informacijos apie kitiems vartotojams iškilusius klausimus, kurie gali sutapti su ieškomais klausimais. Tokiu atveju problemos sprendimas gaunamas labai greitai.
 - (c) Daugiau dėmesio iš didesnių korporacijų, kurios kuria įskiepius klientų naudojimui. Šie įskiepiai gali padėti suteikti vartotojams geresnę svetainės naudojimosi patirtį.
2. WordPress įrašų, jų skirstymas į kategorijas yra paprastas. Naujiems bioinformatikos svetainės turinio kūrimo komandos nariams bus lengva suprasti kaip naudotis platforma bei kurti naują turinį, nes platforma yra draugiška pradedantiesiems.
3. Norintiems prisidėti prie svetainės tvarkymo iš išvaizdos pusės, nebus sudėtinga išmokti naudotis WordPress sistemos įrankiais, nes ši turinio valdymo sistema yra intuityvi.

Bendrai kalbant apie svetainės stilių ir išvaizdą, ji atrodo moderniai. Svetainėje yra pateikiamas labai įvairus turinys. Apsilankius svetainėje iš karto matomas nukreipimas į svetainėje pateiktus straipsnius tiek lietuvių, tiek anglų kalbomis, taip pat trumpas bioinformatikos mokslo apibūdinimas. Yra matoma daug informacijos apie bioinformatikos studijas, nurodomos nuorodos į studijų turinį, kuriame aprašomi visų semestrų dalykai, mokomi studijų metu. Taip pat palankiai nurodoma informacija apie stojimą į bioinformatikos studijas, paaiškinama, ko reikia stojimui ir kaip pasitikrinti, ar norinčio įstoti žmogaus turimi balai pakankami įstojimui. Pagrindiniame svetainės puslapyje taip pat nurodomos nuorodos į tam tikrus skyrelius, kurie suskirstyti pagal sudėtingumą arba turinio tipą, taip pat nurodomas svetainės kūrimo tikslas. Svetainėje daug straipsnių bioinformatikai aktualiomis temomis, kuri pateikiama tvarkingai, pateikiama daug pavyzdžių ir iliustracijų, kas padeda geriau suvokti informaciją straipsniuose.

1.1. Bioinformatikos svetainės tvarkymas

Autorius neturėjo daug galimybių prisidėti prie svetainės tvarkymo, kalbant apie straipsnių vertimą, svetainės dizaino tobulinimą ir informacijos pateikimo vientisumą bei tvarkingumą, nes buvo atliekama straipsnių autoriaus rolė. Buvo sutvarkyti keli elementarūs pastebėti svetainės

trūkumai, tokie kaip netolygi mygtukų stilistika, patvarkyti keli vertimai bei pastebėtos tekstų klaidos, angliškoje svetainės versijoje patvarkyta vieno mygtuko nuoroda, kuri vedė į lietuvišką svetainės puslapį, nors egzistuoja išversta jo versija.

Kadangi autorius turi nemažai patirties svetainių kūrimu, ypač naudojantis WordPress turinio valdymo sistema, tačiau neturėjo galimybės prisidėti prie bioinformatikos svetainės išvaizdos ir bendros informacijos tvarkymo, buvo atidžiai peržiūrėta svetainė ir joje pateikiamas turinys, nuspręsta pateikti kelias rekomendacijas tolimesniam svetainės tvarkymui ir jos tobulinimui:

1. Atidžiau peržvelgti pateikiamą turinį ir ištaisyti rašymo klaidas. Daugiausiai rasta neatidumo klaidų, kai raidės yra sukeistos dėl skubaus rašymo. Gražus, tvarkingai parašytas tekstas yra svarbi svetainės įvaizdžio dalis.
2. Angliškos svetainės versijos meniu neturi submeniu bioinformatikos skyrelių meniu punkte. Tai sumažina straipsnių naršymo efektyvumą. Autoriui neturėjo leidimo tvarkyti svetainės meniu, buvo suteikti tik įrašų ir puslapių kūrimo leidimai.
3. Pašalinti kelis meniu punktus, kurie veda į pagrindinio puslapio vietas. Meniu turi būti aiškūs, reikėtų praktikuoti vientisumą. Jeigu meniu elementai veda į kitus svetainės puslapius, nereikėtų palikti meniu elementų, kuriuos paspaudus slenkama iki tam tikros pagrindinio meniu dalies. Jei pasirenkama tokia praktika, visa informacija turėtų būti viename puslapyje.
4. Svetainėje sukurta antraštės (angl. *header*) informacinė dalis pateikiama abejomis kalbomis, kas nėra bloga praktika, tačiau pasirinktas pateikimo būdas atrodo perteklinis.
5. Sekcijoje apie bioinformatikos studijas mygtukai atrodo stilistiškai netvarkingi.
6. Sutvarkyti poraštę (angl. *footer*), išimant perteklines nuorodas ir paliekant nuorodas į pagrindinius puslapius. Šiuo metu poraštėje yra nurodomos nuorodos į kelis svetainėje pateikiamus straipsnius.
7. Sutvarkyti pagrindinio puslapio skiltį „Turinys“. Deja, bet „Uždaviniai“ neveda į jokių puslapių, tad yra galimybė arba sutvarkyti mokymosi aplinką, arba išimti šią nuorodą.
8. Patarimas iš WordPress entuziasto pusės – pakeisti svetainės temą į lankstesnę. Tokios mokomos temos labai apriboja svetainės galimybes.

Šių rekomendacijų pagrindas – svetainė turi atrodyti patraukliai svetainės lankytojams, tuo labiau pirmą kartą apsilankiusiems svetainėje, nes taip susidaroma pirminė nuomonė apie svetainę. Jei lankytoji svetainė nepatraukli – jis nenorės joje lankytis.

2. 3D modeliavimas

Norint sukurti modelį ar iliustraciją, kuo labiau atitinkantį realų objektą, reikia atlikti analizę. Jei jau egzistuoja kokia nors vaizduojamoji medžiaga, reikia ją peržvelgti, pažiūrėti į tai, kaip pateikiama medžiaga, tačiau vien esamais vaizdiniais negalima pasikliauti, ypač kai modeliavimo taikiny yra akimi nematomi subjektai. Yra svarbu įsigilinti į bandomo kurti objekto aprašymus, bandyti įsivaizduoti, kokių trūkumų gali turėti jau egzistuojanti vaizduojamoji medžiaga, arba kaip atvaizduoti tai, kas autoriaus nuomone yra svarbu.

2.1. Modeliavimo principai

Norint atvaizduoti medžiagą ne visada būtina kurti naujus 3D objektus, nes 3D modeliavimas nėra tik objektų kūrimas. Vieni projektai kuria pavienius objektus, o kiti – utilizuoja jau egzistuojančius objektus, apjungdami juos į vieną didesnę paveikslą. Taip pat egzistuoja 3D modeliavimo šaka, kuri orientuojasi būtent į detalų realaus pasaulio objektų atvaizdavimą juos analizuojant ir perkuriant rankiniu būdu arba naudojantis technologijomis, kurios gali tai padaryti be žmogaus įsikišimo. Toks procesas vadinasi 3D skenavimas.

2.2. Blender programinė įranga

Projektui vykdyti pasirinkta dirbtu su Blender 3D kūrimo rinkiniu, kuris yra licencijuotas kaip GNU GPL (angl. *General Public License*), todėl tai yra nemokama atviro kodo programa. [Com19a] Blender programa naudoja Python programinę kalbą savo API kūrimui, Python taip pat naudojama programos priedų, vartotojo sąsajos kūrimui, failų importavimui bei eksportavimui. Python kalba naudojama visuose vartotojų pasiekiamuose duomenyse ir funkcijose.[Com]

Darbas su Blender programa gali būti palyginamas su vizualiniu programavimu, kai įvairių elementų elgesio apibrėžimui ar nustatymui gali būti naudojami blokai (angl. *nodes*). Blender 3D modeliavimo rinkinys taip pat turi daug modifikatorių, kuriuos naudojant galima gauti labai įvairius rezultatus – nuo dalelių sistemų, viršūnių skaičiaus didinimo, iki objektų masyvų, medžiagų, fizikos kūrimo ir kt.

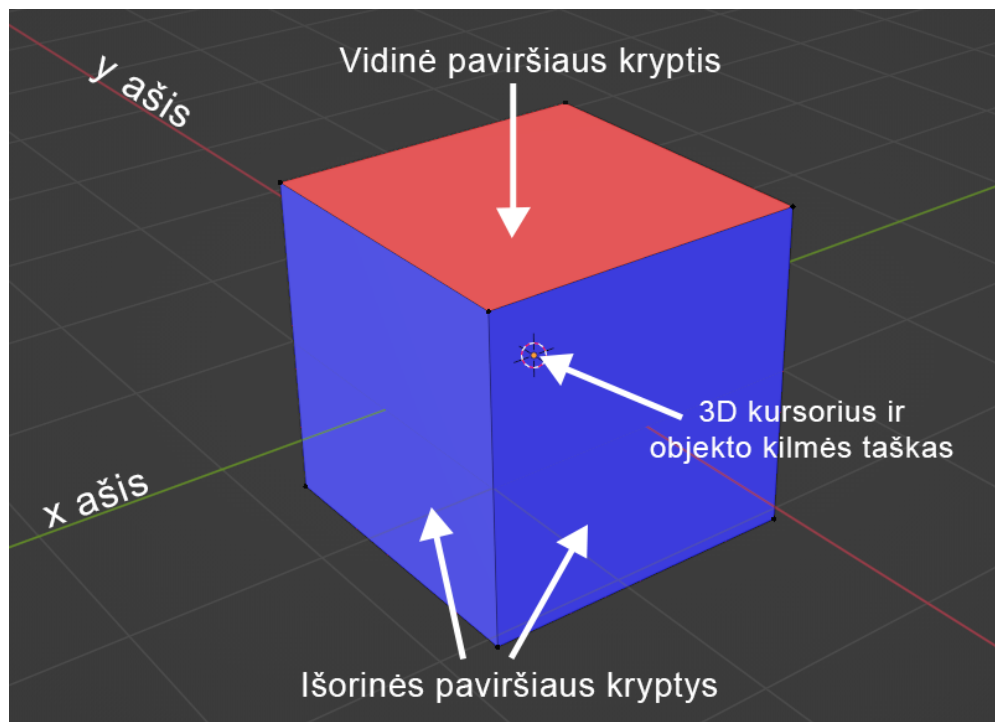
2.2.1. Pagrindiniai terminai

Blender programoje kartojasi keli terminai, kuriuos reikia žinoti norint dirbti su šia programine įranga.

Pagrindiniai ir patys svarbiausi yra trys terminai, apibrėžiantys visą 3D modeliavimo pagrindą. Naudojantis tik šiais trimis elementais jau galima sukurti 3D objektą:

1. Viršūnė (angl. *vertex*) – taškas, turintis savo poziciją 3D ašių sistemoje.
2. Briauna (angl. *edge*) – linija, sudaroma sujungiant dvi viršūnes.

3. Paviršius (angl. *face*) – plotas, sudaromas, kai trys ar daugiau viršūnių sujungiamos briaunomis, turinčiomis po vieną bendrą viršūnę. Pirma ir paskutinė viršūnės **turi** būti sujungtos tarpusavyje briauna.



1 pav. 3D kursorius, objekto kilmės taškas (oranžinis), ašys ir paviršių orientacijos spalvinis atvaizdavimas

Yra ir kitų dažnai sutinkamų terminų (žr. 1 pav.), kuriuos vertėtų žinoti naudojantis Blender programine įranga, keli jų išvardinami šiame sąrašė:

1. Ašis (angl. *axis*) – kiekvienas projektas turi trimatę ašių sistemą viršūnių pozicijų nustatymui.
2. 3D kursorius (angl. *3D cursor*) – taškas 3D ašių sistemoje, kuris gali būti nustatomas naudotojo. Pridedant naujus objektus, 3D kursorius naudojamas kaip objekto pirminė pozicija, tai yra objekto kilmės (angl. *origin*) taškas, kuris dažniausiai būna objekto geometrijos centre.
3. Normalė (angl. *normal*) – naudojama nustatyti paviršiaus arba viršūnių kryptį. Blender programoje normalės yra statmenos geometrijai. Ši informacija yra naudojama nustatyti išorinei ir vidinei paviršių kryptį (angl. *face-orientation*).
4. Tinklelis/tinklas (angl. *mesh*) – rinkinys viršūnių, briaunų ir paviršių, apibrėžiančių 3D objekto formą.
5. Modelis – tinklelis su tekstūrų ir kita svarbia informacija, apibrėžiančių kaip atvaizduoti modelio paviršius.

6. UV žemėlapis (angl. *UV map*) – naudojamas tekstūrų uždėjimui ant objekto paviršiaus.
7. Medžiagų nustatymai (angl. *material properties*) – medžiagų nustatymai leidžia sukurti medžiagą objektui, medžiagos kuriamos naudojantis įvairiais šešėliuotojais (angl. *shaders*) ir jų nustatymais.

2.2.2. Blender režimai

Blender programoje objektų kūrimui, modifikavimui ir objektų komponavimui yra skirtingi režimai, kurių kiekvienas turi savo funkcijas 3D modeliavime. Objekto režimas (angl. *Object mode*) yra įprastinis režimas Blender aplinkoje, šiame režime galima komponuoti objektus, keičiant jų poziciją, rotaciją. Šioje aplinkoje pridodami nauji objektai. Redagavimo režimas (angl. *Edit mode*) skirtas tam tikro pasirinkto objekto tvarkymui, šiame režime galima tvarkyti viršūnių pozicijas, pridėti ar pašalinti viršūnes, pridėti naujus objektus, kurie tampa dalimi redaguojamo objekto. Objekto režime pažymėjus kelis objektus ir perėjus į redagavimo režimą, galima redaguoti visų pasirinktų objektų modelius. Galima atskirti objekte esančius komponentus į atskirus objektus, pavyzdžiui, jei objekte yra du briaunomis nesujungti kubai, juos galima atskirti į du atskirus objektus. Skulptūriniame režime (angl. *sculpt mode*) galima keisti modelį panašiai, kaip skulptūrų gaminyje realiaame gyvenime, galima pridėti masę, nupjauti modelio dalį ir t.t. Šis režimas įprastai nesukuria naujų viršūnių, todėl norint sukurti detales reiktų padalinti modelio briaunas/paviršius ir padidinti viršūnių skaičių. Kiti režimai skirti priskirti svorį viršūnėms ir viršūnių grupėms, piešti tekstūras tiesiai ant objekto tinklelio ar tvarkyti dalelių sistemas.

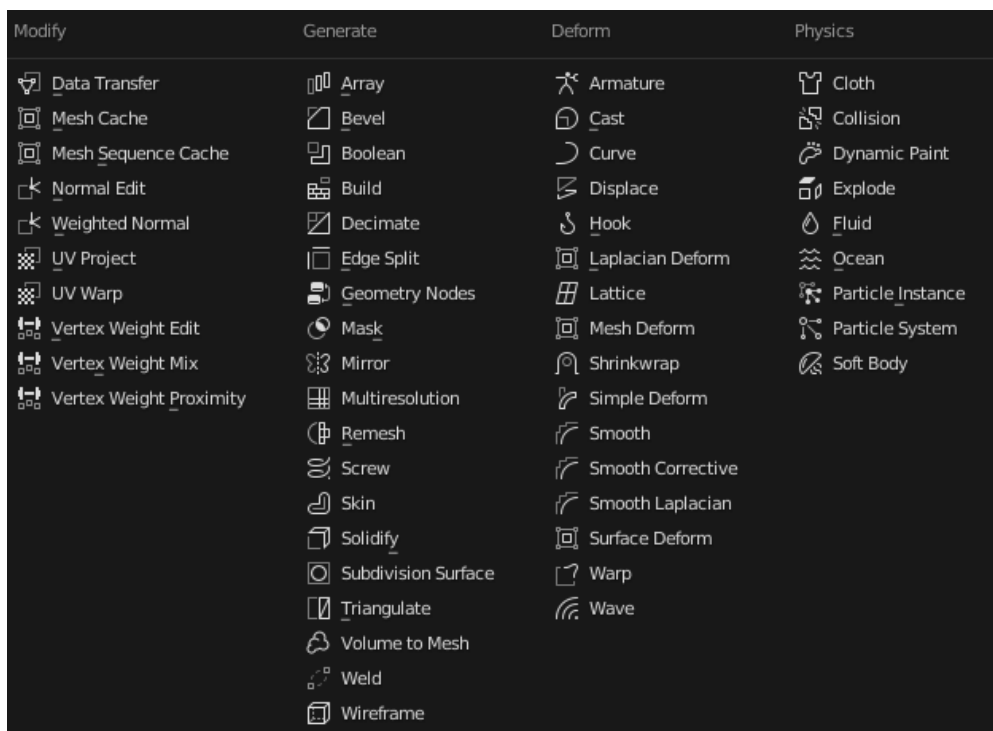
Be objektų režimų dar egzistuoja kitos darbinės erdvės, kurios skirtos įvairiems su modeliavimu, animacija bei galutiniu rezultatu susijusiems dalykams tvarkyti. Pavyzdžiui, animacijos erdvė leidžia kurti raktinius kadrus įvairiai objekto animacijai kurti. Komponavimo erdvė (angl. *compositing*) leidžia pakeisti galutinio vaizdo išvestį naudojantis blokais. Šioje erdvėje galima išvalyti galutinį rezultatą nuo triukšmo, daugiau apie tai tolimesniame poskyryje.

2.2.3. Modifikatoriai

Vieni svarbiausių įrankių Blender programoje yra modifikatoriai, skirti paveikti objektus ir taip keisti jų išvaizdą (žr. 2 pav.). Be papildinių (angl. *add-ons*), Blender programoje yra keturios grupės modifikatorių: modifikuoti (angl. *modify*), generuoti (angl. *generate*), deformuoti (angl. *Deform*) ir fizika (angl. *Physics*). Kiekvienas modifikatorius turi daugybę nustatymų, todėl modifikatoriai gali būti panaudojami labai daug skirtingų būdų.

Autoriaus labiausiai naudojami modifikatoriai, ypač autoriaus, yra generuojantys arba deformuojantys objektą. Iš generuojančių modifikatorių, dažniausiai reikalingi modifikatoriai sukuria objekto masyvą, darydami objekto kopijas pagal pasirinktus nustatymus (ašį, atstumą, susijungimą, kiekį), sukuria storį objektui (šis modifikatorius ypač vertingas kuriant objektus, kuriuos reikia pjauti per pusę, arba kur turi matytis tuščias vidus), padalija tinklelį į smulkesnį (tai sušvelnina modelio briaunas ir padidina viršūnių skaičių), padaro veidrodinę kopiją, susuka. Iš

deformuojančių modifikatorių labiausiai naudojamas kreivės modifikatorius, leidžiantis objektą pritaikyti prie projekte esančios kreivės. Labiausiai naudojama kombinacija yra masyvo ir kreivės modifikatoriai, kuriais naudojantis galima sukurti ilgus, pagal poreikį susukusius objektus, pavyzdžiui, virves, kurie gali būti lengvai redaguojami.



2 pav. Blender programoje prienami modifikatoriai

Modifikatorius nebūtina pritaikyti, jei atvaizduojama per Blender programą, tačiau norint eksportuoti modelį geriausia pritaikyti visus modifikatorius prieš šį procesą.

2.2.4. Svarbios funkcijos

Blender programinė įranga turi labai įvairių naudojimosi metodų, funkcijų. Kai kurie jų apibūdinami šiame skyrelyje.

Aukščiau buvo minėtas terminas normalė ir kaip jis padeda nustatyti paviršiaus orientaciją. Blender programoje galima pasirinkti patikrinti paviršių orientaciją, tada modelio paviršiai yra atitinkamai rodomi raudona arba mėlyna spalvomis. Raudona žymi vidinę paviršiaus orientaciją, o mėlyna – išorinę. Kartais geometrijos kūrimo metu paviršių normalės nustatomos neteisingai, atsitikus tokiai situacijai, Blender turi palankų įrankį normalių perskaičiavimui. Normalių apvertimui einama į modelio tvarkymo režimą, pažymėti galima arba visą objektą, arba viršūnes, sudarančias norimus apversti paviršius. Spaudžiant „Alt“ ir „N“ mygtukus vienu metu atidaromas meniu, kuriame galima pasirinkti apversti normales (labiausiai tinka tada, kai pasirenkami tik paviršiai, kuriuos reikia apversti), perskaičiuoti normales į išorę arba į vidų (šie variantai tinkami visais pasirinkimų atvejais). Geriausia normales patikrinti prieš projekto eksportavimą, arba kai projektas jau paruoštas atvaizdavimui naudojantis Blender vaizdavimo varikliais. Normalių

orientacija yra svarbi tekstūrų bei medžiagų vaizdavimui, nes šios savybės yra „uždedamos“ ant išorinės paviršiaus pusės. Jei paviršiaus vidinė pusė yra išorėje, tas paviršius objekto vaizdavimo rezultate dažniausiai būna nematomas, gaunama tuščia erdvė.

Sukūrus objektą skirtingoms jo dalims galima priskirti skirtingas medžiagas ir tekstūras. Objektui priskiriama bazinė spalva ar tekstūra, pažymimi objekto paviršiai, kuriems norima priskirti kitokią medžiagą, sukuriama nauja medžiaga tam pačiam objektui ir spaudžiama funkcija priskirti (angl. *assign*). Pažymėti paviršiai gauna tos medžiagos informaciją ir nebesinaudoja pagrindinės medžiagos informacija. Blender taip pat turi atskirą darbo erdvę šešėliavimo nustatymams, kuriame gali būti naudojami blokai tekstūrų bei medžiagų gamybai. Kai kuriais atvejais užtenka objekto paviršiui priskirti tik tam tikrą šešėliuoklę, o kartais reikia priskirti iškilimų žemėlapius (angl. *bump maps*) ir kitus blokus, galima atlikti daug testavimų, kol gaunama norima objekto medžiagos išvaizda. Įdomu tai, kad iškilimų žemėlapiai gali suteikti labai daug realistiškumo ir netgi sukurti iliuziją ar net deformuoti objekto tinklę taip, kad medžiagos nelygumai ir tekstūros atsispindėtų galutiniame rezultate. Objekto nelygumams priskirti galima naudojami įvairius Blender programoje egzistuojančius blokus.

UV žemėlapis yra vertinga Blender funkcija, leidžianti uždėti foto-realistines tekstūras ant objekto paviršiaus. Norint tinkamai uždėti tekstūrą, objektas turi būti išardomas į 2D formatą, tai galima įsivaizduoti kaip popierinio kubo gaminimą, tik atvirkščiai. Blender sistemoje esantys paprastieji objektai (kubai, rutuliai, cilindrai ir kt.) jau turi išardytus UV žemėlapius ir yra paruošti tekstūrų uždėjimui, bet statant objektus nuo nulio, juos dar reikia išardyti rankiniu būdu, tam vartotojas pažymi tam tikras briaunas aštriomis (angl. *mark sharp*) ir pasirenka išvyniojimą (angl. *unwrap*). Objektas išvyniojamas pagal pažymėtas briaunas, todėl svarbu atkreipti dėmesį, kuriose vietose būtina pjauti objektą. Geriausiai pjaunamas briaunas nustatyti mažiau matomose vietose, kad uždėdam tekstūras maži tekstūrų neatitikimai, jei tokių yra, nesimatytų galutiniame produkte.

Programoje labai daug objektų animacijų ir judesio nustatymų, objektų fizikos nustatymų, įvairių atvaizdavimo nustatymų, leidžiančių pasirinkti norimą vaizdavimo variklį ir kaip bus atvaizduojami objektai, kameros nustatymų, kurie reikalingi norint pakeisti galutinio rezultato išvaizdą tuo atveju, kai kompoziciją ketinama atvaizduoti Blender programa.

2.2.5. Cycles vaizdavimo variklis

Pirmasis autoriaus bandymas buvo rinkti patirtį kuriant 3D modelius, kurių medžiagos vaizduojamos Blender programos Cycles atvaizdavimo varikliu. Cycles yra spindulių sekimo (angl. *ray-tracing*) principu veikiantis vaizdavimo variklis. Priklausomai nuo norimo rezultato, gali būti sukuriama net tokie atvaizdai, kurie beveik neatskiriami nuo padarytos nuotraukos.

Deja, tokiu principu sukurtas objektas negali būti pateikiamas svetainėje, nebent medžiaga būtų naudojama kaip statinis paveikslėlis ar trumpas filmukas. To priežastis yra didelis techninės įrangos resursų naudojimas atvaizduojant 3D modelius. Priklausomai nuo kompiuterinės įrangos resursų bei 3D modelio vektorių skaičiaus, tekstūrų nustatymo, vieno paveiksluko atvaizdavimas gali užtrukti nuo kelių sekundžių iki 30 minučių ar ilgiau. Jei norima sukurti 40 sekundžių ilgumo,

24 kadrų per sekundę filmuką, kai vieną kadrą atvaizduoti trunka 30 sekundžių, tai užtruktų $40 \times 24 \times 30 = 28800$ sekundžių, kas atitinka 480 minučių arba 8 valandas.

2.2.5.1. Cycles projektas

Blender veikimo suvokimui autorius pasirinko sukurti pusryčių stalą. Atvaizdas sukurtas naudojantis Cycles atvaizdavimo varikliu. Šį kadrą (žr. 3 pav.) atvaizduoti truko 1 minutę ir 46 sekundes, atvaizdavimui naudotasi GPU³, kuris dirba efektyviau nei CPU⁴. GPU atvaizdavimas yra maždaug nuo 50 iki 100 kartų greitesnis nei procesoriaus atvaizdavimas, tačiau yra manoma, kad tradicinis CPU atvaizdavimas yra aukštesnės kokybės, todėl animacijos studijos dažnai naudoja tradicinį atvaizdavimą mikroprocesoriais. [Lee19]



3 pav. Atvaizdavimas su Cycles, su triukšmo išvalymu



4 pav. Atvaizdavimas su Cycles, be triukšmo išvalymo

Dažniausiai atvaizduojamas kadras turi triukšmo (žr. 4 pav.), tačiau autorius savo darbui pasinaudojo *Denoise* mazgu, kuris išvalo triukšmą iš gautos nuotraukos ir taip vaizdas atrodo tvarkingiau. Mazgas naudojamas komponavimo (angl. *compositing*) lange, kur galima dėti įvairius baigiamojo dorojimo (angl. *post-processing*) mazgus, pakeičiančius galutinį atvaizdavimo rezultatą. *Denoise* mazgas veikia naudodamas Intel® Open Image Denoise technologiją. Tai yra atviro kodo biblioteka, naudojama išfiltruoti triukšmą iš spindulių sekimo principu atvaizduotų kadrų. Viena iš šios technologijos naudotojų yra *Walt Disney Animation Studios*. Šioje bibliotekoje esantys triukšmo išvalymo filtrai remiasi gilaus mokymosi (angl. *deep learning*) principais. [Cor]

³grafikos apdorojimo blokas

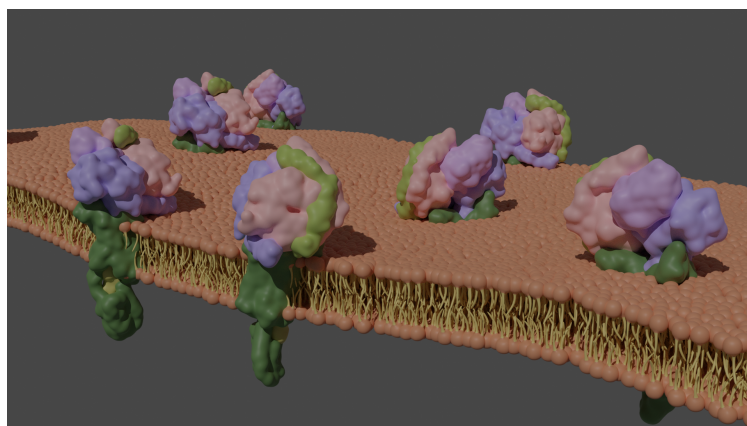
⁴kompiuterio apdorojimo blokas

2.2.6. Procedūrinis modeliavimas

Objektai gali būti kuriami ir procedūriškai, kai modelio kūrimui yra naudojami algoritmai. Algoritmai naudojami nustatyti tam tikras taisykles, kaip turi būti kuriamas 3D modelis. Procedūrinė kompiuterija taip pat naudojama tekstūroms, dažnai ši technologija naudojama augalų, pastatų kūrimui, nes naudojantis algoritmais galima išvengti individualių objektų kūrimo ir sukurti skirtingus vieno tipo objektus pagal nustatytas taisykles. [KBW]

2.2.6.1. Procedūrinio modeliavimo pavyzdys

Autorius išbandė procedūrinį modeliavimą, buvo kuriama plazminė membrana. Modelyje sukurtas fosfolipidų dvisluoksnis ir iš ChimeraX [FHP⁺18] molekulinio vizualizavimo programos buvo gautas vieno iš baltymų 3D modelis. ChimeraX yra puiki sistema molekulių modelių analizei, pateikianti 3D modelį, trumpą informaciją apie molekulę, o pati programa yra labai lanksti. Tai – vertingas įrankis bioinformatikos moksle, kurį vertėtų pristatyti bioinformatikos studentams. ChimeraX leidžia atidaryti PDB⁵ failus tiesiai iš komandinės eilutės. Autorius pasinaudojo įrankiu gauti žmogaus sekretino receptorių Gs komplekso (<https://www.rcsb.org/structure/6wzg>), vieno iš plazminės membranos baltymų, modelį, kuris buvo naudotas membranos atvaizdavime, įstatant membranos baltymą į fosfolipidų dvisluoksnį. Buvo bandoma baltymo kompleksą į membraną įstatyti taip, kaip jis yra išsidėstęs natūraliai membranoje. Išsidėstymas gautas iš PDB duomenų bazės esančios informacijos (<https://www.rcsb.org/3d-view/6WZG/1?preset=membrane>).



5 pav. Plazminės membranos atvaizdavimas su Cycles

Šis projektas buvo įdomus procedūrinio 3D objektų kūrimo bandymas (žr. 5 pav.).

⁵PDB – Protein Data Bank, duomenų bazė, kurioje laikoma informacija apie baltymus

3. 3D modeliai internetinėje erdvėje

Darbo metu buvo išvystyta idėja praturtinti bioinformatikos svetainėje pateikiamus straipsnius interaktyvia informacija. Darbo tikslas – kurti įvairaus lygio straipsnius, kuriuose aprašomos temos būtų papildytos ne tik paprastomis iliustracijomis ar nuotraukomis, bet ir projektais, demonstruojančiais su tema susijusius 3D modelius. Tai reikalauja ne tik kuriamų struktūrų analizės, temos suvokimo bei nuoseklaus informacijos pateikimo, bet ir 3D modeliavimo bei programavimo žinių.

Sukurti 3D modelį nėra vienintelis svarbus šio projekto žingsnis. Sukurtus modelius pateikti svetainėje buvo svarstomi keli metodai. Turinio valdymo sistemos neturi paprasto sprendimo, kaip integruoti 3D modelius į svetainę naudojantis įskiepiais, todėl autorius testavo kitus metodus, kurie galėtų rodyti sukurtus 3D modelių projektus.

3.1. GitHub

Tam, kad būtų galima naudoti sukurtus WebGL projektus internetinėje erdvėje, buvo pasinaudota GitHub hostingu (angl. *hosting*), skirtu programinės įrangos kūrimui bei versijų kontrolei. GitHub suteikia galimybę hostinti puslapius tiesiai iš repozitorijos. Šia funkcija galima pasinaudoti projektų demonstravimui.

3.2. Objektų vizualizavimas su WebGL

Geriausias sprendimas buvo naudotis WebGL technologijomis, kurias dažniausiai palaiko daugelis įrenginių.

WebGL reikalauja daug programavimo, nuo paprasčiausių dalykų kaip šešėliuoklė (angl. *shader*). Tam, kad dirbti su WebGL, reikia suprasti kelias OpenGL API, 2D ir 3D vektorių atvaizdavimo programas, kuria remiasi WebGL API, sritis. Tai apima, bet neapsiriboja anksčiau minėta šešėliuokle ir jos atskira programavimo kalba, matricų skaičiavimu transformacijoms nustatyti, ir viršūnių (angl. *vertex*) buferiais, skirtais laikyti informacijai apie viršūnių pozicijas, normaliąsias vertes, spalvas ir tekstūras. [Con11]

Tačiau nėra būtina viską žinoti puikiai, kad būtų galima naudotis WebGL API. Darbą su WebGL palengvina bibliotekos, sistemos, varikliai bei įrankių rinkiniai, integruojantys WebGL.

3.2.1. Three.js

Three.js yra viena populiariausių JavaScript kalbą parašytų bibliotekų, naudojančių WebGL API vaizdo kūrimui (angl. *rendering*). Bandytas įkelti 3D modelius į svetainę – HTML kalba parašytas skriptas, pritaikantis įvairių JavaScript skriptų, esančių Three.js bibliotekoje, elementus 3D modelio atvaizdavimui svetainėje. Šis variantas atvaizdavo kurtus modelius, reikalavo daug atidumo bei nuoseklumo atsirenkant reikiamus elementus.

Three.js biblioteka yra labai universali ir turi daug panaudojamų elementų, paruoštų naudojimui. Svarbiausia integruoti projektui reikiamus JavaScript skriptus. Kurtam projektui reikalingų WebGL bibliotekos skriptų integravimo kodas, naudotas straipsnio įkėlime:

```
1 <script src="https://cdn.jsdelivr.net/npm/three@0.128.0/build/three.min.js">_
  ↪ </script>
2 <script src="https://cdn.jsdelivr.net/npm/three@0.128.0/examples/js/controls_
  ↪ /OrbitControls.js"></script>
3 <script src="https://cdn.jsdelivr.net/npm/three@0.128.0/examples/js/environm_
  ↪ ents/RoomEnvironment.js"></script>
4 <script src="https://cdn.jsdelivr.net/npm/three@0.128.0/examples/js/loaders/_
  ↪ GLTFLoader.js"></script>
5 <script src="https://cdn.jsdelivr.net/npm/three@0.128.0/examples/js/loaders/_
  ↪ RGBELoader.js"></script>
```

WordPress turinio valdymo sistemoje rašomuose straipsniuose gali būti įterpiami HTML kalba parašytas kodas, todėl yra pakankamai paprasta įterpti sukurto 3D modelio kodą į svetainės straipsnį.

Parašytas Three.js skriptas primena kietąjį kodavimą⁶, be to, tokiu būdu pateikiant objektą reikia daryti daug tokio pačio kodo kopijavimo ir įklijavimo, kas nėra efektyvu, todėl nuspręsta neapsistoti ties pasiektu rezultatu, išbandyti kitą alternatyvą, aprašomą sekančiame poskyryje. Three.js pavyzdžio nuorodos yra priede nr. 3.

3.2.2. Unity WebGL

Buvo nuspręsta išbandyti Unity žaidimų variklį WebGL aplikacijų kūrimui. Unity turi labai palankius įrankius programų kūrimui. Unity – įvairiaplatformis⁷ žaidimų kūrimo įrankis, pritaikytas kurti tiek kompiuteriams, tiek mobiliems telefonams, žaidimų konsolėms bei virtualios realybės platformoms. Tai pasaulyje lyderiaujantis žaidimų variklis. 2018 metais Unity Technologies išplėtė savo taikymo sritį, įtraukdami savo technologijas į kitas pramonės šakas, tokias kaip automobilių, kino, architektūros, inžinerijos bei statybos industrijas. [Com19b]

Dirbant su Unity buvo išsiaiškinti keli trūkumai, vienas jų – Unity WebGL gali neveikti su visomis platformomis. Tam naršyklė turi atitikti kelis standartus:

1. Naršyklė palaiko WebGL 1 arba WebGL 2.
2. Naršyklė atitinka HTML 5 standartus.
3. Naršyklė yra 64 bitų ir palaiko WebAssembly.

⁶ angl. *Hard-coding* – programinės įrangos kūrimo praktika, kai duomenys įterpiami tiesiai į programos šaltinio kodą, o ne generuojami ar gaunami iš išorinio šaltinio kompiliavimo metu

⁷ angl. *Cross-platform* – skirtas veikti per kelias kompiuterijos platformas

Unity dokumentacijoje nurodyta, kad mobilieji įrenginiai nepalaiko Unity WebGL. Taip yra dėl to, nes dabartiniai mobilieji įrenginiai dažnai yra nepakankamai galingi ir neturi pakankamai atminties palaikyti Unity WebGL turinį. [Tec22] Įdomu tai, kad autoriaus kurti projektai veikia naudojamame mobiliajame įrenginyje, galbūt dėl to, nes projektams nereikia daug išteklių atvaizduojant projekte esamą turinį.

3.2.2.1. Unity WebGL projektų kūrimas

Naudotis Unity redaktoriumi nėra sudėtinga, programa yra pakankamai intuityvi naudojimui. Projektų kūrimui reikia turėti C# programavimo kalbos pagrindus. WebGL programų interaktyvumui įgyvendinti C# kalba kuriami skriptai, aprašantys žaidimo objektų (sukurtų 3D modelių, kameros, vartotojo sąsajos elementų) elgesį. Kūrimui taip pat svarbu žinoti, kaip kreiptis į skirtingus elementus, apibrėžtus Unity sistemoje, elementai gali turėti daug skirtingų laukų, funkcijų.

3.2.2.2. Unity projektų naudojimas svetainėje

Naudojantis Unity WebGL modelio kėlimas į svetainę yra lengvesnis nei naudojantis Three.js biblioteka, nereikia daryti pakeitimų pačiame svetainės straipsnyje. Įkėlus projektą ir jį įterpus į straipsnį naudojantis HTML programavimo kalbos `<iframe>` žyma, projektas gali būti toliau tvarkomas lokaliai, o norint įkelti pakeitimus tiesiog pakeičiami į GitHub įkelti failai. Svarbiausia, kad projektas liktų toje pačioje direktorijoje.

Pirmasis parašytas kodas veikė gerai, tačiau nebuvo jautrus ekrano `<iframe>` aukščio (angl. *height*) atributo atžvilgiu. Įsijungus straipsnį per siauresnį įrenginį (pavyzdžiui telefoną), apačioje lieka daug juodos erdvės, nors pats projektas puikiai telpa pagal plotį. Kodo pavyzdys:

```
1 <iframe src="https://skimmeds.github.io/unity-webgl/eukariotine-lastele"  
2 style="border:0; width:100%; height:750px" type="text/html"></iframe>
```

WordPress HTML laukas leidžia apibrėžti stilius ir parašyti daugiau nei vieną eilutę kodo, todėl projekto įterpimui `<iframe>` elementas buvo talpinamas į `<div>` elemento vidų. `<div>` elemento plotis nustatytas 100%. Kadangi visi projektai buvo 16:10 proporcijos, išskyrus pirmąjį, reikiamo aukščio proporcija apskaičiuojama aukštį dalinant iš pločio ir padauginus iš šimto, kad gautume koks procentas pločio atitinka ilgį: $10 \div 16 \times 100 = 62.5$. Skaičius apvalinamas į didžiąją pusę tam, kad nebūtų besidvejinančio vaizdo, kas buvo pastebėta įkėlimo metu, jei `<iframe>` langas yra bent kiek per mažas, šiuo atveju pasirinkta, kad aukštis būtų 64% pločio. Šis skaičius naudojamas padaryti reikalingo ilgio erdvę, kuri keičiasi su pločiu, išlaikydama nustatytą proporciją. `<iframe>` absoliuti pozicija reiškia, kad elemento konteineris yra visa tėvinio elemento erdvė, apibrėžta tėvinio elemento *padding*⁸ krašto, o *padding* zona dėl absoliučios pozicijos įeina į `<iframe>` konteinerį. Todėl galima nenurodyti pačio konteinerio aukščio. Kodo pavyzdys:

⁸ angl. *padding* – zona tarp elemento turinio ir rėmelio.

```
1 <meta name="viewport" content="width=device-width, initial-scale=1" />
2 <style>
3   .container {
4     position: relative;
5     width: 100%;
6     overflow: hidden;
7     padding-top: 64%; /* suapvalinta 16:10 proporcija */
8   }
9   .responsive-iframe {
10    position: absolute;
11    top: 0;
12    left: 0;
13    bottom: 0;
14    right: 0;
15    width: 100%;
16    height: 100%;
17    border: none;
18  }
19 </style>
20
21 <div class="container">
22   <iframe
23     class="responsive-iframe"
24     src="https://skimmeds.github.io/unity-webgl/eukariotine-lastele"
25   ></iframe>
26 </div>
```

4. Straipsniai

Baigiamajam darbui pasirinkta parašyti kelis straipsnius mikrobiologijos bei molekulinės biologijos tema, prie jų pridedant WebGL projektus, susijusius su straipsniais, vizualiam ir interaktyviam informacijos pateikimui. Modeliams kurti buvo pasirinktos skirtingos metodikos, be to, priklausomai nuo modelio funkcijos priderinama ir pasirinkta skirtinga projekto stilistika, todėl modeliai nėra koherentiniai stiliaus atžvilgiu. Tokia metodika pasirinkta norint išbandyti skirtingus 3D modelių pateikimus, be to, modelio pateikimas pritaikomas prie jo paskirties.

Pagrindinė darbo mintis yra suteikti kitokį būdą pateikti informaciją, pridėti daugiau interaktyvumo informacijai, papildyti straipsnių informacinę vertę. Bandymai parodo, kad naudojantis WebGL ir 3D modeliavimu galima praturtinti informaciją įvairiais būdais, vieni gali būti labiau interaktyvūs nei kiti, nes vieni modeliai kurti tik pademonstruoti struktūras, o kiti – pateikti svarbią informaciją. Kai kuriais atvejais matyti 3D modelį gali būti vertingiau nei paprastą nupieštą iliustraciją ar nubraižytą schemą.

Kaip minėta, straipsniams pasirinkta mikrobiologijos ir molekulinės biologijos temos. Prie straipsnių tematikos prieita darbo metu, tam, kad kuriami straipsniai turėtų prasmę ir pateiktų bioinformatikai aktualią informaciją. Temos vystymas pradėtas nuo bakterijos modelio (žr. 4.1 poskyrį). Bioinformatika yra svarbi mikrobiologijos mokslui, todėl parašyti straipsniai yra puikus įvadas į temą, be to, mikrobiologijoje yra labai įdomių subjektų 3D modeliavimui. Kadangi modeliavimas, modelių pateikimas ir straipsnių rašymas yra daug laiko užimantis procesas, yra daug erdvės papildyti esamą turinį rašant naujus straipsnius šia tema. Vienastraipsnių yra orientuojamasi šiek tiek į molekulinės biologijos, gal net šiek tiek biochemijos, temą.

Straipsniai pateikiami tiek anglų, tiek lietuvių kalbomis. Surinkta informacija iš skirtingų šaltinių buvo apjungama, išrinkta tai, kas autoriui atrodė aktualu. Straipsniai skirti vyresnių klasių gimnazistams ir vyresniems biologija besidomintiems žmonėms, nes kai kuri informacija yra pateikiama kiek sudėtingesnė.

4.1. Prokariotinė ląstelė

Pirmasis projektas – bandymas išpildyti naują idėją norint praturtinti straipsnius bioinformatikos svetainėje. Sukurtas 3D modelis demonstruoja bakterijos organeles. Bakterijos struktūra yra pakankamai paprasta, todėl ir buvo pasirinkta kaip pirmasis bandymas 3D modeliavime.

Parašytas HTML kodas, naudojantis Three.js bibliotekos atvaizdavimo skriptais. Atvaizduojant sukurtas interaktyvumas – kameros judinimas apie objektą (kitai vadinama orbitinė kamera), taip pat kamerą galima priartinti ir atitolinti nuo objekto, kas leidžia apžvelgti modelį iš visų pusių. Skriptas parašytas taip, kad pakeitus naršyklės lango dydį keistųsi HTML <canvas> elemento dydis, taip prisitaikoma prie lango dydžio.

Bakterijos (Bacteria) – prokariotai, bakterijų (Bacteria) domeno organizmų karalystė. Paprastai būna kelių mikrometrų ilgio, forma – įvairi: rutulinė, lazdelinė ar spiralinė. Bakterijos yra viena iš pirmųjų gyvybės formų, kuri atsirado Žemėje, ir sutinkama beveik visose buveinėse – dirvožemyje, vandenyje, karštoseiose versmėse, radioaktyviosiose atliekose ir Žemės plutos gilumose. Bakterijos taip pat gyvena simbiozėje ir parazituoja augaluose bei gyvūnuose. Dauguma bakterijų dar nėra aprašytos ir tik maždaug apie pusę bakterijų rūšių gali būti kultūrinu būdu auginamos laboratorijose. Bakterijas tiria mikrobiologijos šaka bakteriologija.



6 pav. Pirmasis WebGL projektas, kurtas naudojantis Three.js biblioteka

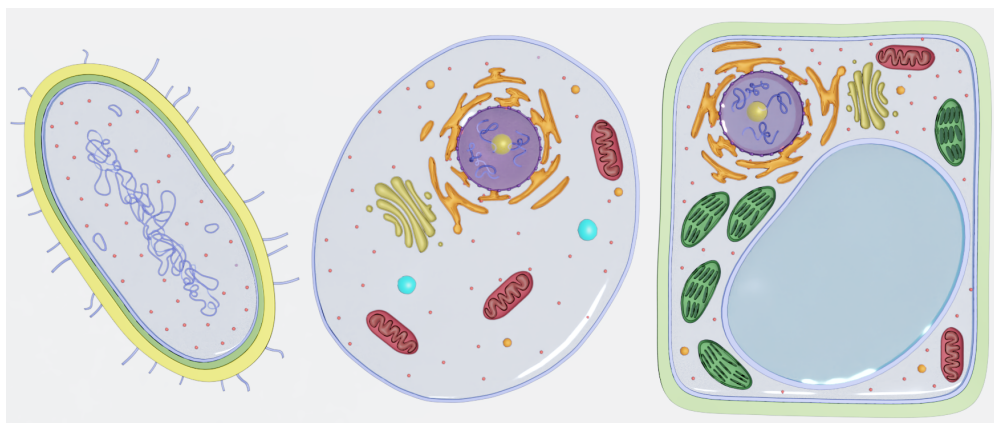
Modelis nebuvo publikuojamas bioinformatikos svetainėje, tačiau rezultatas publikuotas šiam darbui sukurtame projekto puslapyje adresu <https://skimmeds.github.io/unity-webgl/THREE-prokaryotic/>. Buvo bandoma parodyti išorinių ląstelės sluoksnių permatomumą, vaizduojama ląstelės žiedinė chromosoma (žiedinė DNR molekulė), ribosomos. Taip pat bandyta atvaizduoti žiuželio jungtį su bakterijos membrana, sienele ir kapsule. (žr. 6 pav.)

4.2. Mikrobiologijos įvadas

Įvadinis straipsnis. Šis straipsnis neturi 3D modelių, tačiau tai yra svarbus trumpas straipsnis apie mikrobiologijos ir bioinformatikos ryšį. Skirtas temos pristatymui. Straipsnyje bioinformatika apibūdinama kaip mokslas, analizuojantis didelio masto biologinius duomenis, tokius kaip DNR ir aminorūgščių sekos, ir kaip bioinformatikos specialistai gali panaudoti sekų informaciją genų anotacijai. Mikrobiologija trumpai apibūdinama kaip mokslas apie bakterijas, archėjas, virusus, grybus, pirmuonis bei dumblius. Iš to buvo daroma išvada, kad bioinformatikos mokslas svarbus mikrobiologijai dėl galimybės analizuoti sekas ir sieti tas sekas su mikroorganizmų funkcijomis, kas gali padėti geriau suprasti mikroorganizmus ir panaudoti informaciją vaistų, biokuro gamyboje, taip pat užterštumo valymui. [Bis14]

4.3. Ląstelių tipai ir organelės

Skaitytojas supažindinamas su egzistuojančiais ląstelių tipais – prokariotinėmis ir eukariotinėmis ląstelėmis. Straipsnyje aprašoma pagrindinė informacija apie ląstelių tipus, trumpai aprašoma informacija apie abiejuose tipuose randamas organelės ir jų funkcijas. Apibrėžiami ląstelės, organelės terminai, pabrėžiamas pagrindinis skirtumas tarp prokariotinės ir eukariotinės ląstelių – prokariotinės ląstelės neturi branduolio ir kitų membraninių organelių. Kai kurios ląstelės dalys randamos abiejuose ląstelių tipuose. Deoksiribonukleorūgštis (DNR), kurioje kaupiama informacija apie baltymų sintezę. Citoplazma, kuri užpildo ląstelės vidų ir kurioje vyksta didžioji dalis ląstelės procesų. Plazminė membrana, kuri suteikia ląstelei formą ir apsaugą, atlieka kai kurių medžiagų pernešimo funkciją bei turi baltymų, sąveikaujančių su kitomis ląstelėmis. Ribosomos – organelė, sudaryta iš RNR ir baltymų, baltymų sintezės vieta.

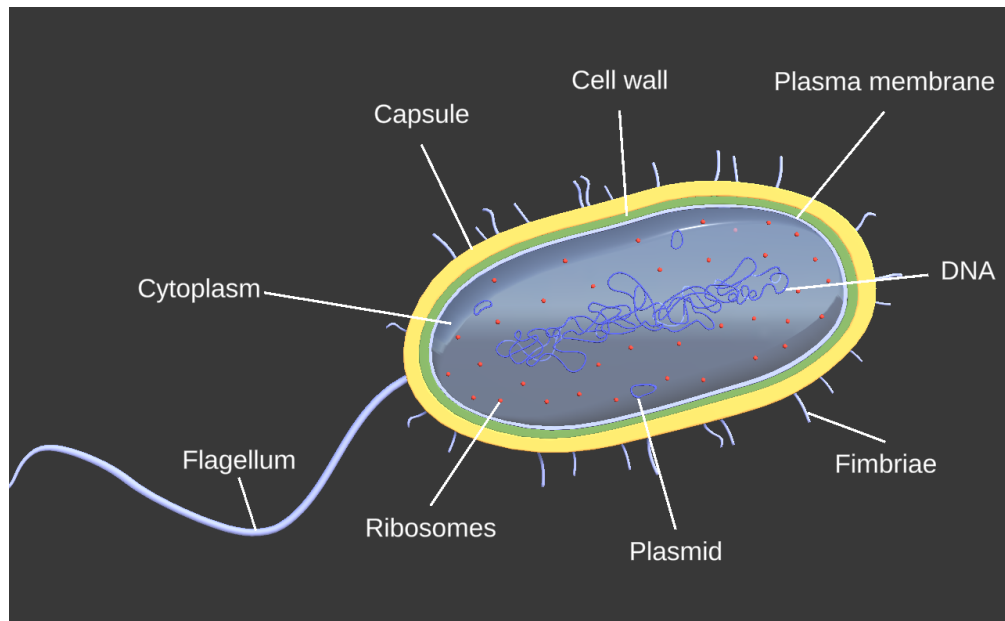


7 pav. Straipsnyje pateikiami trys ląstelių tipai. Blender Cycles atvaizdavimas.

Buvo sukurti trys 3D modeliai, kurie demonstruoja ląstelių organelės. Modeliai buvo sukurti kitiems straipsniams, kurie aprašomi 4.4 ir 4.6 poskyriuose. Kadangi modeliai buvo vertinga iliustracija temoje apie ląstelių tipus, sukurti 3D modeliai sukomponuoti taip, kad naudojantis Cycles atvaizdavimo varikliu būtų sukurta iliustracija (žr. 7 pav.). Prokariotinė, gyvūninė ir augalinė ląstelės buvo sukomponuotos viena šalia kitos, nustatytas palankus apšvietimas. Pasinaudota anksčiau minėtu Denoise filtru vaizdo išvalymui, kad vaizdas atrodytų aiškiau. Taip pat pasinaudota Freestyle nustatymais, kurie veikia kaip papildomas vaizdavimo variklis, veikiantis po Cycles variklio atvaizdavimo. Blender Freestyle nupiešia linijas ties tam tikromis modelių briaunomis, taip paryškindamas atskiras organelės, vaizduojamas modeliuose. Linija nustatyta plona, nes tai tik mažas modelio briaunų paryškimas aiškesnei iliustracijai.

4.4. Prokariotai. Bakterijos ir archėjės

Skaitytojas supažindinamas su dvejais prokariotų domenais – eubakterijomis ir archėjomis. Plačiau apibūdinamos organelės, būdingos prokariotinėms ląstelėms.



8 pav. Straipsnyje pateikto prokariotinės ląstelės WebGL projekto momentinė ekrano kopija

Bakterijos struktūros atvaizdavimas nebuvo numestas į šoną, buvo sukurtas naujas ir visiškai kitaip atrodantis bakterijos struktūros modelis, nei pirmajame bandyme. Pasirinkta nekurti apvalios ląstelės, o sukurti 3D iliustraciją. Ląstelėje esanti DNR kurta naudojantis Bežjė kreive, kuri buvo nupiešta pagal norimą sukurti DNR molekulės išvaizdą, sukūrus cilindrinį objektą, jam pritaikyti masyvo ir kreivės sekimo modifikatoriai, kad sukurti DNR žiedinę chromosomą. Fimbrijos kuriamos naudojantis Blender dalelių sistema. Ribosomos citoplazmoje taip pat išdėstytos naudojantis dalelių sistema, dalelės dedamos ant citoplazmos modelio, o tam, kad ribosomos būtų tik priekyje, buvo naudota *density*, kitaip žinomo kaip tankumo, atributas, kuriam priskirtos viršūnių svorio dažymo (angl. *weight paint*) vertės. Svorio dažymas naudojamas nustatyti dalelių emisiją, plaukų tankį, kai kurių modifikatorių elgesį.

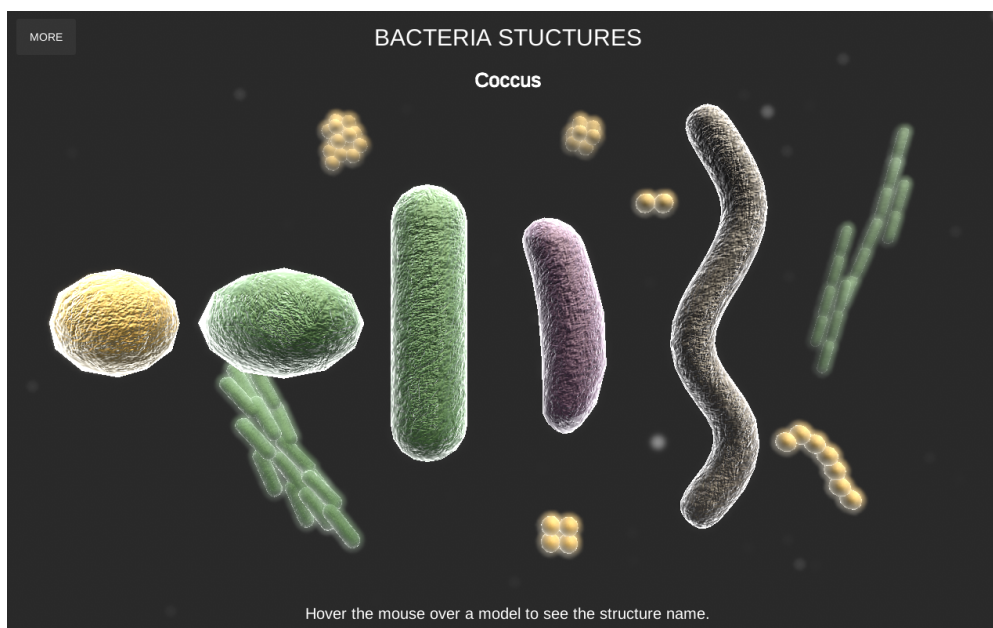
Naudojantis Unity, sukurtas WebGL projektas, kuriame rodomi prokariotinės ląstelės organelių pavadinimai (žr. 8 pav.). Projektui buvo parašyti keturi skriptai C# kalba. Vienas iš skriptų aprašo kameros elgesį, kad judinant pelytės ratuką, būtų atitinkamai priartinamas ir nutolinamas vaizdas. Kad nebūtų per daug priartinama ar nutolinama, nustatyti limitai. Dar vienas skriptas lemia tai, kad laikant pelytės kairįjį mygtuką ir judinant pelytę, objektas yra judinamas. Kadangi modelis yra pritaikytas žiūrėjimui iš vienos pusės, nustatyti rotacijos limitai naudojant *Clamp* funkciją. *Clamp* funkcija leidžia nustatyti limitą *float* tipo kintamiesiems. Judinant modelį, tekstas visada žiūri tiesiai į kamerą, tai yra dar vienas apibrėžtas elgesys. Paskutinis svarbus skriptas leidžia sujungti organelę su jos pavadinimu, nurodant pozicijas, kuriose prasideda ir baigiasi linija.

Buvo atliktas bandymas lokalizuoti projektus, kad būtų galima pakeisti kalbą nekuriant projektų atskiromis kalbomis, tačiau projektas metė klaidą. Lokalizacijos idėja – vėliavėlės paspaudimu pakeičiama projekto lokalizacija ir pakeičiama tekstų kalba. Bandymą galima matyti adresu <https://skimmeds.github.io/unity-webgl/prokaryote-localization/>.

4.5. Prokariotai. Bakterijų klasifikacija

Skaitytojui pristatoma informacija apie gramteigiamas ir gramneigiamas bakterijas, pristatomas Gram dažymo testas ir rezultatų interpretavimo pavyzdys. Pateikiama autoriaus kurta iliustracija, kurioje matomas gramteigiamos ir gramneigiamos sienelių sluoksnių išsidėstymo skirtumai.

Svarbiausia straipsnio dalis – bakterijų klasifikacija pagal jų formą bei ląstelių išsidėstymą ir grupavimąsi bakterijų kolonijose. Yra kelios pagrindinės bakterijų formos, tokios kaip kokas (apvali), bacila (lazdelinė), vibrio (kablelio formos), spirilė (spiralės formos). Klasifikavimas pagal bakterijų išsidėstymą: diplo (kai ląstelės sukibusios poromis), teradai (po keturias ląsteles sukibusios bakterijos), sarcinos (sukibusios po aštuonias bakterijų ląsteles), strepto (bakterijos sukibusios grandine), stafilokokai (vynuogę primenantis kokams būdingas grupavimasis), mikobakterijos (sulipusios lazdelinės bakterijos, struktūra kiek primena stafilokoką) ir kt. Informacija apie bakterijų klasifikaciją buvo parengta remiantis *National Institute of Open Schooling* svetainėje pateiktu mikrobiologijos kursu. [A1-16]



9 pav. Straipsnyje pateikto bakterijų morfologijos WebGL projekto momentinė ekrano kopija

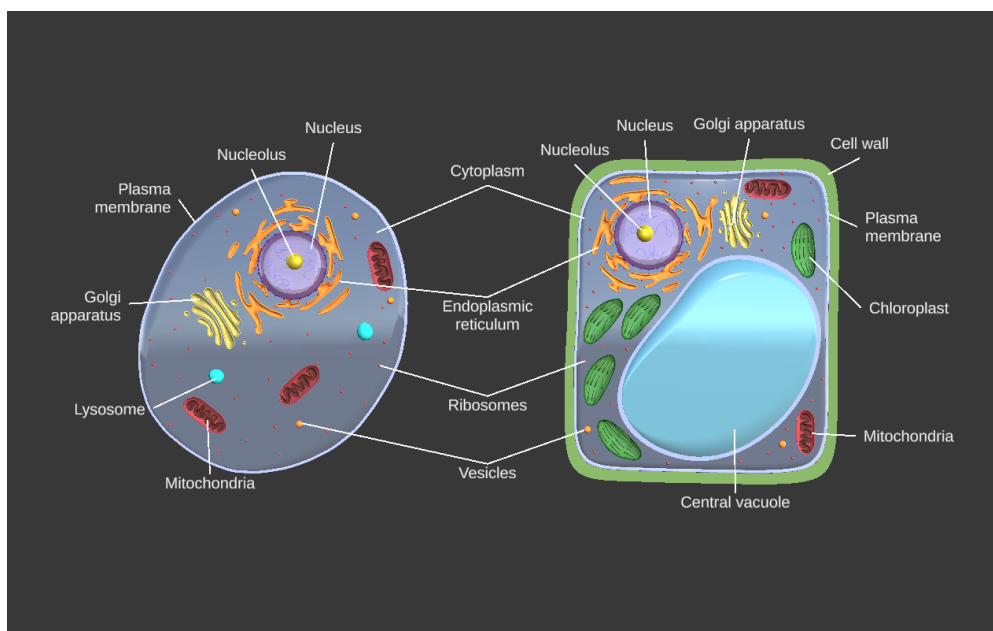
Pats projektas buvo kuriamas kiek kitaip nei ankstesni, norint išbandyti kitas Unity variklio galimybes. Patys 3D modeliai nėra kompleksiniai, sudaryti iš pakankamai paprastų geometrinių formų. Vibrio ląstelės modelis yra modifikuota kapsulė, spirilės modelis kurtas naudojantis Bežjė kreive, pritaikyta cilindru su kreivės ir masyvo modifikatoriais, o pritaikius modifikatorius buvo iškeltos naujos viršūnės (angl. *extrude vertices*) modelio galų užapvalinimui sukurti. Grupavimo si modeliai kurti naudojantis sukurtomis bakterijų formomis, kuriant naujas modelių kopijas ir jungiant gautas kopijas į vieną daugiakampių tinklą (angl. *polygon mesh*). Modeliai išeksportuoti į atskirus failus, medžiagos nebuvo uždėtos.

Buvo norima išbandyti kitas Unity variklio galimybes. Svarbiausi šio projekto bruožai yra parengtos dvi scenos, tekstūrų naudojimo bandymas, Unity dalelių sistemos išbandymas bei tolesnis

apdorojimas (angl. *post-processing*). Scenos keičiamos mygtuko paspaudimu, tam mygtukui buvo priskirtas C# skriptas, perjungiantis scenas. Bakterijų modeliams uždėti kolizijos tinklai (angl. *collider*), tinklas yra naudojamas užfiksuoti, ar pelytė yra užvesta ant objekto. Jei pelytė užvesta ant vienos iš bakterijų kolizijos tinklų, vartotojo sąsajoje rodomas bakterijos struktūros pavadinimas. Projekte taip pat naudojama Unity dalelių sistema, kuri panaudota norint užpildyti tuštumą ir įnešti vaizdinio intereso projektui, lyg bakterijos būtų kokioje nors aplinkoje, kurioje plaukioja baltymai ir panašios struktūros. Kameroms buvo priskirtas vaizdo apdorojimo profilis, kuris fokusuoja į priekyje demonstruojamus modelius, o toliau fone esantys modeliai atrodo neryškūs. Projekto ekrano kopija 9 pav.

4.6. Eukariotai. Gyvūninė ir augalinė ląstelės

Skaitytojas supažindinamas su gyvūninėmis ir augalinėmis ląstelėmis bei jų organelėmis.



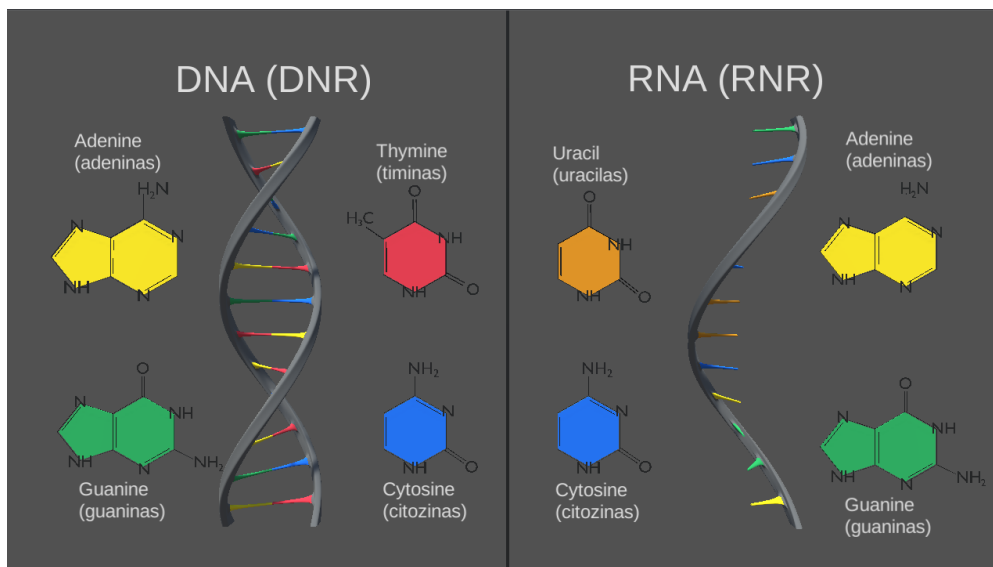
10 pav. Straipsnyje pateikto eukariotinės ląstelės WebGL projekto momentinė ekrano kopija

3D modeliai buvo kuriami kartu su prokariotinės ląstelės modeliu, todėl visi trys modeliai panašūs stilistika. Ribosomų pasiskirstymui naudojama dalelių sistema su *weight paint* informacija ribosomų pasiskirstymui po membraną bei šiuurkštų endoplazminį tinklą. Goldžio aparatas, endoplazminis tinklas ir mitochondrijos vidinės membranos kūrimui buvo pieštos 2D iliustracijos naudojantis Adobe Illustrator, o naudojantis Adobe Photoshop iliustracijos paverstos 3D objektais, kurie vėliau buvo įkelti ir tvarkomi Blender programoje mažinant viršūnių skaičių su *simplify mesh* funkcija. Organelių membranos sustorinimui naudojamas *solidify* modifikatorius, kuriame buvo nustatytas norimas membranų storis.

Šis Unity projektas savo interaktyvumu beveik identiškas į prokariotinės ląstelės organelių projektą (žr. 4.4 poskyrį), nes abu projektai buvo kuriami vienu metu. Naudoti objekto sukimo, linijų kūrimo skriptai.

4.7. DNR ir RNR

Orientuojamasi į molekulinės biologijos temą, aprašomi skirtumai tarp DNR ir RNR, taip pat jų panašumai bei funkcijos. Straipsnis svarbus tuo, nes kituose straipsniuose ne kartą minimi DNR ir RNR polimerai. Skaitytojui suteikiama informacija apie molekulėse esančias jungtis, nukleotidų sandarą, skirtumus tarp deoksiribozės ir ribozės cukrų ir kaip cukrų struktūra daro įtaką polimerų funkcijoms ir grandinėms. Supažindinama su RNR funkcijų svarba ląstelei.



11 pav. Straipsnyje pateikto DNR ir RNR WebGL projekto momentinė ekrano kopija

DNR spiralės modelis nėra tobula DNR spiralės atvaizdas, įprastai deoksiribozių fosfatinės grandinės (cukraus-fosfato stuburas) susisuka taip, kad dvigubos spiralės struktūroje matosi du elementai – didysis ir mažasis grioveliai. Šie elementai nėra atvaizduoti modelyje. Modelio tikslas nebuvo parodyti tikslią DNR spiralės struktūrą – buvo norima atvaizduoti azotinių bazių sąveiką.

Svarbu paminėti, kad įprastai RNR spiralė nėra susisukusi, nes RNR beveik niekada nesudaro dvigubos grandinės ir todėl neturi kaip palaikyti spiralės formos. Nors bazės sukurtos kaip 3D modelis, autorius rinkėsi kitą sprendimą panašioms projektams. Nupiešus 2D iliustraciją ir ją įdėjus modelių atvaizdavime kaip vartotojo sąsajos elementą, galima gauti beveik identišką rezultatą, tačiau būtų naudojama mažiau resursų.

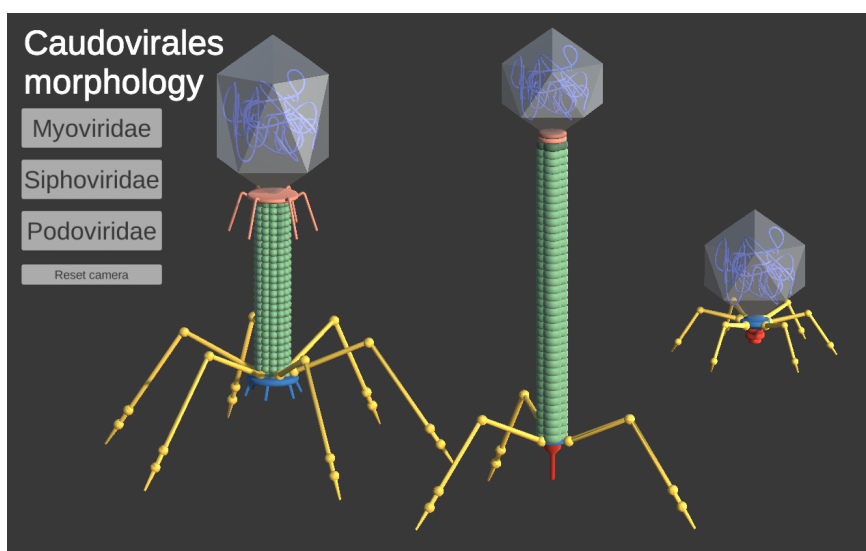
Šis projektas buvo pirmasis, kurtas naudojantis Unity WebGL (žr. 11 pav.). Darbo proceso metu naudojantis Blender sukurti 3D modeliai yra DNR spiralė, RNR grandinė bei azotinės bazės, įeinančios į DNR ir RNR nukleotidų struktūrą. Panaudotas modifikatorius, leidžiantis susukti DNR ir RNR spirales. Šis modifikatorius vadinamas paprastu deformavimu (angl. *simple deform*). Kopetėlių kūrimui naudotas masyvo modifikatorius, DNR dvigubos spiralės sukūrimui taip pat pritaikytas veidrodžio (angl. *mirror*) modifikatorius. Azotinių bazių kūrimui buvo naudotas viešinio atvaizdavimo (angl. *wireframe*) modifikatorius, leidžiantis 3D objektą paversti briaunomis. Taip buvo bandoma parodyti chemines bazių formules.

Unity projekte buvo kurtas tik vienas elgesio skriptas – objekto sukimas aplink ašį, kai ant

vienos iš spiralių yra nuspaudžiamas kairysis pelytės mygtukas ir pelytė yra judinama. Skriptui naudojamas kolizinis tinklas, objekto sukimas apie ašį remiasi fizika, paleidus pelytę objektas toliau sukasi iki tol, kol nustoja sukstis pats dėl objektui priskirtos trinties ir masės. Tekstams pateikti sukurta vartotojo sąsaja.

4.8. Virusai. Bakteriofagai

Straipsnyje skaitytojui pristatoma įprasta virusų struktūra. Virusai gali būti tiek DNR, tiek RNR, priklausomai nuo to, kaip viruse yra laikoma genetinė informacija. Pristatomos kai kurios virusinės ligos bei virusų įtaka žmonijai. Pristatomas žmogaus imuniteto atsakas į virusines infekcijas bei vakcinų veikimo principas, bakteriofagų žalą bakterijoms ir kaip jie gali padėti kovoti su bakterinėmis ligomis.



12 pav. Straipsnyje pateikto bakteriofagų WebGL projekto momentinė ekrano kopija

Buvo sukurti trys bakteriofagų modeliai, parodantys tris skirtingas struktūras, kurias gali turėti *Caudovirales* bakteriofagų šeimos nariai. Modelių kūrimo procese remtasi enterobakterijų fagų T4, T5 ir T7 virusų aprašymais ir egzistuojančia vizualia informacija. Išrinktas toks modeliavimo stilius, kuris nėra per daug detalus (pvz. modelis sukurtas molekuliname lygyje) ar supaprastintas, todėl suteikiantis pakankamai informacijos. Pasirinktas kitas variantas medžiagų nustatymui – per Blender programą tik sukurti 3D modeliai, o medžiagos buvo kuriamos ir taikomos objektų dalims per Unity žaidimų variklį. Toks variantas pasirinktas dėl to, nes dažnai importuojant modelius į Unity, medžiagos atrodo kitaip nei per Blender programą. Virusų genetinė medžiaga kurta tuo pačiu principu, kaip ir bakterijos DNR molekulė – naudojant Bezjė kreivę bei pritaikant reikalingus modifikatorius cilindriniam objektui. Virusų žiedinėms dalims, sudarytoms iš kitų objektų, buvo naudota dalelių emisija, kai dalelės yra įstatomos į viršūnių pozicijas, kuriant uodegų vamzdelius ir apvalkalus naudotasi masyvų modifikatoriumi. Kapsidei pritaikyta permatoma medžiaga, kad būtų galima matyti joje esančią nukleorūgštį.

Unity projekte buvo sukurta vartotojo sąsaja (žr. 12 pav.), kurioje paspaudus vieną iš mygtukų kameros pozicija yra pakeičiama, kad būtų rodoma vienas iš trijų bakteriofagų, rodomi viruso pagrindinių dalių pavadinimai. Pačiam viruso objektui buvo sukurtas elgesio skriptas, kuris suka viruso modelį apie ašį, taip rodydamas modelį iš visų pusių. Pagrindinis projekto langas – visi trys virusų modeliai vienoje kompozicijoje. Teksto ir viruso dalių sujungimui naudotas anksčiau minėtas linijų kūrimo skriptas.

Rezultatai ir išvados

Darbo metu buvo sukurta daug įvairių 3D modelių mikrobiologijos ir molekulinės biologijos temomis, kurie buvo panaudoti interaktyvių projektų kūrimui. Projektai buvo panaudoti su projektų tema susijusiuose straipsniuose, kurie buvo parašyti anglų ir lietuvių kalbomis bei buvo pateikti bioinformatikos svetainėje. Buvo pasiekta norimas rezultatas, išbandyta galimybė panaudoti 3D modeliavimu straipsnyje pateikiamos informacijos praturtinimui, pristatytos kelios galimybės, kaip ši kompiuterinės grafikos šaka gali būti naudojama. Darbas modeliuojant ir kuriant interaktyvius projektus yra vertingas ne tik skaitytojams, bet ir kuriančiajam, nes projekto sukūrimui reikia atidžios temos analizės.

Šis darbas – tik pradžia to, ką galima nuveikti naudojantis 3D modeliavimu. Yra didelis potencialas paruošti informaciją įvairiomis temomis, imtis dar ambicingesnių projektų bioinformatikos svetainės turiniui plėsti. Nors internete informacijos yra daug, be to, galimai jau egzistuoja panašūs projektai, yra potencialu galvoti, kad naudojant 3D technologijas galima kurti trumpus filmukus, demonstruojančius įvairiausių biologinių procesų veikimą ir eigą, tokius kaip ląstelių dalijimasis ir pan. Tai skatina ne tik gilinti žinias apie tam tikras temas, bet ir moko projektų rengimo atsakomybės, be to, leidžia studentams žiūrėti į mokslą iš kūrybiškesnės pusės.

Literatūra

- [Al-16] M. T. Al-mohanna. Morphology and classification of bacteria, 2016. URL: https://www.researchgate.net/publication/315803754_MORPHOLOGY_AND_CLASSIFICATION_OF_BACTERIA.
- [Bis14] Ö. T. Bishop. *Bioinformatics and Data Analysis in Microbiology*. Caister Academic Press, 2014.
- [CAB⁺21] K. Castro, T. F. Amado, C. J. Bidau ir P. A. Martinez. Studying natural history far from the museum: the impact of 3d models on teaching, learning, and motivation. *Journal of Biological Education*:1–11, 2021. URL: <https://www.tandfonline.com/doi/full/10.1080/00219266.2021.1877774>.
- [Com] Blender Online Community. Development. URL: <https://www.blender.org/get-involved/developers/>.
- [Com19a] Blender Online Community. About, 2019. URL: <https://www.blender.org/about/>.
- [Com19b] Fast Company. Unity technologies. *Most Innovative Companies*, 2019. URL: <https://www.fastcompany.com/company/unity-technologies>.
- [Con11] WebGL Public Wiki Contributors. Getting started, 2011. URL: https://www.khronos.org/webgl/wiki/Getting_Started.
- [Cor] Intel Corporation. Intel[®] open image denoise. high-performance denoising library for ray tracing. URL: <https://www.openimagedenoise.org/>.
- [Dyv21] V. Dyvlyash. 7 reasons why 3d modeling is important, 2021. URL: <https://www.linkedin.com/pulse/7-reasons-why-3d-modeling-important-vadim-dyvlyash/>.
- [FHP⁺18] T. Ferrin, C. Huang, E. Pettersen, T. Goddard, G. Couch, E. Meng ir S. Morris *et al.* Ucsf chimerax, 2018. URL: <https://www.rbvi.ucsf.edu/chimerax/>.
- [KBW] A. Klippel, M. Bagher ir J. O. Wallgrün. 2.5 procedural modeling. URL: <https://www.e-education.psu.edu/geogvr/node/558>.
- [Lee19] Tina Lee. Cpu vs. gpu renderer: which is better?, 2019. URL: <https://academyofanimatedart.com/cpu-vs-gpu-renderer-which-is-better/>.
- [LH17] Y. Loke, A. S. Harahsheh ir A. Krieger *et al.* Usage of 3d models of tetralogy of fallot for medical education: impact on learning congenital heart disease. *BMC Medical Education*, 17(54), 2017. URL: <https://doi.org/10.1186/s12909-017-0889-0>.

- [Mac18] R. J Mackenzie. One-way vs two-way anova: differences, assumptions and hypotheses, 2018. URL: <https://www.technologynetworks.com/informatics/articles/one-way-vs-two-way-anova-definition-differences-assumptions-and-hypotheses-306553>.
- [Tec22] Unity Technologies. Unity - manual: webgl browser compatibility, 2022.

Priedas nr. 1

Svetainės straipsnių nuorodos

Straipsniai lietuvių kalba:

1. Žvilgsnis į mikrobiologiją:
<https://www.bioinformatika.lt/mikrobiologija/zvilgsnis-i-mikrobiologija/3534/>
2. Ląstelių tipai ir organelės:
<https://www.bioinformatika.lt/mikrobiologija/lasteliu-tipai-ir-organeles/3544/>
3. Eukariotai. Gyvūninė ir augalinė ląstelė:
<https://www.bioinformatika.lt/mikrobiologija/eukariotai-lasteles/3547/>
4. Prokariotai. Bakterijos ir archėjos:
<https://www.bioinformatika.lt/mikrobiologija/prokariotai-bakterijos-ir-archejos/3554/>
5. Prokariotai. Bakterijų klasifikacija:
<https://www.bioinformatika.lt/mikrobiologija/prokariotai-bakteriju-klasifikacija/3556/>
6. DNR ir RNR:
<https://www.bioinformatika.lt/mikrobiologija/dnr-rnr/3288/>
7. Virusai. Bakteriofagai:
<https://www.bioinformatika.lt/mikrobiologija/virusai-bakteriofagai/3559/>

Straipsniai anglų kalba:

1. A sight into microbiology:
<https://www.bioinformatika.lt/en/microbiology/a-sight-into-microbiology/3561/>
2. Cell types and cellular organelles:
<https://www.bioinformatika.lt/en/microbiology/cell-types-and-cellular-organelles/3563/>
3. Eukaryotes. Animal and plant cells:
<https://www.bioinformatika.lt/en/microbiology/eukaryotes-animal-and-plant-cells/3565/>
4. Prokaryotes. Bacteria and archaea:
<https://www.bioinformatika.lt/en/microbiology/prokaryotes-bacteria-and-archaea/3567/>
5. Prokaryotes. Bacteria classification:
<https://www.bioinformatika.lt/en/microbiology/prokaryotes-bacteria-classification/3569/>
6. DNA and RNA:
<https://www.bioinformatika.lt/en/microbiology/dna-and-rna/3572/>
7. Viruses. Bacteriophages:
<https://www.bioinformatika.lt/en/microbiology/viruses-bacteriophages/3576/>

Priedas nr. 2

Darbo repozitorija

Pagrindinė darbo repozitorija:

<https://github.com/skimmeds/unity-webgl>

Projekto puslapis, kuriame matomos nuorodos į straipsnių projektus bei kitus bandymus:

<https://skimmeds.github.io/unity-webgl/>

Galima rasti Blender failus, tiek panaudotus galutiniuose straipsniuose, tiek nepanaudotus bandymus, taip pat Blender Cycles projekto failus 3D modeliavimo praktikai. Įkelti veikiantys Unity WebGL projektai, nepavykęs lokalizacijos bandymas bei veikiantis Three.js pavyzdys su prokariotine ląstele. Galima rasti kai kuriuos C# kalba parašytus skriptus.

Kai kurie Unity C# skriptai, parašyti projektų interaktyvumui:

<https://github.com/skimmeds/unity-webgl/tree/main/C%23%20scripts>

Priedas nr. 3

Three.js bandymo repozitorija ir kodas

Failai prokariotinės ląstelės 3D modelio atvaizdavimui naudojantis Three.js biblioteka:

<https://gist.github.com/skimmeds/1b6b5d88291e7fad285c62395c3a6b83>

Prokariotinės ląstelės (bakterijos) modelio atvaizdavimo naudojantis Three.js biblioteka rezultatas:

<https://skimmeds.github.io/unity-webgl/THREE-prokaryotic/index.html>