Transformacija 2D SVG elemenata u animirano 3D objekt

Bekavac, Mateo

SVEUČILIŠTE U ZAGREBU
GRAFIČKI FAKULTET

ZAVRŠNI RAD

Mateo Bekavac

Sveučilište u Zagrebu
Grafički fakultet

Smjer: Tehničko-tehnološki

ZAVRŠNI RAD

TRANSFORMACIJA 2D SVG ELEMENATA U ANIMIRANI 3D OBJEKT

Mentor:

doc. dr. sc. Tajana Koren Ivančević

Student:

Mateo Bekavac

Zagreb, 2022

SAŽETAK

Za prikaz grafike na web stranici standard je SVG format - XML jezik koji ima mogućnost zadržavanja visoke rezolucije 2D vektorske grafike neovisno o stupnju *skaliranja* iste. Skalabilnu vektorsku grafiku moguće je izraditi koristeći programe s grafičkim sučeljem, ili s običnim uređivačem teksta. Napretkom preglednika pojavljuje se mogućnost prikaza 3D objekta u 3D okruženju, te postaje nezamjenjiv način virtualne prezentacije proizvoda s raznim mogućnostima interaktivnosti.
Danas postoji mnogo alata posvećenih lakšem modeliranje kompleksnih 3D objekata koji također imaju mogućnost pripreme izrađenog modela za prikaz na webu, jedan od takvih alata je Blender.
U teorijskom dijelu ovog rada proučavaju se alati za izradu vektorske grafike i 3D objekata, te osnovne funkcije tih alata potrebne u svrhu rada.
Eksperimentalni dio rada odnosi se na izradu vektorske grafike u HTML-u koju se animira CSS-om, te zatim izrade animiranog 3D objekta na temelju dobivene grafike koristeći Blender.

Cilj rada je usporediti mogućnosti prikaza 3D objekta, izrađenog i animiranog CSS-om i HTML-om u odnosu na implementaciju 3D animiranog objekta izrađenog u Blenderu.

Ključne riječi: HTML/CSS, vektorska grafika, 2D vektorska grafika, 3D objekti, animacija

ABSTRACT

The standard format for display of graphics on the web is SVG – an XML language with the ability to maintain high quality regardless of the degree it is scaled to. Scalable vector graphics can be made with a dedicated software, or with a simple text editor. The rise of browser capability also made it possible to display 3D objects on a website, which became an irreplaceable way of virtually presenting a product, with many interactivity options. Today there exists a wide range of software dedicated for easier modelling of complex 3D objects which can also be used to prepare the same object for its display on a website, one of these tools is called Blender.

The theoretical part of this thesis studies tools used in creation of vector graphics and 3D objects, as well as the basic functions of those tools necessary for the completion of this work.

The experimental part of the thesis is focused on the creation of vector graphics using HTML, and animating it with CSS, as well as the modelling of a 3D object based on the created graphics using Blender.

The goal of this thesis is to compare display capabilities of a 3D object created and animated using HTML and CSS, as opposed to the implementation of a 3D object created and animated using Blender.

Keywords: HTML/CSS, vector graphics, 2D vector graphics, 3D objects, animation

SADRŽAJ

1. UVOD

Porast korištenja interneta kao medija za dijeljenje informacija značilo je i potrebu za novim, boljim načinima prikaza sadržaja. Rasterska grafika (jpg, png,..) koristi se za prikaz slikovnog sadržaja s mnogo detalja, dok vektorska grafika postaje standard za prikaz dvodimenzionalne grafike - tablica i ilustracija na web stranicama. Svaki dio SVG-a, kao i svako njegovo svojstvo, se može animirati koristeći *animate* oznaku ili CSS, no trodimenzionalne transformacije i animacije CSS-om su donekle ograničene ili zahtijevaju znanje i korištenje JavaScript-a.

Razvoj grafičkih alata za modeliranje 3D objekata donio je mogućnost prikaza u novoj dimenziji – stvaranje 3D modela postojećih 2D grafika, te izrada modela proizvoda koji se nalaze na stranici, se pokazalo kao kvalitetan način privlačenja i zadržavanja posjetioca, te konkuriranja na tržištu. Potencijal ove tehnologije na stranicama nekih proizvođača se očituje mogućnošću mijenjanja svojstava prezentiranih proizvoda u stvarnom vremenu, kao što su boja ili materijal.

Ovaj završni rad teorijskim dijelom dalje opisuje spomenute tehnologije te njihove mogućnosti, a eksperimentalni dio se bavi izradom zamišljene ilustracije HTML-om i animacije CSS-om, te izradom 3D objekta u Blenderu koristeći izrađenu grafiku kao polaznu točku.

2. TEORIJSKI DIO

U ovom dijelu rada se kratko opisuje nastanak HTML-a, CSS-a i SVG-a, što su to alati za 3D modeliranje i za što se koriste. Također se prikazuju primjeri sintakse korištenih jezika, tj. način definiranja HTML oznaka te uređivanja istih CSS-om.

2.1. HTML i CSS

Prvotno jednostavnog izgleda, internet je korišten u svrhu komunikacije unutar jednog, a kasnije između više sveučilišta. Kako bi se prikaz olakšao Tim Berners Lee tijekom 1980.-ih razvija HTML (*HyperText Markup Language*) koji se kasnije nadopunjuje i s CSS-om (*Cascading StyleSheets*).

HTML se koristi za definiranje osnovne raspodjele elemenata na stranici, dok se CSS koristi za daljnju manipulaciju tim elementima.
Dizajn elemenata web-stranice je bitan kako bi se sadržaj prikazao na što bolji, čitljiviji i intuitivniji način kako bi se olakšala navigacija na stranici. Prosječni će korisnik na temelju izgleda stranice u kratkom vremenu odlučiti o nastavku korištenja iste, stoga je uporaba jedinstvenih elemenata dizajna jako bitna. Većina poznatih korporacija u dizajnu koristi: videe, slike, 2D grafike i 3D grafike (animirane i statične), a sve više ih koristi i interaktivna 3D okruženja s modelima visoke rezolucije.

HTML ili *HyperText Markup Language* je kod koji služi za strukturiranje web stranice te sadržaja na web-stranici. Za korištenje HTML-a dovoljan je obični alat za uređivanje teksta (npr. *Notepad*) te poznavanje osnovnih HTML elemenata. Sadržaj neke web-stranice se može oblikovati kao npr. naslov, lista (numerirana ili nenumerirana), paragraf, itd [1].

HTML elementi se definiraju na sljedeći način:

```
<h1> Ovo je naslov </h1>

<a> Ovo je link </a>
```

Naziv HTML elementa se definira unutar izlomljenih zagrada, a svaki element mora imati svoj "zatvarajući" par koji unutar izlomljenih zagrada, prije naziva elementa, sadrži znak "/". Na stranici se prikazuje tekst koji se nalazi između tog para zagrada, pa će se u slučaju primjera iznad na stranici prikazati tekst "Ovo je naslov" i hiperveza "Ovo je link". Neki elementi ne trebaju zatvarajući par, te se definira tako da se prije zatvarajuće izlomljene zagrade stavi znak "/":

```
<img ... />
```

U nekim slučajevima će se tekst prikazati neovisno o tome je li zatvarajući par elementa definiran, iako to nije preporučivo.

HTML elementi također sadrže i atribute, koji ih dalje opisuju te daju više informacija o elementu. Mogu se odnositi na dimenzije elementa, izvor sadržaja elementa, alternativni opis elementa, oblikovanje teksta i sl.
Svi atributi imaju osnovni izgled:

```
ime = “vrijednost”
```

Atributi elemenata se mogu definirati na sljedeći način:

```
<h1 style=”color:red”> Ovo je naslov </h1>

<a href=”https://www.grf.unizg.hr/”> Ovo je link </a>

<img src=”slika.jpg” height=”300” width=”700” />
```

U ovom primjeru je atributu “src” elementa <img> pridana vrijednost “slika.jpg”, što označava izvor sadržaja elementa, odnosno mjesto gdje se nalazi slika koju želimo prikazati. Slike se obično spremaju u jednu datoteku koja sadrži više različitih, kategoriziranih datoteka zbog lakšeg snalaženja, jer se kao lokacija slike navodi putanja od datoteke u kojoj se nalazi HTML dokument do datoteke u kojoj se nalazi slika. Osim lokalno, sliku se može “pozvati” i hipervezom s nekog mjesta na internetu tako da kao izvor navedemo internet vezu slike.

Element <a> sadrži atribut “href” kojim se definira odredišna lokacija, odnosno mjesto na stranici, ili internetu s kojim nas taj element povezuje. U ovom primjeru klikom na tekst “Ovo je link” odlazimo na web stranicu grafičkog fakulteta.

Element <h1> ima atribut “style” kojim se definira CSS nekog elementa. Definiranje CSS svojstava unutar HTML oznake naziva se *inline* način - između navodnika se definiraju svojstva koja se od svojih vrijednosti odvajaju dvotočkom, a vrijednosti se zatim zatvaraju s točkom i zarezom.

Osim kao atribut HTML elementa, “style” se može definirati i kao HTML oznaka unutar <head> elementa, gdje se područje između oznaka <style> upisuje CSS. Također se može pisati eksterno u obliku CSS datoteke.
CSS se sastoji od svojstava kojima se definira vrijednost, te se pridodaju HTML oznakama na način:

```
h1 {color: red;}
```

Osim po svojoj oznaci, svaki HTML element se može klasificirati s jedinstvenim ili djeljivim identifikatorima. Za postavljanje jedinstvene oznake koristi se atribut “id”, a ako želimo definirati ista CSS svojstva za više elemenata koristimo atribut “class”. Klase se u CSS pozivaju tako da se stavi točka pred naziv klase:

```
.imeKlase {svojstvo:vrijednost;}
```

Dok se jedinstveni identifikator “id” poziva sa znakom ljestava:

```
#upisaniID {svojstvo:vrijednost;}
```

2.2. SVG

Skalabilna vektorska grafika (SVG) vektorski je format datoteke male veličine prilagođen webu. Razvijen je od strane World Wide Web Consortium-a s ciljem kompatibilnosti s drugim web formatima kao što je DOM, HTML, CSS i JavaScript.
Relativno mala veličina SVG dokumenata znači i brže učitavanje grafike na stranici čime sama stranica postiže bolje performanse [2].
Kod vektorske grafike slike se pohranjuju putem matematičkih formula temeljenih na koordinatama točaka i linijama na mreži. Time je omogućena značajna promjena veličine odnosno *skaliranje* SVG datoteka bez gubitka razlučivosti što je čini idealnom za izradu ikona i logotipa za web stranice. Skalabilna vektorska grafika na web stranicama se koristi i za ilustracije, animacije, dijelove sučelja, infografike i vizualnu reprezentaciju podataka [3].

SVG je napisan u XML kodu što znači da su sve informacije umjesto kao oblici, pohranjene kao tekst, koji je neovisno o rotaciji i transformaciji, vidljiv tražilicama kao što je Google u procesu traženja ključnih riječi *keywords*. Ova karakteristika omogućava tražilicama da čitaju tekstualne podatke infografika poput dijagrama i grafikona, što dovodi do optimiziranije pretrage i potencijalno veće posjećenosti web stranice [4].

Osim opisivanja SVG-a kodom, koriste se i programi kompatibilni sa SVG formatima kao što su Adobe Illustrator ili Inkscape. Ovi programi omogućuju lakšu manipulaciju grafikom – definiraju se dimenzije SVG dokumenta, a zatim koristeći široki spektar alata moguće je klikom miša nacrtati kompleksnu grafiku u kraćem vremenu u odnosu na direktno pisanje koda. Nakon spremanja dokumenta u SVG formatu, isti je ponovno moguće otvoriti u uređivaču teksta za potrebe animiranja i slično.
SVG element se u HTML dokumentu poziva oznakom <svg> kojoj zatim definiramo dimenzije koje će predstavljati koordinatni sustav:

```
<svg width=”500” height=”200”>

</svg>
```

Elementi koji se definiraju unutar SVG prostora mogu biti:

- Krug - <circle>
- Elipsa - <ellipse>
- Četverokut - <rect>
- Linijski - <line>
- Višelinijski - <polyline>
- Mnogokut - <polygon>
- Put - <path>
- Tekst - <text>

Svaki element se sastoji od različitih atributa koji opisuju njihovu poziciju u SVG prostoru. Krugu se opisuju točke središta X i Y te radijus r:

```
<svg height=”200” width=”200”>
          <circle cx = “X”  cy = “Y” r=”20” />
</svg>
```

Elipsi se također opisuju točke središta, no za razliku od kruga potrebno je definirati radijus “rx” po X i “ry” po Y osi.
Osnovni atributi za definiranje četverokuta su pozicije X i Y nakon kojih slijedi opisivanje visine i širine atributima *width* i *height*.
Linije se sastoje od početne točke definirane atributima *X1* i *Y1,* te točke u kojoj linija završava opisane *X2* i *Y2* atributima.
Višelinijski elementi se opisuju atributom *points* koji se sastoji od više parova točaka koji predstavljaju kretanje linije po koordinatnom sustavu SVG prostora.
Mnogokuti se kao i višelinijski elementi definiraju atributom *points* gdje parovi točaka predstavljaju vrhove mnogokuta.

Put, odnosno <path> element, je jedan od najkorisnijih elemenata SVG-a. Koristeći <path> moguće je prikazati linije, krivulje, kompliciranе oblike i još mnogo toga. Za definiranje puta mogu se koristiti naredbe:

- M – *moveto*
- L – *lineto*
- V – *vertical lineto*
- H – *horizontal lineto*
- C – *curveto*
- S – *smooth curveto*
- Q - *quadratic Bezier curve*
- T – *smooth quadratic Bezier curveto*
- A – *elliptical Arc*
- Z – *closepath*

Navedene naredbe se u oznaci *path* definiraju unutar atributa “d” tako da se nakon pozivanja naredbe upišu željene točke X i Y:

```
<svg height=”200” width=”200”>
        <path d=”M20 20 L40 40 L40 20 Z” />
</svg>
```

Nadalje, SVG datoteke moguće je animirati pomoću JavaScripta ili CSS-a.
SVG elementi i atributi se također mogu animirati korištenjem <animate> oznake.
Unutar izlomljenih zagrada se definiraju svojstva animacija, neka od njih su:

- attributeName – kao vrijednost unosimo ime svojstva koje želimo animirati
- from - početna vrijednost
- to – krajnja vrijednost
- dur – vrijeme trajanja animacije
- begin – označava početak animacije, vrijednost može biti: trajanje vremenske odgode prije početka, prijelaz ili klik mišem, kraj ili početak neke druge animacije
- keyTimes – definira trenutke transformacije za vrijeme animacije, vrijednosti se odvajaju točkom i zarezom
- values – definira vrijednost transformacije tijekom animacije, vrijednosti se odvajaju točkom i zarezom

Atributi “keyTimes” i “values” su povezani tako da svaka vrijednost “keyTimes” atributa odgovara vrijednosti unutar “values” atributa na istoj poziciji. Na ovaj način se animacija dijeli na više sekvenci u kojima se promjene događaju u definiranim trenutcima.
Vrijednosti koje se definiraju atributu “keyTimes” se nalaze u rasponu od 0 do 1 te predstavljaju postotak vremena od ukupnog trajanja animacije.
Za animaciju pulsirajućeg kruga, kod može biti:

```
<circle cx=”50” cy=”50” r=”20”>
        <animate attributeName=”r”
        dur=”8s”
        values=”20; 40; 25; 35; 20”
        keyTimes=”(0; 0.25; 0.5; 0.75; 1” />
```

```
</circle>
```

Animacija CSS-om se također vrši u sekvencama. Koristeći naredbu "@keyframes" definira se tijek animacije u postotcima, a sama animacija se za HTML element koji se animira veže CSS svojstvom "animation-name".
Kod za pulsirajući krug klase "puls", animiran CSS-om u trajanju od 8 sekundi, može biti:

```
 .puls{
width: 20px;
height: 20px;
animation-name: primjer;
animation-duration: 8s;
transform: scale(1);

}

@keyframes primjer{
0%{transform: scale(1);}
25%{transform: scale(2);}
50%{transform: scale(1.3);}
75%{transform: scale(1.6);}
100%{transform: scale(1.2);}
}
```

Koristeći različita svojstva, animaciji se može odrediti: trajanje, odgodu početka, način odvijanja, broj ponavljanja, te ponašanje po završetku animacije.

2.3. Alati za 3D modeliranje

U 3D računalnoj grafici, 3D modeliranje je naziv za proces razvoja matematičke koordinatne reprezentacije bilo kojeg (živog ili neživog) objekta ili površine u tri dimenzije putem specijaliziranog softvera u simuliranom 3D prostoru. Tako dobiveni

objekti nazivaju se 3D modeli, a definirani su kao skup podataka (točke, linije, poligoni) u 3D prostoru. Moguće ih je izraditi manualno, algoritamski ili skeniranjem [5].

Uobičajeno početak procesa 3D modeliranja započinje odabirom osnovnog geometrijskog oblika koji najbolje odgovara zamišljenom modelu, te je potom moguća promjena oblika manipulacijom bilo koje točke na modelu. To je moguće zbog činjenice da softver identificira lokaciju svake okomite ili vodoravne točke, korištenjem koordinatnih podataka, te omogućuje njihovu manipulaciju u odnosu na referentnu točku [5].

3D modeliranje važan je proces u realizaciji mnoštva projekata u različitim industrijama. Možda je najčešća upotreba procesa 3D modeliranja za razvoj video igrica. Na taj način se izrađuju modeli likova, rekvizita i krajolika. Važnost kvalitete ovih modela direktno utječe na tzv. imerzivnost – sposobnost video igrice da korisnik uroni u njen sadržaj.
Važnost imerzivnosti video igrice dodatno se povećava kada su u pitanju igre virtualne stvarnosti. Igre virtualne stvarnosti omogućuju krajnjem korisniku doslovno zakorači u potpuno novi trodimenzionalni svijet zamjenjujući njegovu stvarnost s virtualnim 360 krajolikom [5].

Proces 3D modeliranja uobičajen je i u arhitekturi za prikazivanje trodimenzionalnih modela građevina, dizajnu proizvoda, 3D ispisu i animaciji.

Prikaz samo jednog neživog modela najčešći je u pripremi za 3D ispis, no mnogo su češći prikazi više animiranih 3D modela u interakciji s neživim objektima. Animirani objekti su pokretni likovi u sekvencama animacije, dok neživi objekti s kojima su u interakciji čine raspored scene [6].
Animacija 3D modela se sastoji od mijenjanja svojstava modela kroz neki vremenski period koristeći ključne kadrove na vremenskoj crti animacije. Ključni kadrovi su vremenske oznake u koje su spremljene informacije o transformaciji nekog svojstva, te se promjena odvija između dva ključna kadra, odnosno početne i krajnje vrijednosti transformacije. Jednostavniji modeli s manje pokretnih dijelova se mogu animirati mijenjanjem željenih postavki na ključnim kadrovima, dok je za kompliciranije modele kao što je ljudsko tijelo potrebno izraditi kostur kojim se model dijeli na više dijelova koji se

mogu ciljano animirati.
Neki suvremeni alati za 3D modeliranje podržavaju i tzv. *motion capture* – sposobnost animiranja kostura modela tako da se na stvarnu osobu, na ključnim mjestima, postave senzori čija se kretnja zapisuje na računalo te pretvara u digitalni zapis. Ovom metodom moguće je animirati kretnje na jednostavniji, ali i skuplji način, eliminirajući potrebu za ručnom animacijom svakog zasebnog dijela modela.

2.3.1. Blender

Blender je besplatni alat za 3D modeliranje, te spada u programe otvorenog koda. Omogućuje izvedbu cijelog procesa 3D oblikovanja: modeliranje, opremanje, animaciju, simulaciju, renderiranje, sastavljanje i praćenje pokreta... Blender je višeplatformski i jednako dobro radi na Linux, Windows i Macintosh računalima. Njegovo sučelje koristi OpenGL za pružanje dosljednog iskustva. Kako bi se potvrdila određena kompatibilnost, popis podržanih platformi označava one koje redovito testira razvojni tim [6].

Temeljni cilj korištenja računalnog grafičkog programa, kao što je Blender, jest proizvesti prikaz na zaslonu računala koji se potom renderira u digitalnu sliku ili niz slika za animaciju.

Blender također ima mogućnost otvaranja SVG dokumenta, koju se dalje može uređivati u 3D prostoru. Ovo je moguće zbog toga što su elementi SVG grafike definirani matematički u koordinatnom sustavu, te ih je lako prenijeti u novi prostor. Rezultat *importanja* SVG-a je dvodimenzionalna grafika koja se nalazi na XY površini trodimenzionalnog XYZ prostora. Kako je svaki SVG element definiran kao krivulja za prikaz na 2D površini, u 3D prostoru dolazi do preklapanja. Svaki element ili grupe elemenata je potrebno prvo pretvoriti u 3D mrežu kako bi se mogli odvojiti po novoj osi te olakšala daljnja manipulacija slike u model. Nakon optimiziranja SVG dokumenta, elemente vektorske grafike je moguće istiskivati ili utiskivati čime dobivaju dubinu, mogu se smanjivati ili uvećavati, mijenjati teksture, te kasnije i animirati.

Animacija u Blenderu se izvodi pomoću ključnih kadrova, kojima se definiraju sekvence u vremenu trajanja animacije u kojima se odvija određeni dio animacije. Blender se zbog svoje osebujnosti može koristiti za sve dijelove produkcije u 2D ili 3D prikazu, od izrade scene i modela na sceni do animiranja i renderiranja istih.
Za potrebe animacije kompliciranih modela slažu se kosturi koji odgovaraju modelu koji se animira, a moguće ih je izraditi ručno ili automatski, koristeći funkcije programa.

3. EKSPERIMENTALNI DIO

Eksperimentalni dio rada sastoji se od dva dijela:

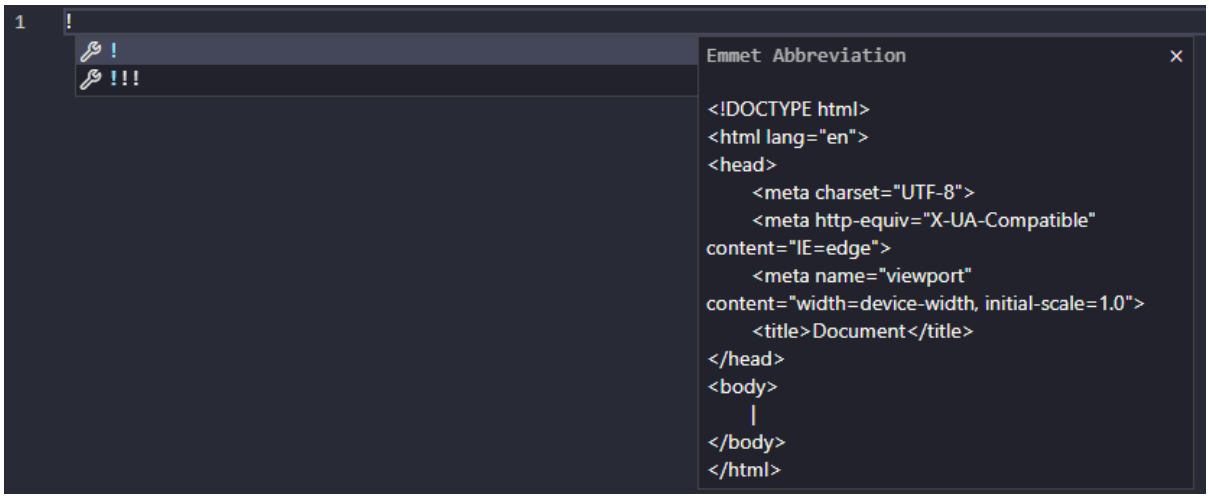
- Izrada SVG-a
- Izrada 3D modela na temelju SVG-a

Kao motiv je odabrana vjetrenjača kojoj je cilj animirati rotor i jedra.

3.1. Izrada SVG-a

Koristeći alat za uređivanje teksta “Visual Studio Code” prvo izrađujemo HTML dokument kojeg nazivamo “index.html” te ga spremamo u datoteku koja će sadržavati i ostale dokumente.

Prvo se definira osnovna HTML struktura. Visual Studio Code ima mogućnost automatskog ispunjavanja te je osnovnu strukturu moguće ispisati tako da unesemo uskličnik i odaberemo prvu opciju iz padajućeg izbornika.



Slika 1 – osnovna HTML struktura

Na slici jedan se vidi kako odabirom prve opcije program nudi ispis osnovne strukture, nakon čega preostaje odrediti naziv stranice.

Za izradu SVG-a koristi se oznaka <svg> kojoj se definiraju širina i visina te “viewBox” svojstvo koje opisuje dimenzije i poziciju prozora za prikaz SVG-a u korisničkom prostoru.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Vjetrenjača</title>
8  </head>
9  <body>
10     <svg width="800" height="800" viewbox="0px 0px 80 80">
11
12
13     </svg>
14 </body>
15 </html>
```

Slika 2 – SVG oznaka unutar body oznake

Nakon definiranja širine i visine od 800 piksela počinje se izradom grafike vjetrenjače. Elementi vjetrenjače će biti podijeljeni u tri skupine različitih klasa, a sve skupine će se nalaziti unutar zajedničke skupine klase “.scena”.

Skupine elemenata grafike su:

1. Tijelo - “.vjtr”
2. Prozori - “.prozori”
3. Rotor i jedra - “.rot”

Skupine elemenata unutar SVG prostora se definiraju koristeći <g> oznaku.

```
<body>
    <svg width="800" height="800" viewbox="0 0 80 80">

        <g id="scena">
            <g class="vjtr">
            </g>

            <g class="prozori">
            </g>

            <g class="rot">
            <g></g>
        </g>

    </svg>
</body>
</html>
```

Slika 3 – označavanje skupina

U skupini *scena* mogu se definirati pozadinski elementi grafike jer će se po slijedu označavanja nalaziti ispod ostalih elemenata. Za pozadinu grafike koristit će se SVG kvadrat stopostotne širine SVG prostora, dok će tlo na kojem se vjetrenjača nalazi biti podijeljeno na dva dijela: tlo i zemlju.

Kako bi se postigao trodimenzionalni efekt koristit će se <polygon> oznaka koja će biti definirana tako da izgleda kao perspektivno distorzirani mnogokut.

```
<g id="scena">

    <rect width="100%" height="100%" fill="#35475f"></rect>
    <polygon class="tlo" points="200,480 500,480 450,370 250,370" fill="teal" />
    <polygon class="zemlja" points="200,480 500,480 500,600 200,600" fill="#653912" />
```

Slika 4 - definiranje pozadine i tla

Slika četiri prikazuje kod za prikaz pravokutnika plave boje (#35475f) koji pokriva cijeli SVG prostor, a zatim dva četverokuta “tlo” i “zemlja”, s tim da je “tlo” definirano kao plavozeleni četverokut trapeznog oblika, a zemlja kao četverokut smeđe boje.

Slika 5 – SVG u pregledniku

Slika pet je prikaz elemenata SVG-a u pregledniku te je vidljivo kako je korištenjem <polygon> oznake postignut izgled polegnutog kvadra.

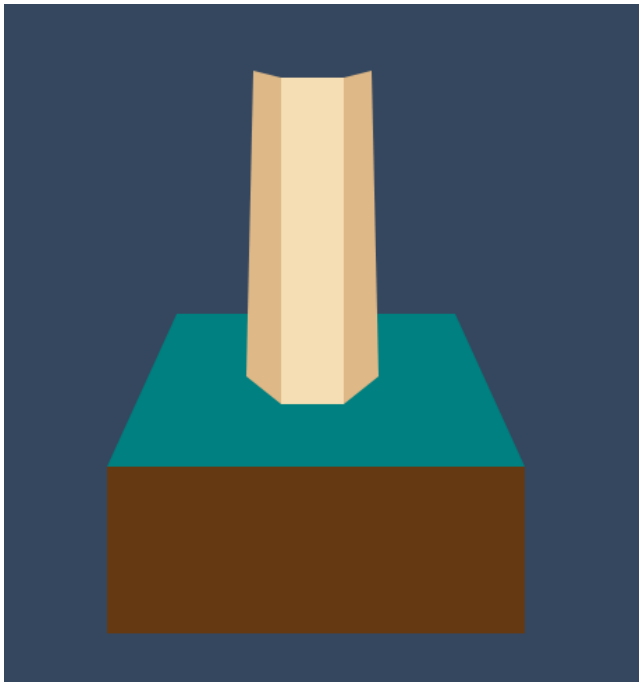
Idući korak je izrada dijelova vjetrenjače, također koristeći <polygon> oznake.
Kako bi se zadržala prividna trodimenzionalnost vjetrenjača je zamišljena s osmerokutnim tlocrtom, a promatraču će biti vidljive tri strane od ukupnih osam.

```
<g class="vjtr">
    <polygon points="300,415 325,435 325,200 305,195" fill="burlywood" />
    <polygon points="325,435 370,435 370,200 325,200" fill="wheat" />
    <polygon points="370,200 390,195 395,415 370,435" fill="burlywood" />
```

Slika 6 – Definiranje tijela vjetrenjače

Slika 6 sadrži kod za izradu triju četverokuta koji će činiti vidljivi dio tijela vjetrenjače.

Za postizanje 3D efekta na tijelu, srednji zid se definira kao pravilni četverokut, a zidovi sa strane kao nepravilni, prividno nagnuti četverokuti.
Kod lijevog zida je donji lijevi kut u odnosu na središnji zid podignut za 20, a gornji lijevi rub za 5, dok je kod desnog zida donji desni kut podignut za 20, a gornji desni za 5.
Vanjski zidovi vjetrenjače ispunjeni su tamnijom nijansom bež boje “burlywood” dok je središnji zid ispunjen svjetlijom nijansom “wheat” kako bi se postigao privid frontalnog osvjetljenja.



Slika 7 – izgled zidova u pregledniku

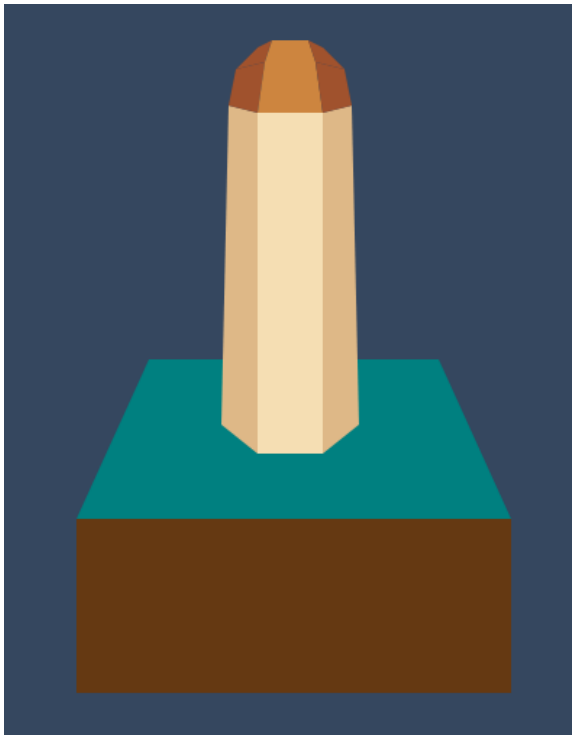
Osim zidova u skupini tijela vjetrenjače nalaze se i elementi krova, odnosno kupole na

kojoj će se nalaziti rotor koji će biti animirani dio grafike. Kako bi rotor i jedra bili paralelni promatraču, kupolu ćemo definirati kao šest četverokuta koji svojim oblikom i položajem simuliraju zakrivljenost.

```
<polygon points="305,195 325,200 330,165 310,170" fill="sienna" />
<polygon points="325,200 330,165 365,165 370,200" fill="peru" />
<polygon points="365,165 370,200 390,195 385,170" fill="sienna" />

<polygon points="310,170 330,165 335,150 325,155" fill="sienna" />
<polygon points="335,150 330,165 365,165 360,150" fill="peru" />
<polygon points="360,150 365,165 385,170 370,155" fill="sienna" />
```

Slika 8 – definiranje kupole

Nakon dodavanja elemenata kupole, skupina tijela vjetrenjače je dovršena te u je u pregledniku prikazana na sljedeći način (slika 9):



Slika 9 – prikaz tijela vjetrenjače na tlu

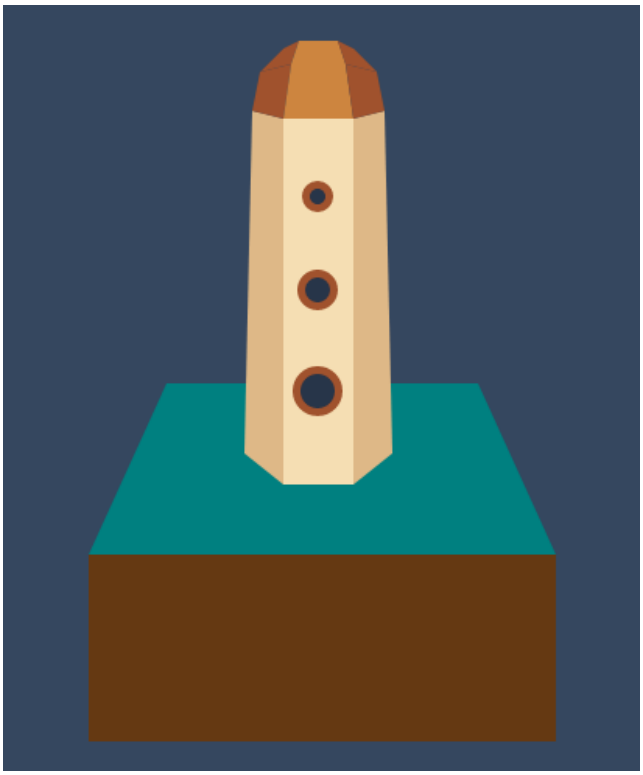
U sljedećem koraku se središnjem zidu vjetrenjače dodaju prozori kako bi se razbila monotonost tijela, a postojati će tri prozora različitih veličina. Prozore je moguće prikazati

oznakom <circle> čijim atributom *stroke* se može definirati izgled okvira prozora, a atributom *fill* ispuniti unutrašnjost bojom koja će predstavljati staklo. Osim na taj način, prozore je moguće prikazati pomoću parova krugova različitih promjera, tako da se definiraju dva <circle> elementa jedan nakon drugog, gdje prvi služi kao okvir sa smeđom ispunom, a drugi kao staklo s tamnijom plavom ispunom. Kako je grafika namijenjena za *importanje* u Blender, zbog lakšeg manipuliranja elementima koristit će se druga metoda.

```
<g class="prozori">
    <circle cx="347" cy="250" r="10" fill="sienna" />
        <circle class="proz" cx="347" cy="250" r="5" fill="#273549" />

    <circle cx="347" cy="310" r="13" fill="sienna" />
        <circle class="proz" cx="347" cy="310" r="8" fill="#273549" />

    <circle cx="347" cy="375" r="16" fill="sienna" />
        <circle class="proz" cx="347" cy="375" r="11" fill="#273549" />
</g>
```

Slika 10 – definiranje prozora

Na slici deset nalazi se kod za tri para krugova pozicioniranih na sredinu tijela po X osi, međusobno udaljenih po Y osi. Svaki par se sastoji od dva kruga različitih dimenzija i boja, a krugovima koji predstavljaju staklo su zadane klase ".proz".



Slika 11 – rezultat dodavanja prozora

Nakon prozora, preostaje definirati skupinu koja se sastoji od rotora i jedara - rotor je dio vjetrenjače na koji su pričvršćena jedra, a prikazat ćemo ga na isti način kao i prozore.

```
<circle cx="347" cy="185" r="10" fill="antiquewhite" />
<circle cx="347" cy="185" r="3" fill="saddlebrown" />
```

Slika 12 – kod za prikaz rotora

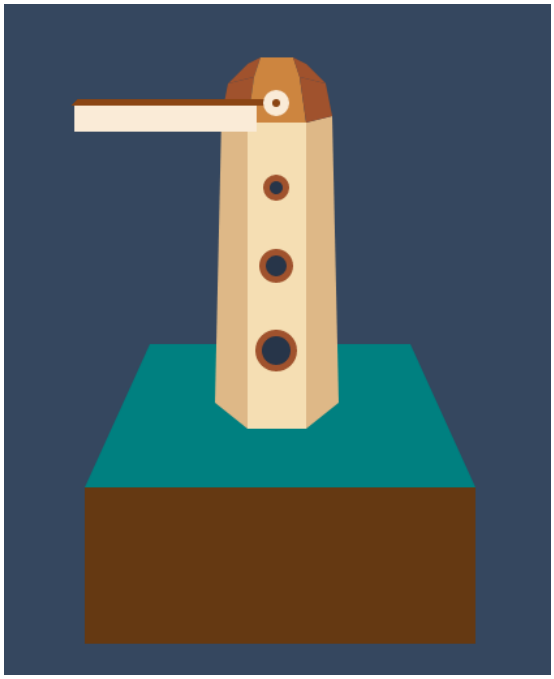
Atributima “cx” i “cy” oba kruga se pozicioniraju na prvi središnji četverokut kupole, X pozicija je ista kao i kod prozora, a Y pozicija je definirana u donjoj polovici elementa kupole.

Svako jedro vjetrenjače će se sastojati od dva elementa: platnenog, i drvenog nosivog dijela jedra. Drveni dio jedra će biti širok 150 točaka, a visok 5, dok će element koji predstavlja platno biti širok 140 točaka, a visok 20.
Kako će svi parovi elemenata biti jednaki, može se definirati jedan par kojeg će se zatim kopirati tri puta, a kasnije ih koristeći CSS transformacije razdvojiti jedne od drugih.

```
<rect class="mreza" x="192" y="187" width="140" height="20" fill="antiquewhite" />
<polygon points="340,187 190,187 195,182 340,182 " fill="saddlebrown" />

<rect class="mreza pr-1" x="192" y="187" width="140" height="20" fill="antiquewhite" />
<polygon class="pr-1" points="340,187 190,187 195,182 340,182 " fill="saddlebrown" />

<rect class="mreza pr-2" x="192" y="187" width="140" height="20" fill="antiquewhite" />
<polygon class="pr-2" points="340,187 190,187 195,182 340,182 " fill="saddlebrown" />

<rect class="mreza pr-3" x="192" y="187" width="140" height="20" fill="antiquewhite" />
<polygon class="pr-3" points="340,187 190,187 195,182 340,182 " fill="saddlebrown" />
```

Slika 13 – kod za jedra

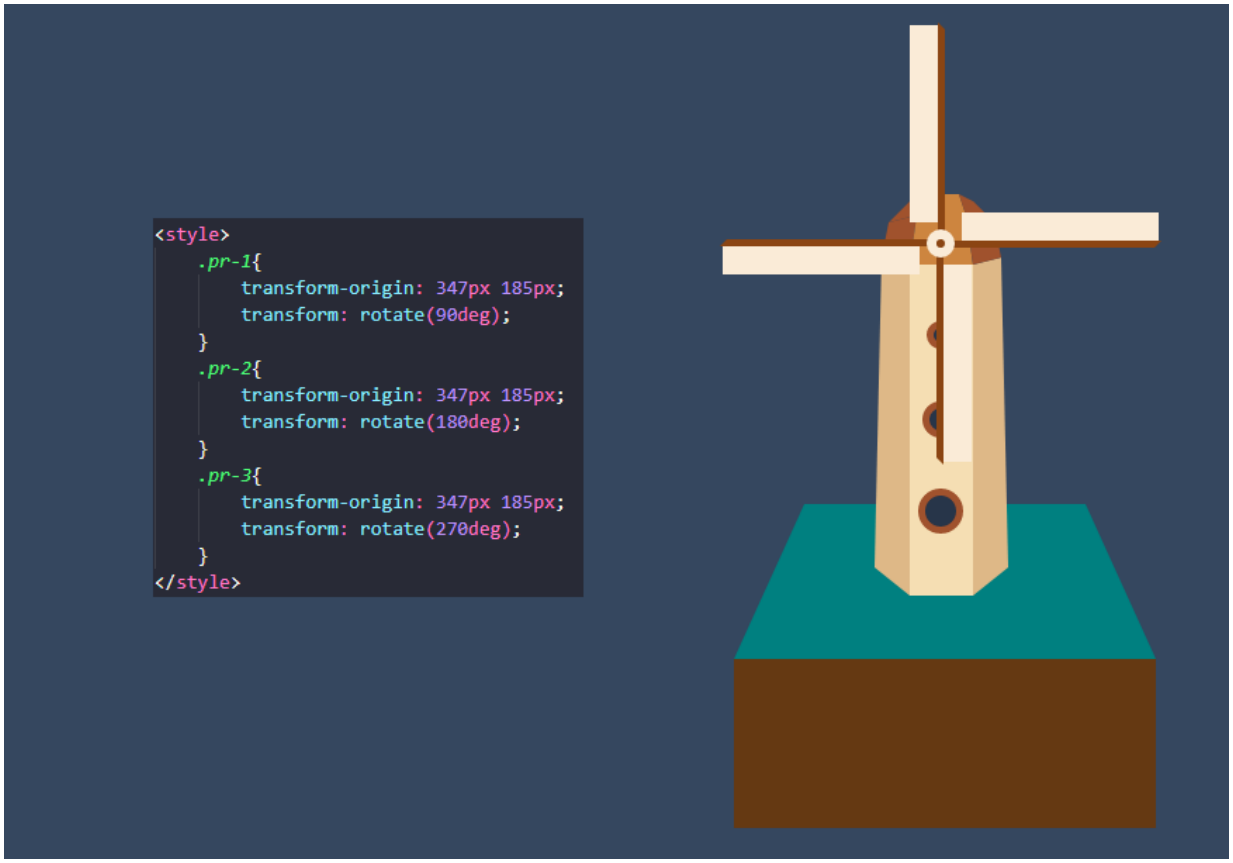
Koristeći oznaku <rect> definiran je platneni odnosno mrežasti dio jedara s ispunom *antiquewhite*, a oznakom <polygon> je definiran drveni dio ispune *saddlebrown*.



Slika 14 – rezultat definiranja i kopiranja jedara

Svaki od elemenata jedara na slici 14 se nalazi na istoj poziciji, no elementima koji su kopirani su zadane klase parova “.pr-1”, “.pr-2” i “.pr-3”.
Pozivajući ove klase u CSS-u, transformacijskim svojstvima se pozicija kopiranih parova može promijeniti na željenu.

Cilj je postaviti kopirana jedra oko rotora, a kako bi se postavilo mjesto oko kojeg se parovi rotiraju koristi se CSS svojstvo “transform-origin” za čije vrijednosti postavljamo X i Y točke rotora. Sama rotacija se izvodi koristeći svojstvo “transform: rotate(x);” gdje je “x” stupanj rotacije koji je za svaki par elemenata 90 stupnjeva viši od prijašnjeg.



Slika 15 – rezultat rotacije jedara CSS-om

Nakon izrade vjetrenjače, preostaje animirati rotor i jedra koristeći CSS “@keyframes” pravilo.

Animaciju je moguće lako provesti tako da ju se CSS-om postavi na cijelu skupinu klase “.rot” u kojoj se nalaze jedra, čime se postiže rotacija cijele skupine oko središnje točke koju predstavlja rotor.

Animacija je jednostavnog izgleda te se sastoji od početne vrijednosti rotacije 0 stupnjeva, i krajnje vrijednosti od 360 stupnjeva.

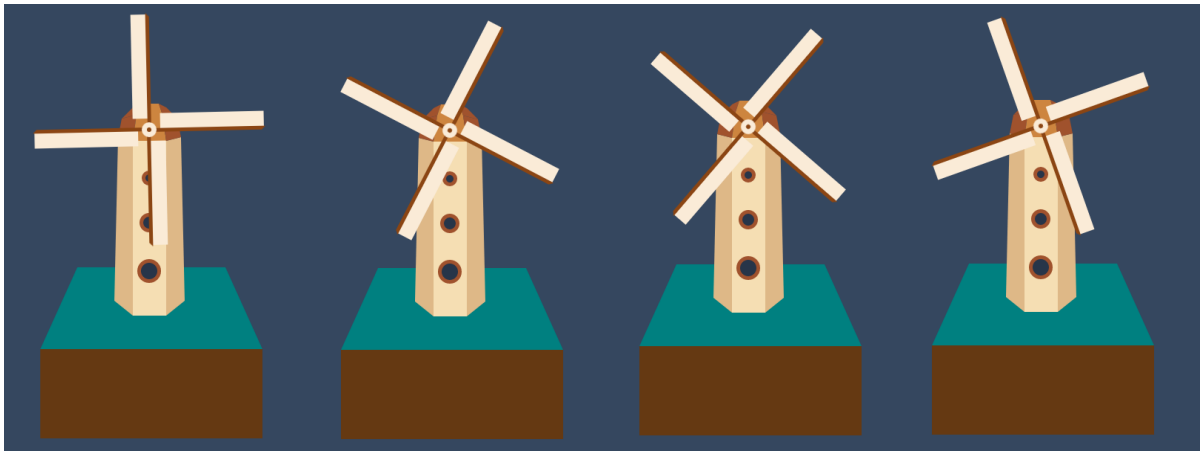
```css
.rot{
    transform-origin: 347px 185px;
    transform: rotate(0deg);
    animation-name:rotacija;
    animation-duration: 4s;
    animation-iteration-count: infinite;
    animation-direction: forwards;
    animation-timing-function: linear;
}

@keyframes rotacija{
    0%{transform:rotate(0deg)}
    100%{transform:rotate(360deg)}
}
```

Slika 16 – CSS kod za animaciju skupine ".rot"

Na slici 16 vidi se da je klasi ".rot" zadana točka izvora transformacije koja je jednaka središtu rotora čime se postiže rotacija skupine oko te točke, a početna vrijednost rotacije je postavljena na 0 stupnjeva.

Dalje se u klasi ".rot" opisuju svojstva animacije: naziv, trajanje, broj ponavljanja te smjer u kojem se odvija animacija. Zadnje svojstvo "animation-timing-function" se odnosi na ponašanje animacije – vrijednost *linear* znači da će se animacija odvijati jednako i linearno od početka do kraja, *ease-in* će animaciju na početku usporiti, *ease-out* na kraju, a *ease-in-out* će ju usporiti na početku i na kraju.

Sama animacija se definira pravilom "@keyframes" nakon kojeg se zadaje ime animacije "rotacija". Unutar vitičastih zagrada *keyframes* pravila prvo se određuju koraci animacije, u ovom slučaju su to 0% i 100%, tj. početak i kraj. Početna vrijednost transformacije je rotacija za 0 stupnjeva, a krajnja za 360 stupnjeva, čime animacija za vrijeme svog trajanja rotira skupinu za jedan cijeli krug.

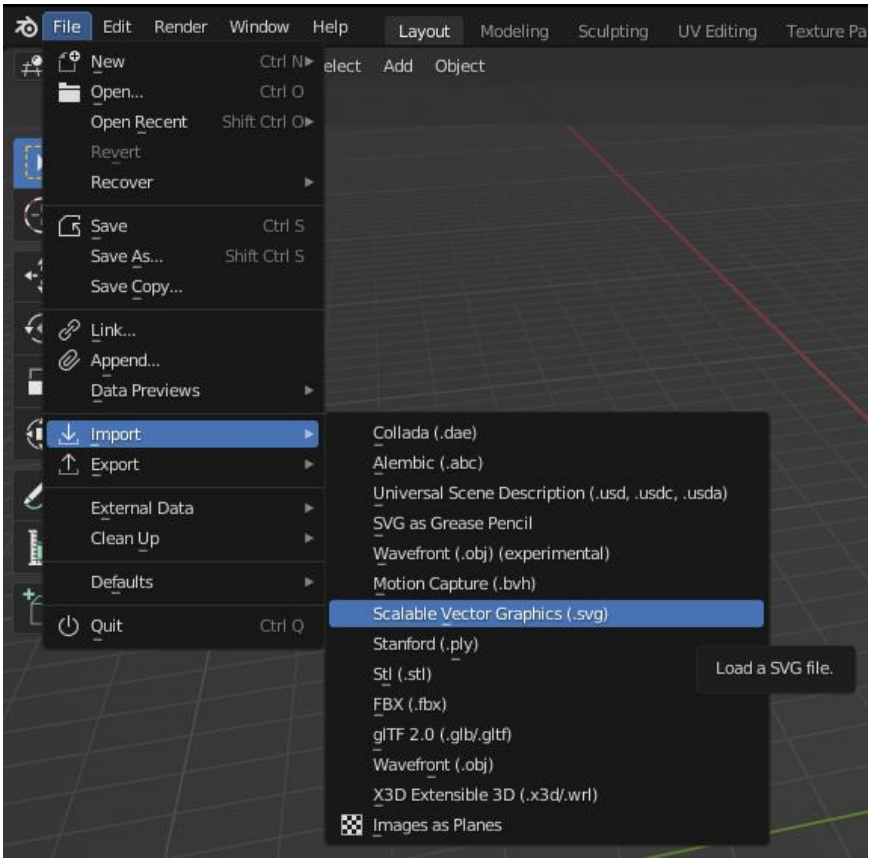


Slika 17 – prikaz dijela animacije

3.2. Izrada 3D modela koristeći 2D SVG

SVG dokument se za učitavanje treba pripremiti tako da iz HTML dokumenta izdvojimo SVG kod te ga spremimo kao vlastitu SVG datoteku. Kako bi se spriječilo moguće probleme, SVG datoteku je preporučivo prvo učitati u program za uređivanje vektorske grafike kao što su Adobe Illustrator ili Inkscape. Nakon učitavanja potrebno je ponovno spremiti datoteku kao vektorsku grafiku formata SVG te je dokument spreman za korištenje u Blenderu.

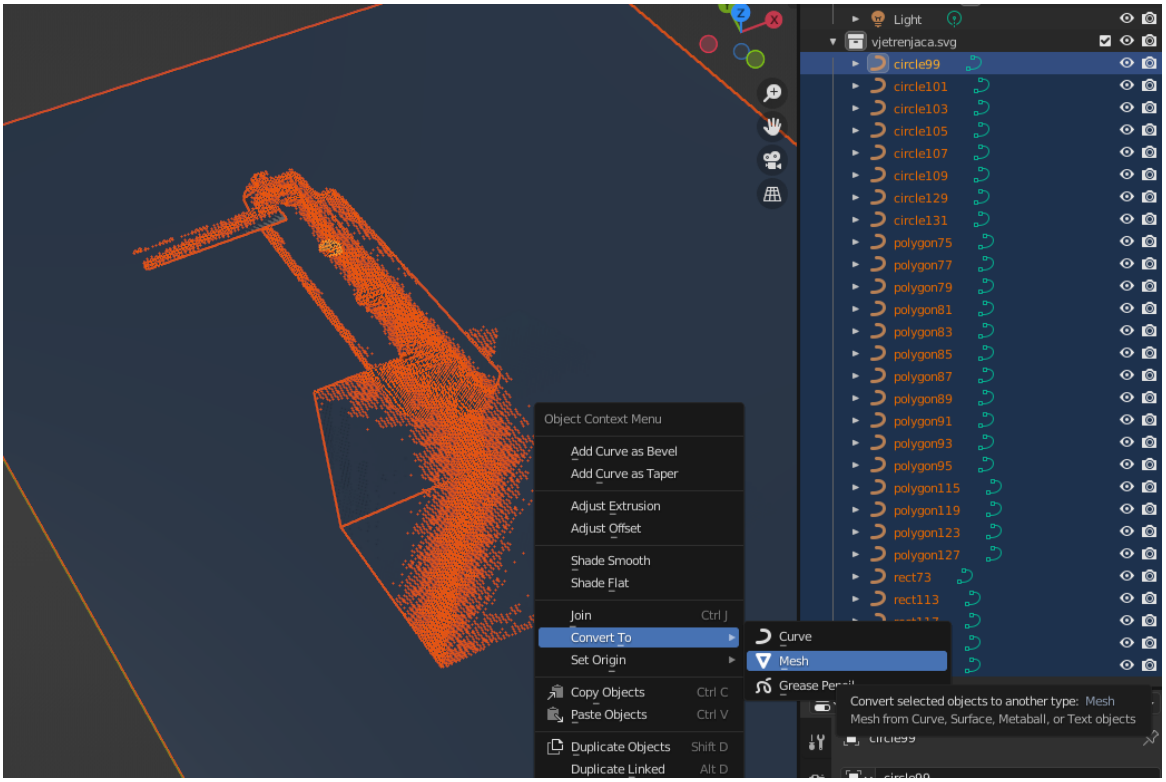
U programu Blender za učitavanje SVG-a potrebno je u alatnoj traci izabrati opciju file, zatim na padajućem izborniku odabrati import, nakon čega odaberemo Scalable Vector Graphics te željeni dokument.



Slika 18 - učitavanje SVG dokumenta

Elementi vektorske grafike u Blender su učitani kao dvodimenzionalne krivulje.

Kako se svi elementi grafike nalaze na istoj ravnini, u novom prostoru se preklapaju i treba ih razdvojiti. Za njihovu 3D manipulaciju prvo ih je potrebno pretvoriti u trodimenzionalne *mesh*-eve tako da se označe svi elementi te desnim klikom miša na sceni otvori izbornik na kojem se odabire opcija *Convert to – Mesh.*



Slika 19 – pretvaranje grafike u 3D objekte

Na slici 19 se nalazi izbornik u kojem se bira radnja za označenu grafiku, a iza izbornika se vidi označena grafika te preklapanje elemenata iste.

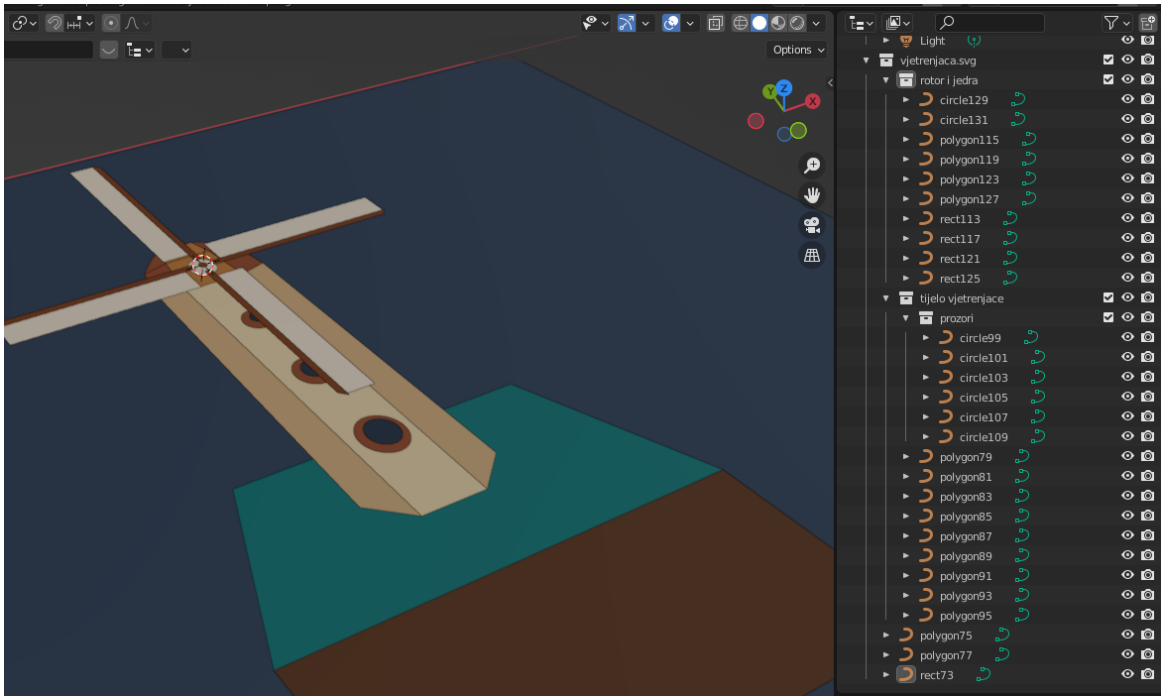
Nakon pretvorbe u objekte, elementi se pomiču jedan od drugog po novoj Z osi te se svakom od njih, koristeći opciju *Object – Set Origin – Origin to geometry,* središte ili izvor transformacija postavlja na sredinu geometrije objekta.

Objekte je također moguće grupirati u kolekcije za bolju preglednost, tijelo vjetrenjače će biti u jednoj kolekciji s jednom podgrupom u kojoj će se nalaziti prozori, a rotor i jedra u drugoj kolekciji.

Iz označenog dijela na slici 19 vidi se da vjetrenjača ima jedno jedro. Razlog tome je što su kod prikaza vektorske grafike na pregledniku jedra rotirana CSS-om, a CSS svojstva se ne prenose u Blender.

To znači da se sva jedra, kao što su definirana SVG-om, nalaze na istoj poziciji te ih je potrebno rotirati oko rotora.

Kao i kod CSS-a, ali u ovom slučaju koristeći 3D pokazivač, izvor transformacije jedara postavlja se na središte rotora te se svako jedro rotira za 90 stupnjeva više od prethodnog.



Slika 20 – razdvajanje elemenata po Z osi

Na slici 20 svi elementi grafike su razdvojeni i grupirani zbog preglednosti, no nije im finalizirana pretvorba u 3D *mesh.*

To je zato što je vektorska grafika rađena za 2D prikaz prividne trodimenzionalnosti, te je kako bi olakšali uređivanje preporučljivo “ispraviti” perspektivno distorzirane elemente. Ovim postupkom se elemente grafike nastoji prikazati kao pravilne oblike što će smanjiti broj novostvorenih točaka i linija – izvede li se postupak pretvorbe na npr. elementu kupole, prije ispravka oblika, stvorit će se linija koja dijagonalno dijeli četverokut na dva trokuta što može otežati njegovu manipulaciju.

Slika 21 - uspravno položena vjetrenjača s korigiranim tijelom

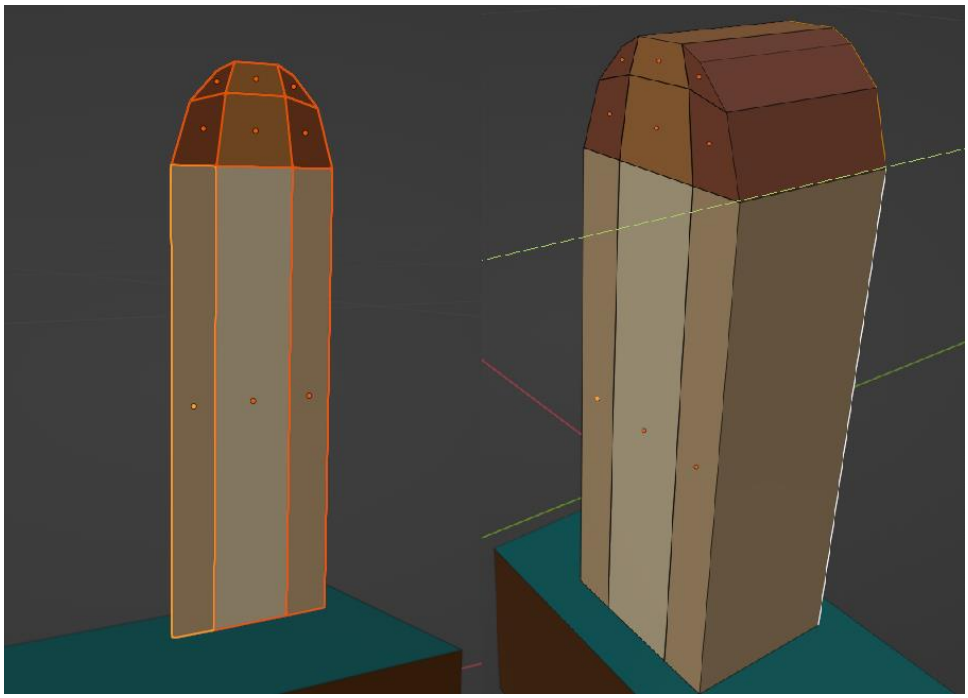
Nakon korigiranja tijela vjetrenjače, svi elementi se pretvaraju u *mesh* te ih se rotira u uspravnu poziciju.

Sljedeći korak se sastoji od istiskivanja svih elemenata kako bi dobili treću dimenziju. Istiskivanje se vrši u sučelju za uređivanje objekata tako da se označe željeni objekti, pritisne tipka tabulator za ulaz u *Edit Mode,* a zatim tipka "E" koja pokreće *Extrude* radnju. Objekte se može slobodno istiskivati po svakoj prostornoj osi, no pritiskom na tipke X, Y ili Z se istiskivanje vrši samo po toj osi.

Ako želimo istiskivanje vršiti na dvije osi, koristeći *shift* + (X/Y/Z) iz postupka se isključuje odabrana os.

Na dnu slike 21 je vidljiva X os (crveno) koja je paralelna vjetrenjači.

Kako bi istisnuli elemente u 3D prostor, *extrude* će se vršiti po Y osi tako da označimo elemente, a zatim toga iskoristimo kombinaciju tipki E-Y koja zaključava transformaciju na Y os.



Slika 22 – tijelo vjetrenjače prije i poslije istiskivanja

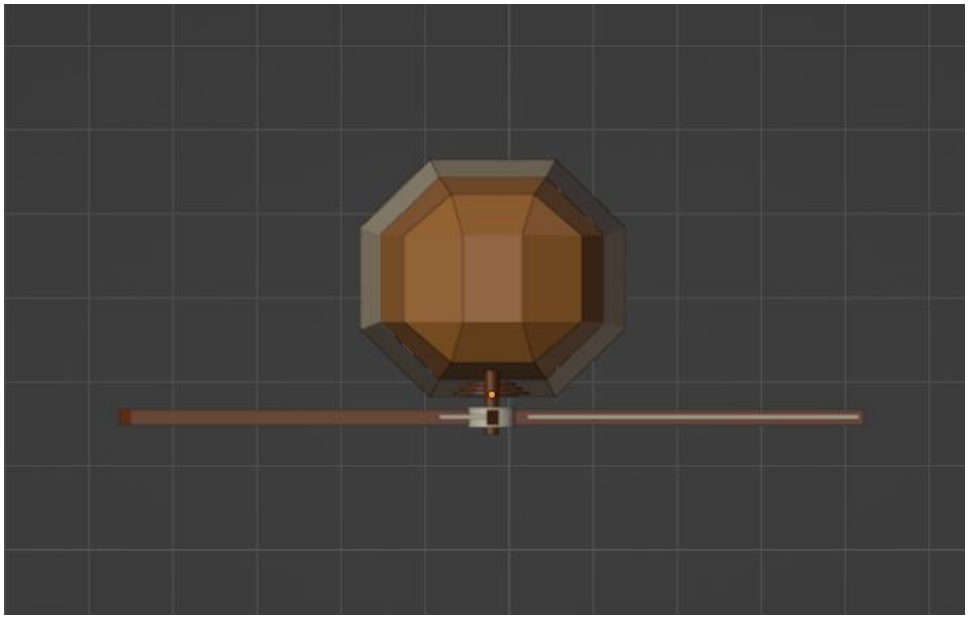
Slika 22 prikazuje označeno dvodimenzionalno tijelo vjetrenjače (lijevo) i rezultat istiskivanja (desno). U ovom prikazu nedostaju ostali elementi koji su prethodno sakriveni opcijom *hide* kako bi se transformacija mogla izvršiti bez blokiranja pregleda modela.

Idući korak se sastoji od istiskivanja ostalih elemenata vjetrenjače koristeći istu metodu. Svaki element, ili grupe jednakih elemenata se označuju te istiskuju do željenog rezultata.

Slika 23 - vjetrenjača nakon istiskivanja svih elemenata

Nakon izvršavanja *extrude* radnje nad svim elementima, na slici 23 se može vidjeti kako cjelokupni model poprima trodimenzionalni izgled. U idućem koraku se tijelo i kupola vjetrenjače modeliraju po zamišljenom osmerokutnom tlocrtu. Osim dimenzija, tijelu i kupoli se mijenja boja ispune. Originalno su elementi ispunjeni različitim bojama kako bi se postiglo prividno osvjetljenje, no kako Blender ima mogućnost postavljanja izvora svjetlosti svi elementi neke skupine se mogu ispuniti istom bojom.

Transformacija tlocrta u osmerokutni oblik se može izvršiti korištenjem *Grab* funkcije kojom je moguće po odabranoj osi pomicati točke, rubove ili cijelo lice lika, te funkcijom *Scale* kojom se odabrane točke, rub ili lice smanjuju ili povećavaju po odabranoj osi.



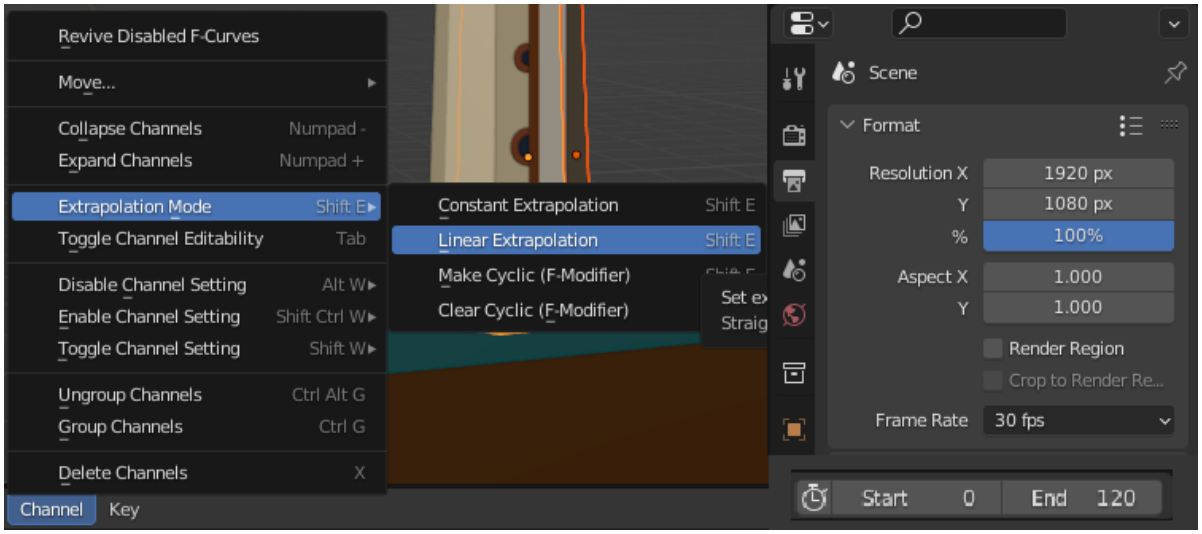
Slika 24 - transformacija stranica u osmerokutni tlocrt

U postupku transformacije tlocrta odabrane su strane tijela dobivene istiskivanjem, te su umanjene po Y osi do odgovarajućih dimenzija. Također je odabiranjem donjih lica tijela, baza vjetrenjače povećana po X i Y osi kako bi se zidove, kao i na vektorskoj grafici, ukosilo prema sredini vjetrenjače. Nakon ukošavanja zidova dolazi do preklapanja s prozorima te ih je potrebno po Y osi podesiti na odgovarajuće mjesto.

Slika 25 - dovršeni model vjetrenjače

Na slici 25 prikazani su svi modelirani dijelovi vjetrenjače, uključujući tlo na kojem se nalazi. Iz scene je uklonjen plavi četverokut koji je u originalnoj vektorskoj grafici predstavljao pozadinu, koja će u Blenderu biti definirana prostorno.

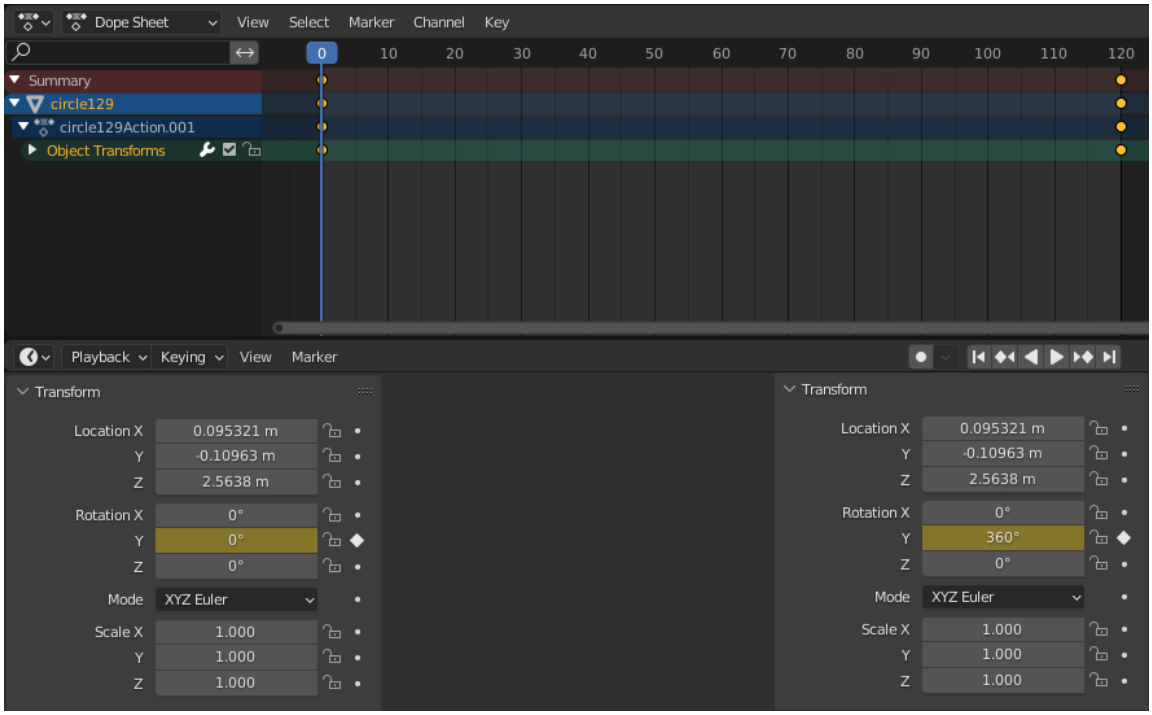
Idući korak ka finalizaciji modela je animiranje rotacije rotora i jedara, za koju se koristi *Animation* sučelje koje se nalazi u alatnoj traci iznad okvira scene.
Kako animacija uključuje rotaciju cijele skupine, pojedinačne modele skupine "rotor i jedra" moguće je spojiti u jedan konkretni model tako da ih se označi, te otvaranjem izbornika desnim klikom izabere opcija *Join.*



Slika 26 – postavke animacije

Slika 26 prikazuje postavke korištene za animaciju, a podešene su da odgovaraju animaciji izrađenoj CSS-om.

Opcija *Extrapolation Mode* se odnosi na način ponašanja animacije, a ekvivalent ovoj opciji je CSS svojstvo *animation-timing-function.* Odabirom *Linear Extrapolation* opcije, transformacija u animaciji će se odvijati jednoliko u svakom trenutku,



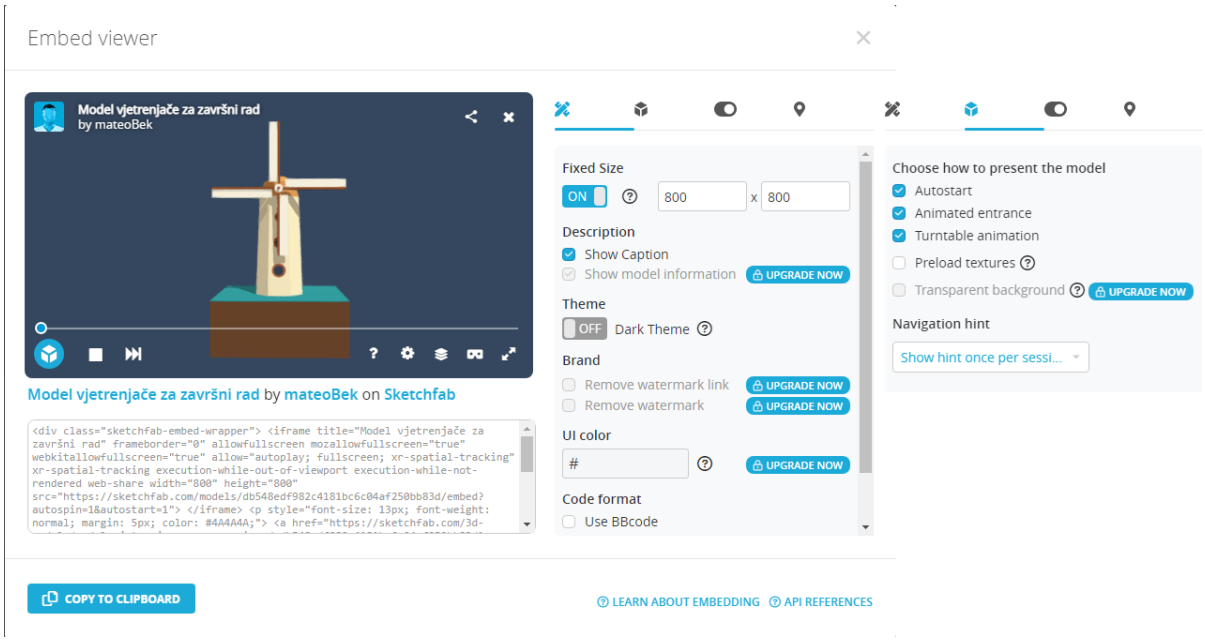
Slika 27 - ključni kadrovi animacije i vrijednosti transformacije

Na slici 27 je prikazana sekvenca animacije s dva ključna kadra. Odabrana je skupina spojenih elemenata modela koja sadržava rotor i jedra te im je u prvom ključnom kadru vrijednost rotacije oko Y osi postavljena na 0, a u drugom ključnom kadru na 360 stupnjeva.

Slika 28 – prikaz dijela animacije u Blenderu

Rezultat animiranja, prikazan na slici 28, odgovara animaciji vektorske grafike CSS-om te je time uspješna. Animaciju je također potrebno spremiti kao ".fbx" format te označiti opciju *Embed Textures*, a opciju *Path mode* podesiti na vrijednost *Copy*.
Model se na stranicu može učitati koristeći "Sketchfab" - web mjesto namijenjeno za dijeljenje 3D modela, koje nakon učitavanja modela i animacije nudi opciju ugrađivanja istih na drugu web stranicu koristeći <iframe> HTML oznaku.

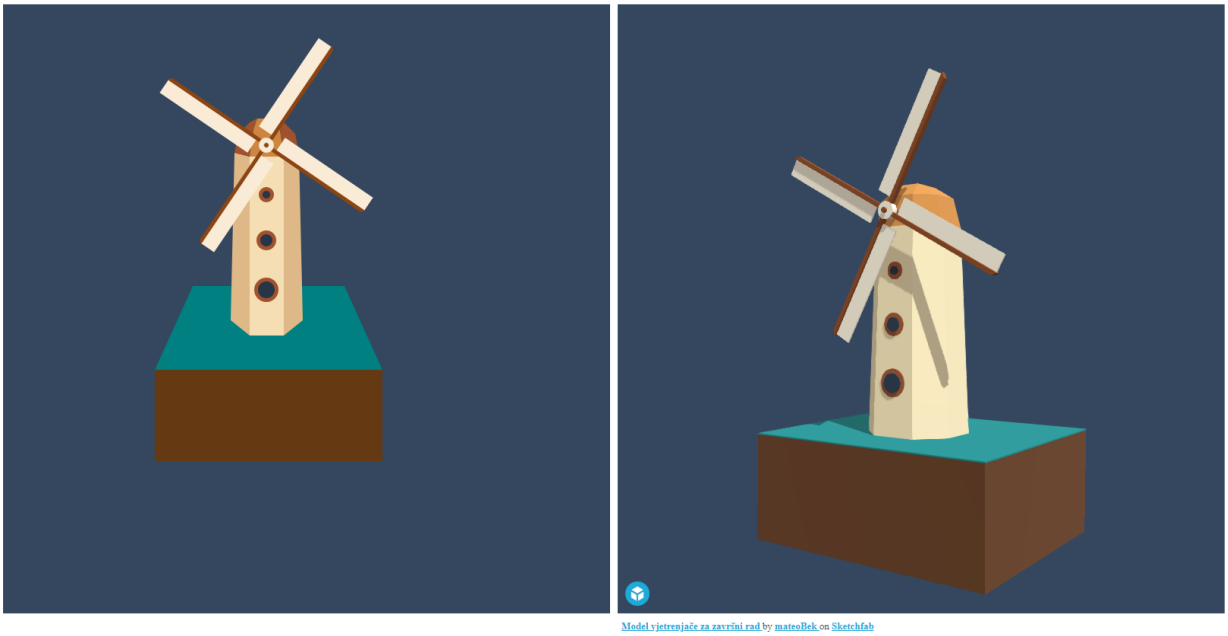
Nakon učitavanja modela i animacije, u korisničkom sučelju Sketchfab-a dovoljno je postaviti pozadinu, broj ponavljanja animacije te objaviti model na vlastiti korisnički profil. Sljedeći korak je podesiti postavke prikaza modela, a zatim kopiranje <iframe> oznake te unošenje iste u kod vlastite stranice.



Slika 29 – odabrane postavke za prikaz 3D objekta na stranici

Na slici 29 prikazane su odabrane postavke oznake za prikaz te su dimenzije prozora za prikaz izjednačene s dimenzijama SVG prozora za prikaz. U pregledu je također omogućena animacija rotacije cijelog modela oko Z osi, animacija učitavanja modela, te automatski početak elemenata animacije.

Nakon određivanja postavki, generirana <iframe> oznaka se kopira, a zatim lijepi u HTML dokument te se CSS-om pozicionira pored SVG prozora.



Slika 30 – prikaz vektorske grafike i trodimenzionalnog objekta u pregledniku

Slika 30 prikazuje vektorsku grafiku izrađenu CSS-om i HTML oznakom SVG (lijevo) i ugrađeni 3D model (desno) izrađen u Blenderu na temelju prethodno definirane grafike.

4. REZULTATI I RASPRAVA

Kako bi se kvalitetno usporedili rezultati prvo je potrebno analizirati i usporediti proces izrade oba modela, a zatim uzimajući to u obzir i krajnje rezultate.

Proces izrade animirane vektorske grafike je započeo s idejom koju je bilo potrebno vizualizirati i realizirati korištenjem HTML jezika. Svi elementi grafike su proizvoljno definirani s ciljem prikaza prividne trodimenzionalnosti, tako da baza vjetrenjače izgleda kao da je udaljena od promatrača dok se rotor i jedra nalaze na promatračevoj visini. Opisivanje svakog elementa strukture vjetrenjače je započelo s vizualizacijom, a zatim izračunom pozicija vrhova elementa s obzirom na "visinu" na kojoj se nalaze. Prozori su zamišljeni kao veći na dnu i manji na vrhu tijela neovisno o perspektivi. Izrada rotora i jedara, kao i njihova animacija bila je jednostavnija u odnosu na ostale elemente vjetrenjače.

Proces izrade 3D objekta te njegova animacija započet je učitavanjem izrađene grafike, koja je zatim optimizirana za manipulaciju u novom 3D prostoru. Svaku skupinu elemenata je bilo potrebno razdvojiti i pretvorbom definirati kao trodimenzionalni objekt. Nakon podešavanja grafike novi dvodimenzionalni 3D objekt se postavlja uspravno te mu se istiskivanjem daje nova dimenzija. Zatim su nove stranice smanjene na željenu širinu kako bi vjetrenjača bila jednolikog izgleda te se postigao željeni tlocrt osmerokutnog oblika. Proces animacije rotora je započeo postavljanjem jednakog izvora transformacije svim elementima, a zatim su odabrane postavke animacije koje odgovaraju prethodno izrađenoj CSS animaciji te je rotoru dodana transformacija između početnog i krajnjeg ključnog kadra.

Na slici rezultata (slika 30) vektorska grafika ima oštre i pravilne linije, dok su na prikazu 3D objekta vidljive nepravilnosti na rubovima modela. Boje na modelu su također svjetlije od boja vektorske grafike, no 3D prikaz sadrži izvor bijele svjetlosti.

5. ZAKLJUČAK

Najveći izazov u oba procesa je bio osmišljavanje i pravilno definiranje vektorske grafike korištenjem HTML jezika kako bi se postigao prividni efekt trodimenzionalnosti. Za postizanje prave trodimenzionalnosti korištenjem SVG-a potrebno je naprednije znanje te korištenje JavaScripta. Na temelju ovoga da se zaključiti kako bi izrada kompliciranije vektorske grafike s 3D efektom na više elemenata bila vremenski zahtjevan proces. Neovisno o tome, rezultat je grafika visoke kvalitete koja će se jednako prikazati na svim medijima neovisno o stupnju skaliranja. Izrada 3D objekta bila je jednostavnija iz razloga što je već postojeća grafika učitana u program te nije bilo potrebe za većim planiranjem.

Glavna razlika između dva procesa je pristup, odnosno način na koji se elementi transformiraju. Dok je SVG rađen tako da se prvo osmisli izgled, a zatim napiše kod i pogleda rezultat, 3D objekt je manipuliran u 3D okruženju s rezultatima vidljivim u stvarnom vremenu. Bilo je dovoljno označiti elemente koje želimo transformirati, a zatim ih odgovarajućim funkcijama programa istisnuti u prostor i modelirati po želji.

Animacija modela se pokazala kao poprilično izravan proces – na vremenskoj crti animacije se klikom odabere vrijeme, a zatim se klikom na *keyframe* ikonu pored željenog svojstva transformacije postavlja ključni kadar. Vrijednosti ključnih kadrova i postavke animacije su definirane kao i kod CSS animacije te je time animacija gotova i spremna za izvoz u obliku ".fbx" dokumenta. Ugrađivanje modela na stranicu se sastojalo od odabira nekoliko opcija prikaza, kopiranja <iframe> oznake te dodavanja par CSS svojstava za prikaz. Vektorska grafika namijenjena je za 2D prikaz te je time ograničena po pitanju mogućnosti 3D transformacija CSS-om.

Trodimenzionalni objekt prikazan je pikselima te je zbog toga manje kvalitete u odnosu na vektorsku grafiku, ali je zato interaktivan te ga je moguće okretati i promatrati iz različitih kuteva. Model također zadržava svoju kvalitetu neovisno o veličini prozora za prikaz kao i kod SVG-a, bez obzira na generalno nižu kvalitetu.
Problem kvalitete se može smanjiti prikazom objekta u većoj rezoluciji, te podešavanjem modela ili postavki prikaza, no to može rezultirati dužem vremenu učitavanja te većoj opterećenosti računala.

Dok je 3D prikaz vektorskom grafikom kvalitetniji, prikaz 3D objekta <iframe> oznakom je realističniji i interaktivniji. Korištenje jedne metode ne treba isključivati drugu, već se mogu međusobno nadopunjavati kako bi se postigli bolji rezultati i nove mogućnosti.

6. LITERATURA

1. Little Chantelle: The History of Web Design, Tiller, 28.9.2021. [https://tillerdigital.com/blog/the-history-of-web-design/], pristupljeno 12.6.2022.
2. Juviler Jamie: SVG Files: What They Are and How to Make One, HubSpot, 2.11.2021 [https://blog.hubspot.com/website/what-is-an-svg-file], pristupljeno 17.6.2022.
3. Libby Alex: Beginning SVG: A Practical Introduction to SVG using Real-World Examples, 2018
4. Više Autora: SVG: Scalable Vector Graphics, Mozilla Developer Network, uređeno 28.7.2022. [https://developer.mozilla.org/en-US/docs/Web/SVG], pristupljeno 25.6.2022..
5. Nepoznat Autor: What is 3D Modelling and What is it Used For?, FutureLearn, 18.3.2022. [https://www.futurelearn.com/info/blog/general/what-is-3d-modelling], pristupljeno 1.7. 2022.
6. Blain M. John: The Complete Guide to Blender Graphics, Computer Modeling & Animation, 2021.

7. PRILOZI