CAPITOLUL 3

Utilizarea obiectelor 3D în scene statice

3.1. Fișiere .OBJ

Formatul de fișier OBJ este unul dintre cele mai importante formate de fișiere în aplicațiile de imprimare 3D și grafică 3D. Este formatul preferat pentru imprimarea 3D multicolor și este utilizat pe scară largă ca format de schimb neutru pentru modelele 3D non-animate în aplicațiile grafice.

Un fișier OBJ este un format standard de imagine 3D care poate fi exportat și deschis de diverse programe de editare 3D. Acesta conține un obiect tridimensional, care include coordonatele 3D, hărțile de texturi, fețele poligonale și alte informații despre obiect.

Pe scurt, formatul de fișier OBJ stochează informații despre modelele 3D. Poate codifica geometria suprafeței unui model 3D și poate stoca, de asemenea, informații despre culoare și textură. Acest format nu stochează nicio informație despre scenă cum ar fi poziția luminii sau animații.

Un fișier OBJ este de obicei generat de un software CAD (Computer Aided Design) ca produs final al procesului de modelare 3D. Extensia de fișier corespunzătoare formatului de fișier OBJ este pur și simplu „.OBJ”.

Formatul de fișier OBJ este open source și neutru. Este adesea utilizat pentru partajarea modelelor 3D în aplicații grafice, deoarece se bucură de un bun suport de import și export din aproape toate software-urile CAD. Acesta este de asemenea utilizat ca format de fișier pentru imprimarea 3D multicolor, deoarece formatul standard de imprimare 3D STL nu include informații despre culoare și textură.

Formatul de fișier OBJ a fost creat inițial de Wavefront Technologies pentru aplicația sa Advanced Visualizer pentru a stoca obiecte geometrice compuse din linii, poligoane și curbe și suprafețe de formă liberă. Cea mai recentă versiune documentată este v3.0, înlocuind versiunea anterioară v2.11.

Cel mai dominant format din lumea tipăririi 3D este STL. Cu toate acestea, STL este un format de fișier vechi care, deși este foarte popular, nu a ținut cont de cerințele moderne. Exactitarea imprimării 3D se apropie rapid de rezoluția la nivel de micron și modelele multicolore devin din ce în ce mai populare. Formatul de fișier STL nu poate suporta destul de bine rezoluțiile mari, deoarece o rezoluție mai mare duce la creșterea dimensiunii fișierului. De asemenea, formatul STL nu este potrivit pentru imprimarea 3D multicolor, deoarece nu suportă informații despre culoare și textură. În schimb, formatul OBJ poate aproxima geometria suprafeței destul de precis, fără a avea un impact semnificativ asupra dimensiuneii fișierului. Acest lucru este posibil folosind curbele Bezier și o metodă numită NURBS. În plus, formatul de fișier OBJ are suport nativ pentru mai multe culori și texturi din același model.

Astfel, utilizarea formatului OBJ are o serie de avantaje în comparație cu utilizarea formatului STL în cazul în care aveți nevoie de modele cu rezoluție înaltă, multi-color. Pe de altă parte, formatul de fișier OBJ nu este la fel de universal ca formatul STL. Aproape toate imprimantele 3D acceptă formatul STL. Nu se poate spune același lucru și despre formatul OBJ, chiar dacă se bucură de suport. Prin urmare, dacă este nevoie de a tipări un model 3D monocrom la o imprimantă standard, este preferabil formatul STL.

Ambele formate OBJ și STL au un spectru larg de utilizare, cu o bază mare de utilizatori fideli și se bucură de suportul unei mulțimi de aplicații terțe.

Concurenții principali în cazul formatelor de fișiere pentru imprimare 3D sunt VRML, AMF și 3MF, care însă nu se bucură de un așa support și compatibilitate și deci nu sunt alternative serioase ale formatelor de fișiere STL și OBJ.

Format fișier OBJ se utilizează în aplicații grafice 3D. Cele mai utilizate formate de fișiere în aplicațiile 3D Graphics sunt OBJ, FBX și COLLADA.

Diferențe între formatele de fișiere OBJ, FBX, COLLADA

Cea mai importantă diferență între formatul de fișier OBJ și celelalte două (FBX și COLLADA) este suportul pentru informații despre scenă cum ar fi sursele de lumină și animații. Formatul de fișier OBJ nu conține informații despre scene și animații, în timp ce FBX și COLLADA o fac. Prin urmare, dacă se dorește întreținerea suportului pentru animații în cazul jocurilor sau filmurilor, atunci se recomandă utilizarea formatului FBX sau COLLADA. În cazul în care însă nu este nevoie de o scena complexă sau de animații, poate fi argumentată utilizarea formatului de fișier OBJ.

Avantajele formatului obj

În primul rând, formatul de fișier OBJ este un format simplu și deschis. Are suport larg pentru export și import printre software CAD. Aceasta înseamnă că dacă partajați modelul 3D ca fișier OBJ, atunci alte programe CAD îl vor interpreta corect și consecvent. Nu se poate spune același lucru despre formatele FBX sau COLLADA. Formatul COLLADA este, de asemenea, open source, dar este destul de complicat. Diferite software CAD îl interpretează diferit și acest lucru poate duce la erori.

Formatul FBX este un format închis și proprietar și oferă un SDK pentru conversia formatelor existente în FBX. Convertirea unui fișier FBX într-un alt format însă se realizează destul de complicat și poate fi însoțită de apariția erorilor.

Un fișier OBJ va fi, mult mai simplu și de dimensiuni reduse comparativ cu un fișier FBX sau COLLADA care conțin același model 3D. Acest lucru se datorează simplității formatului de fișier OBJ în comparație cu celelalte specificații și datorită codificării sale binare native.

Astfel, dacă nu este nevoie de o scenă complexă sau de animații și dar mult mai mlt contează suportul și interpretare corectă de diferite software CAD, formatul de fișier OBJ este formatul potrivit. În aproape toate celelalte cazuri, FBX este formatul optim pentru aplicațiile grafice 3D.

Caracteristici moderne a formatului de fișiere obj

În ceea ce privește caracteristicile moderne, formatul FBX este cel mai progresiv format care oferă o mulțime de caracteristici de ultimă generație, actualizări și îmbunătățiri regulate. Formatul de fișier OBJ este al doilea în ceea ce privește caracteristicile, în timp ce formatul COLLADA nu se bucură de actualizări și îmbunătățiri regulate.

Geometrie

Scopul principal al formatului de fișier OBJ este de a codifica geometria suprafeței unui obiect 3D. Formatul de fișier OBJ este destul de versatil în acest sens. Permite mai multe opțiuni pentru codarea geometriei suprafeței. Fiecare dintre cele trei metode permise descrise mai jos au propriile avantaje și dezavantaje.

Teselare cu fețe poligonale

În forma sa cea mai simplă, formatul de fișier OBJ permite utilizatorului să teseleze suprafața modelului 3D cu forme geometrice simple precum triunghiuri, patrulatere sau poligoane mai complexe. Vârfurile poligoanelor și normala fiecărui poligon sunt apoi stocate într-un fișier care conține geometria suprafeței modelului.

Teselările cu fețe poligonale au o serie de avantaje și dezavantaje. Poligoanele sunt forme geometrice simple și această metodă este de fapt cel mai simplu mod de a descrie geometria suprafeței. Cu toate acestea, aproximarea unei suprafețe curbate cu poligoane duce la modificarea grosimii modelului.

În cazul tipăririi 3D, imprimanta 3D va imprima obiectul cu aceeași grosime specificată de fișier. Desigur, făcând triunghiurile din ce în ce mai mici, aproximarea poate fi făcută din ce în ce mai exact, rezultând imprimări de o calitate mare. Cu toate acestea, pe măsură ce se micșorează dimensiunea triunghiului, crește și numărul de triunghiuri necesare pentru a descrie suprafața. Acest fapt cauzează creșterea semnificativă a dimensiunii fișierului, problemă care încearcă să o soluționeze slicer-ele de imprimare 3D. De asemenea, devine dificilă distribuiția sau gestionarea unor astfel de fișiere uriașe.

Prin urmare, este foarte important să se găsească echilibrul corect între dimensiunea fișierului și calitatea imprimării. Nu are sens să se reducă dimensiunile triunghiurilor la infinit, deoarece la un moment dat nu se va mai putea distinge cu ochiul liber diferența între calitățile tipăririi.

Extensia de fișier .obj se referă în primul rând la Wavefront 3D (.obj) dezvoltat de Wavefront Technologies pentru software-ul Advanced Visualizer. Formatul OBJ este un format textual pentru descrierea geometriei corpurilor tridimensionale care permite modelarea formelor volumetrice complexe și aplicarea diverselor materiale și texturi. Pe lângă fișierul .OBJ, un obiect sau scenă tipică Wavefront 3D va include, de obicei, unul sau mai multe fișiere Material Template Library (.mtl), care definesc materialele obiectului cu referințe la texturi de tip bitmap externe, de obicei stocate într-un subdirector separat.

Formatul OBJ a devenit unul dintre cele mai populare și acceptate formate de modele 3D și funcțiile de export/import de fișiere .OBJ sunt prezente în aproape fiecare editor 3D. Destul de multe utilitare pentru vizualizarea modelelor 3D (există chiar și aplicații web care permit importul, vizualizarea, editarea, exportul și conversia fișierelor .OBJ) oferă posibilitatea de a deschide fișiere .OBJ și a afișa modelele conținute cu redare completă, iar o serie de convertoare permit conversia modelelor OBJ în alte formate. Există colecții întregi și biblioteci de modele în acest format pe Internet. Formatul de geometrie OBJ este un format deschis pentru fișierele de descriere a geometriei.

Specificație OBJ

Formatul de fișier OBJ este un format de fișier ASCII. Editarea acestuia poate fi realizată cu ajutorul oricărui editor de text.

Specificația originală nu indică explicit care ar trebui să fie caracterul de sfârșit de linie, așa că unele software-uri folosesc \r retur de car (CR) și altele folosesc \n linie nouă (NL sau LF). Este posibil să fie necesară conversia caracterelor care indică sfârșitul de linie dacă editorul dvs. de text sau software-ul utilizat nu reușesc să citească fișierul.

Primul caracter al fiecărei linii este foarte important deoarece acesta specifică tipul de comandă. Dacă primul caracter este #, acea linie reprezintă un comentariu și orice altceva din acea linie este ignorat. Liniile goale sunt, de asemenea, ignorate.

Comanda comentariu

o linie de comentarii

După cum s-a menționat anterior, caracterul **#** indică faptul că linia reprezintă un comentariu și ar trebui ignorată. Prima linie este de obicei întotdeauna un comentariu care specifică ce program a generat fișierul.

Comanda vîrf

v x y z

În această versiune simplificată a specificației, vor fi cuprinse doar fețele poligonale, prin urmare comanda **vertex – v** poate fi utilizată pentru a specifica vârfurile poligonului. Când se utilizează suprafețe și curbe de formă liberă, există o comandă similară **vp** care poate fi utilizată pentru a specifica punctele de control ale suprafeței sau curbei.

Comanda **vertex – v** specifică un vârf prin cele trei coordonate carteziene **x**, **y** și **z**. Vârfului *i* se atribuie automat un nume în funcție de ordinea în care este găsit în fișier. Primul vârf din fișier primește numele „1”, al doilea ca „2”, al treilea ca „3” și așa mai departe.

Comanda normala la vîrf

vn x y z

Vertex normal – vn este comanda pentru normala unui vârf. Specifică vectorul normal la suprafață. Parametrii **x**, **y** și **z** sunt componentele vectorului normal. Acest vector normal nu este încă asociat niciunui vârf. Acesta va trebui să fie asociat ulterior unui vârf cu ajutorul unei alte comenzi numită comanda **f**.

Comanda vertex normal poate fi omisă într-o mulțime de fișiere, deoarece atunci când se vor grupa vârfurile în fețe poligonale cu comanda **f**, se va determina automat vectorul normal din coordonatele vârfului și ordinea în care vor apărea vârfurile.

Comanda textură a vîrfului

vt u v [w]

Comanda **vertex texture – vt** specifică un punct din harta texturilor. Parametrii **u** și **v** sunt coordonatele **x** și **y** din harta texturii. Acestea vor fi numere în virgulă mobilă între 0 și 1. Acesteale trebuie asociate unui vârf cu ajutorul comenzii **f**, analogic normalelor la vârfuri.

Comanda suprafață

f v1 [/ vt1] [/ vn1] v2 [/ vt2] [/ vn2] v3 [/ vt3] [/ vn3] ...

Comanda **suprafață – f** este probabil cea mai importantă comandă. Specifică o suprafață poligonală creată din vârfurile specificate anterior.

Pentru a face referire la un vârf, trebuie doar urmat sistemul de numerotare implicit al vârfurilor. De exemplu, „f 23 24 25 27” descrie o față poligonală construită din vârfurile 23, 24, 25 și 27 în ordine.

Pentru fiecare vârf, se poate asocia o comandă **vn**, care apoi asociază acel normala cu vârful corespunzător. În mod similar, poate fi asocită o comandă **vt** cu un vârf, care va determina maparea texturii utilizate pentru acest vârf.

Dacă se indică textura sau normala pentru un vârf, atunci acestea trebuie specificate pentru toate.

Comanda grup

g nume

Comanda **grup – g** indică gruparea unei părți a obiectului într-un grup cu numele indicat ca parametru. Toate comenzile **f** care urmează vor fi în incluse în același grup. Acest lucru este util dacă se dorește reutilizarea unei anumite informații, cum ar fi tipul de material, pentru o parte selectată a unui obiect.

Comanda material utilizat

nume usemtl

Comanda de **utilizare a materialului – usemtl** permite indicarea materialului utilizat. Toate comenzile **f** care vor urma vor folosi același material până nu apare o altă comandă **usemtl**.

3.2 Biblioteca de materiale

Informația despre aspectul obiectelor (materialelor) conținute în fișierul OBJ se conține în fișiere separate în format MTL (Material Library). Fișierul OBJ face referință la un astfel de fișier, dacă este necesar, folosind directiva:

Mtllib [numele fișierului extern MTL]

MTL este un standard stabilit de Wavefront Technologies. Toate informațiile sunt prezentate în formă ASCII și sunt lizibile pentru om. Standardul MTL este, de asemenea, foarte popular și este acceptat de majoritatea pachetelor grafice 3D.

Informațiile despre materialele simple din fișier arată astfel:

Newmtl material_name1

Anunțarea următorului material

Culori Ka 1.000 1.000 0.000

Culoare lumină ambientală (galben) Kd 1.000 1.000 1.000

Culoare difuză (alb)

Parametri de reflecție Ks 0.000 0.000 0.000

Culoare reflexie specular (0; 0; 0 - oprit) Ns 10.000

Coeficient de reflecție specular (de la 0 la 1000)

Parametri de transparență d 0.9

Transparența este specificată folosind directiva d Tr 0.9

sau în alte implementări de format folosind Tr

Următorul material newmtl material_name2 ...

Toți parametrii sunt opționali. În absența oricărui parametru, programul îl setează automat în mod implicit.

Avînd în verere toate cele expuse anterior, mai jos este prezentat conținutul unui fișier OBJ simplu (CUB) construit în editorul 3D Google SketchUp, Fig 3.1 și exportat ca fițier .OBJ cu ajutorul plugin-ului *LIPID Lightsolve* care poate fi ușor găsit și instalat din meniul *Window>Extemsion Warehouse*. Pentru a avea posibilitate de a instala oricare plugin din biblioteca de extensii utilizatorul trebuie să dispună de un account Google și să fie autentificat.

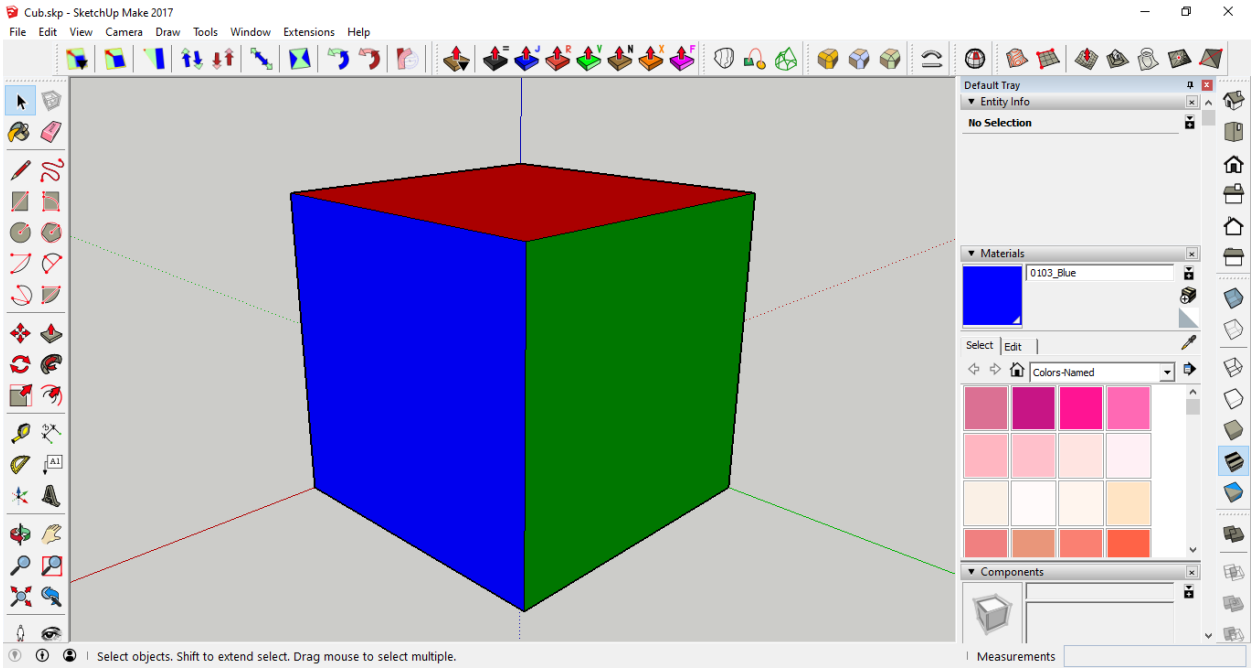


Fig. 3.1 Modelul 3D a unui cub construit în editorul 3D Google SketchUp.

Pentru a exporta modelul 3D al cubului în formatul OBJ se accesează: File-> LIPID OBJ Exporter, Fig3.2, apoi în fereastra LIPIDOBJ se bifează opțiunea Ignore Back face.

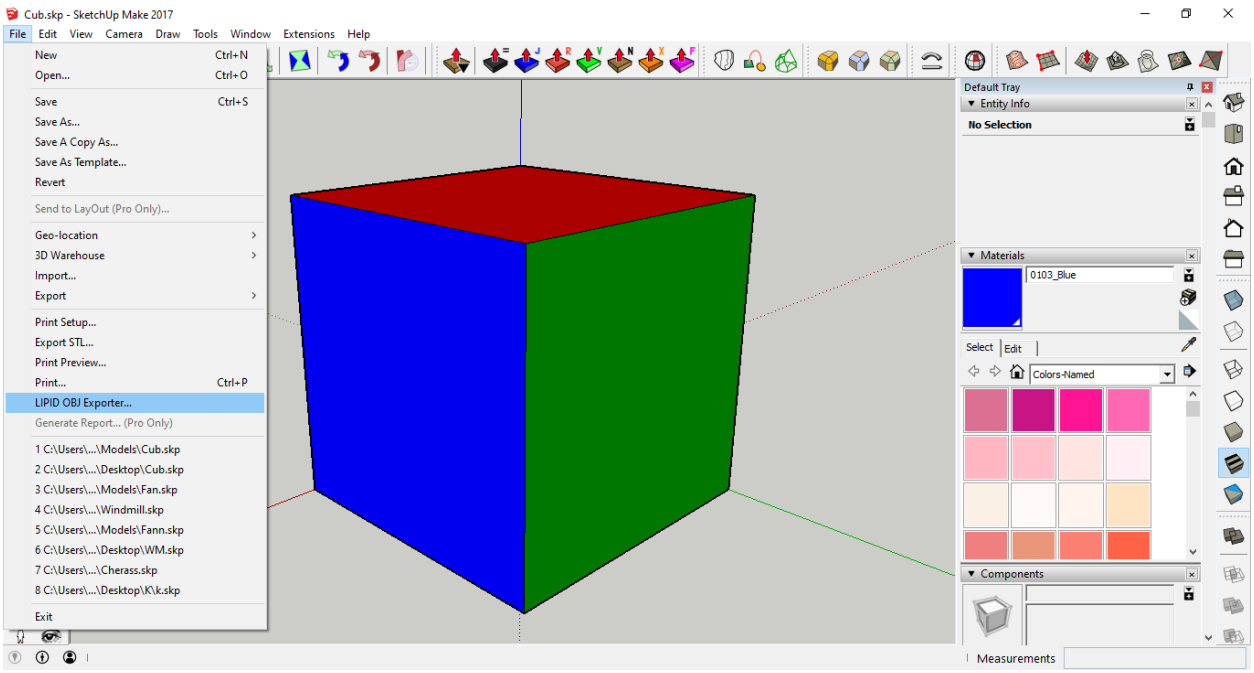


Fig. 3.2 Exportarea modelul 3D al cubului în formatul OBJ cu ajutorul plugin-ului LIPIDOBJ.

Conținutul fișierului .OBJ a modelului 3D a cubului prezentat în Fig. 3.2:

mtllib /Cub.mtl

s 1

o

g

usemtl 0020_Red

v 100 100 0

v 0 0 0

v 0 100 0

vn 0 0 -1

f 1//1 2//1 3//1

v 100 0 0

f 2//1 1//1 4//1

v 100 0 100

v 0 100 100

v 0 0 100

vn 0 0 1

f 5//2 6//2 7//2

v 100 100 100

f 6//2 5//2 8//2

usemtl 0076_Green

vn 0 -1 0

f 5//3 2//3 4//3

f 2//3 5//3 7//3

vn 0 1 0

f 6//4 1//4 3//4

f 1//4 6//4 8//4

usemtl 0103_Blue

vn -1 0 0

f 6//5 2//5 7//5

f 2//5 6//5 3//5

vn 1 0 0

f 1//6 5//6 4//6

f 5//6 1//6 8//6

Analizînd conținutul fișierului .OBJ prezentat anterior se poate observa faptul că în fișierul Cub.obj este utilizat fișerul de materiale Cub.mtl care se află în același. Pe lîngă aceasta mai poate fi observată descrierea coordonatelor tuturor celor 8 vîrfuri ale cubului, indicarea suprafețelor (triunghiurilor) care alcătuiesc cele 6 fețe ale cubului cu specificarea normalelor acestora și atribuirea texturilor *0020_Red, 0076_Green* și *0103_Blue* suprafețelor (triunghiurilor) corespunzătoare.

Conținutul fișierului materialelor utilizate pentru texturarea modelului 3D a cubului prezentat în Fig. 3.2.

#

Alias MTL Material File

 # Converted from SKP by LIPID Lightsolve

newmtl 0020_Red

Ka 0.000000 0.000000 0.000000

Kd 1 0 0

Ks 0.330000 0.330000 0.330000

newmtl 0076_Green

Ka 0.000000 0.000000 0.000000

Kd 0 0.501961 0

Ks 0.330000 0.330000 0.330000

newmtl 0103_Blue

Ka 0.000000 0.000000 0.000000

Kd 0 0 1

Ks 0.330000 0.330000 0.330000

newmtl default_material

Ka 0.000000 0.000000 0.000000

Kd 0.330000 0.330000 0.330000

Ks 0.330000 0.330000 0.330000

Analizînd conținutul fișierului .MTL prezentat anterior poate fi observată definirea texturilor *0020_Red, 0076_Green* și *0103_Blue* utilizate în fișierul Cub.obj pentru texturarea fețelor modelului cubului.

Descrierea obiectelor și metodelor principale pentru lucrul cu fișiere .OBJ cu ajutorul librăriei pentru realizarea graficii WEB treidimensionale în editorul online p5.js.

Biblioteca p5.js este o bibliotecă JavaScript care ușurează procesul de dezvoltare a programelor ce utilizează elemente grafice 2D și 3D și este destinată atît pentru dezvoltatorii experimentați cît și pentru începători. Biblioteca p5.js este gratuită și open-source.

Biblioteca p5.js are un set complet de funcționalități grafice. Cu ajutorul acestei biblioteci grafice întreaga pagină a browserului este privită ca spațiu de lucru în care pot fi utilizate inclusiv obiecte HTML5 pentru text, video, cameră web și sunet.

Funcțiile de bază:

Încărcarea unui model 3D dintr-un fișier OBJ sau STL.

loadModel() trebuie plasat în interiorul funcției ***preload()***. Acest lucru permite modelului să se încarce complet înainte de a rula restul programului.

Una dintre limitările formatului OBJ și STL este faptul că nu ține cont de scară. Aceasta înseamnă că modelele exportate din diferite programe pot avea dimensiuni diferite. Dacă modelul nu se afișează, încercați să apelați ***loadModel()*** cu parametrul de normalizare setat la true. Aceasta va redimensiona modelul la o scară adecvată pentru p5. De asemenea, se pot face modificări suplimentare ale dimensiunii modelului cu funcția ***scale()***.

De asemenea, nu este realizat suportul pentru fișiere colorate STL. Fișierele STL colorate vor fi redate fără proprietăți de culoare.

Sintaxa:

loadModel (path, normalize, [successCallback], [failureCallback], [fileType])

loadModel (path, [successCallback], [failureCallback], [fileType])

Parametri:

path String: Calea modelului încărcat

normalize boolean: dacă este adevărat, se scalează modelul la o dimensiune standardă la încărcare

funcția successCallback Function(p5.Geometry): funcție care trebuie apelată după încărcarea modelului. Va fi returnat obiectul de tip 3D model. (Opțional)

funcția failureCallback Function(Event): apelat cu eveniment de eroare dacă modelul nu reușește să se încarce. (Opțional)

fileType String: extensia fișierului modelului (.stl, .obj). (Opțional)

Se intoarce

p5.Geometry: obiect de tip p5.Geometry

model()

Redă un model 3D pe ecran.

Sintaxa:

model(model)

Parametri:

model p5.Geometrie: Model 3D încărcat care trebuie redat

Sketch Files->Upload file

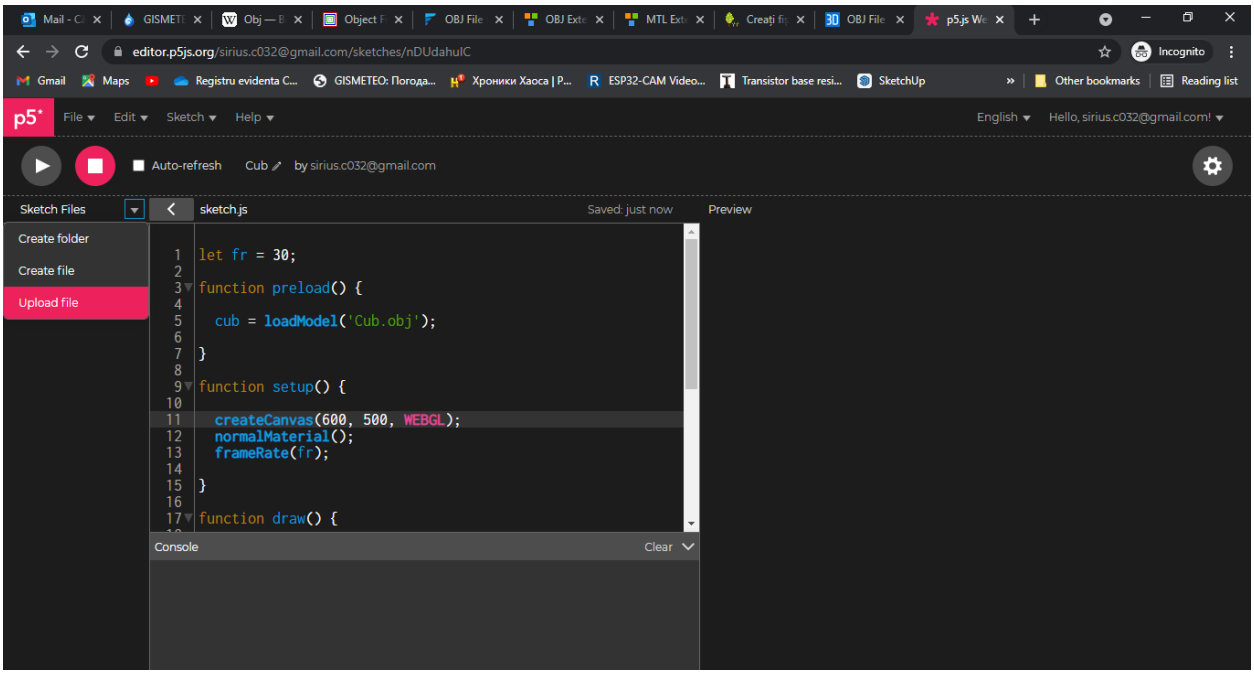


Fig. 3.3 Încarcarea fișierului Cub.obj în directoriul proiectului.

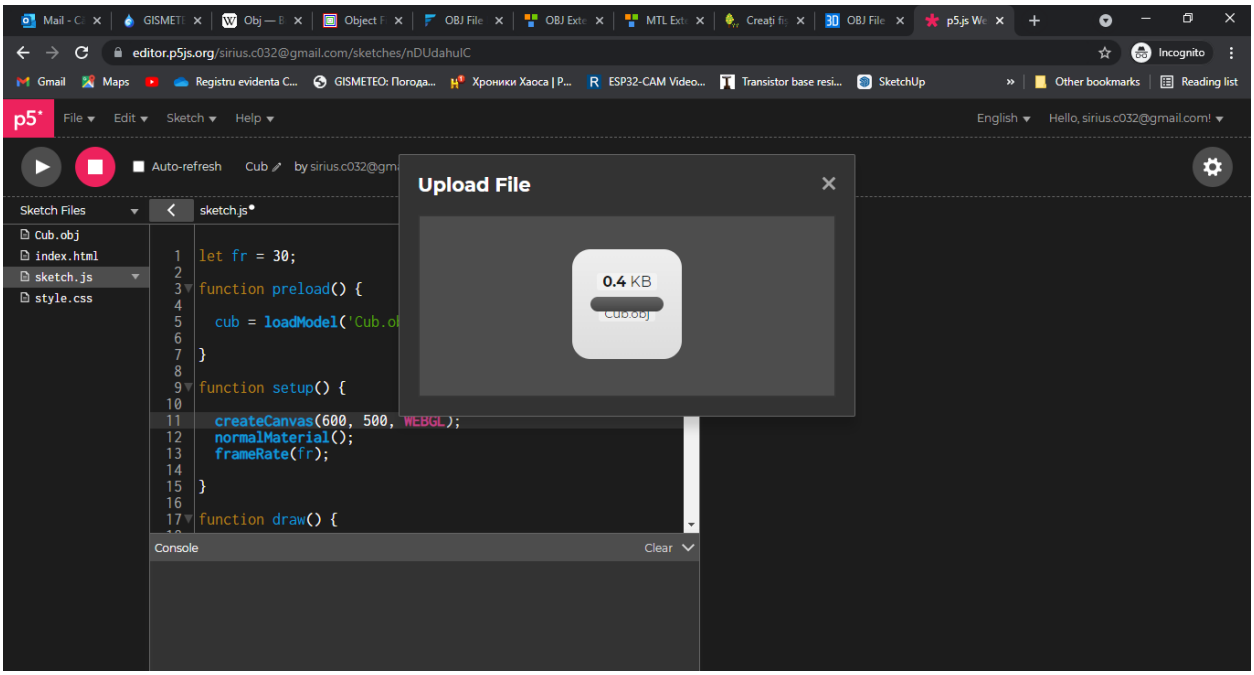


Fig. 3.5 Rezultatul procesului de încarcare a fișierului Cub.obj în directoriul proiectului.

```
let fr = 30;
function preload()
{
	cub = loadModel('Cub.obj');
}
function setup()
 {
	createCanvas(600, 500, WEBGL);
	normalMaterial();
	frameRate(fr);
}

function draw()
{
	orbitControl();
	background(50);
	normalMaterial();
	scale(1.0);
	stroke(100, 100, 100);
	strokeWeight(0.3);
	fill(150, 150, 150);
	model(cub);
}
```

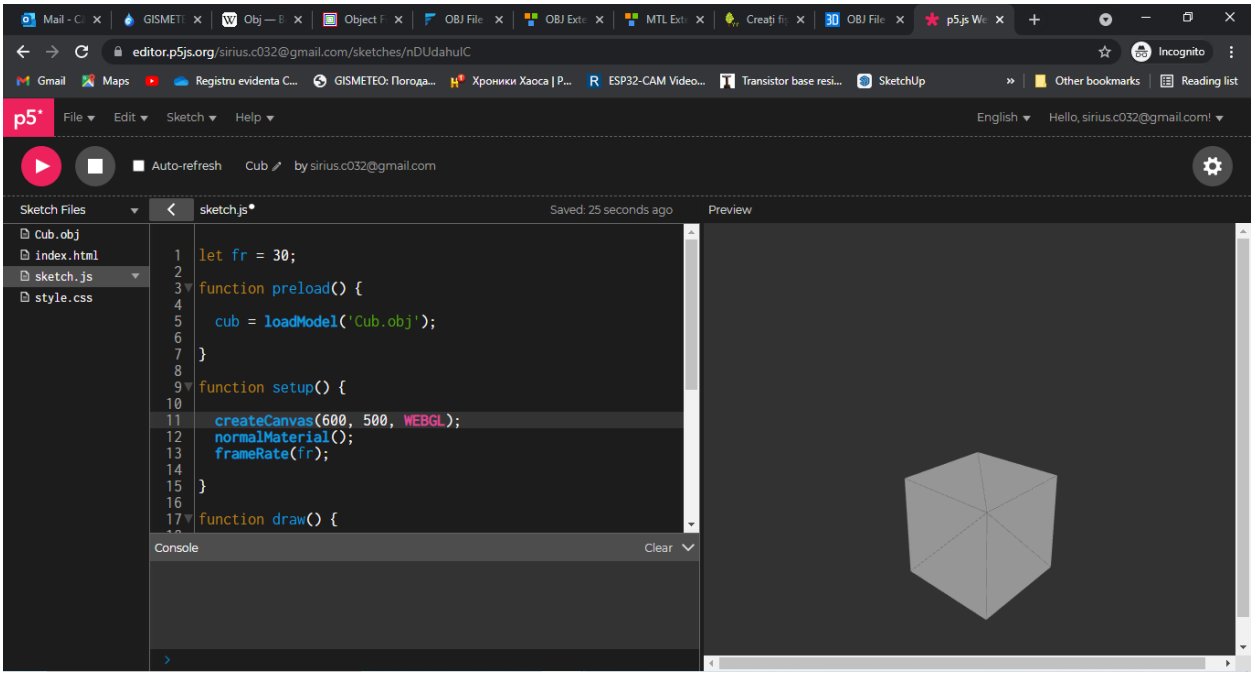


Fig. 3.5 Rezultatul execuției programului de afișare a modelului 3D a cubului.

Crearea unei scene statice 3D.

Sarcina lucrării de laborator constă în conceperea și construirea unei scene statice 3D cu ajutorul editorului online p5.js utilizînd modele de obiecte 3D stocate în fișiere .OBJ create individual sau descărcate din internet.

Pentru exemplificarea mersului lucrării de laborator nr. 3 se pune ca scop crearea unei scene statice care ar reprezenta o moară de vînt îngrădită de un gard în ograda căreia se vor afla cîteva animale deomestice, un arbore și un stog de fîn. Modelele 3D ale acestor obiecte pot fi create de la zero sau importate din repozitoriul 3D Warehouse care poate fi accesat din meniul de bază al editorului grafic 3D Google SketchUp *Window->3D Warehouse*, apoi editate după necesitate.

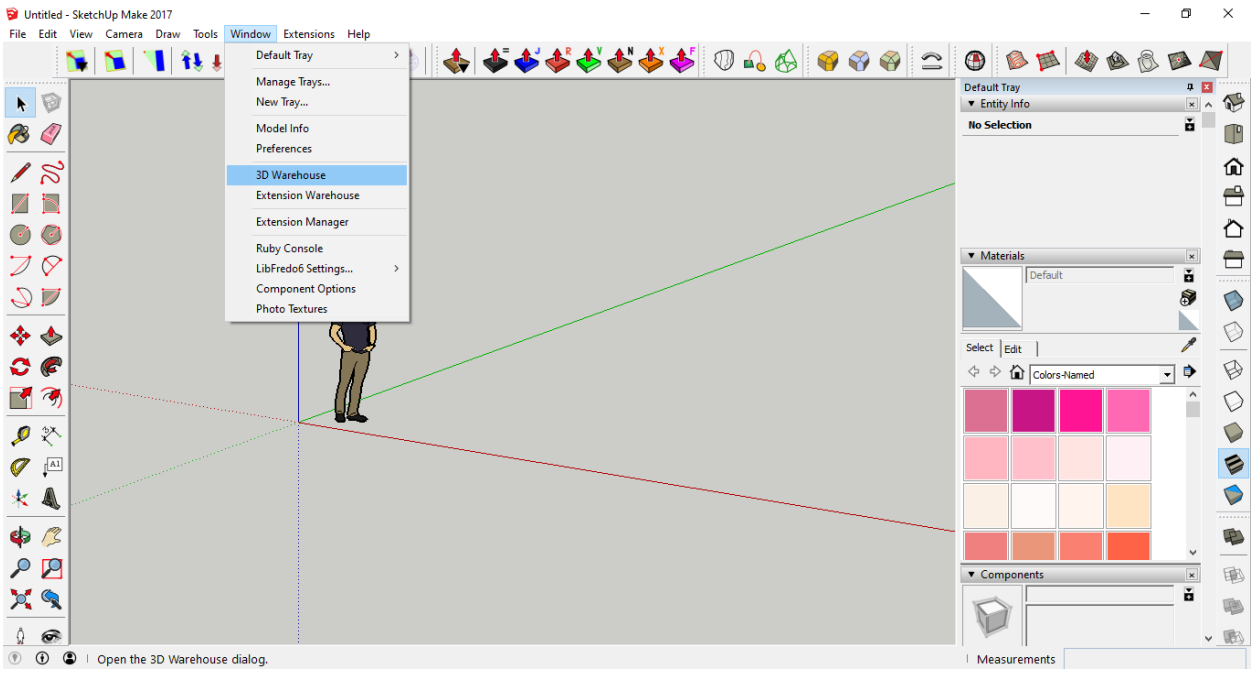


Fig. 3.6 Accesarea repozitoriului 3D Warehouse.

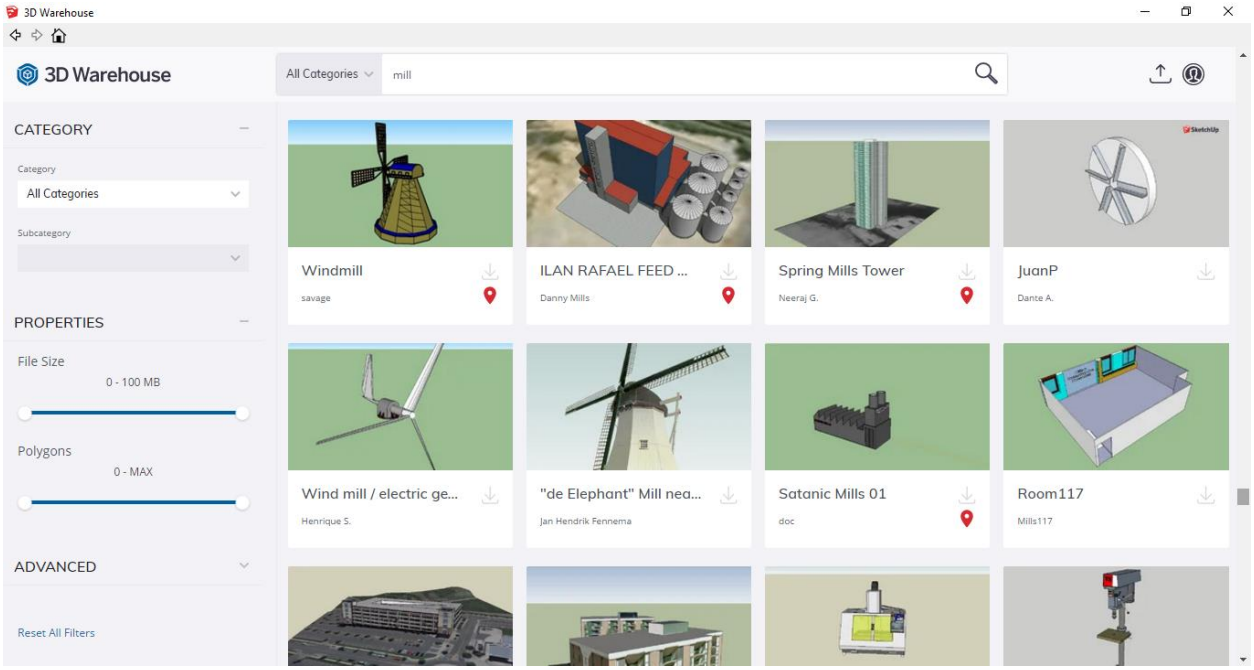


Fig. 3.7 Căutarea modelelor 3D în repozitoriul 3D Warehouse.

Pentru a avea posibilitatea de a downloada sau importa modelul selectat în sketch-ul curent utilizatorul trebuie să dispună de un cont Google Account și să fie autentificat în mediul 3D Warehouse. După ce a fost downloadat modelul/modelele necesare, acestea pot fi editate după necesitate și exportate ca fișiere .OBJ prin intermediului plug-inului *LIPID OBJ Exporter* care

poate fi instalat din meniul *Window->Extension Warehouse*, Fig. 3.7, în cîmpul de căutare al căruia indicăm cuvîntul cheie OBJ iar ca răspuns sînt afișate plugin-urile care satisfac criteriile de căutare printre care poate fi observat și plugin-ul căutat *LIPID OBJ Exporter*, Fig. 3.9. Pentru a avea posibilitatea de a instala plugin-urile, utilizatorul trebuie să dispună de un cont Google Account și să fie autentificat în repozitoriul online *Extension Warehouse*.

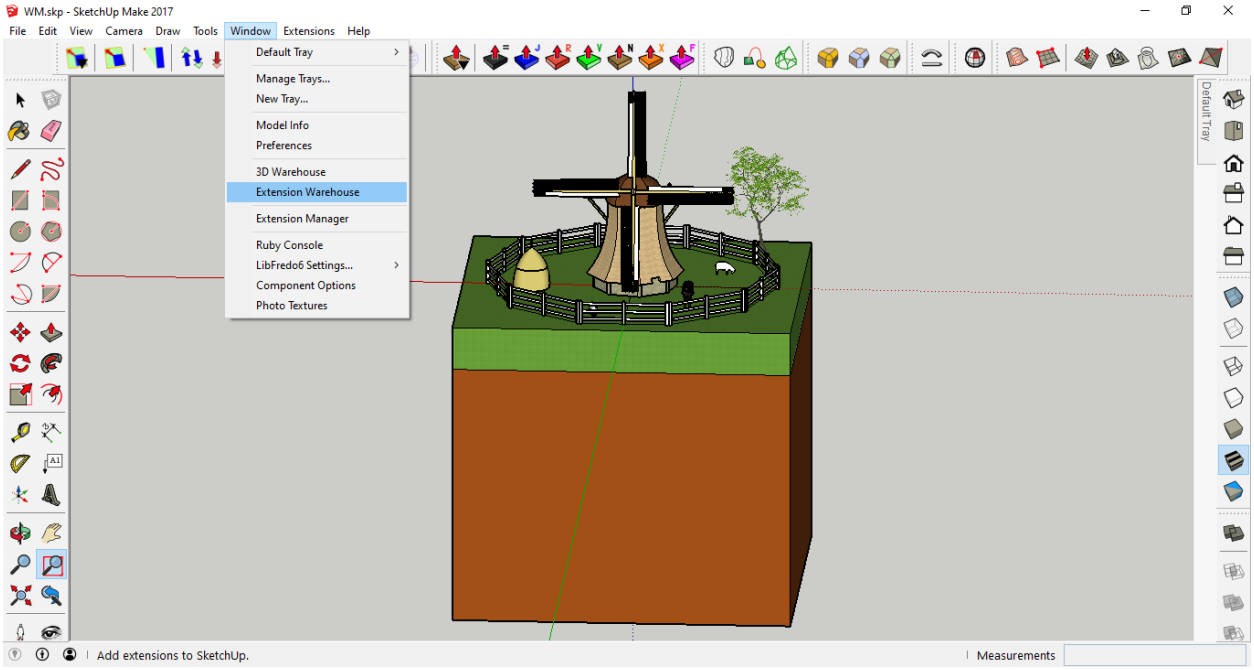


Fig. 3.8 Accesarea opțiunii *Extension Warehouse* din bara de meniu.

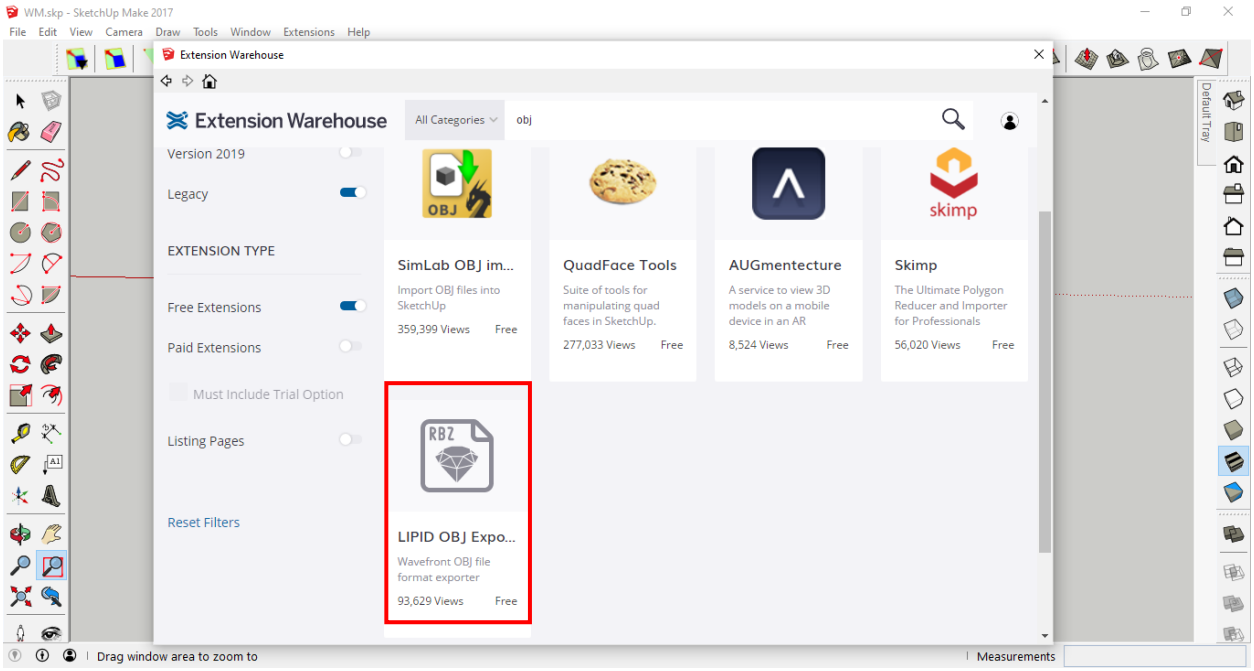


Fig. 3.9 Fereastra de căutare și instalare a plugin-urilor, din repozitoriul online *Extension Warehouse*.

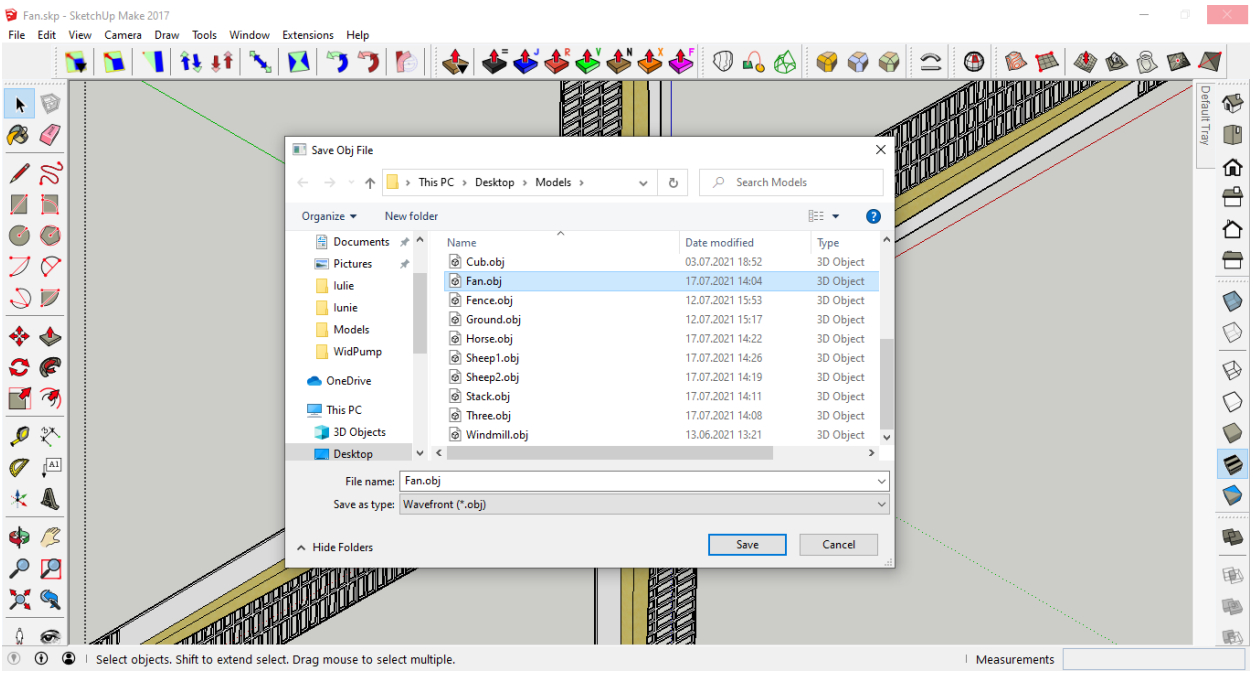


Fig. 3.10 Fereastra de salvare în format .obj a modelelor 3D a obiectelor din scenă.

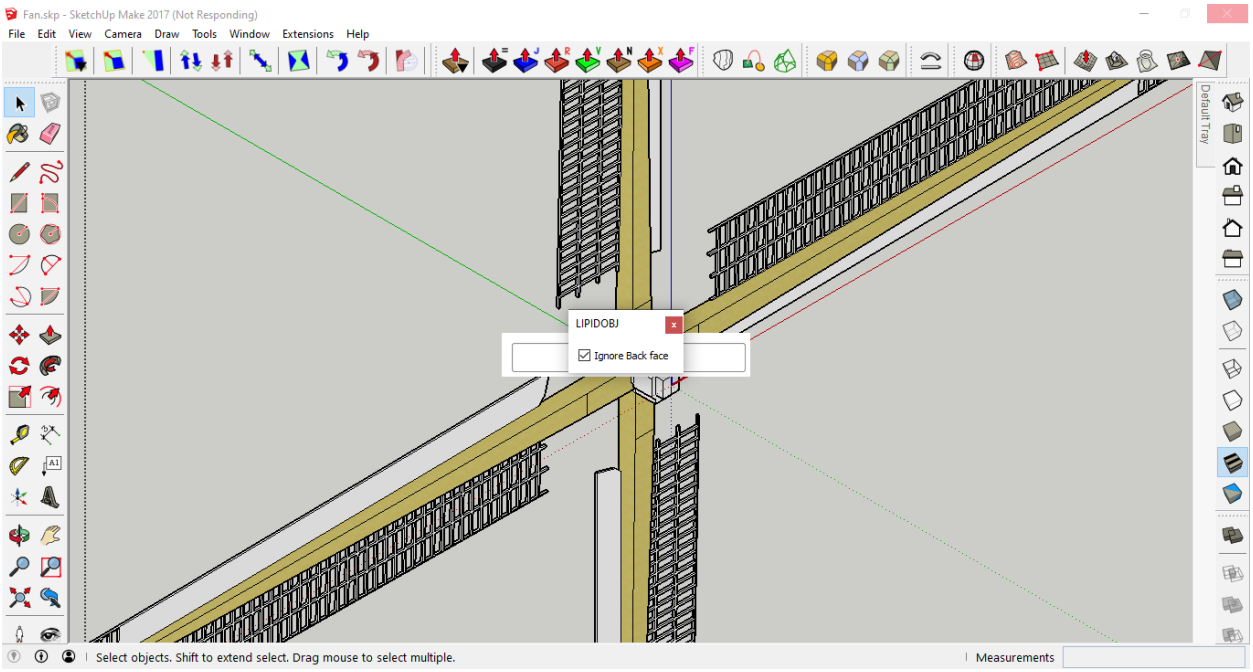


Fig. 3.11 Opțiunea de ignorare a fețelor a plugin-ului LIPIDOBJ la exportarea modelelor 3D în format .obj.

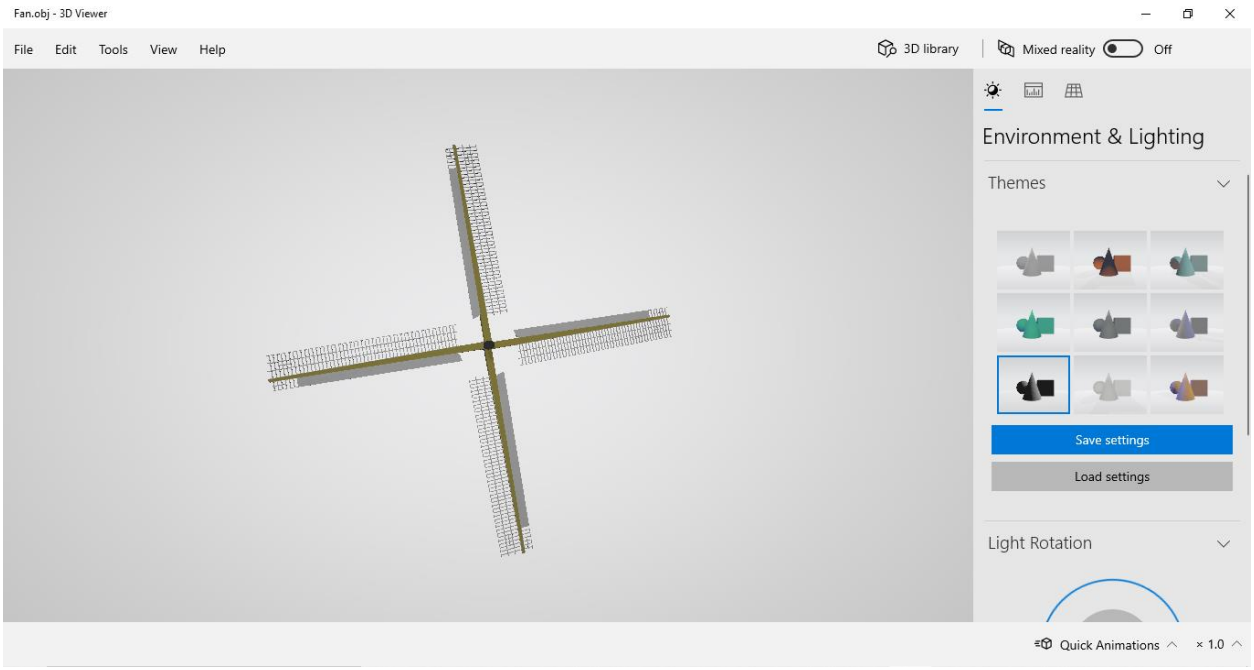


Fig. 3.12 Vizualizarea conținutului fișierelor .OBJ cu ajutorul aplicației standard din cadrul sistemului de operare Windows 10 – 3D Viewer.

```
let fr = 30;
let obj;

function preload() {

  mill = loadModel('Windmill.obj');
  fan = loadModel('Fan.obj');
  gnd = loadModel('Ground.obj');
  sheep1 = loadModel('Sheep1.obj');
  sheep2 = loadModel('Sheep2.obj');
  horse = loadModel('Horse.obj');
  stack = loadModel('Stack.obj');
  fence = loadModel('Fence.obj');
  three = loadModel('Three.obj');
}
```

```
function setup() {

  createCanvas(400, 400, WEBGL);
  normalMaterial();
  //ambientMaterial(150, 150, 150);
  frameRate(fr);
  angleMode(DEGREES);
}

function draw() {
  orbitControl();
  background(50);
  pointLight(255, 255, 255, 100000, 100000, -100000);
  noStroke();

  push();
  scale(0.005);
  translate(0, 0, 0);
  model(mill);
  pop();

  push();
  scale(0.005);
  translate(0, 3000, 13000);
  rotateX(15);
  model(fan);
  pop();

  push();
  scale(0.005);
  model(gnd);
  pop();
```

```
push();
scale(0.005);
translate(-3000, 15000, 0);
rotateZ(-10);
model(sheep1);
pop();

push();
scale(0.005);
translate(10000, 0, 0);
rotateZ(180);
model(sheep2);
pop();

push();
scale(0.005);
rotateZ(-15);
translate(7000, 12000, 0);
model(horse);
pop();

push();
scale(0.005);
translate(-12000, -2000, 0);
model(stack);
pop();

push();
scale(0.005);
translate(0, 0, 0);
model(fence);
pop();
```

```
  push();
  scale(0.005);
  translate(15000, -15000, 0);
  model(three);
  pop();
}
```

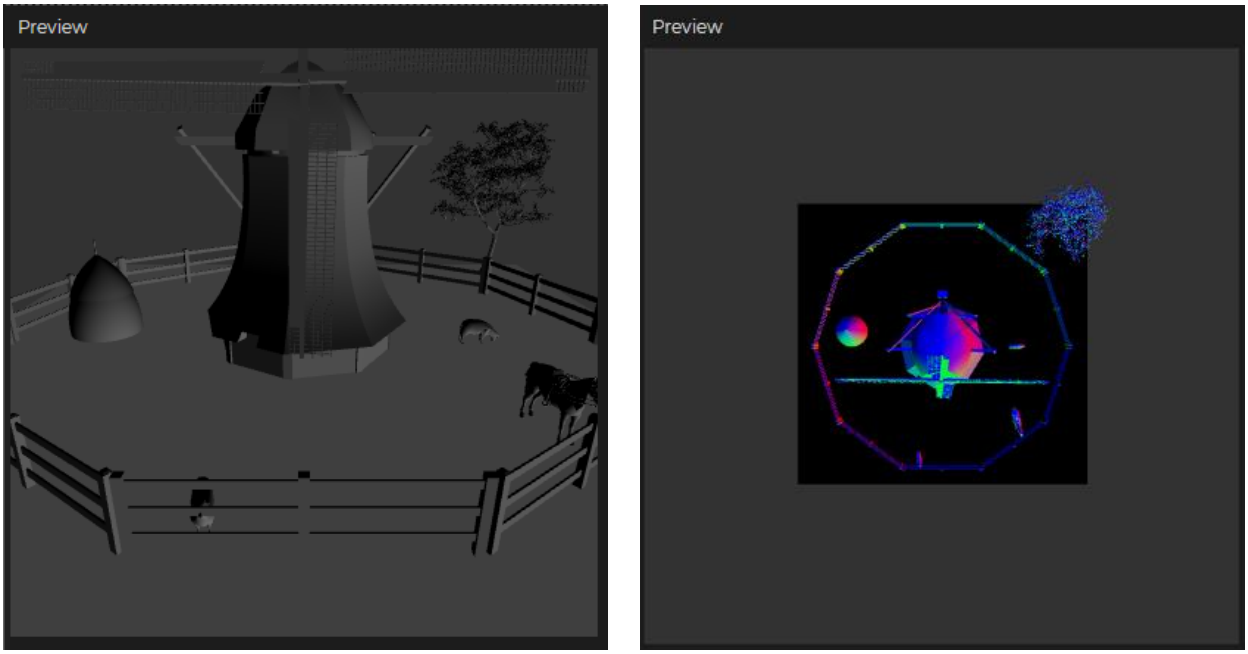
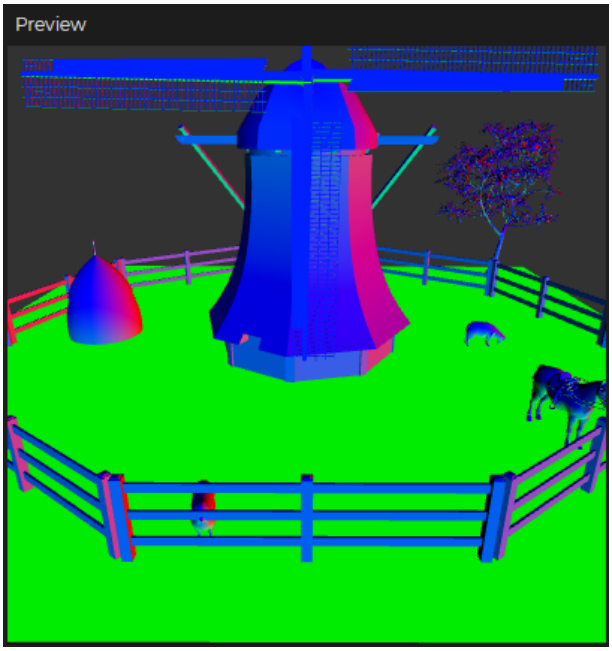


Fig. 3.13 Rezultatul execuției programului de afișare a scenei statice.



Lucrarea de laborator nr. 3

Tema: Scene statice 3D

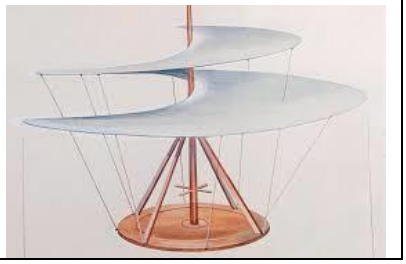
Scopul lucrării: Obţinerea cunoştinţelor practice în sinteza scenelor grafice 3D statice, utilizînd funcțiile standard de reprezentare a modelelor 3D din biblioteca p5.js.

Sarcina lucrării:

1. Elaborați un program pentru sinteza unei scene 3D statice care ar conține cel puțin 5 obiecte utilizîng funcțiile standard de reprezentare a modelelor 3D din biblioteca p5.js.

2. Elaborați un program care crează o scenă 3D statică conform variantei indicate în tabelul 3.1. Pentru crearea scenei pot fi utilizate obiecte grafice 3D existente în repozitoriul 3D Warehouse.

Tabelul 3.1 Variantele pentru realizarea lucrarii de laborator

Nr	Obiect 3D	Model	Nr	Obiect 3D	Model
1	Carusel		6	Morișcă de vînt	
2	Far maritim		7	Turbină eoliană	
3	Automobil blindat		8	Elicopter	

4	Moară de apă		9	Submarină	
5	Avion cu elice		10	Catapultă