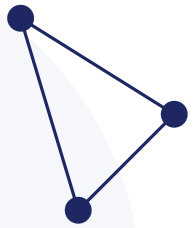
FORMATY DANYCH W GRAFICE 3D

Formaty plików obiektów 3D

Reprezentacja geometrii, sposób zapisu danych i zastosowania



dr inż. Mateusz Pomianek

CELE KSZTAŁCENIA

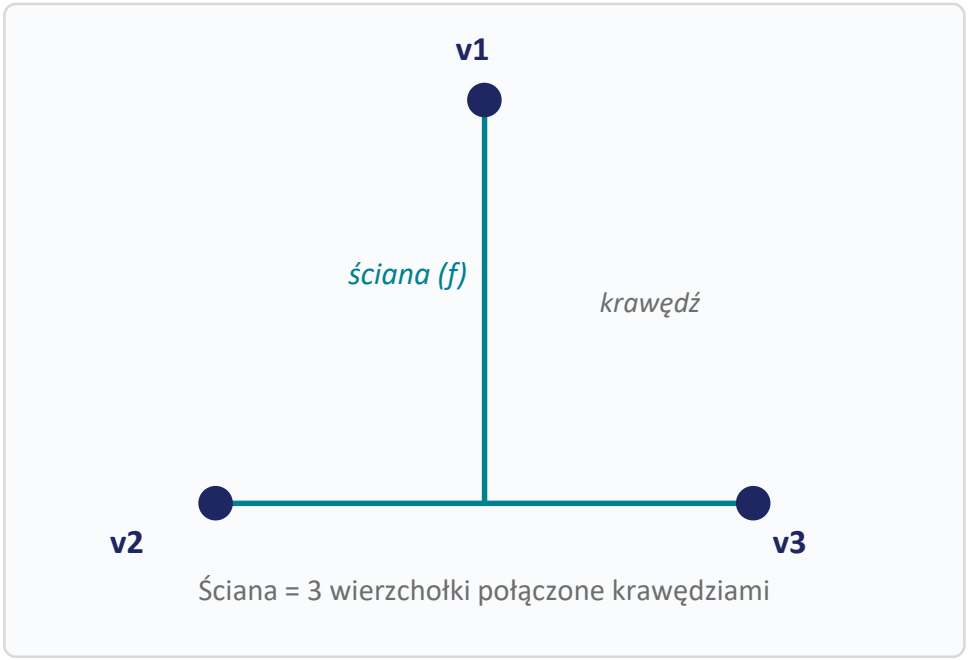
Po zajęciach student samodzielnie analizuje i dobiera format pliku 3D

- Definiuje pojęcie formatu pliku 3D i wskazuje jego podstawowe elementy składowe.
- Klasyfikuje formaty 3D ze względu na sposób reprezentacji geometrii (siatka, scena, CAD).
- Porównuje sposób zapisu danych geometrycznych w formatach OBJ, STL, PLY, glTF, FBX, COLLADA i STEP.
- Dobiera format pliku odpowiedni do zastosowania: druk 3D, grafika czasu rzeczywistego, projektowanie inżynierskie.

Format pliku 3D określa sposób zapisu geometrii i jej atrybutów

- Podstawowym elementem geometrii jest siatka wielokątów (najczęściej trójkątów), zbudowana z wierzchołków (ang. vertices), krawędzi i ścian (ang. faces).
- Format może dodatkowo przechowywać: normalne (kierunek powierzchni), współrzędne tekstury (UV), kolory, materiały, hierarchię obiektów oraz animacje.
- Dane zapisuje się w postaci tekstowej - czytelnej dla człowieka, lub binarnej - bardziej kompaktowej i szybszej do odczytu.

Wierzchołki, krawędzie i ściana



KLASYFIKACJA

Formaty 3D dzieli się na siatkowe, sceniczne oraz CAD (B-rep)

MESH

Formaty siatkowe

Zapisują wyłącznie geometrię: wierzchołki i ściany.

PRZYKŁADY

OBJ, STL, PLY

SCENA

Formaty sceny

Geometria + hierarchia obiektów, materiały, animacje, kamery.

PRZYKŁADY

glTF, FBX, COLLADA

CAD

Formaty CAD (B-rep)

Reprezentacja brzegowa - dokładne powierzchnie i krzywe, bez aproksymacji trójkątami.

PRZYKŁADY

STEP, IGES

OBJ

FORMAT SIATKOWY

OBJ zapisuje geometrię jako czytelną listę wierzchołków i ścian

- Format tekstowy (ASCII) opracowany przez firmę Wavefront Technologies.
- Współrzędne wierzchołków (v), tekstury (vt) i normalnych (vn) zapisywane są jako osobne listy liczb.
- Ściany (f) odwołują się do numerów wierzchołków z tych list - indeksowanie zaczyna się od 1.
- Nie przechowuje animacji ani hierarchii sceny, ale wyłącznie statyczną geometrię i materiały (jako osobny plik .mtl).

Przykład zapisu danych

```
v 0.0 0.0 0.0
v 1.0 0.0 0.0
v 0.0 1.0 0.0
vt 0.0 0.0
vn 0.0 0.0 1.0
f 1/1/1 2/1/1 3/1/1
```

STL

FORMAT SIATKOWY

STL opisuje bryłę jako zbiór niezależnych trójkątów z wektorem normalnym

- Standard powstały na potrzeby druku 3D (stereolitografii), opracowany przez firmę 3D Systems.
- Każdy trójkąt zapisany jest niezależnie, bez wspólnej listy wierzchołków - powoduje to redundancję danych.
- Wersja ASCII jest czytelna tekstowo; wersja binarna zapisuje te same dane jako liczby zmiennoprzecinkowe 32-bitowe i jest znacznie bardziej kompaktowa.
- Podstawowa specyfikacja nie przechowuje kolorów, tekstur ani materiałów.

Przykład zapisu danych

```
solid przyklad
 facet normal 0 0 1
  outer loop
   vertex 0 0 0
   vertex 1 0 0
   vertex 0 1 0
  endloop
 endfacet
endsolid przyklad
```

PLY

FORMAT SIATKOWY

PLY oddziela nagłówek opisujący strukturę danych od samych wartości

- Format opracowany na Uniwersytecie Stanforda do zapisu danych ze skanowania 3D.
- Nagłówek (header) definiuje listę elementów (np. vertex, face) i ich właściwości (property), zanim pojawią się dane liczbowe.
- Taka struktura pozwala zapisać dowolne dodatkowe atrybuty punktu, np. kolor albo pewność pomiaru.
- Występuje w wersji tekstowej (ascii) oraz binarnej (binary_little_endian / binary_big_endian).

Przykład zapisu danych

```
ply
format ascii 1.0
element vertex 3
property float x
property float y
property float z
element face 1
property list uchar int vertex_indices
end_header
0 0 0
1 0 0
0 1 0
3 0 1 2
```

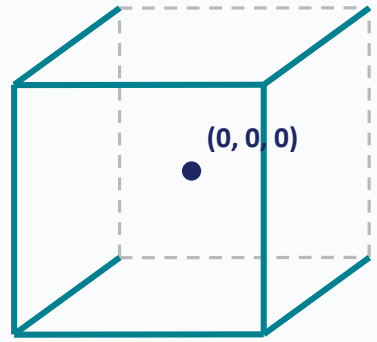
PLY

PLY - PRZYKŁAD ZAPISU BRYŁY

Sześcian 1×1×1 wyśrodkowany w (0, 0, 0): osiem wierzchołków i sześć ścian

- Krawędź sześcianu ma długość 1, więc współrzędne wierzchołków przyjmują wartości ±0,5 w każdej osi.
- Środek bryły znajduje się w (0, 0, 0) - każdy wierzchołek jest przesunięty o połowę krawędzi od środka.
- Lista list (property list uchar int vertex_indices) pozwala zapisać ścianę o dowolnej liczbie wierzchołków - tu każda ściana ma po 4.

Wizualizacja bryły



a = 1

Przykład zapisu danych

```
ply
format ascii 1.0
element vertex 8
property float x
property float y
property float z
element face 6
property list uchar int vertex_indices
end_header
-0.5 -0.5 -0.5
 0.5 -0.5 -0.5
 0.5  0.5 -0.5
-0.5  0.5 -0.5
-0.5 -0.5  0.5
 0.5 -0.5  0.5
 0.5  0.5  0.5
-0.5  0.5  0.5
4 0 1 2 3
4 7 6 5 4
4 0 4 5 1
4 1 5 6 2
4 2 6 7 3
4 3 7 4 0
```

glTF

FORMAT SCENY

glTF przechowuje scenę jako JSON, a dane liczbowe w osobnym buforze binarnym

- Otwarty standard opracowany przez konsorcjum Khronos Group, nazywany „JPEG-iem grafiki 3D”.
- Plik .gltf (JSON) opisuje sceny, węzły i materiały oraz odwołania do danych (accessors); same liczby - współrzędne i indeksy - leżą w pliku binarnym .bin.
- Wariant .glb łączy część opisową i binarną w jednym pliku.
- Obsługuje hierarchię obiektów, animacje szkieletowe i materiały fizyczne (PBR).

Przykład zapisu danych

```
"meshes": [{
  "primitives": [{
    "attributes": {"POSITION": 1},
    "indices": 0
  }]
}],
"accessors": [
  {"bufferView":0,"componentType":5123,
   "count":3,"type":"SCALAR"},
  {"bufferView":1,"componentType":5126,
   "count":3,"type":"VEC3"}
]
```

FBX

FORMAT SCENY

FBX to format Autodesk, przechowujący scenę jako drzewo węzłów

- Występuje w wersji binarnej (domyślna, kompaktowa) oraz tekstowej (ASCII) - obie mają identyczną strukturę węzłów.
- Dane geometrii (Vertices, PolygonVertexIndex) są zagnieżdżone w węzłach opisujących obiekty typu Model, Mesh, Material, AnimationCurve.
- Szeroko stosowany w grafice gier i filmu ze względu na obsługę animacji szkieletowej i złożonych scen.
- Pełna specyfikacja formatu binarnego nie jest publiczna - odczyt i zapis wymagają FBX SDK udostępnianego przez Autodesk.

Przykład zapisu danych

```
Model: "Model::Cube", "Mesh" {
  Vertices: *12 {
    a: 0,0,0, 1,0,0, 1,1,0, 0,1,0
  }
  PolygonVertexIndex: *4 {
    a: 0,1,2,-4
  }
}
```

DAE

FORMAT SCENY

COLLADA zapisuje scenę 3D w ustrukturyzowanym pliku XML

- Plik z rozszerzeniem .dae (Digital Asset Exchange); format ustandaryzowany jako ISO/PAS 17506.
- Dane liczbowe - współrzędne i indeksy - zapisane są jako tekstowe tablice wewnątrz znaczników <float_array> i <source>.
- Struktura XML pozwala jednoznacznie opisać hierarchię, materiały, oświetlenie i animacje, kosztem większego rozmiaru pliku.
- Stosowany jako format pośredni przy wymianie danych między programami do grafiki 3D.

Przykład zapisu danych

```
<mesh>
  <source id="pozycje">
    <float_array count="9">
      0 0 0 1 0 0 0 1 0
    </float_array>
  </source>
  <triangles count="1">
    <p>0 1 2</p>
  </triangles>
</mesh>
```

STEP

FORMAT CAD (B-REP)

STEP opisuje bryłę matematycznie: jako powierzchnie i krawędzie, nie jako siatkę

- Norma ISO 10303; zapis tekstowy w postaci zdefiniowanej przez część ISO 10303-21 (clear text encoding).
- Geometria opisana jest reprezentacją brzegową (B-rep) - dokładnymi powierzchniami i krzywymi, bez aproksymacji trójkątami.
- Każdy element (punkt, krzywa, powierzchnia) ma unikalny numer odniesienia (np. #10), do którego odwołują się inne elementy pliku.
- Standard wymiany danych między systemami CAD/CAM - zachowuje pełną dokładność geometryczną modelu.

Przykład zapisu danych

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('),'2;1');
ENDSEC;
DATA;
#10=CARTESIAN_POINT('',(0.,0.,0.));
#11=CARTESIAN_POINT('',(10.,0.,0.))
;
#12=LINE('',#10,#13);
ENDSEC;
END-ISO-10303-21;
```

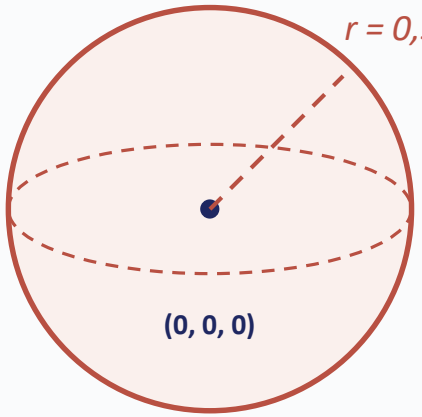
STEP

STEP - PRZYKŁAD ZAPISU BRYŁY

Kula o promieniu 0,5 wyśrodkowana w (0, 0, 0) jako powierzchnia SPHERICAL_SURFACE

- Środek kuli pokrywa się z początkiem układu współrzędnych - punkt #1 CARTESIAN_POINT ma wartość (0., 0., 0.).
- Encja SPHERICAL_SURFACE odwołuje się do lokalnego układu AXIS2_PLACEMENT_3D (#4) oraz do wartości promienia - tu 0,5.
- W odróżnieniu od formatów siatkowych (PLY, OBJ, STL) powierzchnia jest zapisana jako dokładna funkcja matematyczna, a nie zbiór trójkątów.

Wizualizacja bryły



Przykład zapisu danych

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('),'2;1');
ENDSEC;
DATA;
#1=CARTESIAN_POINT('Center',(0.,0.,0.));
#2=DIRECTION('axis',(0.,0.,1.));
#3=DIRECTION('refdir',(1.,0.,0.));
#4=AXIS2_PLACEMENT_3D('',#1,#2,#3);
#5=SPHERICAL_SURFACE('Sphere',#4,0.5);
ENDSEC;
END-ISO-10303-21;
```

PORÓWNANIE

Formaty różnią się kategorią, kodowaniem i przeznaczeniem

Format	Kategoria	Kodowanie	Zapis geometrii	Typowe zastosowanie
OBJ	Siatka	Tekstowe	Wierzchołki + ściany	Wymiana modeli statycznych
STL	Siatka	Tekstowe / binarne	Niezależne trójkąty	Druk 3D
PLY	Siatka	Tekstowe / binarne	Siatka + dowolne atrybuty	Skanowanie 3D, chmury punktów
glTF / GLB	Scena	JSON + binarne	Siatka + hierarchia + animacje	Aplikacje czasu rzeczywistego, AR/VR
FBX	Scena	Binarne / tekstowe (zastrzeżone)	Siatka + hierarchia + animacje	Gry, film, animacja
COLLADA (.dae)	Scena	XML	Siatka + hierarchia + animacje	Wymiana między programami DCC
STEP	CAD (B-rep)	Tekstowe (EXPRESS)	Powierzchnie i krawędzie parametryczne	Projektowanie inżynierskie CAD/CAM

PODSUMOWANIE

Wybór formatu zależy od tego, czy priorytetem jest dokładność, rozmiar pliku czy animacja

- Formaty siatkowe (OBJ, STL, PLY) przechowują wyłącznie geometrię - są proste, ale ograniczone do statycznych modeli.
- Formaty sceny (glTF, FBX, COLLADA) dodają hierarchię obiektów, materiały i animacje kosztem większej złożoności pliku.
- Formaty CAD (STEP, IGES) zachowują pełną precyzję matematyczną powierzchni, niedostępną w reprezentacji siatkowej.
- Kodowanie binarne zmniejsza rozmiar pliku; kodowanie tekstowe ułatwia podgląd, debugowanie i edycję ręczną.

PYTANIA SPRAWDZAJĄCE

Sprawdź, czy potrafisz wyjaśnić różnice między formatami

- Dlaczego plik STL jest zwykle większy niż plik OBJ opisujący ten sam model?
- Jaki problem rozwiązuje podział pliku glTF na część JSON i część binarną?
- Dlaczego format STEP nie przechowuje siatki trójkątów, mimo że większość programów graficznych ostatecznie renderuje bryłę jako siatkę?
- Który format wybrałbyś do zapisu modelu z animacją postaci? Uzasadnij wybór.

BIBLIOGRAFIA

Źródła

- [1] Wavefront Technologies, "Object Files (.obj) File Format Specification," 1995.
- [2] 3D Systems, Inc., "StereoLithography Interface Specification," 1989.
- [3] Stanford University, Computer Graphics Laboratory, "The PLY Polygon File Format," 1994.
- [4] Khronos Group, "glTF 2.0 Specification," 2017.
- [5] Autodesk, Inc., "FBX SDK Documentation."
- [6] Khronos Group, "COLLADA - Digital Asset Schema Release 1.5.0," ISO/PAS 17506:2012.
- [7] International Organization for Standardization, "ISO 10303-21: Industrial automation systems and integration - Product data representation and exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure," 2016.