



Домашна Работа №1

ООП, СУ ФМИ

25/03/2025

1 PLY файл

Тримерен модел ще наричаме съвкупност от точки в тримерното пространство и съвкупност от многоъгълници с върхове измежду дадените върхове.

Често използван формат за съхраняване на такива обекти във файл е Stanford PLY (.ply). Файл от този тип е текстов и започва с хедър:

Listing 1: Код

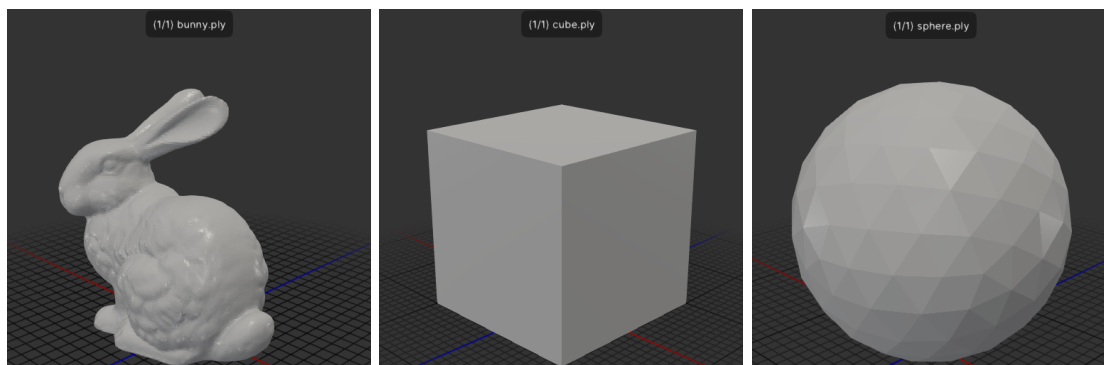
```
1 ply
2 format ascii 1.0
3 element vertex N
4 property float x
5 property float y
6 property float z
7 element face M
8 property list uchar int vertex_indices
9 end_header
```

Тук N се замества с броя точки в модела, а M с броя многоъгълници в него. Между задължителните редове в хедъра може да има коментари. Всеки ред, започващ с думата 'comment' се игнорира. Например: `comment This is a comment`

След това следва списък от координатите на N точки, участващи в модела - по три числа с плаваща запетая на ред - x y z . Накрая следват описанията на M многоъгълника. Всеки многоъгълник се описва с броя на върховете му и индексите на тези върхове. Например, ако искаме да описваме многоъгълник с 4 върха, първият ред ще бъде '4 0 1 2 3', ако върховете с индекси 0, 1, 2 и 3 са точките, които образуват многоъгълника.

Такива файлове може да визуализирате чрез приложението 3D Viewer в Windows 10. Друга добра опция е f3d.

1.1 Примери



Кубът от фигурата е описан във файл cube.ply, който изглежда така:

```
1 ply
2 format ascii 1.0
3 comment I made this by hand
4 element vertex 8
5 property float x
6 property float y
7 property float z
8 element face 6
9 property list uchar int vertex_indices
10 end_header
11 -1 -1 -1
12 1 -1 -1
13 1 -1 1
14 -1 -1 1
15 -1 1 -1
16 1 1 -1
17 1 1 1
18 -1 1 1
19 4 0 1 2 3
20 4 7 6 5 4
21 4 0 4 5 1
22 4 1 5 6 2
23 4 2 6 7 3
24 4 3 7 4 0
```

2 Задача

Реализирайте клас Object3d, който представя тримерен модел в езика C++.

- Реализирайте конструктор, който приема име на файл и зарежда модел от него.
- Реализирайте конструктор, който приема поток за вход и зарежда модел от него.
- Реализирайте методи `int getVertexCount()` и `int getFaceCount()`, които връщат съответно броя върхове и броя многоъгълници в модел.
- Реализирайте метод `void save(const string&)`, който приема име на файл и съхранява модела във файла.
- Реализирайте метод `void print(ostream &)`, който приема изходен поток, в който да се запишат данните на обекта в указания формат.
- Реализирайте метод `Object3d cut(function<bool(float, float, float)> f)`, който връща модел съставен само от тези многоъгълници в модела, координатите на чиито върхове отговарят на предиката f .
- Реализирайте метод `void flip()`, който обръща всички многоъгълници в модела наобратно. (Четириъгълникът 1 2 3 4 става 4 3 2 1)
- Реализирайте статични методи за създаване на куб и сфера с параметри съответно дължина на ръб и радиус. (Не е задължително сферата да изглежда точно като на картинката по-горе).

При опит за прочитане на невалиден файл изведете подходящи съобщения.

2.1 Пример

""

```
1 Object3d b("./bunny.obj");
2 Object3d b_cut = b.cut([](float x, float y, float z) -> bool {
3     return y + z < .12;
4 });
5 b_cut.save("./bunny_cut.obj");
```

Кодът по-горе трябва да произведе файл *bunny_cut.obj*, който съдържа само част от модела на *bunny.obj*, която се намира под плоскостта $y = x + 2.5$. Резултатът изглежда така:



3 Ресурси

Файловете от картинките по-горе са качени в Moodle. Първо тествайте кода си със по-малките файлове *cube.ply* и *sphere.ply* преди да правите каквото и да е с *bunny.obj*. При желание може да използвате и други модели от интернет, например от хранилището на университета на Станфорд.

4 Предаване

Предайте всички '.cpp', '.hpp', '.h' файлове в заданието в мудъл. Ако използвате Make или CMake, качете и съответните файлове, описващи компилацията. Не предавайте '.ply', '.zip' и други бинарни файлове. Разрешено е използването само и единствено на библиотеките *vector*, *string*, *cstring*, *iostream*, *fstream*, *functional*, *cstdint*, *cmath*, *limits* от стандарта на езика.

Краен срок: 27/04/2025

Успех!