

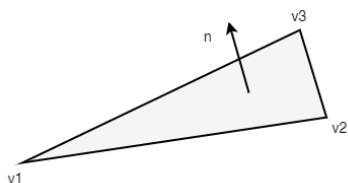
Zadání 22. 4. 2024

Úlohu odevzdejte nejpozději 5. 5. 2024 (do 23:59) přes MS Teams, případně zašlete na e-mail marketa.trneckova@gmail.com s předmětem "ZPC2 uloha 2". Do tohoto data je možné posílat opravu. U řešení odevzdaných později než 3. 5. nezaručuji, že stihnu opravit před termínem odevzdání. Odevzdávejte pouze soubor s příponou .c

Program bude implementovat práci se soubory ve formátu .stl, který popisuje trojrozměrné objekty pomocí množiny trojúhelníků. Více o struktuře souboru ve formátu .stl na konci zadání.

Povinná část

Vytvořte strukturu `trojuhelnik`, která reprezentuje informace o trojúhelníku, jako jsou souřadnice vrcholů (`v1`, `v2` a `v3`) a jeho normálový vektor `n`. Všechny položky budou pole (obsahující 3 hodnoty) typu `float`.



Vytvořte strukturu `objekt`, která bude uchovávat informaci o počtu trojúhelníků pojmenovanou `pocet` (typu `int`), ze kterých se skládá, a ukazateli na pole trojúhelníků (pole typu `trojuhelnik`), pojmenovaný `trojuhelniky`.

Naprogramujte funkci, která je deklarovaná `objekt nacti(char*)`; . Tato funkce pro zadaný řetězec, který představuje cestu k **binárnímu** souboru ve formátu .stl ze souboru načte všechny trojúhelníky a vytvoří objekt typu `objekt`, který bude představovat objekt, který je v souboru uložený. Funkce vrátí tento objekt. V případě, že dojde k chybě, vrátí objekt, který bude mít `pocet` roven 0 a ukazatel `trojuhelniky` bude nastaven na NULL.

Naprogramujte funkci, která je deklarovaná `void vypis(objekt)`; . Tato funkce vypíše předaný objekt jako seznam trojúhelníků, viz následující obrázek, který je výstupem pro soubor `''testbin.stl''`.

```
Trojuhelnik 1
n = [0.447213, 0.447213, -0.774597]
v1 = [-0.500000, 0.866025, 0.000000]
v2 = [0.000000, -0.000000, 1.000000]
v3 = [1.000000, 0.000000, 0.000000]

Trojuhelnik 2
n = [-0.894427, 0.447214, 0.000000]
v1 = [-0.500000, -0.866025, 0.000000]
v2 = [0.000000, -0.000000, 1.000000]
v3 = [-0.500000, 0.866025, 0.000000]

Trojuhelnik 3
n = [0.447213, 0.447213, 0.774597]
v1 = [1.000000, 0.000000, 0.000000]
v2 = [0.000000, -0.000000, 1.000000]
v3 = [-0.500000, -0.866025, 0.000000]

Trojuhelnik 4
n = [0.000000, -1.000000, 0.000000]
v1 = [-0.500000, -0.866025, 0.000000]
v2 = [-0.500000, 0.866025, 0.000000]
v3 = [1.000000, 0.000000, 0.000000]
```

Nepovinná část.

Následující funkce nejsou povinné, viz bodování.

Naprogramujte funkci, která je deklarovaná `int uloz(char*, objekt);`. Funkce objekt reprezentovaný strukturou `objekt` uloží do **ASCII** `.stl` souboru, jehož název je funkci předaný jako řetězec. Funkce vrátí 1, pokud vše proběhlo v pořádku, 0 v případě, že došlo k chybě.

Naprogramujte funkci, která je deklarovaná `objekt sjednoceni(objekt, objekt);`. Tato funkce vrátí objekt, který vznikl sjednocením objektů, které jí byly předány jako argumenty (objekt obsahuje všechny trojúhelníky z obou objektů). V případě chyby vrátí objekt, který bude mít `pocet` roven 0 a ukazatel `trojuhelniky` bude roven `NULL`.

Naprogramujte funkci `smaz()`. Tato funkce zajistí správné smazání objektu, který jí je předaný jako argument. Objektu navíc nastaví položku `pocet` na 0 a `trojuhelniky` na `NULL`.

Poznámka:

- Kód je možné testovat na libovolném binárním `.stl` souboru. Ty je možné stáhnout například na Internetu.
- Není potřeba se zabývat tím, zda je soubor, který načítáme, ve správném formátu.
- Použít můžete pouze knihovny `stdio`, `stdlib` a `stdint`.
- Pro práci se soubory používejte pouze funkce ze seminářů.
- Pokud pracujete na jiném operačním systému, než Windows, je možné, že přiložený binární soubor nebude uložen ve správném formátu. V tom případě mne kontaktujte.

Bodování:

- 5 bodů pokud bude program pracovat dle zadání v povinné části.
- +5 bodů za implementaci funkce `uloz()`.
- +3 body za implementaci funkce `sjednoceni()`.
- +2 body za implementaci funkce `smaz()`.

Upozornění:

- V případě jakýchkoliv nejasností ohledně zadání, neváhejte mě kontaktovat.
- Je nutno dodržet zadání! Včetně všech názvů funkcí, maker, struktury i položek struktury.
- Opisování netoleruji! A to ani od umělé inteligence.

Binární STL

Na začátku souboru je hlavička obsahující 80 bytů (uint8). S touto hlavičkou dále pracovat nebudeme.

Následuje počet trojúhelníků uložených ve 4 bytech (uint32).

Poté pro každý trojúhelník:

12 bytů pro normálový vektor ($3 \times \text{float}$)

12 bytů pro vrchol 1 ($3 \times \text{float}$)

12 bytů pro vrchol 2 ($3 \times \text{float}$)

12 bytů pro vrchol 3 ($3 \times \text{float}$)

2 byty pro kontrolní součet (uint16) – s ním také pracovat nebudeme.

Pro správné fungování použijte místo typu `int` typy `uint8_t`, `uint16_t` a `uint32_t` definované ve standardní knihovně `stdint`.

ASCII STL

Soubor vždy začíná řádkem

```
solid
```

Následuje libovolný počet výpisů trojúhelníků. Každý trojúhelník je zadán následujícími řádky.

```
facet normal ni nj nk
```

```
outer loop
```

```
vertex v1x v1y v1z
```

```
vertex v2x v2y v2z
```

```
vertex v3x v3y v3z
```

```
endloop
```

```
endfacet
```

`ni`, `nj`, `nk`, `v1x`, `v1y`, `v1z`, `v2x`, `v2y`, `v2z`, `v3x`, `v3y` a `v3z` jsou desetinná čísla.

Soubor je ukončen řádkem

```
endsolid
```