Projekt do předmětu **VGE – Výpočetní geometrie**

Triangulace polygonu

řešitelé: **Samuel Repka**, xrepka07
Marek Mudroň, xmudro04
Matej Kunda, xkunda00

Zadání

Po konzultácii sme mali možnosť vytvorenia krokovacieho progamu, ktorý by zobrazoval rôzne metódy triangulácie, programu na dopĺňanie dier v modeloch ako Blender addon, alebo programu triangulácie 3D polygónov ako Blender addon. Rozhodli sme sa pre poslednú možnosť a to vytvoriť addon v programe Blender, ktorý bude triangulovať polygóny. Výsledné zadanie vyzerá nasledovne:

1. Naštudovať algoritmus triangulácie polygónov v 3D priestore.
2. Implementovať algoritmus v jazyku Python ako addon do programu Blender.
3. Vykonať experimenty, otestovať fungovanie programu.
4. Vyhodnotiť výsledky.

Použité technologie

- Blender 3.5.
- Python 3 s knižnicami bpy, bmesh, numpy.

Použité zdroje

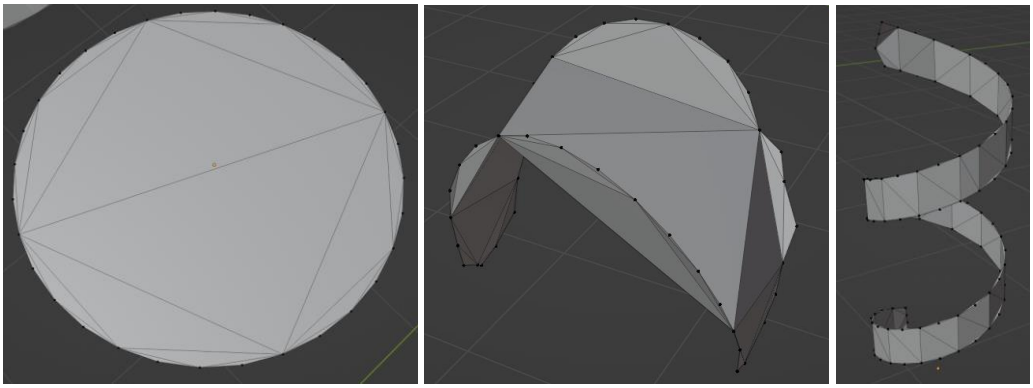
Použitý algoritmus, ktorý sme sa rozhodli implementovať je z tohoto článku:

ZOU, Ming. *An Algorithm for Triangulating 3D Polygons.* Washington, 2013. Diplomová práca. Washington University, School of Engineering and Applied Science. 2013-12-1. Vedúci práce Tao Ju. Dostupné z: https://openscholarship.wustl.edu/cgi/viewcontent.cgi?article=2212&context=etd

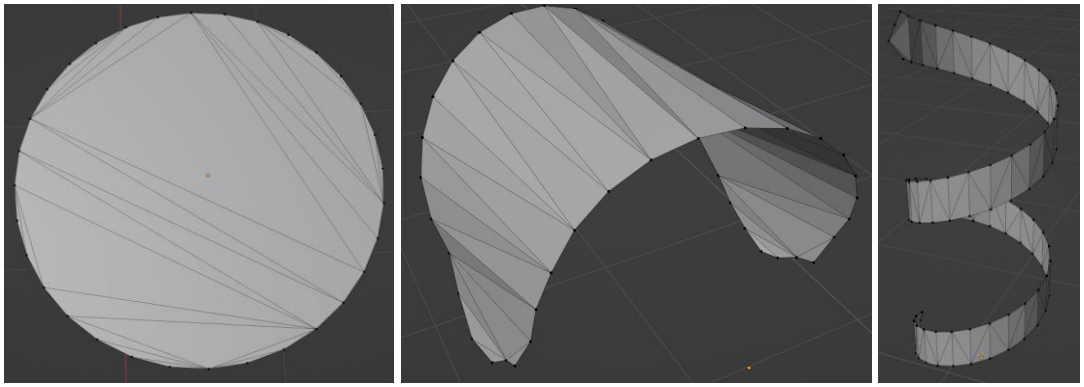
Nejdůležitější výstupy

Projekt obsahuje implementáciu metódy triangulácie polygónov v 3D. Implementované boli 4 váhovacie funkcie:

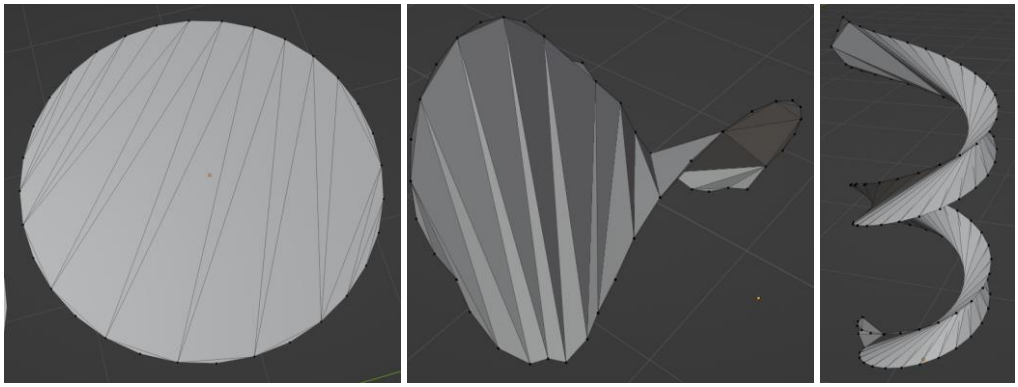
1. **DeLaunay triangulácia –** každý z trojuholníkov musí byť DeLaunay trojuholník.
2. **Minimálny obsah trojuholníka –** trojuholníky sú vyberané spôsobom, aby mali čo najmenší obsah.
3. **Maximálny uhol medzi dvoma trojuholníkmi –** váhovacia funkcia vyberá taký trojuholník, aby bol uhol medzi novým a existujúcim trojuholníkom najväčší.
4. **Minimálny uhol medzi dvoma trojuholníkmi –** rovnaký princíp ako pri maximálnom uhle medzi dvoma trojuholníkmi, ale uhol sa snaží minimalizovať.



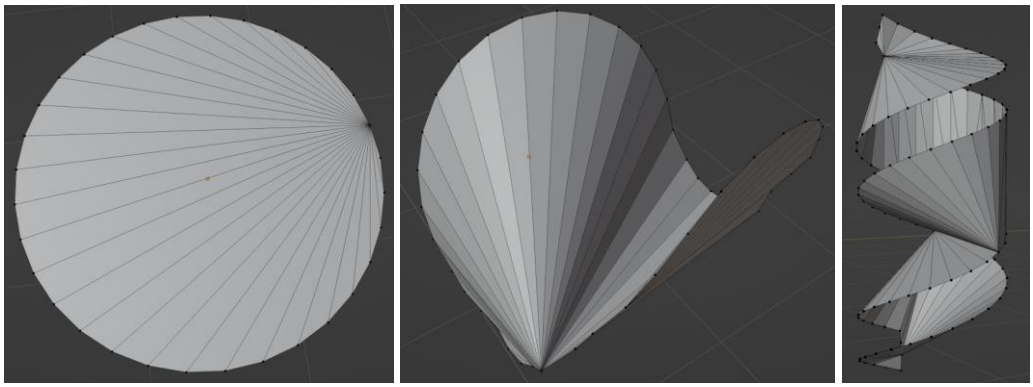
Polygóny triangulované metódou DeLaunay.



Polygóny triangulované metódou najmenšieho obsahu.



Polygóny triangulované metódou maximalizovania uhla medzi dvoma trojuholníkmi.

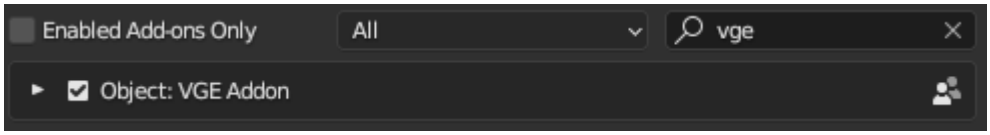


Polygóny triangulované metódou minimalizovania uhla medzi dvoma trojuholníkmi.

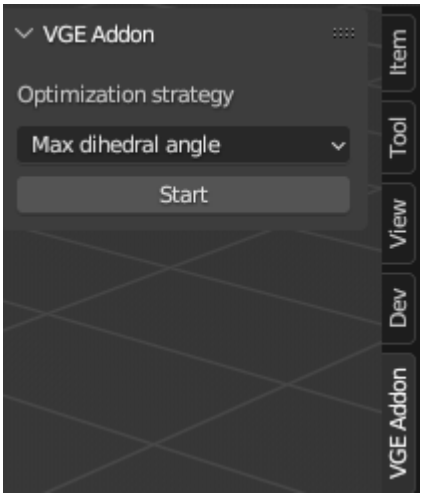
Ovládání vytvořeného programu

Najprv je nutné pripojiť k Blendru vytvorený addon. To je možné urobiť nasledovným spôsobom:

1. Prejdite do okna Edit → Preferences → Add-ons.
2. Kliknite na tlačidlo install a vyberte cestu k .zip súboru s addonom.
3. Následne zo zoznamu povoľte vytvorený addon s názvom „VGE Addon“.



Po úspešnom pripojení addonu do Blendru sa v edit móde po stlačení klávesy N zobrazí na pravom boku panel, kde po kliknutí na tlačidlo VGE Addon sa zobrazí panel s výberom metódy a tlačidlom Start.



Užívateľ následne musí vybrať všetky vertexy polygónu ktorý chce triangulovať, vybrať metódu a po stlačení tlačidla Start sa algoritmus spustí.

Rozdělení práce v týmu

Samuel Repka, Marek Mudroň: študovanie algoritmu, implementácia

Matej Kunda: študovanie algoritmu, testy, dokumentácia

Co bylo nejpracnější

Na začiatku projektu sme museli byť trpezlivý s porozumením cudzieho algoritmu triangulácie. Následne pri implementácii bolo najpracnejšie debugovanie programu.

Zkušenosti získané řešením projektu

- Zdokonalili sme sa v schopnosti čítať vedecký článok a analyzovať cudzí algoritmus.
- Zoznámili sme sa s Blender API.
- Zlepšili sme sa v organizovaní spoločných stretnutí, diskusii problémov a ich riešení.

Autoevaluace

Koncept řešení: 95% (analýza, výběr článků, dekompozice problému, volba vhodných prostředků, …)

Koncept prebehol bez problémov. Ako malý nedostatok vnímam len to, že sme implementovali len časť práce.

Realizace: 100% (kvalita získaných znalostí, kvalita a čitelnost kódu, obecnost řešení, znovupoužitelnost, …)

Kód je prehľadne modularizovaný a riadne komentovaný. Umožňuje jednoduché pridanie nových váhovacích funkcií. Addon má potenciál byť reálne využitý, ak sa rozhodneme zverejniť implementáciu.

Využití zdrojů: 60% (využití existujícího kódu a dat, využití literatury, …)

Literatúru sme využili najviac ako sme mohli, napojenie na Blender GUI bolo inšpirované existujúcim kódom. Samotný algoritmus sme museli implementovať sami.

Hospodaření s časem: 100% (rovnoměrné dotažení částí projektu, míra spěchu, chybějící části, …)

Projekt sme dokončili v niekoľkodennom predstihu a implementovali sme všetko čo sme plánovali.

Spolupráce v týmu: 90% (komunikace, dodržování dohod, vzájemné spolehnutí, rovnoměrnost, …)

Na začiatku projektu bol mierny problém s deľbou práce, avšak tento problém sme prekonali v neskoršej fáze projektu.

Celkový dojem: 90% (pracnost, získané dovednosti, užitečnost, volba zadání, cokoliv, …)

Projekt sme zvádli v dočasnom predstihu s funkcionalitou, ktorú sme si na začiatku určili. Algoritmus, ktorý sme implementovali nám je jasný a sme si istý v oblasti jeho fungovania. Výsledný kód je členený do tried, funkčný, testovaný, komentovaný a umožňuje jednoduché rozširovanie.

Doporučení pro budoucí zadávání projektů

Na projekte nám vyhovovala voľnosť výberu konkrétneho zadania, kedy nám na prvej konzultácii dal vedúci projektu zopár nápadov, čo by sme na túto tému mohli spracovať.