

Projekt do předmětu VGE – Výpočetní geometrie

Triangulace polygonu

řešitelé: Marcel Kiss, xkisss01
Michal Ondrejů, xondre08

Zadání

- Detailnejšie pochopiť princíp triangulácie
- Implementácia triangulácie a demonštrácia algoritmu
- Triangulátory v praxi a netriviálne polygóny

Použité technológie

Počas vývoja aplikácie boli použité Git a CMake. Aplikácia beží na OpenGL 4.1 v aktívnej renderovacej slučke kvôli animáciám.

Potrebné externé knižnice na preloženie projektu:

- GLEW: (<http://glew.sourceforge.net>) – Inicializácia moderného OpenGL API pre Windows
- GLM: (<https://glm.g-truc.net/0.9.9/index.html>) – Vektory, matice, matematika pre geometriu
- SDL2: (<https://www.libsdl.org/download-2.0.php>) – Okno, OpenGL kontext, spracovanie GUI eventov

Aplikácia využíva 2 knižnice (Poly2Tri a Clipper), ktoré sú súčasťou projektu vo forme zdrojových súborov. Cmake vytvorí druhý target 'Libraries', kde zahrnie všetky priložené externé zdroje a zlinkuje ich do hlavného targetu.

Použité zdroje

V časti aplikácii sme použili dve knižnice Poly2Tri na trianguláciu a Clipper na kalkuláciu jednoduchých polygónov. Cieľom je demonštrácia knižníc na problém zobrazenia arbitrárneho útvaru cez trojuholníky:

- Poly2Tri: <https://github.com/greenm01/poly2tri>
 - Sweep-line algorithm for constrained Delaunay triangulation
<https://www.tandfonline.com/doi/abs/10.1080/13658810701492241>
- Clipper: <http://www.angusj.com/delphi/clipper.php>

Pomocné zdroje:

- <http://vterrain.org/Implementation/Libs/triangulate.html>

Exaktný zdrojový kód funkcie **almost_equal()** zo stránok **cppreference.com**. Dôvod použitia je detekcia a odstránenie duplicitných bodov počas procesu triangulácie.

- http://en.cppreference.com/w/cpp/types/numeric_limits/epsilon

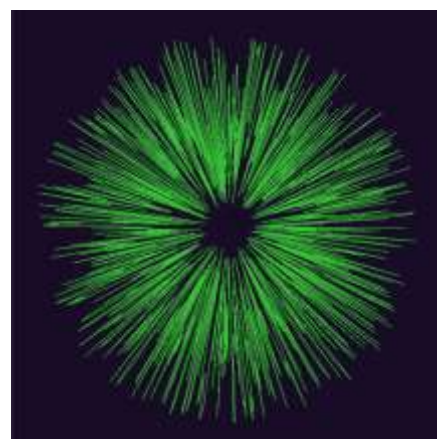
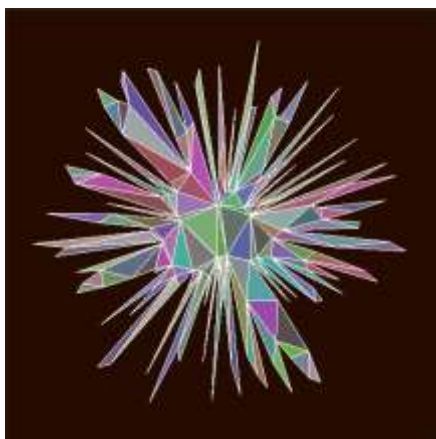
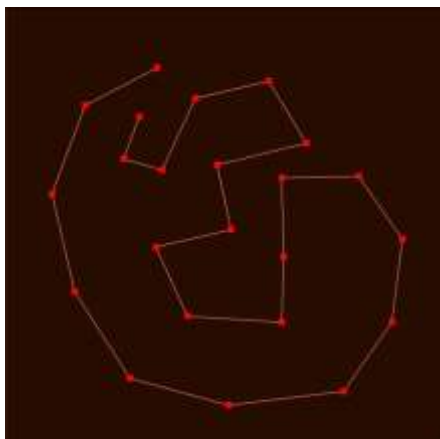
Ear clipping triangulator:

- <http://www.personal.kent.edu/~rmuhamma/Compgeomerty/MyCG/TwoEar/two-ear.htm>
- <https://www.geometrictools.com/Documentation/TriangulationByEarClipping.pdf>

Nejdůležitější výstupy

Interaktívna aplikácia, ktorá vizualizuje trianguláciu polygónov, vhodná na demonštráciu jednoduchých spracovania polygónov. Umožňuje generovať mnoho-bodové polygóny a merať čas triangulácie vlastnej implementácie triangulátora metódou Ear-Clipping a implementácie knižníc. Spojením dvoch samostatných knižníc dokáže netriviálne polygóny rozbiť na formu vhodnú triangulácie.

Triangulácia metódou Ear-Clipping umožňuje postupné krokovanie a zobrazuje body, ktoré sú aktuálne spracovávané.



Ovládání vytvořeného programu

Klik ľavou myšou – Zadávanie bodov polygónu

Klik pravou myšou – Zadávanie bodov diery

ESC, dvojklik ľavou myšou – Ukončenie editácie

1 – prepnutie na EarClipping Triangulátor.

2 – prepnutie na Poly2Tri triangulátor.

3 – prepnutie na Clip2Tri triangulátor.

WSAD – posúvanie sa v rámci kanvasu.

G – Vygenerovanie polygónu s nastavaným počtom bodov (genCount).

_ (Space) – Triangulácia zatvoreného polygónu. Na stdout vypíše čas triangulácie.

R – Reset polygónu, vymazanie aktuálneho výsledku a reset triangulátorov

B – Znižovanie počtu generovaných bodov (genCount) o 25.

N – Zvyšovanie počtu generovaných bodov (genCount) o 25.

X – Štart / Stop samostatného generovania, merania a zvyšovania počtu vygenerovaných bodov.

, (čiarka) – Jeden krok triangulácie EarClipping.

. (bodka) – Preskočenie krokov triangulácie EarClipping do ďalšieho nového trojuholníka.

Rozdelení práce v týmu

Marcel Kiss: výroba CMakeLists.txt, štruktúra projektu, zakladanie Git repozitára, pomocné OpenGL funkcie, vykresľovanie polygónov, návrh štruktúry programu, fúzia knižníc Poly2Tri a Clipper do triangulátora, časovače a merací mechanizmus

Michal Ondrejó: Triangulácia metódou ear clipping.

Co bylo nejpracnější

Vytvoriť cross-platform CMake, spojzduň OpenGL pre Windows cez knižnicu GLEW.

Zkušenosti získané řešením projektu

Triangulačné algoritmy majú veľa limitácií a sú nestabilné pri zlých vstupných dátach. Samostatne nedokážu riešiť netriviálne polygóny, ktoré sa väčšinou vyskytujú v praxi (arbitrárne útvary zadané užívateľom). Napríklad pri zobrazení bėzierových útvarov pomocou trojuholníkov je nevyhnutné najprv útvar teselovať na netriviálny polygón, ktorý musí prejsť fázou zjednodušenia, rozbitia na jednoduché polygóny. Rozbitím na triviálne polygóny vzniká viac možností vyhodnotenia netriviálnych polygónov podľa winding-rule (even-odd, non-zero, positive, negative). Mōžu tak vzniknúť diery v jednoduchých polygónoch, ktoré však podporujú len niektoré algoritmy.

Ďalším problémom je duplicita bodov, efektívne algoritmy nekontrolujú či je bod zhodný z iným bodom alebo sú blízko pri sebe. Počas procesu triangulácie mōže algoritmus zlyhať zasiahnutím do pamäte alebo zlou aritmetikou. Kontrolu 'blízky' bodov je nutné prevádzať pred trianguláciou.

Knižnica Clipper používa zaujímavý prístup odstránenia zhodných bodov a redukovanie implementačných chýb tým, že polygónové body reprezentuje v celých číslach. Každý vstupný bod je pred použitím vynásobí faktorom presnosti, ktorý udáva počet desatinných miest ako presnosť výpočtu. Body po klipnutí sa podelia týmto faktorom na pôvodný priestor.

Triangulácia pomocou metódy Ear clipping postupne prechádza body trojuholníka, a odstriháva „uši“. Na nájdenie ucha polygónu je potrebné zistiť, či je v danom bode polygón konvexný a či sa nenachádza vnútri trojuholníka niektorý z bodov. Algoritmus pre nájdenie ucha má zložitosť $O(kn)$, kde k označuje počet konkávných uhlov a teda v najhoršom prípade dosahuje zložitosť $O(n^2)$. Pre trianguláciu polygónu je teda zložitosť $O(n^3)$

Autoevaluace

Koncept řešení: 90% (analýza, výběr článků, dekompozice problému, volba vhodných prostředků, ...)

Realizace: 95%

Návrh aplikace zahrnuje rychle přidání dalších triangulátorů na porovnání. Implementácia triangulácie je prehľadná a je možné ju znovu použiť.

Využití zdrojů: 95% (využití existujícího kódu a dat, využití literatury, ...)

Využitie externých efektívne implementovaných algoritmov, online recenzie a porovnania rôznych triangulačných knižníc.

Hospodaření s časem: 90% (rovnoměrné dotažení částí projektu, míra spěchu, chybějící části, ...)

Vzhľadom na nával práce s DP, mal projekt menšiu prioritu. S menšou časovou tiesňou by sa mohlo implementovať viac triangulátorov.

Spolupráce v týmu: 100% (komunikace, dodržování dohod, vzájemné spolehnutí, rovnoměrnost, ...)

Celkový dojem: 100% (pracnost, získané dovednosti, užitečnost, volba zadání, cokoliv, ...)

Aplikácia vyzerá pekne, interakcia s animáciami je príjemná a ako demo je postačujúce.

Triangulácia arbitrárnych polygónov je zaujímavá téma a ešte viac s myšlienkou bézierových útvarov.

Doporučení pro budoucí zadávání projektů

Zadanie projektu nie je moc špecifické, ťažko sa odhaduje na akej úrovni má byť spracovanie témy. Z pohľadu time managementu to nie je moc príjemné ale oceňujeme voľnosť.