

# Seminár k záverečnej práci

Patrik Sakáč

3Ib, 2017 - 2018

**Abstrakt.** Práca sa zaoberá rastom stromov, kríkov v počítačových hrách pomocou L-systémov. Naším riešením je vytvoriť obrázok pomocou L-systémov, ktorý bude uchovávať dôležité informácie o raste daného druhu. Tento obrázok následne použiť a implementovať rast.

**Kľúčové slová:** L-systém, vrchol, mesh, shader,

## 1 Úvod

V hrách je rast stromov, kríkov a iných rastlín bežným problémom, s ktorým sa vývojári stretávajú. Pri veľkom počte inštancií sa upúšťa od tejto funkcionality, pretože si vyžaduje veľké nároky na operačnú pamäť, procesor či grafickú kartu. Alebo sa znižujú nároky na konečný grafický vzhľad.

Existuje veľa rôznych riešení, ktoré boli použité pri implementáciách v jednotlivých hrách. Najznámejšie riešenia sú dve. Prvým je nahradenie celých objektov a druhým animácia jedného objektu pomocou kostí alebo pomocou vrcholov<sup>1</sup>. Spomenuté riešenia majú svoje výhody aj nevýhody. Výhodou je jednoduchá implementácia pre vývojárov. Nevýhodou je najmä zložitá výroba pre grafikov. Pri riešení nahradzovaním objektov je pri každom nahradení objektu výrazná zmena pre oko hráča. Riešenie pomocou animovaných objektov, ktoré animátor vytvoril pomocou kostí je optimálnejšie ako pomocou vrcholov, ale je omnoho zložitejšie vytvoriť takúto animáciu. Animácie majú nevýhodu, že rovnaká animácia v inej fáze prehrávania sa považuje za novú a teda musí byť dostupná v operačnej pamäti. Pri väčšom počte animácii ľahko dosiahneme veľké nároky na operačnú pamäť. Preto chceme implementovať riešenie, ktoré bude optimálne a implementácia či výroba nebude časovo náročná. Použijeme pri tom fraktály a to konkrétne L-systémy, ktoré sú určené na modelovanie rastlín a živých organizmov.

L-systémy predstavujú gramatiku, ktorá generuje reťazce. Tieto reťazce v sebe nesú informácie o vykreslení daného obrazca. Poznáme viacero druhov L-systémov napríklad parametrické, stochastické, bezkontextové a iné. Pri vykresľovaní sa používa tzv. korytnačia grafika. Každý symbol v reťazci má svoj význam a presne určuje, čo sa má vykonávať pri vykresľovaní.

Naším cieľom je zoznámiť sa s týmito L-systémami, implementovať niektoré druhy hlavne parametrické L-systémy. Pomocou týchto L-systémov vykresliť textúru<sup>2</sup>

---

<sup>1</sup> Vrchol je trojica desiatinných čísel, ktoré určujú pozíciu, kde sa nachádza.

<sup>2</sup> Lubovoľný obrázok.

ktorá bude predstavovať tzv. masku<sup>3</sup>. Následne vhodne použiť túto masku pri raste rôznych druhov stromov, kríkov. Zanalyzovať scénu pri veľkom počte inštancií.

### 1.1 Ilustračný príklad

Základný typ L-systému je tzv. *DOL-systém*. *D* znamená deterministický a *0* znamená bezkontextový. Formálne je to trojica  $G = (\Sigma, S, P)$ , kde  $\Sigma$  je abeceda symbolov,  $S$  je štartovací symbol z abecedy,  $P$  je množina prepisovacích pravidiel.

Príklad

$\Sigma = \{0, 1, [, ]\}$

$S = 0$

$P = \{(1 \rightarrow 11), (0 \rightarrow 1[0]0)\}$

1. iterácia: 1[0]0

2. iterácia: 11[1[0]0]1[0]0

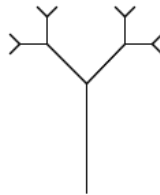
3. iterácia: 1111[11[1[0]0]1[0]0]11[1[0]0]1[0]0

0 – kresli čiaru (skonči v liste)

1 – kresli čiaru

[ - vlož do zásobníka a otoč o 45° vľavo

] – vyber zo zásobníka a otoč o 45° vpravo



Obrázok 1. Vykreslená tretia iterácia.

### 1.2 Prehľad súčasného stavu

Sú implementované základné a parametrické L-systémy. Je vytvorený testovací objekt stromu, ktorý sa animuje na grafickej karte pomocou prelínania meshov<sup>4</sup>. Taktiež je vytvorený shader<sup>5</sup>, ktorý pomocou informácií v maske vytvára rast konárov a listov. Je vytvorená prvá maska pomocou implementovaných L-systémov. Je vytvorená

<sup>3</sup> Obrázok obsahujúci rôzne informácie v každom kanáli. Nezáleží ako ten obrázok bude vyzeráť.

<sup>4</sup> Objekt alebo model reprezentovaný skupinou vrcholov, trojuholníkov a niektorých ďalších informácií.

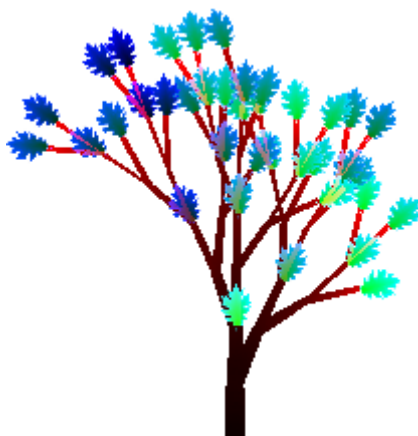
<sup>5</sup> Súbor s pravidlami a funkciami pre grafickú kartu, ktorým definujeme pokročilejšie funkcie pri zobrazovaní objektov.

testovacia scéna s väčším počtom inštancií. Ďalej je potrebná úprava kódu, zjednodušenie a oprava nejakých nedostatkov.

## 2 Návrh riešenia

Riešenie sa skladá z viacerých častí. Prvou časťou je vygenerovať reťazec, kde dostaneme vstupné parametre. Vstupné parametre sú pravidlá, štartovací reťazec a počet iterácií. Chceme implementovať parametrické generovanie reťazcov, takže počas generovania budeme matematicky vyhodnocovať reťazce. Výsledný reťazec je vstupom pre nasledujúcu časť.

Nasledujúcou časťou je vykreslenie. Vykreslenie rozdelíme na prípravu dát a samotné vykreslenie. Príprava dát spočíva vo vytvorení objektových údajov reprezentujúcich konár (čiaru), list, spolu s ich vlastnosťami ako sú pozícia, rotácia, veľkosť. Vykreslenie je už len grafické zobrazenie údajov. Zobrazené údaje následne uložíme do súboru s príponou “.png”. Tento súbor budeme nazývať maska. Maska obsahuje v každom kanáli inú informáciu. V červenom je postupný rast konára. V zelenom je informácia o tom, že kedy ma začať rásť list. V modrom je postupný rast listu.



Obrázok 2. Textúra masky.

Ďalšou časťou je samotný rast. Implementovať to budeme v hernom vývojovom prostredí Unity. Unity poskytuje pre vykresľovanie široké možnosti. Každý objekt, ktorý sa v scéne nachádza a má byť viditeľný na kamere obsahuje nejaký “Render Component”. Na naše rastúce objekty použijeme konkrétne “Mesh Renderer”, ktorého súčasťou je materiál<sup>6</sup>. Materiál obsahuje vstupné dáta pre shader a samotný shader. Našu masku použijeme ako vstup pre shader, pomocou takéhoto materiálu. Potrebujeme si napísať vlastný shader, ktorý bude čítať informácie z masky a

---

<sup>6</sup> Materiál je dátová štruktúra obsahujúca informácie ako sú obrázky, farby, čísla, vektory a shader.

zobrazovať rastúci strom alebo krík. Použijeme štandardný model PBS<sup>7</sup>, ktorý poskytuje Unity. V shadery si definujeme funkciu na úpravu vrcholov, aby sme mohli presúvať pozíciami vrcholov. Presúvať vrcholmi potrebujeme aby sme vedeli prelínať medzi dvoma meshmi. Pre výpočet výslednej “surface”<sup>8</sup> štruktúry použijeme vzorec z jednotlivých kanálov masky a farebných zložiek. Prelínanie medzi dvoma meshmi<sup>9</sup> vieme spraviť, len ak máme informácie z obidvoch meshov. Preto potrebujeme program ktorý nám vytvorí mesh obsahujúci informácie aj o nasledujúcom meshy. Výstupom tohto programu bude jeden mesh, ktorý obsahuje dodatočné informácie. Problémom je, že máme k dispozícii šesť voľných desiatinných čísel a potrebujeme 9 desiatinných čísel. V meshi sú voľné uv2,uv3,uv4<sup>10</sup> mapovania, ktoré použijeme na prenos potrebných informácií. Potrebné informácie sú pozícia, normála<sup>11</sup> a tangenta<sup>12</sup> vrcholu. Pozíciu vrcholu uložíme pôvodnú, ale zvyšné čísla zakódujeme z dvoch desiatinných čísel do jedného tak, aby sme neskôr v shadery ich vedeli odkódovať. Vo finálnom shadery budeme dvoma hodnotami ovládať prelínanie meshov, čo predstavuje rast hlavného kmeňa stromu spolu s konármi a rast jednotlivých listov. Niektoré kríky môžu vynechať prelínanie meshov a použijú iba rast pomocou masky. Na koniec je potrebné vytvoriť scénu, kde bude možné zanalyzovať implementáciu celkového rastu na väčšom počte inštancií. Prípadne porovnať so známou implementáciou pomocou animácií.

### 3 Záver

Zatiaľ sa spravili jednotlivé časti, ktoré sú vyššie spomenuté. Každá časť je implementovaná a je potrebné len doladiť niektoré detaily. Rovnako je potrebné zjednodušiť niektoré implementácie. Je potrebné zanalyzovať scénu s väčším počtom inštancií a porovnať s implementáciou pomocou animácií.

---

<sup>7</sup> PBS je skratka pre Physically-Based Shading, čo simuluje interakcie medzi materiálmi a svetlom realistickým spôsobom.

<sup>8</sup> Surface je výstupná štruktúra, ktorá je následne vykreslená grafickou kartou.

<sup>9</sup> Prelínanie dvoch meshov sa nazýva “Blend Shapes”, ktoré sa najbežnejšie využíva pri mimických výrazoch osôb.

<sup>10</sup> uv2,uv3,uv4 predstavujú dvojicu desiatinných čísel, ktoré sa používajú pri mapovaní obrázka na nejaký objekt (model).

<sup>11</sup> Normála je trojica desiatinných čísel, ktoré určujú normálový vektor vrcholu.

<sup>12</sup> Tangenta je štvorica desiatinných čísel, ale pre nás potrebné sú len prvé 3 čísla, ktoré predstavujú dotyčnicový vektor pre vrchol.

## **Literatúra**

1. Pedrosa D.S.: GALSYS - Procedural Creation of Trees - through Combination of Genetic Algorithms and Lindenmayer Systems, 2014, ISBN 978-3639643763
  2. Quigley E., et al.: Real-time Interactive Tree Animation, IEEE Transactions on Visualization and Computer Graphics Vol 22, 2017
  3. Fornander P.: Game Mechanics Integrated with a Lindenmayer System, thesis, School of Computing, Blekinge Institute of Technology, Sweden, 2013
  4. Hornby G.S., Pollack J.B.: Evolving L-systems to generate virtual creatures, Computer & Graphics (25) 6, 2001, pp. 1041-1048
  5. Trube B.: Fractals - A Programmer's Approach, Amazon Digital Services, 2013
- Prusinkiewicz P., Hanan J.: Lindenmayer systems, fractals, and plants, Springer Science & Business Media, Springer Verlag, 1989, ISBN 978-0387970929