Milan Mroczkowski
sata150@gmail.com
yesit.cz

Obrázek: řádkové vyplňování převzdato z UJEPu.

Globální osvětlovací metody

- aby scéna vypadala dobře, potřebujeme hodit hodně fotonů do scény, tyto všechny fotony sledujeme, jsou zde několika násobné odrazy. Náročné na počítač, na začátku hry se vzhled 3D scény uloží (vyrenderuje). Když pak jdu postavičkou, dopomáhám si lokálními osvětlovacími metodami.

PHONGŮV OSVĚTLOVACÍ MODEL
- empirický vyhodnocovací model, není na fyzikální podstatě

Sečtu odrazy:
Dokonalý zrcadlový odraz: úhel dopadu = úhel odrazu, intenzita největší (závisí na intenzitě a úhlu dopadu)
Dokonalý difúzní odraz: paprsek, který dopadl na povrch se šíří do všech stran stejnou intenzitou. (závidí na intenzitě)
Teplé ambientní (okolní) světlo: malé osvětlení za koulí, aby nebyla absolutně černá

násobím empirickým koeficientem menším než 1.

MONTE CARLO
- vygeneruji náhodné číslo s určitou pravděpodobností, foton se láme s určitou pravděpodobností.
- používáme ROZEHRÁNÍ NÁHODNÉ VELIČINY

SLEDOVÁNÍ FOTONŮ
- sleduji foton vržený do scény: jak odrážet, co bude dělat na poloprůhledném povrchu, těžko zjistím všechny možnosti kudy může jít.
- připravím si sled na základě vygenerovaného náhodného čísla – např.: nejpravděpodobněji se odrazí zrcadlově, pak difúzně, málo pravděpodobné: dovnitř materiálu.

Lokální osvětlovací modely

- nevšímají si: osvětlení jako celku
- nedokonalé vystínování tělesa: slunce osvítí pixel, bude světlí, konec

RENDEROVÁNÍ SNÍMKŮ: vypočítá intenzitu pixelu podle geometrie, závisí na:
- zdroji světla
- úhel pohledu
- normále (přímka kolmá na daný prostor) - je určena normálovým vektorem

Milan Mroczkowski
sata150@gmail.com
yesit.cz

Redukce barevného prostoru

- z původního obrázku chceme, aby na obrázku byla pouze bílá a černá (bez šedi)

Metody:

a) POLOTÓNY: podle intenzity kolečka (větší intenzita – větší kolečko)

b) ÚPRAVY GRADAČNÍ KŘIVKOU: ovlivňuje souvislost mezi jasem a kontrastem, transformace barev, Photoshop

c) PRAHOVÁNÍ: pixel porovnávám s odstínem šedi (prahová hodnota), intenzita větší než ½ – bíle.
Ztráta detailů, vidět rastr.

POMOCÍ MATICE: každý pixel nahradím čtveřicí pixelu, pokud je ze čtvrtiny bílí, nahradím ho maticí – z dálky bude vypadat jako šedá.

d) NÁHODNÉ ROZPTÝLENÍ
Vygeneruji si čísla a budu porovnávat s náhodnou hodnotou od nula do jedné.
Původní pixel s větší intenzitou než náhodná hodnota – bílí.

Přejdu o pixel doprava a znovu vygeneruji náhodné číslo. Velký šum, vhodný pro větší obrázky.

e) BAYERŮV FILTR – přiřadí hodnotu podle intenzity od 0 do 1.
Pixel 0,7 – vybarví se jako 1 (0,3 zbytek)
U sousedních pixelů odebereme 0,3

Vylepšení obrázku:

PŘEVZORKOVÁNÍ – histogram v digitální podobě nahradím analogovou funkcí. S analogovou funkcí provádím matematické operace. Analogové hodnoty zpracuji a znovu provedu převzorkování. Dostanu vhodnější histogram.

(neumíme přímo pracovat s digitálním signálem)

WARFING – protáhnout kočku nebo vytvořit jiné zvíře při prolínání obrazů

MORFING – warfing provedu na 2 obrázcích, hodně prohnu, na alfa kanálu nastavím prolínání

Milan Mroczkowski
sata150@gmail.com
yesit.cz

Vyšrafování obrázku barvou (stínování)

KONSTANTNÍ STÍNOVÁNÍ – podle intenzity vybarvil plošku

GOURAUDOVO STÍNOVÁNÍ – na úsečkách provedu lineární interpolaci, uvnitř plochy použiji bilineární interpolaci s kterou vybarvím. (Plynulejší barevné přechody.)

PHONGOVO STÍNOVÁNÍ – bilineární interpolací spočítám v každém bodě normálu, pak spočítám intenzitu pomocí Phongova osvětlovacího modelu.

Kde nedopadá světlo (není vidět ze zdroje světla) : stíny

PSEUDOSTÍN – nepravý stín, ve tvaru elipsy tmavší oblast, postava člověka
VRŽENÝ STÍN – objekt vrhá stín na ostatní
VLASTNÍ STÍN – objekt vrhá stín sám na sebe

Nejdokonalejší zobrazení stínů pomocí: GLOBÁLNÍCH ZOBRAZOVACÍCH METOD. Řešení stínů převedu na řešení viditelnosti.

Použiji Z-BUFFER pro zjištění viditelnosti, pak pro zjištění zda je objekt ve stínu.

Textury

(stíny)

- jako bitmapa, kterou nanesu na povrch
- základní prvek textur pro povrchy (rovinu nebo prostorový objekt 3D): TEXEL

- vytvářet: mlhy, mohu ovlivňovat normálu povrchu

TEXTURY JEDNOROZMĚRNÉ: opakující se déšť
TEXTURY DVOUROZMĚRNÉ: pláštěnka na člověku ve hře
TEXTURY TROJROZMĚRNÉ: šachová figurka, barva definována ve všech bodech prostoru

Textury podle vlastností:
DIFÚZNÍ TEXTURY – všímají si barvy povrchu
NORMÁLOVÉ TEXTURY – mění tvar objektu
ODRAZOVÉ TEXTURY – dám tam odrazovou plochu
HYPERTEXTURA – modelování ohně, určuje optické vlastnosti nad povrchem objektu

Mapování textur – nanášení textur na povrch těles:
PROCEDURÁLNÍ TEXTURA – pomocí fraktálu
HRBOLATÁ TEXTURA – vytváříme hrboly na obrázku, je to rychlejší než užití sítí trojúhelníků, kvalitou lepší.

Trojrozměrné modely a plochy

HERMITOVSKÉ KUBIKY
COONSOVY KUBIKY
BEZIEROVY KUBIKY

Fraktály

PROCEDURÁLNÍ MODELOVÁNÍ

Máme nějaký **předpis** a podle něj tvoříme.
Jejich popis je založen na **Braunově pohybu** – na **náhodném čísle** to závisí, který vygenerujeme.
Bod se pohybuje do **všech směrů s** určitou **pravděpodobností nebo** na základě **matematických vztahů** (řekneme želvo nahoru).

Modelujeme (do nekonečna): **rostliny**, mlhu, erozi, …

KOCHOVY VLOČKY
SHARPINSKÝ FRAKTÁL

3D scéna

Mám geometricky popsanou scénu.
chci: objekt blíž vidět, objekt dál nevidět

Rastrové algoritmy:
MALÍŘŮV ALGORITMUS: nakreslím pozadí, pak popředí, jen se překreslí, nefunguje dál, blíž.

Z-BUFFER – **paměť hloubky** (grafická karta), u objektu nastavím RGB + hodnotu Z (vzdálenost od kamery, koukám ze shora), ukládá se nejbližší vzdálenost. Defaultní hodnota nekonečno.

Vykreslují se ze Z-bufferu 4 matice:
barvy R G B + matice nejbližší vzdálenosti k nám

Plošková reprezentace:
Plošky rozdělím na dva typy:
Přivrácené – vektor normály směřuje k pozorovateli
Odvrácené – nezajímají nás (od pozorovatele)

Plochy se setkávají na hranách:
hrany zadní – svírají odvrácené plochy, nevidím plochy
hrany přední – svírají přivrácené plochy, vidím plochy
hrany obrysové – svírá přední a odvrácenou plochu – hrana mezi viditelnou a neviditelnou plochou

Kouli vykreslujeme pomocí Nurbs křivek a Nurbs ploch. Pro hru ji programátoři převedou na síť trojúhelníků – zobrazováno na grafické kartě.

ROBERTŮV ALGORITMUS – vektorový algoritmus, testuje viditelnost hran.
Pokud nejde vidět celá hrana, rozdělí ji a testuje části hrany.

Dvourozměrné objekty, vykreslit:

při výpočtu používám celá čísla

KRUŽNICI, ELIPSU – pomocí symetrie, stačí vykreslit 1/8 kruhu. Využije se osová a středová souměrnost.

ÚSEČKU – příkaz DrawLine, **Bresenhamův algoritmus**
Maluji od prvního bodu (pixelu), který vybarvím. Postupuji doprava, když

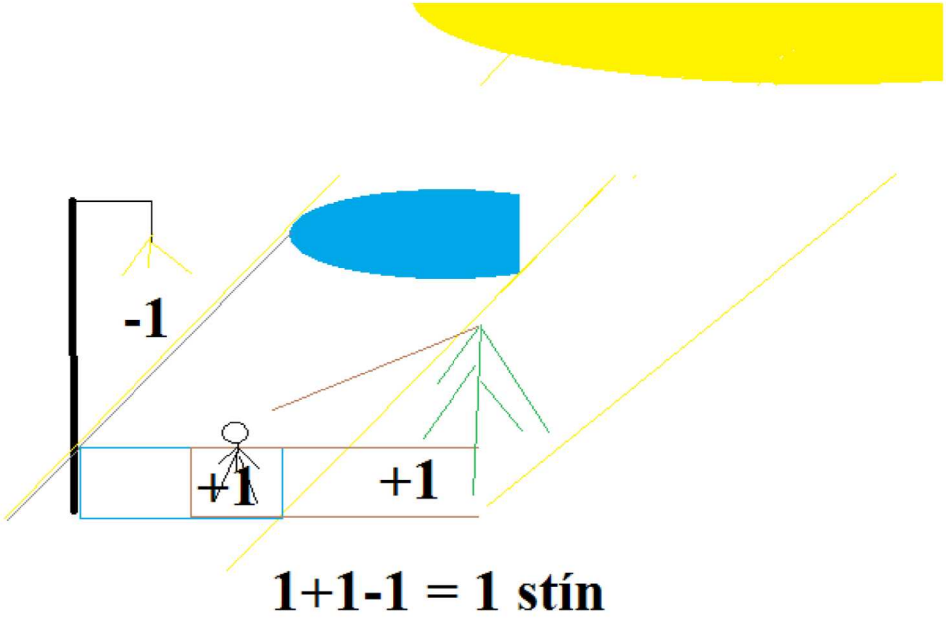
- čára bude svírat **s osou x úhel větší než 45 stupňů**, tak se vykreslí podle řídící osy nebo
- **podle vzdálenosti od středu**.

Počítání stínů

Máme více zdrojů světla.
STÍNOVÉ TĚLESO – ohraničuje prostor ze kterého je stín

Počítáme: Kolikrát jsme vlezli do stínového tělesa?
0 – objekt osvětlen

používáme stínovou paměť hloubky Z-BUFFER, pomocná matice – vzdálenosti od světelného zdroje, blíž osvíceno.



Odstranění šumu a ostření

operace k sobě inverzní

rozostřování – šumu ubývá
zaostřování – šumu přibývá

Položí se maska na obrázek, spočítá se průměrná intenzita (medián) a porovná se s intenzitou uprostřed masky, pokud tam bude jediná tmavá tečka, nahradí se tou intenzitou.

KONVOLUČNÍ MASKA – mívá vyšší hodnotu uprostřed, přiložím na obrázek s hodnotami jasu 0 až 255. Číslo vynásobím s koeficientem. (využívám sumu), běžně integrály

dále na konci pdf dokumentu

Vyplňování (vybarvování) oblastí:

ŘÁDKOVÉ VYPLŇOVÁNÍ

SEMÍNKOVÉ VYPLŇOVÁNÍ – testuji, pokud sousední pixely neleží na hranici, vybarvím a půjdu dál.

KONVOLUČNÍ MASKA

$$c[m,n] = a[m,n] \otimes b[m,n] = \sum_{j=0}^{J-1} \sum_{k=0}^{k-1} a[j,k], b[m-j, n-k]$$

KDE a JE KONVOLUČNÍ JÁDRO O ROZMĚRU $J \times K$, b je OBRAZ, KTERÝ CHCEME FILTROVAT

ORIGINÁLNÍ OBRAZ

8	12	18
6	**10**	12
1	4	5

$\times$ KONVOLUČNÍ JÁDRO

-1	-1	-1
-1	9	-1
-1	-1	-1

$\longrightarrow$

-8	-12	-18
-6	90	-12
-1	-4	-5

SEČTU VŠE

$-8-12-18-6+90-12-1-4-5=24$

VÝSLEDNÝ OBRAZ

	24	

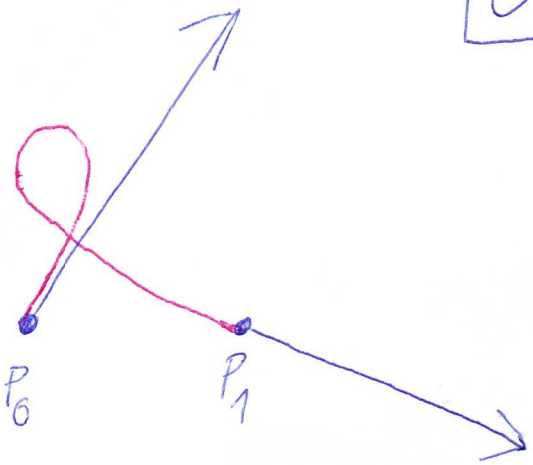
KUBIKA = 3. STUPEŇ
C3
ZARUČUJE SPOJITOST
1. A 2. DERIVACE

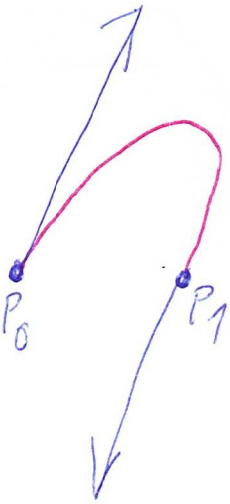
HERMITOVSKÉ KUBIKY

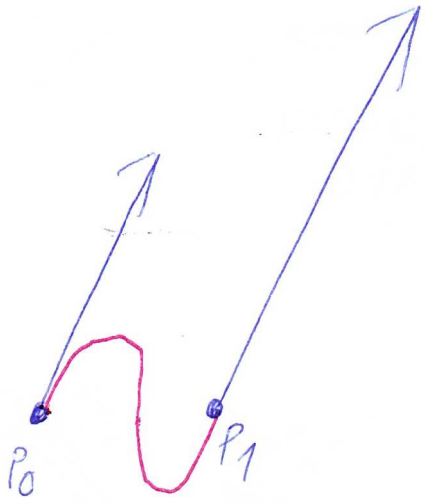
2 BODY (POČÁTEČNÝ, KONCOVÝ) ŘÍDÍCÍ
2 TEČNÉ VEKTORY

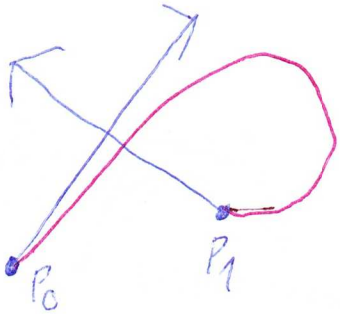
TVAR KŘIVKY OVLIVŇUJE: VELIKOST
SMĚR
TEČNÝCH VEKTORŮ

C^1





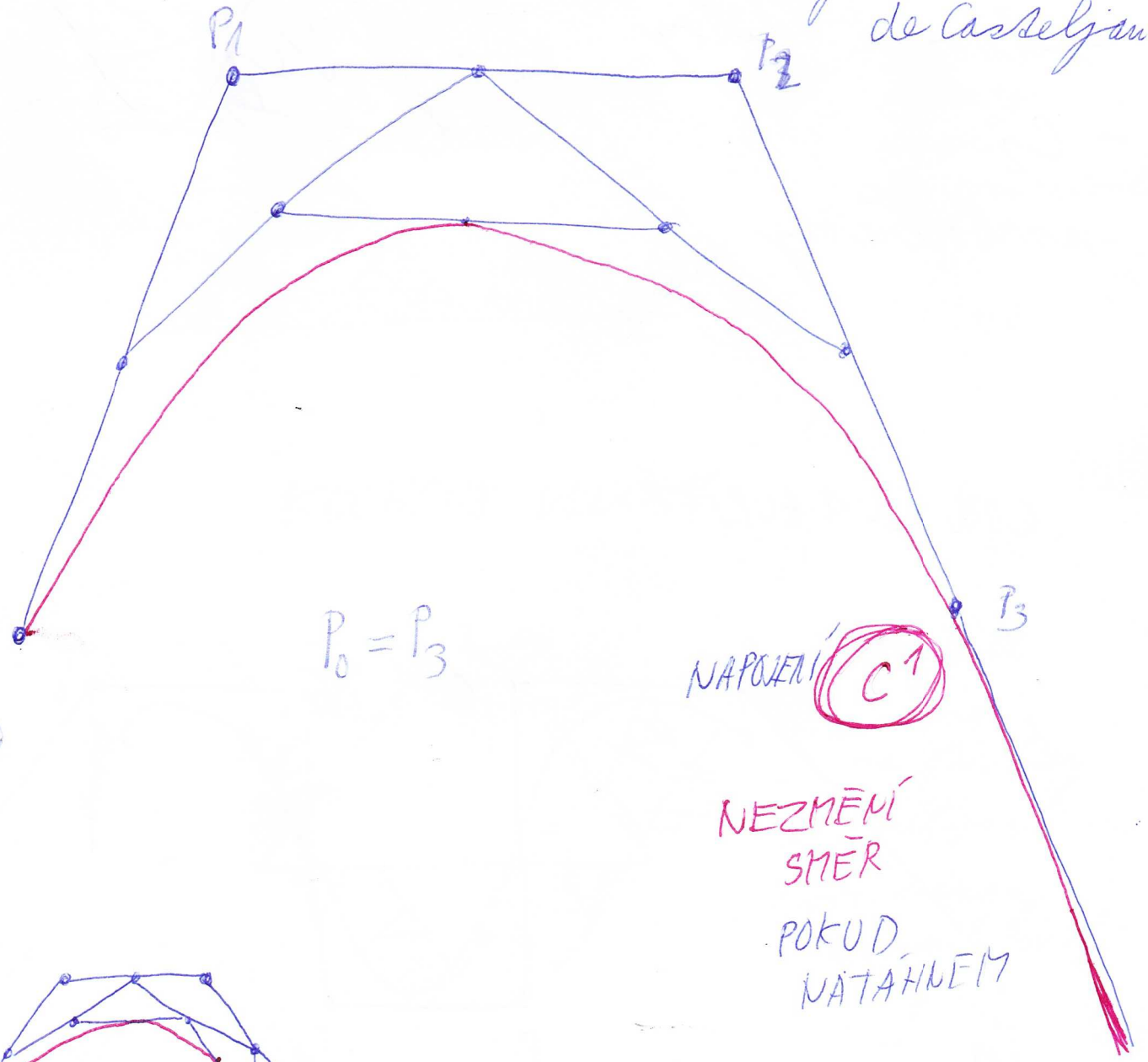


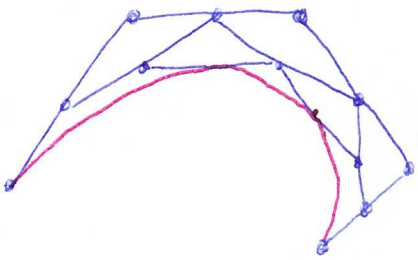


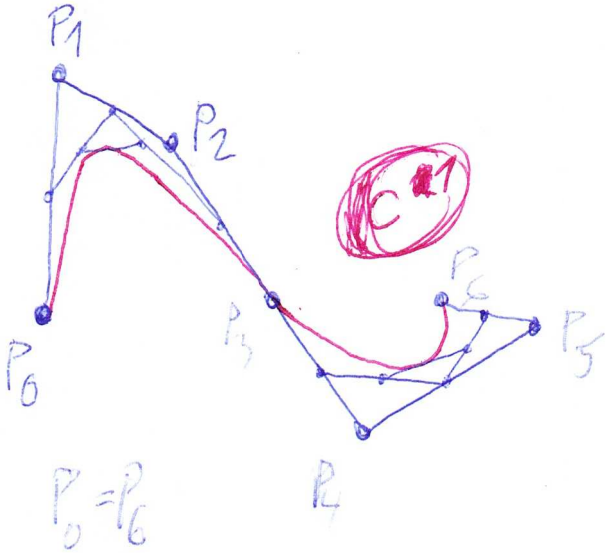
NULOVÉ VEKTORY

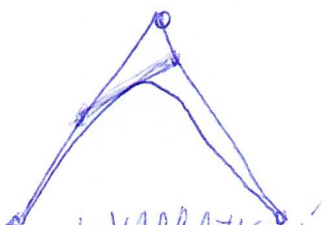
BEZIEROVA ~~KŘIVKA~~ KUBIKA

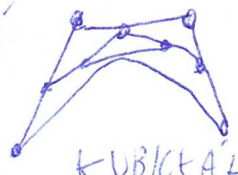
- algoritmus de Casteljau



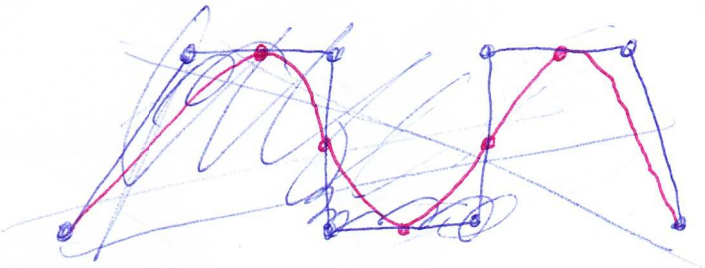






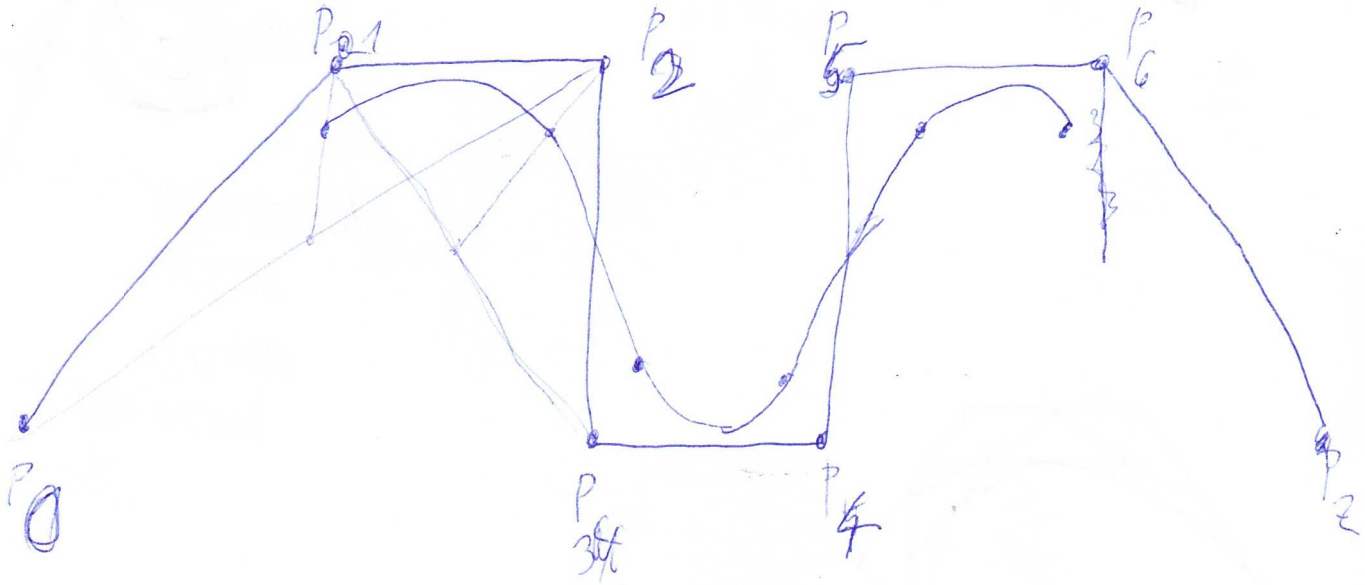


COONSOVY ~~KŘIVKY~~ KUBIKY

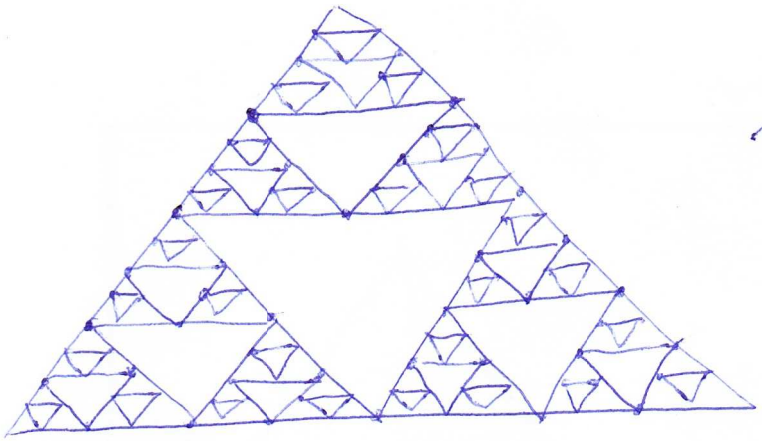


- vektorové editory
- C_2
- nurbs křivky

COONSOVA KUBIKA:

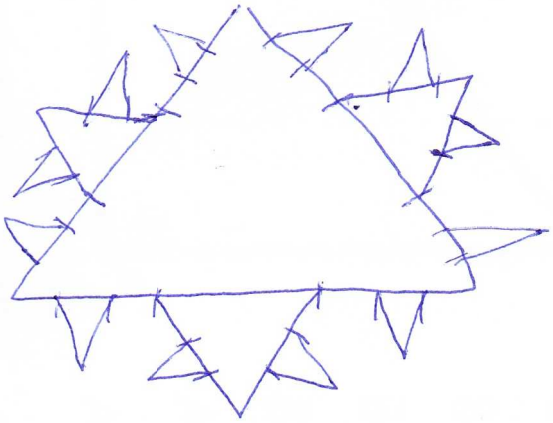


SHERPINSKÝ FRAKTÁL



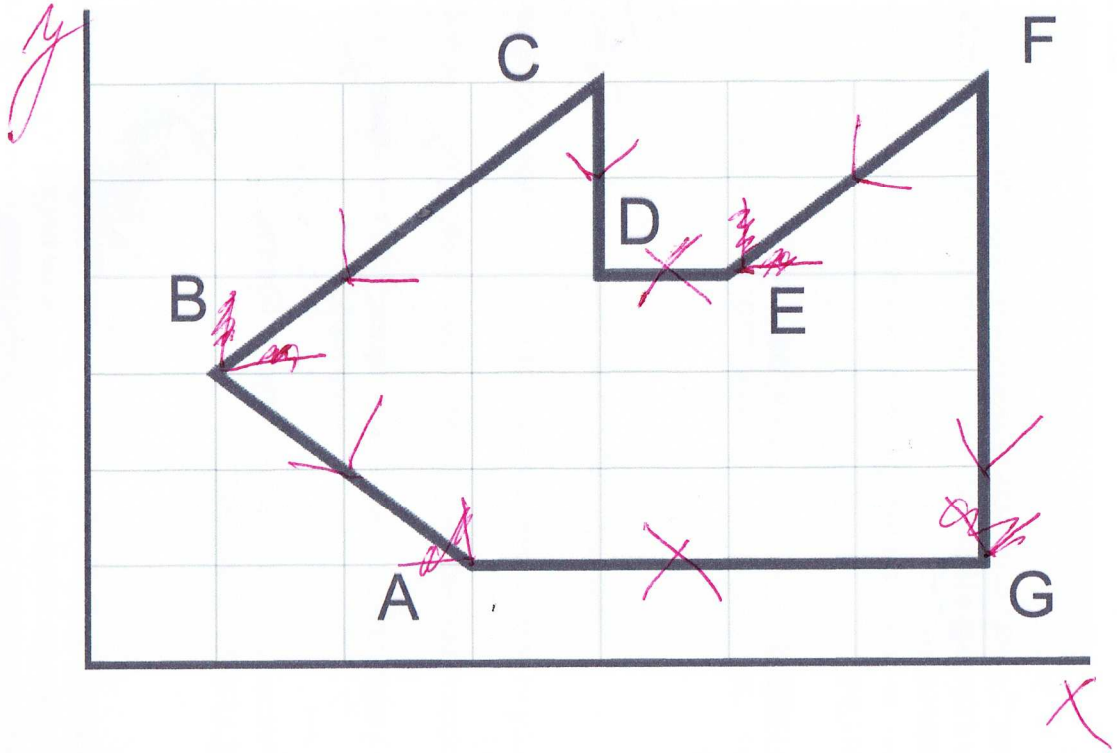
spojíš středy
hran
trojúhelníku

KOCHOVÝ VLOČKY



rozdělíš
hrany
na třetiny
a vytvoříš
rovnostranný
trojúhelník

Řádkové vyplňování



BC-CD, EF-FG

BC-CD, EF-FG

BC-CD, (DE), EF-FG

BC?AB-FG

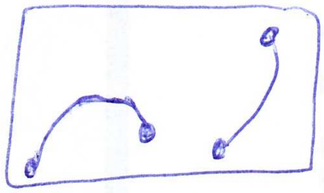
AB-FG

AB-(GA)-FG

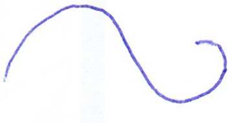
- Předzpracování hraničních úseček:
  - + vynechat vodorovné
  - + ostatní orientovat shora dolů a zkrátit o pixel u dolního vrcholu

SEMÍNKOVÉ - VYPLŇOVÁNÍ

VYBERU PIXEL A PAK VŽDY SOUSEDY OPAKUJEME

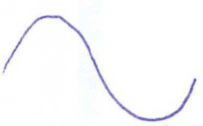


C^0 PRUDKÁ ZMĚNA SMĚRU



ZMĚNY POZVOLNÉ

C_1



NEMĚNÍ SE NIKDY RYCHLOST

C_2