**Modelovanie telies a reprezentácie**

Čo je to modelovanie?

* Reprezentacia a manipulacia objektov
  + Ziskat, upravit, transformovat, vyhladit, zobrazit, deformovat, morfovat, komprimovat, prenasat, analyzovat
* Ako:
  + Reprezentovat 2d a 3d objekty v pocitaci?
  + Ziskat pocitacove reprezentacie objektov?
  + Manipulovat s reprezentaciami predmetov?

Reprezentacia 2D objektov

* Reprezentacia 2D obrazu ako matica pixelov
* Ako definovat 2D tvary?
  + Trojuholnik: 3 body, nejaka informacia ako maju byt spojene atd
* Raster – bitmapy, raster, pixely explicitne
* Vector – tvary, vektory, krivky parametricke, implicitne

Reprezentacia 3D objektov

* Polygony (trojuholniky)
* Voxely (objemove kocky)
* Bodova reprezentacia
  + Hlbkove obrazy
  + Mracna bodov (point cloud)
* Povrchova reprezentacia
  + Polygonalne, delene, parametricke, implicitne
* Pevne latky a telesa (rigid bodies)
  + Voxels, BSP tree, CSG, sweep
* Makke latky a telesa
  + Tkaniny, deformovatelne objekty, kvapaliny
* Hierarchicke struktury
  + Graafy sceny, aplikacne zavisle

Prečo tolko roznych reprezentacii?

* Efektivnost pre rozne ulohy
  + Ziskavanie, analyza, manipulacia, vykreslovanie, animacia
* Datove struktury urcuju algoritmy

**Krivky**

Sposob zapisu

* Explicitne y = f(x)
* Implicitne F(x,y) = 0 ; kružnica: F(x,y) = (x-xs)^2+(y-ys)ˆ2-rˆ2 = O
* Parametricky x = x(t), y = y(t), z = z(t)

**Vlastnosti kriviek**

* Dotycnicovyy vector v bode
  + Derivacia kazdeho bodu = (x´(tˇ0) atd )
* Rovnica dotycnice
  + P(u) = Q(t0) + u
* Spajanie kriviek a spojitost
  + Uzol Q1(1) = Q2(0)
  + Q(t) je triedy C^n, ked ma spojite derivacie podla t do radu n

Pozadovane vlastnosti

* Invariancia k lin. Transformaciam a projekciam
  + Napr otocenie controlnych Bodov = otocenie krivky
* Vlastnost konvexnej obalky
  + Krivka (cast) lezi v konvexnom obale svojich kontrolnych bodov
* Lokalita zmien
  + Zmena polohy / vahy kontrolnych bodov -> zmena len casti krivky
* Krivka prechadza krajnymi bodmi riadiaceho polygonu (def. kontrolnymi bodmi)

Modelovanie kriviek

* Polynomicke krivky
  + Kubiky – stupen n=3
  + Qn(t) = ant^n + .. +a1t+a0
* Kontrolne body
  + Aproximacne krivky
  + Interpolacne krivky

Interpolacne krivky

* Hermitovska (Fergusonova) kubika
* Vstupuju tam body a vektory a nasobia sa nejakymi polynomalnymi funkciami
* Vsetky maju rovnaku maticku z tych polynomalnych funkcii, menia sa jedine body a vektory

Aproximacne krivky

* Bezierove krivky
* kde to Bi^n su bernsteinove polynomy n teho stupna

Bezierove kubiky

* Definovana bodmi P0,P1,P2,P3
* Nadvazovanie kubik
  + Riesi problem zmeny celej krivky
* C^1 spojitost, ked Pn = Q0 je stredom usecky Pn-1 a Q1

Coonsove kubiky

* Uniformy ne-racionalny B-spline
* Neprechadza krajnymi bodmi
* Q(0) = (P0+4P1+P2)/6
* Q(1) = P1+4P2+P3)/6
* Vector q derivovany (0) = P2-P0)/2
* -//-(1) = P3-P1/2

Spline krivky

* Spline krivka stupna n je po castiach polynomicka krivka, ktora je triedy Cˆn
* Prirodzeny spline – interpoluje riadiace body
  + Sklada z kriviek stupna 3 -> C^2 spojita v uzloch
* B-spline krivky
  + Aproximacne krivky, tj nie su prirodzene spline

Uniformny kubicky B-spline

* Vznikne naviazanim niekolkych coonsovych kubik
* N>= 4 kontrolnych bodov
* N – 3 segmentov
* Kontrolne body P0,P1,…,Pn
* Segment Q3,Q4,…,Qn
* Uniformny ti+1 -ti = k
* Lokalita zmien
  + Rozdiel v p4 a p´4 -> zmena uzlov t5,t6,t7 a 4-roch segmentov

B-spline

* B-spline krivka stupna p je nejak zadefinovana, kukni si prezentaciu
  + N+1 kontrolnych bodov
  + Ni,p(u) su B-spline bazove polynomy
  + Uzlovy vector u = {u0,u1,…,um}
    - Napr u={-2,-1,0,1,2,3,4,5} uniformna postupnost
  + Mozno uzavriet

NURBS

* Neuniformne racionalne B-spline krivky
* Motivacia
  + Kruznica (krivka stupna 2) je pomocou B-spline kriviek prilis komplikovana:
    - 4 B-spline krivky, 8 kontrolnych bodov
    - Stupna 2,3,5 a 10

Ako zobrazit krivky?

Hrubou silou

* Dosadenie parametra t do vzorca, napr pre bezierovu krivku do vzorca
* Nevyhody:
  + Rovnomerny krok t
  + Dlza usecky rovnaka -> nepresne
  + Dlha usecka: krivka je “hranata” (lebo menej krokov)
  + Kratka usecka: pomale vykreslovanie (vela krokov)
* Algoritmus de Casteljau – toto implementujeme na cvikach
  + Postavene na linearnych interpolaciach

**Plochy**

Parametricke plochy

* Bodova rovnica plochy Q(u,v) dvoch parametrov u,v patria do <0,1>
* X(u,v), y(u,v), z(u,v) polynomicke funkcie dvoch parametrov u,v
* Rohy plochy Q(0,1), Q(1,0), Q(0,1), Q(1,1)
* Strany Q(u,v) – hlavne krivky v smer u pre u=0, v = 1, podobne pre v
* Predstav si zakriveny papier
* Dotycnica v smere u
* Dotycnica v smere v

Pozadovane vlastnosti

* Iinvariancia k linearnymc transformaciam a projekciam
* Bla blab la idk

Modelovanie ploch

* Plocha Q(u,v)
* MB – bazova matica, obsahuje komplikovane koeficienty ai polynomov bazovych funkcii (nemeni sa)
* P – matica riadiace body a dalsie prvky urcujuce geometriu plochy
* To iste co krivky len ideme v dvoch smeroch

Spajanie zaplat

* C^0 spojitost
  + Maju spolocnu stranu, ktora je krivkou triedy C^0
* C^1 spojitost
  + Maju spolocnu stranu
  + Strana je C^1 spojita
  + Priecne parcialne derivacie ku strane su identicke

Aproximacne plochy

* Dvanast-vektorova plocha definovana 4 rohovymi bodmi
  + Dotycnice v oboch smeroch
  + Rovnica plochy
* Plochy spajajuce 2 krivky
  + Priamkova ploch
* Billinearna coonsova plocha
* Bi-kubicka plocha
* Vseobecna bi-kubicka plocha

**Bezierova plocha**

* N x m-teho stupna je urcena (n+1)x(m+1) riadiacimi bodmi Pi a vztahom nejakym kukni prez

B-spline plochy

* Lahko sa nadvazuju
* B-spline plochy n-teho stupna zarucuju C^n-1 spojitost
* Vlastnosti ma nejake no

**NURBSS plochy**

* Nejak jebnuto definovana prilis tazke na mna