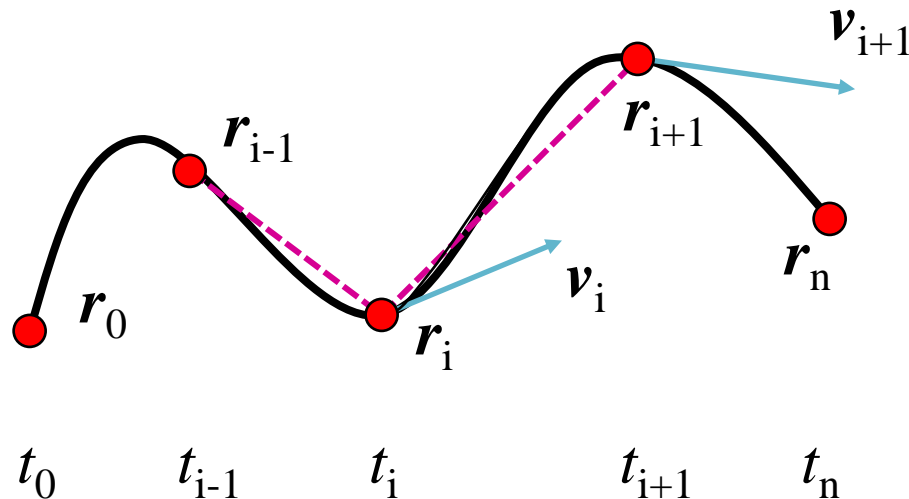


# OpenGL III

# Catmull-Rom spline

Minden két vezérlőpont közé egy görbe szegmens

Simaság: a sebesség is legyen közös két egymás utánira



$$\mathbf{v}_i = \frac{1}{2} \left[ \frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{t_{i+1} - t_i} + \frac{\mathbf{r}_i - \mathbf{r}_{i-1}}{t_i - t_{i-1}} \right]$$

Egy görbeszegmens: Hermite interpoláció

Legeslegelső és legutolsó sebesség explicit

# Catmull-Rom spline

- Egyenletes sebesség esetén

$$p(t) = [t^3 \quad t^2 \quad t \quad 1] \begin{bmatrix} -s & 2-s & s-2 & s \\ 2s & s-3 & 3-2s & -s \\ -s & 0 & s & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix}$$

- $s$  a görbeszegmensek találkozásánál a sebesség arány
  - $\frac{1}{2}$  általában jó választás

# Catmull-Rom spline

- Kontroll pontok
  - GL\_LINE\_STRIP\_ADJACENCY
- A görbe interpoláció a geometria shaderben

```
#version 330
#extension GL_EXT_gpu_shader4: enable
#extension GL_EXT_geometry_shader4: enable

layout(lines_adjacency) in;
layout(line_strip, max_vertices=200) out;

Void main()
{
    ...
}
```

# Catmull-Rom spline

## ■ Feladat

- A kontroll pontok az egér kattintással rakhatóak le
- Jelenítsük meg a kontrol pontokat összekötve
- Jelenítsük meg az interpolált görbét
  - Speciális és (esetleg) általános eset

# Részecske rendszer

- Részecskék mint pontok
  - Pozíció és sebesség
- A geometria shaderben a pozíció módosítása
  - A részecske kitörlődik, ha kimegy az ablakból
  - Transform feedbackkel iteráljunk!
  - 2D eset
    - Pozíció és sebesség egy float4
    - Egyetlen buffer elég a feedbackhez
  - 3D eset
    - Több bufferbe feedbackelünk párhuzamosan