Documentation

[Curve Interpolation]

Jiayi Pu

More information (video) at https://u.osu.edu/pu.127/3d-unity/

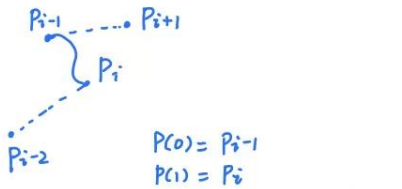
Cubic interpolations are used so that the cube moves through the points in a curved pattern. You have two choices of different patterns:
CatmullRomCurveInterpolation.cs and BezierCurves.cs.

Initially, there will randomly create points. The object is created by re-parameterizing with arc length (when distanceInc = time in the update function). Then, EaseInAndOut function makes the cube slowly start from point, and down slow to end point.

Also, the object rotates to face forward in the direction it is moving.

For Catmull Rom Curve Interpolation:

Catmull-Rom spline derivation



$P(0) = P_{i-1}$
$P(1) = P_i$
$P'(0) = \tau(P_i - P_{i-2})$
$P'(1) = \tau(P_{i+1} - P_{i-1})$

With cubic equation $C_3u^3 + C_2u^2 + C_1u + C_0$

$C_0 = P_{i-1}$

$C_0 + C_1 + C_2 + C_3 = P_i$

$C_1 = \tau(P_i - P_{i-2})$

$C_1 + 2C_2 + 3C_3 = \tau(P_{i+1} - P_{i-1})$

We can solve and get.

$C_3 = (-\tau)P_{i-2} + (2-\tau)P_{i-1} + (\tau-2)P_i + (\tau)P_{i+1}$

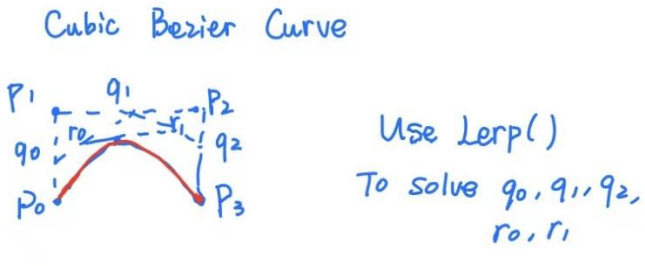
$C_2 = (2\tau)P_{i-2} + (\tau-3)P_{i-1} + (3-2\tau)P_i + (-\tau)P_{i+1}$

$C_1 = (-\tau)P_{i-2} + (\tau)P_i$

$C_0 = P_{i-1}$

Create 8 points. P0->P1->P2…->P8(P0). And always there are 4 points need to be considered, Pi-2, Pi-1, Pi and Pi+1. Therefore, as an example, if i=0, i-2 = -2 which need to set to 6; i-1 = -1 need to set to 7.

For Bezier Curve:



Create 9 points. Because P0 -> P3, P3->P6, P6->P9(P0) is a great loop.