

Math (EM4)

Definierte Funktionen

Quellcode

```
1. namespace Math
2. {
3.     extern const int RANDMAX;
4.     float sqrt(float f); //gibt die Quadratwurzel einer Zahl zurück
5.     int sqrti(int i); //gibt die Quadratwurzel einer Zahl zurück
6.     float sqr(float f); //??? Vermutlich ebenfalls Wurzel
7.     float VectorLen(float x, float y, float z); //Gibt die Länge eines Vektors zurück. Veraltet, besser: float
        Vector::GetLen();
8.     void NormalizeVector(float &x, float &y, float &z); //Normiert Vektor (Länge = 1). Veraltet, besser: Vector
        Vector::GetNormal();
9.     int rand(); //Gibt Zufallszahl von 0-1 zurück
10.    int abs(int i); //Gibt Betrag (Zahl ohne Vorzeichen) zurück
11.    float acos(float f); //arccos
12.    float asin(float f); //arcsin
13.    float atan(float f); //arctan
14.    float atan2(float f, float g); //arctan2
15.    float cos(float f); //cos
16.    float cosh(float f); //cosh
17.    float exp(float f); //Exponent zur Basis e: e^f
18.    float fabs(float f); //Gibt Betrag zurück
19.    float fmod(float f, float g); //???
20.    float log(float f); //natürlicher Logarithmus zur Basis e
21.    float log10(float f); //Logarithmus zur Basis 10
22.    float pow(float f, float g); //f^g
23.    float sin(float f); //sin
24.    float sinh(float f); //sinh
25.    float tan(float f); //tan
26.    float tanh(float f); //tanh
27.    float atof(const char *s); //Char zu Float
28.    int atoi(const char *s); //Char zu Integer
29.    float ceil(float f); //Gibt die kleinste Ganzzahl zurück, die nicht kleiner als f ist (f = 4.8 -> 5)
30.    float floor(float f); //Gibt die größte Ganzzahl zurück, die nicht größer als f ist (f = 4.8 -> 4)
31.    float dist2(float x0, float y0, float x1, float y1); //Entfernung zwischen zwei 2D-Vektoren.
32.    float dist(float x0, float y0, float x1, float y1); //wie dist2()
33.    float dist2(float x0, float y0, float z0, float x1, float y1, float z1); //Entfernung zwischen zwei 3D-Vektoren. Veraltet,
        besser: GetLen(vecA - vecB);
34.    float dist(float x0, float y0, float z0, float x1, float y1, float z1); //wie dist2()
35.    void RotateVector(float &vectorx_, float &vectory_, float &vectorz_, const float *matrix3x3_); //Vektor mit
        Rotationsmatrix drehen
36.    void CrossProduct(float x, float y, float z, float &cx, float &cy, float &cz); //Kreuzprodukt zweier Vektoren (vecA x
        vecB)
37.    void EulerToMatrix(float yaw, float pitch, float roll, float *matrix3x3_); //drei Rotationsvektoren (r/p/y) zu Rotations-
        Matrix
38.    void MultiplyMatrices(float *mat1_, const float *mat2_); //multipliziert 2 Matrixen
39. }
```

Alles anzeigen

Inhaltsverzeichnis

- [1 Definierte Funktionen](#)