zCViewShaped : zCViewInteractive

enum zTMatType

MAT\_SUMMARY,

MAT\_TRANSLATION,

MAT\_ROTATION,

MAT\_SCALE,

MAT\_MAX

Перечислитель, определяющий типы разделенных матриц объекта.

Это класс объектов, форма которых задается произвольным набором вершин. Вершины определяются в виртуальном диапазоне от 0 до 8192 относительно данного объекта, а текстурные координаты от 0 до 1. Например, чтобы полностью заполнить объект, потребуется прямоугольник с точками min(0;0) max(8192;8192).

Защищенные члены:

Свойства:

zTBBox2D BBox;

Занимаемый объектом объем.

Array<zTRndSimpleVertex> Modified;

Подготовленные для рендера вершины.

Публичные члены:

Свойства:

Array<zTRndSimpleVertex> Vertexes;

Исходный список вершин объекта. Вершины объекта указываются в виртуальных координатах относительно текущего объекта. То есть левая верхняя точка будет соответствовать {0; 0}, а нижняя правая – {8192, 8192}.

zMAT3 TrafoShape[MAT\_MAX];

Набор разделенных матриц фигуры на 4 части: общую матрицу, матрицу смещения, матрицу поворота и матрицу масштаба.

zMAT3 TrafoUV[MAT\_MAX];

Набор разделенных матриц UV на 4 части: общую матрицу, матрицу смещения, матрицу поворота и матрицу масштаба.

bool\_t UseParentTrafo;

Зарезервировано.

virtual bool\_t IsOnBounds( int x, int y );

Определяет, находится ли координаты в пределах объекта.

* x – пиксельная координата X
* y – пиксельная координата Y

void GetShapeBounds( zVEC2& min, zVEC2& max );

Возвращает границы объекта в виртуальных координатах.

* min – минимальные виртуальные координаты объекта
* max – максимальные виртуальные координаты объекта

void CreateFlatUV();

Вписывает UV по крайним точкам объекта.

void CreateBoxShape( const zVEC2& min, const zVEC2& max );

Создает прямоугольную форму объекта.

* min – левый верхний угол прямоугольника
* max – правый нижний угол прямоугольника

void CreateRoundShape( const zVEC2& center, const float& radius, const uint& edges );

Создает округлую форму объекта.

* center – центр окружности в виртуальных координатах
* radius – радиус окружности в виртуальных координатах
* edges – количество граней окружности

void MoveShape( const zVEC2& direction );

Сдвигает фигуру объекта в заданном направлении.

* direction – направление сдвига в виртуальных координатах

void RotateShape( const float& angle, const zVEC2& pivot = 4096.0f );

Поворачивает фигуру на заданный угол.

* angle – угол поворота в градусах
* pivot – опорная точка поворота в виртуальных координатах

void ScaleShape( const zVEC2& direction, const zVEC2& pivot = 4096.0f, const float& angle = 0.0f );

Масштабирует фигуру в заданном направлении.

* direction – направление масштабирования в виртуальных координатах
* pivot – опорная точка масштабирования
* angle – угол поворота направления direction в градусах

void MoveUV( const zVEC2& direction );

Сдвигает текстуру объекта в заданном направлении.

* direction – направление сдвига в виртуальных координатах.

void RotateUV( const float& angle, const zVEC2& pivot = 4096.0f );

Поворачивает текстуру на заданный угол.

* angle – угол поворота в градусах
* pivot – опорная точка поворота

void ScaleUV( const zVEC2& direction, const zVEC2& pivot = 4096.0f, const float& angle = 0.0f );

Масштабирует текстуру в заданном направлении.

* direction – направление масштабирования в виртуальных координатах
* pivot – опорная точка масштабирования
* angle – угол поворота направления direction в градусах

void MoveUVLocal( const zVEC2& direction );

Сдвигает текстуру в заданном направлении учитывая положение самой фигуры.

* direction – направление сдвига в виртуальных координатах.

void RotateUVLocal( const float& angle, const zVEC2& pivot = 4096.0f );

Поворачивает текстуру на заданный угол учитывая положение самой фигуры.

* angle – угол поворота в градусах
* pivot – опорная точка поворота

void ScaleUVLocal( const zVEC2& direction, const zVEC2& pivot = 4096.0f, const float& angle = 0.0f );

Масштабирует текстуру в заданном направлении учитывая положение самой фигуры.

* direction – направление масштабирования в виртуальных координатах
* pivot – опорная точка масштабирования
* angle – угол поворота направления direction в градусах

virtual void InsertVertex( const zVEC2& point );

Добавляет вершину к текущей фигуре данного объекта.

* point – положение точки в виртуальных координатах

virtual void InsertVertex( const zVEC2& point, const zVEC2& uv );

Добавляет вершину к текущей фигуре данного объекта и назначает UV.

* point – положение точки в виртуальных координатах
* uv – текстурная координата в диапазоне от 0 до 1

void ResetShapeTranslation( const zVEC2& trans = 0.0f );

Сбрасывает смещение фигуры и устанавливает новое.

void ResetShapeRotation( const float& angle = 0.0f, const zVEC2& pivot = 4096.0f );

Сбрасывает поворот фигуры и устанавливает новый.

void ResetShapeScaling( const zVEC2& direction = 0.0f, const zVEC2& pivot = 4096.0f, const float& angle = 0.0f );

Сбрасывает масштабирование фигуры и устанавливает новое.

void ResetUVTranslation( const zVEC2& direction = 0.0f );

Сбрасывает смещение текстуры и устанавливает новое.

void ResetUVRotation( const float& angle = 0.0f, const zVEC2& pivot = 4096.0f );

Сбрасывает поворот текстуры и устанавливает новый.

void ResetUVScaling( const zVEC2& direction = 0.0f, const zVEC2& pivot = 0.0f, const float& angle = 0.0f );

Сбрасывает масштабирование текстуры и устанавливает новое.