

Complex Logarithmic View

Generated by Doxygen 1.8.1.1

Thu Jun 14 2012 22:56:45

Contents

1	Namespace Index	1
1.1	Packages	1
2	Class Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	Namespace Documentation	7
4.1	Package Clv	7
4.2	Package Clv.Viewer	7
4.3	Package Clv.Viewer.Math	8
4.3.1	Enumeration Type Documentation	8
4.3.1.1	SplitCase	8
4.3.1.2	SplitCaseFocus	8
4.3.1.3	SplitState	8
5	Class Documentation	9
5.1	Clv.Viewer.App Class Reference	9
5.1.1	Detailed Description	10
5.1.2	Constructor & Destructor Documentation	10
5.1.2.1	App	10
5.1.3	Member Function Documentation	10
5.1.3.1	AddTexture	10
5.1.3.2	Draw	10
5.1.3.3	Initialize	10
5.1.3.4	LoadContent	10
5.1.3.5	MoveFocusPoint	10
5.1.3.6	NextTexture	11
5.1.3.7	PrevTexture	11
5.1.3.8	ResetProjection	11
5.1.3.9	Update	11

5.2	Clv.Viewer.Math.Bounds2 Class Reference	11
5.2.1	Detailed Description	11
5.3	Clv.Viewer.ComplexLogView Class Reference	11
5.4	Clv.Viewer.ConcreteRenderable Class Reference	12
5.4.1	Detailed Description	12
5.4.2	Member Function Documentation	12
5.4.2.1	Dispose	12
5.4.3	Property Documentation	13
5.4.3.1	IndexBuffer	13
5.4.3.2	Transform	13
5.4.3.3	VertexBuffer	13
5.5	Clv.Viewer.Controls Class Reference	13
5.5.1	Detailed Description	13
5.5.2	Member Function Documentation	13
5.5.2.1	UpdateKeyBoardControls	13
5.5.2.2	UpdateMouseControls	14
5.6	Clv.Viewer.CutOffFisheyeEffect Class Reference	14
5.7	Clv.Viewer.DefaultEffect Class Reference	14
5.8	Clv.Viewer.Math.Edge2 Class Reference	15
5.8.1	Detailed Description	15
5.9	Clv.Viewer.Grid Class Reference	15
5.9.1	Detailed Description	16
5.9.2	Constructor & Destructor Documentation	16
5.9.2.1	Grid	16
5.9.2.2	Grid	16
5.9.3	Member Function Documentation	16
5.9.3.1	DisposeBuffers	16
5.9.3.2	GetConcreteRenderable	16
5.9.3.3	Split	17
5.9.4	Property Documentation	17
5.9.4.1	Bounds	17
5.9.4.2	Indices	17
5.9.4.3	LastSplitPlane	17
5.9.4.4	SplitGrid	17
5.9.4.5	Transform	17
5.9.4.6	Triangles	17
5.9.4.7	VertexData	17
5.9.4.8	Vertices	17
5.10	Clv.Viewer.IRenderable Interface Reference	18
5.10.1	Detailed Description	18

5.10.2	Member Function Documentation	18
5.10.2.1	GetConcreteRenderable	18
5.10.3	Property Documentation	18
5.10.3.1	Indices	18
5.10.3.2	Transform	19
5.10.3.3	VertexData	19
5.11	Clv.Viewer.Math.Line2 Class Reference	19
5.11.1	Detailed Description	19
5.12	Clv.Viewer.Renderable Class Reference	19
5.12.1	Member Function Documentation	20
5.12.1.1	GetConcreteRenderable	20
5.12.2	Property Documentation	20
5.12.2.1	Indices	20
5.12.2.2	Transform	20
5.12.2.3	VertexData	20
5.13	Clv.Viewer.Math.SplitInfo Class Reference	20
5.13.1	Detailed Description	21
5.13.2	Member Function Documentation	21
5.13.2.1	GetNewPoints	21
5.13.2.2	GetSplitCase	21
5.13.3	Member Data Documentation	21
5.13.3.1	NewPoints	21
5.13.3.2	NewPointsFocus	21
5.13.3.3	State	21
5.13.3.4	StateFocus	22
5.14	Clv.Viewer.Math.SplitPlane2 Class Reference	22
5.14.1	Detailed Description	22
5.14.2	Member Function Documentation	22
5.14.2.1	Height	22
5.15	Clv.Viewer.Math.Triangle2 Class Reference	23
5.15.1	Detailed Description	23
5.15.2	Member Function Documentation	23
5.15.2.1	ComputeBaryCentric	23
5.15.2.2	ContainsPoint	23
5.15.2.3	GetEdges	24
5.16	Clv.Viewer.Math.Vertex2 Class Reference	24
5.16.1	Detailed Description	24
5.16.2	Member Function Documentation	24
5.16.2.1	Interpolate	24
5.16.2.2	ToVertexPositionTexture	25

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Clv	7
Clv.Viewer	7
Clv.Viewer.Math	8

Chapter 2

Class Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

- Clv.Viewer.App 9
- Clv.Viewer.Math.Bounds2 11
- Clv.Viewer.ComplexLogView 11
- Clv.Viewer.ConcreteRenderable 12
- Clv.Viewer.Controls 13
- Clv.Viewer.CutOffFisheyeEffect 14
- Clv.Viewer.DefaultEffect 14
- Clv.Viewer.Math.Edge2 15
- Clv.Viewer.IRenderable 18
 - Clv.Viewer.Grid 15
 - Clv.Viewer.Renderable 19
- Clv.Viewer.Math.Line2 19
- Clv.Viewer.Math.SplitInfo 20
- Clv.Viewer.Math.SplitPlane2 22
- Clv.Viewer.Math.Triangle2 23
- Clv.Viewer.Math.Vertex2 24

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Clv.Viewer.App	class for main application logic	9
Clv.Viewer.Math.Bounds2	2dim bounding rectangle defined between 2 points (min,max	11
Clv.Viewer.ComplexLogView	11
Clv.Viewer.ConcreteRenderable	ConcreteRenderable are the actual filled buffer declarations which are then uploaded to the Graphics card. the extension "ToConcreteRenderable" translates a renderable to a concrete renderable.	12
Clv.Viewer.Controls	Includes user interaction with mouse and keyboard Mousecontrols handle the selection of a new focuspoint on click keyboard controls handle changes in function root, culling mode, wireframe to solid mode and vice versa depth and texture changes and goes to fullscreen mode	13
Clv.Viewer.CutOffFisheyeEffect	14
Clv.Viewer.DefaultEffect	14
Clv.Viewer.Math.Edge2	2dim Line defined between 2 vertices	15
Clv.Viewer.Grid	Tesselated squared regular 2d grid.	15
Clv.Viewer.IRenderable	Basic interface a class must implement to be sent to the graphics card and be rendered	18
Clv.Viewer.Math.Line2	2dim Line defined between 2 points	19
Clv.Viewer.Renderable	19
Clv.Viewer.Math.SplitInfo	Contains splitting information from the triangle split stage	20
Clv.Viewer.Math.SplitPlane2	SplitPlane defined by focus point and normal vector n. Concept similiar to normal plane but is delimited by FocusPoint. Triangle hit by splitplane at focuspoint needs special split processing.	22
Clv.Viewer.Math.Triangle2	Class to handle a 2D triangle	23
Clv.Viewer.Math.Vertex2	Class to handle a 2-Dimensional vertex with an index and texture coordinates	24

Chapter 4

Namespace Documentation

4.1 Package Clv

Packages

- package [Viewer](#)

4.2 Package Clv.Viewer

Packages

- package [Math](#)

Classes

- class [App](#)
class for main application logic
- class [Controls](#)
Includes user interaction with mouse and keyboard Mousecontrols handle the selection of a new focuspoint on click keyboard controls handle changes in function root, culling mode, wireframe to solid mode and vice versa depth and texture changes and goes to fullscreen mode
- class [Grid](#)
Tesselated squared regular 2d grid.
- class [Renderable](#)
- interface [IRenderable](#)
Basic interface a class must implement to be sent to the graphics card and be rendered
- class [ConcreteRenderable](#)
ConcreteRenderable are the actual filled buffer declarations which are then uploaded to the Graphics card. the extension "ToConcreteRenderable" translates a renderable to a concrete renderable.
- class **RenderableExtensions**
- class [ComplexLogView](#)
- class [DefaultEffect](#)
- class [CutOffFisheyeEffect](#)
- class **Utils**

4.3 Package Clv.Viewer.Math

Classes

- class [SplitInfo](#)
Contains splitting information from the triangle split stage
- class **Constants**
Determines the width of the split in order to avoid numerical issues
- class **IntersectionUtils**
- class [Vertex2](#)
Class to handle a 2-Dimensional vertex with an index and texture coordinates
- class [Triangle2](#)
Class to handle a 2D triangle
- class [Edge2](#)
2dim Line defined between 2 vertices
- class [Line2](#)
2dim Line defined between 2 points
- class [Bounds2](#)
2dim bounding rectangle defined between 2 points (min,max
- class [SplitPlane2](#)
SplitPlane defined by focus point and normal vector n. Concept similar to normal plane but is delimited by FocusPoint. Triangle hit by splitplane at focuspoint needs special split processing.

Enumerations

- enum [SplitState](#) { **NoSplit**, **Split**, **FocusSplit** }
Determines if a Triangle was split at the focus point(FocusSplit), if it was cleanly split along the plane (Split), or if it wasn't split at all (NoSplit)
- enum [SplitCase](#) { **XOX**, **XXO**, **OXX**, **Invalid** }
Determines which edges of a triangle were split in case of a clean split.
- enum [SplitCaseFocus](#) { **XOO**, **O XO**, **OOX**, **Invalid** }
Determines which edge of a triangle were split in case of a focus split.

4.3.1 Enumeration Type Documentation

4.3.1.1 enum Clv.Viewer.Math.SplitCase

Determines which edges of a triangle were split in case of a clean split.

4.3.1.2 enum Clv.Viewer.Math.SplitCaseFocus

Determines which edge of a triangle were split in case of a focus split.

4.3.1.3 enum Clv.Viewer.Math.SplitState

Determines if a Triangle was split at the focus point(FocusSplit), if it was cleanly split along the plane (Split), or if it wasn't split at all (NoSplit)

Chapter 5

Class Documentation

5.1 Clv.Viewer.App Class Reference

class for main application logic

Public Member Functions

- [App](#) ()
create main window and start application
- void [AddTexture](#) (Texture2D newTexture)
Add the given texture to the complex logarithmic view effect
- void [NextTexture](#) ()
change texture to next item in texture list
- void [PrevTexture](#) ()
change texture to previous item in texture list
- void [MoveFocusPoint](#) (float x, float y)
Move focus point with given parameters if in range

Protected Member Functions

- void [OnClientSizeChanged](#) (object sender, EventArgs e)
- void [ResetProjection](#) ()
set projection to default
- override void [Initialize](#) ()
create grid and initialize main controls
- override void [LoadContent](#) ()
xna routine to get content for the application Load standard textures, shaders reset the current projection, set initial texture and texture filter
- override void [UnloadContent](#) ()
- override void [Draw](#) (GameTime gameTime)
xna draw routine - pass information over to shader give information from renderable to graphicsdevice
- override void [Update](#) (GameTime gameTime)
Update scene logic every frame create new split plane if focus point has moved add grid which has to be rendered if focus point was changed fly to new focus point in FlyTime - interpolate new focus point between finished and current time afterwards save elapsed time update keyboard and mouse controls

Properties

- bool **IsFullScreen** [get, set]
- int **NthRoot** [get, set]
- Matrix **CameraTransform** [get, set]
- float **Depth** [get, set]

5.1.1 Detailed Description

class for main application logic

5.1.2 Constructor & Destructor Documentation

5.1.2.1 Clv.Viewer.App.App ()

create main window and start application

5.1.3 Member Function Documentation

5.1.3.1 void Clv.Viewer.App.AddTexture (Texture2D *newTexture*)

Add the given texture to the complex logarithmic view effect

Parameters

<i>newTexture</i>	created texture
-------------------	-----------------

5.1.3.2 override void Clv.Viewer.App.Draw (gameTime *gameTime*) [protected]

xna draw routine - pass information over to shader give information from renderable to graphicsdevice

Parameters

<i>gameTime</i>	gameTime with information of total game time and elapsed game time
-----------------	--

5.1.3.3 override void Clv.Viewer.App.Initialize () [protected]

create grid and initialize main controls

5.1.3.4 override void Clv.Viewer.App.LoadContent () [protected]

xna routine to get content for the application Load standard textures, shaders reset the current projection, set initial texture and texture filter

5.1.3.5 void Clv.Viewer.App.MoveFocusPoint (float *x*, float *y*)

Move focus point with given parameters if in range

Parameters

<i>x</i>	movement in x direction
<i>y</i>	movement in y direction

5.1.3.6 void Clv.Viewer.App.NextTexture ()

change texture to next item in texture list

5.1.3.7 void Clv.Viewer.App.PrevTexture ()

change texture to previous item in texture list

5.1.3.8 void Clv.Viewer.App.ResetProjection () [protected]

set projection to default

5.1.3.9 override void Clv.Viewer.App.Update (GameTime gameTime) [protected]

Update scene logic every frame create new split plane if focus point has moved add grid which has to be rendered if focus point was changed fly to new focus point in FlyTime - interpolate new focus point between finished and current time afterwards save elapsed time update keyboard and mouse controls

Parameters

<i>gameTime</i>	gameTime with information of total game time and elapsed game time
-----------------	--

The documentation for this class was generated from the following file:

- Clv/Clv.Viewer/Clv.Viewer/App.cs

5.2 Clv.Viewer.Math.Bounds2 Class Reference

2dim bounding rectangle defined between 2 points (min,max)

Public Member Functions

- **Bounds2** (Vector2 min, Vector2 max)
- void **ExtendBy** (Vector2 v)

Public Attributes

- Vector2 **Min**
- Vector2 **Max**

5.2.1 Detailed Description

2dim bounding rectangle defined between 2 points (min,max)

The documentation for this class was generated from the following file:

- Clv/Clv.Viewer/Clv.Viewer/Math/Math.cs

5.3 Clv.Viewer.ComplexLogView Class Reference

Public Member Functions

- **ComplexLogView** (Effect effect)

Properties

- Matrix **World** [get, set]
- Matrix **View** [get, set]
- Matrix **Projection** [get, set]
- Texture2D **Texture** [get, set]
- Vector4 **FocusPoint** [get, set]
- int **NthRoot** [get, set]
- float **YOffset** [get, set]

The documentation for this class was generated from the following file:

- Clv/Clv.Viewer/Clv.Viewer/Shaders/ComplexLogView.cs

5.4 Clv.Viewer.ConcreteRenderable Class Reference

[ConcreteRenderable](#) are the actual filled buffer declarations which are then uploaded to the Graphics card. the extension "ToConcreteRenderable" translates a renderable to a concrete renderable.

Public Member Functions

- void [Dispose](#) ()
Disposes vertex and index gpu resources

Properties

- VertexBuffer [VertexBuffer](#) [get, set]
Contains the VertexData
- IndexBuffer [IndexBuffer](#) [get, set]
Contains the IndexData
- Matrix [Transform](#) [get, set]
Transformation which is applied before rendering

5.4.1 Detailed Description

[ConcreteRenderable](#) are the actual filled buffer declarations which are then uploaded to the Graphics card. the extension "ToConcreteRenderable" translates a renderable to a concrete renderable.

5.4.2 Member Function Documentation

5.4.2.1 void Clv.Viewer.ConcreteRenderable.Dispose ()

Disposes vertex and index gpu resources

5.4.3 Property Documentation

5.4.3.1 IndexBuffer `Clv.Viewer.ConcreteRenderable.IndexBuffer` `[get]`, `[set]`

Contains the IndexData

5.4.3.2 Matrix `Clv.Viewer.ConcreteRenderable.Transform` `[get]`, `[set]`

Transformation which is applied before rendering

5.4.3.3 VertexBuffer `Clv.Viewer.ConcreteRenderable.VertexBuffer` `[get]`, `[set]`

Contains the VertexData

The documentation for this class was generated from the following file:

- `Clv/Clv.Viewer/Clv.Viewer/IRenderable.cs`

5.5 Clv.Viewer.Controls Class Reference

Includes user interaction with mouse and keyboard Mousecontrols handle the selection of a new focuspoint on click keyboard controls handle changes in function root, culling mode, wireframe to solid mode and vice versa depth and texture changes and goes to fullscreen mode

Public Member Functions

- **Controls** (GraphicsDeviceManager graphics, double depth)
- void [UpdateMouseControls](#) ([ComplexLogView](#) effect, [App](#) game)

On Click a new focus point is selected, the calculation is based on a ray shot to the main plane this point is used to calculate the undistorted point on the plane therefore an inverse shader calculation is used to calculate the distance r from old focus point and y coordinate of the new focus point x is calculated simply with r and y

- void [UpdateKeyBoardControls](#) ([App](#) game)

Keyboard controls in Detail: [Escape]: exit program [F11]: toggle Fullscreen [+]: increase nth root up to a maximum of 8 [-]: decrease nth root down to a minimum of 1 [f]: toggle fill mode between wireframe and solid [c]: toggle cullmode (counterclockwise and clockwise) [PageUp]: decrease depth value (zoom out of view) [PageDown]: increase depth value (zoom into view) [I]: load new texture out of file (gif, png or jpg) [LeftArrow]: use previous loaded texture [RightArrow]: use next loaded texture

5.5.1 Detailed Description

Includes user interaction with mouse and keyboard Mousecontrols handle the selection of a new focuspoint on click keyboard controls handle changes in function root, culling mode, wireframe to solid mode and vice versa depth and texture changes and goes to fullscreen mode

5.5.2 Member Function Documentation

5.5.2.1 void `Clv.Viewer.Controls.UpdateKeyBoardControls` ([App](#) game)

Keyboard controls in Detail: [Escape]: exit program [F11]: toggle Fullscreen [+]: increase nth root up to a maximum of 8 [-]: decrease nth root down to a minimum of 1 [f]: toggle fill mode between wireframe and solid [c]: toggle cullmode (counterclockwise and clockwise) [PageUp]: decrease depth value (zoom out of view) [PageDown]: increase depth value (zoom into view) [I]: load new texture out of file (gif, png or jpg) [LeftArrow]: use previous loaded texture [RightArrow]: use next loaded texture

Parameters

<i>game</i>	game gives values for several control options and functions to change them
-------------	--

5.5.2.2 void Clv.Viewer.Controls.UpdateMouseControls (ComplexLogView effect, App game)

On Click a new focus point is selected, the calculation is based on a ray shot to the main plane this point is used to calculate the undistorted point on the plane therefore an inverse shader calculation is used to calculate the distance r from old focus point and y coordinate of the new focus point x is calculated simply with r and y

Parameters

<i>effect</i>	effect gives paramaeters for unmapping
<i>game</i>	game gives needed information for back calculation of distortion and the new focus point is set there

The documentation for this class was generated from the following file:

- Clv/Clv.Viewer/Clv.Viewer/Controls.cs

5.6 Clv.Viewer.CutOffFisheyeEffect Class Reference

Public Member Functions

- **CutOffFisheyeEffect** (Effect effect)

Properties

- Matrix **World** [get, set]
- Matrix **View** [get, set]
- Matrix **Projection** [get, set]
- Texture2D **Texture** [get, set]
- Vector4 **FocusPoint** [get, set]

The documentation for this class was generated from the following file:

- Clv/Clv.Viewer/Clv.Viewer/Shaders/FisheyeEffect.cs

5.7 Clv.Viewer.DefaultEffect Class Reference

Public Member Functions

- **DefaultEffect** (Effect effect)

Properties

- Matrix **World** [get, set]
- Matrix **View** [get, set]
- Matrix **Projection** [get, set]
- Texture2D **Texture** [get, set]

The documentation for this class was generated from the following file:

- Clv/Clv.Viewer/Clv.Viewer/Shaders/DefaultEffect.cs

5.8 Clv.Viewer.Math.Edge2 Class Reference

2dim Line defined between 2 vertices

Public Member Functions

- **Edge2** ([Vertex2](#) a, [Vertex2](#) b)

Public Attributes

- [Vertex2](#) A
- [Vertex2](#) B

5.8.1 Detailed Description

2dim Line defined between 2 vertices

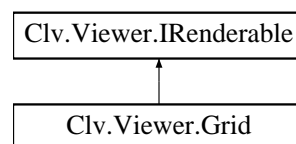
The documentation for this class was generated from the following file:

- Clv/Clv.Viewer/Clv.Viewer/Math/Math.cs

5.9 Clv.Viewer.Grid Class Reference

Tesselated squared regular 2d grid.

Inheritance diagram for Clv.Viewer.Grid:



Public Member Functions

- **Grid** (float size, short tessellation)
Creates a squared regular grid of certain world size and with a specific tessellation count;
- **Grid** (float size, short tessellation, [Vector2](#) offset)
Creates a squared regular grid of certain world size and with a specific tessellation count;
- void **Split** ([SplitPlane2](#) plane)
Computes a split version of the present grid by splitting it against the given SplitPlane.
- void **DisposeBuffers** ()
Disposes ressources allocated on the graphics card.
- **ConcreteRenderable** **GetConcreteRenderable** ([GraphicsDevice](#) device)
Either returns the cached concreteRenderable or creates a new one from SplitGrid.

Properties

- [Vertex2\[\] Vertices](#) [get, set]
Contains all Vertices
- [Triangle2\[\] Triangles](#) [get, set]
Contains all Triangles
- [SplitPlane2 LastSplitPlane](#) [get, set]
The last plane the Grid was split against
- [IRenderable SplitGrid](#) [get, set]
Holds the split version of the present grid
- [Bounds2 Bounds](#) [get, set]
Specifies the min and max extends of the grid
- Matrix [Transform](#) [get, set]
Transformation which is applied before rendering
- [VertexPositionTexture\[\] VertexData](#) [get, set]
Contains VertexData to be uploaded as a vertexbuffer
- [short\[\] Indices](#) [get, set]
Contains index information to be uploaded as an indexbuffer, indexing into the vertexbuffer

5.9.1 Detailed Description

Tesselated squared regular 2d grid.

5.9.2 Constructor & Destructor Documentation

5.9.2.1 Clv.Viewer.Grid.Grid (float size, short tessellation)

Creates a squared regular grid of certain world size and with a specific tessellation count;

5.9.2.2 Clv.Viewer.Grid.Grid (float size, short tessellation, Vector2 offset)

Creates a squared regular grid of certain world size and with a specific tessellation count;

5.9.3 Member Function Documentation

5.9.3.1 void Clv.Viewer.Grid.DisposeBuffers ()

Disposes ressources allocated on the graphics card.

5.9.3.2 ConcreteRenderable Clv.Viewer.Grid.GetConcreteRenderable (GraphicsDevice device)

Either returns the cached concreteRenderable or creates a new one from SplitGrid.

Parameters

<i>device</i>	GraphicsDevice for buffer conversion
---------------	--------------------------------------

Returns

[ConcreteRenderable](#)

Implements [Clv.Viewer.IRenderable](#).

5.9.3.3 void Clv.Viewer.Grid.Split (SplitPlane2 plane)

Computes a split version of the present grid by splitting it against the given SplitPlane.

Parameters

<i>plane</i>	SplitPlane to split against
--------------	-----------------------------

5.9.4 Property Documentation

5.9.4.1 Bounds2 Clv.Viewer.Grid.Bounds [get], [set]

Specifies the min and max extends of the grid

5.9.4.2 short [] Clv.Viewer.Grid.Indices [get], [set]

Contains index information to be uploaded as an indexbuffer, indexing into the vertexbuffer

Implements [Clv.Viewer.IRenderable](#).

5.9.4.3 SplitPlane2 Clv.Viewer.Grid.LastSplitPlane [get], [set]

The last plane the [Grid](#) was split against

5.9.4.4 IRenderable Clv.Viewer.Grid.SplitGrid [get], [set]

Holds the split version of the present grid

5.9.4.5 Matrix Clv.Viewer.Grid.Transform [get], [set]

Transformation which is applied before rendering

Implements [Clv.Viewer.IRenderable](#).

5.9.4.6 Triangle2 [] Clv.Viewer.Grid.Triangles [get], [set]

Contains all Triangles

5.9.4.7 VertexPositionTexture [] Clv.Viewer.Grid.VertexData [get], [set]

Contains VertexData to be uploaded as a vertexbuffer

Implements [Clv.Viewer.IRenderable](#).

5.9.4.8 Vertex2 [] Clv.Viewer.Grid.Vertices [get], [set]

Contains all Vertices

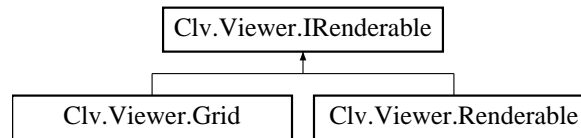
The documentation for this class was generated from the following file:

- Clv/Clv.Viewer/Clv.Viewer/Grid.cs

5.10 Clv.Viewer.IRenderable Interface Reference

Basic interface a class must implement to be sent to the graphics card and be rendered

Inheritance diagram for Clv.Viewer.IRenderable:



Public Member Functions

- [ConcreteRenderable GetConcreteRenderable](#) (GraphicsDevice device)
Transforms the given renderable into a concrete renderable containing buffer data.

Properties

- Matrix [Transform](#) [get, set]
Transformation which is applied before rendering
- VertexPositionTexture[] [VertexData](#) [get, set]
Contains VertexData to be uploaded as a vertexbuffer
- short[] [Indices](#) [get, set]
Contains index information to be uploaded as an indexbuffer, indexing into the vertexbuffer

5.10.1 Detailed Description

Basic interface a class must implement to be sent to the graphics card and be rendered

5.10.2 Member Function Documentation

5.10.2.1 ConcreteRenderable Clv.Viewer.IRenderable.GetConcreteRenderable (GraphicsDevice device)

Transforms the given renderable into a concrete renderable containing buffer data.

Parameters

<i>device</i>	GraphicsDevice for buffer conversion
---------------	--------------------------------------

Returns

[ConcreteRenderable](#)

Implemented in [Clv.Viewer.Grid](#), and [Clv.Viewer.Renderable](#).

5.10.3 Property Documentation

5.10.3.1 short [] Clv.Viewer.IRenderable.Indices [get], [set]

Contains index information to be uploaded as an indexbuffer, indexing into the vertexbuffer

Implemented in [Clv.Viewer.Grid](#), and [Clv.Viewer.Renderable](#).

5.10.3.2 Matrix Clv.Viewer.IRenderable.Transform [get], [set]

Transformation which is applied before rendering

Implemented in [Clv.Viewer.Grid](#), and [Clv.Viewer.Renderable](#).

5.10.3.3 VertexPositionTexture [] Clv.Viewer.IRenderable.VertexData [get], [set]

Contains VertexData to be uploaded as a vertexbuffer

Implemented in [Clv.Viewer.Grid](#), and [Clv.Viewer.Renderable](#).

The documentation for this interface was generated from the following file:

- Clv/Clv.Viewer/Clv.Viewer/IRenderable.cs

5.11 Clv.Viewer.Math.Line2 Class Reference

2dim Line defined between 2 points

Public Member Functions

- **Line2** (Vector2 a, Vector2 b)

Public Attributes

- Vector2 **A**
- Vector2 **B**

5.11.1 Detailed Description

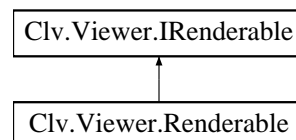
2dim Line defined between 2 points

The documentation for this class was generated from the following file:

- Clv/Clv.Viewer/Clv.Viewer/Math/Math.cs

5.12 Clv.Viewer.Renderable Class Reference

Inheritance diagram for Clv.Viewer.Renderable:



Public Member Functions

- [ConcreteRenderable GetConcreteRenderable](#) (GraphicsDevice device)
Transforms the given renderable into a concrete renderable containing buffer data.

Properties

- Matrix [Transform](#) [get, set]
Transformation which is applied before rendering
- VertexPositionTexture[] [VertexData](#) [get, set]
Contains VertexData to be uploaded as a vertexbuffer
- short[] [Indices](#) [get, set]
Contains index information to be uploaded as an indexbuffer, indexing into the vertexbuffer

5.12.1 Member Function Documentation

5.12.1.1 ConcreteRenderable Clv.Viewer.Renderable.GetConcreteRenderable (GraphicsDevice device)

Transforms the given renderable into a concrete renderable containing buffer data.

Parameters

<i>device</i>	GraphicsDevice for buffer conversion
---------------	--------------------------------------

Returns

[ConcreteRenderable](#)

Implements [Clv.Viewer.IRenderable](#).

5.12.2 Property Documentation

5.12.2.1 short [] Clv.Viewer.Renderable.Indices [get], [set]

Contains index information to be uploaded as an indexbuffer, indexing into the vertexbuffer

Implements [Clv.Viewer.IRenderable](#).

5.12.2.2 Matrix Clv.Viewer.Renderable.Transform [get], [set]

Transformation which is applied before rendering

Implements [Clv.Viewer.IRenderable](#).

5.12.2.3 VertexPositionTexture [] Clv.Viewer.Renderable.VertexData [get], [set]

Contains VertexData to be uploaded as a vertexbuffer

Implements [Clv.Viewer.IRenderable](#).

The documentation for this class was generated from the following file:

- Clv/Clv.Viewer/Clv.Viewer/IRenderable.cs

5.13 Clv.Viewer.Math.SplitInfo Class Reference

Contains splitting information from the triangle split stage

Public Member Functions

- [SplitCase GetSplitCase \(\)](#)
Derives SplitCase from NewPoints array after split
- [IEnumerable](#)
[< VertexPositionTexture > GetNewPoints \(\)](#)
Flattens out the NewPoints array

Public Attributes

- [SplitState State](#) = [SplitState.NoSplit](#)
Type of Split
- [Tuple< Vertex2, Vertex2 > \[\] NewPoints](#)
Contains the new points for each edge, in case there are any. Otherwise null.
- [Tuple< Vertex2, Vertex2 > NewPointsFocus](#)
Contains new points in case of a focus split
- [SplitCaseFocus StateFocus](#) = [SplitCaseFocus.Invalid](#)
States which edge was split during focus split

5.13.1 Detailed Description

Contains splitting information from the triangle split stage

5.13.2 Member Function Documentation

5.13.2.1 [IEnumerable<VertexPositionTexture> Clv.Viewer.Math.SplitInfo.GetNewPoints \(\)](#)

Flattens out the NewPoints array

Returns

Returns an IEnum of XnaVertexPositionTexture of the new points

5.13.2.2 [SplitCase Clv.Viewer.Math.SplitInfo.GetSplitCase \(\)](#)

Derives SplitCase from NewPoints array after split

5.13.3 Member Data Documentation

5.13.3.1 [Tuple<Vertex2, Vertex2> \[\] Clv.Viewer.Math.SplitInfo.NewPoints](#)

Contains the new points for each edge, in case there are any. Otherwise null.

5.13.3.2 [Tuple<Vertex2, Vertex2> Clv.Viewer.Math.SplitInfo.NewPointsFocus](#)

Contains new points in case of a focus split

5.13.3.3 [SplitState Clv.Viewer.Math.SplitInfo.State = SplitState.NoSplit](#)

Type of Split

5.13.3.4 SplitCaseFocus Clv.Viewer.Math.SplitInfo.StateFocus = SplitCaseFocus.Invalid

States which edge was split during focus split

The documentation for this class was generated from the following file:

- Clv/Clv.Viewer/Clv.Viewer/Math/Intersection.cs

5.14 Clv.Viewer.Math.SplitPlane2 Class Reference

SplitPlane defined by focus point and normal vector n. Concept similar to normal plane but is delimited by Focus-Point. Triangle hit by splitplane at focuspoint needs special split processing.

Public Member Functions

- **SplitPlane2** (Vector2 fp, Vector2 n)
- float **Height** (Vector2 p)
Normal distance of a point to plane
- bool **IsTriangleAbovePlane** (Triangle2 tri)
- bool **Intersects** (Triangle2 tri)
- bool **Intersects** (Bounds2 b)

Properties

- Vector2 **FocusPoint** [get, set]
- Vector2 **Normal** [get, set]
- float **Distance** [get, set]

5.14.1 Detailed Description

SplitPlane defined by focus point and normal vector n. Concept similar to normal plane but is delimited by Focus-Point. Triangle hit by splitplane at focuspoint needs special split processing.

5.14.2 Member Function Documentation

5.14.2.1 float Clv.Viewer.Math.SplitPlane2.Height (Vector2 p)

Normal distance of a point to plane

Parameters

p

Returns

distance

The documentation for this class was generated from the following file:

- Clv/Clv.Viewer/Clv.Viewer/Math/Math.cs

5.15 Clv.Viewer.Math.Triangle2 Class Reference

Class to handle a 2D triangle

Public Member Functions

- **Triangle2** ([Vertex2](#) a, [Vertex2](#) b, [Vertex2](#) c, short index=-1)
- [Edge2\[\]](#) **GetEdges** ()
get three 2D triangle edges
- bool **ContainsPoint** ([Vector2](#) p)
<http://www.blackpawn.com/texts/pointinpoly/default.html> *Check if triangle contains given point based on barycentric coordinates*
- [Vector2](#) **ComputeBaryCentric** ([Vector2](#) p)
calculate barycentric of the given point inside the triangle2

Public Attributes

- short **Index**
- [Vertex2](#) **A**
- [Vertex2](#) **B**
- [Vertex2](#) **C**

5.15.1 Detailed Description

Class to handle a 2D triangle

5.15.2 Member Function Documentation

5.15.2.1 [Vector2](#) Clv.Viewer.Math.Triangle2.ComputeBaryCentric ([Vector2](#) p)

calculate barycentric of the given point inside the triangle2

Parameters

p

Returns

uv vector with barycentric coordinates

5.15.2.2 bool Clv.Viewer.Math.Triangle2.ContainsPoint ([Vector2](#) p)

<http://www.blackpawn.com/texts/pointinpoly/default.html> Check if triangle contains given point based on barycentric coordinates

Parameters

p

Returns

true if point is inside the triangle

5.15.2.3 Edge2 [] Clv.Viewer.Math.Triangle2.GetEdges ()

get three 2D triangle edges

Returns

Edge-Array with three edges based on vertices

The documentation for this class was generated from the following file:

- Clv/Clv.Viewer/Clv.Viewer/Math/Math.cs

5.16 Clv.Viewer.Math.Vertex2 Class Reference

Class to handle a 2-Dimensional vertex with an index and texture coordinates

Public Member Functions

- **Vertex2** (short index, Vector2 pos, Vector2 coord)
- VertexPositionTexture [ToVertexPositionTexture](#) ()
create an xna VertexPositionTexture with a Vector2

Static Public Member Functions

- static [Vertex2 Interpolate](#) (Vertex2 a, Vertex2 b, float t, short newIndex=-1)
Interpolation of two given vectors

Public Attributes

- short **Index**
- Vector2 **Position**
- Vector2 **TexCoord**

5.16.1 Detailed Description

Class to handle a 2-Dimensional vertex with an index and texture coordinates

5.16.2 Member Function Documentation

5.16.2.1 static Vertex2 Clv.Viewer.Math.Vertex2.Interpolate (Vertex2 a, Vertex2 b, float t, short newIndex = -1) [static]

Interpolation of two given vectors

Parameters

<i>a</i>	first interpolation vector
<i>b</i>	second interpolation vector
<i>t</i>	interpolant
<i>newIndex</i>	new Index if changed

Returns

interpolated vector

5.16.2.2 VertexPositionTexture Clv.Viewer.Math.Vertex2.ToVertexPositionTexture ()

create an xna VertexPositionTexture with a Vector2

Returns

the created VertexPositionTexture

The documentation for this class was generated from the following file:

- Clv/Clv.Viewer/Clv.Viewer/Math/Math.cs

Index

- AddTexture
 - Clv::Viewer::App, 10
- App
 - Clv::Viewer::App, 10
- Bounds
 - Clv::Viewer::Grid, 17
- Clv, 7
- Clv.Viewer, 7
- Clv.Viewer.App, 9
- Clv.Viewer.ComplexLogView, 11
- Clv.Viewer.ConcreteRenderable, 12
- Clv.Viewer.Controls, 13
- Clv.Viewer.CutOffFisheyeEffect, 14
- Clv.Viewer.DefaultEffect, 14
- Clv.Viewer.Grid, 15
- Clv.Viewer.IRenderable, 18
- Clv.Viewer.Math, 8
- Clv.Viewer.Math.Bounds2, 11
- Clv.Viewer.Math.Edge2, 15
- Clv.Viewer.Math.Line2, 19
- Clv.Viewer.Math.SplitInfo, 20
- Clv.Viewer.Math.SplitPlane2, 22
- Clv.Viewer.Math.Triangle2, 23
- Clv.Viewer.Math.Vertex2, 24
- Clv.Viewer.Renderable, 19
- Clv::Viewer::App
 - AddTexture, 10
 - App, 10
 - Draw, 10
 - Initialize, 10
 - LoadContent, 10
 - MoveFocusPoint, 10
 - NextTexture, 11
 - PrevTexture, 11
 - ResetProjection, 11
 - Update, 11
- Clv::Viewer::ConcreteRenderable
 - Dispose, 12
 - IndexBuffer, 13
 - Transform, 13
 - VertexBuffer, 13
- Clv::Viewer::Controls
 - UpdateKeyboardControls, 13
 - UpdateMouseControls, 14
- Clv::Viewer::Grid
 - Bounds, 17
 - DisposeBuffers, 16
 - GetConcreteRenderable, 16
 - Grid, 16
 - Indices, 17
 - LastSplitPlane, 17
 - Split, 16
 - SplitGrid, 17
 - Transform, 17
 - Triangles, 17
 - VertexData, 17
 - Vertices, 17
- Clv::Viewer::IRenderable
 - GetConcreteRenderable, 18
 - Indices, 18
 - Transform, 18
 - VertexData, 19
- Clv::Viewer::Math
 - SplitCase, 8
 - SplitCaseFocus, 8
 - SplitState, 8
- Clv::Viewer::Math::SplitInfo
 - GetNewPoints, 21
 - GetSplitCase, 21
 - NewPoints, 21
 - NewPointsFocus, 21
 - State, 21
 - StateFocus, 21
- Clv::Viewer::Math::SplitPlane2
 - Height, 22
- Clv::Viewer::Math::Triangle2
 - ComputeBaryCentric, 23
 - ContainsPoint, 23
 - GetEdges, 23
- Clv::Viewer::Math::Vertex2
 - Interpolate, 24
 - ToVertexPositionTexture, 25
- Clv::Viewer::Renderable
 - GetConcreteRenderable, 20
 - Indices, 20
 - Transform, 20
 - VertexData, 20
- ComputeBaryCentric
 - Clv::Viewer::Math::Triangle2, 23
- ContainsPoint
 - Clv::Viewer::Math::Triangle2, 23
- Dispose
 - Clv::Viewer::ConcreteRenderable, 12
- DisposeBuffers
 - Clv::Viewer::Grid, 16
- Draw
 - Clv::Viewer::App, 10

- GetConcreteRenderable
 - Clv::Viewer::Grid, 16
 - Clv::Viewer::IRenderable, 18
 - Clv::Viewer::Renderable, 20
- GetEdges
 - Clv::Viewer::Math::Triangle2, 23
- GetNewPoints
 - Clv::Viewer::Math::SplitInfo, 21
- GetSplitCase
 - Clv::Viewer::Math::SplitInfo, 21
- Grid
 - Clv::Viewer::Grid, 16
- Height
 - Clv::Viewer::Math::SplitPlane2, 22
- IndexBuffer
 - Clv::Viewer::ConcreteRenderable, 13
- Indices
 - Clv::Viewer::Grid, 17
 - Clv::Viewer::IRenderable, 18
 - Clv::Viewer::Renderable, 20
- Initialize
 - Clv::Viewer::App, 10
- Interpolate
 - Clv::Viewer::Math::Vertex2, 24
- LastSplitPlane
 - Clv::Viewer::Grid, 17
- LoadContent
 - Clv::Viewer::App, 10
- MoveFocusPoint
 - Clv::Viewer::App, 10
- NewPoints
 - Clv::Viewer::Math::SplitInfo, 21
- NewPointsFocus
 - Clv::Viewer::Math::SplitInfo, 21
- NextTexture
 - Clv::Viewer::App, 11
- PrevTexture
 - Clv::Viewer::App, 11
- ResetProjection
 - Clv::Viewer::App, 11
- Split
 - Clv::Viewer::Grid, 16
- SplitCase
 - Clv::Viewer::Math, 8
- SplitCaseFocus
 - Clv::Viewer::Math, 8
- SplitGrid
 - Clv::Viewer::Grid, 17
- SplitState
 - Clv::Viewer::Math, 8
- State
 - Clv::Viewer::Math::SplitInfo, 21
- StateFocus
 - Clv::Viewer::Math::SplitInfo, 21
- ToVertexPositionTexture
 - Clv::Viewer::Math::Vertex2, 25
- Transform
 - Clv::Viewer::ConcreteRenderable, 13
 - Clv::Viewer::Grid, 17
 - Clv::Viewer::IRenderable, 18
 - Clv::Viewer::Renderable, 20
- Triangles
 - Clv::Viewer::Grid, 17
- Update
 - Clv::Viewer::App, 11
- UpdateKeyBoardControls
 - Clv::Viewer::Controls, 13
- UpdateMouseControls
 - Clv::Viewer::Controls, 14
- VertexBuffer
 - Clv::Viewer::ConcreteRenderable, 13
- VertexData
 - Clv::Viewer::Grid, 17
 - Clv::Viewer::IRenderable, 19
 - Clv::Viewer::Renderable, 20
- Vertices
 - Clv::Viewer::Grid, 17