AdVdGlyphRecognition

Ángel David Monteagudo Jiménez

Version 1.1
11/24/2015 11:54:00 AM

Table of Contents

Class Index

Class List

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

Namespace Documentation

AdVd Namespace Reference

Namespaces

- namespace GlyphRecognition

Classes

- class UIEditorUtility

UI editor utility. Finds or creates a canvas to be the parent of a new UI object.

Class Documentation

CapStretchStrokeGraphic Class Reference

Draws glyphs and strokes with caps and stretches the center of the texture.

Public Member Functions

- void SetStrokes (Glyph glyph)
  Sets the renderer to draw the strokes of a glyph.
- void SetStrokes (Stroke[] strokes)
  Sets the renderer to draw a set of strokes.
- void ClearStrokes ()
  Sets the renderer to draw nothing.

Public Attributes

- float capSize = 0.5f
  The size of the cap relative to the width.
- float relativeWidth = 0.02f
  Relative width of the strokes.

Protected Member Functions

- override void BuildStrokeMesh (Stroke s, VertexHelper vh)
  Fills the vertex helper to build the stroke mesh.
- override void OnPopulateMesh (VertexHelper vh)

Protected Attributes

- Vector2 scale = Vector2.one
- float width

Properties

- override Texture mainTexture `[get]`
- bool IsClear `[get]`
  Check whether the renderer should be clear or drawing strokes.

Detailed Description

Draws glyphs and strokes with caps and stretches the center of the texture.

Member Function Documentation

override void BuildStrokeMesh (Stroke *s*, VertexHelper *vh*) `[protected]`, `[virtual]`

Fills the vertex helper to build the stroke mesh.

Parameters:

s	Stroke to draw.
vh	Vertex helper.

Implements StrokeGraphic.

void ClearStrokes () `[inherited]`

Sets the renderer to draw nothing.

override void OnPopulateMesh (VertexHelper *vh*) `[protected]`, `[inherited]`

void SetStrokes (Glyph *glyph*) `[inherited]`

Sets the renderer to draw the strokes of a glyph.

Parameters:

glyph	Glyph.

void SetStrokes (Stroke[] *strokes*) `[inherited]`

Sets the renderer to draw a set of strokes.

Parameters:

strokes	Strokes.

Member Data Documentation

float capSize = 0.5f

The size of the cap relative to the width.

float relativeWidth = 0.02f `[inherited]`

Relative width of the strokes.

Vector2 scale = Vector2.one `[protected]`, `[inherited]`

float width `[protected]`, `[inherited]`

Property Documentation

bool IsClear `[get]`, `[inherited]`

Check whether the renderer should be clear or drawing strokes.

`true` if this renderer is clear; otherwise, `false`.

override Texture mainTexture `[get]`, `[inherited]`

Glyph Class Reference

Public Member Functions

- void DrawGlyph (Vector2 position, Vector2 scale)
  Draws the glyph using gizmos.
- void Normalize ()
  Normalize this glyph.
- void Resample (float sampleDistance=0.05f)
  Resample this glyph with the specified sampleDistance. A sample distance sorter than 1e-3 does nothing.
- override string ToString ()
- override bool Equals (object obj)
- override int GetHashCode ()

Static Public Member Functions

- static Glyph CreateGlyph (Stroke[] strokes=null)
  Creates a normalized glyph.
- static Glyph CreateGlyph (Stroke[] strokes, float sampleDistance)
  Creates a glyph and resamples its strokes.
- static bool operator== (Glyph a, Glyph b)
- static bool operator!= (Glyph a, Glyph b)

Properties

- int Length `[get]`
- Stroke this[int index] `[get]`

Member Function Documentation

static Glyph CreateGlyph (Stroke[] *strokes* = `null`) `[static]`

Creates a normalized glyph.

Returns:

The glyph.

Parameters:

strokes	Strokes.

static Glyph CreateGlyph (Stroke[] *strokes*, float *sampleDistance*) `[static]`

Creates a glyph and resamples its strokes.

Returns:

The glyph.

Parameters:

strokes	Strokes.
sampleDistance	Sample distance.

void DrawGlyph (Vector2 *position*, Vector2 *scale*)

Draws the glyph using gizmos.

Parameters:

position	Position.
scale	Scale.

override bool Equals (object *obj*)

override int GetHashCode ()

void Normalize ()

Normalize this glyph.

static bool operator!= (Glyph *a*, Glyph *b*) `[static]`

static bool operator== (Glyph *a*, Glyph *b*) `[static]`

void Resample (float *sampleDistance* = `0.05f`)

Resample this glyph with the specified sampleDistance. A sample distance sorter than 1e-3 does nothing.

Parameters:

sampleDistance	Sample distance.

override string ToString ()

Property Documentation

int Length `[get]`

Stroke this[int index] `[get]`

GlyphDisplay Class Reference

Glyph display component. Needs a stroke graphic component to work.

Public Member Functions

- void RebuildGlyph ()
  Rebuilds the glyph.

Properties

- Glyph glyph `[get, set]`
  Gets or sets the glyph to display.

Detailed Description

Glyph display component. Needs a stroke graphic component to work.

Member Function Documentation

void RebuildGlyph ()

Rebuilds the glyph.

Property Documentation

Glyph glyph`[get], [set]`

Gets or sets the glyph to display.

The glyph.

GlyphDisplayEditor Class Reference

Public Member Functions

- override void OnInspectorGUI ()

Static Public Member Functions

- static void CreateGlyphDisplay (MenuCommand menuCommand)

Member Function Documentation

static void CreateGlyphDisplay (MenuCommand *menuCommand*)`[static]`

override void OnInspectorGUI ()

GlyphDrawInput Class Reference

UI component to draw glyphs and find the closest match within a set of stored glyphs using a specific matching method.

Classes

- class GlyphCastEvent

Glyph cast event.

It contains the index of the closest glyph matched and the info of the match.

Public Types

- enum Matching_Method { None =-1, SqrDistanceDTWMatchingMethod, SqrDTWMatchingMemoryCostMethod, SqrDistanceMatchingMethod, SqrMemoryMatchingMethod, LegendreMatchingMethod }
- enum Series_Generator { None =-1, LegendreSeries, LegendreSobolevSeries }

Public Member Functions

- delegate void StrokeDraw (Stroke[] strokes)
- delegate void PointDraw (Vector2[] points)
- void OnBeginDrag (PointerEventData eventData)
- void OnDrag (PointerEventData eventData)
- void OnEndDrag (PointerEventData eventData)
- void OnPointerClick (PointerEventData eventData)
- bool Cast ()

  Casts the currently drawn glyph. Return false if there is no glyph to cast. Use PerformCast(true) to recast.
- bool Cast (Glyph glyph)

  Cast the specified glyph. Return true if glyph is not null.
- void PerformCast (bool recast=false)

  Cast the currently drawn glyph or recast previous glyph if there is no glyph drawn.
- void ClearInput ()

  Clears the current input.

Public Attributes

- Glyph castedGlyph
- GlyphMatch currentMatch
- GlyphSet targetGlyphSet

  The set of glyphs to compare with the casted glyph.
- float normalizedGlyphSize =0.8f

  The size of a normalized glyph relative to the component.
- float sampleDistance =0.05f

  The sample distance when drawing and resampling glyphs.
- bool castOnTap =true

  Set castOnTap to true if you want to trigger a glyph cast by tapping on the component.
- bool overrideThreshold =false

  Set to true to override the default theshold used by the matching method.
- GlyphCastEvent OnGlyphCast

  The event to listen for glyph casts.
- StrokeDraw OnStrokeDraw

  Delegate called when a new stroke is finished.
- PointDraw OnPointDraw

  Delegate called when the stroke currently being drawn changes.

Properties

- float Threshold `[get, set]`

  Gets or sets the threshold used by the matching method. The field overrideThreshold must be true in order to set the threshold.
- Matching_Method Method `[get, set]`

  Gets or sets the matching method. Re-setting the method (Method=Method) re-instances it.
- Series_Generator SeriesGenerator `[get, set]`

  Gets or sets the series generator.
- float Alpha `[get, set]`

Gets or sets the alpha value used in "memory" matching methods. The bigger is alpha, more error is forgiven.

- float SobolevFactor `[get, set]`
  Gets or sets the factor used in Legendre-Sobolev series generator. For a value of 0 Legendre-Sobolev and Legendre series are the same.

Detailed Description

UI component to draw glyphs and find the closest match within a set of stored glyphs using a specific matching method.

Member Enumeration Documentation

enum Matching_Method `[strong]`

Enumerator

None
SqrDistanceDTWMatchingMethod
SqrDTWMatchingMemoryCostMethod
SqrDistanceMatchingMethod
SqrMemoryMatchingMethod
LegendreMatchingMethod

enum Series_Generator `[strong]`

Enumerator

None
LegendreSeries
LegendreSobolevSeries

Member Function Documentation

bool Cast ()

Casts the currently drawn glyph. Return false if there is no glyph to cast. Use PerformCast(true) to recast.

bool Cast (Glyph *glyph*)

Cast the specified glyph. Return true if glyph is not null.

Parameters:

glyph	Glyph.

void ClearInput ()

Clears the current input.

void OnBeginDrag (PointerEventData *eventData*)

void OnDrag (PointerEventData *eventData*)

void OnEndDrag (PointerEventData *eventData*)

void OnPointerClick (PointerEventData *eventData*)

void PerformCast (bool *recast* = `false`)

Cast the currently drawn glyph or recast previous glyph if there is no glyph drawn.

Parameters:

recast	If set to `true` and there is no glyph drawn recast previous glyph.

delegate void PointDraw (Vector2[] *points*)

delegate void StrokeDraw (Stroke[] *strokes*)

Member Data Documentation

Glyph castedGlyph

bool castOnTap =true

Set castOnTap to true if you want to trigger a glyph cast by tapping on the component.

GlyphMatch currentMatch

float normalizedGlyphSize =0.8f

The size of a normalized glyph relative to the component.

GlyphCastEvent OnGlyphCast

The event to listen for glyph casts.

PointDraw OnPointDraw

Delegate called when the stroke currently being drawn changes.

StrokeDraw OnStrokeDraw

Delegate called when a new stroke is finished.

bool overrideThreshold =false

Set to true to override the default theshold used by the matching method.

float sampleDistance =0.05f

The sample distance when drawing and resampling glyphs.

GlyphSet targetGlyphSet

The set of glyphs to compare with the casted glyph.

Property Documentation

float Alpha`[get], [set]`

Gets or sets the alpha value used in "memory" matching methods. The bigger is alpha, more error is forgiven.

The alpha.

Matching_Method Method`[get], [set]`

Gets or sets the matching method. Re-setting the method (Method=Method) re-instances it.

The method.

Series_Generator SeriesGenerator`[get], [set]`

Gets or sets the series generator.

The series generator.

float SobolevFactor `[get]`, `[set]`

Gets or sets the factor used in Legendre-Sobolev series generator. For a value of 0 Legendre-Sobolev and Legendre series are the same.

The Sovolev Factor.

float Threshold `[get]`, `[set]`

Gets or sets the threshold used by the matching method. The field overrideThreshold must be true in order to set the threshold.

The threshold.

GlyphDrawInput.GlyphCastEvent Class Reference

Glyph cast event. It contains the index of the closest glyph matched and the info of the match.

Detailed Description

Glyph cast event. It contains the index of the closest glyph matched and the info of the match.

GlyphDrawInputEditor Class Reference

Public Member Functions

- override void OnInspectorGUI ()

Static Public Member Functions

- static void CreateGlyphInput (MenuCommand menuCommand)
  Create a glyph input.

Member Function Documentation

static void CreateGlyphInput (MenuCommand *menuCommand*) `[static]`

Create a glyph input.

Parameters:

menuCommand	Menu command.

override void OnInspectorGUI ()

GlyphEditor Class Reference

Public Member Functions

- override void OnInspectorGUI ()
- void GLDrawGlyph ()
  Draw the glyph using GL calls.
- void DrawGlyphHandleLines ()
  Draws the glyph handle lines.
- void Resample (float sampleDist)
  Resample glyph and record undo.
- void Normalize ()
  Normalize glyph and record undo.
- void DrawGlyphPointHandles ()
  Draws the glyph point handles.
- void DrawGlyphPointDeleteHandles ()
  Draws the glyph point delete handles.
- void DrawGlyphEdgeHandles ()
  Draws the glyph add-point-to-edge handles.
- void DrawGlyphStrokeHandles (bool delete=false)
  Draws the glyph stroke handles.
- void AddStroke (Vector2[] newStroke)
  Adds a stroke.
- override void OnPreviewSettings ()
- override bool HasPreviewGUI ()
- override void OnPreviewGUI (Rect r, GUIStyle background)

Static Public Member Functions

- static void CreateGlyph ()

Member Function Documentation

void AddStroke (Vector2[] *newStroke*)

Adds a stroke.

Parameters:

newStroke	New stroke.

static void CreateGlyph () `[static]`

void DrawGlyphEdgeHandles ()

Draws the glyph add-point-to-edge handles.

void DrawGlyphHandleLines ()

Draws the glyph handle lines.

void DrawGlyphPointDeleteHandles ()

Draws the glyph point delete handles.

void DrawGlyphPointHandles ()

Draws the glyph point handles.

void DrawGlyphStrokeHandles (bool *delete* = `false`)

Draws the glyph stroke handles.

Parameters:

delete	If set to `true` draws delete handles, if `flase` draws move handles.

void GLDrawGlyph ()

Draw the glyph using GL calls.

override bool HasPreviewGUI ()

void Normalize ()

Normalize glyph and record undo.

override void OnInspectorGUI ()

override void OnPreviewGUI (Rect *r*, GUIStyle *background*)

override void OnPreviewSettings ()

void Resample (float *sampleDist*)

Resample glyph and record undo.

Parameters:

sampleDist	Sample dist.

GlyphEditorWindow Class Reference

Public Attributes

- GlyphEditor glyphEditor

Member Data Documentation

GlyphEditor glyphEditor

GlyphMatch Class Reference

Classes

- class StrokeMatch

Public Member Functions

- GlyphMatch (Glyph src, Glyph tgt, StrokeMatch[] matches, float cost, float threshold)
- void DrawLerp (float t, Vector2 position, Vector2 scale)
  Draws the gizmos of a step of the morphing from the source glyph to the target glyph.
- Stroke[] GetLerpStrokes (float t)
  Gets the stroke array of a step of the morphing from the source glyph to the target glyph.

Public Attributes

- Glyph source

Properties

- float Cost `[get]`
  Gets the cost of the match.
- float Threshold `[get]`
  Gets the max cost threshold for the match to be valid.
- bool Valid `[get]`
  Gets a value indicating whether this GlyphMatch is a valid match.

Constructor & Destructor Documentation

GlyphMatch (Glyph *src*, Glyph *tgt*, StrokeMatch[] *matches*, float *cost*, float *threshold*)

Member Function Documentation

void DrawLerp (float *t*, Vector2 *position*, Vector2 *scale*)

Draws the gizmos of a step of the morphing from the source glyph to the target glyph.

Parameters:

t	T.
position	Position.
scale	Scale.

Stroke [] GetLerpStrokes (float *t*)

Gets the stroke array of a step of the morphing from the source glyph to the target glyph.

Returns:

The glyph strokes.

Parameters:

t	T.

Member Data Documentation

Glyph source

Property Documentation

float Cost `[get]`

Gets the cost of the match.

The cost.

float Threshold `[get]`

Gets the max cost threshold for the match to be valid.

The threshold.

bool Valid `[get]`

Gets a value indicating whether this GlyphMatch is a valid match.

`true` if valid; otherwise, `false` .

GlyphMatch.StrokeMatch Class Reference

Public Member Functions

- StrokeMatch (float c, float w, Stroke a, Stroke b, int[] aIndices, int[] bIndices)
- Stroke Lerp (float t)

Public Attributes

- float cost

Properties

- int Length `[get]`
- Vector2 this[int index, float t] `[get]`

Constructor & Destructor Documentation

StrokeMatch (float ***c*****, float** ***w*****, Stroke** ***a*****, Stroke** ***b*****, int[]** ***aIndices*****, int[]** ***bIndices*****)**

Member Function Documentation

Stroke Lerp (float ***t*****)**

Member Data Documentation

float cost

Property Documentation

int Length `[get]`

Vector2 this[int index, float t] `[get]`

GlyphSet Class Reference

Public Member Functions

- IEnumerator GetEnumerator ()

Static Public Member Functions

- static implicit operator Glyph[] (GlyphSet gs)

Properties

- int Length `[get]`
- Glyph this[int index] `[get]`
- Glyph[] Glyphs `[get, set]`
  Gets a copy of the glyphs array or sets the glyphs array.

Member Function Documentation

IEnumerator GetEnumerator ()

static implicit operator Glyph[] (GlyphSet *gs*) `[static]`

Property Documentation

Glyph [] Glyphs `[get], [set]`

Gets a copy of the glyphs array or sets the glyphs array.

The glyphs.

int Length `[get]`

Glyph this[int index] `[get]`

LegendreMatchingMethod Class Reference

A stroke matching method that uses Legendre series distance as the feature distance between strokes. The coefficients of the targets can be precomputed and reused, saving time in the long term. Stroke match time cost: O(k)

Public Member Functions

- LegendreMatchingMethod (int degree, float threshold=defaultThreshold)
- LegendreMatchingMethod (LegendreSeries generator, float threshold=defaultThreshold)
- virtual void SetTargets (params Glyph[] targets)
  Sets the targets. The coefficients of modified glyphs won't be updated if the instance is the same.
- override int MultiMatch (Glyph src, Glyph[] targets, out GlyphMatch bestMatch)
  Set targets, then try to match a glyph with them and get the best match.
- virtual int MultiMatch (Glyph src, out GlyphMatch bestMatch)
  Try to match a glyph with the current targets and get the best match.
- override GlyphMatch Match (Glyph src, Glyph tgt)
  Try to match the specified glyphs. Returns null if the match fails.

Static Public Member Functions

- static int[] HungarianMethod (float[,] costMatrix)
  Perform the Hungarian method with a square cost matrix.

Public Attributes

- const float defaultThreshold = 1.6f
- float threshold
  The max cost threshold of a valid match.

Protected Member Functions

- virtual void InitCoefficientsGenerator ()
- virtual int[] MatchStrokes (Vector2[][] srcGlyphCoeffs, Vector2[][] tgtGlyphCoeffs)
- virtual GlyphMatch FinalizeMatch (Glyph src, Glyph tgt, int[] indexMatch)
- GlyphMatch.StrokeMatch GetStrokeMatch (float error, bool direct)
- virtual float StrokeCoeffDiff (Vector2[] aCoeffs, Vector2[] bCoeffs)
- virtual float InvStrokeCoeffDiff (Vector2[] aCoeffs, Vector2[] bCoeffs)

Protected Attributes

- LegendreSeries legendreGenerator
- int degree
- float[,] error
- bool[,] directMatch
- Stroke srcStroke = null

Properties

- override string Name `[get]`

Detailed Description

A stroke matching method that uses Legendre series distance as the feature distance between strokes. The coefficients of the targets can be precomputed and reused, saving time in the long term. Stroke match time cost: O(k)

Constructor & Destructor Documentation

LegendreMatchingMethod (int *degree*, float *threshold* = `defaultThreshold`)

LegendreMatchingMethod (LegendreSeries *generator*, float *threshold* = `defaultThreshold`)

Member Function Documentation

virtual GlyphMatch FinalizeMatch (Glyph *src*, Glyph *tgt*, int[] *indexMatch*) `[protected]`, `[virtual]`

GlyphMatch.StrokeMatch GetStrokeMatch (float *error*, bool *direct*) `[protected]`

static int [] HungarianMethod (float *costMatrix*[,]) `[static]`, `[inherited]`

Perform the Hungarian method with a square cost matrix.

Returns:

The best match.

Parameters:

costMatrix	Cost matrix.

virtual void InitCoefficientsGenerator () `[protected]`, `[virtual]`

virtual float InvStrokeCoeffDiff (Vector2[] *aCoeffs*, Vector2[] *bCoeffs*) `[protected]`, `[virtual]`

override GlyphMatch Match (Glyph *src*, Glyph *tgt*) `[virtual]`

Try to match the specified glyphs. Returns null if the match fails.

Parameters:

src	Source.
tgt	Target.

Implements MatchingMethod.

virtual int [] MatchStrokes (Vector2 *srcGlyphCoeffs*[][], Vector2 *tgtGlyphCoeffs*[][])`[protected], [virtual]`

override int MultiMatch (Glyph *src*, Glyph[] *targets*, out GlyphMatch *bestMatch*)`[virtual]`

Set targets, then try to match a glyph with them and get the best match.

Returns:

The index of the best match, or -1 if there is no match.

Parameters:

src	Source.
targets	Targets.
bestMatch	Best match info, or null if there is no match.

Reimplemented from MatchingMethod.

virtual int MultiMatch (Glyph *src*, out GlyphMatch *bestMatch*)`[virtual]`

Try to match a glyph with the current targets and get the best match.

Returns:

The index of the best match, or -1 if there is no match.

Parameters:

src	Source.
bestMatch	Best match info, or null if there is no match.

virtual void SetTargets (params Glyph[] *targets*)`[virtual]`

Sets the targets. The coefficients of modified glyphs won't be updated if the instance is the same.

Parameters:

targets	Targets.

virtual float StrokeCoeffDiff (Vector2[] *aCoeffs*, Vector2[] *bCoeffs*)`[protected], [virtual]`

Member Data Documentation

const float defaultThreshold = 1.6f

int degree `[protected]`

bool [,] directMatch `[protected]`

float [,] error `[protected]`

LegendreSeries legendreGenerator `[protected]`

Stroke srcStroke = null `[protected]`

float threshold `[inherited]`

The max cost threshold of a valid match.

Property Documentation

override string Name `[get]`

LegendreSeries Class Reference

Legendre series coefficients generator.

Public Member Functions

- LegendreSeries (int degree)
- void Init ()
  Initialize this instance. Initialize once before using Compute().
- Vector2[] Compute (Stroke stroke)
  Compute the coefficients for the specified stroke. The coefficients of the inverse stroke can be obtained as: ICi =~ Ci·(-1)^i, i = 0, 1, ...
- override string ToString ()

Protected Member Functions

- virtual float LegendreSqrNorm (int k)
- virtual float PolyInnerProduct (int polyA, int polyB)
- void Reset ()
- virtual void GetMoments (Stroke stroke)

Protected Attributes

- int degree
- float[][] legendrePolynomials
- float[] xMomentIntegrals
- float[] yMomentIntegrals

Properties

- int Degree `[get]`
  Gets the degree of the series.

Detailed Description

Legendre series coefficients generator.

Constructor & Destructor Documentation

LegendreSeries (int *degree*)

Member Function Documentation

Vector2 [] Compute (Stroke *stroke*)

Compute the coefficients for the specified stroke. The coefficients of the inverse stroke can be obtained as: ICi =~ Ci·(-1)^i, i = 0, 1, ...

Parameters:

stroke	Stroke.

virtual void GetMoments (Stroke *stroke*) `[protected]`, `[virtual]`

Reimplemented in LegendreSobolevSeries.

void Init ()

Initialize this instance. Initialize once before using Compute().

virtual float LegendreSqrNorm (int *k*) `[protected]`, `[virtual]`

Reimplemented in LegendreSobolevSeries.

virtual float PolyInnerProduct (int *polyA*, int *polyB*) `[protected]`, `[virtual]`

Reimplemented in LegendreSobolevSeries.

void Reset () `[protected]`

override string ToString ()

Member Data Documentation

int degree `[protected]`

float [][] legendrePolynomials `[protected]`

float [] xMomentIntegrals `[protected]`

float [] yMomentIntegrals `[protected]`

Property Documentation

int Degree `[get]`

Gets the degree of the series.

The degree.

LegendreSobolevSeries Class Reference

Legendre-Sobolev series coefficients generator.

Public Member Functions

- LegendreSobolevSeries (int degree, float mu=1f)
- override string ToString ()
- void Init ()
  Initialize this instance. Initialize once before using Compute().
- Vector2[] Compute (Stroke stroke)
  Compute the coefficients for the specified stroke. The coefficients of the inverse stroke can be obtained as: ICi =~ Ci·(-1)^i, i = 0, 1, ...

Protected Member Functions

- override float LegendreSqrNorm (int k)
- override float PolyInnerProduct (int polyA, int polyB)

- override void GetMoments (Stroke stroke)
- void Reset ()

Protected Attributes

- int degree
- float[][] legendrePolynomials
- float[] xMomentIntegrals
- float[] yMomentIntegrals

Properties

- int Degree `[get]`
  Gets the degree of the series.

Detailed Description

Legendre-Sobolev series coefficients generator.

Constructor & Destructor Documentation

LegendreSobolevSeries (int *degree*, float *mu* = `1f`)

Member Function Documentation

Vector2 [] Compute (Stroke *stroke*) `[inherited]`

Compute the coefficients for the specified stroke. The coefficients of the inverse stroke can be obtained as: ICi =~ Ci·(-1)^i, i = 0, 1, ...

Parameters:

stroke	Stroke.

override void GetMoments (Stroke *stroke*) `[protected]`, `[virtual]`

Reimplemented from LegendreSeries.

void Init () `[inherited]`

Initialize this instance. Initialize once before using Compute().

override float LegendreSqrNorm (int *k*) `[protected]`, `[virtual]`

Reimplemented from LegendreSeries.

override float PolyInnerProduct (int *polyA*, int *polyB*) `[protected]`, `[virtual]`

Reimplemented from LegendreSeries.

void Reset () `[protected]`, `[inherited]`

override string ToString ()

Member Data Documentation

int degree `[protected]`, `[inherited]`

float [][] legendrePolynomials `[protected]`, `[inherited]`

float [] xMomentIntegrals `[protected]`, `[inherited]`

float [] yMomentIntegrals `[protected]`, `[inherited]`

Property Documentation

int Degree `[get]`, `[inherited]`

Gets the degree of the series.

The degree.

MatchingMethod Class Reference

Base matching method class with standard MultiMatch method and HungarianMethod implemented.

Public Member Functions

- abstract GlyphMatch Match (Glyph src, Glyph tgt)
  Try to match the specified glyphs. Returns null if the match fails.
- virtual int MultiMatch (Glyph src, Glyph[] targets, out GlyphMatch bestMatch)
  Try to match a glyph with multiple targets and get the best match.

Static Public Member Functions

- static int[] HungarianMethod (float[,] costMatrix)
  Perform the Hungarian method with a square cost matrix.

Public Attributes

- float threshold
  The max cost threshold of a valid match.

Properties

- abstract string Name `[get]`
  Gets the name of the method.

Detailed Description

Base matching method class with standard MultiMatch method and HungarianMethod implemented.

Member Function Documentation

static int [] HungarianMethod (float *costMatrix*[,]) `[static]`

Perform the Hungarian method with a square cost matrix.

Returns:

The best match.

Parameters:

costMatrix	Cost matrix.

abstract GlyphMatch Match (Glyph *src*, Glyph *tgt*) `[pure virtual]`

Try to match the specified glyphs. Returns null if the match fails.

Parameters:

src	Source.
tgt	Target.

Implemented in LegendreMatchingMethod, and StrokeToStrokeMatchingMethod.

virtual int MultiMatch (Glyph *src*, Glyph[] *targets*, out GlyphMatch *bestMatch*) `[virtual]`

Try to match a glyph with multiple targets and get the best match.

Returns:

The index of the best match, or -1 if there is no match.

Parameters:

src	Source.

targets	Targets.
bestMatch	Best match info, or null if there is no match.

Reimplemented in LegendreMatchingMethod.

Member Data Documentation

float threshold

The max cost threshold of a valid match.

Property Documentation

abstract string Name `[get]`

Gets the name of the method.

The name of the method.

RepeatStrokeGraphic Class Reference

Draws glyphs and strokes repeating a texture in the U axis.

Public Member Functions

- void SetStrokes (Glyph glyph)
  Sets the renderer to draw the strokes of a glyph.
- void SetStrokes (Stroke[] strokes)
  Sets the renderer to draw a set of strokes.
- void ClearStrokes ()
  Sets the renderer to draw nothing.

Public Attributes

- float relativeWidth = 0.02f
  Relative width of the strokes.

Protected Member Functions

- override void BuildStrokeMesh (Stroke s, VertexHelper vh)
  Fills the vertex helper to build the stroke mesh.
- override void OnPopulateMesh (VertexHelper vh)

Protected Attributes

- Vector2 scale = Vector2.one
- float width

Properties

- override Texture mainTexture `[get]`
- bool IsClear `[get]`
  Check whether the renderer should be clear or drawing strokes.

Detailed Description

Draws glyphs and strokes repeating a texture in the U axis.

Member Function Documentation

override void BuildStrokeMesh (Stroke *s*, VertexHelper *vh*) `[protected]`, `[virtual]`

Fills the vertex helper to build the stroke mesh.

Parameters:

s	Stroke to draw.
vh	Vertex helper.

Implements StrokeGraphic.

void ClearStrokes () `[inherited]`

Sets the renderer to draw nothing.

override void OnPopulateMesh (VertexHelper *vh*) `[protected]`, `[inherited]`

void SetStrokes (Glyph *glyph*) `[inherited]`

Sets the renderer to draw the strokes of a glyph.

Parameters:

glyph	Glyph.

void SetStrokes (Stroke[] *strokes*) `[inherited]`

Sets the renderer to draw a set of strokes.

Parameters:

strokes	Strokes.

Member Data Documentation

float relativeWidth = 0.02f `[inherited]`

Relative width of the strokes.

Vector2 scale = Vector2.one `[protected]`, `[inherited]`

float width `[protected]`, `[inherited]`

Property Documentation

bool IsClear `[get]`, `[inherited]`

Check whether the renderer should be clear or drawing strokes.

`true` if this renderer is clear; otherwise, `false` .

override Texture mainTexture `[get]`, `[inherited]`

SqrDistanceDTWMatchingMethod Class Reference

DTW matching method using square distance as the feature distance. Stroke match time cost: O(n^2)

Classes

- struct DTWNode

Public Member Functions

- SqrDistanceDTWMatchingMethod (float threshold=defaultThreshold)
- override GlyphMatch Match (Glyph src, Glyph tgt)
  Try to match the specified glyphs. Returns null if the match fails.
- virtual int MultiMatch (Glyph src, Glyph[] targets, out GlyphMatch bestMatch)
  Try to match a glyph with multiple targets and get the best match.

Static Public Member Functions

- static int[] HungarianMethod (float[,] costMatrix)

Perform the Hungarian method with a square cost matrix.

Public Attributes

- const float defaultThreshold = 0.09f
- float threshold
  The max cost threshold of a valid match.

Protected Types

- enum DTWPrev : byte { None = 0, PrevI, PrevJ, PrevIJ }

Protected Member Functions

- virtual void BuildDTW ()
- override GlyphMatch.StrokeMatch GetStrokeMatch ()
- virtual int[] MatchStrokes (Glyph src, Glyph tgt)

Protected Attributes

- DTWNode[,] directDTW
- float[,] error
- GlyphMatch.StrokeMatch[,] matchMatrix
- Stroke srcStroke = null

Properties

- override string Name `[get]`

Detailed Description

DTW matching method using square distance as the feature distance. Stroke match time cost: O(n^2)

Member Enumeration Documentation

enum DTWPrev : byte`[strong]`, `[protected]`

Enumerator

None
PrevI
PrevJ
PrevIJ

Constructor & Destructor Documentation

SqrDistanceDTWMatchingMethod (float *threshold* = `defaultThreshold`)

Member Function Documentation

virtual void BuildDTW () `[protected]`, `[virtual]`

override GlyphMatch.StrokeMatch GetStrokeMatch () `[protected]`, `[virtual]`

Implements StrokeToStrokeMatchingMethod.

Reimplemented in SqrDTWMatchingMemoryCostMethod.

static int [] HungarianMethod (float *costMatrix*[,]) `[static]`, `[inherited]`

Perform the Hungarian method with a square cost matrix.

Returns:

The best match.

Parameters:

costMatrix	Cost matrix.

override GlyphMatch Match (Glyph *src*, Glyph *tgt*) `[virtual]`, `[inherited]`

Try to match the specified glyphs. Returns null if the match fails.

Parameters:

src	Source.
tgt	Target.

Implements MatchingMethod.

virtual int [] MatchStrokes (Glyph *src*, Glyph *tgt*) `[protected]`, `[virtual]`, `[inherited]`

virtual int MultiMatch (Glyph *src*, Glyph[] *targets*, out GlyphMatch *bestMatch*) `[virtual]`, `[inherited]`

Try to match a glyph with multiple targets and get the best match.

Returns:

The index of the best match, or -1 if there is no match.

Parameters:

src	Source.
targets	Targets.
bestMatch	Best match info, or null if there is no match.

Reimplemented in LegendreMatchingMethod.

Member Data Documentation

const float defaultThreshold = 0.09f `[inherited]`

DTWNode [,] directDTW `[protected]`

float [,] error `[protected], [inherited]`

GlyphMatch.StrokeMatch [,] matchMatrix `[protected], [inherited]`

Stroke srcStroke = null `[protected], [inherited]`

float threshold `[inherited]`

The max cost threshold of a valid match.

Property Documentation

override string Name `[get]`

SqrDistanceDTWMatchingMethod.DTWNode Struct Reference

Public Member Functions

- DTWNode (float cost, DTWPrev prevNode=DTWPrev.None)
- DTWNode (float costPI, float costPJ, float costPIJ)

Static Public Member Functions

- static implicit operator float (DTWNode n)
- static DTWNode operator+ (DTWNode node, float c)

Public Attributes

- float cost
- DTWPrev prevNode

Constructor & Destructor Documentation

DTWNode (float *cost*, DTWPrev *prevNode* = `DTWPrev.None`)

DTWNode (float *costPI*, float *costPJ*, float *costPIJ*)

Member Function Documentation

static implicit operator float (DTWNode *n*)`[static]`

static DTWNode operator+ (DTWNode *node*, float *c*)`[static]`

Member Data Documentation

float cost

DTWPrev prevNode

SqrDistanceMatchingMethod Class Reference

Matching method using square distance as the feature distance. Not as good as DTW but faster. Certain deformations may lead to wrong matchings. Stroke match time cost: O(n)

Public Member Functions

- SqrDistanceMatchingMethod (float threshold=defaultThreshold)
- override GlyphMatch Match (Glyph src, Glyph tgt)
  Try to match the specified glyphs. Returns null if the match fails.
- virtual int MultiMatch (Glyph src, Glyph[] targets, out GlyphMatch bestMatch)
  Try to match a glyph with multiple targets and get the best match.

Static Public Member Functions

- static int[] HungarianMethod (float[,] costMatrix)
  Perform the Hungarian method with a square cost matrix.

Public Attributes

- const float defaultThreshold = 0.09f
- float threshold
  The max cost threshold of a valid match.

Protected Member Functions

- override GlyphMatch.StrokeMatch GetStrokeMatch ()
- virtual int[] MatchStrokes (Glyph src, Glyph tgt)

Protected Attributes

- float[,] error
- GlyphMatch.StrokeMatch[,] matchMatrix
- Stroke srcStroke = null

Properties

- override string Name `[get]`

Detailed Description

Matching method using square distance as the feature distance. Not as good as DTW but faster. Certain deformations may lead to wrong matchings. Stroke match time cost: O(n)

Constructor & Destructor Documentation

SqrDistanceMatchingMethod (float ***threshold*** **=** **`defaultThreshold`)**

Member Function Documentation

override GlyphMatch.StrokeMatch GetStrokeMatch () **`[protected]`, `[virtual]`**

Implements StrokeToStrokeMatchingMethod.

static int [] HungarianMethod (float ***costMatrix*****[,])** **`[static]`, `[inherited]`**

Perform the Hungarian method with a square cost matrix.

Returns:

The best match.

Parameters:

costMatrix	Cost matrix.

override GlyphMatch Match (Glyph ***src*****, Glyph** ***tgt*****)** **`[virtual]`, `[inherited]`**

Try to match the specified glyphs. Returns null if the match fails.

Parameters:

src	Source.
tgt	Target.

Implements MatchingMethod.

virtual int [] MatchStrokes (Glyph ***src*****, Glyph** ***tgt*****)** **`[protected]`, `[virtual]`, `[inherited]`**

virtual int MultiMatch (Glyph ***src*****, Glyph[]** ***targets*****, out GlyphMatch** ***bestMatch*****)** **`[virtual]`, `[inherited]`**

Try to match a glyph with multiple targets and get the best match.

Returns:

The index of the best match, or -1 if there is no match.

Parameters:

src	Source.
targets	Targets.
bestMatch	Best match info, or null if there is no match.

Reimplemented in LegendreMatchingMethod.

Member Data Documentation

const float defaultThreshold = 0.09f `[inherited]`

float [,] error `[protected]`, `[inherited]`

GlyphMatch.StrokeMatch [,] matchMatrix `[protected]`, `[inherited]`

Stroke srcStroke = null `[protected]`, `[inherited]`

float threshold `[inherited]`

The max cost threshold of a valid match.

Property Documentation

override string Name `[get]`

SqrDTWMatchingMemoryCostMethod Class Reference

DTW matching method using square distance as the feature distance. A special feature distance that "forgives" errors is used to get the cost. Stroke match time cost: O(n^2)

Public Member Functions

- SqrDTWMatchingMemoryCostMethod (float alpha, float threshold=defaultThreshold)
- override GlyphMatch Match (Glyph src, Glyph tgt)
  Try to match the specified glyphs. Returns null if the match fails.
- virtual int MultiMatch (Glyph src, Glyph[] targets, out GlyphMatch bestMatch)
  Try to match a glyph with multiple targets and get the best match.

Static Public Member Functions

- static int[] HungarianMethod (float[,] costMatrix)
  Perform the Hungarian method with a square cost matrix.

Public Attributes

- const float defaultThreshold = 0.09f
- float threshold
  The max cost threshold of a valid match.

Protected Types

- enum DTWPrev : byte { None = 0, PrevI, PrevJ, PrevIJ }

Protected Member Functions

- float FeatureDistance (Vector2 a, Vector2 b)
- override GlyphMatch.StrokeMatch GetStrokeMatch ()
- virtual void BuildDTW ()
- virtual int[] MatchStrokes (Glyph src, Glyph tgt)

Protected Attributes

- float half1PlusSqrAlpha
- DTWNode[,] directDTW
- float[,] error
- GlyphMatch.StrokeMatch[,] matchMatrix
- Stroke srcStroke = null

Properties

- override string Name `[get]`

Detailed Description

DTW matching method using square distance as the feature distance. A special feature distance that "forgives" errors is used to get the cost. Stroke match time cost: O(n^2)

Member Enumeration Documentation

enum DTWPrev : byte `[strong]`, `[protected]`, `[inherited]`

Enumerator

None
PrevI
PrevJ
PrevIJ

Constructor & Destructor Documentation

SqrDTWMatchingMemoryCostMethod (float *alpha*, float *threshold* = `defaultThreshold`)

Member Function Documentation

virtual void BuildDTW () `[protected], [virtual], [inherited]`

float FeatureDistance (Vector2 *a*, Vector2 *b*) `[protected]`

override GlyphMatch.StrokeMatch GetStrokeMatch () `[protected], [virtual]`

Reimplemented from SqrDistanceDTWMatchingMethod.

static int [] HungarianMethod (float *costMatrix*[,]) `[static], [inherited]`

Perform the Hungarian method with a square cost matrix.

Returns:

The best match.

Parameters:

costMatrix	Cost matrix.

override GlyphMatch Match (Glyph *src*, Glyph *tgt*) `[virtual], [inherited]`

Try to match the specified glyphs. Returns null if the match fails.

Parameters:

src	Source.
tgt	Target.

Implements MatchingMethod.

virtual int [] MatchStrokes (Glyph *src*, Glyph *tgt*) `[protected], [virtual], [inherited]`

virtual int MultiMatch (Glyph *src*, Glyph[] *targets*, out GlyphMatch *bestMatch*) `[virtual], [inherited]`

Try to match a glyph with multiple targets and get the best match.

Returns:

The index of the best match, or -1 if there is no match.

Parameters:

src	Source.

targets	Targets.
bestMatch	Best match info, or null if there is no match.

Reimplemented in LegendreMatchingMethod.

Member Data Documentation

const float defaultThreshold = 0.09f `[inherited]`

DTWNode [,] directDTW `[protected]`, `[inherited]`

float [,] error `[protected]`, `[inherited]`

float half1PlusSqrAlpha `[protected]`

GlyphMatch.StrokeMatch [,] matchMatrix `[protected]`, `[inherited]`

Stroke srcStroke = null `[protected]`, `[inherited]`

float threshold `[inherited]`

The max cost threshold of a valid match.

Property Documentation

override string Name `[get]`

SqrMemoryMatchingMethod Class Reference

Matching method using a special feature distance that "forgives" previous errors. Not as good as DTW but faster. Slightly better than its square distance counterpart. Stroke match time cost: O(n)

Public Member Functions

- SqrMemoryMatchingMethod (float alpha, float threshold=defaultThreshold)
- override GlyphMatch Match (Glyph src, Glyph tgt)
  Try to match the specified glyphs. Returns null if the match fails.
- virtual int MultiMatch (Glyph src, Glyph[] targets, out GlyphMatch bestMatch)
  Try to match a glyph with multiple targets and get the best match.

Static Public Member Functions

- static int[] HungarianMethod (float[,] costMatrix)

Perform the Hungarian method with a square cost matrix.

Public Attributes

- const float defaultThreshold = 0.09f
- float threshold
  The max cost threshold of a valid match.

Protected Member Functions

- float FeatureDistance (Vector2 a, Vector2 b)
- override GlyphMatch.StrokeMatch GetStrokeMatch ()
- virtual int[] MatchStrokes (Glyph src, Glyph tgt)

Protected Attributes

- float half1PlusSqrAlpha
- float[,] error
- GlyphMatch.StrokeMatch[,] matchMatrix
- Stroke srcStroke = null

Properties

- override string Name `[get]`

Detailed Description

Matching method using a special feature distance that "forgives" previous errors. Not as good as DTW but faster. Slightly better than its square distance counterpart. Stroke match time cost: O(n)

Constructor & Destructor Documentation

SqrMemoryMatchingMethod (float *alpha*, float *threshold* = `defaultThreshold`)

Member Function Documentation

float FeatureDistance (Vector2 *a*, Vector2 *b*) `[protected]`

override GlyphMatch.StrokeMatch GetStrokeMatch () `[protected]`, `[virtual]`

Implements StrokeToStrokeMatchingMethod.

static int [] HungarianMethod (float *costMatrix*[,]) `[static]`, `[inherited]`

Perform the Hungarian method with a square cost matrix.

Returns:

The best match.

Parameters:

costMatrix	Cost matrix.

override GlyphMatch Match (Glyph *src*, Glyph *tgt*) `[virtual], [inherited]`

Try to match the specified glyphs. Returns null if the match fails.

Parameters:

src	Source.
tgt	Target.

Implements MatchingMethod.

virtual int [] MatchStrokes (Glyph *src*, Glyph *tgt*) `[protected], [virtual], [inherited]`

virtual int MultiMatch (Glyph *src*, Glyph[] *targets*, out GlyphMatch *bestMatch*) `[virtual], [inherited]`

Try to match a glyph with multiple targets and get the best match.

Returns:

The index of the best match, or -1 if there is no match.

Parameters:

src	Source.
targets	Targets.
bestMatch	Best match info, or null if there is no match.

Reimplemented in LegendreMatchingMethod.

Member Data Documentation

const float defaultThreshold = 0.09f `[inherited]`

float [,] error `[protected], [inherited]`

float half1PlusSqrAlpha `[protected]`

GlyphMatch.StrokeMatch [,] matchMatrix `[protected], [inherited]`

Stroke srcStroke = null `[protected], [inherited]`

float threshold `[inherited]`

The max cost threshold of a valid match.

Property Documentation

override string Name `[get]`

Stroke Class Reference

Public Member Functions

- Stroke (Vector2[] points=null)
- void DrawStroke (Vector2 position, Vector2 scale)
  Draws the stroke using gizmos.
- void Translate (Vector2 position)
  Translate this stroke to the specified position.
- void Scale (Vector2 scale)
  Scale this stroke by specified value.
- void Resample (float sampleDistance)
  Resample this stroke by the specified sampleDistance. A sample distance sorter than 1e-3 does nothing.
- override bool Equals (object obj)
- override int GetHashCode ()

Static Public Member Functions

- static bool operator== (Stroke a, Stroke b)
- static bool operator!= (Stroke a, Stroke b)

Public Attributes

- const float minSampleDistance = 1e-3f

Properties

- int Length `[get]`
- Vector2 this[int index] `[get]`
- Rect Bounds `[get]`
  Gets the bounds of the stroke.

Constructor & Destructor Documentation

Stroke (Vector2[] *points* = `null`)

Member Function Documentation

void DrawStroke (Vector2 *position*, Vector2 *scale*)

Draws the stroke using gizmos.

Parameters:

position	Position.
scale	Scale.

override bool Equals (object *obj*)

override int GetHashCode ()

static bool operator!= (Stroke *a*, Stroke *b*)`[static]`

static bool operator== (Stroke *a*, Stroke *b*)`[static]`

void Resample (float *sampleDistance*)

Resample this stroke by the specified sampleDistance. A sample distance sorter than 1e-3 does nothing.

Parameters:

sampleDistance	Sample distance.

void Scale (Vector2 *scale*)

Scale this stroke by specified value.

Parameters:

scale	Scale.

void Translate (Vector2 *position*)

Translate this stroke to the specified position.

Parameters:

position	Position.

Member Data Documentation

const float minSampleDistance = 1e-3f

Property Documentation

Rect Bounds `[get]`

Gets the bounds of the stroke.

The bounds.

int Length `[get]`

Vector2 this[int index] `[get]`

StrokeGraphic Class Reference

Component for graphical visualization of glyphs and strokes.

Public Member Functions

- void SetStrokes (Glyph glyph)
  Sets the renderer to draw the strokes of a glyph.
- void SetStrokes (Stroke[] strokes)
  Sets the renderer to draw a set of strokes.
- void ClearStrokes ()
  Sets the renderer to draw nothing.

Public Attributes

- float relativeWidth = 0.02f
  Relative width of the strokes.

Protected Member Functions

- override void OnPopulateMesh (VertexHelper vh)
- abstract void BuildStrokeMesh (Stroke s, VertexHelper vh)
  Fills the vertex helper to build the stroke mesh.

Protected Attributes

- Vector2 scale = Vector2.one
- float width

Properties

- override Texture mainTexture `[get]`
- bool IsClear `[get]`
  Check whether the renderer should be clear or drawing strokes.

Detailed Description

Component for graphical visualization of glyphs and strokes.

Member Function Documentation

abstract void BuildStrokeMesh (Stroke *s*, VertexHelper *vh*) `[protected]`, `[pure virtual]`

Fills the vertex helper to build the stroke mesh.

Parameters:

s	Stroke to draw.
vh	Vertex helper.

Implemented in CapStretchStrokeGraphic, and RepeatStrokeGraphic.

void ClearStrokes ()

Sets the renderer to draw nothing.

override void OnPopulateMesh (VertexHelper *vh*) `[protected]`

void SetStrokes (Glyph *glyph*)

Sets the renderer to draw the strokes of a glyph.

Parameters:

glyph	Glyph.

void SetStrokes (Stroke[] *strokes*)

Sets the renderer to draw a set of strokes.

Parameters:

strokes	Strokes.

Member Data Documentation

float relativeWidth = 0.02f

Relative width of the strokes.

Vector2 scale = Vector2.one `[protected]`

float width `[protected]`

Property Documentation

bool IsClear `[get]`

Check whether the renderer should be clear or drawing strokes.

`true` if this renderer is clear; otherwise, `false` .

override Texture mainTexture `[get]`

StrokeToStrokeMatchingMethod Class Reference

Stroke to stroke base matching method. GetStrokeMatch must be implemented.

Public Member Functions

- override GlyphMatch Match (Glyph src, Glyph tgt)
  Try to match the specified glyphs. Returns null if the match fails.
- virtual int MultiMatch (Glyph src, Glyph[] targets, out GlyphMatch bestMatch)
  Try to match a glyph with multiple targets and get the best match.

Static Public Member Functions

- static int[] HungarianMethod (float[,] costMatrix)
  Perform the Hungarian method with a square cost matrix.

Public Attributes

- const float defaultThreshold = 0.09f
- float threshold
  The max cost threshold of a valid match.

Protected Member Functions

- virtual int[] MatchStrokes (Glyph src, Glyph tgt)
- abstract GlyphMatch.StrokeMatch GetStrokeMatch ()

Protected Attributes

- float[,] error

- GlyphMatch.StrokeMatch[,] matchMatrix
- Stroke srcStroke = null

Properties

- abstract string Name `[get]`
  Gets the name of the method.

Detailed Description

Stroke to stroke base matching method. GetStrokeMatch must be implemented.

Member Function Documentation

abstract GlyphMatch.StrokeMatch GetStrokeMatch () `[protected], [pure virtual]`

Implemented in SqrDistanceDTWMatchingMethod, SqrMemoryMatchingMethod, SqrDTWMatchingMemoryCostMethod, and SqrDistanceMatchingMethod.

static int [] HungarianMethod (float *costMatrix*[,]) `[static], [inherited]`

Perform the Hungarian method with a square cost matrix.

Returns:

The best match.

Parameters:

costMatrix	Cost matrix.

override GlyphMatch Match (Glyph *src*, Glyph *tgt*) `[virtual]`

Try to match the specified glyphs. Returns null if the match fails.

Parameters:

src	Source.
tgt	Target.

Implements MatchingMethod.

virtual int [] MatchStrokes (Glyph *src*, Glyph *tgt*) `[protected], [virtual]`

virtual int MultiMatch (Glyph *src*, Glyph[] *targets*, out GlyphMatch *bestMatch*) `[virtual], [inherited]`

Try to match a glyph with multiple targets and get the best match.

Returns:

The index of the best match, or -1 if there is no match.

Parameters:

src	Source.
targets	Targets.
bestMatch	Best match info, or null if there is no match.

Reimplemented in LegendreMatchingMethod.

Member Data Documentation

const float defaultThreshold = 0.09f

float [,] error `[protected]`

GlyphMatch.StrokeMatch [,] matchMatrix `[protected]`

Stroke srcStroke = null `[protected]`

float threshold `[inherited]`

The max cost threshold of a valid match.

Property Documentation

abstract string Name `[get]`, `[inherited]`

Gets the name of the method.

The name of the method.

UIEditorUtility Class Reference

UI editor utility. Finds or creates a canvas to be the parent of a new UI object.

Static Public Member Functions

- static GameObject GetOrCreateCanvasAndEventSystem (MenuCommand menuCommand)
- static void CreateEventSystem ()

Detailed Description

UI editor utility. Finds or creates a canvas to be the parent of a new UI object.

Member Function Documentation

static void CreateEventSystem () `[static]`

static GameObject GetOrCreateCanvasAndEventSystem (MenuCommand *menuCommand*) `[static]`

GlyphDrawer Class Reference

Utility monobehaviour to draw glyphs and strokes. The user may re-implement this class in order to draw get the strokes drawn in a custom way.

Public Attributes

- GlyphDrawInput glyphInput
- StrokeGraphic targetGlyphGraphic

Detailed Description

Utility monobehaviour to draw glyphs and strokes. The user may re-implement this class in order to draw get the strokes drawn in a custom way.

Member Data Documentation

GlyphDrawInput glyphInput

StrokeGraphic targetGlyphGraphic

SampleGlyphDrawer Class Reference

Public Member Functions

- void OnGlyphCast (int index, GlyphMatch match)

Public Attributes

- GlyphDrawInput glyphInput
- StrokeGraphic targetGlyphGraphic
- Material drawMaterial
- AnimationCurve morph
- float morphDuration =1f

Member Function Documentation

void OnGlyphCast (int *index*, GlyphMatch *match*)

Member Data Documentation

Material drawMaterial

GlyphDrawInput glyphInput

AnimationCurve morph

float morphDuration =1f

StrokeGraphic targetGlyphGraphic