Table of Contents

Namespace GTTG.Core.Base

Classes

ObservableObject

Observable object, using .

ViewElement

Element which can be drawn, arranged in layout and placed by strategies.

Visual

Base class for drawable object with draw access checking and draw ownership with hit-testing.

Structs

ViewElement.MinMax

Determines maximal and minimal width and height values considering various user properties states.

Interfaces

IVisual

Contract for object which can be drawn and hit-tested.

Interface IVisual

Contract for object which can be drawn and hit-tested.

Namespace: GTTG.Core.Base

Assembly: cs.temp.dll.dll

Syntax

```
public interface IVisual
```

Properties

CurrentDrawingLayer

Current drawing layer which visual belongs to.

Declaration

```
IDrawingLayer CurrentDrawingLayer { get; }
```

Property Value

TYPE	DESCRIPTION
IDrawingLayer	

Methods

Draw(DrawingCanvas)

Draws content of this visual to `drawingCanvas` if GTTG.Core.Drawing.Canvases.DrawingCanvas.DrawingLayer is same as CurrentDrawingLayer or CurrentDrawingLayer is DefaultDrawingLayer.

Declaration

```
void Draw(DrawingCanvas drawingCanvas)
```

Parameters

TYPE	NAME	DESCRIPTION
DrawingCanvas	drawingCanvas	Canvas which belongs to this visual for drawing.

HasHit(SKPoint)

Hit-tests this target against provided point.

Declaration

```
bool HasHit(SKPoint contentPoint)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	contentPoint	Point against which target is tested, in coordinate system of ContentDrawingCanvas.

Returns

TYPE	DESCRIPTION
System.Boolean	True if target was hit; otherwise false.

PopDrawingLayer()

Change CurrentDrawingLayer which visual belongs by removing top layer from drawing layer stack. If no previous value available, DefaultDrawingLayer is selected.

Declaration

```
void PopDrawingLayer()
```

ProvideVisuals()

Returns all visual children of this visual.

Declaration

```
IEnumerable<IVisual> ProvideVisuals()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<IVisual>	

ProvideVisualsInSameLayer()

Returns visual children of this visual that has same CurrentDrawingLayer.

Declaration

```
IEnumerable<IVisual> ProvideVisualsInSameLayer()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<IVisual>	

PushDrawingLayer(IDrawingLayer)

Change CurrentDrawingLayer to which visual belongs to by adding new value on drawing layer stack.

Declaration

```
void PushDrawingLayer(IDrawingLayer drawingLayer)
```

Parameters

TYPE	NAME	DESCRIPTION
IDrawingLayer	drawingLayer	New drawing layer to which visual belongs to.

Class ObservableObject

Observable object, using .

Inheritance

System.Object
ObservableObject
Visual

Namespace: GTTG.Core.Base

Assembly: cs.temp.dll.dll

Syntax

```
public abstract class ObservableObject : INotifyPropertyChanged
```

Methods

Notify(String)

Notify observers about property changes.

Declaration

```
public void Notify(string propertyName = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	propertyName	Name of property that is changed.

Update<T>(ref T, T, Boolean, String)

Update property backing field and notify observers about property change.

Declaration

```
public bool Update<T>(ref T field, T value, bool notifyIfEqual = false, string propertyName = null)
```

Parameters

TYPE	NAME	DESCRIPTION
T	field	The field to update.
T	value	New value assigned to the field.
System.Boolean	notifyIfEqual	If true, notify is triggered even if `field` and `value` are equal.
System.String	propertyName	Name of property that is changed.

Returns

TYPE	DESCRIPTION
System.Boolean	True if backing field value changed; otherwise false.

Type Parameters

NAME	DESCRIPTION
T	The type of field.

Events

PropertyChanged

Add handler to receive notification if property of derived class is updated.

Declaration

```
public event PropertyChangedEventHandler PropertyChanged
```

Event Type

TYPE	DESCRIPTION
PropertyChangedEventHandler	

Class ViewElement

Element which can be drawn, arranged in layout and placed by strategies.

Inheritance

System.Object

ObservableObject

Visual

ViewElement

Implements

IVisual

Inherited Members

Visual.CurrentDrawingLayer

Visual.PushDrawingLayer(IDrawingLayer)

Visual.PopDrawingLayer()

Visual.IsInDrawingLayer(IDrawingLayer)

Visual.IsDrawableOnCanvas(DrawingCanvas)

Visual.IsInSameLayer(IVisual)

Visual.OnDraw(DrawingCanvas)

Visual.ProvideVisualsInSameLayer()

Visual.ProvideVisuals()

ObservableObject.PropertyChanged

ObservableObject.Notify(String)

ObservableObject.Update<T>(T, T, Boolean, String)

Namespace: GTTG.Core.Base

Assembly: cs.temp.dll.dll

Syntax

```
public abstract class ViewElement : Visual, IVisual
```

Fields

ArrangeMatrix

Matrix used for arrange cycle, equal to parent PlacementMatrix. In comparison to instance's PlacementMatrix does not account margins and transformations.

Declaration

```
protected SKMatrix ArrangeMatrix
```

Field Value

TYPE	DESCRIPTION
SKMatrix	

Properties

ArrangedHeight

Height of element set by or other similar methods.

Declaration

```
public float ArrangedHeight { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

ArrangedWidth

Width of element set by or other similar methods.

Declaration

```
public float ArrangedWidth { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

ArrangeSize

Latest size given to view element in Arrange(SKPoint, SKSize) or Arrange(SKPoint, SKSize, ViewElement). When Rotate(Single, Boolean) and similar operations applied, rearrange uses this value.

Declaration

```
public SKSize ArrangeSize { get; }
```

Property Value

TYPE	DESCRIPTION
SKSize	

BottomMargin

Bottom margin of the element.

Declaration

```
public float BottomMargin { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

BoundingRect

Minimal bounding rectangle with edges perpendicular to axes of cartesian graph around ContentLeftTop, ContentRightTop, ContentRightBottom, ContentLeftBottom.

Declaration

```
public SKRect BoundingRect { get; }
```

Property Value

TYPE	DESCRIPTION
SKRect	

Clip

If applied, any drawing in view element outside this clip area is not visible. Origin and size of clip are both not scaled and rotated; clip is applied to PlacementMatrix.

Declaration

```
public SKRect Clip { get; }
```

Property Value

TYPE	DESCRIPTION
SKRect	

ContentHeight

Width which element occupies on canvas after applied rotate and scale transformations. Margin measures included.

Declaration

```
public float ContentHeight { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

ContentLeftBottom

Position of lower left vertex (with 0 rad rotation) on ContentDrawingCanvas, after applied rotate and scale transformations. Margin measures included.

Declaration

```
public SKPoint ContentLeftBottom { get; }
```

Property Value

TYPE	DESCRIPTION
SKPoint	

ContentLeftTop

Position of upper left vertex (with 0 rad rotation) on ContentDrawingCanvas, after applied rotate and scale transformations. Margin measures included.

Declaration

```
public SKPoint ContentLeftTop { get; }
```

Property Value

TYPE	DESCRIPTION
SKPoint	

ContentRightBottom

Position of right bottom vertex (with 0 rad rotation) on ContentDrawingCanvas, after applied rotate and scale transformations. Margin measures included.

Declaration

```
public SKPoint ContentRightBottom { get; }
```

Property Value

TYPE	DESCRIPTION
SKPoint	

ContentRightTop

Position of right top vertex (with 0 rad rotation) on ContentDrawingCanvas, after applied rotate and scale transformations. Margin measures included.

Declaration

```
public SKPoint ContentRightTop { get; }
```

Property Value

TYPE	DESCRIPTION
SKPoint	

ContentWidth

Height which element occupies on canvas after applied rotate and scale transformations. Margin measures included.

Declaration

```
public float ContentWidth { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

DesiredSize

Desired size of element after latest Measure(SKSize) call.

Declaration

```
public SKSize DesiredSize { get; }
```

Property Value

TYPE	DESCRIPTION
SKSize	

HasClipEnabled

If true, Clip is applied before drawing of the element.

Declaration

```
public bool HasClipEnabled { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Height

Preferred height of element for layout, contains TopMargin and BottomMargin. If set and ArrangeCore(SKSize) not overriden and falls between values of MaxHeight and MinHeight, has same priority behaviour as MaxHeight and MinHeight for ArrangeOverride(SKSize).

Declaration

```
public float Height { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

LeftMargin

Left margin of the element.

Declaration

```
public float LeftMargin { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

MaxHeight

Preferred maximal height of element content for layout. If set and ArrangeCore(SKSize) not overriden, height value from ArrangeOverride(SKSize) has lower priority if max is lower.

Declaration

```
public float MaxHeight { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

MaxWidth

Preferred maximal width of element content for layout. If set and ArrangeCore(SKSize) not overriden, width value from ArrangeOverride(SKSize) has lower priority if max is lower.

Declaration

```
public float MaxWidth { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

MinHeight

Preferred minimal height of element content for layout. If set and ArrangeCore(SKSize) not overriden, height value from ArrangeOverride(SKSize) has lower priority if min is higher.

Declaration

```
public float MinHeight { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

MinWidth

Preferred minimal width of element content for layout. If set and ArrangeCore(SKSize) not overriden, width value from ArrangeOverride(SKSize) has lower priority if min is higher.

Declaration

```
public float MinWidth { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

PlacementMatrix

TransformationMatrix applied to canvas to map to start of element's content. Allows drawing of element's content in it's coordinate system, with Rotation and ScaleFactor applied in the matrix.

Declaration

```
public SKMatrix PlacementMatrix { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
SKMatrix	

RightMargin

Right margin of the element.

Declaration

```
public float RightMargin { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Rotation

Clockwise rotation of view element in radians.

Declaration

```
public float Rotation { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

ScaleFactor

Scale of element, values between 0 and 1 makes element smaller to original scale.

Declaration

```
public float ScaleFactor { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

TopMargin

Top margin of the element.

Declaration

```
public float TopMargin { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

UnscaledHeight

Height of element's content without applying rotate and scale transformations. Does not include margin measures.

Declaration

```
public float UnscaledHeight { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

UnscaledWidth

Width of element's content without applying rotate and scale transformations. Does not include margin measures.

Declaration

```
public float UnscaledWidth { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Width

Preferred width of element for layout, contains LeftMargin and RightMargin. If set and ArrangeCore(SKSize) not overriden and falls between values of MaxWidth and MinWidth, has same priority behaviour as MaxWidth and MinWidth for ArrangeOverride(SKSize).

Declaration

```
public float Width { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

Arrange(SKPoint, SKSize)

Arranges element directly to coordinate system of ContentDrawingCanvas. Resets ScaleFactor and Rotation to default values.

Declaration

```
public void Arrange(SKPoint origin, SKSize size)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	origin	Position in ContentDrawingCanvas.

TYPE	NAME	DESCRIPTION
SKSize	size	Size provided to element.

Arrange(SKPoint, SKSize, ViewElement)

Arranges element to coordinate system of `parentElement`. Resets ScaleFactor and Rotation to default values.

Declaration

```
public void Arrange(SKPoint origin, SKSize size, ViewElement parentElement)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	origin	Position in `parentElement`.
SKSize	size	Size provided to `parentElement`
ViewElement	parentElement	ViewElement whose coordinate system is used to place this view element.

ArrangeCore(SKSize)

Override-able base logic for ArrangeOverride(SKSize) calls.

Declaration

```
protected virtual (SKRect Content, SKSize Size)ArrangeCore(SKSize size)
```

Parameters

TYPE	NAME	DESCRIPTION
SKSize	size	Available size to use. If exceeded, element has empty size and is not arranged.

Returns

TYPE	DESCRIPTION
System.ValueTuple<SKRect, SKSize>	Content positioned in size (size contains margins). Content is moved from [0,0] by TopMargin and LeftMargin.

ArrangeOverride(SKSize)

User measure for this element called from Arrange(SKPoint, SKSize) or similar methods.

Declaration

```
protected virtual SKSize ArrangeOverride(SKSize finalSize)
```

Parameters

TYPE	NAME	DESCRIPTION
SKSize	finalSize	Available size to use.

Returns

TYPE	DESCRIPTION
SKSize	Final size of element which should not exceed `finalSize`.

Draw(DrawingCanvas)

Draws content of this visual to `drawingCanvas` if GTTG.Core.Drawing.Canvases.DrawingCanvas.DrawingLayer is same as CurrentDrawingLayer or CurrentDrawingLayer is DefaultDrawingLayer.

Declaration

```
public override sealed void Draw(DrawingCanvas drawingCanvas)
```

Parameters

TYPE	NAME	DESCRIPTION
DrawingCanvas	drawingCanvas	Canvas which belongs to this visual for drawing.

Overrides

Visual.Draw(DrawingCanvas)

GetBoundingRectangle()

Creates bounding rectangle around points of rectangle determined by ContentLeftTop and ContentRightBottom.

Declaration

```
public SKRect GetBoundingRectangle()
```

Returns

TYPE	DESCRIPTION
SKRect	

HasHit(SKPoint)

Hit-tests this target against provided point.

Declaration

```
public override bool HasHit(SKPoint contentPoint)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
SKPoint	contentPoint	Point against which target is tested, in coordinate system of ContentDrawingCanvas.

Returns

TYPE	DESCRIPTION
System.Boolean	True if target was hit; otherwise false.

Overrides

Visual.HasHit(SKPoint)

IsInView(SKRect)

Determines if this element is in visible area of ContentDrawingCanvas.

Declaration

```
public bool IsInView(SKRect view)
```

Parameters

TYPE	NAME	DESCRIPTION
SKRect	view	Rectangular area in ContentDrawingCanvas representing visible area.

Returns

TYPE	DESCRIPTION
System.Boolean	True if this element is in area of `view`; otherwise false.

Measure(SKSize)

Measures view element. Assigns virtual MeasureCore(SKSize) with no modifications to DesiredSize.

Declaration

```
public void Measure(SKSize availableSize)
```

Parameters

TYPE	NAME	DESCRIPTION
SKSize	availableSize	Recommended available size to use.

MeasureCore(SKSize)

Override-able base logic for MeasureOverride(SKSize) calls. Calls MeasureOverride(SKSize) and modified it's return size if needed. If MeasureOverride(SKSize) returned size does not falls into MaxHeight, MaxWidth, MinHeight, MinWidth, modified to fit. Newly min-max-modified value is then also modified to fit `availableSize`.

Declaration

```
protected virtual SKSize MeasureCore(SKSize availableSize)
```

Parameters

TYPE	NAME	DESCRIPTION
SKSize	availableSize	Maximum to be returned.

Returns

TYPE	DESCRIPTION
SKSize	Value to be assigned to DesiredSize.

MeasureOverride(SKSize)

User measure for this element called from Measure(SKSize).

Declaration

```
protected virtual SKSize MeasureOverride(SKSize availableSize)
```

Parameters

TYPE	NAME	DESCRIPTION
SKSize	availableSize	Recommended available size to use.

Returns

TYPE	DESCRIPTION
SKSize	Measured size of this element by user.

Reposition(SKPoint)

Changes position of view element and then arranges it and it's children with ArrangeSize with the changed position.

Declaration

```
public void Reposition(SKPoint origin)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	origin	Position in ContentDrawingCanvas.

Reposition(SKPoint, ViewElement)

Changes position of view element and then arranges it and it's children with ArrangeSize with the changed position.

Declaration

```
public void Reposition(SKPoint origin, ViewElement viewElement)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	origin	Origin in content of `viewElement` coordinate system.
ViewElement	viewElement	ViewElement in whose content is this view element repositioned.

Rotate(Single, Boolean)

Rotates view element and it's content. Does not rearrange view element to fits it's parent after rotation. Arranges the view element and it's children with ArrangeSize with changed rotation.

Declaration

```
public void Rotate(float radRotation, bool isCombined = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	radRotation	Clockwise rotation in radians to be added to current Rotation.
System.Boolean	isCombined	If true, add rotation to current Rotation. Otherwise resets rotation and sets this value instead.

Scale(Single, Boolean)

Scales view element and it's content. Does not re-arrange view element to fits it's parent after scale. Combines with current ScaleFactor. Arranges the view element and it's children with ArrangeSize with changed scale.

Declaration

```
public void Scale(float scaleMultiple, bool isCombined = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	scaleMultiple	If higher than 0, resizes element by scaling it by provided value. Otherwise does nothing.
System.Boolean	isCombined	If true, multiplies ScaleFactor with current ScaleFactor. Otherwise resets scale and sets this value instead.

Implements

IVisual

Struct ViewElement.MinMax

Determines maximal and minimal width and height values considering various user properties states.

Inherited Members

System.ValueType.Equals(System.Object)

System.ValueType.GetHashCode()

System.ValueType.ToString()

System.Object.Equals(System.Object, System.Object)

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.GetType()

Namespace: GTTG.Core.Base

Assembly: cs.temp.dll.dll

Syntax

```
protected struct MinMax
```

Class Visual

Base class for drawable object with draw access checking and draw ownership with hit-testing.

Inheritance

System.Object
ObservableObject
Visual
ViewElement

Implements

IVisual

Inherited Members

ObservableObject.PropertyChanged
ObservableObject.Notify(String)
ObservableObject.Update<T>(T, T, Boolean, String)

Namespace: GTTG.Core.Base

Assembly: cs.temp.dll.dll

Syntax

```
public abstract class Visual : ObservableObject, IVisual
```

Constructors

Visual()

Creates visual with CurrentDrawingLayer initialized to DefaultDrawingLayer.

Declaration

```
protected Visual()
```

Properties

CurrentDrawingLayer

Current drawing layer which visual belongs to.

Declaration

```
public IDrawingLayer CurrentDrawingLayer { get; }
```

Property Value

TYPE	DESCRIPTION
IDrawingLayer	

Methods

Draw(DrawingCanvas)

Draws content of this visual to `drawingCanvas` if GTTG.Core.Drawing.Canvases.DrawingCanvas.DrawingLayer is same as CurrentDrawingLayer or CurrentDrawingLayer is DefaultDrawingLayer.

Declaration

```
public virtual void Draw(DrawingCanvas drawingCanvas)
```

Parameters

TYPE	NAME	DESCRIPTION
DrawingCanvas	drawingCanvas	Canvas which belongs to this visual for drawing.

HasHit(SKPoint)

Hit-tests this target against provided point.

Declaration

```
public abstract bool HasHit(SKPoint contentPoint)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	contentPoint	Point against which target is tested, in coordinate system of ContentDrawingCanvas.

Returns

TYPE	DESCRIPTION
System.Boolean	True if target was hit; otherwise false.

IsDrawableOnCanvas(DrawingCanvas)

Determines if visual can be drawn by `drawingCanvas`.

Declaration

```
public bool IsDrawableOnCanvas(DrawingCanvas drawingCanvas)
```

Parameters

TYPE	NAME	DESCRIPTION
DrawingCanvas	drawingCanvas	

Returns

TYPE	DESCRIPTION
System.Boolean	True if IsInDrawingLayer(IDrawingLayer) returns true for GTTG.Core.Drawing.Canvases.DrawingCanvas.DrawingLayer.

IsInDrawingLayer(IDrawingLayer)

Returns true if visual lies in `drawingLayer`.

Declaration

```
public bool IsInDrawingLayer(IDrawingLayer drawingLayer)
```

Parameters

TYPE	NAME	DESCRIPTION
IDrawingLayer	drawingLayer	

Returns

TYPE	DESCRIPTION
System.Boolean	True if CurrentDrawingLayer equals to . If CurrentDrawingLayer is DefaultDrawingLayer, returns also true; otherwise false.

IsInSameLayer(IVisual)

Determines if visual has same layer as this instance.

Declaration

```
public bool IsInSameLayer(IVisual visual)
```

Parameters

TYPE	NAME	DESCRIPTION
IVisual	visual	IVisual to compare.

Returns

TYPE	DESCRIPTION
System.Boolean	True if has same layer; otherwise false.

OnDraw(DrawingCanvas)

After drawing checks are done, this method is called for user to apply drawing on .

Declaration

```
protected virtual void OnDraw(DrawingCanvas drawingCanvas)
```

Parameters

TYPE	NAME	DESCRIPTION
DrawingCanvas	drawingCanvas	

PopDrawingLayer()

Change CurrentDrawingLayer which visual belongs by removing top layer from drawing layer stack. If no previous value available, DefaultDrawingLayer is selected.

Declaration

```
public void PopDrawingLayer()
```

ProvideVisuals()

Provides elements drawn in this element in draw order.

Declaration

```
public abstract IEnumerable<IVisual> ProvideVisuals()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<IVisual>	

ProvideVisualsInSameLayer()

Returns visual children of this visual that has same CurrentDrawingLayer.

Declaration

```
public IEnumerable<IVisual> ProvideVisualsInSameLayer()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<IVisual>	

PushDrawingLayer(IDrawingLayer)

Change CurrentDrawingLayer to which visual belongs to by adding new value on drawing layer stack.

Declaration

```
public void PushDrawingLayer(IDrawingLayer drawingOwner)
```

Parameters

TYPE	NAME	DESCRIPTION
IDrawingLayer	drawingOwner	

Implements

IVisual

Namespace GTTG.Core.Component

Classes

GraphicalComponent

Represents scene of outer content rectangle with inner view rectangle inside. Modifications changes layout of those rectangles. ContentHeight maps to ContentDateTimeInterval and ViewHeight maps to ViewDateTimeInterval, forming DateTimeContext.

ViewModifier

Interactive view with scale and translate operations limited by border bounds.

Interfaces

IViewProvider

Providing state and conversions tools of graphical component.

Enums

ResizeTransformationResult

Specifies the resize modification result.

ScaleTransformationResult

Specifies the scale modification result.

TimeModificationResult

Specifies the time modification result.

TranslationTransformationResult

Specifies the translation modification result.

Class GraphicalComponent

Represents scene of outer content rectangle with inner view rectangle inside. Modifications changes layout of those rectangles. ContentHeight maps to ContentDateTimeInterval and ViewHeight maps to ViewDateTimeInterval, forming DateTimeContext.

Inheritance

System.Object
GraphicalComponent

Implements

IViewProvider
INotifyPropertyChanged

Namespace: GTTG.Core.Component

Assembly: cs.temp.dll.dll

Syntax

```
public class GraphicalComponent : ObservableObject, IViewProvider, INotifyPropertyChanged
```

Constructors

GraphicalComponent()

Creates graphical component which must configured be modified for further use.

Declaration

```
public GraphicalComponent()
```

GraphicalComponent(ViewModifier, DateTimeContext)

Creates graphical component configured by provided parameters.

Declaration

```
public GraphicalComponent(ViewModifier viewModifier, DateTimeContext dateTimeContext)
```

Parameters

TYPE	NAME	DESCRIPTION
ViewModifier	viewModifier	Instance of ViewModifier providing component sizes and used as engine.
DateTimeContext	dateTimeContext	DateTimeContext of component.

Properties

ContentDateTimeInterval

ContentDateTimeInterval value of DateTimeContext.

Declaration

```
public DateTimeInterval ContentDateTimeInterval { get; }
```

Property Value

TYPE	DESCRIPTION
DateTimeInterval	

ContentHeight

Height of ContentDrawingCanvas.

Declaration

```
public float ContentHeight { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

ContentMatrix

Transformation matrix for setting accurate area of ContentDrawingCanvas into graphical component.

Declaration

```
public SKMatrix ContentMatrix { get; }
```

Property Value

TYPE	DESCRIPTION
SKMatrix	

ContentWidth

Width of ContentDrawingCanvas.

Declaration

```
public float ContentWidth { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

DateTimeContext

Representation of for ContentDrawingCanvas and ViewDrawingCanvas. Updates via .

Declaration

```
public DateTimeContext DateTimeContext { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
DateTimeContext	

DpiScale

Scale factor of current DPI and device independent pixel DPI. If device independent pixel is 96 DPI and current is 192, value is 2.

Declaration

```
public float DpiScale { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Scale

Scale factor of current view scale and unscaled view. If view is zoomed in twice, scale is 2.

Declaration

```
public float Scale { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

ViewDateTimeInterval

ViewDateTimeInterval value of DateTimeContext.

Declaration

```
public DateTimeInterval ViewDateTimeInterval { get; }
```

Property Value

TYPE	DESCRIPTION
DateTimeInterval	

ViewHeight

Height of ViewDrawingCanvas.

Declaration

```
public float ViewHeight { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

ViewWidth

Width of ViewDrawingCanvas.

Declaration

```
public float ViewWidth { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

ConvertViewToContentLocation(SKPoint)

Converts ViewDrawingCanvas position to ContentDrawingCanvas position.

Declaration

```
public SKPoint ConvertViewToContentLocation(SKPoint viewPoint)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	viewPoint	Position in ViewDrawingCanvas.

Returns

TYPE	DESCRIPTION
SKPoint	Position in ContentDrawingCanvas.

ConvertViewToContentLocation(Single, Single)

Converts ViewDrawingCanvas position to ContentDrawingCanvas position.

Declaration

```
public SKPoint ConvertViewToContentLocation(float x, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	x	Horizontal position in ViewDrawingCanvas.
System.Single	y	Vertical position in ViewDrawingCanvas.

Returns

TYPE	DESCRIPTION
SKPoint	Position in ContentDrawingCanvas.

GetContentHorizontalPosition(DateTime)

Converts to ContentDrawingCanvas position.

Declaration

```
public float GetContentHorizontalPosition(DateTime dateTime)
```

Parameters

TYPE	NAME	DESCRIPTION
DateTime	dateTime	to convert.

Returns

TYPE	DESCRIPTION
System.Single	ContentDrawingCanvas horizontal position.

GetDateTimeFromContent(Single)

Converts ContentDrawingCanvas horizontal position to .

Declaration

```
public DateTime GetDateTimeFromContent(float globalHorizontalPosition)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	globalHorizontalPosition	

Returns

TYPE	DESCRIPTION
DateTime	representation of `contentHorizontalPosition`.

GetDateTimeFromView(Single)

Converts ViewDrawingCanvas horizontal position to .

Declaration

```
public DateTime GetDateTimeFromView(float viewHorizontalPosition)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	viewHorizontalPosition	Horizontal position on ViewDrawingCanvas.

Returns

TYPE	DESCRIPTION
DateTime	representation of `viewHorizontalPosition`.

GetViewHorizontalPosition(DateTime)

Converts to ViewDrawingCanvas position.

Declaration

```
public float GetViewHorizontalPosition(DateTime dateTime)
```

Parameters

TYPE	NAME	DESCRIPTION
DateTime	dateTime	to convert.

Returns

TYPE	DESCRIPTION
System.Single	ViewDrawingCanvas horizontal position.

GetViewRect()

Returns visible area of ContentDrawingCanvas in graphical component in coordinates of ContentDrawingCanvas.

Declaration

```
public SKRect GetViewRect()
```

Returns

TYPE	DESCRIPTION
SKRect	

TryChangeBorderHeight(Single)

Changes ContentHeight.

Declaration

```
public ResizeTransformationResult TryChangeBorderHeight(float height)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	height	New height of ContentHeight.

Returns

TYPE	DESCRIPTION
ResizeTransformationResult	ViewModified if applied; otherwise ViewUnmodified.

TryChangeContentTime(DateTimeInterval)

Assigns new to ContentHeight in content area. Does not modify view area.

Declaration

```
public TimeModificationResult TryChangeContentTime(DateTimeInterval contentDateTimeInterval)
```

Parameters

TYPE	NAME	DESCRIPTION
DateTimeInterval	contentDateTimeInterval	New ContentDateTimeInterval value. If not in ViewDateTimeInterval, modification is not applied.

Returns

TYPE	DESCRIPTION
TimeModificationResult	TimeModified if successful; otherwise TimeUnmodified.

TryChangeDateTimeContext(DateTimeContext)

Assigns new ViewDateTimeInterval to ViewHeight and then proportionally resizes ContentHeight to map itself to ContentDateTimeInterval with view area in right position.

Declaration

```
public TimeModificationResult TryChangeDateTimeContext(DateTimeContext dateTimeContext)
```

Parameters

TYPE	NAME	DESCRIPTION
DateTimeContext	dateTimeContext	New DateTimeContext.

Returns

TYPE	DESCRIPTION
TimeModificationResult	TimeModified if successful; otherwise TimeUnmodified.

TryChangeViewTime(DateTimeInterval)

Assigns new to ViewHeight in view area. ContentHeight is proportionally resized to match new DateTimeContext. Vertical position of view area in content area is not modified.

Declaration

```
public TimeModificationResult TryChangeViewTime(DateTimeInterval viewDateTimeInterval)
```

Parameters

TYPE	NAME	DESCRIPTION
DateTimeInterval	viewDateTimeInterval	New ViewDateTimeInterval value. If not in ContentDateTimeInterval, modification is not applied.

Returns

TYPE	DESCRIPTION
TimeModificationResult	TimeModified if successful; otherwise TimeUnmodified.

TryResizeContentArea(Single, Single)

Resizes the content area. View area is not resized or moved. Modification is not executed if view area would exceed content area after modification.

Declaration

```
public ResizeTransformationResult TryResizeContentArea(float globalWidth, float globalHeight)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	globalWidth	New ContentWidth value.
System.Single	globalHeight	New ContentHeight value.

Returns

TYPE	DESCRIPTION
ResizeTransformationResult	ViewModified if successful; otherwise ViewUnmodified.

TryResizeView(Single, Single)

Resizes the view area. Global area's ContentWidth and ContentHeight are proportionally resized to have same view:global ratio as before. View is placed in global area proportionally on the same position.

Declaration

```
public ResizeTransformationResult TryResizeView(float viewWidth, float viewHeight)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
System.Single	viewWidth	New ViewWidth value.
System.Single	viewHeight	New ViewHeight value.

Returns

TYPE	DESCRIPTION
ResizeTransformationResult	ViewModified if successful; otherwise ViewUnmodified.

TryScale(SKPoint, Single)

Changes view area by scaling ViewWidth and ViewHeight. Scaling is applied on `origin` point in view which has the same visual position in scaled view. If view area after scaling exceeds content area, translation of view area is applied with `origin` no longer being on visually same position.

Declaration

```
public ScaleTransformationResult TryScale(SKPoint origin, float delta)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	origin	Point in view which has the same visual position after scaling.
System.Single	delta	Difference between original and new Scale value. If new value is 3 and old is 1, difference is 2. In reverse -2.

Returns

TYPE	DESCRIPTION
ScaleTransformationResult	ViewModifiedWithSameOrigin if view is modified and `origin` in same view's position. ViewModifiedWithTransformedOrigin if view is modified and `origin` was transformed as view would be out of border bounds. ViewUnmodified otherwise.

TryTranslate(SKPoint)

Moves the view area in global area by translating it with provided vector. If view would leave the content area, translation is not applied.

Declaration

```
public TranslationTransformationResult TryTranslate(SKPoint translationVector)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	translationVector	Vector represented by in which points of view area are translated. Y-axis increases downwards, X-axis increases left to right.

Returns

TYPE	DESCRIPTION
TranslationTransformationResult	ViewUnmodified if translated view would leave content area; otherwise ViewModified.

Implements

IViewProvider

INotifyPropertyChanged

Interface IViewProvider

Providing state and conversions tools of graphical component.

Namespace: GTTG.Core.Component

Assembly: cs.temp.dll.dll

Syntax

```
public interface IViewProvider : INotifyPropertyChanged
```

Properties

ContentHeight

Height of ContentDrawingCanvas.

Declaration

```
float ContentHeight { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

ContentMatrix

Transformation matrix for setting accurate area of ContentDrawingCanvas into graphical component.

Declaration

```
SKMatrix ContentMatrix { get; }
```

Property Value

TYPE	DESCRIPTION
SKMatrix	

ContentWidth

Width of ContentDrawingCanvas.

Declaration

```
float ContentWidth { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

DateTimeContext

Representation of for ContentDrawingCanvas and ViewDrawingCanvas. Updates via .

Declaration

```
DateTimeContext DateTimeContext { get; }
```

Property Value

TYPE	DESCRIPTION
DateTimeContext	

DpiScale

Scale factor of current DPI and device independent pixel DPI. If device independent pixel is 96 DPI and current is 192, value is 2.

Declaration

```
float DpiScale { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Scale

Scale factor of current view scale and unscaled view. If view is zoomed in twice, scale is 2.

Declaration

```
float Scale { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

ViewHeight

Height of ViewDrawingCanvas.

Declaration

```
float ViewHeight { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

ViewWidth

Width of ViewDrawingCanvas.

Declaration

```
float ViewWidth { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

ConvertViewToContentLocation(SKPoint)

Converts ViewDrawingCanvas position to ContentDrawingCanvas position.

Declaration

```
SKPoint ConvertViewToContentLocation(SKPoint viewPoint)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	viewPoint	Position in ViewDrawingCanvas.

Returns

TYPE	DESCRIPTION
SKPoint	Position in ContentDrawingCanvas.

ConvertViewToContentLocation(Single, Single)

Converts ViewDrawingCanvas position to ContentDrawingCanvas position.

Declaration

```
SKPoint ConvertViewToContentLocation(float x, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	x	Horizontal position in ViewDrawingCanvas.
System.Single	y	Vertical position in ViewDrawingCanvas.

Returns

TYPE	DESCRIPTION
SKPoint	Position in ContentDrawingCanvas.

GetContentHorizontalPosition(DateTime)

Converts to ContentDrawingCanvas position.

Declaration

```
float GetContentHorizontalPosition(DateTime dateTime)
```

Parameters

TYPE	NAME	DESCRIPTION
DateTime	dateTime	to convert.

Returns

TYPE	DESCRIPTION
System.Single	ContentDrawingCanvas horizontal position.

GetDateTimeFromContent(Single)

Converts ContentDrawingCanvas horizontal position to .

Declaration

```
DateTime GetDateTimeFromContent(float contentHorizontalPosition)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	contentHorizontalPosition	Horizontal position on ContentDrawingCanvas.

Returns

TYPE	DESCRIPTION
DateTime	representation of `contentHorizontalPosition`.

GetDateTimeFromView(Single)

Converts ViewDrawingCanvas horizontal position to .

Declaration

```
DateTime GetDateTimeFromView(float viewHorizontalPosition)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	viewHorizontalPosition	Horizontal position on ViewDrawingCanvas.

Returns

TYPE	DESCRIPTION
DateTime	representation of `viewHorizontalPosition`.

GetViewHorizontalPosition(DateTime)

Converts to ViewDrawingCanvas position.

Declaration

```
float GetViewHorizontalPosition(DateTime dateTime)
```

Parameters

TYPE	NAME	DESCRIPTION
DateTime	dateTime	to convert.

Returns

TYPE	DESCRIPTION
System.Single	ViewDrawingCanvas horizontal position.

GetViewRect()

Returns visible area of ContentDrawingCanvas in graphical component in coordinates of ContentDrawingCanvas.

Declaration

```
SKRect GetViewRect()
```

Returns

TYPE	DESCRIPTION
SKRect	

Enum ResizeTransformationResult

Specifies the resize modification result.

Namespace: GTTG.Core.Component

Assembly: cs.temp.dll.dll

Syntax

```
public enum ResizeTransformationResult
```

Fields

NAME	DESCRIPTION
ViewModified	Equal to modification success.
ViewUnmodified	Equal to modification failure.

Enum ScaleTransformationResult

Specifies the scale modification result.

Namespace: GTTG.Core.Component

Assembly: cs.temp.dll.dll

Syntax

```
public enum ScaleTransformationResult
```

Fields

NAME	DESCRIPTION
ViewModifiedWithSameOrigin	Equal to modification success with origin point on same position in view.
ViewModifiedWithTransformedOrigin	Equal to modification success with translated origin point which no longer has same position in view.
ViewUnmodified	Equal to modification failure.

Enum TimeModificationResult

Specifies the time modification result.

Namespace: GTTG.Core.Component

Assembly: cs.temp.dll.dll

Syntax

```
public enum TimeModificationResult
```

Fields

NAME	DESCRIPTION
TimeModified	Equal to modification success.
TimeUnmodified	Equal to modification failure.

Enum TranslationTransformationResult

Specifies the translation modification result.

Namespace: GTTG.Core.Component

Assembly: cs.temp.dll.dll

Syntax

```
public enum TranslationTransformationResult
```

Fields

NAME	DESCRIPTION
ViewModified	Equal to modification success.
ViewUnmodified	Equal to modification failure.

Class ViewModifier

Interactive view with scale and translate operations limited by border bounds.

Inheritance

System.Object
ViewModifier

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: GTTG.Core.Component

Assembly: cs.temp.dll.dll

Syntax

```
public class ViewModifier
```

Constructors

ViewModifier(Single, Single, Single, Single)

Creates view in border of specified sizes with view attached to the top left corner of the border.

Declaration

```
public ViewModifier(float viewWidth, float viewHeight, float borderWidth, float borderHeight)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	viewWidth	Initial ViewWidth value.
System.Single	viewHeight	Initial ViewHeight value.
System.Single	borderWidth	Initial BorderWidth value.
System.Single	borderHeight	Initial BorderHeight value.

Properties

BorderHeight

Vertical length of border where view exists.

Declaration

```
public float BorderHeight { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

BorderWidth

Horizontal length of border where view exists.

Declaration

```
public float BorderWidth { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

ViewHeight

Vertical length of view.

Declaration

```
public float ViewHeight { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

ViewMatrix

TransformationMatrix which positions view to border.

Declaration

```
public SKMatrix ViewMatrix { get; }
```

Property Value

TYPE	DESCRIPTION
SKMatrix	

ViewWidth

Horizontal length of view.

Declaration

```
public float ViewWidth { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

ComputeBorderWidth(Single, DateTimeContext)

Computes border width from and ViewWidth values. Can be used to get the value for ViewModifier construction.

Declaration

```
public static float ComputeBorderWidth(float viewWidth, DateTimeContext dateTimeContext)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	viewWidth	Horizontal length of view.
DateTimeContext	dateTimeContext	represents border values. represents view values.

Returns

TYPE	DESCRIPTION
System.Single	Length of BorderWidth determined from parameters.

ConvertViewPositionToContentPosition(SKPoint)

Converts location of point in view to it's canvas location.

Declaration

```
public SKPoint ConvertViewPositionToContentPosition(SKPoint viewLocation)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	viewLocation	Location of point in view.

Returns

TYPE	DESCRIPTION
SKPoint	Location of point in canvas.

TryResizeBorder(Single, Single)

Resizes the border where view exists. If view's position would be out of border bounds, view is transitioned.

Declaration

```
public ResizeTransformationResult TryResizeBorder(float newBorderWidth, float newBorderHeight)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	newBorderWidth	New horizontal width of the border.
System.Single	newBorderHeight	New vertical width of the border.

Returns

TYPE	DESCRIPTION
ResizeTransformationResult	ViewUnmodified if new border is not fitting into view, otherwise ViewModified.

TryResizeBorder(Single, Single, Single, Single)

Resizes the border where view exists and positions view at new provided location.

Declaration

```
public ResizeTransformationResult TryResizeBorder(float newBorderWidth, float newBorderHeight, float
viewHorizontalOffset, float viewVerticalOffset)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	newBorderWidth	New horizontal width of the border.
System.Single	newBorderHeight	New vertical width of the border.
System.Single	viewHorizontalOffset	Horizontal offset from border origin [0,0] where view should be positioned.
System.Single	viewVerticalOffset	Vertical offset from border origin [0,0] where view should be positioned.

Returns

TYPE	DESCRIPTION
ResizeTransformationResult	ViewUnmodified if new border is not fitting into view or view with provided offset is out of border bounds, otherwise ViewModified

TryResizeView(Single, Single)

Resizes the current view and border.

Declaration

```
public ResizeTransformationResult TryResizeView(float newViewWidth, float newViewHeight)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	newViewWidth	New horizontal width of view.
System.Single	newViewHeight	New vertical width of view.

Returns

TYPE	DESCRIPTION
ResizeTransformationResult	Result state of the operation. If operation cannot be applied, view is left unmodified.

TryScale(SKPoint, Single)

Transforms view by scaling it in both axes with added delta. Scale values applied only in positive numbers greater or equal one.

Declaration

```
public ScaleTransformationResult TryScale(SKPoint origin, float delta)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	origin	Point of the current view which is not transformed by scale operation. If scale operation leaves border bounds, origin is also transformed.
System.Single	delta	Difference between original and new scale

Returns

TYPE	DESCRIPTION
ScaleTransformationResult	ViewModifiedWithSameOrigin if view is modified and `origin` in same view's position. ViewModifiedWithTransformedOrigin if view is modified and `origin` was transformed as view would be out of border bounds. ViewUnmodified otherwise.

TryTranslate(SKPoint)

Transforms view by translating it with provided vector. If view would leave border, transformation is not applied.

Declaration

```
public TranslationTransformationResult TryTranslate(SKPoint translationVector)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
SKPoint	translationVector	Vector in which direction is the move translated.

Returns

TYPE	DESCRIPTION
TranslationTransformationResult	ViewUnmodified if translated view would leave border; otherwise ViewModified.

Namespace GTTG.Core.Drawing.Canvases

Classes

CanvasFactory

Implementation of ICanvasFactory for ContentDrawingCanvas, ViewDrawingCanvas and DefaultDrawingLayer.

ContentDrawingCanvas

Represents canvas covering whole displayable content.

DefaultDrawingCanvas

Singleton abstract canvas for DefaultDrawingLayer with no drawn content.

SpecificDrawingCanvas

Represents specific canvas used for some scenario of drawing.

ViewDrawingCanvas

Represents canvas covering currently displayed content.

Structs

DrawingCanvas

Represents abstract canvas wrapping with applied transformation of to enable drawing in specific area of ContentDrawingCanvas content.

Interfaces

ICanvasFactory

Contract for object creating DrawingCanvas from provided layer and canvas.

Class CanvasFactory

Implementation of ICanvasFactory for ContentDrawingCanvas, ViewDrawingCanvas and DefaultDrawingLayer.

Inheritance

System.Object
CanvasFactory

Implements

ICanvasFactory

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: GTTG.Core.Drawing.Canvases

Assembly: cs.temp.dll.dll

Syntax

```
public class CanvasFactory : ICanvasFactory
```

Constructors

CanvasFactory(IViewProvider)

Allows configuration of supported canvases by state of .

Declaration

```
public CanvasFactory(IViewProvider viewProvider)
```

Parameters

TYPE	NAME	DESCRIPTION
IViewProvider	viewProvider	Receives state of represented by this instance.

Methods

CreateCanvas(IDrawingLayer, SKCanvas)

Creates drawing canvas from drawing layer and canvas.

Declaration

```
public DrawingCanvas CreateCanvas(IDrawingLayer drawingLayer, SKCanvas skCanvas)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
IDrawingLayer	drawingLayer	IDrawingLayer to create canvas from. GTTG.Core.Drawing.Canvases.DrawingCanvas.DrawingLayer is set to this value.
SKCanvas	skCanvas	whose can be modified by calling this method.

Returns

TYPE	DESCRIPTION
DrawingCanvas	Instance of DrawingCanvas.

Implements

ICanvasFactory

Class ContentDrawingCanvas

Represents canvas covering whole displayable content.

Inheritance

System.Object

SpecificDrawingCanvas

ContentDrawingCanvas

Inherited Members

SpecificDrawingCanvas.DrawingCanvas

SpecificDrawingCanvas.ChangeDrawingLayer(IDrawingLayer)

System.Object.ToString()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

Namespace: GTTG.Core.Drawing.Canvases

Assembly: cs.temp.dll.dll

Syntax

```
public sealed class ContentDrawingCanvas : SpecificDrawingCanvas
```

Methods

Update(IViewProvider, SKCanvas)

Updates DrawingCanvas with provided parameters.

Declaration

```
public override void Update(IViewProvider viewProvider, SKCanvas canvas)
```

Parameters

TYPE	NAME	DESCRIPTION
IViewProvider	viewProvider	Information about state of .
SKCanvas	canvas	Canvas which state can be modified as used for transformed drawing.

Overrides

SpecificDrawingCanvas.Update(IViewProvider, SKCanvas)

Class DefaultDrawingCanvas

Singleton abstract canvas for DefaultDrawingLayer with no drawn content.

Inheritance

System.Object
SpecificDrawingCanvas
DefaultDrawingCanvas

Inherited Members

SpecificDrawingCanvas.DrawingCanvas
SpecificDrawingCanvas.ChangeDrawingLayer(IDrawingLayer)
System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: GTTG.Core.Drawing.Canvases

Assembly: cs.temp.dll.dll

Syntax

```
public sealed class DefaultDrawingCanvas : SpecificDrawingCanvas
```

Properties

Get

Get instance of singleton.

Declaration

```
public static DefaultDrawingCanvas Get { get; }
```

Property Value

TYPE	DESCRIPTION
DefaultDrawingCanvas	

Methods

Update(IViewProvider, SKCanvas)

Updates DrawingCanvas with provided parameters.

Declaration

```
public override void Update(IViewProvider viewProvider, SKCanvas canvas)
```

Parameters

TYPE	NAME	DESCRIPTION
IViewProvider	viewProvider	Information about state of .

TYPE	NAME	DESCRIPTION
SKCanvas	canvas	Canvas which state can be modified as used for transformed drawing.

Overrides

SpecificDrawingCanvas.Update(IViewProvider, SKCanvas)

Struct DrawingCanvas

Represents abstract canvas wrapping with applied transformation of to enable drawing in specific area of ContentDrawingCanvas content.

Inherited Members

System.ValueType.Equals(System.Object)

System.ValueType.GetHashCode()

System.ValueType.ToString()

System.Object.Equals(System.Object, System.Object)

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.GetType()

Namespace: GTTG.Core.Drawing.Canvases

Assembly: cs.temp.dll.dll

Syntax

```
public struct DrawingCanvas
```

Constructors

DrawingCanvas(IDrawingLayer, SKCanvas, SKSize, SKRect)

Creates drawing canvas backed by Skia canvas.

Declaration

```
public DrawingCanvas(IDrawingLayer drawingLayer, SKCanvas canvas, SKSize size, SKRect view)
```

Parameters

TYPE	NAME	DESCRIPTION
IDrawingLayer	drawingLayer	GTTG.Core.Drawing.Canvases.DrawingCanvas.DrawingLayer value.
SKCanvas	canvas	Backing canvas. Should be transformation matrix modified, is called outside.
SKSize	size	Size value.
SKRect	view	Positioned rectangle which is compared to placement of to skip it's draw call out of canvas.

Fields

Canvas

wrapped by this canvas.

Declaration

```
public readonly SKCanvas Canvas
```

Field Value

TYPE	DESCRIPTION
SKCanvas	

Properties

Height

Height of this canvas.

Declaration

```
public float Height { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Size

Size of this canvas.

Declaration

```
public SKSize Size { get; }
```

Property Value

TYPE	DESCRIPTION
SKSize	

SourceCanvasMatrix

TransformationMatrix of content canvas like ContentDrawingCanvas or ViewDrawingCanvas where this canvas resides.

Declaration

```
public float[] SourceCanvasMatrix { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single[]	

Width

Width of this canvas.

Declaration

```
public float Width { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

Draw(IVisual)

Draws visual element on canvas.

Declaration

```
public void Draw(IVisual visual)
```

Parameters

TYPE	NAME	DESCRIPTION
IVisual	visual	

Interface ICanvasFactory

Contract for object creating DrawingCanvas from provided layer and canvas.

Namespace: GTTG.Core.Drawing.Canvases

Assembly: cs.temp.dll.dll

Syntax

```
public interface ICanvasFactory
```

Methods

CreateCanvas(IDrawingLayer, SKCanvas)

Creates drawing canvas from drawing layer and canvas.

Declaration

```
DrawingCanvas CreateCanvas(IDrawingLayer drawingLayer, SKCanvas skCanvas)
```

Parameters

TYPE	NAME	DESCRIPTION
IDrawingLayer	drawingLayer	IDrawingLayer to create canvas from. GTTG.Core.Drawing.Canvases.DrawingCanvas.DrawingLayer is set to this value.
SKCanvas	skCanvas	whose can be modified by calling this method.

Returns

TYPE	DESCRIPTION
DrawingCanvas	Instance of DrawingCanvas.

Class SpecificDrawingCanvas

Represents specific canvas used for some scenario of drawing.

Inheritance

System.Object

SpecificDrawingCanvas

ContentDrawingCanvas

DefaultDrawingCanvas

ViewDrawingCanvas

Inherited Members

System.Object.ToString()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

Namespace: GTTG.Core.Drawing.Canvases

Assembly: cs.temp.dll.dll

Syntax

```
public abstract class SpecificDrawingCanvas
```

Properties

DrawingCanvas

Instance of DrawingCanvas representing this canvas.

Declaration

```
public DrawingCanvas DrawingCanvas { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
DrawingCanvas	

Methods

ChangeDrawingLayer(IDrawingLayer)

Updates drawing layer of DrawingCanvas.

Declaration

```
public void ChangeDrawingLayer(IDrawingLayer drawingLayer)
```

Parameters

TYPE	NAME	DESCRIPTION
IDrawingLayer	drawingLayer	New `drawingLayer` of DrawingCanvas.

Update(IViewProvider, SKCanvas)

Updates DrawingCanvas with provided parameters.

Declaration

```
public abstract void Update(IViewProvider viewProvider, SKCanvas canvas)
```

Parameters

TYPE	NAME	DESCRIPTION
IViewProvider	viewProvider	Information about state of .
SKCanvas	canvas	Canvas which state can be modified as used for transformed drawing.

Class ViewDrawingCanvas

Represents canvas covering currently displayed content.

Inheritance

System.Object
SpecificDrawingCanvas
ViewDrawingCanvas

Inherited Members

SpecificDrawingCanvas.DrawingCanvas
SpecificDrawingCanvas.ChangeDrawingLayer(IDrawingLayer)
System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: GTTG.Core.Drawing.Canvases

Assembly: cs.temp.dll.dll

Syntax

```
public sealed class ViewDrawingCanvas : SpecificDrawingCanvas
```

Methods

Update(IViewProvider, SKCanvas)

Updates DrawingCanvas with provided parameters.

Declaration

```
public override void Update(IViewProvider viewProvider, SKCanvas canvas)
```

Parameters

TYPE	NAME	DESCRIPTION
IViewProvider	viewProvider	Information about state of .
SKCanvas	canvas	Canvas which state can be modified as used for transformed drawing.

Overrides

SpecificDrawingCanvas.Update(IViewProvider, SKCanvas)

Namespace GTTG.Core.Drawing.Layers

Classes

ContentDrawingLayer

IDrawingLayer representing layer of ContentDrawingCanvas.

DefaultDrawingLayer

Singleton helper structure representing layer which is always positive on check in Visual.

DrawingLayer

Represents drawing of one layer as logical set of components.

DrawingManager

Manages drawing of added IDrawingLayers by defined order.

ViewDrawingLayer

IDrawingLayer representing layer of ViewDrawingCanvas.

Interfaces

IDrawingLayer

Represents drawing of one layer as logical set of components.

IRegisteredLayersOrder

Defines order of registered IDrawingLayer drawing layers.

Class ContentDrawingLayer

IDrawingLayer representing layer of ContentDrawingCanvas.

Inheritance

System.Object

DrawingLayer

ContentDrawingLayer

Implements

IDrawingLayer

IVisual

Inherited Members

DrawingLayer.Draw(DrawingCanvas)

DrawingLayer.OnDraw(DrawingCanvas)

System.Object.ToString()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

Namespace: GTTG.Core.Drawing.Layers

Assembly: cs.temp.dll.dll

Syntax

```
public abstract class ContentDrawingLayer : DrawingLayer, IDrawingLayer, IVisual
```

Properties

CurrentDrawingLayer

Declaration

```
public IDrawingLayer CurrentDrawingLayer { get; }
```

Property Value

TYPE	DESCRIPTION
IDrawingLayer	

Methods

HasHit(SKPoint)

Declaration

```
public bool HasHit(SKPoint contentPoint)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	contentPoint	

Returns

TYPE	DESCRIPTION
System.Boolean	

PopDrawingLayer()

Does nothing as object represents.

Declaration

```
public void PopDrawingLayer()
```

ProvideVisuals()

Declaration

```
public abstract IEnumerable<IVisual> ProvideVisuals()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<IVisual>	

ProvideVisualsInSameLayer()

Declaration

```
public IEnumerable<IVisual> ProvideVisualsInSameLayer()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<IVisual>	

PushDrawingLayer(IDrawingLayer)

Does nothing as object represents.

Declaration

```
public void PushDrawingLayer(IDrawingLayer drawingLayer)
```

Parameters

TYPE	NAME	DESCRIPTION
IDrawingLayer	drawingLayer	

Implements

IDrawingLayer
IVisual

Class DefaultDrawingLayer

Singleton helper structure representing layer which is always positive on check in Visual.

Inheritance

System.Object
DrawingLayer
DefaultDrawingLayer

Implements

IDrawingLayer

Inherited Members

DrawingLayer.Draw(DrawingCanvas)
System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: GTTG.Core.Drawing.Layers

Assembly: cs.temp.dll.dll

Syntax

```
public sealed class DefaultDrawingLayer : DrawingLayer, IDrawingLayer
```

Properties

Get

Gets singleton instance.

Declaration

```
public static DefaultDrawingLayer Get { get; }
```

Property Value

TYPE	DESCRIPTION
DefaultDrawingLayer	

Methods

OnDraw(DrawingCanvas)

Draws layer's inner content.

Declaration

```
protected override void OnDraw(DrawingCanvas drawingCanvas)
```

Parameters

TYPE	NAME	DESCRIPTION
DrawingCanvas	drawingCanvas	

Overrides

DrawingLayer.OnDraw(DrawingCanvas)

Implements

IDrawingLayer

Class DrawingLayer

Represents drawing of one layer as logical set of components.

Inheritance

System.Object
DrawingLayer
ContentDrawingLayer
DefaultDrawingLayer
ViewDrawingLayer

Implements

IDrawingLayer

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: GTTG.Core.Drawing.Layers

Assembly: cs.temp.dll.dll

Syntax

```
public abstract class DrawingLayer : IDrawingLayer
```

Methods

Draw(DrawingCanvas)

Draws the layer onto `drawingCanvas`.

Declaration

```
public virtual void Draw(DrawingCanvas drawingCanvas)
```

Parameters

TYPE	NAME	DESCRIPTION
DrawingCanvas	drawingCanvas	

OnDraw(DrawingCanvas)

Draws layer's inner content.

Declaration

```
protected abstract void OnDraw(DrawingCanvas drawingCanvas)
```

Parameters

TYPE	NAME	DESCRIPTION
DrawingCanvas	drawingCanvas	

Implements

IDrawingLayer

Class DrawingManager

Manages drawing of added IDrawingLayers by defined order.

Inheritance

System.Object
DrawingManager

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: GTTG.Core.Drawing.Layers

Assembly: cs.temp.dll.dll

Syntax

```
public class DrawingManager
```

Constructors

DrawingManager(ICanvasFactory, IRegisteredLayersOrder)

Creates empty DrawingManager with registered layers from IRegisteredLayersOrder.

Declaration

```
public DrawingManager(ICanvasFactory canvasFactory, IRegisteredLayersOrder registeredLayersOrder)
```

Parameters

TYPE	NAME	DESCRIPTION
ICanvasFactory	canvasFactory	Factory to create DrawingCanvas for each IDrawingLayer.
IRegisteredLayersOrder	registeredLayersOrder	Ordered list of types of registered layers to be placed in defined order by ReplaceRegisteredDrawingLayer(IDrawingLayer, Int32)

Properties

Layers

Ordered list of layers starting from the undermost one.

Declaration

```
public IReadOnlyList<(IDrawingLayer DrawingLayer, bool IsRegistered)> Layers { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.IReadOnlyList<System.ValueTuple<IDrawingLayer, System.Boolean>>	

Methods

AddOnCurrentBottom(IDrawingLayer)

Inserts `drawingLayer` into the drawn layers as the current undermost layer.

Declaration

```
public void AddOnCurrentBottom(IDrawingLayer drawingLayer)
```

Parameters

TYPE	NAME	DESCRIPTION
IDrawingLayer	drawingLayer	The drawing layer to insert.

Exceptions

TYPE	CONDITION
System.ArgumentNullException	`drawingLayer` is null reference.

AddOnCurrentTop(IDrawingLayer)

Inserts `drawingLayer` into the drawn layers as the current topmost layer.

Declaration

```
public void AddOnCurrentTop(IDrawingLayer drawingLayer)
```

Parameters

TYPE	NAME	DESCRIPTION
IDrawingLayer	drawingLayer	The drawing layer to insert.

Exceptions

TYPE	CONDITION
System.ArgumentNullException	`drawingLayer` is null reference.

Draw(SKSurface)

Draws layers in order defined in Layers on surface.

Declaration

```
public void Draw(SKSurface surface)
```

Parameters

TYPE	NAME	DESCRIPTION
SKSurface	surface	

GetDrawingLayer<T>()

Get layer by type.

Declaration

```
public T GetDrawingLayer<T>()
    where T : IDrawingLayer
```

Returns

TYPE	DESCRIPTION
T	Instance of IDrawingLayer of type `T` if found; otherwise default of `T`.

Type Parameters

NAME	DESCRIPTION
T	SegmentType of layer to get.

GetDrawingLayersFromUndermostOne()

Returns all drawing layers in order from undermost to topmost one.

Declaration

```
public IEnumerable<IDrawingLayer> GetDrawingLayersFromUndermostOne()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<IDrawingLayer>	

Insert(Int32, IDrawingLayer)

Inserts `drawingLayer` into the drawn layers at the specified `index`.

Declaration

```
public void Insert(int index, IDrawingLayer drawingLayer)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	The zero-based index at which `drawingLayer` should be inserted.
IDrawingLayer	drawingLayer	The drawing layer to insert.

Exceptions

TYPE	CONDITION
System.ArgumentNullException	`drawingLayer` is null reference.
System.ArgumentOutOfRangeException	`index` is less than 0.-or- `index` is greater than List<T>.Count.

RemoveDrawingLayer(Int32)

Removes instance of drawing layer from Layers.

Declaration

```
public void RemoveDrawingLayer(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	If entry under index is registered, entry is not removed and instance is set to DefaultDrawingLayer. Otherwise, entry is removed from Layers.

Exceptions

TYPE	CONDITION
System.ArgumentOutOfRangeException	Provided `index` out of range in Layers.

ReplaceRegisteredDrawingLayer(IDrawingLayer, Int32)

Replaces instance of drawing layer in registration of type of `drawingLayer` from IRegisteredLayersOrder.

Declaration

```
public void ReplaceRegisteredDrawingLayer(IDrawingLayer drawingLayer, int registeredIndex = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
IDrawingLayer	drawingLayer	Instance of drawing layer that replaces previous instance under the registration.
System.Int32	registeredIndex	In case of multiple registrations of same type, zero-based index selects the registration.

Exceptions

TYPE	CONDITION
System.ArgumentException	`drawingLayer` was not registered in IRegisteredLayersOrder

TYPE	CONDITION
System.ArgumentOutOfRangeException	`registeredIndex` is present in IRegisteredLayersOrder, but count of registrations does not corresponds to the registered index.

Interface IDrawingLayer

Represents drawing of one layer as logical set of components.

Namespace: GTTG.Core.Drawing.Layers

Assembly: cs.temp.dll.dll

Syntax

```
public interface IDrawingLayer
```

Methods

Draw(DrawingCanvas)

Draws the layer onto `drawingCanvas`.

Declaration

```
void Draw(DrawingCanvas drawingCanvas)
```

Parameters

TYPE	NAME	DESCRIPTION
DrawingCanvas	drawingCanvas	

Interface IRegisteredLayersOrder

Defines order of registered IDrawingLayer drawing layers.

Namespace: GTTG.Core.Drawing.Layers

Assembly: cs.temp.dll.dll

Syntax

```
public interface IRegisteredLayersOrder
```

Properties

DrawingLayerTypeList

Ordered set of drawing layers types - IDrawingLayer. First index - 0 is visually on bottom.

Declaration

```
ImmutableList<Type> DrawingLayerTypeList { get; }
```

Property Value

TYPE	DESCRIPTION
ImmutableList<Type>	

Class ViewDrawingLayer

IDrawingLayer representing layer of ViewDrawingCanvas.

Inheritance

System.Object

DrawingLayer

ViewDrawingLayer

Implements

IDrawingLayer

Inherited Members

DrawingLayer.Draw(DrawingCanvas)

DrawingLayer.OnDraw(DrawingCanvas)

System.Object.ToString()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

Namespace: GTTG.Core.Drawing.Layers

Assembly: cs.temp.dll.dll

Syntax

```
public abstract class ViewDrawingLayer : DrawingLayer, IDrawingLayer
```

Implements

IDrawingLayer

Namespace GTTG.Core.Extensions

Classes

SkPathExtensions

Extensions method for .

SkRectExtensions

Extension methods for .

VisualsEnumerableExtensions

Extensions method for hit test of collections.

Class SkPathExtensions

Extensions method for .

Inheritance

System.Object
SkPathExtensions

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: GTTG.Core.Extensions

Assembly: cs.temp.dll.dll

Syntax

```
public static class SkPathExtensions
```

Methods

CalculateDistanceFromPath(SKPath, SKPoint)

Calculates distance of from by finding closest point on the path relative to the point.

Declaration

```
public static float CalculateDistanceFromPath(this SKPath path, SKPoint point)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPath	path	
SKPoint	point	

Returns

TYPE	DESCRIPTION
System.Single	

Class SkRectExtensions

Extension methods for .

Inheritance

System.Object
SkRectExtensions

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: GTTG.Core.Extensions

Assembly: cs.temp.dll.dll

Syntax

```
public static class SkRectExtensions
```

Methods

ContainsWithDelta(SKRect, SKRect, Single)

Determines if rectangle contains another rectangle.

Declaration

```
public static bool ContainsWithDelta(this SKRect containing, SKRect inside, float comparisonDelta = 0.001F)
```

Parameters

TYPE	NAME	DESCRIPTION
SKRect	containing	Rectangle that should contain `inside` rectangle.
SKRect	inside	Rectangle that should be in `containing` rectangle.
System.Single	comparisonDelta	Tolerance of floating point subtraction.

Returns

TYPE	DESCRIPTION
System.Boolean	

Class VisualsEnumerableExtensions

Extensions method for hit test of collections.

Inheritance

System.Object
VisualsEnumerableExtensions

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: GTTG.Core.Extensions

Assembly: cs.temp.dll.dll

Syntax

```
public static class VisualsEnumerableExtensions
```

Methods

HitTest<THitTestTarget>(IEnumerable<THitTestTarget>, SKPoint)

Hit-tests collection of elements.

Declaration

```
public static IEnumerable<THitTestTarget> HitTest<THitTestTarget>(this IEnumerable<THitTestTarget> targets,
SKPoint contentPoint)
    where THitTestTarget : IVisual
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<THitTestTarget>	targets	Collection of elements to be hit-tested.
SKPoint	contentPoint	Point in ContentDrawingCanvas tested for hit.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<THitTestTarget>	Elements which were hit. Elements are returned by order in `targets`.

Type Parameters

NAME	DESCRIPTION

NAME	DESCRIPTION
THitTestTarget	Type of elements in collection to be processed.

OrderByLayers<TVisualType>(IEnumerable<TVisualType>, IEnumerable<IDrawingLayer>, IDrawingLayer)

Orders collection of `TVisualType` elements by their IDrawingLayer.

Declaration

```
public static IEnumerable<TVisualType> OrderByLayers<TVisualType>(this IEnumerable<TVisualType> visuals,
IEnumerable<IDrawingLayer> drawingLayersOrder, IDrawingLayer sourceDrawingLayer)
    where TVisualType : IVisual
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<TVisualType>	visuals	Collection of `TVisualType` elements to be sorted, each belonging to some IDrawingLayer.
System.Collections.Generic.IEnumerable<IDrawingLayer>	drawingLayersOrder	IDrawingLayers are ordered by order in this enumeration. Returns first elements from first layer in enumeration.
IDrawingLayer	sourceDrawingLayer	If IDrawingLayer of visual element is DefaultDrawingLayer, treat this visual element as one belonging to `sourceDrawingLayer` layer.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<TVisualType>	Sorted collection of visual elements by their drawing layers. First element returned is in layer which comes first in `drawingLayersOrder` enumeration. If drawing layers are same, order is defined by order in `visuals`.

Type Parameters

NAME	DESCRIPTION
TVisualType	SegmentType of visual elements in collection to be processed.

OrderByLayers<TVisualType, TSourceDrawingLayer>(IEnumerable<TVisualType>, IEnumerable<IDrawingLayer>)

Orders collection of `TVisualType` elements by their IDrawingLayer.

Declaration

```
public static IEnumerable<TVisualType> OrderByLayers<TVisualType, TSourceDrawingLayer>(this
IEnumerable<TVisualType> visuals, IEnumerable<IDrawingLayer> drawingLayersOrder)
    where TVisualType : IVisual where TSourceDrawingLayer : IDrawingLayer
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<TVisualType>	visuals	Collection of `TVisualType` elements to be sorted, each belonging to some IDrawingLayer.
System.Collections.Generic.IEnumerable<IDrawingLayer>	drawingLayersOrder	Layers are ordered by order in this enumeration. Returns first elements from first layer in enumeration.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<TVisualType>	Sorted collection of visual elements by their drawing layers. First element returned is in layer which comes first in `drawingLayersOrder` enumeration. If drawing layers are same, order is defined by order in `visuals`.

Type Parameters

NAME	DESCRIPTION
TVisualType	SegmentType of visual elements in collection to be processed.
TSourceDrawingLayer	If IDrawingLayer of visual element is DefaultDrawingLayer, treat this visual element as one belonging to instance of layer of this type.

Namespace GTTG.Core.HitTest

Classes

HitTestManager

Provides tools for hit testing of visual elements.

Enums

HitTestFilterBehavior

Specifies the return behavior of a hit test in a hit test filter callback method.

HitTestResultBehavior

Determines whether to continue the enumeration of any remaining visual objects during a hit test.

ResultTraversalOrder

Defines result order of tree traversal.

Delegates

HitTestFilterCallback

Represents the callback method that specifies parts of the visual tree to omit from hit test processing

HitTestResultCallback

Represents a callback that is used to customize hit testing. GTTG invokes the HitTestResultCallback to report hit test intersections to the user.

Enum HitTestFilterBehavior

Specifies the return behavior of a hit test in a hit test filter callback method.

Namespace: GTTG.Core.HitTest

Assembly: cs.temp.dll.dll

Syntax

```
public enum HitTestFilterBehavior
```

Fields

NAME	DESCRIPTION
Continue	Hit test against the current and its descendants.
ContinueSkipChildren	Hit test against the current , but not its descendants.
ContinueSkipSelf	Do not hit test against the current , but hit test against its descendants.
ContinueSkipSelfAndChildren	Do not hit test against the current or its descendants.
Stop	Stop hit testing at the current .

Delegate HitTestFilterCallback

Represents the callback method that specifies parts of the visual tree to omit from hit test processing

Namespace: GTTG.Core.HitTest

Assembly: cs.temp.dll.dll

Syntax

```
public delegate HitTestFilterBehavior HitTestFilterCallback(IVisual target, SKPoint contentPoint);
```

Parameters

TYPE	NAME	DESCRIPTION
IVisual	target	The visual to hit test.
SKPoint	contentPoint	Point in ContentDrawingCanvas against which hit test.

Returns

TYPE	DESCRIPTION
HitTestFilterBehavior	A HitTestResultBehavior that represents the action resulting from the hit test.

Class HitTestManager

Provides tools for hit testing of visual elements.

Inheritance

System.Object
HitTestManager

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: GTTG.Core.HitTest

Assembly: cs.temp.dll.dll

Syntax

```
public class HitTestManager
```

Constructors

HitTestManager(DrawingManager)

Creates hit test manager with provided drawing layers order.

Declaration

```
public HitTestManager(DrawingManager drawingManager)
```

Parameters

TYPE	NAME	DESCRIPTION
DrawingManager	drawingManager	Drawing manager with layers for DrawingLayers to enumerate.

Properties

DrawingLayers

Provides order of drawing layers in visual order with undermost one being enumerated as first.

Declaration

```
public IEnumerable<IDrawingLayer> DrawingLayers { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<IDrawingLayer>	

Methods

HitTest(HitTestFilterCallback, HitTestResultCallback, SKPoint)

Hit tests tree of DrawingManager of ContentDrawingLayer layers content with provided callbacks for tree pruning and positive hit

Hit tests tree of DrawingManager or ContentDrawingLayer layers content with provided callbacks for tree pruning and positive hit tests. Order of tree traversal is defined as follows: Apply `filterCallback` on root element. If tree should be processed, hit tests the root element. If the root element is positive on the hit-test, first process the root element and then children enumerated from (considering applied filters). Else return from the tree traversal.

Declaration

```
public void HitTest(HitTestFilterCallback filterCallback, HitTestResultCallback resultCallback, SKPoint
contentPoint)
```

Parameters

TYPE	NAME	DESCRIPTION
HitTestFilterCallback	filterCallback	Called when element is found traversing the tree.
HitTestResultCallback	resultCallback	Called when hit test on element in tree is positive.
SKPoint	contentPoint	Point in ContentDrawingCanvas tested for hit.

HitTest(IVisual, HitTestFilterCallback, HitTestResultCallback, SKPoint)

Hit tests tree of elements, with provided callbacks for tree pruning and positive hit tests. Order of tree traversal is defined as follows: Apply `filterCallback` on root element. If tree should be processed, hit tests the root element. If the root element is positive on the hit-test, first process the root element and then children enumerated from (considering applied filters). Else return from the tree traversal.

Declaration

```
public static void HitTest(IVisual hitTestRoot, HitTestFilterCallback filterCallback, HitTestResultCallback
resultCallback, SKPoint contentPoint)
```

Parameters

TYPE	NAME	DESCRIPTION
IVisual	hitTestRoot	Root of element tree also hit tested by default.
HitTestFilterCallback	filterCallback	Called when element is found traversing the tree.
HitTestResultCallback	resultCallback	Called when hit test on element in tree is positive.
SKPoint	contentPoint	Point in ContentDrawingCanvas tested for hit.

HitTest(IVisual, SKPoint, ResultTraversalOrder)

Hit tests a element tree with provided element as root to find elements with positive hit test by enumerating it's children with .

Declaration

```
public static IVisual HitTest(IVisual target, SKPoint contentPoint, ResultTraversalOrder resultTraversalOrder
= ResultTraversalOrder.Last)
```

Parameters

TYPE	NAME	DESCRIPTION
IVisual	target	Target as root of hit tested tree.
SKPoint	contentPoint	Point in ContentDrawingCanvas tested for hit.
ResultTraversalOrder	resultTraversalOrder	Determines if element should be first or last from order defined in default tree traversal.

Returns

TYPE	DESCRIPTION
IVisual	First hit tested element defined by `resultTraversalOrder`. By default, last.

Enum HitTestResultBehavior

Determines whether to continue the enumeration of any remaining visual objects during a hit test.

Namespace: GTTG.Core.HitTest

Assembly: cs.temp.dll.dll

Syntax

```
public enum HitTestResultBehavior
```

Fields

NAME	DESCRIPTION
Continue	Continue hit testing against the next visual in the visual tree hierarchy.
Stop	Stop any further hit testing and return from the callback.

Delegate HitTestResultCallback

Represents a callback that is used to customize hit testing. GTTG invokes the HitTestResultCallback to report hit test intersections to the user.

Namespace: GTTG.Core.HitTest

Assembly: cs.temp.dll.dll

Syntax

```
public delegate HitTestResultBehavior HitTestResultCallback(IVisual target);
```

Parameters

TYPE	NAME	DESCRIPTION
IVisual	target	The object that is returned from a hit test.

Returns

TYPE	DESCRIPTION
HitTestResultBehavior	A HitTestResultBehavior that represents the action resulting from the hit test.

Enum ResultTraversalOrder

Defines result order of tree traversal.

Namespace: GTTG.Core.HitTest

Assembly: cs.temp.dll.dll

Syntax

```
public enum ResultTraversalOrder
```

Fields

NAME	DESCRIPTION
First	First element found in traversal is returned first.
Last	Last element found in traversal is returned first.

Namespace GTTG.Core.Strategies

Classes

StrategyException

Initializes a new instance of the StrategyException class with a specified error message.

Class StrategyException

Initializes a new instance of the StrategyException class with a specified error message.

Inheritance

System.Object

StrategyException

Namespace: GTTG.Core.Strategies

Assembly: cs.temp.dll.dll

Syntax

```
public class StrategyException : ArgumentException
```

Constructors

StrategyException(String)

Declaration

```
public StrategyException(string message)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	

Namespace GTTG.Core.Strategies.Implementations

Classes

BasicStrategyManager<TPlacementType, TElement, TSegmentType>

Represents storage of registered elements under placement type which are mapped to particular segment type.

MeasurableStrategyManager<TPlacementType, TElement, TSegmentType>

Represents strategy manager for segments with measureable height. Manager registers measure methods to segments as it's resources.

MeasureableSegment

Represents measurable segment used for strategies. Measures content from added event handlers.

Segment

Represents segment which can be positioned in ContentDrawingCanvas or particular .

SegmentRegistry<TSegmentType, TSegment>

SegmentRegistry<TSegmentType, TSegment>.SegmentRegistrationBuilder

Implementation of registration builder for adding instances after specified type.

StrategyManager<TPlacementType, TElement, TSegmentType, TSegment>

Represents strategy manager that maps content to particular segment instance.

Delegates

HeightMeasureHelper

Measures height of element connected with this delegate's implementation.

Class BasicStrategyManager<TPlacementType, TElement, TSegmentType>

Represents storage of registered elements under placement type which are mapped to particular segment type.

Inheritance

System.Object

BasicStrategyManager<TPlacementType, TElement, TSegmentType>

StrategyManager<TPlacementType, TElement, TSegmentType, TSegment>

Implements

System.Collections.Generic.IReadOnlyDictionary<TPlacementType, TElement>

System.Collections.Generic.IReadOnlyCollection<System.Collections.Generic.KeyValuePair<TPlacementType, TElement>>

System.Collections.Generic.IEnumerable<System.Collections.Generic.KeyValuePair<TPlacementType, TElement>>

System.Collections.IEnumerable

Namespace: GTTG.Core.Strategies.Implementations

Assembly: cs.temp.dll.dll

Syntax

```
public class BasicStrategyManager<TPlacementType, TElement, TSegmentType> : Visual,
IReadOnlyDictionary<TPlacementType, TElement>, IReadOnlyCollection<KeyValuePair<TPlacementType, TElement>>,
IEnumerable<KeyValuePair<TPlacementType, TElement>>, IEnumerable where TElement : IVisual
```

Type Parameters

NAME	DESCRIPTION
TPlacementType	Type of placement type.
TElement	Type of element of .
TSegmentType	Type of segment type.

Constructors

BasicStrategyManager(ITypeConverter<TPlacementType, TSegmentType>)

Creates empty basic strategy manager.

Declaration

```
public BasicStrategyManager(ITypeConverter<TPlacementType, TSegmentType> typeConverter)
```

Parameters

TYPE	NAME	DESCRIPTION
ITypeConverter<TPlacementType, TSegmentType>	typeConverter	Converter for placement type to segment type.

Fields

Elements

Collection of registered `TElement` under `TPlacementType`.

Declaration

```
protected readonly Dictionary<TPlacementType, TElement> Elements
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<TPlacementType, TElement>	

SegmentTypes

Collection of registered `TElement` under `TSegmentType`.

Declaration

```
protected readonly Dictionary<TPlacementType, TSegmentType> SegmentTypes
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<TPlacementType, TSegmentType>	

TypeConverter

Converter between `TPlacementType` and `TSegmentType` values.

Declaration

```
protected readonly ITypeConverter<TPlacementType, TSegmentType> TypeConverter
```

Field Value

TYPE	DESCRIPTION
ITypeConverter<TPlacementType, TSegmentType>	

Properties

Count

Declaration

```
public int Count { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Item[TPlacementType]

Declaration

```
public TElement this[TPlacementType key] { get; }
```

Parameters

TYPE	NAME	DESCRIPTION
TPlacementType	key	

Property Value

TYPE	DESCRIPTION
TElement	

Keys

Declaration

```
public IEnumerable<TPlacementType> Keys { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<TPlacementType>	

ManagedSegmentTypes

Maps registered `TPlacementType` types to `TSegmentType`.

Declaration

```
public IReadOnlyDictionary<TPlacementType, TSegmentType> ManagedSegmentTypes { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.IReadOnlyDictionary<TPlacementType, TSegmentType>	

Values

Declaration

```
public IEnumerable<TElement> Values { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<TElement>	

Methods

Add(TPlacementType, TElement)

Adds element to manager under key.

Declaration

```
public virtual void Add(TPlacementType key, TElement value)
```

Parameters

TYPE	NAME	DESCRIPTION
TPlacementType	key	Placement type of added element.
TElement	value	Element to be added.

Exceptions

TYPE	CONDITION
StrategyException	Element under the key already registered.

Add(KeyValuePair<TPlacementType, TElement>)

Adds element to manager under key wrapped as System.Collections.Generic.KeyValuePair<TKey, TValue>.

Declaration

```
public virtual void Add(KeyValuePair<TPlacementType, TElement> item)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.KeyValuePair<TPlacementType, TElement>	item	

Exceptions

TYPE	CONDITION
StrategyException	Element under the key already registered.

Clear()

Clears all added elements and cleans held resources and registrations.

Declaration

```
public virtual void Clear()
```

Contains(KeyValuePair<TPlacementType, TElement>)

Determines if elem

Declaration

```
public bool Contains(KeyValuePair<TPlacementType, TElement> item)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.KeyValuePair<TPlacementType, TElement>	item	

Returns

TYPE	DESCRIPTION
System.Boolean	

ContainsKey(TPlacementType)

Declaration

```
public bool ContainsKey(TPlacementType key)
```

Parameters

TYPE	NAME	DESCRIPTION
TPlacementType	key	

Returns

TYPE	DESCRIPTION
System.Boolean	

Draw(DrawingCanvas)

Declaration

```
public override void Draw(DrawingCanvas drawingCanvas)
```

Parameters

TYPE	NAME	DESCRIPTION
DrawingCanvas	drawingCanvas	

GetEnumerator()

Declaration

```
public IEnumerator<KeyValuePair<TPlacementType, TElement>> GetEnumerator()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerator<System.Collections.Generic.KeyValuePair<TPlacementType, TElement>>	

HasHit(SKPoint)

Declaration

```
public override bool HasHit(SKPoint contentPoint)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	contentPoint	

Returns

TYPE	DESCRIPTION
System.Boolean	

ProvideVisuals()

Declaration

```
public override IEnumerable<IVisual> ProvideVisuals()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<IVisual>	

Remove(TPlacementType)

Removes item from manager and removes all it's registrations and resources.

Declaration

```
public virtual bool Remove(TPlacementType key)
```

Parameters

TYPE	NAME	DESCRIPTION
TPlacementType	key	Item to remove under key.

Returns

TYPE	DESCRIPTION
System.Boolean	True if removed; otherwise false.

Remove(KeyValuePair<TPlacementType, TElement>)

Removes item from manager and removes all it's registrations and resources.

Declaration

```
public virtual bool Remove(KeyValuePair<TPlacementType, TElement> item)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.KeyValuePair<TPlacementType, TElement>	item	Item to remove.

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
System.Boolean	True if removed; otherwise false.

TryGetValue(TPlacementType, out TElement)

Declaration

```
public bool TryGetValue(TPlacementType key, out TElement value)
```

Parameters

TYPE	NAME	DESCRIPTION
TPlacementType	key	
TElement	value	

Returns

TYPE	DESCRIPTION
System.Boolean	

Implements

System.Collections.Generic.IReadOnlyDictionary<TKey, TValue>
System.Collections.Generic.IReadOnlyCollection<T>
System.Collections.Generic.IEnumerable<T>
System.Collections.IEnumerable

Delegate HeightMeasureHelper

Measures height of element connected with this delegate's implementation.

Namespace: GTTG.Core.Strategies.Implementations

Assembly: cs.temp.dll.dll

Syntax

```
public delegate float HeightMeasureHelper();
```

Returns

TYPE	DESCRIPTION
System.Single	Height of particular element.

Class MeasurableStrategyManager<TPlacementType, TElement, TSegmentType>

Represents strategy manager for segments with measureable height. Manager registers measure methods to segments as it's resources.

Inheritance

System.Object
BasicStrategyManager<TPlacementType, TElement, TSegmentType>
StrategyManager<TPlacementType, TElement, TSegmentType, MeasureableSegment>
MeasurableStrategyManager<TPlacementType, TElement, TSegmentType>

Implements

System.Collections.Generic.IReadOnlyDictionary<TPlacementType, TElement>
System.Collections.Generic.IReadOnlyCollection<System.Collections.Generic.KeyValuePair<TPlacementType, TElement>>
System.Collections.Generic.IEnumerable<System.Collections.Generic.KeyValuePair<TPlacementType, TElement>>
System.Collections.IEnumerable

Inherited Members

StrategyManager<TPlacementType, TElement, TSegmentType, MeasureableSegment>.ManagedSegments
StrategyManager<TPlacementType, TElement, TSegmentType, MeasureableSegment>.SegmentRegistry
StrategyManager<TPlacementType, TElement, TSegmentType, MeasureableSegment>.Segments
StrategyManager<TPlacementType, TElement, TSegmentType, MeasureableSegment>.Add(KeyValuePair<TPlacementType, TElement>)
StrategyManager<TPlacementType, TElement, TSegmentType, MeasureableSegment>.Add(TPlacementType, TElement)
StrategyManager<TPlacementType, TElement, TSegmentType, MeasureableSegment>.Clear()
StrategyManager<TPlacementType, TElement, TSegmentType, MeasureableSegment>.Remove(KeyValuePair<TPlacementType, TElement>)
StrategyManager<TPlacementType, TElement, TSegmentType, MeasureableSegment>.Remove(TPlacementType)
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Item[TPlacementType]
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Keys
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Values
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Count
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.ManagedSegmentTypes
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Elements
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.SegmentTypes
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.TypeConverter
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.HasHit(SKPoint)
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.ProvideVisuals()
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Draw(DrawingCanvas)
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.GetEnumerator()
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.ContainsKey(TPlacementType)
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.TryGetValue(TPlacementType, TElement)
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Clear()
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Contains(KeyValuePair<TPlacementType, TElement>)
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Add(TPlacementType, TElement)
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Add(KeyValuePair<TPlacementType, TElement>)
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Remove(KeyValuePair<TPlacementType, TElement>)
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Remove(TPlacementType)

Namespace: GTTG.Core.Strategies.Implementations

Assembly: cs.temp.dll.dll

Syntax

```
public class MeasurableStrategyManager<TPlacementType, TElement, TSegmentType> :
StrategyManager<TPlacementType, TElement, TSegmentType, MeasureableSegment>,
IReadOnlyDictionary<TPlacementType, TElement>, IReadOnlyCollection<KeyValuePair<TPlacementType, TElement>>,
IEnumerable<KeyValuePair<TPlacementType, TElement>>, IEnumerable where TElement : IVisual
```

Type Parameters

NAME	DESCRIPTION
TPlacementType	
TElement	
TSegmentType	

Constructors

MeasurableStrategyManager(ISegmentRegistry<TSegmentType, MeasureableSegment>, ITypeConverter<TPlacementType, TSegmentType>, IElementMeasureProvider<TPlacementType, TElement, TSegmentType>)

Creates strategy manager with measureable content.

Declaration

```
public MeasurableStrategyManager(ISegmentRegistry<TSegmentType, MeasureableSegment> segmentRegistry,
ITypeConverter<TPlacementType, TSegmentType> typeConverter, IElementMeasureProvider<TPlacementType, TElement,
TSegmentType> measureProvider)
```

Parameters

TYPE	NAME	DESCRIPTION
ISegmentRegistry<TSegmentType, MeasureableSegment>	segmentRegistry	Segments registry to receive segments of `TSegmentType`.
ITypeConverter<TPlacementType, TSegmentType>	typeConverter	Instance of converter between specified types.
IElementMeasureProvider<TPlacementType, TElement, TSegmentType>	measureProvider	Interface to which are passed added elements to be measured.

Fields

MeasureHelperRegistrations

Instances of HeightMeasureHelper under registration of `TPlacementType`.

Declaration

```
protected readonly Dictionary<TPlacementType, HeightMeasureHelper> MeasureHelperRegistrations
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<TPlacementType, HeightMeasureHelper>	

MeasureProvider

Implementation of measure method for `TElement` values.

Declaration

```
protected readonly IElementMeasureProvider<TPlacementType, TElement, TSegmentType> MeasureProvider
```

Field Value

TYPE	DESCRIPTION
IElementMeasureProvider<TPlacementType, TElement, TSegmentType>	

Methods

Add(TPlacementType, TElement)

Declaration

```
public override void Add(TPlacementType key, TElement value)
```

Parameters

TYPE	NAME	DESCRIPTION
TPlacementType	key	
TElement	value	

Overrides

GTTG.Core.Strategies.Implementations.StrategyManager<TPlacementType, TElement, TSegmentType, GTTG.Core.Strategies.Implementations.MeasureableSegment>.Add(TPlacementType, TElement)

Add(KeyValuePair<TPlacementType, TElement>)

Declaration

```
public override void Add(KeyValuePair<TPlacementType, TElement> item)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.KeyValuePair<TPlacementType, TElement>	item	

Overrides

GTTG.Core.Strategies.Implementations.StrategyManager<TPlacementType, TElement, TSegmentType, GTTG.Core.Strategies.Implementations.MeasureableSegment>.Add(System.Collections.Generic.KeyValuePair<TPlacementType, TElement>)

Clear()

Declaration

```
public override void Clear()
```

Overrides

GTTG.Core.Strategies.Implementations.StrategyManager<TPlacementType, TElement, TSegmentType, GTTG.Core.Strategies.Implementations.MeasureableSegment>.Clear()

Remove(TPlacementType)

Declaration

```
public override bool Remove(TPlacementType key)
```

Parameters

TYPE	NAME	DESCRIPTION
TPlacementType	key	

Returns

TYPE	DESCRIPTION
System.Boolean	

Overrides

GTTG.Core.Strategies.Implementations.StrategyManager<TPlacementType, TElement, TSegmentType, GTTG.Core.Strategies.Implementations.MeasureableSegment>.Remove(TPlacementType)

Remove(KeyValuePair<TPlacementType, TElement>)

Declaration

```
public override bool Remove(KeyValuePair<TPlacementType, TElement> item)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.KeyValuePair<TPlacementType, TElement>	item	

Returns

TYPE	DESCRIPTION
System.Boolean	

Overrides

GTTG.Core.Strategies.Implementations.StrategyManager<TPlacementType, TElement, TSegmentType, GTTG.Core.Strategies.Implementations.MeasureableSegment>.Remove(System.Collections.Generic.KeyValuePair<TPlacementType, TElement>)

Implements

System.Collections.Generic.IReadOnlyDictionary<TKey, TValue>
System.Collections.Generic.IReadOnlyCollection<T>
System.Collections.Generic.IEnumerable<T>
System.Collections.IEnumerable

Class MeasureableSegment

Represents measurable segment used for strategies. Measures content from added event handlers.

Inheritance

System.Object

Segment

MeasureableSegment

Inherited Members

Segment.ContentUpperBoundPosition

Segment.ContentLowerBoundPosition

Segment.LocalUpperBound

Segment.LocalLowerBound

Segment.SegmentLocalMiddle

Segment.SegmentContentMiddle

Segment.SegmentLocalHeight

Segment.SegmentContentHeight

Segment.SetBounds(ViewElement, Single, Single)

Segment.SetBounds(Single, Single)

Namespace: GTTG.Core.Strategies.Implementations

Assembly: cs.temp.dll.dll

Syntax

```
public class MeasureableSegment : Segment
```

Properties

DesiredHeight

Height of segment's content set from MeasureHeight() calls.

Declaration

```
public float DesiredHeight { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

MeasureHeight()

Measures height of segment by selecting maximum from HeightMeasureHelpers invocations. Measured value is set to DesiredHeight.

Declaration

```
public void MeasureHeight()
```

Events

HeightMeasureHelpers

Add HeightMeasureHelper method to measure content of particular element to be placed in this segment; invoked on MeasureHeight().

Declaration

```
public event HeightMeasureHelper HeightMeasureHelpers
```

Event Type

TYPE	DESCRIPTION
HeightMeasureHelper	

Class Segment

Represents segment which can be positioned in ContentDrawingCanvas or particular .

Inheritance

System.Object
Segment
MeasureableSegment

Namespace: GTTG.Core.Strategies.Implementations

Assembly: cs.temp.dll.dll

Syntax

```
public class Segment : ISegment
```

Properties

ContentLowerBoundPosition

Declaration

```
public float ContentLowerBoundPosition { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

ContentUpperBoundPosition

Declaration

```
public float ContentUpperBoundPosition { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

LocalLowerBound

If segment is placed from SetBounds(ViewElement, Single, Single), this value is in coordinates of . Otherwise equal to ContentLowerBoundPosition.

Declaration

```
public float LocalLowerBound { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

LocalUpperBound

If segment is placed from SetBounds(ViewElement, Single, Single), this value is in coordinates of . Otherwise equal to ContentUpperBoundPosition.

Declaration

```
public float LocalUpperBound { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

SegmentContentHeight

Declaration

```
public float SegmentContentHeight { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

SegmentContentMiddle

Position of the middle of segment in ContentDrawingCanvas.

Declaration

```
public float SegmentContentMiddle { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

SegmentLocalHeight

Distance between LocalLowerBound and LocalUpperBound.

Declaration

```
public float SegmentLocalHeight { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

SegmentLocalMiddle

If segment is placed from SetBounds(ViewElement, Single, Single), value is in middle of LocalUpperBound and LocalLowerBound; otherwise same as SegmentContentMiddle.

Declaration

```
public float SegmentLocalMiddle { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

SetBounds(Single, Single)

Places segment in ContentDrawingCanvas.

Declaration

```
public void SetBounds(float globalUpperBound, float globalLowerBound)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	globalUpperBound	Value of ContentUpperBoundPosition.
System.Single	globalLowerBound	Value of ContentLowerBoundPosition.

Exceptions

TYPE	CONDITION
StrategyException	If bounds does not form valid segment; lower is above upper.

SetBounds(ViewElement, Single, Single)

Places segment in ContentDrawingCanvas from perspective of particular .

Declaration

```
public void SetBounds(ViewElement viewElement, float localUpperBound, float localLowerBound)
```

Parameters

TYPE	NAME	DESCRIPTION
ViewElement	viewElement	from which position of segment is determined.
System.Single	localUpperBound	Position in `viewElement` area translated to ContentUpperBoundPosition.
System.Single	localLowerBound	Position in `viewElement` area translated to ContentLowerBoundPosition.

Class SegmentRegistry<TSegmentType, TSegment>

Inheritance

System.Object

SegmentRegistry<TSegmentType, TSegment>

Namespace: GTTG.Core.Strategies.Implementations

Assembly: cs.temp.dll.dll

Syntax

```
public class SegmentRegistry<TSegmentType, TSegment> : ISegmentRegistry<TSegmentType, TSegment> where TSegment
: Segment
```

Type Parameters

NAME	DESCRIPTION
TSegmentType	
TSegment	

Constructors

SegmentRegistry()

Creates empty SegmentRegistry<TSegmentType, TSegment> with no segment registrations.

Declaration

```
public SegmentRegistry()
```

Fields

RegisteredSegments

Internal structure to book keep the registered segments.

Declaration

```
protected readonly Dictionary<TSegmentType, TSegment> RegisteredSegments
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<TSegmentType, TSegment>	

Methods

Register(TSegment)

Declaration

```
public virtual ISegmentRegistrationBuilder<TSegmentType> Register(TSegment segment)
```

Parameters

TYPE	NAME	DESCRIPTION
TSegment	segment	

Returns

TYPE	DESCRIPTION
ISegmentRegistrationBuilder<TSegmentType>	

Resolve(TSegmentType)

Declaration

```
public virtual TSegment Resolve(TSegmentType segmentType)
```

Parameters

TYPE	NAME	DESCRIPTION
TSegmentType	segmentType	

Returns

TYPE	DESCRIPTION
TSegment	

Class SegmentRegistry<TSegmentType, TSegment>.SegmentRegistrationBuilder

Implementation of registration builder for adding instances after specified type.

Inheritance

System.Object

SegmentRegistry<TSegmentType, TSegment>.SegmentRegistrationBuilder

Namespace: GTTG.Core.Strategies.Implementations

Assembly: cs.temp.dll.dll

Syntax

```
protected class SegmentRegistrationBuilder : ISegmentRegistrationBuilder<TSegmentType>
```

Constructors

SegmentRegistrationBuilder(Dictionary<TSegmentType, TSegment>, TSegment)

Creates builder with registration instances.

Declaration

```
public SegmentRegistrationBuilder(Dictionary<TSegmentType, TSegment> registeredSegments, TSegment newSegment)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<TSegmentType, TSegment>	registeredSegments	Dictionary of segments where instance under new type is added
TSegment	newSegment	Instance of segment whose type must be specified.

Fields

NewSegment

New segment added be added to RegisteredSegments.

Declaration

```
protected readonly TSegment NewSegment
```

Field Value

TYPE	DESCRIPTION
TSegment	

RegisteredSegments

Internal structure to book keep the registered segments.

Declaration

```
protected readonly Dictionary<TSegmentType, TSegment> RegisteredSegments
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<TSegmentType, TSegment>	

Class StrategyManager<TPlacementType, TElement, TSegmentType, TSegment>

Represents strategy manager that maps content to particular segment instance.

Inheritance

System.Object
BasicStrategyManager<TPlacementType, TElement, TSegmentType>
StrategyManager<TPlacementType, TElement, TSegmentType, TSegment>
MeasurableStrategyManager<TPlacementType, TElement, TSegmentType>

Implements

System.Collections.Generic.IReadOnlyDictionary<TPlacementType, TElement>
System.Collections.Generic.IReadOnlyCollection<System.Collections.Generic.KeyValuePair<TPlacementType, TElement>>
System.Collections.Generic.IEnumerable<System.Collections.Generic.KeyValuePair<TPlacementType, TElement>>
System.Collections.IEnumerable

Inherited Members

BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Item[TPlacementType]
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Keys
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Values
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Count
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.ManagedSegmentTypes
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Elements
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.SegmentTypes
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.TypeConverter
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.HasHit(SKPoint)
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.ProvideVisuals()
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Draw(DrawingCanvas)
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.GetEnumerator()
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.ContainsKey(TPlacementType)
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.TryGetValue(TPlacementType, TElement)
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Clear()
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Contains(KeyValuePair<TPlacementType, TElement>)
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Add(TPlacementType, TElement)
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Add(KeyValuePair<TPlacementType, TElement>)
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Remove(KeyValuePair<TPlacementType, TElement>)
BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Remove(TPlacementType)

Namespace: GTTG.Core.Strategies.Implementations

Assembly: cs.temp.dll.dll

Syntax

```
public class StrategyManager<TPlacementType, TElement, TSegmentType, TSegment> :
BasicStrategyManager<TPlacementType, TElement, TSegmentType>, IReadOnlyDictionary<TPlacementType, TElement>,
IReadOnlyCollection<KeyValuePair<TPlacementType, TElement>>, IEnumerable<KeyValuePair<TPlacementType,
TElement>>, IEnumerable where TElement : IVisual where TSegment : ISegment
```

Type Parameters

NAME	DESCRIPTION
TPlacementType	Type of placement type.

NAME	DESCRIPTION
TElement	Type of added elements.
TSegmentType	Type of segment type.
TSegment	Type of segment instances.

Constructors

StrategyManager(ISegmentRegistry<TSegmentType, TSegment>, ITypeConverter<TPlacementType, TSegmentType>)

Creates empty StrategyManager<TPlacementType, TElement, TSegmentType, TSegment>.

Declaration

```
public StrategyManager(ISegmentRegistry<TSegmentType, TSegment> segmentRegistry,
ITypeConverter<TPlacementType, TSegmentType> typeConverter)
```

Parameters

TYPE	NAME	DESCRIPTION
ISegmentRegistry<TSegmentType, TSegment>	segmentRegistry	Segments registry to receive segments of `TSegmentType`.
ITypeConverter<TPlacementType, TSegmentType>	typeConverter	Instance of converter between specified types.

Fields

SegmentRegistry

A registry from where segments are received.

Declaration

```
protected readonly ISegmentRegistry<TSegmentType, TSegment> SegmentRegistry
```

Field Value

TYPE	DESCRIPTION
ISegmentRegistry<TSegmentType, TSegment>	

Segments

Collection of information about `TElement` registration under `TPlacementType`.

Declaration

```
protected readonly Dictionary<TPlacementType, ISegment> Segments
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<TPlacementType, ISegment>	

Properties

ManagedSegments

Maps registered `TPlacementType` types to it's .

Declaration

```
public IReadOnlyDictionary<TPlacementType, ISegment> ManagedSegments { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.IReadOnlyDictionary<TPlacementType, ISegment>	

Methods

Add(TPlacementType, TElement)

Declaration

```
public override void Add(TPlacementType key, TElement value)
```

Parameters

TYPE	NAME	DESCRIPTION
TPlacementType	key	
TElement	value	

Overrides

GTTG.Core.Strategies.Implementations.BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Add(TPlacementType, TElement)

Add(KeyValuePair<TPlacementType, TElement>)

Declaration

```
public override void Add(KeyValuePair<TPlacementType, TElement> item)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.KeyValuePair<TPlacementType, TElement>	item	

Overrides

GTTG.Core.Strategies.Implementations.BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Add(System.Collections.Generic.KeyValuePair<TPlacementType, TElement>)

Clear()

Declaration

```
public override void Clear()
```

Overrides

GTTG.Core.Strategies.Implementations.BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Clear()

Remove(TPlacementType)

Declaration

```
public override bool Remove(TPlacementType key)
```

Parameters

TYPE	NAME	DESCRIPTION
TPlacementType	key	

Returns

TYPE	DESCRIPTION
System.Boolean	

Overrides

GTTG.Core.Strategies.Implementations.BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Remove(TPlacementType)

Remove(KeyValuePair<TPlacementType, TElement>)

Declaration

```
public override bool Remove(KeyValuePair<TPlacementType, TElement> item)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.KeyValuePair<TPlacementType, TElement>	item	

Returns

TYPE	DESCRIPTION
System.Boolean	

Overrides

GTTG.Core.Strategies.Implementations.BasicStrategyManager<TPlacementType, TElement, TSegmentType>.Remove(System.Collections.Generic.KeyValuePair<TPlacementType, TElement>)

Implements

System.Collections.Generic.IReadOnlyDictionary<TKey, TValue>
System.Collections.Generic.IReadOnlyCollection<T>
System.Collections.Generic.IEnumerable<T>
System.Collections.IEnumerable

Namespace GTTG.Core.Strategies.Interfaces

Interfaces

IElementMeasureProvider<TPlacementType, TElement, TSegmentType>

Contract for height measure of particular element with provided information about registered type and segment.

ISegment

Represents segment as horizontal stripe bounded by upper and lower lines in ContentDrawingCanvas. As ContentDrawingCanvas Y-axis is increasing downwards, ContentLowerBoundPosition is always greater than ContentUpperBoundPosition.

ISegmentRegistrationBuilder<T>

Helper structure for fluent syntax registration in .

ISegmentRegistry<TSegmentType, TSegment>

Storage of ISegment of type `TSegmentType`.

IStrategyDocker

Implements strategy by docking provided elements into segments.

ITypeConverter<TPlacementType, TSegmentType>

Converts type to different representation.

Interface IElementMeasureProvider<TPlacementType, TElement, TSegmentType>

Contract for height measure of particular element with provided information about registered type and segment.

Namespace: GTTG.Core.Strategies.Interfaces

Assembly: cs.temp.dll.dll

Syntax

```
public interface IElementMeasureProvider<in TPlacementType, in TElement, in TSegmentType>
```

Type Parameters

NAME	DESCRIPTION
TPlacementType	
TElement	
TSegmentType	

Methods

MeasureHeight(TPlacementType, TElement, TSegmentType, ISegment)

Measures height of `TElement` by provided parameters.

Declaration

```
float MeasureHeight(TPlacementType placementType, TElement element, TSegmentType segmentType, ISegment
segment)
```

Parameters

TYPE	NAME	DESCRIPTION
TPlacementType	placementType	
TElement	element	
TSegmentType	segmentType	
ISegment	segment	

Returns

TYPE	DESCRIPTION
System.Single	

Interface ISegment

Represents segment as horizontal stripe bounded by upper and lower lines in ContentDrawingCanvas. As ContentDrawingCanvas Y-axis is increasing downwards, ContentLowerBoundPosition is always greater than ContentUpperBoundPosition.

Namespace: GTTG.Core.Strategies.Interfaces

Assembly: cs.temp.dll.dll

Syntax

```
public interface ISegment
```

Properties

ContentLowerBoundPosition

Position of segment's horizontal line bounding it's content from below.

Declaration

```
float ContentLowerBoundPosition { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

ContentUpperBoundPosition

Position of segment's horizontal line bounding it's content from above.

Declaration

```
float ContentUpperBoundPosition { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

SegmentContentHeight

Distance between ContentUpperBoundPosition and ContentLowerBoundPosition.

Declaration

```
float SegmentContentHeight { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Interface ISegmentRegistrationBuilder<T>

Helper structure for fluent syntax registration in .

Namespace: GTTG.Core.Strategies.Interfaces

Assembly: cs.temp.dll.dll

Syntax

```
public interface ISegmentRegistrationBuilder<in T>
```

Type Parameters

NAME	DESCRIPTION
T	

Methods

As(T)

Register added segment under type instance of `T`.

Declaration

```
ISegmentRegistrationBuilder<T> As(T segmentType)
```

Parameters

TYPE	NAME	DESCRIPTION
T	segmentType	

Returns

TYPE	DESCRIPTION
ISegmentRegistrationBuilder<T>	

Interface ISegmentRegistry<TSegmentType, TSegment>

Storage of ISegment of type `TSegmentType`.

Namespace: GTTG.Core.Strategies.Interfaces

Assembly: cs.temp.dll.dll

Syntax

```
public interface ISegmentRegistry<in TSegmentType, TSegment>
    where TSegment : ISegment
```

Type Parameters

NAME	DESCRIPTION
TSegmentType	Type of segments for distinguishing particular instances.
TSegment	Class type of segment.

Methods

Register(TSegment)

Register segment and determine other values in subsequent calls by returned structure.

Declaration

```
ISegmentRegistrationBuilder<TSegmentType> Register(TSegment segment)
```

Parameters

TYPE	NAME	DESCRIPTION
TSegment	segment	Instances of `TSegment`

Returns

TYPE	DESCRIPTION
ISegmentRegistrationBuilder<TSegmentType>	Registration fluent syntax structure to determine particular `TSegmentType` and other values.

Resolve(TSegmentType)

Get instance of ISegment previously registered by Register(TSegment).

Declaration

```
TSegment Resolve(TSegmentType segmentType)
```

Parameters

TYPE	NAME	DESCRIPTION
TSegmentType	segmentType	Type of registered instance.

Returns

TYPE	DESCRIPTION
TSegment	Registered instance under `segmentType`.

Interface IStrategyDocker

Implements strategy by docking provided elements into segments.

Namespace: GTTG.Core.Strategies.Interfaces

Assembly: cs.temp.dll.dll

Syntax

```
public interface IStrategyDocker
```

Methods

Dock()

Docks elements added to strategy managed by this docker.

Declaration

```
void Dock()
```

Interface ITypeConverter<TPlacementType, TSegmentType>

Converts type to different representation.

Namespace: GTTG.Core.Strategies.Interfaces

Assembly: cs.temp.dll.dll

Syntax

```
public interface ITypeConverter<in TPlacementType, out TSegmentType>
```

Type Parameters

NAME	DESCRIPTION
TPlacementType	
TSegmentType	

Methods

Convert(TPlacementType)

Converts of `TPlacementType` to `TSegmentType`.

Declaration

```
TSegmentType Convert(TPlacementType placementType)
```

Parameters

TYPE	NAME	DESCRIPTION
TPlacementType	placementType	

Returns

TYPE	DESCRIPTION
TSegmentType	

Namespace GTTG.Core.Time

Classes

DateTimeContext

DateTimeInterval intervals as representation of ContentDrawingCanvas and ViewDrawingCanvas scope.

DayHoursInterval

Represents day interval as sequence of hours from which are determined available hour windows.

Structs

DateTimeInterval

Represents interval of values from Start to End.

Class DateTimeContext

DateTimeInterval intervals as representation of ContentDrawingCanvas and ViewDrawingCanvas scope.

Inheritance

System.Object
DateTimeContext

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: GTTG.Core.Time

Assembly: cs.temp.dll.dll

Syntax

```
public class DateTimeContext
```

Constructors

DateTimeContext(DateTimeInterval, DateTimeInterval)

Constructs DateTimeContext.

Declaration

```
public DateTimeContext(DateTimeInterval contentDateTimeInterval, DateTimeInterval viewDateTimeInterval)
```

Parameters

TYPE	NAME	DESCRIPTION
DateTimeInterval	contentDateTimeInterval	Value representing ContentDateTimeInterval.
DateTimeInterval	viewDateTimeInterval	Value representing ViewDateTimeInterval.

Properties

ContentDateTimeInterval

Time scope of data available to display in ContentDrawingCanvas. Contains ContentDateTimeInterval.

Declaration

```
public DateTimeInterval ContentDateTimeInterval { get; }
```

Property Value

TYPE	DESCRIPTION
DateTimeInterval	

ViewDateTimeInterval

Time scope of view, the content being displayed in ViewDrawingCanvas.

Declaration

```
public DateTimeInterval ViewDateTimeInterval { get; }
```

Property Value

TYPE	DESCRIPTION
DateTimeInterval	

Struct DateTimeInterval

Represents interval of values from Start to End.

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetType()

Namespace: GTTG.Core.Time

Assembly: cs.temp.dll.dll

Syntax

```
public struct DateTimeInterval
```

Constructors

DateTimeInterval(DateTime, DateTime)

Creates DateTimeInterval of values.

Declaration

```
public DateTimeInterval(DateTime start, DateTime end)
```

Parameters

TYPE	NAME	DESCRIPTION
DateTime	start	
DateTime	end	

Exceptions

TYPE	CONDITION
System.ArgumentOutOfRangeException	`end` is earlier or same as `start`.

Properties

End

End of DateTimeInterval. This value is also in interval of this instance.

Declaration

```
public DateTime End { get; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Start

Start of DateTimeInterval. This value is also in the interval of this instance.

Declaration

```
public DateTime Start { get; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

TimeSpan

Time elapsed between Start and End.

Declaration

```
public TimeSpan TimeSpan { get; }
```

Property Value

TYPE	DESCRIPTION
TimeSpan	

Methods

Contains(DateTime)

Determines whether an provided `dateTime` is in the DateTimeInterval.

Declaration

```
public bool Contains(DateTime dateTime)
```

Parameters

TYPE	NAME	DESCRIPTION
DateTime	dateTime	The provided .

Returns

TYPE	DESCRIPTION
System.Boolean	true if `dateTime` is found in the DateTimeInterval; otherwise, false.

Contains(DateTimeInterval)

Determines whether an provided `dateTimeInterval` is in the interval of this instance.

Declaration

```
public bool Contains(DateTimeInterval dateTimeInterval)
```

Parameters

TYPE	NAME	DESCRIPTION
DateTimeInterval	dateTimeInterval	The provided DateTimeInterval.

Returns

TYPE	DESCRIPTION
System.Boolean	true if `dateTimeInterval` is in interval of this instance; otherwise, false.

GetDateTimesByPeriod(DateTime, TimeSpan)

Provides values in interval of this instance defined by start and repeating period.

Declaration

```
public IEnumerable<DateTime> GetDateTimesByPeriod(DateTime start, TimeSpan period)
```

Parameters

TYPE	NAME	DESCRIPTION
DateTime	start	Start included in returned values from which enumeration starts.
TimeSpan	period	TimeSpan period separating returned values.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<DateTime>	values in DateTimeInterval separated by `period`. Contains `start`.

Exceptions

TYPE	CONDITION
System.ArgumentOutOfRangeException	`start` is not in .

GetMultiple(DateTime)

Converts to multiple of this instance interval.

Declaration

```
public float GetMultiple(DateTime dateTime)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
DateTime	dateTime	to convert.

Returns

TYPE	DESCRIPTION
System.Single	Values [0.00f - 1.00f] for for which Contains(DateTimeInterval) returns true. Otherwise for outside the interval returns values greater or lower than mentioned return value interval.

IntersectsWith(DateTimeInterval)

Determines whether an provided `dateTimeInterval` contains at least one same value from DateTimeInterval.

Declaration

```
public bool IntersectsWith(DateTimeInterval dateTimeInterval)
```

Parameters

TYPE	NAME	DESCRIPTION
DateTimeInterval	dateTimeInterval	The provided DateTimeInterval.

Returns

TYPE	DESCRIPTION
System.Boolean	true if `dateTimeInterval` intersects with the DateTimeInterval; otherwise, false.

ToString()

Converts this interval to string representation using default .

Declaration

```
public override string ToString()
```

Returns

TYPE	DESCRIPTION
System.String	String representation of DateTimeInterval.

Overrides

System.ValueType.ToString()

ToString(String)

Converts this interval to string using format pattern.

Declaration

```
public string ToString(string format)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	format	A standard or custom date and time format string to be used as format on both Start and End.

Returns

TYPE	DESCRIPTION
System.String	String representation of this interval.

Exceptions

TYPE	CONDITION
System.FormatException	The length of `format` is 1, and it is not one of the format specifier characters defined for System.Globalization.DateTimeFormatInfo.-or- `format` does not contain a valid custom format pattern.
System.ArgumentOutOfRangeException	The date and time is outside the range of dates supported by the calendar used by the current culture.

Class DayHoursInterval

Represents day interval as sequence of hours from which are determined available hour windows.

Inheritance

System.Object
DayHoursInterval

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: GTTG.Core.Time

Assembly: cs.temp.dll.dll

Syntax

```
public class DayHoursInterval
```

Constructors

DayHoursInterval(Int32, Int32, ICollection<Int32>)

Creates hour windows for provided hour interval.

Declaration

```
public DayHoursInterval(int startHour, int endHour, ICollection<int> windowHours = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	startHour	Value assigned to StartHour.
System.Int32	endHour	Value assigned to EndHour.
System.Collections.Generic.ICollection<System.Int32>	windowHours	If provided, only those hour windows are allowed to be in WindowHours.

Fields

AllIntervalHours

All possible hour windows of day.

Declaration

```
public static readonly ImmutableArray<int> AllIntervalHours
```

Field Value

TYPE	DESCRIPTION
ImmutableArray<System.Int32>	

WholeDay

Representation of whole day.

Declaration

```
public static DayHoursInterval WholeDay
```

Field Value

TYPE	DESCRIPTION
DayHoursInterval	

Properties

EndHour

End hour of day hours sequence.

Declaration

```
public int EndHour { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

MaxWindowHour

Longest available hour window.

Declaration

```
public int MaxWindowHour { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

StartHour

Start hour of day hours sequence.

Declaration

```
public int StartHour { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

WindowHours

Available hour windows.

Declaration

```
public IReadOnlyList<int> WindowHours { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.IReadOnlyList<System.Int32>	

Methods

Equals(DayHoursInterval)

Compares intervals by start and end values.

Declaration

```
public bool Equals(DayHoursInterval other)
```

Parameters

TYPE	NAME	DESCRIPTION
DayHoursInterval	other	

Returns

TYPE	DESCRIPTION
System.Boolean	

Equals(Object)

Declaration

```
public override bool Equals(object obj)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	obj	

Returns

TYPE	DESCRIPTION
System.Boolean	

Overrides

System.Object.Equals(System.Object)

GetHashCode()

Declaration

```
public override int GetHashCode()
```

Returns

TYPE	DESCRIPTION
System.Int32	

Overrides

System.Object.GetHashCode()

ToDateTimeInterval(DateTime)

Converts this day interval to DateTimeInterval.

Declaration

```
public DateTimeInterval ToDateTimeInterval(DateTime date = null)
```

Parameters

TYPE	NAME	DESCRIPTION
DateTime	date	Date value with date representing this interval.

Returns

TYPE	DESCRIPTION
DateTimeInterval	DateTimeInterval with `date` date and day hours interval of this instance.

Namespace GTTG.Core.Utils

Classes

LayoutConstants

Constant values used for layout cycle.

PlacementUtils

Math functions used for implementation.

Class LayoutConstants

Constant values used for layout cycle.

Inheritance

System.Object
LayoutConstants

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: GTTG.Core.Utils

Assembly: cs.temp.dll.dll

Syntax

```
public static class LayoutConstants
```

Fields

HorizontalLineVector

Vector of horizontal line.

Declaration

```
public static readonly SKPoint HorizontalLineVector
```

Field Value

TYPE	DESCRIPTION
SKPoint	

InfiniteSize

with Single.PositiveInfinity values.

Declaration

```
public static readonly SKSize InfiniteSize
```

Field Value

TYPE	DESCRIPTION
SKSize	

Class PlacementUtils

Math functions used for implementation.

Inheritance

System.Object
PlacementUtils

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: GTTG.Core.Utils

Assembly: cs.temp.dll.dll

Syntax

```
public static class PlacementUtils
```

Methods

ComputeAcuteRadAngle(SKPoint, SKPoint)

Computes acute angle of two vectors in radians.

Declaration

```
public static double ComputeAcuteRadAngle(SKPoint u, SKPoint v)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	u	First vector.
SKPoint	v	Second vector.

Returns

TYPE	DESCRIPTION
System.Double	Acute angle in radians.

ComputeCosine(SKPoint, SKPoint)

Computes cosine of two vectors.

Declaration

```
public static double ComputeCosine(SKPoint u, SKPoint v)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	u	First vector.
SKPoint	v	Second vector.

Returns

TYPE	DESCRIPTION
System.Double	Cosine value in radians.

ComputeDiagonal(SKSize)

Computes diagonal of rectangle.

Declaration

```
public static float ComputeDiagonal(SKSize rect)
```

Parameters

TYPE	NAME	DESCRIPTION
SKSize	rect	The rectangle whose diagonal is determined.

Returns

TYPE	DESCRIPTION
System.Single	Diagonal of the `rect`.

ComputeHorizontalLineIntersection(SKPoint, SKPoint, Single)

Computes position of intersection of line and horizontal line.

Declaration

```
public static SKPoint ComputeHorizontalLineIntersection(SKPoint vector, SKPoint vectorPoint, float
horizontalLineY)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	vector	Vector which sets direction of the intersecting line.
SKPoint	vectorPoint	Point found on `vector`, forming the intersecting line.

TYPE	NAME	DESCRIPTION
System.Single	horizontalLineY	Vertical position of the horizontal line.

Returns

TYPE	DESCRIPTION
SKPoint	Point on the horizontal line.

ComputeHypotenuseLength(SKPoint, SKPoint, Single)

Computes length of hypotenuse in perpendicular triangle formed by acute angle of two vectors.

Declaration

```
public static float ComputeHypotenuseLength(SKPoint u, SKPoint v, float opposedToAngleLegLength)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	u	First vector forming the acute angle.
SKPoint	v	Second vector forming the acute angle.
System.Single	opposedToAngleLegLength	Length of the leg opposed to the formed angle.

Returns

TYPE	DESCRIPTION
System.Single	Length of hypotenuse in formed triangle.

ComputeLegLength(SKPoint, SKPoint, Single)

Computes length of leg adjacent to acute angle of two vectors in perpendicular triangle.

Declaration

```
public static float ComputeLegLength(SKPoint u, SKPoint v, float opposedToAngleLegLength)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	u	First vector forming the acute angle.

TYPE	NAME	DESCRIPTION
SKPoint	v	Second vector forming the acute angle.
System.Single	opposedToAngleLegLength	Length of leg opposed to the formed angle.

Returns

TYPE	DESCRIPTION
System.Single	Length of adjacent leg to the formed angle.

ComputesVectorLength(SKPoint)

Computes length of vector.

Declaration

```
public static float ComputesVectorLength(SKPoint vector)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	vector	The vector whose length is determined.

Returns

TYPE	DESCRIPTION
System.Single	Length of `vector`.

MoveInLine(SKPoint, SKPoint, Single)

Moves point in line by specified length.

Declaration

```
public static SKPoint MoveInLine(SKPoint vector, SKPoint vectorPoint, float length)
```

Parameters

TYPE	NAME	DESCRIPTION
SKPoint	vector	Vector describing the line; in which the translation of `vectorPoint` happens.
SKPoint	vectorPoint	The point in line which is translated.

TYPE	NAME	DESCRIPTION
System.Single	length	The length of translation.

Returns

TYPE	DESCRIPTION
SKPoint	Translated `vectorPoint` by specified length.