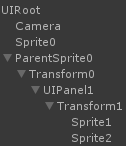
重要属性：

public List<UIWidget> widgets = new List<UIWidget>();

public List<UIDrawCall> drawCalls = new List<UIDrawCall>();

[HideInInspector][SerializeField] int mDepth = 0;

UIPanel mParentPanel = null;（实际上在基类UIRect还有一个parent（UIRect））



Debug.Log("---------------------------------begin");

Debug.Log("name = " + gameObject.name);

if (parent) {

Debug.Log("parent.name = " + parent.name);

} else

{

Debug.Log("parent is null");

}

if (parentPanel) {

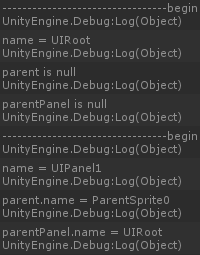
Debug.Log("parentPanel.name = " + parentPanel.name);

} else

{

Debug.Log("parentPanel is null");

}



更新函数：

void LateUpdate ()

{

#if UNITY\_EDITOR

if (mUpdateFrame != Time.frameCount || !Application.isPlaying)

#else

if (mUpdateFrame != Time.frameCount)

#endif

{

mUpdateFrame = Time.frameCount;

// Update each panel in order

for (int i = 0, imax = list.Count; i < imax; ++i)

list[i].**UpdateSelf**();

int rq = 3000;

// Update all draw calls, making them draw in the right order

for (int i = 0, imax = list.Count; i < imax; ++i)

{

UIPanel p = list[i];

if (p.renderQueue == RenderQueue.Automatic)

{

p.startingRenderQueue = rq;

p.UpdateDrawCalls();

rq += p.drawCalls.Count;

}

else if (p.renderQueue == RenderQueue.StartAt)

{

p.UpdateDrawCalls();

if (p.drawCalls.Count != 0)

rq = Mathf.Max(rq, p.startingRenderQueue + p.drawCalls.Count);

}

else // Explicit

{

p.UpdateDrawCalls();

if (p.drawCalls.Count != 0)

rq = Mathf.Max(rq, p.startingRenderQueue + 1);

}

}

}

}

UpdateSelf函数：

void UpdateSelf ()

{

mHasMoved = cachedTransform.hasChanged;

UpdateTransformMatrix();

UpdateLayers();

**UpdateWidgets()**;

if (mRebuild)

{

mRebuild = false;

FillAllDrawCalls();

}

else

{

for (int i = 0; i < drawCalls.Count; )

{

UIDrawCall dc = drawCalls[i];

if (dc.isDirty && !FillDrawCall(dc))

{

UIDrawCall.Destroy(dc);

drawCalls.RemoveAt(i);

continue;

}

++i;

}

}

if (mUpdateScroll)

{

mUpdateScroll = false;

UIScrollView sv = GetComponent<UIScrollView>();

if (sv != null) sv.UpdateScrollbars();

}

if (mHasMoved)

{

mHasMoved = false;

mTrans.hasChanged = false;

}

}

更新所有的Widgets函数UpdateWidgets:

void UpdateWidgets()

{

bool changed = false;

bool forceVisible = false;

bool clipped = hasCumulativeClipping;

if (!cullWhileDragging)

{

for (int i = 0; i < UIScrollView.list.size; ++i)

{

UIScrollView sv = UIScrollView.list[i];

if (sv.panel == this && sv.isDragging) forceVisible = true;

}

}

if (mForced != forceVisible)

{

mForced = forceVisible;

mResized = true;

}

// Update all widgets

int frame = Time.frameCount;

for (int i = 0, imax = widgets.Count; i < imax; ++i)

{

UIWidget w = widgets[i];

// If the widget is visible, update it

if (w.panel == this && w.enabled)

{

#if UNITY\_EDITOR

// When an object is dragged from Project view to Scene view, its Z is...

// odd, to say the least. Force it if possible.

if (!Application.isPlaying)

{

Transform t = w.cachedTransform;

if (t.hideFlags != HideFlags.HideInHierarchy)

{

t = (t.parent != null && t.parent.hideFlags == HideFlags.HideInHierarchy) ?

t.parent : null;

}

if (t != null)

{

for (; ; )

{

if (t.parent == null) break;

if (t.parent.hideFlags == HideFlags.HideInHierarchy) t = t.parent;

else break;

}

if (t != null)

{

Vector3 pos = t.localPosition;

pos.x = Mathf.Round(pos.x);

pos.y = Mathf.Round(pos.y);

pos.z = 0f;

if (Vector3.SqrMagnitude(t.localPosition - pos) > 0.0001f)

t.localPosition = pos;

}

}

}

#endif

// First update the widget's transform

if (w.**UpdateTransform**(frame) || mResized || (mHasMoved && !alwaysOnScreen))

{

// Only proceed to checking the widget's visibility if it actually moved

bool vis = forceVisible || (w.CalculateCumulativeAlpha(frame) > 0.001f);

w.UpdateVisibility(vis, forceVisible || alwaysOnScreen || ((clipped || w.hideIfOffScreen) ? IsVisible(w) : true));

}

// Update the widget's geometry if necessary

if (w.**UpdateGeometry**(frame))

{

changed = true;

//Debug.Log("Geometry changed: " + w.name + " " + frame, w);

if (!mRebuild)

{

// Find an existing draw call, if possible

if (w.drawCall != null) w.drawCall.isDirty = true;

else FindDrawCall(w);

}

}

}

}

// Inform the changed event listeners

if (changed && onGeometryUpdated != null) onGeometryUpdated();

mResized = false;

}