Smart Slicer 2D

Updated: 2018/10/07
Current Asset Version: 1.1.3
Roadmap: https://trello.com/b/MxDEYWIU/smart-slicer-2d

- **Slicer2D Components**
  - Components
  - Component Reference

- **Slicer2D Main Scripting API**

  This API is used to interact with existing game objects that got Slicer2D component attached
  - Functions
  - Function Reference
  - Slice2D - Slice Result

- **Slicer2D Events Handling**
  - AddEvent
  - AddResultEvent

- **Slicer2D Low Level Scripting API**

  This API is used to create your own custom slicer controllers and unique behaviour
  - Functions
  - Function Reference

- **Math2D Scripting API** *<not fully documented yet>*
  - Functions

- **Polygon2D Scripting API** *<not fully documented yet>*
  - Functions

- **Vector2D Scripting API** *<not fully documented yet>*
  - Functions

- **Code Samples** *<not documented yet>*

Forum Discussion

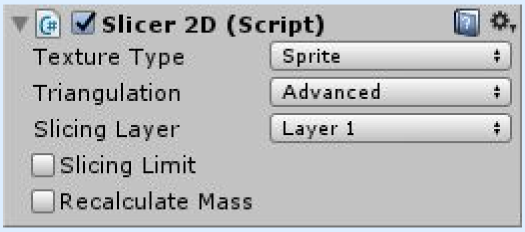
Contact: simonas@kuzmickas.lt

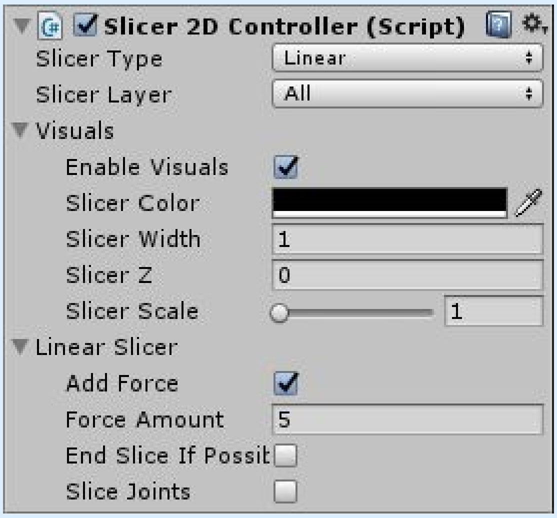
Slicer2D Components

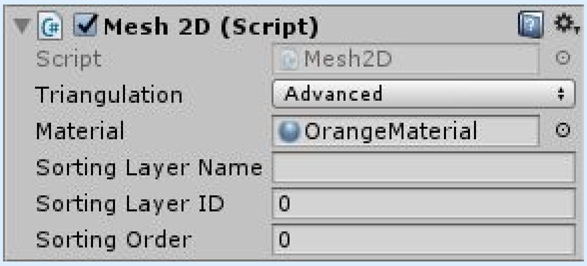
Components

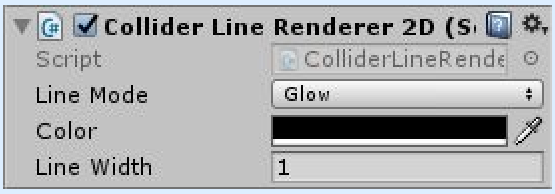
Slicer2D	*Physics component for slicing sprites & meshes (Require Collider2D)*
Mesh2D	*Creates a mesh from attached Collider2D*
ColliderLineRenderer2D	*Draw lines for the attached Collider2D*
JointRenderer2D	*Draws lines for the attached joint components*
Slicer2DAnchors	*Attach Slice2D component to the specific colliders*
Slicer2DController	*Mainly for mouse input which is used to slice or affect slice-able objects*
Slicer2DLinearController	*Simplified version of Slicer2DController for linear mode slicing only*
Slicer2DComplexController	*Simplified version of Slicer2DController for complex mode slicing only*

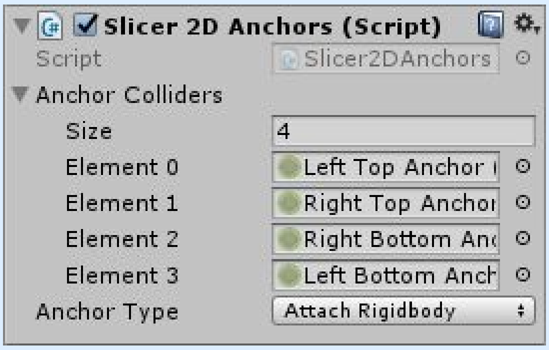
Component Reference

Slicer2D	
appearance	 ☑ Slicer 2D (Script) Texture Type Sprite Triangulation Advanced Slicing Layer Layer 1 ☐ Slicing Limit ☐ Recalculate Mass
description	*Physics component for Slicing sprites & meshes*
information	***Parameters:*** ● *Texture Type* ○ *Sprite* ○ *Mesh* ○ *None* ● *Slicing Layer* ○ *8 layers* ● *Slicing Limit* ○ *Optionable (after 1-10 slices object won't be slice-able)*

Slicer2DController	
appearance	 ☑ Slicer 2D Controller (Script) Slicer Type Linear Slicer Layer All Visuals Enable Visuals ☑ Slicer Color ■ Slicer Width 1 Slicer Z 0 Slicer Scale 1 Linear Slicer Add Force ☑ Force Amount 5 End Slice If Possit ☐ Slice Joints ☐
description	*Mainly for mouse input which is used to slice or affect slice-able objects*
information	***Parameters:*** ● *Slicer Type* ○ *Linear* ○ *Linear Cut* ○ *Complex* ○ *Complex Tracked* ○ *Point* ○ *Polygon* ○ *Explode* ○ *Creator* ● *Slicer Layer:* ○ *All* ○ *Selected (checkbox list for 8 layers)*

Mesh2D	
appearance	 Mesh 2D (Script) Script Mesh2D Triangulation Advanced Material OrangeMaterial Sorting Layer Name Sorting Layer ID 0 Sorting Order 0
description	*Creates a mesh from already attached Collider2D*
information	***Parameters:*** ● *Triangulation* ○ *Advanced* ○ *Legacy* ● *Material*

ColliderLineRenderer2D	
appearance	 Collider Line Renderer 2D (S Script ColliderLineRende Line Mode Glow Color Line Width 1
description	*Draws lines for attached Collider2D*
information	***Parameters:*** ● *Line Mode* ○ *Smooth* ○ *Glow* ○ *Default* ● *Color* ● *Line Width*

Slicer2DAnchors	
appearance	 Slicer 2D Anchors (Script) Script Slicer2DAnchors Anchor Colliders Size 4 Element 0 Left Top Anchor (Element 1 Right Top Anchor Element 2 Right Bottom Anc Element 3 Left Bottom Anch Anchor Type Attach Rigidbody
description	*Attach Slicer2D component to a specific anchor*
information	***Parameters:*** ● *Anchor Colliders* ● *Anchor Type*

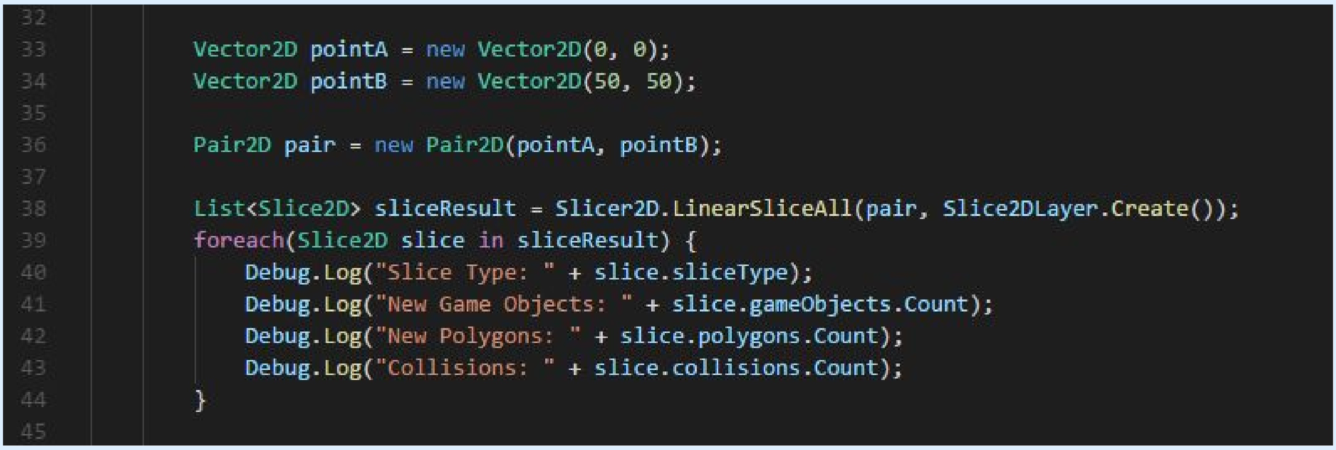
Slice2D - Slice Result

This object is used to return results and other additional information after slicing bodies

Slice2D	
object fields	● Slice2D ○ polygonList List <Polygon2D> ○ gameObjects List <GameObject> ○ collisionList List<Vector2D> ○ slice List<Vector2D> ○ slices List<List<Vector2D>>
description	Contain information about the slice. Used in scripting Slicer2D API and events handling system.

Sample 1

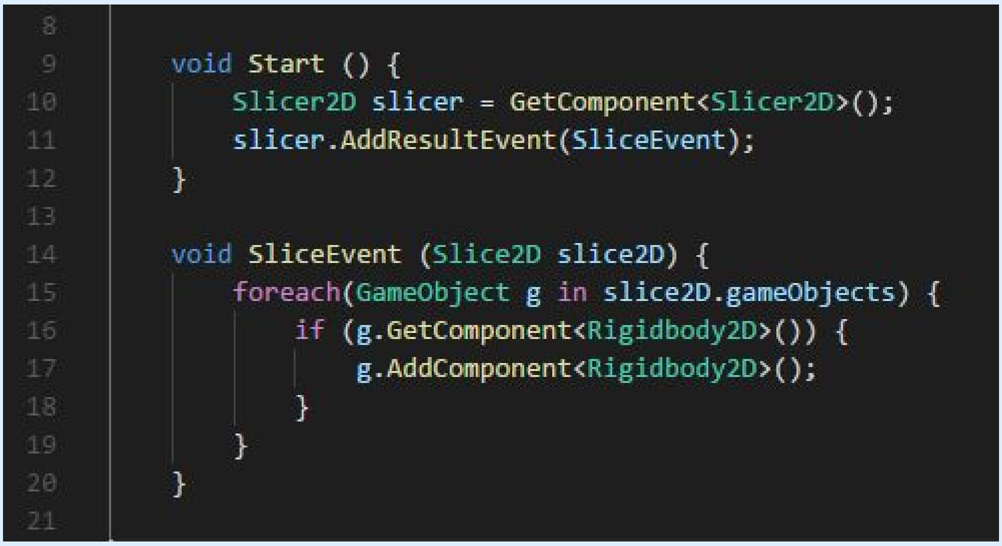
Prints all information about the slice into console



```
Vector2D pointA = new Vector2D(0, 0);
Vector2D pointB = new Vector2D(50, 50);

Pair2D pair = new Pair2D(pointA, pointB);

List<Slice2D> sliceResult = Slicer2D.LinearSliceAll(pair, Slice2DLayer.Create());
foreach(Slice2D slice in sliceResult) {
    Debug.Log("Slice Type: " + slice.sliceType);
    Debug.Log("New Game Objects: " + slice.gameObjects.Count);
    Debug.Log("New Polygons: " + slice.polygons.Count);
    Debug.Log("Collisions: " + slice.collisions.Count);
}
```

Sample 2

Uses event system to add Rigidbody2D component after slicing the GameObject



```
void Start () {
    Slicer2D slicer = GetComponent<Slicer2D>();
    slicer.AddResultEvent(SliceEvent);
}

void SliceEvent (Slice2D slice2D) {
    foreach(GameObject g in slice2D.gameObjects) {
        if (g.GetComponent<Rigidbody2D>()) {
            g.AddComponent<Rigidbody2D>();
        }
    }
}
```

Slicer2D Main Scripting API

This API is used to interact with existing game objects that are having Slicer2D component attached

Functions

LinearSliceAll	*Slice all objects linearly using a line by 2 given points*
LinearCutSliceAll	*Slice all objects linearly using a line by 2 given points & width*
PointSliceAll	*Slice all objects linearly by a point and given rotation*
ComplexSliceAll	*Slice all objects non-linearly by given points list*
PolygonSliceAll	*Slice a piece of area from a polygon - using polygon as a parameter*
ExplodingSliceAll	*Explode a polygon by a given point*
ExplodeSliceAll	*Explode all objects*

Function Reference

Static List <Slice2D> LinearSliceAll (**Pair2D** slice)	
returns	List of *Slice2D* objects
description	*Slice all GameObjects in the scene with Slicer2D component attached. Slices linearly using a line by 2 given points*
information	*<not documented yet>*

Static List <Slice2D> LinearCutSliceAll (**LinearCut** slice)	
returns	List of *Slice2D* objects
description	*Slice all GameObjects in the scene with Slicer2D component attached. Slices linearly using a line by 2 given points & width*
information	*<not documented yet>*

Static List <Slice2D> PointSliceAll (**Vector2D** position, **Vector2D** rotation)	
returns	List of *Slice2D* objects
description	*Slice all GameObjects in the scene with Slicer2D component attached. Slice linearly by a point and given rotation*
information	*<not documented yet>*

Static List <Slice2D> ComplexSliceAll (**List<Vector2D>** slice)	
returns	List of *Slice2D* objects
description	*Slice all GameObjects in the scene with Slicer2D component attached. Slice an object non-linearly by given points list*
information	*<not documented yet>*

Static List <Slice2D> PolygonSliceAll (**Polygon2D** polygon)	
returns	List of *Slice2D* objects
description	*Slice all GameObjects in the scene with Slicer2D component attached. Slice a piece of area from a polygon - using polygon as a parameter*
information	*<not documented yet>*

Static List <Slice2D> ExplodingSliceAll (**Vector2D** point)	
returns	List of *Slice2D* objects
description	*Explode all GameObjects in the scene with Slicer2D component attached. Explode a polygon by a given point*
information	*<not documented yet>*

Static List <Slice2D> ExplodeSliceAll ( )	
returns	List of *Slice2D* objects
description	*Explode all GameObjects in the scene with Slicer2D component attached. Explode all objects*
information	*<not documented yet>*

Slicer2D Events Handling

AddEvent

description

This event is applied before new sliced objects are created.

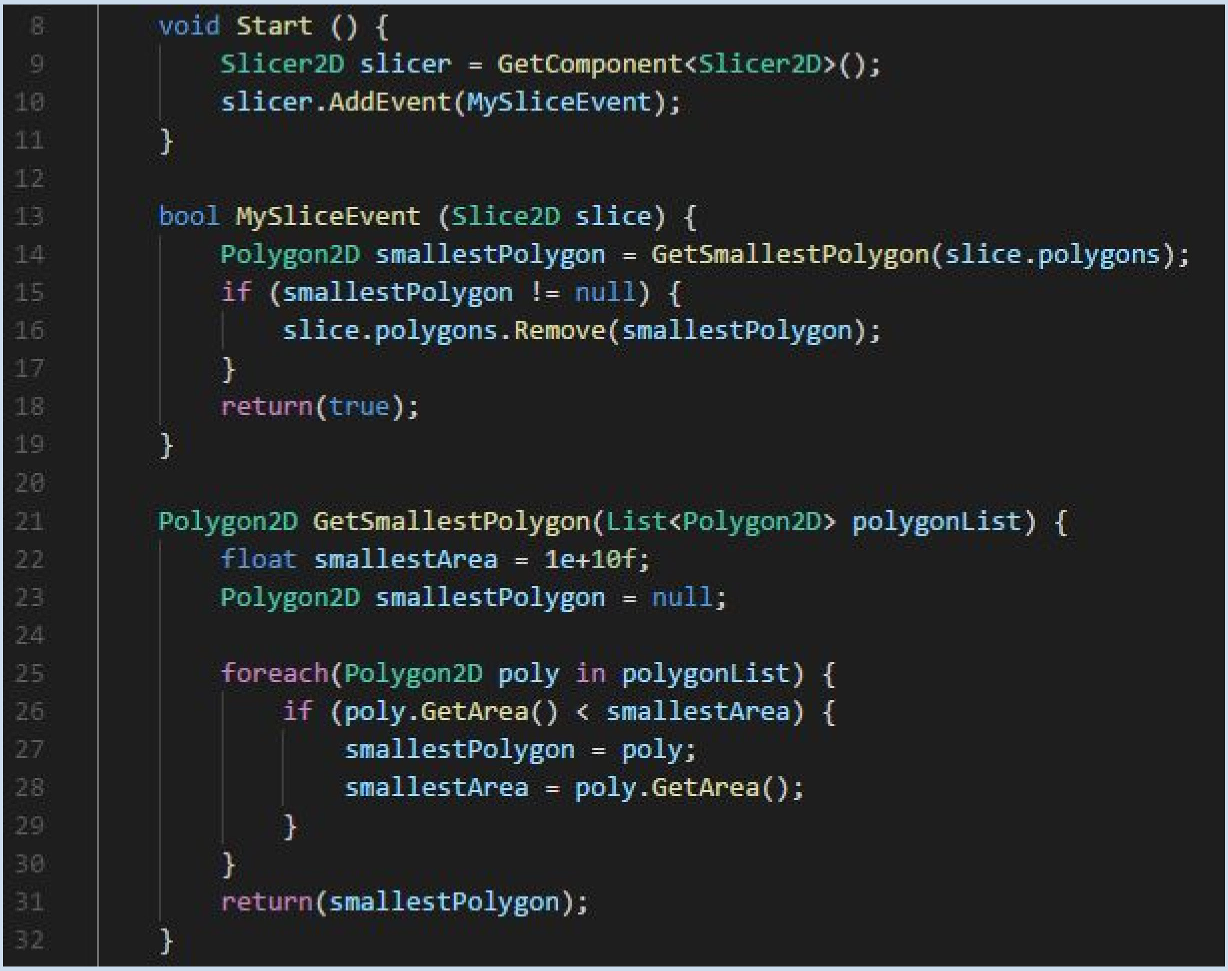
If this event method

- Returns “false” - the slice will be canceled
- Returns “true” - the slice will be performed

Slice2D object contains fields:

- SliceType
- Polygons
- Collisions
- Slice
- Slices

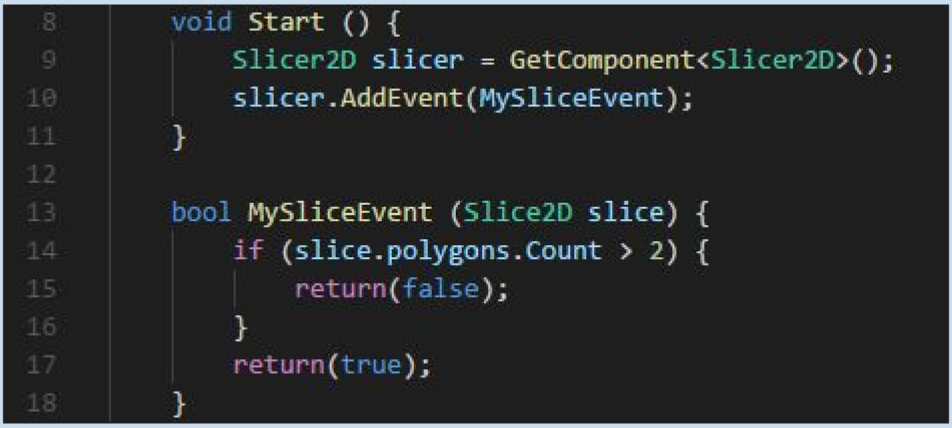
Sample 1



```
void Start () {
    Slicer2D slicer = GetComponent<Slicer2D>();
    slicer.AddEvent(MySliceEvent);
}

bool MySliceEvent (Slice2D slice) {
    Polygon2D smallestPolygon = GetSmallestPolygon(slice.polygons);
    if (smallestPolygon != null) {
        slice.polygons.Remove(smallestPolygon);
    }
    return(true);
}

Polygon2D GetSmallestPolygon(List<Polygon2D> polygonList) {
    float smallestArea = 1e+10f;
    Polygon2D smallestPolygon = null;

    foreach(Polygon2D poly in polygonList) {
        if (poly.GetArea() < smallestArea) {
            smallestPolygon = poly;
            smallestArea = poly.GetArea();
        }
    }
    return(smallestPolygon);
}
```

Removes smallest slice from result
Smallest new polygon wont be created

Sample 2

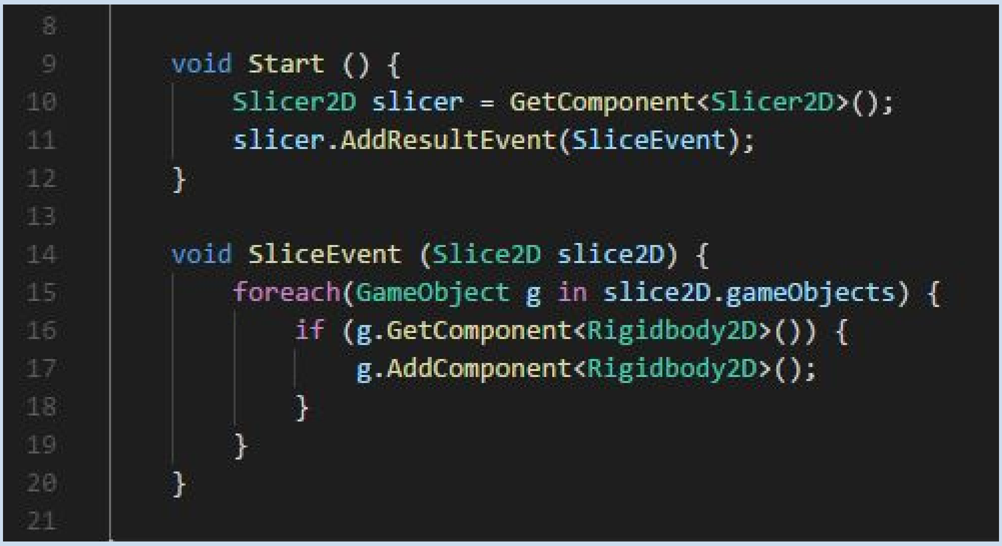


```
    void Start () {
        Slicer2D slicer = GetComponent<Slicer2D>();
        slicer.AddEvent(MySliceEvent);
    }

    bool MySliceEvent (Slice2D slice) {
        if (slice.polygons.Count > 2) {
            return(false);
        }
        return(true);
    }
```

This sample canceles the slice if it creates more than 2 new objects

AddResultEvent	
description	This event is applied after new sliced objects are created. This event method cannot cancel slice. Slice2D object contains fields: • GameObjects • SliceType • Polygons • Collisions • Slice • Slices

Sample 1



```

    void Start () {
        Slicer2D slicer = GetComponent<Slicer2D>();
        slicer.AddResultEvent(SliceEvent);
    }

    void SliceEvent (Slice2D slice2D) {
        foreach(GameObject g in slice2D.gameObjects) {
            if (g.GetComponent<Rigidbody2D>()) {
                g.AddComponent<Rigidbody2D>();
            }
        }
    }

```

This sample attaches a rigidbody for every new slice of the object

Slicer2D Low Level Scripting API

This API is used to create your own custom slicer controllers and unique behaviour

Functions

LinearSlice	*Slice an object linearly using a line by 2 given points*
LinearCutSlice	*Slice an object linearly using a line by 2 given points & width*
PointSlice	*Slice an object linearly by a point and given rotation*
ComplexSlice	*Slice an object non-linearly by given points list*
PolygonSlice	*Slice a piece of area from a polygon - using polygon as a parameter*
ExplodingSlice	*Explode a polygon by a given point*
Explode	*Explode a polygon*
CreatorSlice	*Convert a slice (points list) into a polygon*

Function Reference

Static Slice2D LinearSlice (**Polygon2D** polygon, **Pair2D** slice)	
returns	*Slice2D* object
description	*Slice an object linearly using a line by 2 given points*
information	*<not documented yet>*

Static Slice2D LinearCutSlice (**Polygon2D** polygon, **LinearCut** slice)	
returns	*Slice2D* object
description	*Slice an object linearly using a line by 2 given points & width*
information	*<not documented yet>*

Static Slice2D PointSlice (**Polygon2D** polygon, **Vector2D** point, **float** rotation)	
returns	*Slice2D* object
description	*Slice an object linearly by a point and given rotation*
information	*<not documented yet>*

Static List<Slice2D> ComplexSlice (**Polygon2D** polygon, **List<Vector2D>** slicePoints)	
returns	*Slice2D* Object
description	*Slice an object non-linearly by given points list*
information	*<not documented yet>*

Static Slice2D PolygonSlice (**Polygon2D** polygon, **Polygon2D** slicePolygon)	
returns	*Slice2D* object
description	*Slice a piece of area from a polygon - using polygon as a parameter*
information	*<not documented yet>*

Static Slice2D ExplodingSlice (**Polygon2D** polygon, **Vector2D** point)	
returns	*Slice2D* object
description	*Explode a polygon by a given point*
information	*<not documented yet>*

Static Slice2D Explode(**Polygon2D** polygon)	
returns	*Slice2D* object
description	*Explode a polygon*
information	*<not documented yet>*

Static Slice2D CreatorSlice (**List<Vector2D>** pointsList)	
returns	*Slice2D* object
description	*Convert a slice (points list) into a polygon*
information	*<not documented yet>*

Math2D Scripting API

Functions

PointInPoly	*<not documented yet>*
PolyInPoly	*<not documented yet>*
PolyIntersectPoly	*<not documented yet>*
SliceIntersectPoly	*<not documented yet>*
LineIntersectPoly	*<not documented yet>*
LineIntersectLine	*<not documented yet>*
LineIntersectCircle	*<not documented yet>*
GetPointLineIntersectLine	*<not documented yet>*
GetListLineIntersectPoly	*<not documented yet>*
GetListPolyIntersectSlice	*<not documented yet>*
GetListLineIntersectSlice	*<not documented yet>*

Polygon2D Scripting API

Functions

AddHole	*<not documented yet>*
AddPoint	*<not documented yet>*
PolyInPoly	*<not documented yet>*
PointInPoly	*<not documented yet>*
PointInHole	*<not documented yet>*
ToLocalSpace	*<not documented yet>*
ToWorldSpace	*<not documented yet>*
ToOffset	*<not documented yet>*
Normalize	*<not documented yet>*
GetArea	*<not documented yet>*
GetBounds	*<not documented yet>*
LineIntersectHoles	*<not documented yet>*
SliceIntersectPoly	*<not documented yet>*
SliceIntersectHoles	*<not documented yet>*
GetListSliceIntersectPoly	*<not documented yet>*
Polygon2D.GetColliderType	*<not documented yet>*
Polygon2D.CreateFromCollider	*<not documented yet>*
Polygon2D.Create	*<not documented yet>*
Polygon2D.GetListFromCollider	*<not documented yet>*

Vector2D Scripting API

Functions

Set	*<not documented yet>*
Push	*<not documented yet>*
Inc	*<not documented yet>*
Dec	*<not documented yet>*
Distance	*<not documented yet>*
Atan2	*<not documented yet>*

FAQ:
can I calculate slice's area using your utils library?
That's my small fix for the problem. +1 is because result contains an original piece.
Within the API, which is the best method to access the the properties (specifically the rigidbody) of the objects that has been created after a slice
Is it possible to have 2 joints attached to the same object, slice it, and have 2 slices appear each with their respective joint still attached?