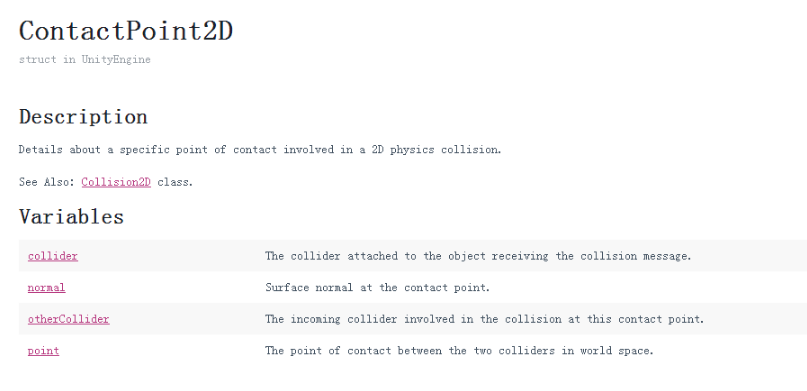
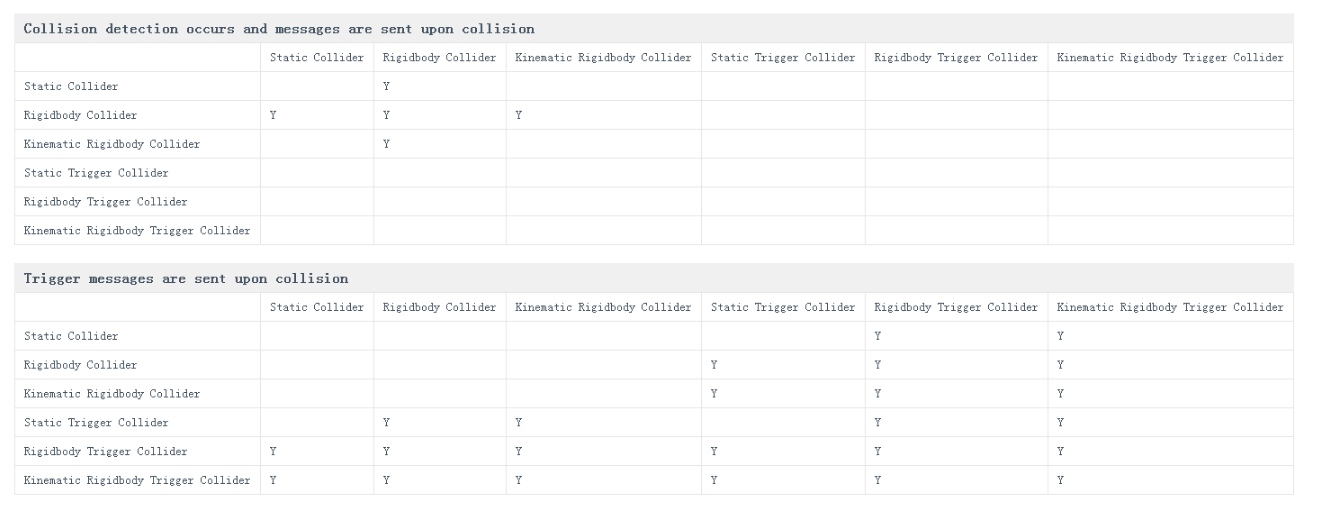
# **Unity中获得碰撞体碰撞的位置**

在纯物理环境中，为了获得碰撞体碰撞的位置，我们可以使用Collider2D.OnCollisionEnter2D(Collision2D)来接受碰撞，Collision2D中存有碰撞信息Collision2D.contacts(The specific points of contact with the incoming Collider2D), contacts的类型是ContactPoint2D[]



通过ContactPoint2D，我们可以获得世界坐标系中两个碰撞体碰撞的位置。

当然，出发Collider2D.OnCollisionEnter2D(Collision2D)是有条件的，且较触发器更为苛刻，官网文档给出的条件（Unity ManualPhysicsPhysics OverviewColliders）如下：



我现在有特殊的需求，一个静态碰撞体和一个鼠标拖动的超小碰撞体（近似认为是一个点），要求他们碰撞的位置。当然，我设置碰撞体为IsKinematic的时候，OnCollisionEnter2D就会失效，所以无法用上面的方法获得碰撞点，那么来思考一下使用OnTriggerEnter2D是否好使？查看文档可以知道，OnTriggerEnter触发的时候，并不会记录点信息，故我们需要自己思考如何获得碰撞点。为了使用触发器来获得碰撞点，我也试了很多方法，试过将刚体的Collision Detection置为Continuous，试过修改差值方式等，均不好使……每次OnTriggerEnter2D触发的时候，一个触发器已经嵌入到另一个触发器之中，ORZ, 所以能得到的信息就是在触发的这一帧的位置。不过上一帧的位置我们也能得到，自己测试了一下，想在LateUpdate中记录上一帧的位置，不过这个位置也就是触发时候的位置，所以还需要一个变量记录真正的上一帧的位置。现在得到了两个有用的变量：触发器触发时候的位置 currentPos以及上移帧的位置 lastPos，静态碰撞体的边缘就在这两个位置之间，我从lastPos像currentPos发射一条射线：

1

Vector3 origin = m\_LastLastPos;//上一帧的位置

2

Vector3 end = GameContext.MouseFollowPosition;//触发时的位置

3

Vector3 direction = end - origin;//射线方向

4

float distance = Vector3.Distance(origin, end);//射线检测距离

5

RaycastHit2D hit = Physics2D.Raycast(origin, direction, distance, 1 &lt;&lt; LayerMask.NameToLayer("Target"));//发射射线，只检测与"Target"层的碰撞

6

Debug.DrawRay(origin, direction, Color.red, 2);//绘制射线

7

Debug.Assert(hit.collider != null, "未检测到起点");

8

if (hit.collider != null)

9

｛

10

m\_StartPoint = hit.point;//获得该碰撞点

11

｝

这就得到了所需要的碰撞点了。