2D Collider PRO

ArmNomads Games

2D Camera Collider

To use this component , select your game camera (orthographic only) and apply - Tools / 2DColliderPRO > Add > CameraCollider

This component creates 2D camera border collider. If you move or resize your camera, collider will dynamically change its points.

*Demo scene is included.

Public Methodes.

- `SetCollider(bool _is_trigger , bool _use_effector , PhysicsMaterial2D _physics_Material_2D)`

  Creates and initializes 2D edge collider for camera borders.This method will be automatically called at Awake() , if set_collider_at_start is checked in inspector.

- `Update_Collider_Size()`

  Updates 2D collider points positions when camera moves or orthographic size is changed. This method will be automatically called in FixedUpdate() , if is_dynamic is checked in inspector.

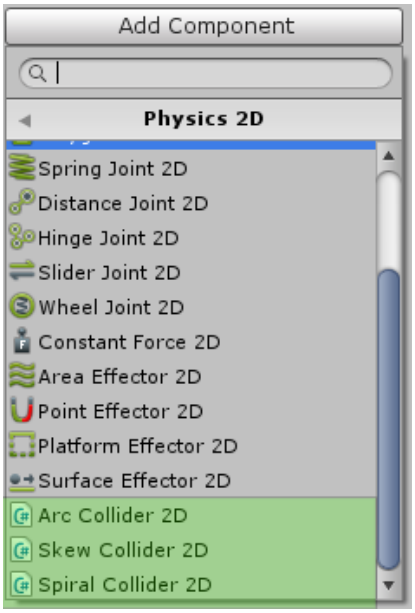
2D Custom Colliders

To use this component , select your gameobject and apply -

Add Component > Physics2D > ArcCollider2D

Add Component > Physics2D > SkewCollider2D

Add Component > Physics2D > SpiralCollider2D



ArcCollider2D

This component creates 2D edge collider with Arc settings.

Skew Collider2D

This component creates 2D polygon collider with Skew settings.

Spiral Collider2D

This component creates 2D edge collider with Spiral settings.

Arc Collider 2D, Skew Collider 2D and Spiral Collider 2D are Highly customizable and can be combined with Show2DCollider.

*Demo scene is included.

Show 2D Collider

To use this component , select your gameobject (which contains 2D Collider) and apply – Tools / 2DColliderPRO > Show 2D Collider

This component creates visual zone of your 2D collider, which is perfect for debugging. You can set individual settings for all your 2D colliders. If your gameobject has 2 or more 2D colliders , you can add this component multiple times for all of them. Also it has collision time settings (changes color when colliding). Works with all types of 2D Colliders.
If you want to see it in game window, just enable Gizmos button.

*Demo scene is included.

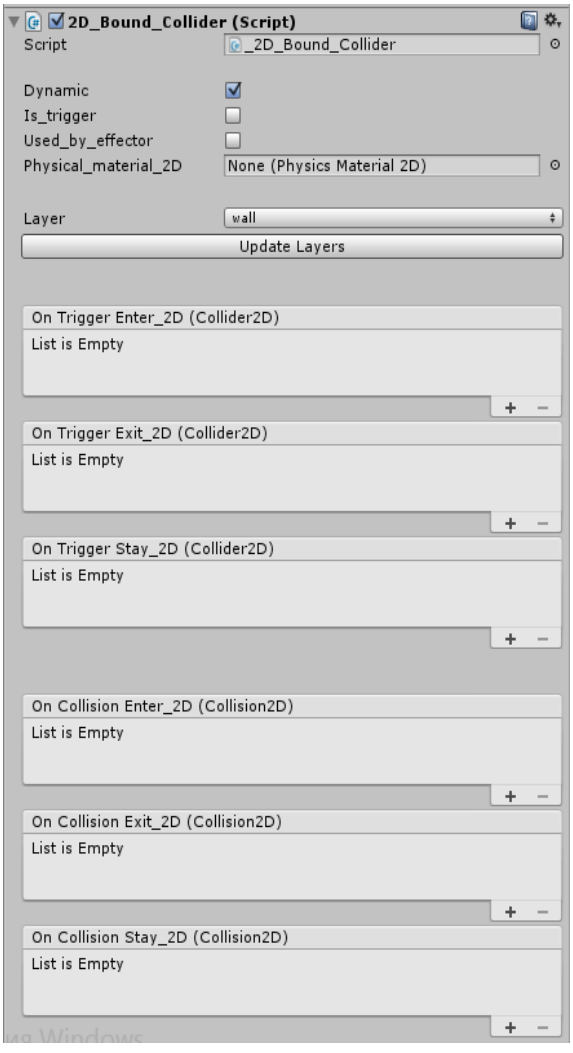
2D SortingRaycast

2D Sorting Raycast is a static method of _2D_Collider_Prostatic class. It is a raycastbased on 2D Sorting, that returns sprite's collider hit info with highest priority sorted value (returns RaycastHit2D). For the best result SpriteRenderer and Collider2D should be on the same gameobject. All parameters are the same with Physics2D.RaycastAll() . See Unity3D Scripting API.

_2D_Collider_Pro.Sorting_Raycast(Vector2 origin , Vector2 direction , float distance, int layerMask, float minDepth, float maxDepth)

*Demo scene is included.

2D Bound Collider



To use this component , select your Sprite Renderer gameobject and apply – Tools / 2DColliderPRO > Add > 2D Bound Collider

It will create 2D collider from sprite bounds at runtime. Works with static and animated sprites.

If Dynamic is checked , it will update collider size based on your sprite changes. Also it provides all 2D collider events (triggers and collisions).

You can register your trigger and collision events via code by this way .

```
2D_Bound_Collider boundCol ;
boundCol.OnTriggerEnter_2D += YourTriggerHandler ;
```

*Demo scene is included.

Public Methodes.

- public Collider2D Get_Bound_Collider()
  returns 2D bound collider.
- public void Start_Update()
  starts updating 2d collider size in FixedUpdate.
- public void Stop_Update()
  stops updating 2d collider size in FixedUpdate
- public void Set_Trigger(bool b)
  sets 2D collider to trigger
- public void Use_Effector(bool b)
  sets 2D collider to work with effectors

2D Reflection

To use this component , select your gameobject (empty) and apply –

Tools / 2DColliderPRO >Add > 2D Reflection

This component creates 2D ray, which is reflected by mirrors (layer), and stopped by obstacles (layer). Rays number can be set from inspector or via code. You can use Line Renderer for ray visualization. Also it can trigger colliders that have ActiveObject component.

Public Methodes.

- public void Activate()

  starts casting the ray.
- public void Deactivate()

  stops casting the ray.
- public void Set_Rays_Count(int count)

  sets the current rays count.
- public RaycastHit2D[ ] Get_All_Active_Hits()

  gets the raycast hits, that have colliders.
- public RaycastHit2D Get_Current_Obstacle_Hit()

  gets the current obstacle hit info.

*Demo scene is included.