

# Unity 2D Experimental Preview R1 Release Notes

## Release Notes

- **New:** 9 Slice
  - Allows Sprites to be drawn as 9-sliced image with SpriteRenderer.
- **New:** Sprite Outline Editor
  - Allow user to adjust sprite outline in Sprite Editor Window to control tight mesh generation.
- **New:** Grid
  - Allows users to specify a layout for organisation of object placement.
  - Supports 3D coordinates.
  - Provides space conversions from grid cell, local and world.
  - Supported layouts:
    - Rectangular
    - Isometric
    - Hexagonal
    - Custom (Not implemented currently)
  - Adjust cell sizes, gaps and swizzles between coordinate axes.
- **New:** Tile Map Editor
  - Programmable Tiles
    - Paint tiles on the tile map and tiles will decide what sprites, orientation and colour it needs to be based on placement.
    - Able to update data and logic for tiles and have painted tile maps refresh itself to use the updated information.
  - Programmable Brushes
    - Paint and update tiles based on provided logic.
    - Brush decides which tiles should be painted onto the target location, add details or properties to the target location.
  - **Note:** TileMapCollider2D is still under works and is coming soon!
    - As a placeholder, adding a PolygonCollider2D component to the TileMap. GameObject will generate a collider for the TileMap. This is temporary and will not be in the final production.
- **New:** SortingGroup Component
  - The SortingGroup is a component that alters the order in which Renderers are rendered.
- **New:** Sprite Masking
  - Allows Sprites to be masked in presence of mask(s).
  - Masks shapes can be driven by sprites. More control is available using the alpha threshold of the sprite driving the shape of the mask.
  - The Mask interaction for each sprite can be set to “Show under mask” or “Show outside mask”. Default is no interaction with any mask.
- **New:** CapsuleCollider2D Component
  - Capsule shaped collider defined as an area and direction (to control end-caps)
  - ‘*Physics2D.CapsuleCast ()*’, ‘*Physics2D.CapsuleCastAll ()*’ & ‘*Physics2D.CapsuleCastNonAlloc ()*’ API.

- *NOTE: This collider does not stop Box2D from producing contacts on adjacent box-colliders; that will be solved by the collider-merging that is currently WIP!*
- **New:** Edge-Radius property for BoxCollider2D & EdgeCollider2D components
  - BoxCollider2D & EdgeCollider2D now have an 'edgeRadius' property allowing a circular radius extending from all edges to be used for collision detection. This results in capsule-like edges and rounded vertices/corners.
- **Improvement:** Rewritten contact handling back-end
  - Due to the fact that Unity allows a lot of dynamic changes to happen to both Rigidbody2D and Collider2D, changes which Box2D wasn't designed for, it makes contact handling and reporting fairly complex.
  - The back-end contact handling mechanism has been completely rewritten to ensure that all contact reports coming from Box2D are handled efficiently and correctly no matter what front-end changes Unity makes to Collider2D or Rigidbody2D.
  - All 'CollisionExit' and 'TriggerExit' contact conditions are now handled correctly when deleting, re-positioning or resizing a Collider2D.
- **Improvement:** Rigidbody2D component
  - New '*Rigidbody2D.bodyType*' property allowing the selection of Dynamic, Kinematic or Static body type. ('*Rigidbody2D.isKinematic*' still available but will be deprecated later)
  - New '*Rigidbody2D.simulated*' property exposed to inspector allowing the body, collider & joint simulation to be turned on or off really fast.
  - New '*Rigidbody2D.useFullKinematicContacts*' property allowing kinematic bodies to contact other static/kinematic bodies (in addition to already contacting dynamic bodies) returning full contacts in 'OnCollisionXXX' callbacks.
  - New '*Rigidbody2D.material*' property allowing the body to set the physics material for all attached colliders that do not specify a physics material.
- **Improvement:** PlatformEffector2D component
  - Now has a '*PlatformEffector2D.rotationalOffset*' property allowing local 'up' to change.
- **Improvement:** PolygonCollider2D & EdgeCollider2D Inspector point editing
  - The contacts roll-out shows in realtime, all contacts for the Rigidbody2D (all its colliders) and Collider2D.
  - Both now allow paths/points to be modified directly inside the inspector.
- **Improvement:** All Rigidbody2D and Collider2D have inspector 'Info' roll-out
  - Useful Information shown in the inspector for all rigidbodies and colliders.
- **Improvement:** All Rigidbody2D and Collider2D have inspector 'Contacts' roll-out.
- **New:** 2D Physics Casting API
  - New '*Collider2D.Raycast ()*' - Cast a ray from the collider position
  - New '*Collider2D.Cast ()*' - Cast the collider through the world
  - New '*Rigidbody2D.OverlapPoint ()*' - Check if a point overlaps any of the colliders attached to the body
  - New '*Rigidbody2D.Cast ()*' - Cast the body (all its colliders) through the world
  - New '*Physics2D.OverlapBox ()*', '*Physics2D.OverlapBoxAll ()*' & '*Physics2D.OverlapBoxNonAlloc ()*' - Box overlaps using point, size & angle

- **New:** 2D Physics Contacts API
  - Rewritten contacts back-end allows fast querying of all contacts either on a Rigidbody2D or Collider2D using a non-allocating API as highlighted here: <https://oc.unity3d.com/index.php/s/ruNynjAFTQ4AYe9>
  - Physics2D/Collider2D/Rigidbody2D 'GetContacts()' method can return full ContactPoint2D or just Collider2D.
  - A more convenient and faster method to filter results is provided by the new ContactFilter2D. Filtering happens before the results are passed back and can filter on contact-enabled state, layer-mask, min/max Z-depth & collision-normal range.
  - ContactPoint2D expanded to include relative velocity at each contact point, both normal and tangent impulses applied by the solver as well as the full Collider2D/Rigidbody2D pair.
- **New:** 2D Physics Gizmo Control
  - Gizmos roll-out in Physics 2D setting fully configurable from script:
    - Should all colliders be shown without the need to be selected?
    - Should collider sleep-state be shown?
    - Configurable enabled (awake)/sleep collider colors
    - Should collider contacts be shown?
    - Configurable collider contact scale and color
    - Collider AABB and color