bpy_extras submodule (bpy_extras.view3d_utils)

bpy_extras.view3d_utils.**region_2d_to_vector_3d(region, rv3d, coord)**

Return a direction vector from the viewport at the specific 2d region coordinate.

PARAMETERS:

- **region** (`bpy.types.Region`) – region of the 3D viewport, typically bpy.context.region.
- **rv3d** (`bpy.types.RegionView3D`) – 3D region data, typically bpy.context.space_data.region_3d.
- **coord** (*2d vector*) – 2d coordinates relative to the region: (event.mouse_region_x, event.mouse_region_y) for example.

RETURNS:

normalized 3d vector.

RETURN TYPE:

`mathutils.Vector`

bpy_extras.view3d_utils.**region_2d_to_origin_3d(region, rv3d, coord, *, clamp=None)**

Return the 3d view origin from the region relative 2d coords.

Note

Orthographic views have a less obvious origin, the far clip is used to define the viewport near/far extents. Since far clip can be a very large value, the result may give with numeric precision issues.

To avoid this problem, you can optionally clamp the far clip to a smaller value based on the data you're operating on.

PARAMETERS:

- **region** (`bpy.types.Region`) – region of the 3D viewport, typically bpy.context.region.
- **rv3d** (`bpy.types.RegionView3D`) – 3D region data, typically bpy.context.space_data.region_3d.
- **coord** (*Sequence[float]*) – 2D coordinates relative to the region; (event.mouse_region_x, event.mouse_region_y) for example.
- **clamp** (*float | None*) – Clamp the maximum far-clip value used. (negative value will move the offset away from the view_location)

RETURNS:

The origin of the viewpoint in 3d space.

RETURN TYPE:

`mathutils.Vector`

bpy_extras.view3d_utils.**region_2d_to_location_3d(region, rv3d, coord, depth_location)**

Return a 3d location from the region relative 2d coords, aligned with *depth_location*.

PARAMETERS:

- **region** (`bpy.types.Region`) – region of the 3D viewport, typically bpy.context.region.
- **rv3d** (`bpy.types.RegionView3D`) – 3D region data, typically bpy.context.space_data.region_3d.
- **coord** (*2d vector*) – 2d coordinates relative to the region; (event.mouse_region_x, event.mouse_region_y) for example.
- **depth_location** (*3d vector*) – the returned vectors depth is aligned with this since there is no defined depth with a 2d region input.

RETURNS:

normalized 3d vector.

RETURN TYPE:

`mathutils.Vector`

bpy_extras.view3d_utils.**location_3d_to_region_2d(region, rv3d, coord, *, default=None)**

Return the *region* relative 2d location of a 3d position.

PARAMETERS:

- **region** (`bpy.types.Region`) – region of the 3D viewport, typically bpy.context.region.
- **rv3d** (`bpy.types.RegionView3D`) – 3D region data, typically bpy.context.space_data.region_3d.
- **coord** (*3d vector*) – 3d world-space location.
- **default** – Return this value if `coord` is behind the origin of a perspective view.

RETURNS:

2d location

RETURN TYPE:

`mathutils.Vector` | Any