Skip to content

BMesh Utilities (bmesh.utils)

This module provides access to blenders bmesh data structures.

bmesh.utils.**edge_rotate(edge, ccw=False)**

Rotate the edge and return the newly created edge. If rotating the edge fails, None will be returned.

PARAMETERS:

- **edge** (`bmesh.types.BMEdge`) – The edge to rotate.
- **ccw** (*bool*) – When True the edge will be rotated counter clockwise.

RETURNS:

The newly rotated edge.

RETURN TYPE:

`bmesh.types.BMEdge`

bmesh.utils.**edge_split(edge, vert, fac)**

Split an edge, return the newly created data.

PARAMETERS:

- **edge** (`bmesh.types.BMEdge`) – The edge to split.
- **vert** (`bmesh.types.BMVert`) – One of the verts on the edge, defines the split direction.
- **fac** (*float*) – The point on the edge where the new vert will be created [0 - 1].

RETURNS:

The newly created (edge, vert) pair.

RETURN TYPE:

tuple[`bmesh.types.BMEdge`, `bmesh.types.BMVert`]

bmesh.utils.**face_flip(faces)**

Flip the faces direction.

PARAMETERS:

face (`bmesh.types.BMFace`) – Face to flip.

bmesh.utils.**face_join(faces, remove=True)**

Joins a sequence of faces.

PARAMETERS:

- **faces** (`bmesh.types.BMFace`) – Sequence of faces.
- **remove** (*bool*) – Remove the edges and vertices between the faces.

RETURNS:

The newly created face or None on failure.

RETURN TYPE:

`bmesh.types.BMFace`

bmesh.utils.**face_split(face, vert_a, vert_b, coords=(), use_exist=True, example=None)**

Face split with optional intermediate points.

PARAMETERS:

- **face** (`bmesh.types.BMFace`) – The face to cut.
- **vert_a** (`bmesh.types.BMVert`) – First vertex to cut in the face (face must contain the vert).
- **vert_b** (`bmesh.types.BMVert`) – Second vertex to cut in the face (face must contain the vert).

- **coords** (*Sequence[Sequence[float]]*) – Optional sequence of 3D points in between *vert_a* and *vert_b*.
- **use_exist** (*bool*) – .Use an existing edge if it exists (Only used when *coords* argument is empty or omitted)
- **example** (`bmesh.types.BMEdge`) – Newly created edge will copy settings from this one.

RETURNS:

The newly created face or None on failure.

RETURN TYPE:

tuple[`bmesh.types.BMFace`, `bmesh.types.BMLoop`]

bmesh.utils.**face_split_edgenet(face, edgenet)**

Splits a face into any number of regions defined by an edgenet.

PARAMETERS:

- **face** (`bmesh.types.BMFace`) – The face to split.
- **face** – The face to split.
- **edgenet** (Sequence[`bmesh.types.BMEdge`]) – Sequence of edges.

RETURNS:

The newly created faces.

RETURN TYPE:

tuple[`bmesh.types.BMFace`, …]

Note

Regions defined by edges need to connect to the face, otherwise they’re ignored as loose edges.

bmesh.utils.**face_vert_separate(face, vert)**

Rip a vertex in a face away and add a new vertex.

PARAMETERS:

- **face** (`bmesh.types.BMFace`) – The face to separate.
- **vert** (`bmesh.types.BMVert`) – A vertex in the face to separate.

RETURN VERT:

The newly created vertex or None on failure.

RTYPE VERT:

`bmesh.types.BMVert`

Note

This is the same as loop_separate, and has only been added for convenience.

bmesh.utils.**loop_separate(loop)**

Rip a vertex in a face away and add a new vertex.

PARAMETERS:

loop (`bmesh.types.BMLoop`) – The loop to separate.

RETURN VERT:

The newly created vertex or None on failure.

RTYPE VERT:

`bmesh.types.BMVert`

bmesh.utils.**vert_collapse_edge(vert, edge)**

Collapse a vertex into an edge.

PARAMETERS:

- **vert** (`bmesh.types.BMVert`) – The vert that will be collapsed.
- **edge** (`bmesh.types.BMEdge`) – The edge to collapse into.

RETURNS:

The resulting edge from the collapse operation.

RETURN TYPE:

`bmesh.types.BMEdge`

bmesh.utils.**vert_collapse_faces(vert, edge, fac, join_faces)**

Collapses a vertex that has only two manifold edges onto a vertex it shares an edge with.

PARAMETERS:

- **vert** (`bmesh.types.BMVert`) – The vert that will be collapsed.
- **edge** (`bmesh.types.BMEdge`) – The edge to collapse into.
- **fac** (*float*) – The factor to use when merging customdata [0 - 1].
- **join_faces** (*bool*) – When true the faces around the vertex will be joined otherwise collapse the vertex by merging the 2 edges this vertex connects to into one.

RETURNS:

The resulting edge from the collapse operation.

RETURN TYPE:

`bmesh.types.BMEdge`

bmesh.utils.**vert_dissolve(vert)**

Dissolve this vertex (will be removed).

PARAMETERS:

vert (`bmesh.types.BMVert`) – The vert to be dissolved.

RETURNS:

True when the vertex dissolve is successful.

RETURN TYPE:

bool

bmesh.utils.**vert_separate(vert, edges)**

Separate this vertex at every edge.

PARAMETERS:

- **vert** (`bmesh.types.BMVert`) – The vert to be separated.
- **edges** (`bmesh.types.BMEdge`) – The edges to separated.

RETURNS:

The newly separated verts (including the vertex passed).

RETURN TYPE:

tuple[`bmesh.types.BMVert` , …]

bmesh.utils.**vert_splice(vert, vert_target)**

Splice vert into vert_target.

PARAMETERS:

- **vert** (`bmesh.types.BMVert`) – The vertex to be removed.
- **vert_target** (`bmesh.types.BMVert`) – The vertex to use.

Note

The verts mustn’t share an edge or face.

Copyright © Blender Authors
Made with Furo