

[Skip to content](#)

Freestyle Chaining Iterators (freestyle.chainingiterators)

This module contains chaining iterators used for the chaining operation to construct long strokes by concatenating feature edges according to selected chaining rules. The module is also intended to be a collection of examples for defining chaining iterators in Python.

class freestyle.chainingiterators.ChainPredicateIterator

Class hierarchy: `freestyle.types.Iterator` > `freestyle.types.ViewEdgeIterator` > `freestyle.types.ChainingIterator` > `ChainPredicateIterator`

A “generic” user-controlled ViewEdge iterator. This iterator is in particular built from a unary predicate and a binary predicate. First, the unary predicate is evaluated for all potential next ViewEdges in order to only keep the ones respecting a certain constraint. Then, the binary predicate is evaluated on the current ViewEdge together with each ViewEdge of the previous selection. The first ViewEdge respecting both the unary predicate and the binary predicate is kept as the next one. If none of the potential next ViewEdge respects these two predicates, None is returned.

`__init__(upred, bpred, restrict_to_selection=True, restrict_to_unvisited=True, begin=None, orientation=True)`

`__init__(brother)`

Builds a ChainPredicateIterator from a unary predicate, a binary predicate, a starting ViewEdge and its orientation or using the copy constructor.

PARAMETERS:

- **upred** (`freestyle.types.UnaryPredicateID`) – The unary predicate that the next ViewEdge must satisfy.
- **bpred** (`freestyle.types.BinaryPredicateID`) – The binary predicate that the next ViewEdge must satisfy together with the actual pointed ViewEdge.
- **restrict_to_selection** (*bool*) – Indicates whether to force the chaining to stay within the set of selected ViewEdges or not.
- **restrict_to_unvisited** (*bool*) – Indicates whether a ViewEdge that has already been chained must be ignored or not.
- **begin** (`freestyle.types.ViewEdge` | None) – The ViewEdge from where to start the iteration.
- **orientation** (*bool*) – If true, we’ll look for the next ViewEdge among the ViewEdges that surround the ending ViewVertex of begin. If false, we’ll search over the ViewEdges surrounding the ending ViewVertex of begin.
- **brother** (`ChainPredicateIterator`) – A ChainPredicateIterator object.

class freestyle.chainingiterators.ChainSilhouetteIterator

Class hierarchy: `freestyle.types.Iterator` > `freestyle.types.ViewEdgeIterator` > `freestyle.types.ChainingIterator` > `ChainSilhouetteIterator`

A ViewEdge Iterator used to follow ViewEdges the most naturally. For example, it will follow visible ViewEdges of same nature. As soon, as the nature or the visibility changes, the iteration stops (by setting the pointed ViewEdge to 0). In the case of an iteration over a set of ViewEdge that are both Silhouette and Crease, there will be a precedence of the silhouette over the crease criterion.

`__init__(restrict_to_selection=True, begin=None, orientation=True)`

`__init__(brother)`

Builds a ChainSilhouetteIterator from the first ViewEdge used for iteration and its orientation or the copy constructor.

PARAMETERS:

- **restrict_to_selection** (*bool*) – Indicates whether to force the chaining to stay within the set of selected ViewEdges or not.
- **begin** (`freestyle.types.ViewEdge` | None) – The ViewEdge from where to start the iteration.
- **orientation** (*bool*) – If true, we’ll look for the next ViewEdge among the ViewEdges that surround the ending ViewVertex of begin. If false, we’ll search over the ViewEdges surrounding the ending ViewVertex of begin.
- **brother** (`ChainSilhouetteIterator`) – A ChainSilhouetteIterator object.

class freestyle.chainingiterators.pyChainSilhouetteIterator

Natural chaining iterator that follows the edges of the same nature following the topology of objects, with decreasing priority for silhouettes, then borders, then suggestive contours, then all other edge types. A ViewEdge is only chained once.

`init()`

traverse(iter)

class freestyle.chainingiterators.pyChainSilhouetteGenericIterator

Natural chaining iterator that follows the edges of the same nature following the topology of objects, with decreasing priority for silhouettes, then borders, then suggestive contours, then all other edge types.

__init__(self, stayInSelection=True, stayInUnvisited=True)

Builds a pyChainSilhouetteGenericIterator object.

PARAMETERS:

- **stayInSelection** (*bool*) – True if it is allowed to go out of the selection
- **stayInUnvisited** (*bool*) – May the same ViewEdge be chained twice

init()

traverse(iter)

class freestyle.chainingiterators.pyExternalContourChainingIterator

Chains by external contour

checkViewEdge(ve, orientation)

init()

traverse(iter)

class freestyle.chainingiterators.pySketchyChainSilhouetteIterator

Natural chaining iterator with a sketchy multiple touch. It chains the same ViewEdge multiple times to achieve a sketchy effect.

__init__(self, nRounds=3, stayInSelection=True)

Builds a pySketchyChainSilhouetteIterator object.

PARAMETERS:

- **nRounds** (*int*) – Number of times every Viewedge is chained.
- **stayInSelection** (*bool*) – if False, edges outside of the selection can be chained.

init()

make_sketchy(ve)

Creates the sketchy effect by causing the chain to run from the start again. (loop over itself again)

traverse(iter)

class freestyle.chainingiterators.pySketchyChainingIterator

Chaining iterator designed for sketchy style. It chains the same ViewEdge several times in order to produce multiple strokes per ViewEdge.

init()

traverse(iter)

class freestyle.chainingiterators.pyFillOcclusionsRelativeChainingIterator

Chaining iterator that fills small occlusions

__init__(self, percent)

Builds a pyFillOcclusionsRelativeChainingIterator object.

Builds a `pyFillOcclusionsRelativeChainingIterator` object.

PARAMETERS:

percent (*float*) – The maximal length of the occluded part, expressed in a percentage of the total chain length.

init()

traverse(iter)

class `freestyle.chainingiterators.pyFillOcclusionsAbsoluteChainingIterator`

Chaining iterator that fills small occlusions

__init__(self, length)

Builds a `pyFillOcclusionsAbsoluteChainingIterator` object.

PARAMETERS:

length (*int*) – The maximum length of the occluded part in pixels.

init()

traverse(iter)

class `freestyle.chainingiterators.pyFillOcclusionsAbsoluteAndRelativeChainingIterator`

Chaining iterator that fills small occlusions regardless of the selection.

__init__(self, percent, l)

Builds a `pyFillOcclusionsAbsoluteAndRelativeChainingIterator` object.

PARAMETERS:

- **percent** (*float*) – The maximal length of the occluded part as a percentage of the total chain length.
- **l** (*float*) – Absolute length.

init()

traverse(iter)

class `freestyle.chainingiterators.pyFillQi0AbsoluteAndRelativeChainingIterator`

Chaining iterator that fills small occlusions regardless of the selection.

__init__(self, percent, l)

Builds a `pyFillQi0AbsoluteAndRelativeChainingIterator` object.

PARAMETERS:

- **percent** (*float*) – The maximal length of the occluded part as a percentage of the total chain length.
- **l** (*float*) – Absolute length.

init()

traverse(iter)

class `freestyle.chainingiterators.pyNoIdChainSilhouetteIterator`

Natural chaining iterator that follows the edges of the same nature following the topology of objects, with decreasing priority for silhouettes, then borders, then suggestive contours, then all other edge types. It won't chain the same `ViewEdge` twice.

__init__(self, stayInSelection=True)

Builds a `pyNoIdChainSilhouetteIterator` object.

PARAMETERS:

stayInSelection (*bool*) – True if it is allowed to go out of the selection

init()

traverse(iter)

[Previous](#)

[Freestyle Functions \(freestyle.functions\)](#)

[Report issue on this page](#)

Copyright © Blender Authors

Made with [Furo](#)

[Freestyle Shaders \(freestyle.shader\)](#)

[Next](#)