

Autoparter

Dokumentation

Einleitung

Dieses Plugin ist im Rahmen des Software-Projekt-Praktikums "Multi Media Madness" an der RWTH Aachen im Sommersemester 2020 entstanden.

Aufbau

User Interface

Autoparter.py

Alle Abläufe des UI's werden hier geregelt. Die Datei teilt sich in allgemeine Events, Welcome, Cutsurface Selection, Create Connectors sowie allgemeine Getter/Setter.

AutoparterToolHandle.py

In dieser Datei wird das Erscheinungsbild der "Handles" festgelegt.

Einstiegspunkte (Buttons Actions):

Welcome.qml



<u>Sub-Window</u>	<u>Button</u>	<u>Aktion</u>
-	Part object	Autoparter._startPressed







PlaneSelection.qml

<u>Sub-Window</u>	<u>Button</u>	<u>Aktion</u>
-	Select	Autoparter._selectPlanePressed
-	x	Autoparter._cancelPressed

CreateConnectors.qml

<u>Sub-Window</u>	<u>Button</u>	<u>Aktion</u>
1	Create Connectors	Autoparter._createConnectorsPressed
1	..or complete	Autoparter._cancelPressed

2	Add -> Cubic	Autoparter._beginAddingCubicConnector
2	Create	Autoparter._finallyCreateConnectorsPressed
2	x	Autoparter._cancelPressed
2		Autoparter._switchToCutSurfaceA
2		Autoparter._switchToCutSurfaceB

3	Add/Edit	Autoparter._addCubicConnector
3	x	Autoparter._cancelAddingConnector
3		Autoparter._editCubicConnector
3	x	Autoparter._deleteConnector
3	    	Autoparter.setCameraView

Jobs (Operationen in einem Thread)

Jobs.py

SliceMeshJob

- Operations.toTriMesh
- Operations.slice_mesh
- Operations.toMeshData

CreateCubicConnectorsJob

- Operations.add_cubic_connectors
- Operations.toMeshData

IsRectInPolyJob

→ Operations.is_vertex_in_poly

IsCollidingWithOtherRectsJob

IsCollidingWithMeshJob

Operationen

Operations.py

slice_mesh(Trimesh, plane_normal=[], plane_origin=[]) ->

Trimesh, (Trimesh, Trimesh, (Path, Matrix))

-Erhält als Eingabe ein Mesh und teilt dieses in zwei Meshes auf. Teilungswerkzeug ist eine Schnittebene, die durch ihren Orts- und Normalenvektor beschrieben wird. Gibt das Teil zurück, welches positiv im Verhältnis zum Normalenvektor der Schnittebene ist. Das Teil wird sowohl mit abgeschlossener als auch offener Schnittfläche zurückgegeben. Außerdem wird die Schnittfläche als Trimesh (3D) und Path auf der xy-Ebene zurückgegeben.

add_cubic_connectors((Path,Matrix),rectanglesData, rectanglesPlane, Trimesh, Trimesh) -> Trimesh, Trimesh

Eingabe:

- Schnittfläche als Pfad auf der xy-Ebene
- Rechteckdaten: Alle Positionen, Größen und Tiefen der neuen Rechteckverbinder, sowie deren Geschlecht.
- Rechteckebene: Die Ebene auf der die Rechteckverbinder liegen
- Beide offenen Teile als Trimesh

Benutzt:

- cut_rectangle_out_from_polygons
- calculate_cubic_cap

Ausgabe:

Beide fertigen Teile als Trimesh

toTriMesh(MeshData) -> Trimesh

-Konvertiert ein Trimesh-Objekt zu MeshData-Objekt

toMeshData(Trimesh) -> MeshData

-Konvertiert MeshData-Objekt zu Trimesh-Objekt

is_vertex_in_poly ((Path, Matrix), [], ([], [])) -> Boolean

-Überprüft ob gegebene Punkte innerhalb gegebenen Polygonen liegen

cut_rectangle_out_from_polygons ([], []) -> []

-Schneidet angegebenes Rechteck aus gegebenen Polygonen aus und gibt neu entstandene Polygone zurück

calculate_cubic_cap([], Boolean) -> []

-Berechnet Faces für einen Stecker bzw. Buchse

Libraries

trimesh

Da die in Cura installierte Version von trimesh zu alt ist, wird eine neuere Version mitgeliefert

shapely, numpy

Bereits vorhanden

Einschränkungen

- Die Funktionen `Operations.add_cubic_connectors` sowie `Operations.slice_mesh` sind relativ speziell für das Plugin und müssten entkoppelt werden, um sie auch in einem anderen Kontext nutzen zu können. Zudem könnten sie durch das Verwenden von C++ Libraries beschleunigt werden. Es wurde sich bewusst dagegen entschieden, um eine einfache Installation zu gewährleisten.
- Beim erneuten Teilen muss momentan berücksichtigt werden, dass die Schnittfläche nicht einen zuvor erstellten Rechteckverbinder teilt. (Der Bug tritt nur dann auf, wenn mehr als ein Verbinder auf einer Schnittfläche zuvor hinzugefügt wurde)
- Grundsätzlich sollten die Meshes von Objekten wasserdicht sein (komplett geschlossen). Die Möglichkeit undichte Meshes zu teilen wurde zwar gewährt, kann jedoch zu Problemen führen. Außerdem sollten die Meshes keine weiteren verborgenen Meshes im Inneren aufweisen.
- Nach dem Teilen sollten die Einzelteile nicht mit anderen Tools (Skalieren/Drehen/Bewegen etc.) von Cura manipuliert werden, wenn man noch Steckverbindungen hinzufügen möchte. Erst nachdem das Hinzufügen vollständig abgeschlossen ist (bzw. auf das Hinzufügen verzichtet wurde), können die neuen Objekte wieder frei manipuliert werden.
- Während der Erstellung eines neuen Rechteckverbinders wird überprüft, ob dessen Buchse zu tief ist und auf der anderen Seite wieder aus dem Mesh hinausragt. Der entsprechende Algorithmus testet, ob ausgehend von allen vier Ecken sowie der Mitte jeweils eine Linie mit dem Mesh "hinten" kollidiert.

Vor allem bei großen Rechteck-Verbindern kann dieses zu einem negativen Ergebnis führen, obwohl der Rechteck-Verbinder mit dem Mesh kollidiert. Da auf C++ Libraries etc. verzichtet wird und eine native Implementation eines Algorithmus, der Spatial-Indexing bzw. einen AABBTree verwendet, recht umständlich ist, wurde sich für diese Variante entschieden. Der naive Test, alle Faces des Rechtecks vs. alle Faces des Meshes, ist zu langsam.

Plan / Zukunft

- Toleranz auch für die Tiefe der Rechteckverbinder (Nach einem ersten Testdruck erscheint dies ebenfalls nötig)
- Nach dem Schneiden überprüfen, ob die Einzelteile in weitere Teile zerfallen und entsprechend aufteilen. (Vor allem damit neu entstandene Einzelteile auf der Druckfläche aufliegen)
- Die Möglichkeit zusätzlich die Schnittfläche zu verschieben / drehen
- Nach dem Schneiden eines der beiden Teile spiegeln, um eine angenehmere Ansicht zu ermöglichen
- Bessere Färbung der Texturen der Einzelteile nach dem Schnitt
- Snap-Mechanismus für die Rechteckverbinder
- Shaped Connector (Der Kontur der Schnittfläche nachempfundener Verbinder mit einem definierten Abstand zum Rand)
- Sliding-Connector (Rechteckverbinder von links nach rechts oder von oben nach unten, der sich nach außen verjüngt.
- Spatial Indexing

Lizenzen

- **Flugzeug aus der Anleitung und dem Heroshot:**
World War Two Fighter Planes (<https://www.thingiverse.com/thing:2812575>) by grumpusbumpus is licensed under the Creative Commons - Attribution license.
<http://creativecommons.org/licenses/by/3.0/>
- **trimesh**
The MIT License (MIT) Copyright (c) 2019 Michael Dawson-Haggerty

Version

- Getestet mit Cura Version 4.5.0 / SDK 7.0.0, wahrscheinlich auch mit früheren Versionen kompatibel.