



Leopold-Franzens-Universität
Innsbruck

Institut für Grundlagen der
Technischen Wissenschaften

Arbeitsbereich für
Geometrie und CAD

Automatische Generierung und Bewertung von Würfelgitterobjekten für Rissleseübungen

Diplomarbeit

Betreuer: assoz. Prof. Mag. Dr. Hans-Peter Schröcker

Daniel Gasteiger (0916600)

daniel.gasteiger@student.uibk.ac.at

Innsbruck
12. Jänner 2016

Inhaltsverzeichnis

1	Realisierung des Prototyps	1
1.1	Anforderungen an den Prototyp	1
1.2	Verwendete Programme	1
1.2.1	Entwicklungsumgebung	2
1.2.2	Bibliotheken	2
1.3	Generierung der Würfelgitterobjekte	3
1.3.1	Random Lattice Polytope	3
1.3.2	Extrusionskörper	7
1.3.3	Generische Algorithmen	8
1.4	Ein- und Ausgabe	10
1.4.1	Hauptfenster	10
1.4.2	Menü	11
1.4.3	Einstellungen	12
1.4.4	Exportieren der Objekte	15
1.5	Allgemeine Features	16
1.5.1	Internationalisierung	16
1.5.2	Logging	16
A	Beispielobjekte	19

1 Realisierung des Prototyps

Beginnend mit einer Auflistung der benötigten Features und Eigenschaften, sowie den effektiv verwendeten Programmen und Bibliotheken, wird in diesem Kapitel die Implementierung vorgestellt und die einzelnen Komponenten werden dabei genauer betrachtet.

1.1 Anforderungen an den Prototyp

Die drei wichtigsten Ziele für den Prototyp sind die Generierung und Präsentation von zufälligen Würfelgitterobjekten, sowie die Bereitstellung von Wegen diese Objekte im Unterricht zu verwenden.

Außerdem sollte der Prototyp nicht an eine bestimmte Plattform/Betriebssystem gebunden sein und auch sprachliche Barrieren überwinden können. Um diese Ziele zu erreichen soll der Prototyp folgende Features umsetzen:

- Verwendung einer plattformübergreifenden Programmiersprache zur Implementierung
- Implementierung verschiedener Algorithmen zur pseudozufälligen Erzeugung neuer Würfelgitterobjekte mit Hilfe eines Startwertes
- Die genauen Generierungsparameter sollen für den Benutzer frei konfigurierbar sein
- Repräsentation der generierten Objekte als drehbares 3D-Modell und ihrer drei Risse im Programm
- Einbau von Exportfunktionen der generierten Objekte; sowohl als digitale Datei, als auch in druckbarer Form
- Verwendung von austauschbaren Sprachdateien für sämtliche angezeigten Informationen

1.2 Verwendete Programme

In diesem Teil der Arbeit werden die verwendeten Programme und Bibliotheken zur Erstellung des Prototyps kurz vorgestellt.

Für die Wahl der Programmiersprache für das Programm standen viele Kandidaten zur Verfügung. Die Programmiersprache sollte weit verbreitet sein, damit der Prototyp auch nach Fertigstellung der Diplomarbeit eventuell noch von anderen Programmierern verbessert werden kann. Außerdem sollte die graphische Umsetzung einer Benutzeroberfläche samt Visualisierung des 3D Objektes und seiner Risse möglich sein.

Diese Voraussetzungen wurden von Java¹ (Version 1.7) zusammen mit den zwei zusätzlichen Bibliotheken Java3D und UnBBoolean erfüllt.

1.2.1 Entwicklungsumgebung

Die Entwicklungsumgebung Eclipse² wurde sowohl für das Programm als auch für die Erstellung der Diplomarbeit (mit der Erweiterung TeXlipse³) verwendet.

1.2.2 Bibliotheken

Die folgenden Bibliotheken werden zusätzlich zur Java-Standardbibliothek verwendet:

Java3D

Java3D⁴ wird für die Visualisierung des 3D Modelles im Programm verwendet.

UnBBoolean

UnBBoolean⁵ wird für die binären Operationen (Vereinigung, Durchschnitt und Subtraktion) über zwei 3D Körpern verwendet.

iText

Zum Abspeichern der generierten Übungen wird die Möglichkeit der Generierung von pdf und jpg Dateien geboten. Für die Exportierung als pdf wird iText in der Version 4.2.0⁶ verwendet.

XStream

Die Bibliothek XStream⁷ wird im Programm für die Serialisierung und Deserialisierung der Benutzereinstellungen in eine XML-Datei verwendet.

¹<https://www.java.com/de/>

²<http://www.eclipse.org/home/index.php>

³<http://texlipse.sourceforge.net/>

⁴<https://java3d.java.net/>

⁵<http://unbboolean.sourceforge.net/>

⁶<https://github.com/yasory/iText-4.2.0>

⁷<http://x-stream.github.io/>

Logback

Logback⁸ wird als logging Framework verwendet um sowohl während der Entwicklung als auch im Betrieb eine bessere Nachverfolgung der internen Abläufe zu ermöglichen.

JUnit

Zur Qualitätsicherung während der Programmierungsphase wurden Unit Tests mit dem Testframework JUnit 4⁹ umgesetzt. Diese Tests haben sich während der Entwicklung als sehr nützlich erwiesen, da sie die wichtigsten Grenzfälle bei den verschiedenen Algorithmen überprüfen und gerade bei Umbauarbeiten im Code so manchen Programmfehler aufgezeigt haben.

1.3 Generierung der Würfelgitterobjekte

In diesem Abschnitt werden die verschiedenen Arten zur Generierung der Würfelgitterobjekte vorgestellt, die im Prototyp eingebaut wurden. Alle Verfahren erzeugen nur 3D-Objekte, die den Kriterien der Würfelgitterobjekten entsprechen.

1.3.1 Random Lattice Polytope

Bei der Generierung als Random Lattice Polytope wird für jeden der Würfelgitterpunkte zufällig entschieden, ob er Teil des Objektes ist, oder nicht. Sobald dieser Schritt für alle Punkte ausgeführt wurde, wird die konvexe Hülle aller gesetzten Punkte berechnet. Diese Hülle ist dann das generierte Random Lattice Polytope.

Zur Bestimmung der zufällig gesetzten Punkte wird die Standardfunktionalität zur Generierung von Zufallszahlen von Java verwendet. Dieses Verfahren benötigt einen Startwert (oft auch *random seed* genannt). Dieser Wert kann entweder zufällig (z.B. unter Einbeziehung der aktuellen Zeit) gewählt werden oder vorgegeben sein. So ist es z.B. durch Eingabe eines fixen Startwertes in den Einstellungen möglich ein bestimmtes Objekt neu zu erzeugen. Das Random Lattice Polytope in Abbildung 1 wurde mit Startwert -2027329128 erzeugt und kann durch dessen Vorgabe in den Einstellungen auch erneut generiert werden.

Das Kernstück dieser Generierungsart ist die Berechnung der konvexen Hülle. Dazu wurde der Quickhull Algorithmus (siehe [1]) verwendet, der in eine Initialisierungs- und Iterationsphase unterteilt werden kann. Für eine

⁸<http://logback.qos.ch/>

⁹<http://junit.org/>

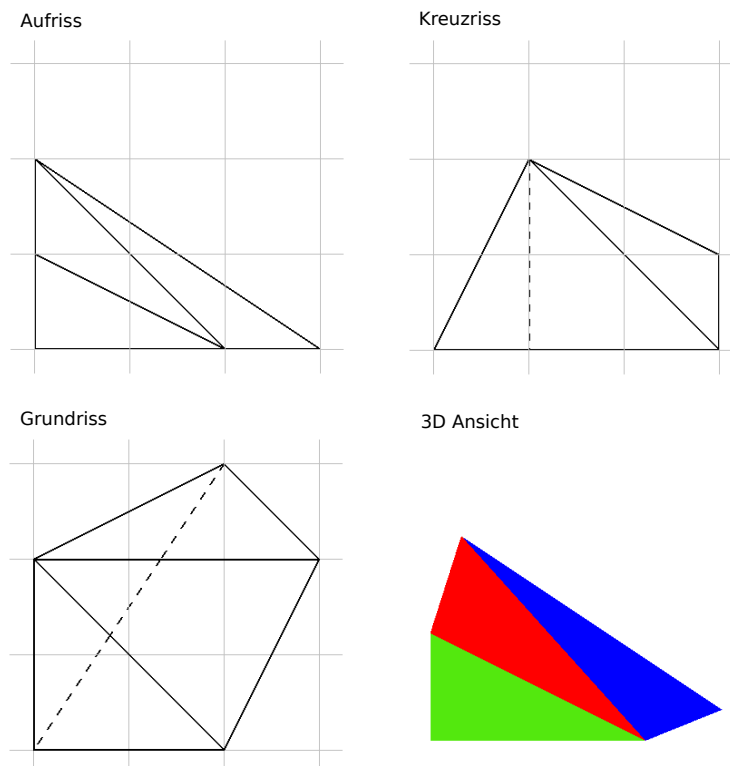


Abbildung 1: Beispiel für einen Random Lattice Polytope

bessere Vorstellung können die entsprechenden Schritte der 2D-Variante in Abbildung 2 nachverfolgt werden. Diese Variante funktioniert sehr ähnlich, kann aber wesentlich anschaulicher dargestellt werden.

Initialisierung

Als erster Schritt werden die Punkte gesucht, die am nächsten bei den 6 Seitenwänden des Würfelgitters liegen. Liegt ein Punkt an einem Eck des umschließenden Würfels könnte er drei Mal als solcher Randpunkt ausgewählt werden, was aber kein Problem ist.

Von diesen sechs Punkten werden die zwei Punkte ausgewählt, die die höchste euklidische Distanz zueinander haben (vgl. 2(b)). Diese zwei Punkte bilden die erste Kante des entstehenden Tetraeders.

Danach wird der Punkt, der am weitesten von dieser Kante entfernt ist als dritter Eckpunkt für die erste Seitenfläche ausgewählt (vgl. 2(c)). Von allen gesetzten Punkten wird nun der Punkt als vierter Eckpunkt des Tetraeders ausgewählt, der wiederum am weitesten von der Seitenfläche ent-

fernt ist. Damit ist die initiale Hülle vollständig.

Alle gesetzten Punkte werden nun überprüft ob sie außerhalb des Tetraeders liegen und wenn ja genau einer Seitenfläche zugeordnet, über der sie liegen (vgl. 2(d)). Jede Seitenfläche besitzt eine eigene Liste an zugeordneten Punkten.

Als letzter Schritt der Initialisierung wird nun noch eine Liste mit allen Seitenflächen angelegt, die in den folgenden Iterationen betrachtet werden müssen. Dieser Liste werden die vier Seitenflächen des Tetraeders hinzugefügt.

Iteration

In jeder Iteration wird der Reihe nach die Liste der Seitenflächen abgearbeitet. Hat die aktuelle Seitenfläche keine zugeordneten Punkte, so wird sie aus der Liste entfernt und die nächste Seitenfläche betrachtet (vgl. 2(f)). Sobald alle Seitenflächen aus der Liste entfernt wurden, ist die Iterationsphase beendet.

Von den zugeordneten Punkten der aktuellen Seitenfläche (in Folge S1 genannt) wird der Punkt mit der größten Entfernung (in Folge P1 genannt) zu ihr ausgewählt (vgl. 2(e)). Danach werden alle Seitenflächen gesucht, die von diesem Punkt aus gesehen werden und an die aktuelle Seitenfläche angrenzen.

Von diesen zusammenhängenden Seitenflächen werden nun alle Kanten gesucht, die am Rand liegen. Anders gesagt: All jene Kanten, die in der Sammlung aller Kanten dieser zusammenhängenden Seitenflächen genau einmal vorkommen.

All diese Kanten bilden jeweils mit Punkt P1 eine neue Seitenfläche der konvexen Hülle.

Nun müssen noch die zugewiesenen Punkte von Seitenfläche S1 den neuen Seitenflächen zugewiesen werden, falls sie sich oberhalb einer dieser Flächen befinden.

Als letzter Schritt werden die alten zusammenhängenden Seitenflächen von der Hülle entfernt und die neu generierten Seitenflächen sowohl der Hülle, als auch der Liste über die gerade iteriert wird, hinzugefügt.

Nach dem Ende der Iterationsphase ist die konvexe Hülle über allen

zufällig gesetzten Punkten generiert(vgl. 2(g)) und damit auch das Random Lattice Polytope fertig gestellt.

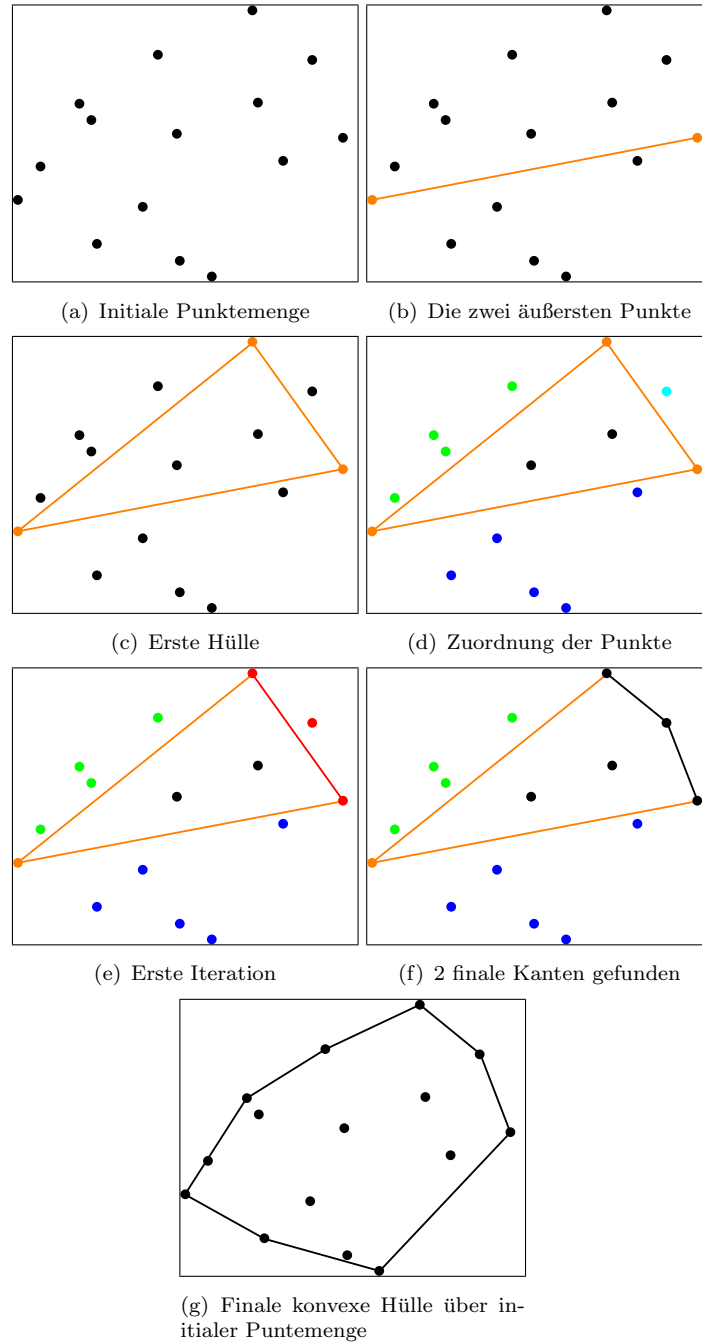


Abbildung 2: Quick Hull Algorithmus in 2D

1.3.2 Extrusionskörper

Bei der Erzeugung eines Extrusionskörpers werden zuerst auf einer der Seitenflächen des Würfels zufällig Punkte ausgewählt. Aus diesen Punkten wird die konvexe Hülle berechnet und das entstandene Polygon zur gegenüberliegenden Würfelseite hin extrudiert. Um die Körper einfach zu halten wird garantiert, dass die konvexe Hülle genau ein Drei- oder Viereck ist.

Dieses Verfahren wird noch zwei Mal angewendet, wobei die Hälfte der erzeugten Extrusionskörper um die Y-Achse um 90 Grad gedreht werden. Abschließend wird mit Hilfe der UnBBoolean Bibliothek die Vereinigung der Körper erzeugt. Die voreingestellte Anzahl von drei vereinigten Extrusionskörpern wurde als gutes Mittelmaß gewählt, vor allem da dadurch sowohl komplexe Körper, als auch immer wieder einfachere Körper entstehen können, wenn sich die Körper überlagern.

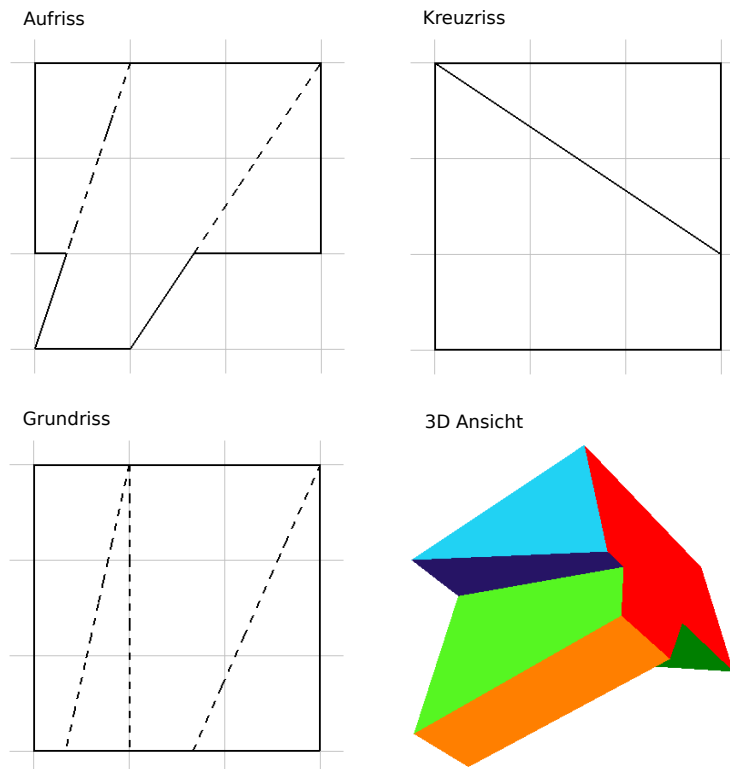


Abbildung 3: Beispiel für einen generierten Extrusionskörper

In Abbildung 3 ist als Beispiel den Extrusionskörper, der mit Startwert -2142940721 generiert wurde dargestellt.

1.3.3 Generische Algorithmen

Bei der Verwendung eines generischen Algorithmus wird zuerst eine bestimmte Anzahl an Objekten erzeugt um die Startpopulation zu bilden. Dabei kann in den Einstellungen festgelegt werden, wie viele Objekte welcher Generierungsart (Random Lattice Polytopes oder Extrusionskörper) erzeugt werden sollen.

Nach der Erzeugung der Startpopulation muss von allen Objekten die Fitness ausgerechnet werden, d.h. wie gut sie den Erwartungen entsprechen. Die Zielfitness wird in den Konfigurationen festgelegt. Zur Berechnung der Fitness werden die Anzahl der Punkte, der Kanten und der Flächen, sowie die Anzahl der Punkte, die nicht auf dem Würfelgitter liegen (i.F. Nichtgitterpunkte genannt), der Kanten in allgemeiner Lage (i.F. allgemeine Kante) und der Flächen in allgemeiner Lage (i.F. allgemeine Fläche) mit der folgenden Formel verwendet:

$$\begin{aligned} \text{Fitness} = & \text{Punkte} + \text{Kanten} + \text{Flächen} + \text{Nichtgitterpunkte} * 5 + \\ & + \text{allgemeineKanten} * 3 + \text{allgemeineFlächen} * 3 \end{aligned} \quad (1)$$

Die Gewichtung und Auswahl der einzelnen Parameter wurde anhand verschiedener generierter Objekte und ihrem erwartetem Schwierigkeitsgrad gewählt, aber bietet noch Raum für weiterführende Forschung.

Nachdem die Fitness der Population bestimmt wurde, werden sie aufsteigend der Abweichung zur Zielfitness sortiert. Solange nun nicht ein optimales Objekt (innerhalb einer definierten Toleranzgrenze) gefunden wurde und eine konfigurierte Anzahl an Iterationen nicht erreicht wurde, wird versucht das Ergebnis zu verbessern.

Dazu werden im ersten Schritt jeweils zwei Objekte zufällig ausgewählt und miteinander kombiniert. Die Kombination ist im Prototyp ein simpler Schnitt durch die X-Z-Ebene und die Vereinigung der oberen Hälfte vom ersten Objekt mit der unteren Hälfte des Zweiten. In der Abbildung 4 sieht man solch eine Kombination. Diese Möglichkeit zur Erzeugung von neuen Objekten kann auch ohne den Generischen Algorithmus direkt im Menü ausgewählt werden.

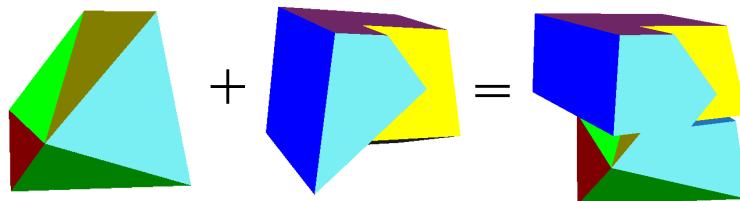


Abbildung 4: Beispiel für Kombination

Neben der Kombination kann auch eine Mutation der Population durchgeführt werden, wobei wiederum zufällig Objekte ausgewählt werden. In Folge werden deren Punkte innerhalb der Würfelgittergrenzen mit einer geringen Wahrscheinlichkeit um einen Gitterpunkt verschoben. Da diese Mutation invalide Objekte erzeugen kann, muss anschließend ihre Validität überprüft werden.

Anschließend wird die Fitness der neu generierten Objekte bestimmt und die Objekte werden nun zusammen mit der alten Population sortiert. Von dieser neuen Population werden nur die besten Objekte behalten, so dass wieder die ursprüngliche Anzahl der Population erreicht wird.

Am Ende dieser Iterationen wird das Würfelgitterobjekt mit der besten Fitness zurückgeliefert. In Abbildung 5 sieht man das Ergebnis eines Durchlaufs mit Zielfitness 100 und Toleranz 5.

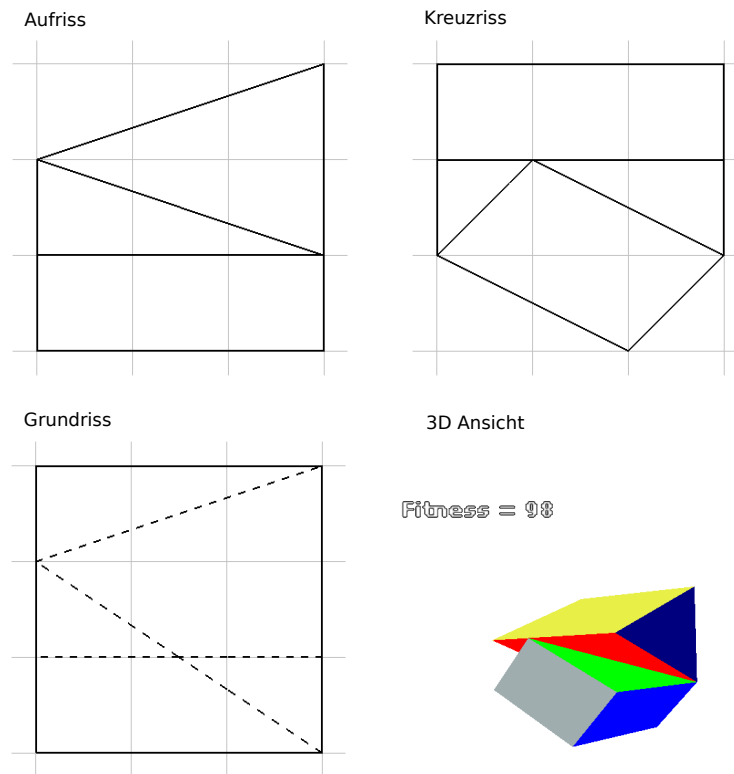


Abbildung 5: Beispiel für den Generischen Algorithmus mit Zielfitness 100

1.4 Ein- und Ausgabe

Zur Bedienung des Programmes wurde eine Kombination aus Java-Swing¹⁰ Elementen und Java3D erstellt. Im Folgenden werden die einzelnen grafischen Teile des Prototyps kurz präsentiert.

1.4.1 Hauptfenster

Im Hauptfenster ist die klassische Menüzeile am oberen Programmrand und die vier Ansichten darunter zu sehen. Am unteren Rand des Fensters befindet sich ein Knopf durch den eine neue Generierung des Beispielobjektes mit den aktuellen Einstellungen ausgelöst werden kann.

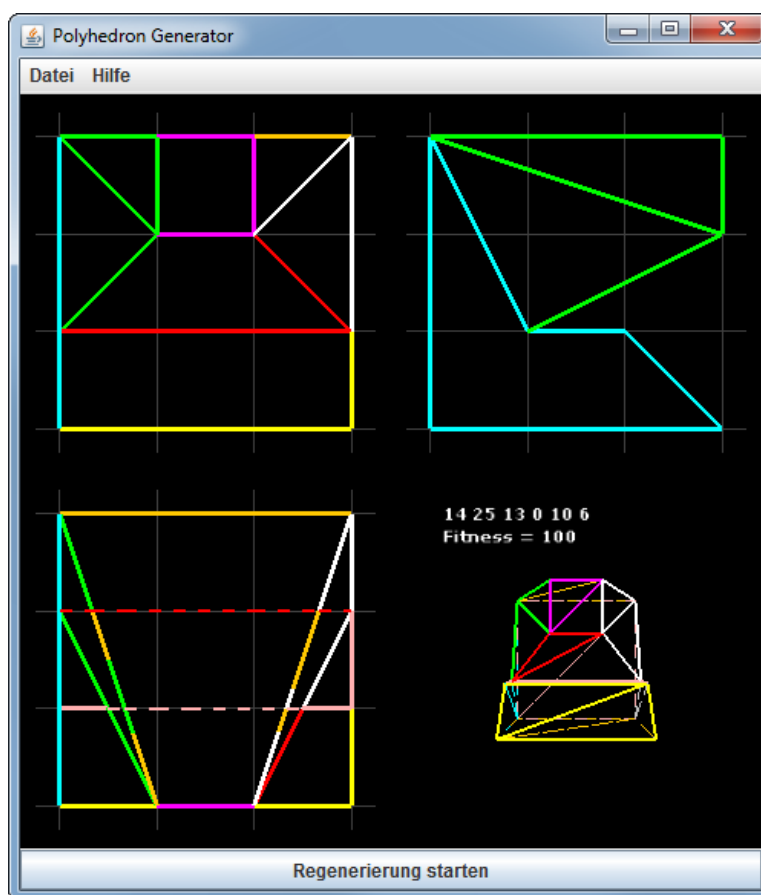


Abbildung 6: Programmfenster mit den drei Rissen und der 3D Ansicht

Die 2D Ansichten sind Projektionen in europäischer Darstellung und werden als Dreitafelprojektion (links oben Aufriss, rechts oben Kreuzriss

¹⁰<http://docs.oracle.com/javase/tutorial/uiswing/index.html>

und links unten Grundriss) angeordnet. Rechts unten befindet sich eine 3D Ausgabe des Objektes, die durch die Maus drehbar ist. Dazu ist ein Klick auf das Objekt mit der linken Maustaste notwendig. Anschließend kann das Objekt dann, ohne die Maustaste loszulassen, frei gedreht werden.

1.4.2 Menü

Über das Menü können die verschiedenen Aktionen des Programmes aufgerufen werden.



Abbildung 7: Hauptmenü

Unter dem Menüpunkt *Neu* können die verschiedenen Arten zur Generierung aufgerufen werden. Dabei sind die ersten drei Auswahlmöglichkeiten hartcodierte Beispielobjekte während die anderen Möglichkeiten eine neue Generierung auslösen. Je nach ausgewählter Generierungsart werden die hinterlegten Einstellungen des Programmes verwendet.

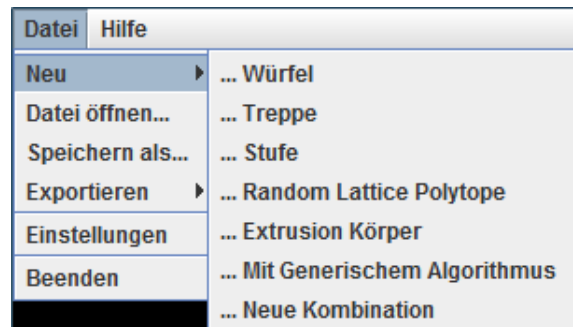


Abbildung 8: Auswahl von Beispielen und den verschiedenen Generierungsarten

1.4.3 Einstellungen

Über den Menüpunkt *Einstellungen* können über drei Tabs aufgeteilt die meisten Einstellungen des Programmes gewählt werden.

Die Einstellungen werden in einer Datei *config.xml* - im selben Ordner wie sich das Programm befindet - gespeichert.

In Abbildung 9 sind die 5 Einstellungen, die zur Repräsentation der Würfelgitterobjekte getroffen werden können, dargestellt.

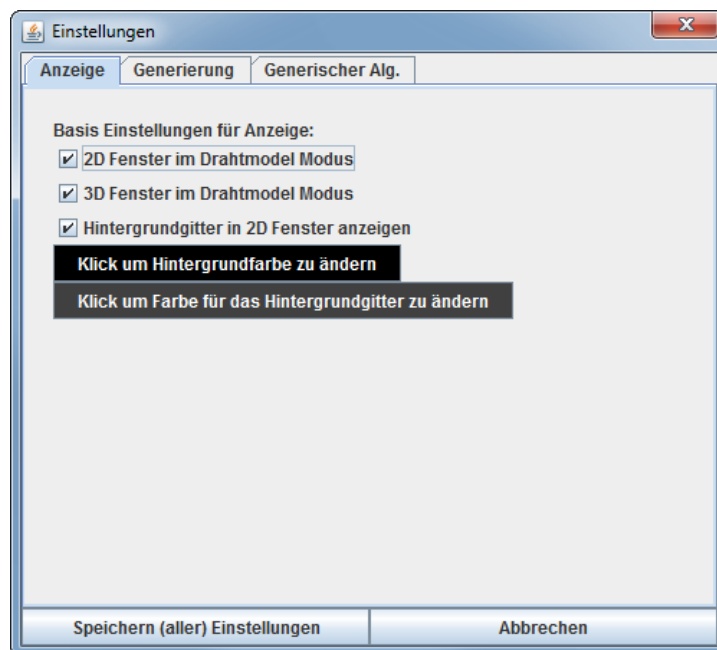


Abbildung 9: Einstellungen zur graphischen Ausgabe

Im zweiten Tab befinden sich die Einstellungen zur Generierung von Random Lattice Polytops und Extrusionskörpern.

Die Anzahl der Würfelgitterpunkte hat einen großen Einfluss auf die Komplexität der generierten Objekte. Die Einstellung gibt die Anzahl der Punkte pro Dimension vor. Das zulässige Minimum liegt dabei bei 2 Punkten. In Abbildung 6 wurden 3 Punkte gewählt, in Abbildung 5 waren es 4 Punkte.

Die Dichte der Punkte im Random Lattice Polytope bestimmt mit welcher Wahrscheinlichkeit jeder Punkt beim Random Lattice Polytope Algorithmus gesetzt wird. Diese Einstellung sollte in Berücksichtigung mit der Anzahl der Würfelgitterpunkte getroffen werden, da bei mehr Punkten eine geringere Wahrscheinlichkeit geeignete Ergebnisse liefert.

Die zwei Seeds bestimmen den Startpunkt für die Erzeugung der Pseudo-Zufallszahlen, die für die Generierung der Random Lattice Polytops und Extrusionskörpern verwendet werden. Der erste Seed wird dabei immer verwendet, der zweite nur wenn eine Kombination der Objekte gewünscht ist. Der Seed 0 bedeutet die Wahl eines zufälligen Seeds.

Bei der Wahl der Kombination zweier Polytopen können die Typen festgelegt werden. Dabei können alle möglichen Kombinationen zwischen Random Lattice Polytops und Extrusionskörpern gewählt werden.

Die Schnitthöhe bei Kombination gibt die Höhe (bzgl. der Y-Achse) an, an der die zwei generierten Polytopen geschnitten werden. Diese Höhe muss zwischen 0 und der Anzahl der Würfelgitterpunkte liegen. Die Eingabe einer Dezimalzahl ist möglich, aber nicht empfohlen

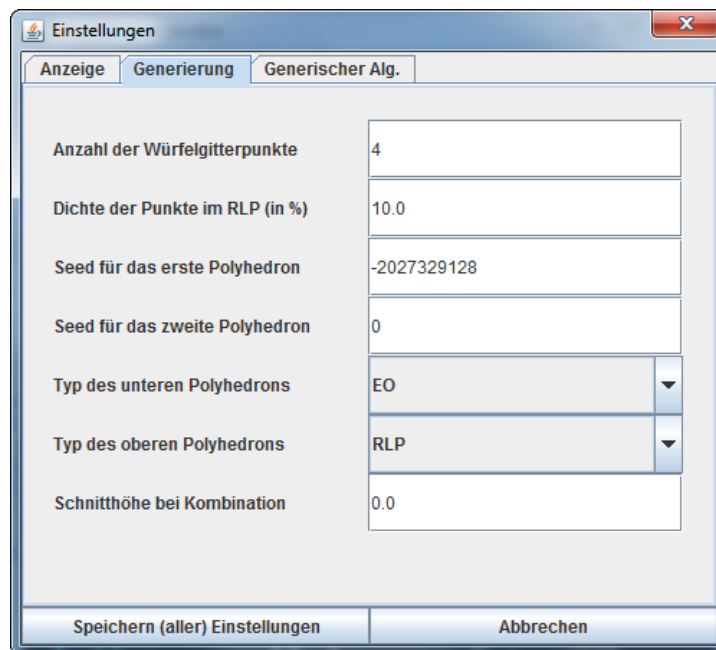


Abbildung 10: Einstellungen für die Generierung der Beispiele

Über den dritten Tab (siehe Abbildung 11) können schließlich die Einstellungen bezüglich des generischen Algorithmus gesetzt werden.

Die optimale Fitness gibt das Ziel des Schwierigkeitsgrades vor. Sobald ein Objekt diese Fitness \pm der gewählten Fitness Toleranz erreicht hat, wird es als gesuchtes Objekt ausgegeben.

Die verschiedenen Populationsgrößen legen die Anzahl der einzelnen generierten Objekte in den verschiedenen Schritten des generischen Algorithmus fest.

mus fest. Je größer die Anzahl, desto größer die Vielfalt, allerdings dauert die Generierung auch entsprechend länger.

Die maximale Anzahl an Iterationen gibt eine Obergrenze vor, nach der der Algorithmus das bisher beste Ergebnis zurückliefert und nicht mehr weiter ausgeführt wird.

Wie bei den allgemeinen Einstellungen zur Generierung muss auch für den generischen Algorithmus die Anzahl der Würfelgitterpunkte pro Dimension und die Dichte der Punkte für Random Lattice Polytopes festgelegt werden. Eine ungünstige Wahl dieser Einstellungen kann verhindern, dass der Algorithmus ein Objekt mit der gewünschten Fitness erzeugen kann.

In der letzten Einstellung kann noch ausgewählt werden, ob Extrusionskörper oder Random Lattice Polytopes als Grundlage für den generischen Algorithmus verwendet werden. Will man beide Arten als Eingabe, kann man durch gedrückt halten der Strg-Taste beide Zeilen auswählen.

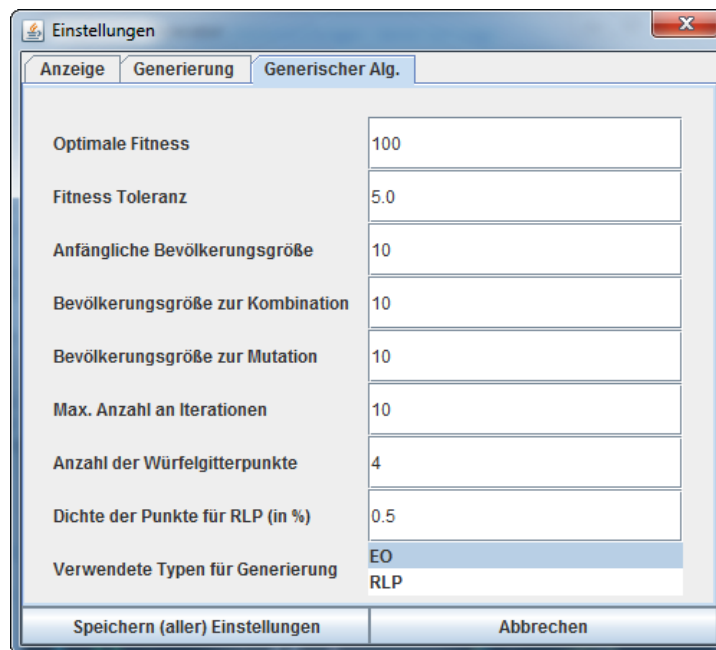


Abbildung 11: Einstellungen für den generischen Algorithmus

Zusätzlich zu den Einstellungen in diesem Menü gibt es noch folgende Einstellungen, die nur direkt über die *config.xml* verändert werden können:

- localeLanguage, localeCountry und localeVariant:
Angabe eines ISO 3166 codes für die Sprachauswahl. Siehe auch 1.5.1.

- `genericAlgorithmWeightPoints`, `genericAlgorithmWeightEdges`, `genericAlgorithmWeightFaces`, `genericAlgorithmWeightGeneralEdges`, `genericAlgorithmWeightNonGridPoints`, `genericAlgorithmWeightGeneralFaces`: Die Gewichtung der einzelnen Parameter der Fitnessfunktion kann hier eingestellt werden. Durch setzen der Gewichtung auf 0 kann ein Parameter auch komplett bei der Berechnung ignoriert werden.
- `eoNumberOfPolyhedra`: Die Anzahl der generierten Extrusionskörper, deren Vereinigung bei der Verwendung des entsprechenden Algorithmus ausgegeben wird.
- `gaEoNumberOfPolyhedra`: Die Anzahl der erstellten Extrusionskörper, deren Vereinigung für die Grundpopulation des generischen Algorithmus verwendet wird.
- `gaUseCombination`: Angabe ob beim generischen Algorithmus Kombinationen der bestehenden Population erzeugt werden.
- `gaUseMutation`: Angabe ob beim generischen Algorithmus die Mutationen durchgeführt werden.
- `genericAlgorithmSeed`: Eingabe des Startwertes für den generischen Algorithmus

1.4.4 Exportieren der Objekte

Die generierten Objekte können sowohl als pdf, als auch als png-Bilddateien exportiert werden. Beim Export als pdf ist es möglich (siehe Abbildung 12) sich zusätzlich noch die berechnete Schwierigkeit ausgeben lassen und zu bestimmen welche Risse angezeigt werden. Die Schwierigkeiten sind momentan entsprechend des Wertes der Fitnessfunktion wie folgt eingestuft: < 60 leicht, < 120 mittel und ≥ 120 schwierig.

Außerdem gibt es die Möglichkeit die generierten Objekte durch Auswahl der Menüpunkte *Speichern als...* und *Datei öffnen...* in Form einer XML-Datei (mit eigener Endung `.polyxml`) zu speichern und wieder in das Programm zu laden. Auf diesem Wege können Objekte auch noch nachbearbeitet werden, was allerdings händisch - ohne weitere Unterstützung durch das Programm - erfolgen muss.



Abbildung 12: Dialog beim Export als pdf

1.5 Allgemeine Features

Abschließend werden nun noch verschiedene allgemeine Features des Programmes vorgestellt.

1.5.1 Internationalisierung

Um das Programm einer größeren Anzahl an Interessierten zur Verfügung stellen zu können wurde eine einfache Internationalisierung eingebaut. So kann durch Eingabe eines ISO 3166 Codes die Programmsprache gewählt werden. Der Code *de-AT-7* steht z.B. für Tirol. Die Sprachen *de* und *en* sind im Programm festimplementiert und können über diese Einstellung ausgewählt werden. Als Standardsprache ist Deutsch eingestellt.

Neue Sprachdateien werden dynamisch ins Programm eingebunden, in dem sie in den Ordner *at.ac.uibk.math.polytope.i18n* des Java-Archivs kopiert werden. Die erstellte Sprachdatei muss dabei mit dem Schema „Strings-*_SprachCode_LandCode_RegionCode.properties*“ abgespeichert werden, wobei die letzten zwei Codes optional sind. Als Grundlage wird am besten eine der bestehenden Dateien verwendet.

1.5.2 Logging

Um die Programmentwicklung zu unterstützen und eine spätere Fehlersuche und Nachverfolgung der internen Abläufe zu erleichtern wurde das logging Framework Logback verwendet. In der Datei *log4j2.xml* kann sowohl für die Konsolenausgabe, als auch für die Ausgabe in eigenen Log-Dateien festgelegt werden, welche Details ausgegeben werden. Für den späteren Betrieb sollte das Log-Level *WARN* verwendet werden.

Literatur

- [1] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The Quickhull Algorithm for Convex Hulls. *ACM Trans. Math. Softw.*, 22(4):469–483, 1996.

Abbildungsverzeichnis

1	Beispiel für einen Random Lattice Polytope	4
2	Quick Hull Algorithmus in 2D	6
3	Beispiel für einen generierten Extrusionskörper	7
4	Beispiel für Kombination	8
5	Beispiel für den Generischen Algorithmus mit Zielfitness 100 .	9
6	Programmfenster mit den drei Rissen und der 3D Ansicht . .	10
7	Hauptmenü	11
8	Auswahl von Beispielen und den verschiedenen Generierungs- arten	11
9	Einstellungen zur graphischen Ausgabe	12
10	Einstellungen für die Generierung der Beispiele	13
11	Einstellungen für den generischen Algorithmus	14
12	Dialog beim Export als pdf	16