



Diplomarbeit

Multikriterielle Grundrissgenerierung in der Anwendung an Schulgebäuden Nutzen, Grenzen und Potenziale

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Diplom-Ingenieurs unter der
Leitung von

Privatdozent Dipl.Ing. Dr. Gabriel Wurzer
E272

Institut für EDV-gestützte Methoden in Architektur und Raumplanung

eingereicht an der Technischen Universität Wien

Fakultät für Architektur und Raumplanung

von

Bernadette Arendt
Matr. Nr. 0525752

Wien, am

Bernadette Arendt

Inhalt

| | |
|---|----|
| Abstract | 3 |
| 1. Einleitung..... | 6 |
| 2. Grundlagen der Problemstellung und vorhandene Lösungsansätze | 8 |
| 3. Konzept..... | 15 |
| 4. Implementierung..... | 22 |
| 4.1 Benutzeroberfläche | 22 |
| 4.2 - Prozessablauf | 25 |
| 4.2 Datenübertragung / Daten-Import..... | 26 |
| 4.3 Grundeigenschaften und Aufbau der Flächen | 27 |
| 4.4 Gebäudetypologien | 28 |
| 4.5 Schnittflächenerkennung | 29 |
| 4.6 Nachbarschaftsanordnung | 29 |
| 4.7 Anordnung im Grundriss | 30 |
| 4.8 Interaktion..... | 31 |
| 4.9 Evaluierung..... | 31 |
| 4.10 Export | 32 |
| 5. Diskussion..... | 33 |
| 5.1 Charakteristik der Testlandschaft..... | 33 |
| 5.2 Analyse des gesamten Lösungssatzes | 37 |
| 5.3 Layoutbandbreite | 52 |
| 5.5 Vergleich zum Einfluss von Interaktion und Randomisierung..... | 55 |
| 5.6 Maßstabswechsel und Auswirkungen..... | 56 |
| 5.7 Auswirkungen der Evaluierung..... | 60 |
| 5.8 Anmerkungen zur Schnittstelle | 62 |
| 6. Zusammenfassung und Ausblick | 63 |
| 7. Schlussbetrachtung | 67 |
| Bibliographie | 68 |
| Abbildungsverzeichnis..... | 69 |
| Tabellenverzeichnis | 71 |
| Quellcode für NetLogo | 72 |

Abstract - DEUTSCH

Im Zuge der Diplomarbeit wird in Anlehnung an den abstrakt gehaltene Algorithmus von Elezkurtaĵ und Franck zur generischen Grundrissentwicklung (2002) und den umfassend dokumentierten Layoutsolver "Dichte Packung" von Reinhard König ein eigenes Programm entwickelt und um konkretisierende Elemente erweitert, um diese an einem praktischen Fall testen zu können. Die daraus ableitbaren Nutzen und Grenzen sowie entstehende Vorteile und Einschränkungen werden analysiert, woraus eine Basis für Optimierung und Anforderungen zukünftiger Algorithmen entstehen kann.

Anhand computergenerierter Außenhüllen von Schulen, deren genauer Entstehungsprozess Thema einer separaten Diplomarbeit (Daniel Elsner, "SimSchool - Optimierung der Abläufe in Schulbauten") ist, wird das Programm zur Layoutgenerierung eingesetzt. Eine Datenschnittstelle wird errichtet, die sämtliche nötigen Parameter zu Außengrenzen, Größen und deren -verhältnissen sowie Funktionsbeziehungen übergibt. Das Programm "entwirft" daraufhin seinen "idealen" Grundriss in einem zellenbasierten System, wobei es erlaubt ist, manuell (und '*sinnvoll*' lt. Elezkurtaĵ und Franck) in den Prozess einzugreifen und das Ergebnis zu verändern bzw. damit verschiedene Ergebnisse zu erzeugen. Die entstehenden Grundrissvarianten werden mittels "Agenten", in diesem Fall vereinfachte simulierte SchülerInnen überprüft und miteinander verglichen.

*„Das Ziel der Analyse eines Problems ist eine Beschreibung, die so genau wird, dass sie die Lösung enthält.“*¹ In diesem Sinne soll im Zuge der Arbeit auf folgende Fragen eingegangen werden: Welche Parameterdichte bringt die spannendsten Ergebnisse? Welchen Einfluss hat der interaktive Ansatz der Anwendung, der es erlaubt in den "Entwurfsprozess" des Computers manuell einzugreifen und damit das Ergebnis mitzusteuern? Gibt es ablesbare Grundvoraussetzungen, die zu guten oder schlechten Ergebnissen führen? Welches Potenzial kann dem Algorithmus aus den Erfahrungen zugesprochen werden?

¹ Georg Franck und Tomor Elezkurtaĵ, 2011, Design Methods und computergestützte Grundrissentwicklung, verfügbar unter: http://www.iemar.tuwien.ac.at/wp-content/uploads/2011/12/design_methods.pdf [zuletzt aufgerufen am 21.2.2016]

Abstract - ENGLISH

The core of the master thesis is an independently developed cell-based program that is referring to the algorithm of Elezkurtaj and Frank for abstract generic floorplanning (2002) and the extensive, profound Layoutsolver "Packing" by Reinhard König. Besides the alteration from a vectorized world to a cell-based world the algorithm is extended with functions for practical use in order to test the algorithm for viability. The derived benefits and limits are analysed to create a basis for optimizations and requirements of further algorithms.

The practical example used is a computer-generated building complex of a school which is developed in another master thesis of the institute (Daniel Elsner, "SimSchool - Optimierung der Abläufe in Schulbauten"). An interface is developed which processes the necessary parameters such as borders, sizes and relations of size and functions. The program "designs" an "ideal" floorplan in a cell-based world. It is possible to interact manually (and 'reasonable' according to Elezkurtaj and Franck) during the design process in order to influence the result or to get a variety of results. Using simulated students and teachers, so called "agents", the resulting floorplans are tested and therefore comparable.

"The goal of a problems' analysis is a description so accurate it already contains the solution."² Along this lines the thesis regards the following questions: Which density of parameters bares the most surprising results? What is the impact of the interactive part of the application? Can parameters be derived that grant the quality of the results? Which potentials of the algorithm are obtained out of the approach?

^{2 2} Georg Franck und Tomor Elezkurtaj, 2011, Design Methods und computergestützte Grundrissentwicklung, available at: http://www.iemar.tuwien.ac.at/wp-content/uploads/2011/12/design_methods.pdf [accessed 21.2.2016]

Hinweis

Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung männlicher und weiblicher Sprachformen verzichtet. Sämtliche Personenbezeichnungen gelten selbstverständlich gleichermaßen für beiderlei Geschlecht.

1. Einleitung

Computergenerierte Layouts sind seit Jahrzehnten ein wiederkehrend auftretendes Forschungsgebiet. Einer der vielversprechendsten und interessantesten Algorithmen zur computerunterstützten Grundrissgenerierung entstand an der TU Wien im Zuge der Dissertation von Tomor Elezkurtaj und wurde 2010 von Reinhard König an der Bauhaus-Universität Weimar hinsichtlich der verwendeten Prozesse ausführlich dokumentiert und in der Grundfunktionalität nachgebildet.

Die Leistungsfähigkeit der Basisfunktionen kann - kontextgerecht verwendet - als belegt angesehen werden. Spannend ist die Frage nach erweiternden Anforderungen und die Anwendung der verfügbaren Tools in neuem Kontext.

Hauptziel dieser Arbeit ist, ein Set aus multikriteriell erzeugten Layoutvarianten hinsichtlich Verwertbarkeit und Einsatzmöglichkeit generell zu analysieren sowie untereinander zu vergleichen. Dafür sind folgende Schritte für die Arbeit maßgebend:

- Die Definition der Kriterien, nach denen die Layouts erzeugt werden sollen, ergibt sich aus der Analyse der Forschungen aus den letzten fünfzig Jahren (siehe Kapitel 2). Hierbei werden besonders reizvolle Erkenntnisse aus der Analyse aufgegriffen und angepasst, sowie das in diesem Kontext noch nicht behandelte Element der Gebäudetypologie als Neuheit eingeführt (siehe Kapitel 3 - Konzept)
- Ein komplexes Testbeispiel, welches sich von den in der Analyse gebrauchten Beispielen sowohl in Größenverhältnis als auch Verwendungszweck unterscheidet, wird von einer parallel entstehenden Diplomarbeit (SimSchool - Optimierung der Abläufe in Schulbauten von Daniel Elsner) übernommen. *"Der erhebliche Sanierungsbedarf, der dadurch entsteht, dass die Gebäude des Neubaubooms der sechziger und siebziger Jahre des letzten Jahrhunderts nun in die Jahre kommen, wird aufgrund der schwierigen kommunalen Finanzverhältnisse vielerorts nur unzureichend bewältigt. Renovierungen und Umbauten werden aufgeschoben, solange es eben noch geht. Schülerinnen und Schüler sind ebenso wie die Lehrerinnen und Lehrer dadurch häufig räumlichen Umgebungen ausgesetzt, die an der Grenze des Zumutbaren, und manchmal auch weit jenseits davon liegen."*³ Dieser insbesondere auch in Wien zutreffende Zustand in Kombination mit den vielen neuen Ansätzen zur Strukturierung der Bildungseinrichtungen in Österreich in den

³ Riklef Rambow, Zur Bedeutung der Architektur für den Lernort Schule, 2006, leicht verändert in: Starke Eltern - Starke Kinder, Jahresheft 2006, Deutscher Kinderschutzbund Seite 88-90, verfügbar unter <http://www.tu-cottbus.de/theoriederarchitektur/Lehrstuhl/deu/rambow/SESK.pdf> [zuletzt aufgerufen am: 09.03.2016]

letzten Jahren machen das Thema Schulbau zu einem besonders geeigneten Thema, da ein hohes Konzeptentwicklungs- und Planungsaufkommen für die kommenden Jahre abzusehen ist. (siehe Kapitel 5.1 - Diskussion - Aufbau der Testlandschaft)

- Um diese Aspekte zielführend zusammenführen zu können und eine rasche kontinuierliche Layoutgenerierung erzeugen zu können, muss eine Schnittstelle für den Datenaustausch entwickelt werden, die dem Programmablauf vorangestellt ist. Hierbei ist ein Augenmerk darauf zu legen, welche Parameter in welcher Ausprägung vom Programm weiterverarbeitet werden können oder müssen. Welche Angaben müssen immer definiert sein, welche können auch dem Zufall überlassen werden? (siehe Kapitel 4 - Implementierung)
- Nach Erzeugung eines Layoutsets, bestehend aus 126 Einzellösungen, können objektive Aussagen getroffen werden. Ableitbare Tendenzen und daraus entstehende Fragestellungen werden in tiefergehenden Analysen mit weiteren Lösungserzeugungen aufgearbeitet. (siehe ab Kapitel 5.2 - Diskussion)

unvollständigem Problemwissen zu plausiblen und brauchbaren Ergebnissen führen. Eine breite Lösungsvarianz wird einem Optimum vorgezogen. Die Funktionsweise kann dem Bereich der zellulären Automaten zugeordnet werden. Der Nutzer gibt Bereiche an, die eine Angabe zu Größe und gereihten präferierten Nachbarbereichen erhalten. Das Programm arbeitet diese Bereiche wie folgt ab: es weist Zellen Bereiche zu, bis deren Größe erreicht ist und fährt anschließend mit einem präferierten Bereich fort, bis schlussendlich ein gesamtes Geschoß "gewachsen" ist. Ein Set aus vier hierarchisch abzuarbeitenden Regeln zur Zellbelegung, von häufigster Zellnachbarschaft über Distanz zum Zentrum und leeren Zellpositionen bis zu einer Regel, die dem Zufall in den Schoß spielt, reicht dem Programm aus um unverhältnismäßige Proportionen abzufangen und auf den ersten Blick sehr konkrete Grundrisse zu erzeugen (Abb. 2). Bewusst betrachten diese Regeln nur die lokale unmittelbare Situation der Zellnachbarn, um die den 70er Jahren entsprechende Rechenkapazität niedrig und unabhängig von Gebäudegröße zu halten. Im Ausblick erstreben die Autoren Interaktivität und eine tiefere Ausreizung der Zellbelegungsregeln.

Bemerkenswert dabei ist, dass DOMINO im Planungsbüro von Welton Becket and Associates unter anderem für die Krankenhausplanung im Einsatz war. Offen bleibt jedoch die Frage, ob der Mangel an Information und Publikationen zu den damit verbundenen Erfahrungen zu bedauern oder Antwort genug ist.

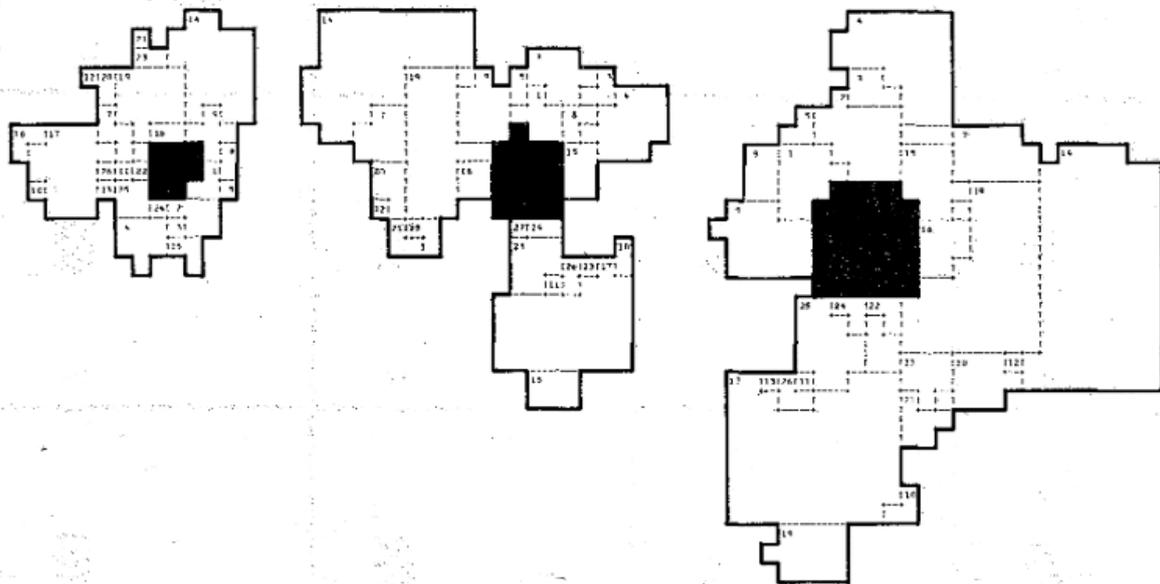


Abbildung 2 - Ergebnisse von DOMINO in unterschiedlichen Maßstäben⁵

Einen auf physikalischen Gesetzen basierenden, interaktiven Ansatz wählen Scott A. Arvin und

⁵ William J. Mitchell, Robert L. Dillon, 1972, A Polyomino Assembly Procedure for Architectural Floor Planning, aus: EDRA 3: Research and Practice: Proceedings of the edra3/ar8 Conference (1972), Seite 23-5-10.

Donald H. House. In ihrem Lösungsansatz werden die erzeugten Flächen über gegenseitige Kraft- einwirkung angeordnet. Die Möglichkeit der Interaktion soll gerade durch die für den Nutzer intuitiv verständlichen Gesetze von Anziehungskraft und Masse gefördert und interessant gestaltet werden. Die einzelnen Formen haben jeweils eine Masse, deren Mittelpunkte über Beziehungsverhältnisse miteinander verbunden sind. Diese Adjazenzen verhalten sich im Programmhintergrund wie Federn mit integriertem Dämpfer, woraus sich Anziehungskräfte ergeben, die je nach Distanz zueinander die Geschwindigkeit bzw. Bewegungsdringlichkeit beeinflussen. Wird eine Form vom Nutzer verschoben, greifen die neu wirkenden Kräfte umgehend und das Gesamtergebnis wird dementsprechend beeinflusst und ummodelliert. Das schnell erklärte Modell führt zu einer Bandbreite unterschiedlicher und mit kleinen Eingriffen wandelbarer Ergebnisse. Auch hier ist der Schwerpunkt auf das Entstehen von Design an sich gesetzt und nicht auf das "beste" Design (Abb.3).

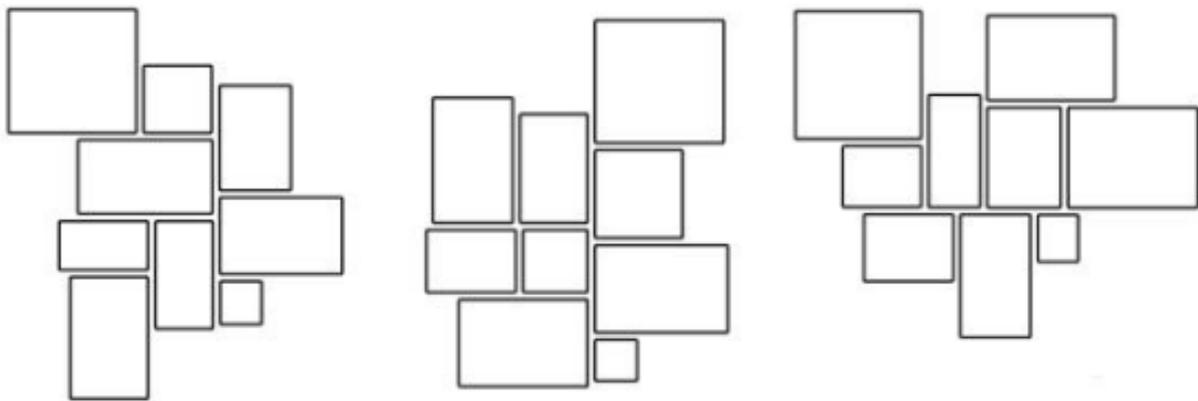


Abbildung 3 - Ergebnisse von Physics in der geometrischen Ebene⁶

Während sich bei DOMINO die Reihenfolge von "Geometrie einer Fläche" und "Topologie der nächsten Fläche" blockweise abwechseln und die Arbeit von Arvin und House die Parallelität der beiden Kernprobleme betont, findet man bei ARCHiPLAN von Benachir Medjdoub und Bernad Yannou einen Prozess, der erst die Topologie in weiter Bandbreite für den Nutzer erarbeitet, um anschließend die topologischen Ergebnisse auf geometrischer Ebene zu optimieren. Für die Erstellung der möglichen Topologien (die ab einer umsetzbaren weiterführenden geometrischen Variante für gültig erklärt sind) werden Lagebeschränkungen für die Räume definiert und iterativ vom eingeschränktsten Raum beginnend die Topologie entwickelt (Abb. 4 - links). Weiterführend definiert der Nutzer ein oder mehrere messbare Optimierungskriterien. In dem dokumentierten Beispiel ist der Suchraum auf eine möglichst kleine Erschließungsfläche ausgelegt (Abb. 4 - rechts).

⁶ Scott A. Arvin und Donald H. House, 1999, Making Designs Come Alive: Using Physically Based Modeling Techniques in Space Layout Planning, aus: Computers in Building (1999), Seite 258



Abbildung 4 - Bandbreite möglicher Topologien für ein zweigeschoßiges Einfamilienhaus (links) und eine geometrisch optimale Lösung der Topologie 3 (rechts)⁷

Ein Branch-and-Bound Algorithmus ermittelt zu jeder Topologie aus dem, dem Programm zugrundeliegenden, Lösungsspektrum je ein global optimales geometrisches Ergebnis. Da die Beschränkungen auf sehr projektspezifische Anforderungen gesetzt werden, bedarf es zum erstmaligen Einsatz pro Vorhaben eine exakte topologische Analyse der Ansprüche. Wie mit sich widersprechenden Bedingungen umgegangen wird, ist leider nicht dokumentiert.

Die "Dichte Packung" von Reinhard König basiert auf Forschungen von Tomor Elez Kurtaj und Georg Franck sowie der Dissertation von Elez Kurtaj aus 2004. In beiden resultierenden Programmen wird eine Evolutionsstrategie eingesetzt, um eine überlappungsfreie Anordnung bestimmter erzeugter Flächen zu erreichen. Die graphische Repräsentation (Abb. 5) sowie eine Formel zur Berechnung der geringsten Schnittfläche sind bei Elez Kurtaj und Franck wie folgt definiert:

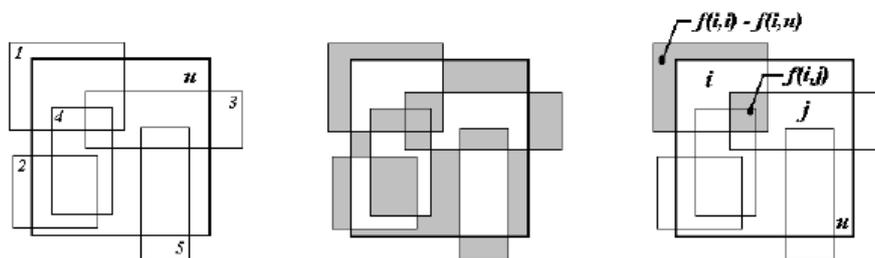


Abbildung 5 - Schnittflächenberechnung graphisch aufbereitet (oben)⁸

⁷ Benachir Medjdoub und Bernard Yannou, 2001, Dynamic Space ordering at a Topological Level in Space Planning, aus: Artificial Intelligence in Engineering Vol. 15 Issue 1, Januar 2001, Seite 12 + Seite 14

⁸ Tomor Elez Kurtaj und Georg Franck, 2001, Evolutionary Algorithms in Urban Planning, aus: Computergestützte Raumplanung CORP 2001 Geo.Multimedia.01, 2001, Seite 270, verfügbar unter http://www.corp.at/archive/CORP2001_ElezKurtaj_DO.pdf [zuletzt aufgerufen am: 12.02.2016]

$$Sg = \sum_{i=1}^{n-1} \sum_{j=i+1}^n (S[i] \cap S[j]) + \lambda \sum_{i=1}^n (S[i] \setminus S[u])^9$$

Um das definierte Ziel (Sg soll sich gegen 0 bewegen) zu erreichen, sind in Königs Arbeitspapier Nr. 4¹⁰ die umgesetzten Programmbestandteile umfangreich dokumentiert. Da sich Königs "Dichte Packung" an Elezkurtajs Lösungsstrategie orientiert, können zusammenfassende Aussagen über beide Programme getroffen werden. Sich überlappende Rechtecke stoßen sich ab, indem sie sich verschieben oder ihre Proportionen ändern. Die Lösungsfindung nutzt eine evolutionäre Strategie: von der ersten Anordnungsvariante P wird eine Kopie C angelegt, welche mutiert und mit P verglichen wird. Die jeweils bessere Lösung bringt nun erneut eine mutierte Kopie hervor, sodass sich das Ergebnis kontinuierlich verbessert. Mit der Einführung von Raumbeziehungen entsteht ein multikriterielles Optimierungsproblem, welches nur pareto-optimal gelöst werden kann da sich die Zielfunktionen widersprechen. Das heißt, letztendlich wird ein Zustand erreicht, bei dem jede Verbesserung der Lösung für Problem x eine Verschlechterung der Lösung für Problem y und umgekehrt bedeutet. Elezkurtajs Programm GAMES (Genetic algorithmic modeling and exploration system) birgt eine umfassende

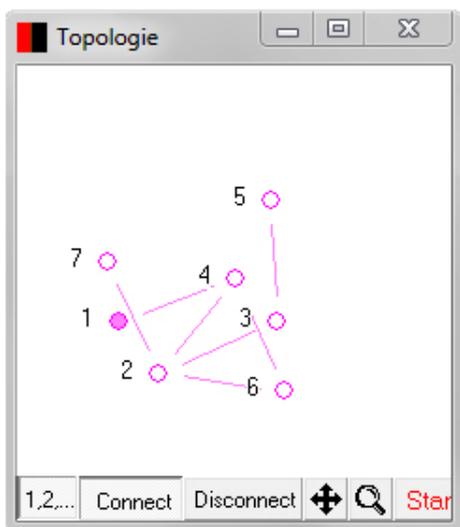


Abbildung 7 - Screenshot Topologie



Abbildung 6.1 - Screenshot Flächen, Abbildung 6.2 - Screenshot Wände, Abbildung 6.3 - Screenshot Polygone

⁹ siehe Fußnote 6

¹⁰ Reinhard König, 2011, Generierung von Grundriss-Layouts mittels Hybrider Evolutionsstrategie, aus: Arbeitspapiere Nr. 4, März 2011, Bauhaus Universität Weimar, verfügbar unter: http://infar.architektur.uni-weimar.de/service/drupal-infar/sites/default/files/upload/Publikationen/Arbeitspapiere/InfAR_04_Grundriss-Layouts.pdf [zuletzt aufgerufen am 10.01.2016]

Bandbreite an Funktionalitäten (Abb. 8). Die Adjazenzen können interaktiv bearbeitet (Abb. 7), Räume verschoben, deren Proportionen verändert oder deren Position verankert werden. Auch im laufenden Prozess können Räume hinzugefügt oder gelöscht werden. Der Darstellungsgrad bewegt sich von abstrakten bunten Kästchen (Abb. 6a) bis hin zu Mauern mit Verbindungsdurchbrüchen, die der aktuellen Topologie entsprechen (Abb. 6b). Reizvoll und bemerkenswert ist außerdem, dass die Orthogonalität aufgehoben werden kann (Abb. 6c).

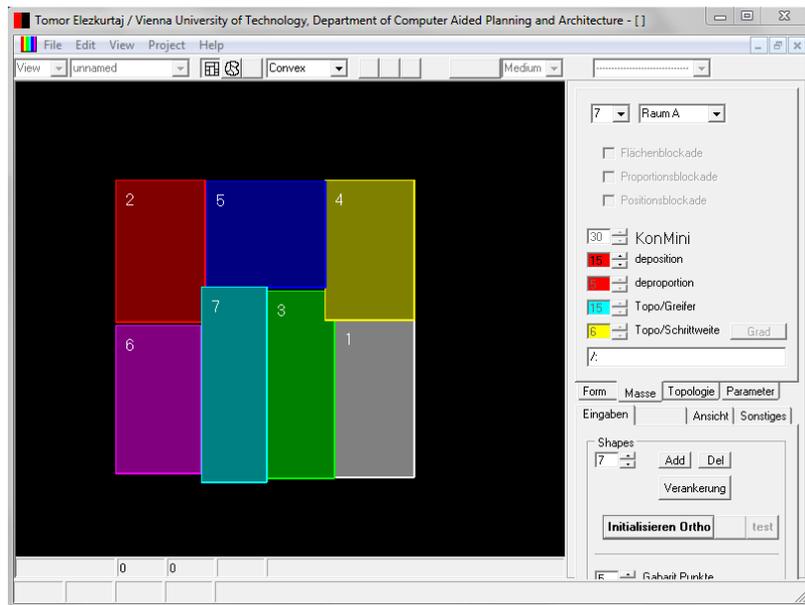


Abbildung 8 - Screenshot GAMES

Königs Umsetzung beschränkt sich darauf, eine frei wählbare Anzahl gleich großer, vom Nutzer verschiebbarer Rechtecke mit einer sternförmigen Typologie zu testen (Abb. 9). Sie besteht in der Möglichkeit und der publizierten Dokumentation, den Algorithmus mit Setzung von unterschiedlichen Einstellungen bezüglich der verwendeten und verglichenen Evolutionsstrategien hinlänglich zu testen.

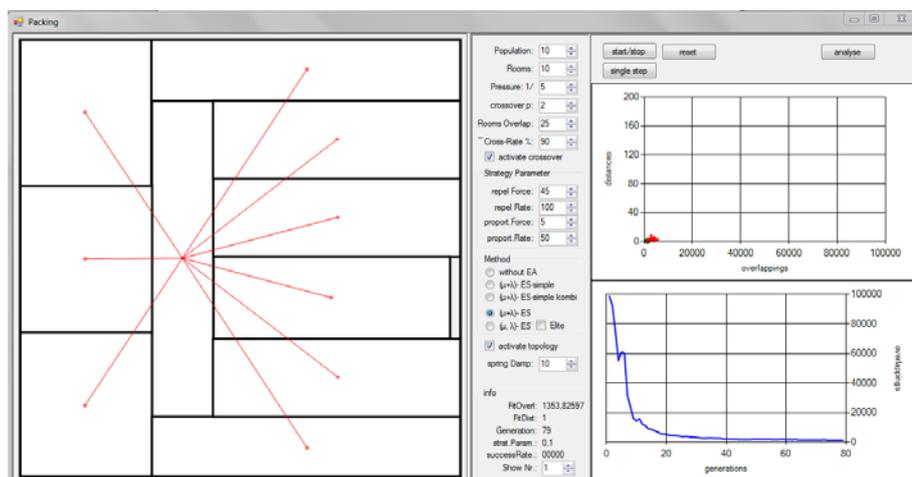


Abbildung 9 - Screenshot von Dichte Packing: Sterntopologie mit einer zentralen Fläche, 10 erstellte Flächen

Die Erfahrungen aus diesem Layoutsolver werden von Reinhard König in einer weiteren Publikation¹¹ anhand einfacher Testbeispiele an Kriterien wie Lösungsvarianz und Performance beurteilt.

Eine direkte Gegenüberstellung der unterschiedlichen Strategien der vorgestellten Programme kann der nachfolgenden Tabelle 1 entnommen werden. Ein Programm wird dann als interaktiv angesehen, wenn während des Programmablaufs vom Nutzer eingegriffen und das Layout in Echtzeit und Korrelation mit dem Algorithmus idealerweise per Maussteuerung verändert werden kann. Das zusätzliche Problemwissen bezeichnet Wissen, welches über Anzahl und Größe der Räume hinaus zwingend nötig ist.

| | DOMINO | Physics | ARCHiPLAN | GAMES | Dichte Packung |
|---|---|----------------|----------------------------|--|----------------------------------|
| Optimum | lokal | lokal | global | global (pareto-optimal) | |
| Topologie/ Geometrie | Geometrie vor Topologie | gleichwertig | Topologie vor Geometrie | gleichwertig | |
| Verfahren | iterativ | nicht dokum. | iterativ | Evolutionstrategie | |
| Interaktiv | nein | ja | nein | ja | ja |
| - Position | | ja | | ja | ja |
| - Proportion | | nein | | ja | indirekt über Position |
| - Topologie | | nein | | ja | nein (theoretisch möglich) |
| Fallbeispiel mit Lösungsvorschlägen | ja | nein | ja | nein | ja |
| zusätzlich erforderliches Problemwissen | Reihenfolge Wunschnachb ar-schaften | keines | Orientierung | keines | keines |
| Geschoßanzahl | einfach | einfach | mehrfach | einfach (über "Fixierung" mehr-fach theoretisch | einfach |

Tabelle 1 - Zusammenfassung und Vergleich der Programmeigenschaften

¹¹ Reinhard König, 2011, KREMLAS Entwicklung einer kreativen evolutionären Entwurfsmethode für Layoutprobleme in Architektur und Städtebau, 2011, Hrsg. Dirk Donath, Reinhard König und Frank Petzold, Bauhaus Universität Weimar, Seite 141 ff.

3. Konzept

Aus der Analyse geht hervor, dass konkrete Fallbeispiele bei jenen Arbeiten besonders intensiv dokumentiert sind, wo diese Beispiele das Regelwerk des darauf folgenden Ablaufs bereits definieren. Exakt das vorgegebene Problemwissen muss zwingend vor Programmstart erarbeitet werden. Insbesondere bei den letzten beiden Programmen ist jedoch überhaupt kein Problemwissen notwendig, jedoch genügend Entwicklungspotenzial vorhanden, die Programme um eine solche verfügbare Angabe zu ergänzen und in die Lösungsfindung miteinzubinden. Reinhard König schreibt hierzu: *"Das Potenzial des MOOP-Ansatzes liegt darin, dass theoretisch beliebig viele weitere Kriterien in das System integriert werden können. Für alle Kriterienkombinationen kann nach pareto-optimalen Lösungen gesucht werden. [...] Die genannten Kriterien können im Rahmen des iterativen Verfahrens beliebig aktiviert oder deaktiviert werden."*¹²

Ein geeignetes Testbeispiel findet sich in Form einer aktuellen Wettbewerbsauslobung der Stadt Wien zum Bau eines Bildungscampus am Wiener Nordbahnhof. Als ein Projekt aus dem Bereich des Wiener Campusmodells, welches Kindergarten, Volks- bzw. neue Mittelschule sowie Freizeitangebote für Kinder und Jugendliche ineinander verschränkt¹³, erfüllt die Auslobung zwei hilfreiche Aspekte: Aus den Unterlagen kann ein konkretes Set an Anforderungen, sowohl Raumgrößen als auch die hierarchische Gliederung der Betriebsstruktur betreffend, abgeleitet werden. Gleichmaßen wird das Konzept des Wiener Campusmodells damit stellvertretend abgebildet und eventuelle Nutzen können auf zukünftige Projekte angewandt werden.

Die parallel entstehende Diplomarbeit "SimSchool - Optimierung der Abläufe in Schulbauten" von Daniel Elsner beschäftigt sich mit der geschoßweisen Funktionsaufteilung bei unterschiedlicher Geschoßgröße des Bildungscampus. Die Resultate dieser Arbeit werden übergeben und in einem der "Dichten Packung" ähnlichen Programm verarbeitet und damit einfache Grundrissvariationen erzeugt.

Hinzu kommt die Ergänzung des Programms um zwei Aspekte, die in den analysierten Arbeiten nicht oder nur kaum behandelt wurden. Einerseits werden die Proportionalitätseigenschaften der Flächen als beschränkendes Kriterium eingeführt um zu schmale, unbrauchbare Raumlösungen vermeiden zu können. Andererseits verlässt das Programm den vordefinierten quadratischen Gesamtraum und

¹² Reinhard König, 2011, Generierung von Grundriss-Layouts mittels Hybrider Evolutionsstrategie, aus: Arbeitspapiere Nr. 4, März 2011, Bauhaus Universität Weimar, Seite 35, verfügbar unter: http://infar.architektur.uni-weimar.de/service/drupal-infar/sites/default/files/upload/Publikationen/Arbeitspapiere/InfAR_04_Grundriss-Layouts.pdf [zuletzt aufgerufen am 10.01.2016]

¹³ Wiener Schulen (Magistratsabteilung 56), Das Wiener Campusmodell, verfügbar unter: <https://www.wien.gv.at/bildung/schulen/schulbau/campus/> [zuletzt aufgerufen am 12.2.2016]

bietet eine Bandbreite an Außenhüllen. Die multikriterielle Optimierung betrifft somit drei Bereiche: Überlappungsfreiheit der Räume, Anordnung innerhalb festgelegter Gebäudegrenzen und Ausrichtung aufgrund von Beziehungspräferenzen der Räume.

"Die Beschreibung eines komplexen Problems enthält dessen Lösung dann, wenn sie es in Teilprobleme zerlegt, die sich erstens unabhängig voneinander lösen lassen, und deren Lösungen sich zweitens zur Gesamtlösung zusammensetzen lassen."¹⁴ Die oben beschriebenen Abläufe werden gebündelt durchgeführt und ergeben letztendlich eine kontinuierliche Bewegung, bis alle Bedingungen pareto-optimal erfüllt sind.

Zusammengefasst kann nach folgenden Anforderungen optimiert werden:

| | |
|--|---|
| - Überlappungsfreie Anordnung | $Sg = \sum_{i=1}^{n-1} \sum_{j=i+1}^n (S[i] \cap S[j])$ |
| - Anordnung in der vorgegebenen Grundrisstypologie | $Ig = \sum_{i=1}^n (S[i] \cap S[t])$ |
| - Anziehen und Abstoßen der Funktion | $Lg = \sum_{i=1}^n \lambda l \binom{i}{j}$ |

Der abgewandte Algorithmus wird nachstehend an einem einfachen Beispiel mit drei Räumen, die in einem länglichen Grundriss angeordnet werden sollen, demonstriert. Das Beispiel verwendet folgende Eigenschaften:

| Bezeichnung | Farbe | Größe | Beziehungen: | A | B | C |
|-------------|-------|-------|--------------|-----|-----|---|
| Raum A | rot | 10 | | | +/- | - |
| Raum B | blau | 10 | | +/- | | + |
| Raum C | gelb | 10 | | - | + | |

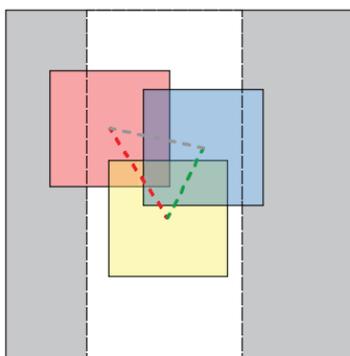


Abbildung 10.1 - Schritt 1 Initialisierung

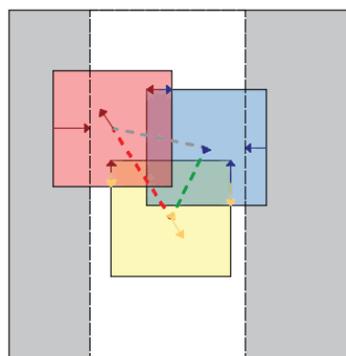


Abbildung 10.2 - Schritt 2 Analyse

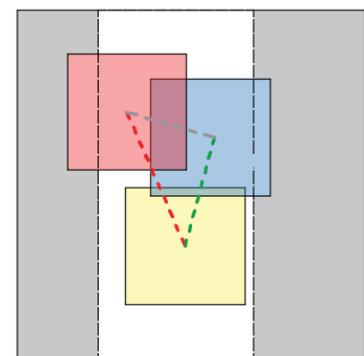


Abbildung 10.3 - Schritt 3 Bewegung

¹⁴ Georg Franck und Tomor Elezkurtaj, 2011, Design Methods und computergestützte Grundrissentwicklung, Seite 2, verfügbar unter: http://www.iemar.tuwien.ac.at/wp-content/uploads/2011/12/design_methods.pdf [zuletzt aufgerufen am: 15.02.2016]

Schritt 1: Die Räume werden mit beliebigen, randomisierten Koordinaten erzeugt. Ihnen werden ihre jeweiligen Werte zugeordnet, in diesem Beispiel die gleiche Größe, die Farbe sowie die Werte aus der Adjazenzmatrix (wobei grün positiv, rot negativ und grau neutral repräsentiert) (Abb. 10.1).

Schritt 2: Ein jeder Raum prüft seine Bedürfnisse aufgrund von Überlappung, Adjazenz und Außentypologie. Daraus ergeben sich - für jedes Bedürfnis - Richtungen und Längen. Diese Vektoren können auch gegenläufig sein und sich damit gegenseitig aufheben (Abb. 10.2)

Schritt 3: Die Räume bewegen sich aufgrund ihrer errechneten Vektoren; im Beispiel dargestellt jeweils um die Hälfte ihrer Vektorlänge (Abb. 10.3)

Die einzelnen Schritte mit einer Bewegung um mindestens die Hälfte der wirkenden Vektoren von der Initialisierung bis zum Erreichen eines Zustandes vollständiger Zufriedenheit verhalten sich für das obige Beispiel wie folgt (Abb. 11):

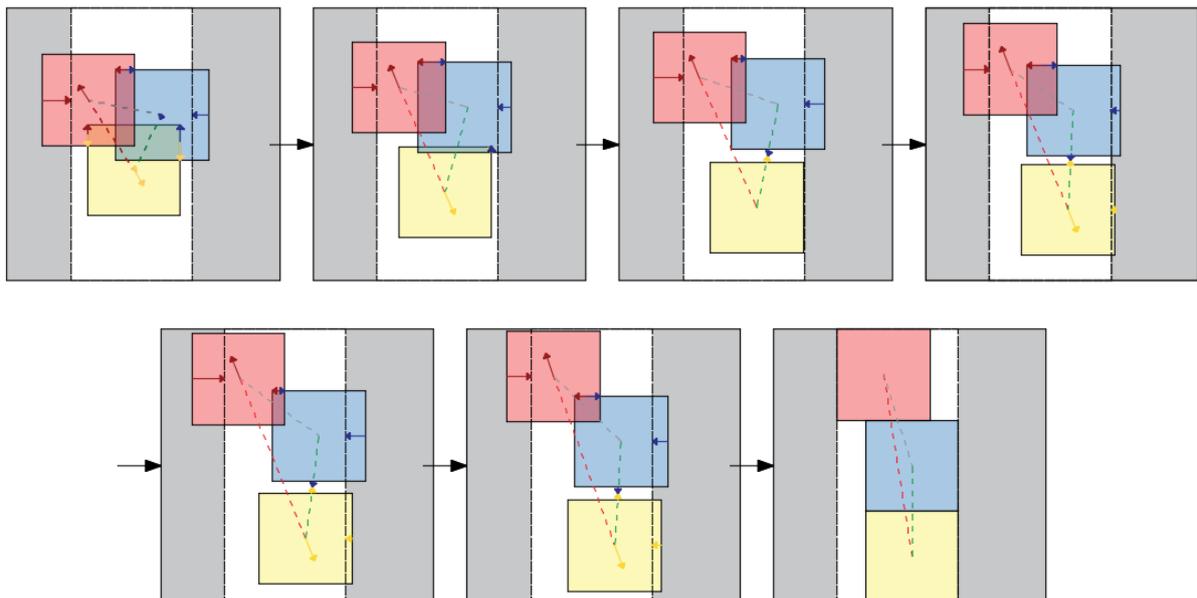


Abbildung 11 - Von der Initialisierung zum Layout

Sucht jedes der Kriterien seine alleinige Ideallösung, unterscheiden sich die gefundenen Layouts sowohl untereinander als auch von der gemeinsamen Lösung stark (Abb. 12).

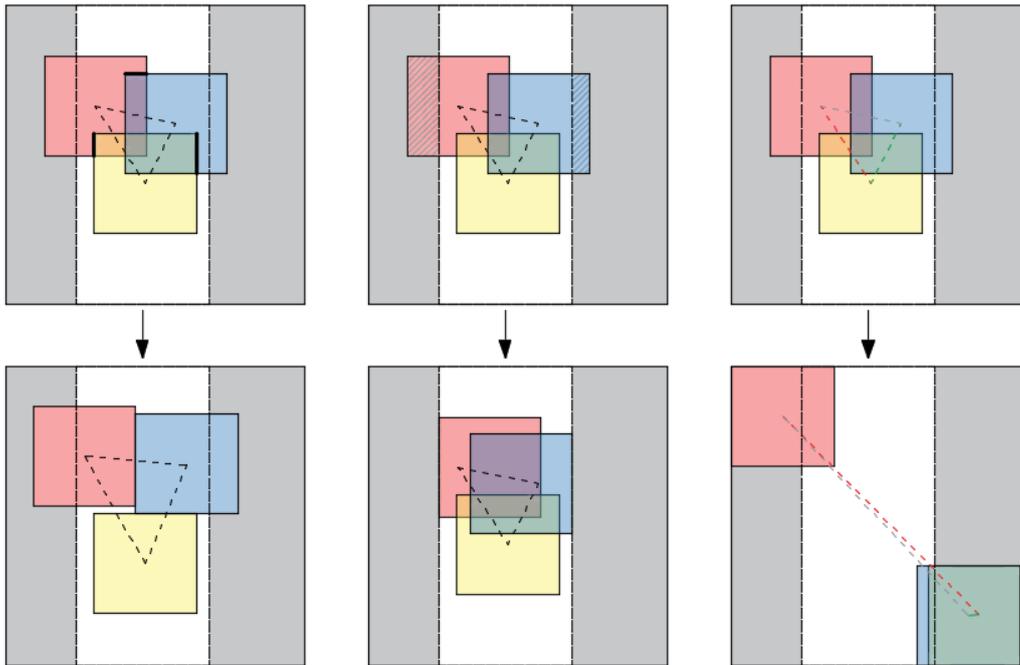


Abbildung 12 - Optimierung Überlappung, Typologie und Adjazenzen

Die zu erreichenden Zustände errechnen sich aus der Geometrie der jeweiligen Objekte, nachfolgend am Beispiel der Überschneidung beschrieben: Tritt eine Überschneidung auf, wird ein Vektor mit dem jeweils nächst liegenden Eckpunkt berechnet (Abb. 13.1). Bewegen sich nun beide Flächen um die Hälfte der jeweiligen Vektorlänge in die Richtung, die vom Eckpunkt zum Schnittpunkt verläuft, wird die Überlappung eliminiert (Abb. 13.2). Bereits in Reinhard Königs Arbeitspapier Nr. 4 wird diese Methode als "effiziente Heuristik"¹⁵ zum Minimieren der Schnittflächen bezeichnet. Auch die zusätzliche Anpassung der Proportionen, die König beschreibt, wird übernommen. Anstelle des Verschiebens werden hierbei, um einen überlappungsfreien Zustand zu erreichen, die sich überschneidenden Seiten um den halben Vektor verkürzt und die zweite Seitenlänge über die gleichbleibende Größe und die verkürzte Seite neu errechnet (Abb. 13.3). Die Gewichtungen und Wahrscheinlichkeitswerte für Verschiebung und Proportion entsprechen dem Programm "Dichte Packung".

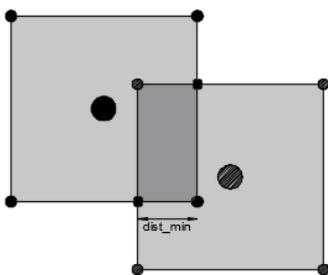


Abbildung 13.1 - Überlappung

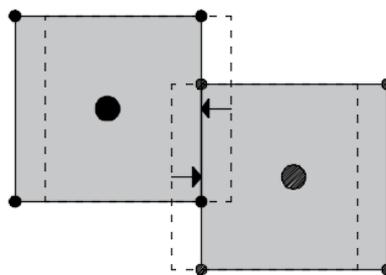


Abbildung 13.2 - Verschiebung

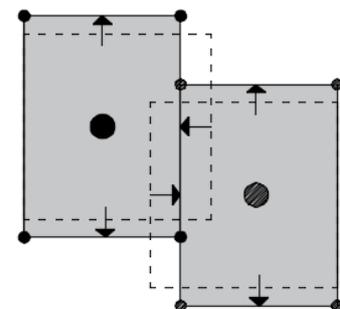


Abbildung 13.2 - Proportionsänderung

¹⁵ Reinhard König, 2011, Generierung von Grundriss-Layouts mittels Hybrider Evolutionsstrategie, aus: Arbeitspapiere Nr. 4, März 2011, Bauhaus Universität Weimar, Seite 12

Die Analysen des Layoutsolver "Dichte Packung", auf der das Konzept u.a. beruht, weisen darauf hin, dass bestimmte Proportionsverhältnisse bevorzugt generiert werden.¹⁶ Dieser Effekt soll mit einer einzuhaltenden Bedingung pro Lösungsvorschlag oder in weiter fortgeschrittener Problemanalyse pro Raum unterbunden werden und die Lösungsqualität steigern. Letztendlich wird damit vermieden, dass die gefundenen Lösungen den Eindruck erwecken, aus einer Ansammlung langer Gänge zu bestehen. Die Bedingung bewirkt, dass die auferlegte Eigenschaft während der Lösungsfindung immer eingehalten wird und nicht dahingehend optimiert wird. Eine manuelle Anpassung der Seitenproportionen wie sie z.B. in GAMES zu finden ist, wird bewusst nicht umgesetzt, da die Programmtests in der Analyse gezeigt haben, dass Veränderungen weitestgehend schon nach wenigen Iterationen kaum mehr sichtbar sind. Dieser Parameter fällt ohnehin bereits durch die vorab festgelegten Werte der maximalen Seitenverhältnisse pro Flächeneinheit in einen wohlüberlegten Rahmen.

Um die Anwendung von der hypothetischen quadratischen Programmwelt, wie man sie aus "Dichte Packung" kennt, zu lösen, werden zusätzlich auch vereinfachte Gebäudeschemata herangezogen, um die Resultatfindung auf geänderte äußere Begrenzungen hin zu testen. Hierbei wurden als Grundtypologien folgende Formationen für das Programm vorbereitet: Dehnt sich das Quadrat, entsteht ein "O", und dessen Hälfte, ein "C". Streckt ("I") und schiebt man anschließend das Quadrat, entstehen Gebilde wie "L", "T", "H" und "Z" sowie durch Zweiteilung ein "Pavillon" (Abb. 15). Die gesamte innenliegende Fläche ist der Wert, der nach oben hin optimiert werden soll. Hierzu wird, ähnlich zum Abprallvektor, der kürzeste Vektor zum Grundriss gefunden und die Fläche in dessen Richtung bewegt (Abb. 14).

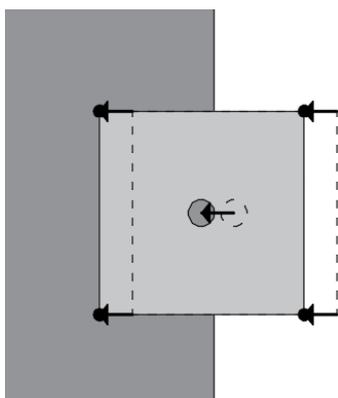


Abbildung 14 - Bewegung des Raums in den Grundriss



Abbildung 15 - ausgewählte Gebäudetypologien

¹⁶ Reinhard König, 2011, KREMLAS Entwicklung einer kreativen evolutionären Entwurfsmethode für Layoutprobleme in Architektur und Städtebau, 2011, Hrsg. Dirk Donath, Reinhard König und Frank Petzold, Bauhaus Universität Weimar, Seite 150

Die hinterlegten Beziehungen zu anderen Räumen können positiv, negativ oder neutral sein. Tendenziell ziehen sich positive Nachbarschaften an sowie sich negative Nachbarschaften abstoßen (Abb. 16). Daraus leitet sich auch die Optimierungsfunktion ab, mittels derer die durchschnittliche Länge von einer Raumaußengrenze zur nächsten für positive Beziehungen gegen Null und für Negative gegen Unendlich angestrebt wird.

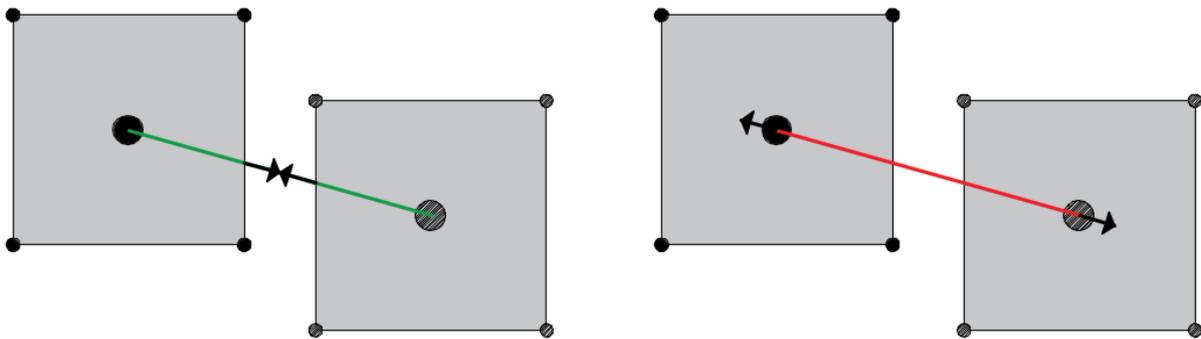


Abbildung 16 - Anziehung und Abstoßung durch positive und negative Nachbarschaftsbeziehungen

Da die Erschließung nicht als eigene räumliche Funktion importiert wird, sondern innerhalb einer dafür vorgesehenen freibleibenden Restfläche erst später planerisch berücksichtigt werden soll, fällt diese Beziehung als zwingende Nachbarschaft aus (siehe Kapitel 5.1).

Ein besonders reizvoller Aspekt der kreativen Problemlösung durch das Programm liegt in der Möglichkeit, in den Lösungsfindungsprozess eingreifen und Anpassungen an das Layout vorzunehmen zu können. Das Programm kann jederzeit pausiert und die Räume innerhalb der Weltgrenzen durch den Benutzer mittels Drag & Drop verschoben werden. Genauso kann die Gebäude-Typologie (Grundtypen bzw. Typen mit gleicher Größe) auch ohne Neuimport der Flächen während der Lösungsfindung geändert werden. Nimmt man an, der Nutzer hat seine Gründe um den gelben Raum C in Schritt Drei an die Oberseite zu verschieben (Abb. 17), vernimmt die Suche nach der Lösung umgehend einen anderen Verlauf.

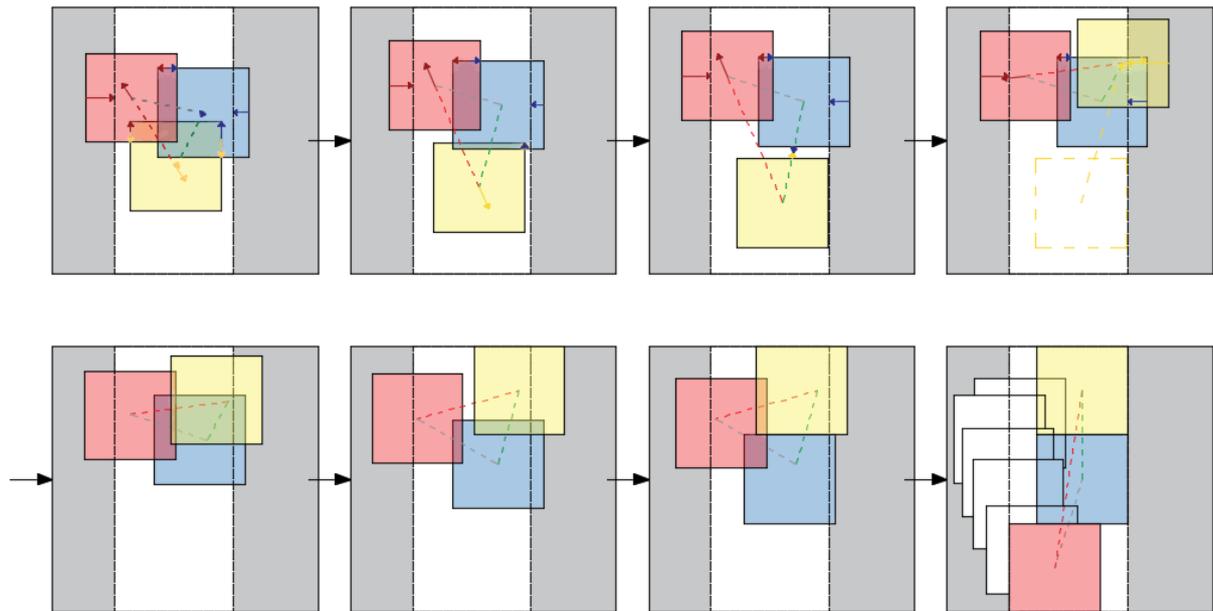


Abbildung 17- Lösungssuche mit Interaktion in Schritt 4

Bewusst werden bestimmte Elemente ausgeklammert, auf die bisherige Publikationen ein Augenmerk gelegt haben. Aufgrund des Anspruchs, in einem sehr frühen Entwurfsstadium eine Vielzahl an unterschiedlichen, komplexen Anordnungen zu generieren, wird von minimalen Kontaktlängen für Raumverbindungen abgesehen. Des weiteren wird die Funktion der Erschließung nicht als eigenständige Fläche aufgenommen, da gerade diese Kriterien die Ergebnisse zwar gerne "real" aber umso deutlicher nicht mehr interpretierbar wirken lassen und die Kreativität des Nutzers selbst unter Umständen dadurch eingeschränkt wird. Die Erschließung wird als innerhalb eines mindestens freibleibenden Zehntels der verfügbaren Fläche liegend angenommen.

Die Arbeit erhebt keinesfalls den Anspruch, nachzubauende Schemata zu kreieren sondern vielmehr einen raschen computerunterstützten Entwurfsprozess zu simulieren, der ein Vielfaches der Rahmenbedingungen schon mitabbildet.

4. Implementierung

Für die Anwendung der im vorherigen Kapitel zusammengefassten Funktionalitäten wurde NetLogo¹⁷ gewählt. Die Wahl basiert auf dem Schwerpunkt des Programms: Simulation durch Multi-Agents¹⁸ über Zeiträume, die auf Logo basierende einfach erlernbare Programmiersprache und eine für die genannten Zwecke mit allen absehbaren Funktionalitäten ausgestattete wirkende Modellierwelt. Insbesondere das Command Center mag manchem Architekten an AutoCAD erinnern und ist für kurzfristige Codeadaptierung ein sympathisches, hilfreiches Tool.

Zum Verständnis der gewählten Variablennamen sei darauf hingewiesen, dass Eigenschaften von Objekten (Patches, Agenten und Links), die zusätzlich zu Systemvariablen benötigt werden, mit einem vorangestellten Unterstrich "_" gekennzeichnet sind. Allgemeine Variablen sind nicht speziell gekennzeichnet. Die Implementierung erfolgt im 2D-Arbeitsbereich. Ergänzend zu NetLogo 5.1.0 wird die Extension "Vectorview"¹⁹ von Ahmad Esmaeili verwendet.

4.1 Benutzeroberfläche

Die Benutzeroberfläche (Abb. 18) besteht aus zwei Steuerungsbereichen, einer interaktiven Grundrissansicht und einem Kontrollplot.

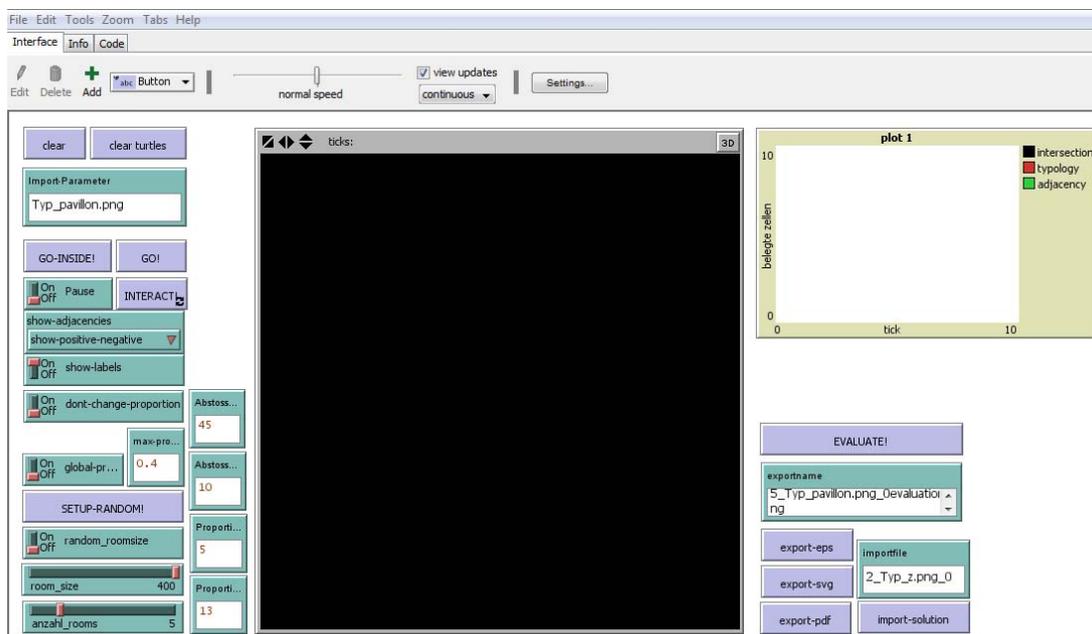


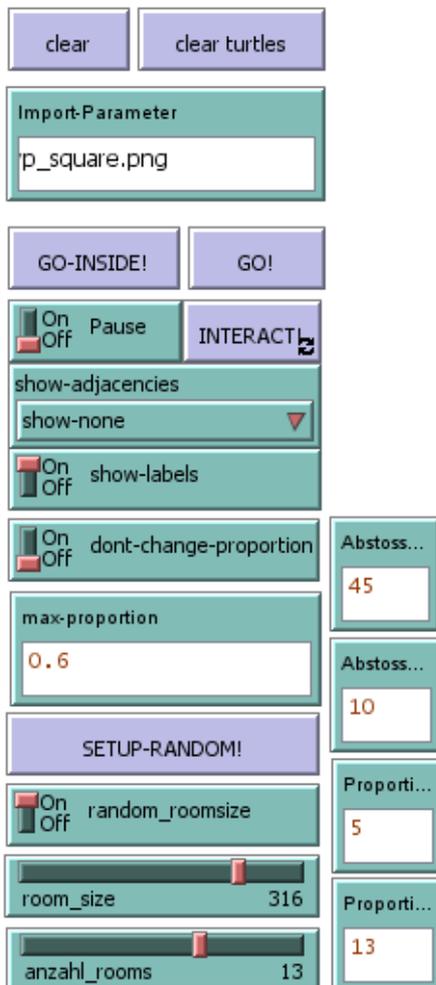
Abbildung 18 - Screenshot der Benutzeroberfläche

¹⁷ NetLogo von Uri Wilensky, 1999, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. Software unter: <http://ccl.northwestern.edu/netlogo/> [zuletzt abgerufen am: 13.02.2015]

¹⁸ Bei einem Multiagentensystem oder MAS handelt es sich um ein System aus mehreren gleichartigen oder unterschiedlich spezialisierten handelnden Einheiten, Software-Agenten, die kollektiv ein Problem lösen. (verfügbar unter: <https://de.wikipedia.org/wiki/Multiagentensystem>, zuletzt abgerufen am: 19.01.2016)

¹⁹ Ahmad Esmaeili, 2015, Software unter: <https://github.com/aesmaeili/vectorview> [abgerufen am 07.03.2016]

Im linken Teil werden einige Voreinstellungen getroffen (Abb.19). Hier kann der Nutzer aus mehreren Optionen zu Lösungsfindung hinsichtlich Import, globaler oder individueller Steuerung sowie der Darstellung im Viewport bzw. dem späteren Export wählen.



Bereinigen der aktuell geladenen Inhalte mittels "clear".

Wird eine Gebäudetypologie als Außenhülle eingehalten (Import-Parameter + GO-INSIDE!) oder wird die gesamte quadratische Programmwelt als möglicher Grundriss gesehen (GO!)? (siehe Punkt 4.4)

Pausenfunktion der Berechnung und Aktivierung der Interaktion.

Einstellung für Anzeige von Beziehungen (positiv, negativ, positiv und negativ, alle, keine) sowie Ein-/Aus für Raumbeschriftung.

Optional werden die ursprünglichen Proportionen garnicht verändert oder über eine globale Variable gesteuert.

Die Flächen betreffend kann anstelle eines speziellen Anwendungsfalls auch ein randomisiertes Beispiel mit variabler Raumgröße und -anzahl erzeugt werden. (siehe Punkt 4.2)

rechts: Optional werden die Wahrscheinlichkeitswerte für Verschiebung und Proportionsänderung verändert (siehe Punkt 4.5)

Abbildung 19 - Screenshot der Schaltflächen zu Programmstart und Einstellungen

Im Programm zentral positioniert befindet sich ein Viewport, in dem die Lösungsgenerierung in Echtzeit angezeigt wird (Abb. 20). Hierbei wird ob der leichteren Lesbarkeit der Funktionsbezeichnungen die Typologie in Schwarz, die Restfläche in Weiß dargestellt. Direkt in diesem Feld kommt es auch zu einer möglichen Interaktion durch den Nutzer.

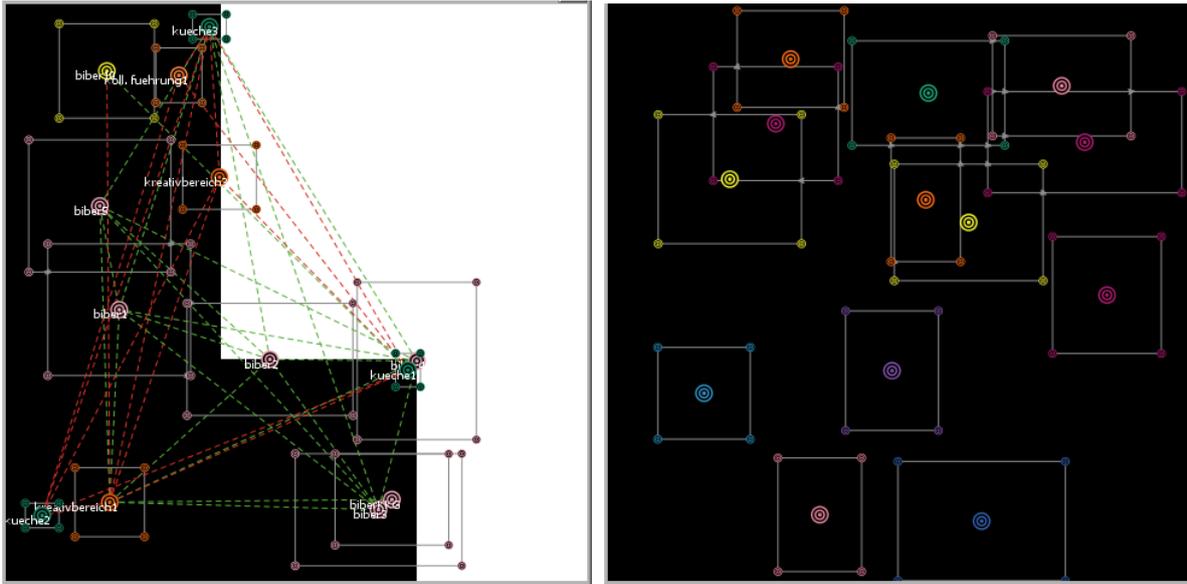


Abbildung 20 - Lösungsfenster (links Importdatei mit Typologie L mit Darstellung der Beziehungen, rechts randomisiertes Beispiel in Gesamtwelt)

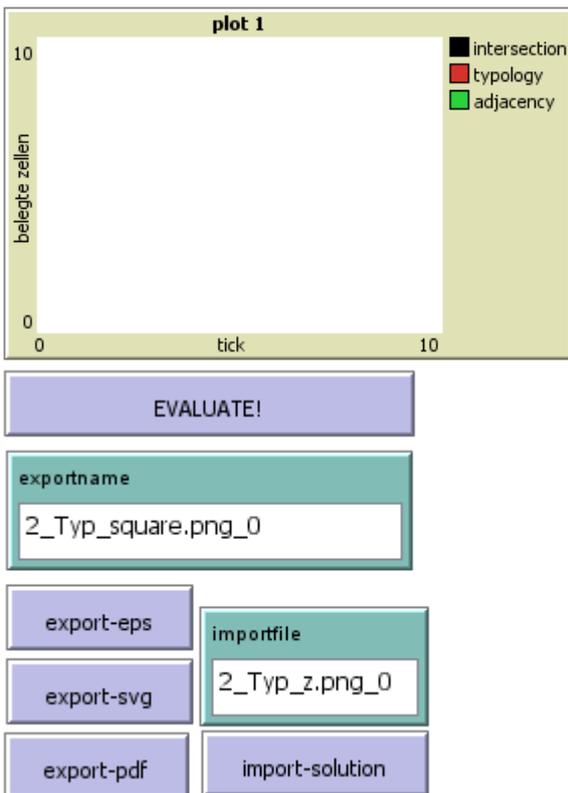


Abbildung 21 - Fenster für Nachverfolgungsgraph und Import/Exportbereich

Grafisch können die multikriteriellen Optimierungswerte über eine Plot-Tabelle mitverfolgt werden (Abb. 21 oben).

Die Tendenz der Plotlinien soll kontinuierlich schrumpfen.

Der Name für die Exportdateien wird aus dem Importdateinamen sowie der Typologie bereits vom Programm vorbereitet (Abb. 21 unten). Unter diesem vom Nutzer überschreibbaren Namen befinden sich die Möglichkeiten, die Lösung zu exportieren oder eine bereits exportierte Lösung wieder zu importieren. Entspricht die Lösung zwar noch nicht dem Maximum an Optimierbarkeit, reicht dem Nutzer jedoch für seine weitere

Verwendung, ist jederzeit ein Stop mit Weiterverarbeitung möglich.

4.2 - Prozessablauf

Der auf die wesentlichen Schritte reduzierte Programmablauf zur Lösungsgenerierung kann aus dem Diagramm (Abb. 22) entnommen werden. In den nachfolgenden Unterkapiteln finden sich jeweils detaillierte Informationen zur Umsetzung.

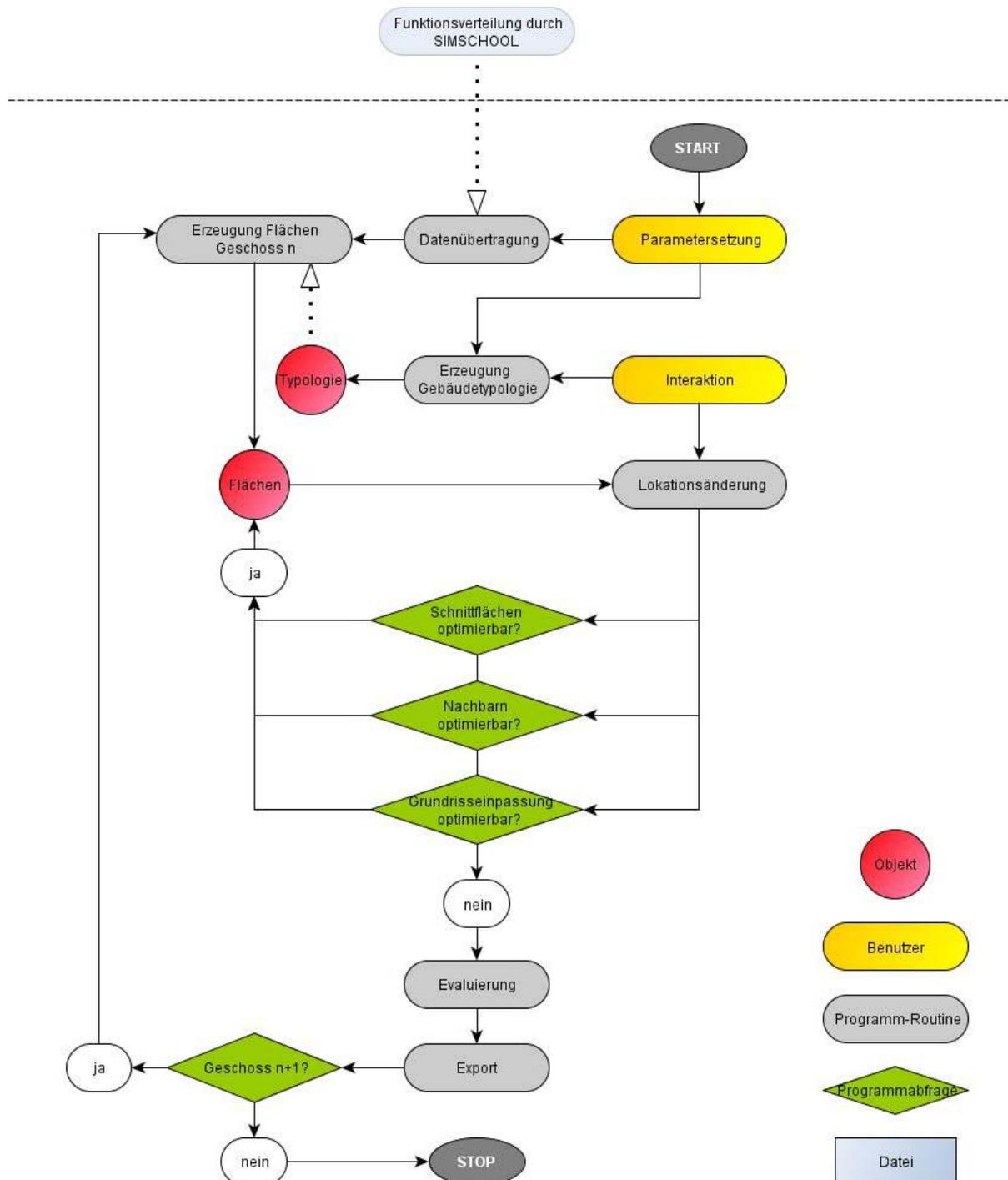


Abbildung 22 - Überblick Programmablauf

4.2 Datenübertragung / Daten-Import

Als Schnittstelle zwischen dem Geschoßaufteilungsalgorithmus von Daniel Elsner und dem Raumanordnungsalgorithmus werden aus Ergebnissen von Elsners Arbeit jeweils CSV-Dateien (comma-separated values) erstellt. Die Detailschritte dafür sind der Abb. 23 zu entnehmen. Diese CSV enthält bereits die Angaben zu Raumbezeichnung, Größe, Geschoß, Nachbarschaftspräferenz und dargestellter Farbe. Aus einer separat erstellten Datei holen sich die Räume Angaben zu ihren bevorzugten Seitenverhältnissen, desweiteren wird die prozentuale Größe am Geschoß ermittelt. Mittels VBA (Visual Basic for Applications) in Microsoft Excel werden so die nachbearbeiteten Daten (Abb. 24) geschoßweise in neue CSV-Dateien (Abb. 25) umgewandelt, die daraufhin wieder in NetLogo importiert werden können.

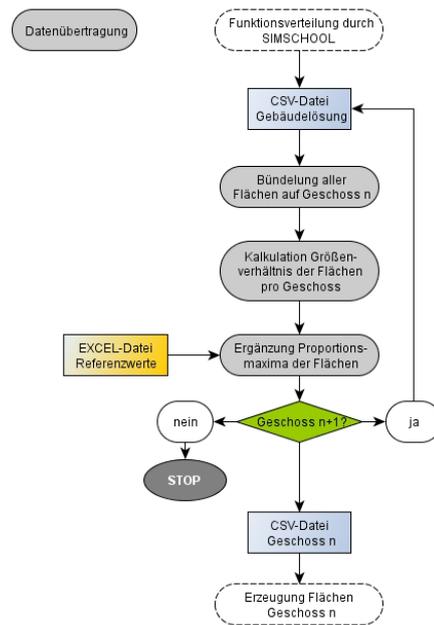


Abbildung 23 - Diagramm Ablauf Import

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | | | | | |
|----|---------|----------------------|----------|---------|-------------|------------------------------|-----------|------------|-------------|-------------|-------------|---------|-----------|--------|----------|---------|---------|----------|----|------------------------------|----|
| 1 | CSV-MEI | | | | | | | | | | | | | | | | | | | | |
| 2 | | "Geschoss" | "Funktio | "Breed" | "Mein Berei | "Farbe" | "Flaeche" | "Hoehe | "Propc | "globa | "local | "Link-I | "alle Lir | anzahl | reallist | Fixnumr | | | | | |
| 3 | 0 | "3-Fach-Turnhalle" | 7.2 | 0.66 | 1619 | [10 2 0 2 2 2 0 1 1] | 0 | "3-Fach-T | bereiche | "3-Fach-Tur | [192 192 19 | 1619 | 7.5 | 0 | 100 | 100 | [["Gyn | [["Gymi | 10 | [10 2 0 2 2 2 0 1 1] | 1 |
| 4 | 0 | "Gymnastiksaele" | 7.2 | 0.66 | 892 | [2 10 0 2 2 2 0 1 1] | 0 | "Gymnas | bereiche | "Gymnastik | [192 192 19 | 892 | 5.5 | 0 | 99 | 99 | [["3-Fa | [["3-Fac | 10 | [2 10 0 2 2 2 0 1 1] | 2 |
| 5 | 0 | "kreativbereich2" | 24.8 | 0.5 | 1185 | [0 0 10 2 2 2 2 0 1 2] | 0 | "kreativ | unterbereic | "NMS / FMS | [234 102 28 | 1185 | 3 | 0 | 82 | 100 | [["bibi | [["bibi | 10 | [0 0 10 2 2 2 2 0 1 2] | 3 |
| 6 | 0 | "biber7" | 44.7 | 0.5 | 896 | [2 2 2 10 2 2 2 2 1] | 0 | "biber7" | unterbereic | "NMS / FMS | [253 204 26 | 896 | 3 | 0 | 67 | 82 | [["krei | [["kreat | 10 | [2 2 2 10 2 2 2 2 1] | 4 |
| 7 | 0 | "biber8" | 44.7 | 0.5 | 914 | [2 2 2 2 10 2 2 2 1] | 0 | "biber8" | unterbereic | "NMS / FMS | [253 204 26 | 914 | 3 | 0 | 66 | 80 | [["krei | [["kreat | 10 | [2 2 2 2 10 2 2 2 1] | 5 |
| 8 | 0 | "biber9" | 44.7 | 0.5 | 914 | [2 2 2 2 10 2 2 2 1] | 0 | "biber9" | unterbereic | "NMS / FMS | [253 204 26 | 914 | 3 | 0 | 65 | 80 | [["krei | [["kreat | 10 | [2 2 2 2 10 2 2 2 1] | 6 |
| 9 | 0 | "biber6" | 44.7 | 0.5 | 896 | [2 2 2 2 10 2 2 2 1] | 0 | "biber6" | unterbereic | "NMS / FMS | [253 204 26 | 896 | 3 | 0 | 63 | 77 | [["krei | [["kreat | 10 | [2 2 2 2 10 2 2 2 1] | 7 |
| 10 | 0 | "kuecheSpeisesaal" | 74.3 | 0.4 | 649 | [0 0 0 2 2 2 10 0 0] | 0 | "kueche | unterbereic | "NMS / FMS | [18 128 129 | 649 | 3 | 0 | 54 | 65 | [["Gyn | [["Gymi | 10 | [0 0 0 2 2 2 10 0 0] | 8 |
| 11 | 0 | "koll. fuehrung2" | 25.8 | 0.75 | 96 | [1 1 1 2 2 2 0 10 1] | 0 | "koll. fue | unterbereic | "NMS / FMS | [238 131 59 | 96 | 3 | 0 | 0 | 0 | [] | [] | 10 | [1 1 1 2 2 2 0 10 1] | 9 |
| 12 | 0 | "Veranstaltungssaal" | 24.8 | 0.66 | 283 | [1 1 2 1 1 1 10 1 10] | 0 | "Veranst | bereiche | "Veranstalt | [234 102 28 | 283 | 4 | 0 | 0 | 0 | [] | [] | 10 | [1 1 2 1 1 1 10 1 10] | 10 |
| 13 | 1 | "kreativbereich1" | 24.8 | 0.5 | 267 | [10 2 2 2 2 0 0 2 0 1 0 0] | 1 | "kreativ | unterbereic | "VS + KDG" | [234 102 28 | 267 | 3 | 0 | 85 | 100 | [["bibi | [["3-Fac | 13 | [10 2 2 2 2 0 0 2 0 1 0 0] | 11 |
| 14 | 1 | "biber3" | 136.3 | 0.5 | 1121 | [2 10 2 2 2 2 1 1 1 1 1 1] | 1 | "biber3" | unterbereic | "VS + KDG" | [240 154 19 | 1121 | 3 | 0 | 50 | 58 | [["krei | [["kreat | 13 | [2 10 2 2 2 2 1 1 1 1 1 1] | 12 |
| 15 | 1 | "biber2" | 136.3 | 0.5 | 1117 | [2 2 10 2 2 2 1 1 1 1 1 1] | 1 | "biber2" | unterbereic | "VS + KDG" | [240 154 19 | 1117 | 3 | 0 | 50 | 58 | [["krei | [["kreat | 13 | [2 2 10 2 2 2 1 1 1 1 1 1] | 13 |
| 16 | 1 | "biber4" | 136.3 | 0.5 | 1121 | [2 2 2 10 2 2 1 1 1 1 1 1] | 1 | "biber4" | unterbereic | "VS + KDG" | [240 154 19 | 1121 | 3 | 0 | 48 | 57 | [["krei | [["kreat | 13 | [2 2 2 10 2 2 1 1 1 1 1 1] | 14 |
| 17 | 1 | "biber5" | 136.3 | 0.5 | 1115 | [2 2 2 2 10 2 1 1 1 1 1 1] | 1 | "biber5" | unterbereic | "VS + KDG" | [240 154 19 | 1115 | 3 | 0 | 48 | 57 | [["krei | [["kreat | 13 | [2 2 2 2 10 2 1 1 1 1 1 1] | 15 |
| 18 | 1 | "biber1" | 136.3 | 0.5 | 1117 | [2 2 2 2 10 2 1 1 1 1 1 1] | 1 | "biber1" | unterbereic | "VS + KDG" | [240 154 19 | 1117 | 3 | 0 | 31 | 36 | [["krei | [["kreat | 13 | [2 2 2 2 10 2 1 1 1 1 1 1] | 16 |
| 19 | 1 | "kueche3" | 74.3 | 0.4 | 63 | [0 2 2 2 2 10 0 1 0 0 0 1] | 1 | "kueche | unterbereic | "VS + KDG" | [18 128 129 | 63 | 3 | 0 | 0 | 0 | [] | [] | 13 | [0 2 2 2 2 10 0 1 0 0 0 1] | 17 |
| 20 | 1 | "kueche1" | 74.3 | 0.4 | 63 | [0 1 1 1 1 1 10 1 0 0 0 2] | 1 | "kueche | unterbereic | "VS + KDG" | [18 128 129 | 63 | 3 | 0 | 0 | 0 | [] | [] | 13 | [0 1 1 1 1 1 10 1 0 0 0 2] | 18 |
| 21 | 1 | "biberKKG" | 136.3 | 0.5 | 622 | [2 1 1 1 1 1 1 1 10 1 1 1] | 1 | "biberK | unterbereic | "VS + KDG" | [240 154 19 | 622 | 3 | 0 | 0 | 0 | [] | [] | 13 | [2 1 1 1 1 1 1 1 10 1 1 1] | 19 |
| 22 | 1 | "kueche2" | 74.3 | 0.4 | 63 | [0 1 1 1 1 1 10 1 0 0 0 1] | 1 | "kueche | unterbereic | "VS + KDG" | [18 128 129 | 63 | 3 | 0 | 0 | 0 | [] | [] | 13 | [0 1 1 1 1 1 10 1 0 0 0 1] | 20 |
| 23 | 1 | "koll. fuehrung1" | 25.8 | 0.75 | 168 | [1 1 1 1 1 1 10 0 10 10 1 1] | 1 | "koll. fue | unterbereic | "VS + KDG" | [238 131 59 | 168 | 3 | 0 | 0 | 0 | [] | [] | 13 | [1 1 1 1 1 1 10 0 10 10 1 1] | 21 |
| 24 | 1 | "kreativbereich3" | 24.8 | 0.5 | 300 | [0 1 1 1 1 1 10 0 10 10 1 1] | 1 | "kreativ | unterbereic | "Musikschul | [234 102 28 | 300 | 3 | 0 | 59 | 100 | [["bibi | [["bibi | 13 | [0 1 1 1 1 1 10 0 10 10 1 1] | 22 |
| 25 | 1 | "biber10" | 44.7 | 0.5 | 500 | [0 1 1 1 1 1 1 2 1 1 1 1 10] | 1 | "biber10" | unterbereic | "Musikschul | [253 204 26 | 500 | 3 | 0 | 58 | 98 | [["krei | [["kreat | 13 | [0 1 1 1 1 1 1 2 1 1 1 1 10] | 23 |

Abbildung 24 - Screenshot der zusammengeführten Excel-Tabelle für eine Lösung mit 2 Geschoßen

```

0 "3-Fach-Turnhalle" 7.2 0.66 1619 [10 2 0 2 2 2 0 1 1 ]
0 "Gymnastiksaele" 7.2 0.66 892 [2 10 0 2 2 2 0 1 1 ]
0 "kreativbereich2" 24.8 0.5 1185 [0 0 10 2 2 2 2 0 1 2 ]
0 "biber7" 44.7 0.5 896 [2 2 2 10 2 2 2 2 1 ]
0 "biber8" 44.7 0.5 914 [2 2 2 2 10 2 2 2 1 ]
0 "biber9" 44.7 0.5 914 [2 2 2 2 10 2 2 2 1 ]
0 "biber6" 44.7 0.5 896 [2 2 2 2 2 10 2 2 1 ]
0 "kuecheSpeisesaal" 74.3 0.4 649 [0 0 0 2 2 2 10 0 0 ]
0 "koll. fuehrung2" 25.8 0.75 96 [1 1 1 2 2 2 0 10 1 ]
0 "Veranstaltungssaal" 24.8 0.66 283 [1 1 2 1 1 1 10 1 10 ]
  
```

Abbildung 25 - die aus Abbildung 24 erzeugte CSV für Geschoß 0

4.3 Grundeigenschaften und Aufbau der Flächen

Eine jede Fläche besteht aus insgesamt fünf Agenten (Turtles) und vier Links (Abb. 26). Bei der Initialisierung wird die angegebene Anzahl von Flächen pro Geschoß erzeugt: Jede der Flächen erhält die neben der Skizze genannten Werte, wobei diese Angaben aus dem Dateiimport stammen, global gesetzt oder randomisiert erzeugt werden (können).

Eigenschaften/Variablen der Fläche:

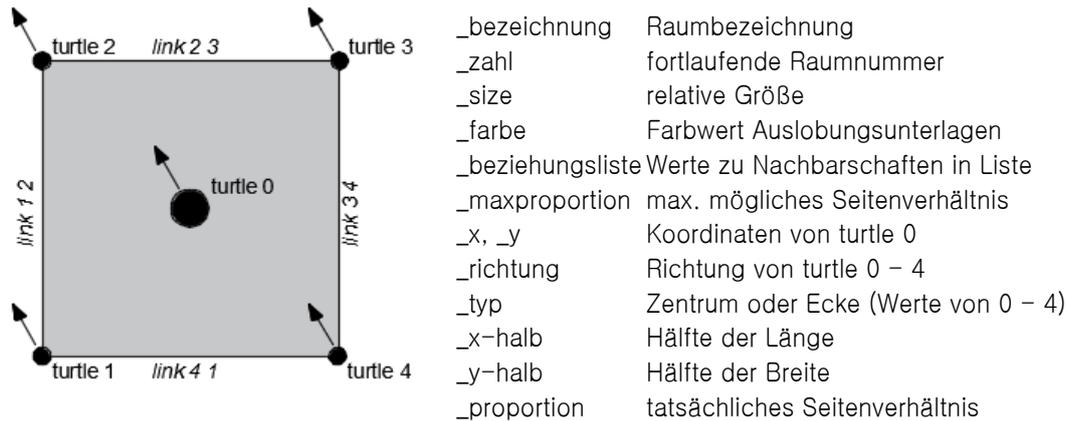


Abbildung 26 - Skizze einer Fläche

Raumbezeichnung bis Nachbarschaftsliste sind Importwerte. Das maximale Seitenverhältnis kann, wenn die Analyse fortgeschritten genug ist, mitimportiert werden. Alternativ kann sie über einen globalen Wert über alle Räume gesteuert werden. X-Koordinate, Y-Koordinate und Richtung werden per Zufallszahl für den Zentrums-Agenten frei im Raum gesetzt und können innerhalb und außerhalb gegebener Grundrissgrenzen liegen.

Länge und Breite sowie die daraus ableitbaren Werte für die Offsets der jeweiligen Eckpunkte `_x-halb` und `_y-halb` ergeben sich aus der vorgegebenen relativen Größe. Dies bedeutet, dass die Größe dem prozentuellen Anteil an der verfügbaren Fläche entspricht, wobei die gewählte Typologie (siehe Punkt 4.3) mitberücksichtigt wird. Die Seitenlängen der Flächen dürfen sich nur innerhalb eines festgesetzten Verhältnisses bewegen, wobei sie ihre relative Größe beibehalten müssen. Das standardmäßige anfängliche tatsächliche Seitenverhältnis ist 1:1, das heißt, es werden quadratische Flächen werden erzeugt. Somit kann das vorgegebene maximale Seitenverhältnis bei der Initialisierung nicht überschritten werden.

Über die Längen- und Breitenangaben werden auch die Eckpunkte der stets orthogonalen Flächen gesetzt. Der Typ gibt Auskunft über die Lage der Agenten: Das Zentrum trägt den Typ "0", die Ecken im Uhrzeigersinn von links unten aus die Typen "1" bis "4". Damit können Grenzen der Außenhülle in einzelnen Richtungen lokalisiert werden. Darüber hinaus werden zwischen den Eckpunkten Links

aufgespannt, die in diesem Anwendungsfall ungerichteten Verbindungen entsprechen, welche der Fläche ebenfalls eindeutig über ihre idente Linknummer zugewiesen werden.

Die Flächennummer wird als Index benutzt, über den sich alle einzelnen Bestandteile einer Fläche gleichzeitig bzw. gezielt ansprechen lassen. Auch die implementierte Adjazenzmatrix nimmt auf die jeweiligen Flächennummern Bezug.

4.4 Gebäudetypologien

Die neun Typen werden mittels Bilddateien in das Layoutprogramm geladen (Abb. 27). Die Anwendung interpretiert weiße Fläche als Teil der Typologie und schwarze Fläche als Außenbereich. Die jeweiligen Flächeninhalte stimmen bis auf geringe unvermeidbare Toleranzwerte überein, um auch während des Programmablaufs die gewählte Typologie ändern zu können. Die Abweichung hat die in Patches (Raster) aufgebaute Welt in NetLogo zugrunde liegen. Geometrische Formen ließen sich nur dann mit identen Maßen abbilden, wenn die Welt unendlich fein gerastert wäre. Im Beispiel aus Punkt 5 wird eine Welt aus 65x65 Patches, in Summe 4225, verwendet; die angegebenen Größenangaben sind ebenfalls auf eine solche Welt bezogen. Die als Teil des Grundrisses zur Verfügung stehenden Patches liegen in einem Bereich zwischen 2080 und 2130. Dabei wurde berücksichtigt, dass komplexere Geometrien tendenziell eine höhere Patchanzahl erhalten. Sowohl Mittel als auch Median liegen bei 2110 Patches, die mittlere Abweichung bei 10.

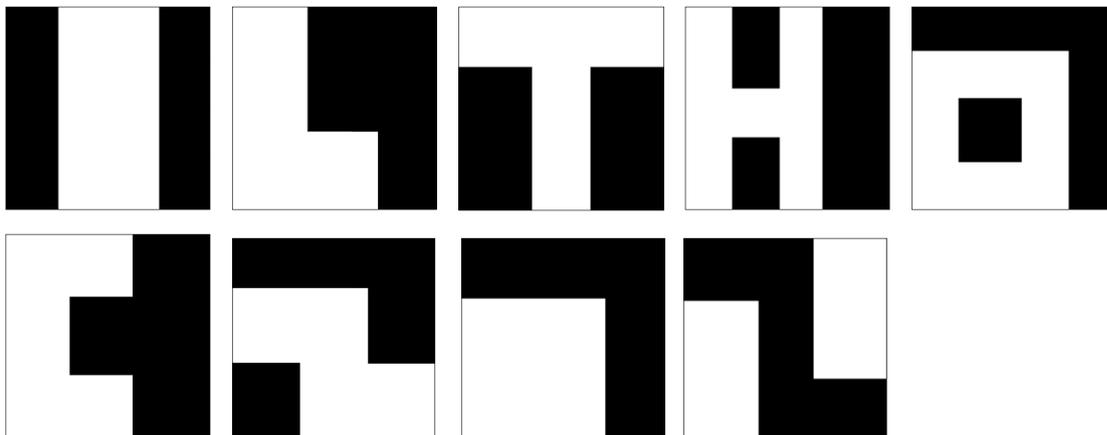


Abbildung 27 - implementierte Grundrisstypologien. Die Zahl in der Klammer gibt die Anzahl der Grundriss-Patches von an. Typologien von links oben nach rechts unten: I (2080), L (2110), T (2109), H (2125), O (2130), C (2100), Z (2107), Quadrat (2116), Pavillon (2115)

Beliebige weitere Typologien bis hin zu spezifischen Außenformen lassen sich im Bedarfsfall jederzeit mit einer einfachen Grafik in geläufigen PC-Programmen erstellen und können importiert werden, ohne Änderungen am Programm vornehmen zu müssen. Die Detailliertheit im Programm hängt dabei stets mit der gewählten Auflösung zusammen. Sind sehr feine (oder schräge) Formen notwendig, empfiehlt es sich, die Welt auf eine höhere Auflösung zu setzen, was jedoch eine steigenden Rechenkapazität bei allen Programmabläufen zu Folge hat.

4.5 Schnittflächenerkennung

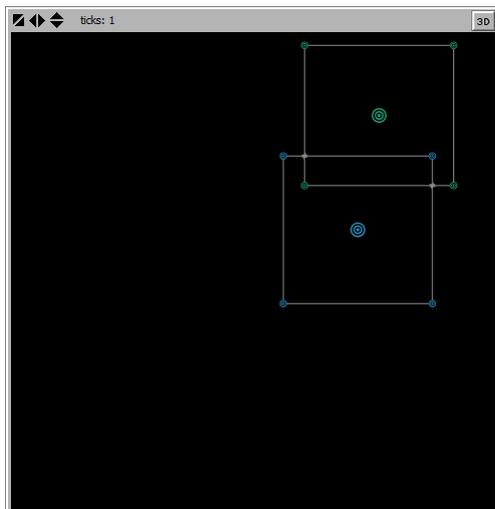


Abbildung 28 - Screenshot zweier sich schneidender Flächen

Die erzeugten Flächen sollen sich ohne Überschneidungen in dem ihnen zur Verfügung stehenden Raum anordnen. Dafür wird getestet, ob sich Links der jeweiligen Flächen untereinander kreuzen: Links schneiden sich stets dann, wenn sich auf ihrer gesamten Länge l ein mit einer abweichenden Flächennummer $_{zahl}$ bezeichneter Link befindet. Schneiden sich zwei Räume, ergibt dies vier Überschneidungen, wobei zwei davon jeweils im selben Punkt liegen (Abb. 28). Die Suche basiert auf dem Code des NetLogo-Examples "Intersecting Links Example" von Uri Wilensky.²⁰

4.6 Nachbarschaftsanordnung

Die Umsetzung von gewünschten oder nicht gewünschten Naheverhältnissen der Räume werden durch ein Set aus Beziehungslinks realisiert. Jeder Flächenmittelpunkt ist mit jedem weiteren Flächenmittelpunkt mit einem solchen Link verbunden. Diese Links erhalten im Zuge der Flächeninitialisierung einen Wert aus der mitgegebenen Importmatrix in Listenform, der negativ ("0"), neutral ("1") oder positiv ("2") sein kann.

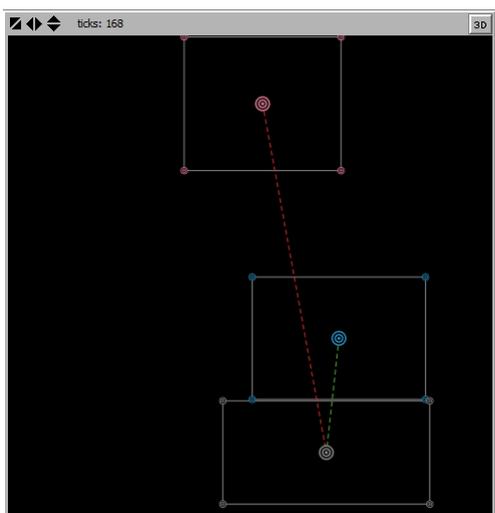


Abbildung 29 - Screenshots Beziehungsanordnung randomisierter Beispiele

Da die Beziehungswertliste aufgrund der vereinfachten Elementansprache idealerweise auch einen Beziehungswert zu sich selbst aufweist, liegt dieser standardmäßig bei "10" auch wenn keine Beziehungen zu sich selbst erzeugt werden. Während der Lösungssuche werden positive und negative Verbindungen analysiert, welche sich gegenseitig anziehen bzw. abstoßen solange sie aufgrund der Typologie oder Flächenüberschneidung einen Drang haben, sich zu bewegen. Dafür werden die jeweiligen (gegenläufigen) Richtungen der Beziehungslinks analysiert und, sofern der entsprechende Platz zur

²⁰ Uri Wilensky, verfügbar unter: http://modelingcommons.org/browse/one_model/2314#model_tabs_browse_info [zuletzt abgerufen am 06.03.2016]

Verfügung steht, erfolgt eine Bewegung der Fläche (Abb. 29).

Im folgenden Codebeispiel wird davon ausgegangen, dass positive Adjazenzen nur noch dann näher angezogen werden müssen, wenn ihr Beziehungslink länger ist als die addierten Wurzeln der beiden Flächen.

```
ask my-beziehungen with [_beziehungswert = 2 and link-length > (sqrt thissize + sqrt othersize)] [
  let xface [xcor] of other-end
  let yface [ycor] of other-end
  set thisheading thisheading + (atan (xface - thisx) (yface - thisy))
]
set thisheading thisheading / count my-links with [_beziehungswert = 2 and link-length > (sqrt thissize _
+ sqrt othersize)]
[...]
ask turtles with [_zahl = zahl] [set heading thisheading]
if can-move-all? [
fd moving-distance
]
[...]
```

4.7 Anordnung im Grundriss

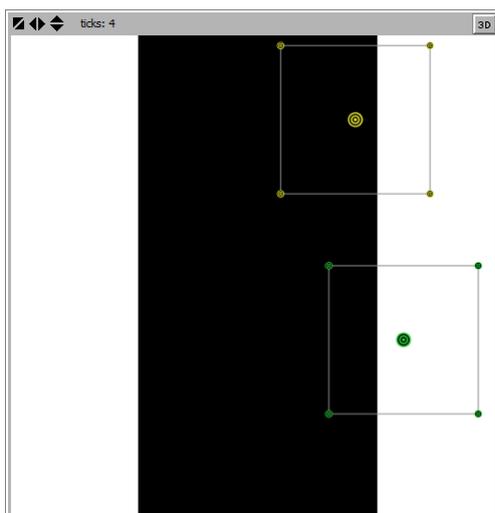


Abbildung 30 - Screenshot zweier Flächen auf dem Weg in eine I-Typologie

Um die Typologie-Anforderungen zu erfüllen, sucht das Programm die mittigen Turtles ($_typ = 0$), welche auf Patches außerhalb der Typologie liegen. Diese müssen zwingend bewegt werden, da dadurch zumindest die Hälfte der Fläche innerhalb der Typologie liegt. Der Abstand bis zum nächst gelegenen Patch innerhalb der Typologie wird daraufhin ermittelt und als neuer Verschiebevektor hinzugefügt. Räume, die noch nicht vollständig innerhalb der Grenzen liegen, suchen sich nach der gleichen Methode eine Richtung, in die sie sich kontinuierlich weiterbewegen möchten (Abb. 30).

```
to-report outside-border?
  let outside-border 0
  ask patch-here [if _grundriss = false [
    set outside-border 1]
  ifelse outside-border = 1
  [report true]
  [report false]
end

ask turtles with [_typ = 0 and shape = "target"] [
  set zahl = _zahl
  if outside-border? = true [
    face min-one-of patches with [_grundriss = _
      true] [distance myself]
    set currentheading heading
    ask turtles with [_zahl = zahl]
    set heading currentheading
    fd distspeed]]
```

4.8 Interaktion

Eine manuelle Verschiebung eines oder mehrerer Räume kann permanent in Echtzeit erfolgen. Der Nutzer pausiert den Berechnungsvorgang für den Vorgang der Verschiebung mittels Drag&Drop des Mittelpunkts (Abb. 31). Sobald alle gewünschten Änderungen direkt in der Grundrissvorschau durchgeführt sind, setzt er die Berechnung fort. Die Benutzeroberfläche wird währenddessen nicht verlassen. Ausgangsbasis für die Programmierung ist das NetLogo-Example "Mouse Drag One"²¹ von Uri Wilensky.

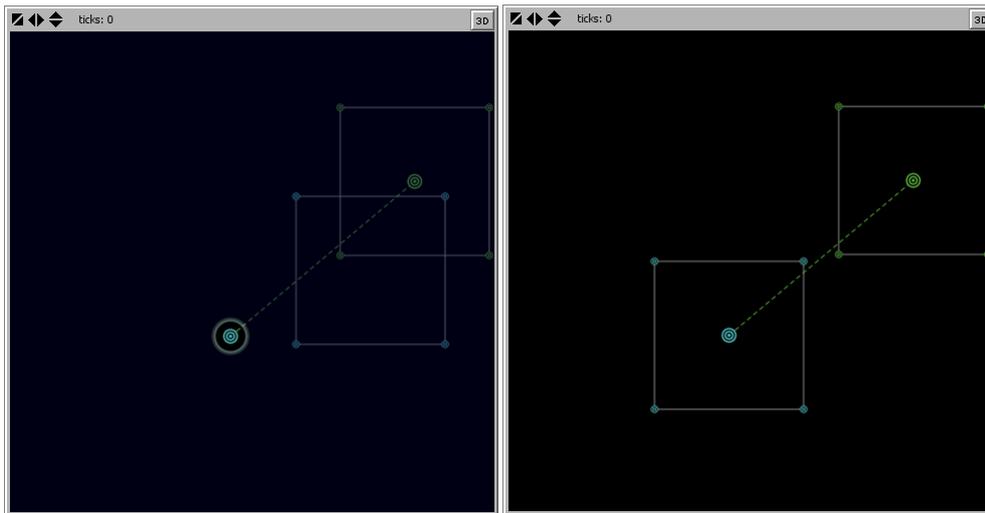


Abbildung 31 - Screenshot Verschiebung

4.9 Evaluierung

Als messbares sekundäres Beurteilungskriterium der verschiedenen Lösungen wurde eine Angabe zur Zentralität einer jeden Fläche implementiert.

Die Zentralität wird errechnet, indem je ein Turtle von jeder Fläche in jede andere Fläche geht. Ein Counter zählt, wie viele Turtles Patches innerhalb einer Fläche passieren, bis sie ihr Ziel erreicht haben. Je höher der Counter, desto zentraler liegt die damit verbundene Funktion. Die optische Darstellung erfolgt über die Vergrößerung der mittigen Turtles (`_typ "0"`) je nach Zentralität sowie der farbigen Darstellung der von den Flächen belegten Patches (Abb. 32).

```
ask turtles with [shape = "target"] [set _goal ""]
ask students [set size 2
  set goal _goal
  ask turtles with [shape = "target" and _typ = 0 and _zahl = goal] [set zielrichtung who]
  face turtle zielrichtung
  while [distance turtle zielrichtung > 1] [
    ask patch-here [set oldroom _roomnumber]
    fd 1
    test-patches
  ]
]
```

²¹ Uri Wilensky, verfügbar unter: http://modelingcommons.org/browse/one_model/2330#model_tabs_browse_procedures [zuletzt abgerufen am 06.03.2016]

```
]
ask turtles with [shape = "target" and _typ = 0] [
  set size _passierende-tester / anzahl_rooms ]
```

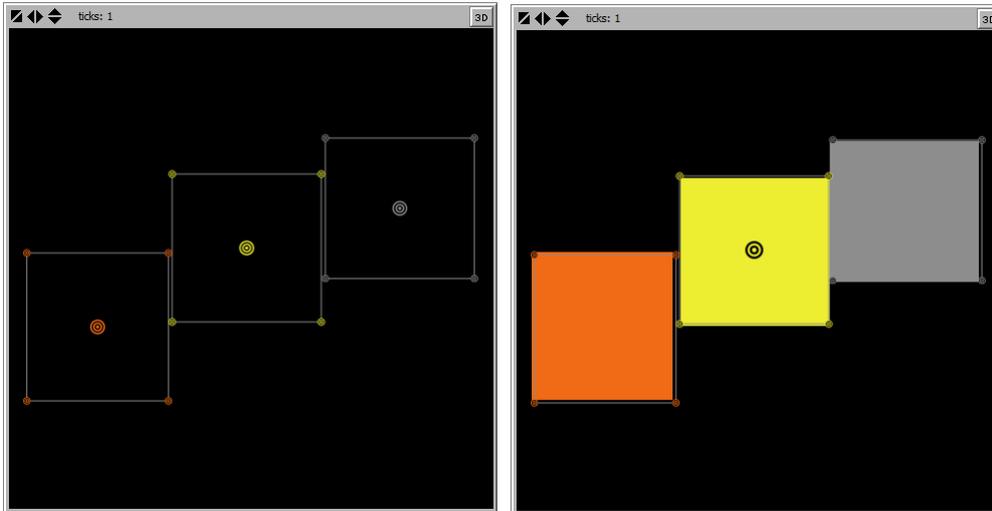


Abbildung 32 - Screenshot einer Lösung von drei Flächen und ihre Evaluierung zur Zentralität

4.10 Export

Der erzeugte Grundriss wird sowohl grafisch (ESP und PDF stehen zur Option) als auch als Werteliste (CSV) exportiert. Der grafische Export berücksichtigt die gesetzten Einstellungen zum Anzeigen von Adjazenzen und erzeugt jeweils das Ergebnis und seine Evaluierung. In der CSV findet sich der Wert aus der Evaluierung als weitere Eigenschaft des jeweiligen Raumes.

5. Diskussion

5.1 Charakteristik der Testlandschaft

Das hier verwendete Testbeispiel entstand in Kooperation mit Daniel Elsner im Zuge der Diplomarbeit "SimSchool - Optimierung der Abläufe in Schulbauten". In Elsners Arbeit wurden die Auslobungsunterlagen der Stadt Wien für den Bau eines Wiener Bildungscampus am Nordbahnhof genau analysiert und ein geschößweiser Funktionsverteilungs-algorithmus anhand des gegebenen Raumprogramms implementiert. Hierbei sind folgende Eigenschaften berücksichtigt worden:

- In einem ersten Schritt wurde anhand der Wege, die SchülerInnen in einer Schulwoche im Gebäude zurücklegen, ein Nachbarschaftsdiagramm erstellt, welches die räumlichen Beziehungen der unterschiedlichen Bereiche des Bildungscampus abbildet (Abb. 33).

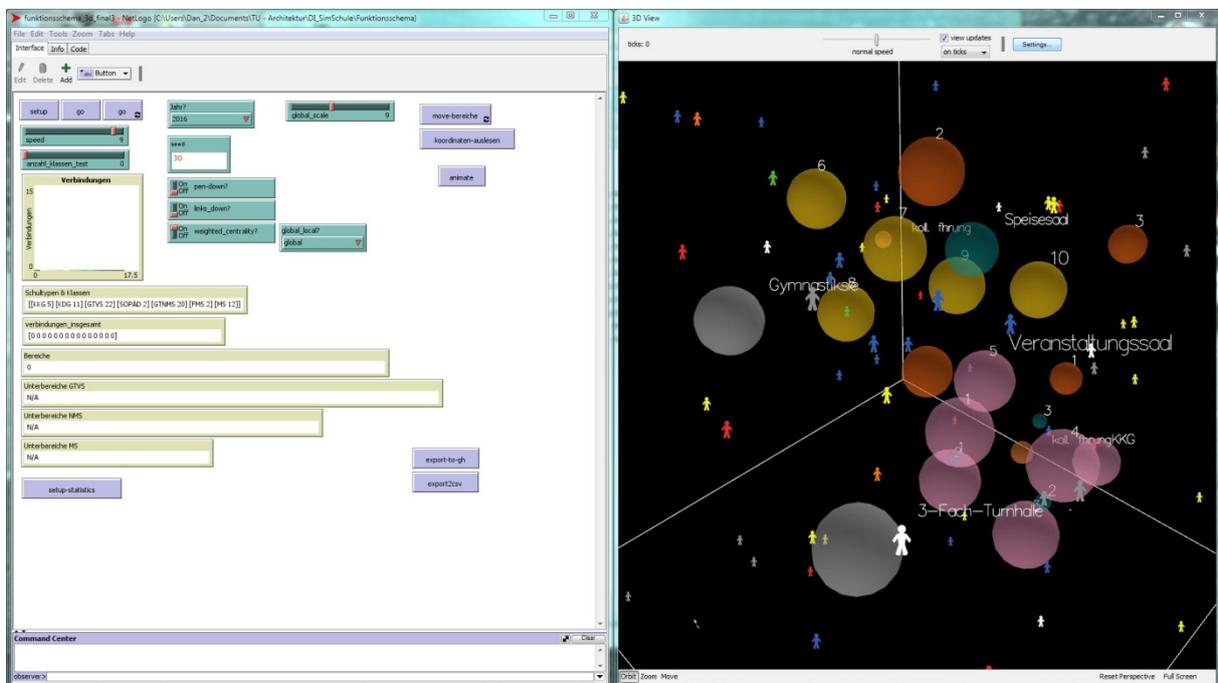


Abbildung 33 - Screenshot "SimSchool" Wegesimulationsprogramm²²

- Dieses Diagramm dient in einem nächsten Schritt als Grundlage zur Verteilung der Bereiche auf Geschoße (Abb. 34). Die zur Verfügung stehende Geschoßfläche wird vom Nutzer gesetzt und die benötigte Geschoßanzahl ergibt sich daraus.

- Der Algorithmus zur Verteilung der Funktionen basiert auf einer Kombination mehrerer Faktoren. Einer davon ist der vorgegebene hierarchische Aufbau des Bildungscampus in Über- und Unterbereiche (Abb. 35); ein weiterer die "Weighted Closeness-Centrality" der Bereiche, einem berechneten Wert, der ihre Zentralität in Abhängigkeit der Beziehungen im Nachbarschaftsdiagramm

²² Daniel Elsner, 2016 (in Druck), SimSchool - Optimierung der Abläufe in Schulbauten, Diplomarbeit, TU Wien, Fakultät für Architektur und Raumplanung

widerspiegelt. Ein letzter Faktor ist die Idee eines Hauptgeschoßes jedes Überbereichs, von dem aus keiner seiner Unterbereiche mehr als ein Geschoß entfernt sein darf.

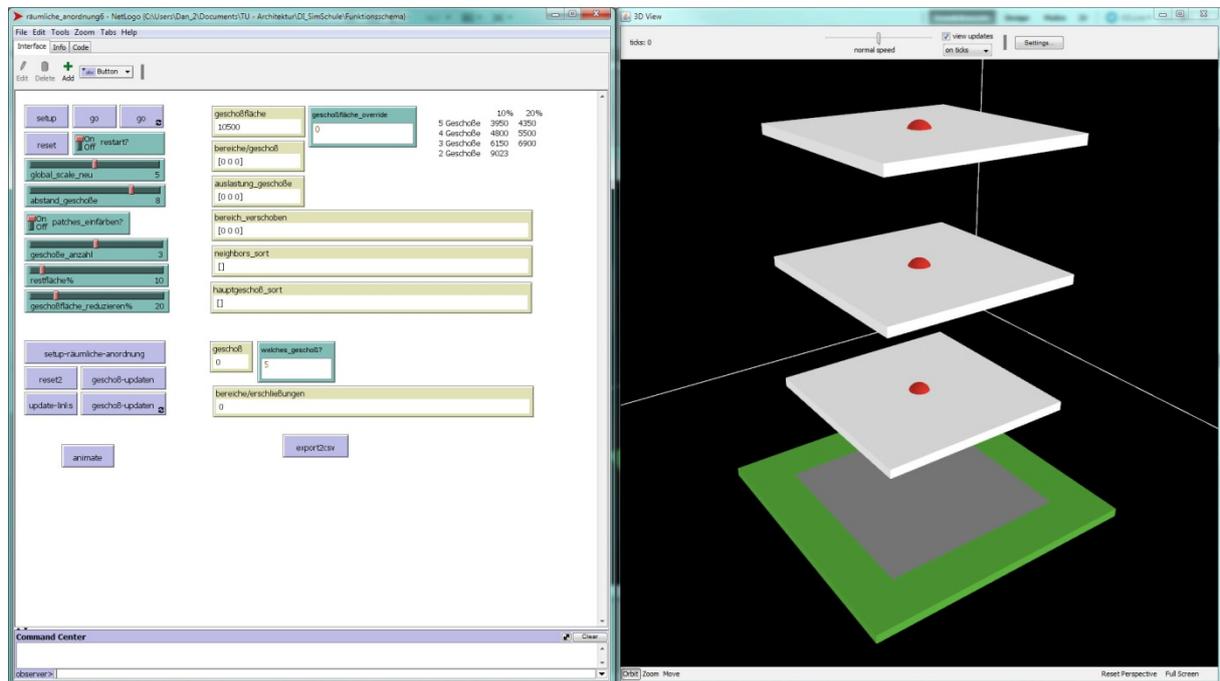


Abbildung 34 - Screenshot "SimSchool" Funktionsaufteilungsprogramm²³

- ein individuell gesetzter prozentualer Wert der Geschoßfläche bleibt mindestens für Restflächen frei, deren Funktionen in der Simulation keine Berücksichtigung finden.

Übergeben wurden insgesamt vier Lösungen mit zwei bis fünf Geschoßen. Diese enthalten eine Aufteilung der aus dem Diagramm ersichtlichen Funktionsflächen (exkl. in Abb. 35 blau dargestellter Musikschule, die durch Bildungsbereich 10 und Kreativbereich 3 ersetzt wurden) unter Berücksichtigung ihrer Größe sowie jeweils mindestens zehn Prozent Freifläche, unter anderem für die Erschließung der Bereiche untereinander. Daraus ergibt sich für das in Kapitel 4 beschriebene Programm mit den neun standardmäßigen Typologien eine Bandbreite von 36 skizzenhaften Gebäuden bzw. 126 automatisiert erzeugten unterschiedlichen Geschoßlösungen, noch ohne die Möglichkeit von Interaktion und unterschiedlicher Parametersetzung für Einzelergebnisse in Betracht gezogen zu haben. Je eine mögliche Lösung zu den 36 Gebäuden findet sich in Punkt 5.2 dargestellt. Über den Datenimport kommt es zu keiner negativen Nachbarschaftsbeziehung, weshalb zu Testzwecken die durchaus legitime Abstoßung von Speisesaal und Kollegialer Führung ergänzt wurde, um das Verhalten einer solchen Negativbeziehung ebenfalls testen zu können.

²³ Daniel Elsner, 2016 (in Druck), SimSchool - Optimierung der Abläufe in Schulbauten, Diplomarbeit, TU Wien, Fakultät für Architektur und Raumplanung

Die in der Diskussion teilweise angemerkten Einzellösungen sind mit Geschoßanzahl, Typologie und tatsächlichem Geschoß bezeichnet. Für Geschoß 0 der Lösung "H" in der dreigeschoßigen Variante findet man dementsprechend die Abkürzung 3-H-0 vor.

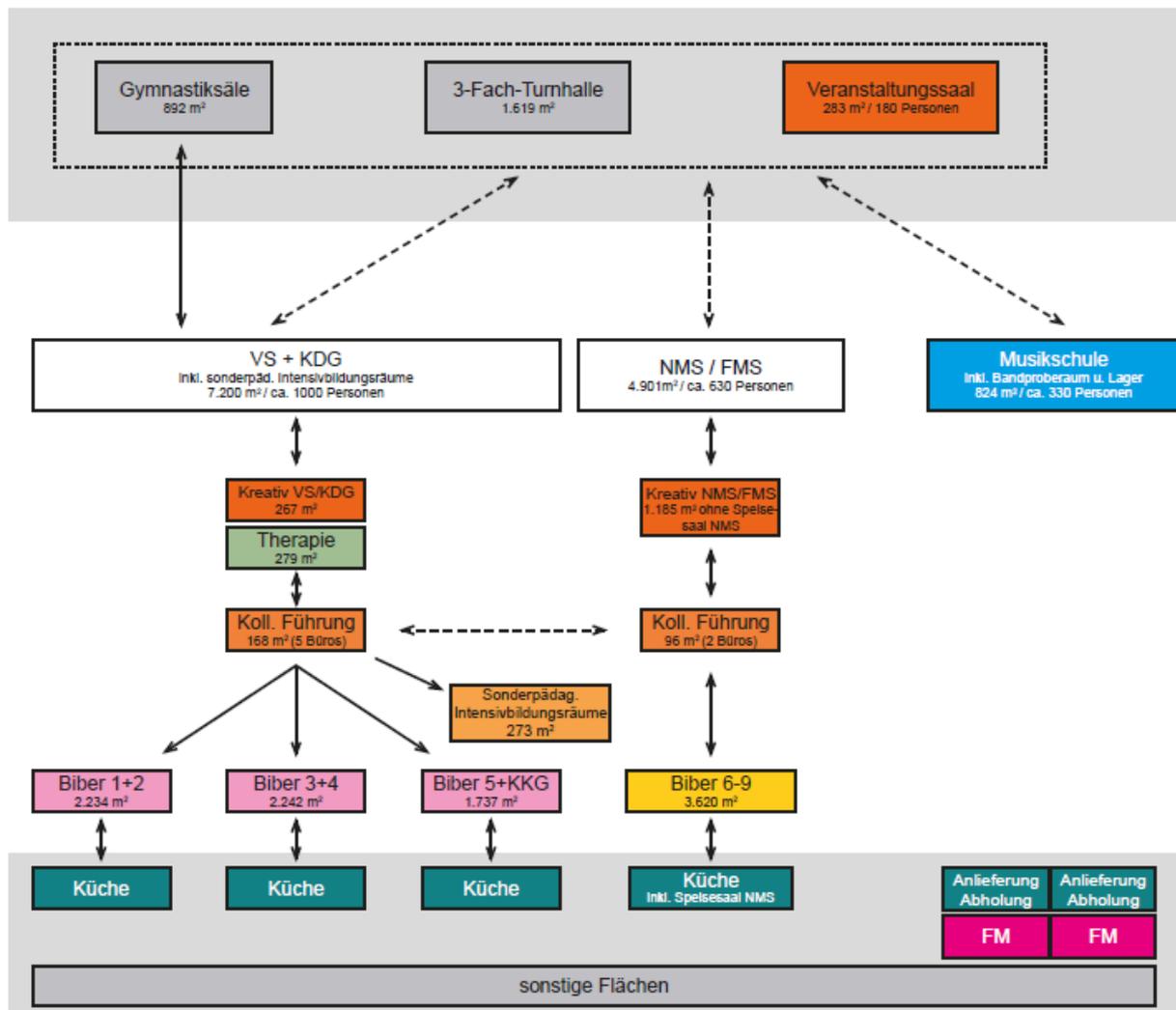


Abbildung 35 - Diagramm zur hierarchischen Gliederung des Bildungscampus²⁴

Die daraus abgeleiteten Funktionseinheiten, die zuerst von Elsners Programm geschoßweise verteilt und anschließend in dieser Arbeit pro Geschoß angeordnet werden, sind in Tabelle 2 vollständig aufgelistet. Wie daraus ersichtlich ist, werden nebst der internen und externen Erschließung in die prozentual festgelegte Restfläche auch die Flächen für Technik, Facility Management, Therapie und Sonderpädagogik gezählt.

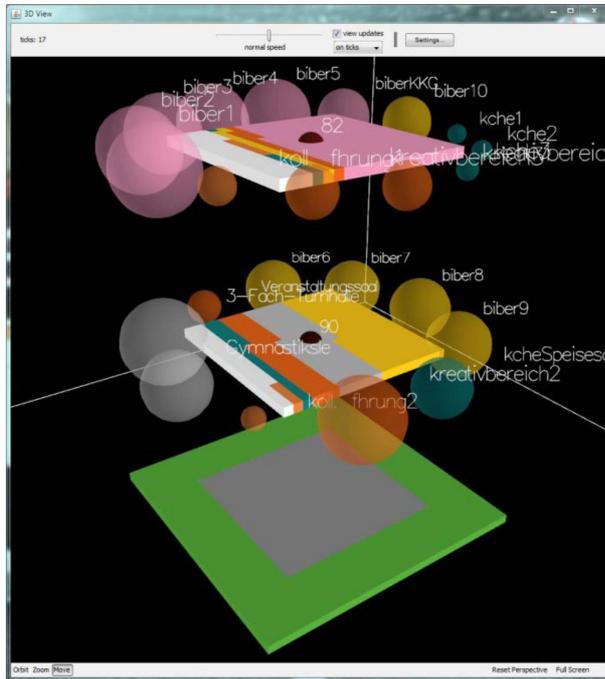
²⁴ Magistrat der Stadt Wien, Magistratsabteilung 19, 2015, Neubau Bildungscampus Nordbahnhof 1020 Wien - Auslobung, Wien, 18.11.2015

| Bereich | Funktionsfläche | Farbe |
|--------------------------------------|--------------------|-------|
| 3-Fach-Turnhalle | 3-Fach-Turnhalle | |
| Gymnastiksäle | Gymnastiksaale | |
| Neue Mittelschule / Fachmittelschule | biber6 | |
| Neue Mittelschule / Fachmittelschule | biber7 | |
| Neue Mittelschule / Fachmittelschule | biber8 | |
| Neue Mittelschule / Fachmittelschule | biber9 | |
| Neue Mittelschule / Fachmittelschule | koll. fuehrung2 | |
| Neue Mittelschule / Fachmittelschule | kreativbereich2 | |
| Neue Mittelschule / Fachmittelschule | kuecheSpeisesaal | |
| Veranstaltungssaal | Veranstaltungssaal | |
| Volksschule / Kindergarten | biber1 | |
| Volksschule / Kindergarten | biber2 | |
| Volksschule / Kindergarten | biber3 | |
| Volksschule / Kindergarten | biber4 | |
| Volksschule / Kindergarten | biber5 | |
| Volksschule / Kindergarten | biberKKG | |
| Volksschule / Kindergarten | koll. fuehrung1 | |
| Volksschule / Kindergarten | kreativbereich1 | |
| Volksschule / Kindergarten | kueche1 | |
| Volksschule / Kindergarten | kueche2 | |
| Volksschule / Kindergarten | kueche3 | |
| Musikschule | biber10 | |
| Musikschule | kreativbereich3 | |

Tabelle 2 - verwendetes Funktionsflächenprogramm mit Farbkennzeichnung

5.2 Analyse des gesamten Lösungssatzes

Layouts mit 2 Geschoßen - Input



Geschoßfläche: 9315m²

Ebene 0: 10 Funktionsbereiche

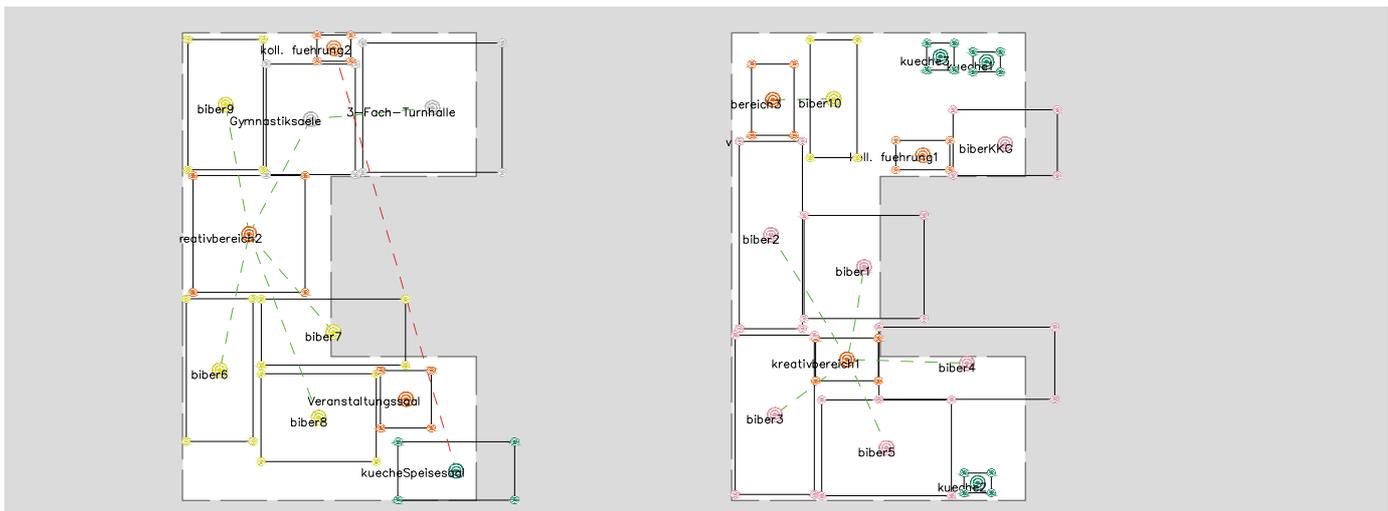
Ebene 1: 13 Funktionsbereiche

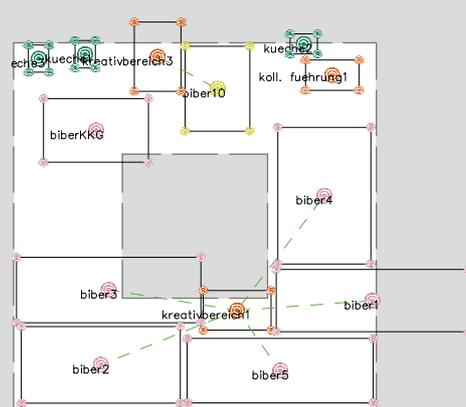
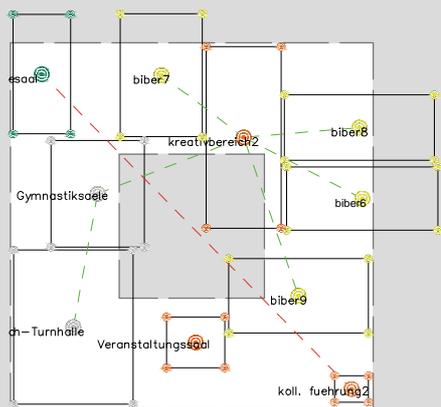
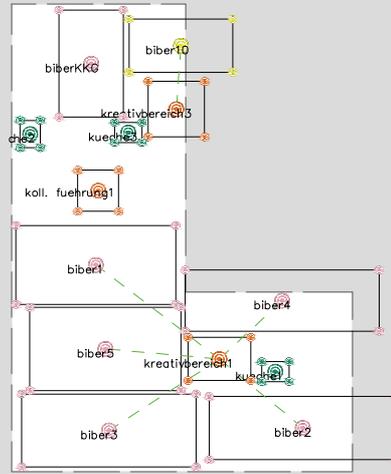
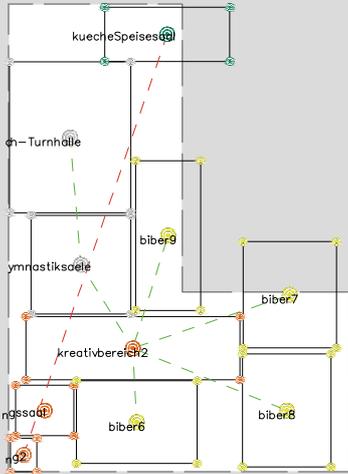
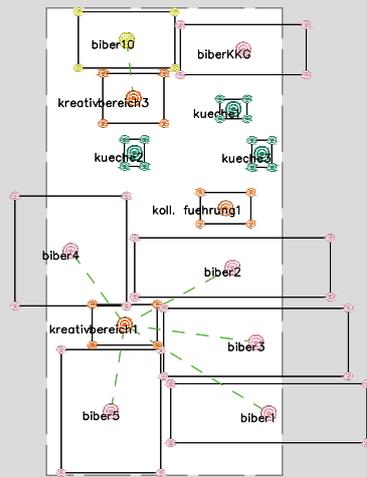
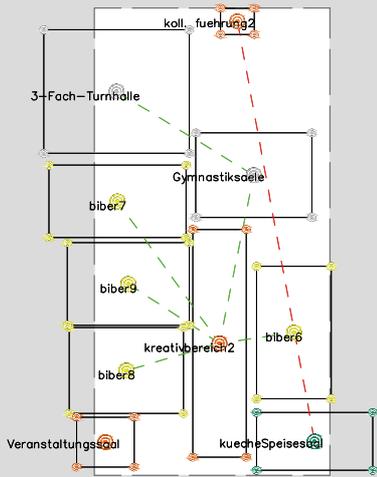
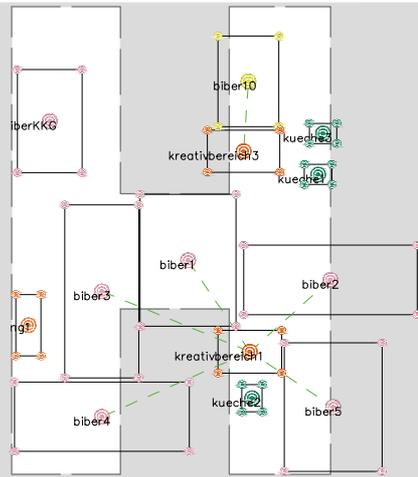
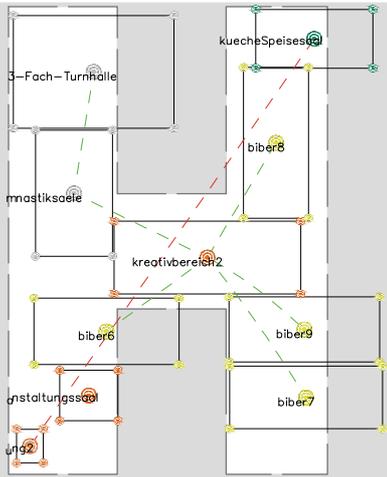
Eigenschaften:

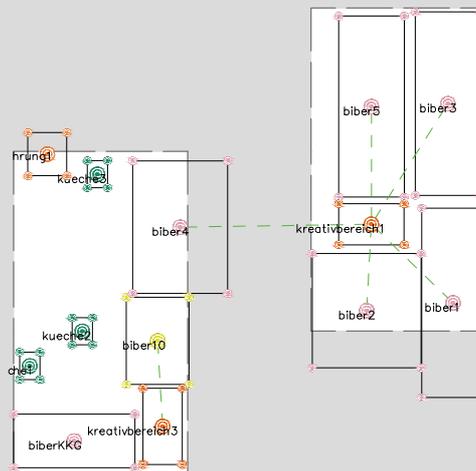
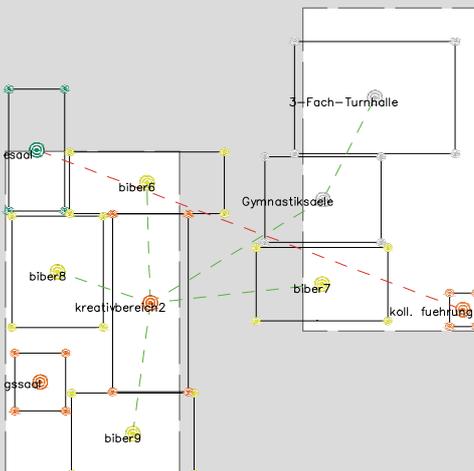
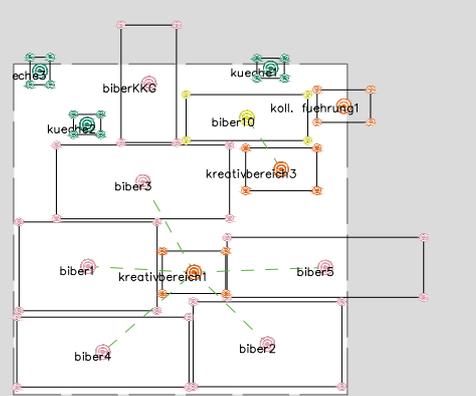
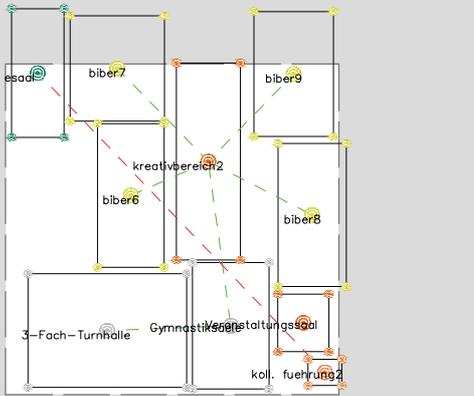
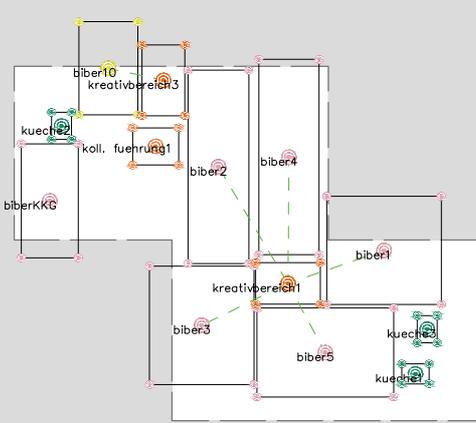
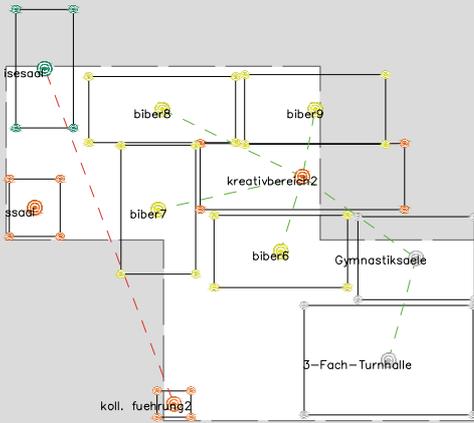
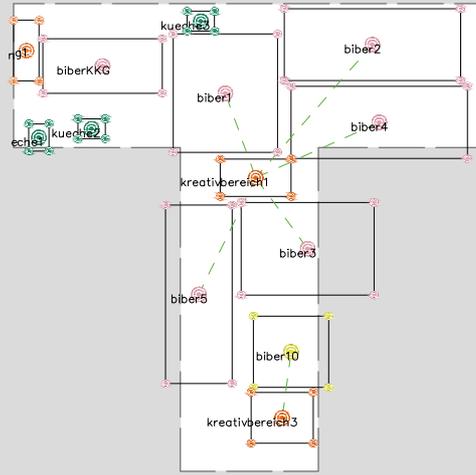
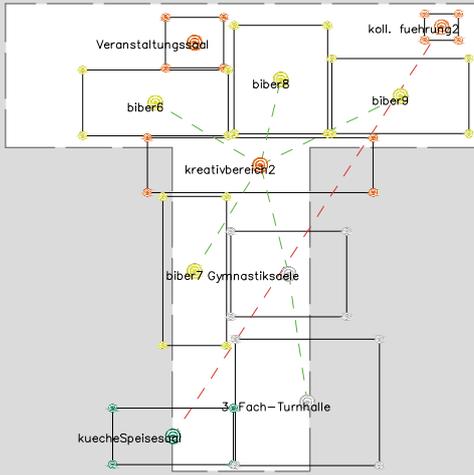
- die meisten Funktionsbereiche pro Ebene
- dadurch entstehen viele und intensive Nachbarschaftsverhältnisse

Abbildung 36 - mögliche Aufteilung für zwei Geschoße von Daniel Elsner

Layouts mit 2 Geschoßen - Output



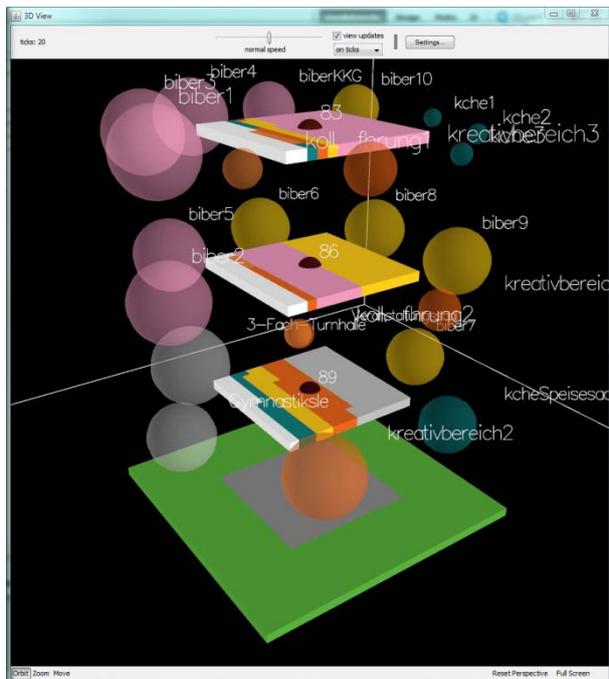




Layouts mit 2 Geschoßen - Fazit

Der Algorithmus kommt mit der geforderten Zielvorgabe zurecht und findet für alle Typologien eine Lösung. Auch bei komplexen bzw. nicht vollständig zufriedenstellend lösbaren Typologien (2-Pav-0 oder 2-Pav-1) wird eine Lösung gefunden, die alle gegebenen Anforderungen letztendlich so gut es geht erfüllt. Die negative Adjazenz in Ebene 0 fällt dem Programm besonders leicht zu bewältigen. Die zwei sternförmigen Beziehungsmittelpunkte in Ebene 0 und Ebene 1 haben zwar unterschiedliche Größen, die Anordnung der Bildungsbereiche erfolgt jedoch in beiden Fällen stimmig. Adjazenzen werden nach ihrer Länge bearbeitet. Das bedeutet auch, dass nicht zusammenhängende Gebäudekomplexe teilweise als gute Nachbarschaft beurteilt werden (2-H-1). Gerade dadurch ist eine Anregung zur Kreativität des Planers zu finden.

Layouts mit 3 Geschoßen - Input



Geschoßfläche: 6200m²

Ebene 0: 10 Funktionsflächen

Ebene 1: 6 Funktionsflächen

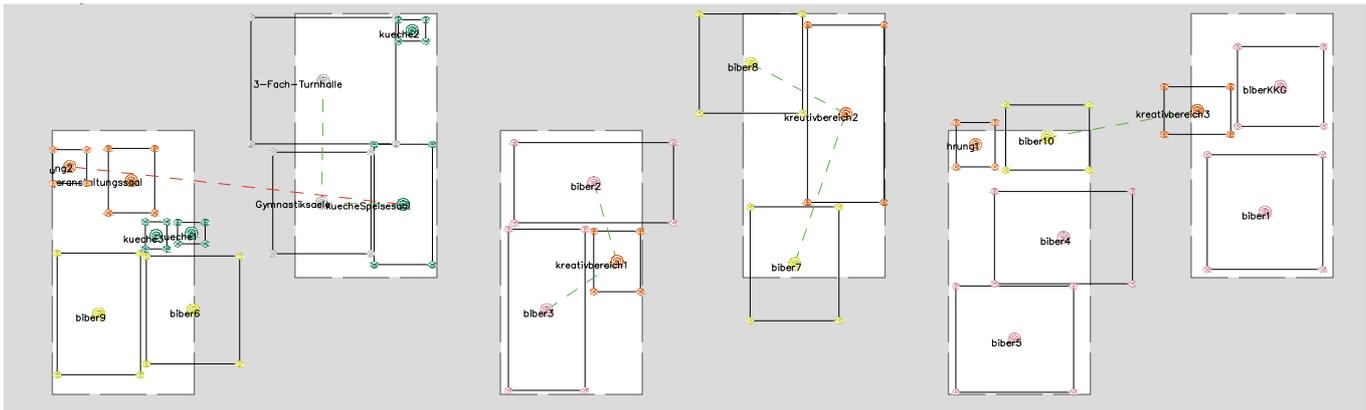
Ebene 2: 7 Funktionsflächen

Eigenschaften:

- Ebene 1 hat je zwei Dreier-Adjazenzen
wohingegen Ebene 2 nur eine Adjazenz enthält
- Ebene 0 beinhaltet sowohl die größten als
auch kleinsten Funktionsflächen

Abbildung 37 - mögliche Aufteilung für drei Geschoße von Daniel Elsner

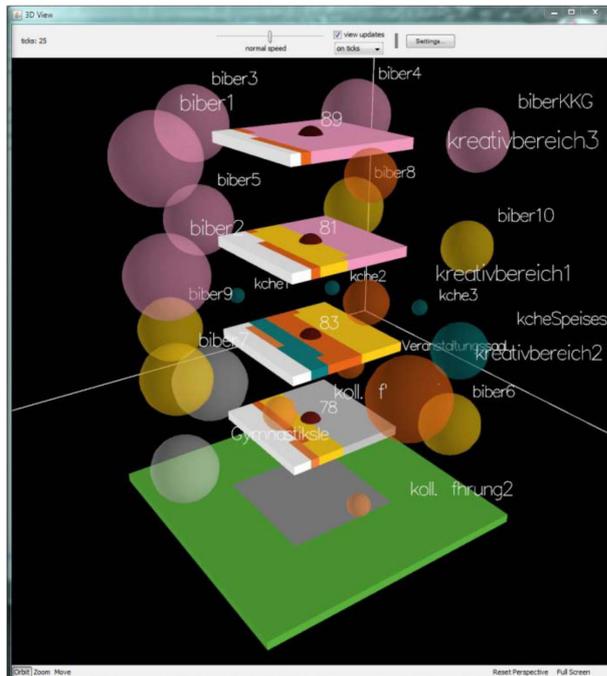




Layouts mit 3 Geschossen - Fazit

Es fällt auf, dass das Programm mit mehr Adjazenzen zu, auf den ersten Blick, geordneteren Ergebnissen gelangt (vergleiche Ebene 1 mit Ebene 2). Bei Ebene 1 entstehen zwei Blöcke aus jeweils einem Kreativbereich und seinen angeschlossenen Bildungsbereichen. Diese Trennung fällt besonders bei Grundrissen auf, die eine solche Entwicklung unterstützen (3-C-1, 3-H-1, 3-Pav-1). Bei Lösung 3-O-0 "schummelt" das Programm - für den Algorithmus liegen alle fünf Agenten der Fläche 3-Fach-Turnhalle innerhalb der Typologie.

Layouts mit 4 Geschossen - Input



Geschoßfläche: 4829m²

Ebene 0: 5 Funktionsflächen

Ebene 1: 8 Funktionsflächen

Ebene 2: 5 Funktionsflächen

Ebene 3: 5 Funktionsflächen

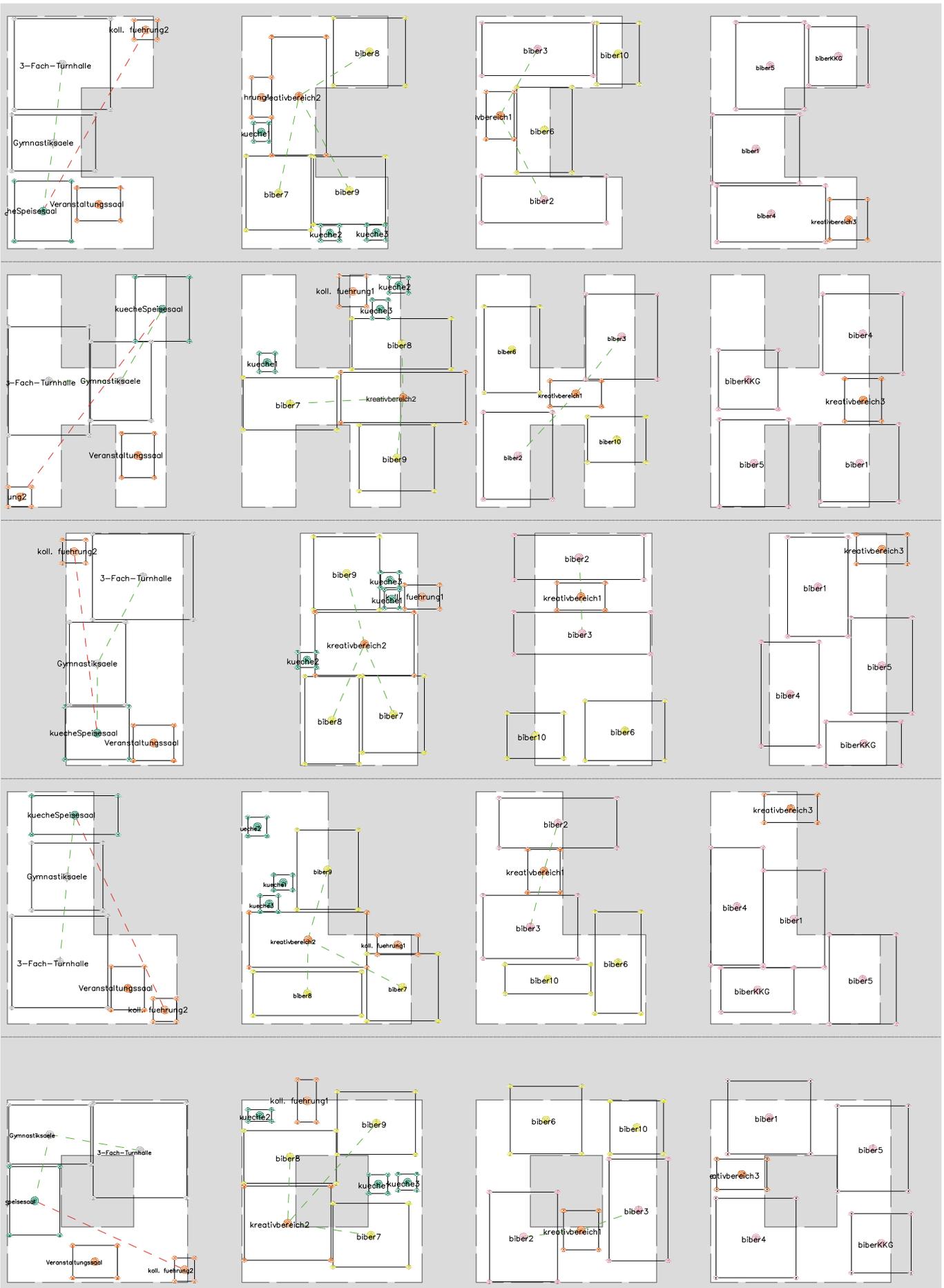
Eigenschaften:

- Ebene 1 weist eine einfache, kleine

Sternadjazenz auf, Ebene 3 keinerlei Adjazenz

Abbildung 38 - mögliche Aufteilung für vier Geschosse von Daniel Elsner

Layouts mit 4 Geschoßen - Output





Layouts mit 4 Geschoßen - Fazit

Die geringe Belegung von 74% der Ebene 0 fällt optisch nur in Lösung 4-O-0, 4-T-0 und 4-Pav-0 auf. Die Sterntopologie der Adjazenz der größten Flächen und die zufällig gesetzten Startpositionen führen bei 4-H-1 zu einer besonders eigenwilligen Lösung. Interessant ist die Entwicklung und der Einfluss von Adjazenzen bei den I-Ebenen. 4-I-0 und 4-I-1 weisen gewisse charakteristische Ähnlichkeiten auf wohingegen die Adjazenz und eine anfängliche Überlappung bei 4-I-2 im oberen Bereich zu länglichen Flächen führt, die der Typologie entgegengesetzt arbeiten.

Layouts mit 5 Geschoßen - Input

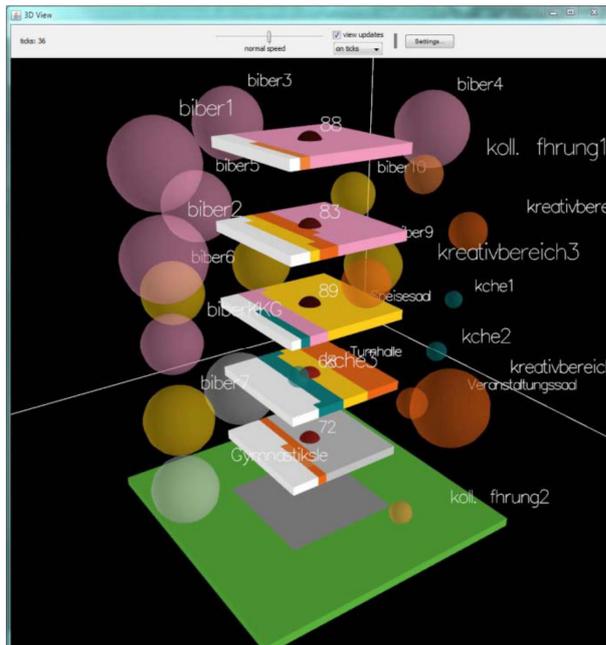


Abbildung 39 - mögliche Aufteilung für fünf Geschoße von Daniel Elsner

Geschoßfläche: 3989m²

Ebene 0: 5 Funktionsflächen

Ebene 1: 3 Funktionsflächen

Ebene 2: 7 Funktionsflächen

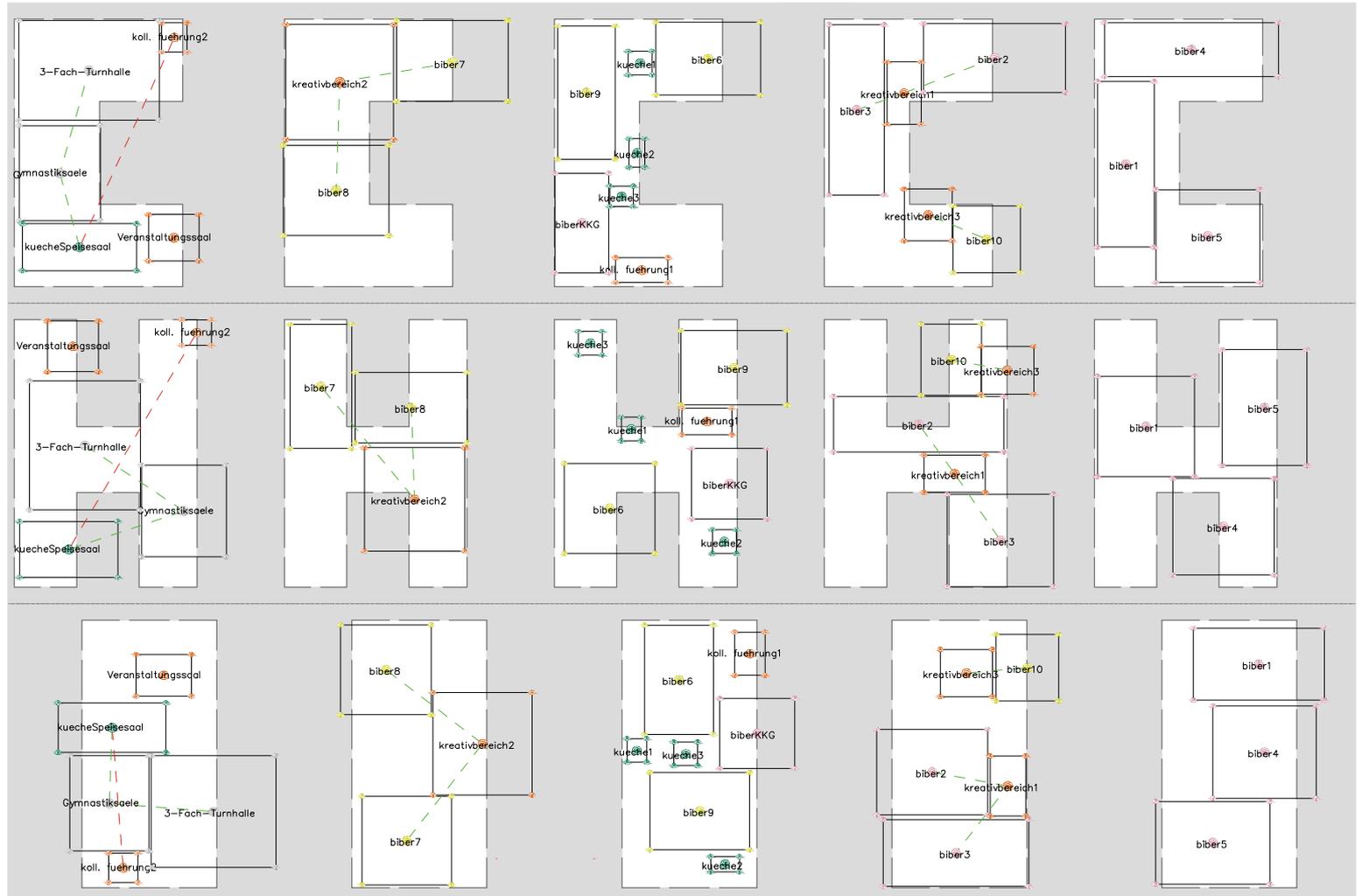
Ebene 3: 5 Funktionsflächen

Ebene 4: 3 Funktionsflächen

Eigenschaften:

- zwei Ebenen mit je nur drei Flächen

Layouts mit 5 Geschoßen - Output





Layouts mit 5 Geschoßen - Fazit

Durch die Skalierung der Patches auf die Gesamtgeschoßfläche (siehe Kapitel 4.4) werden besonders große Räume mit geringem Proportionsspielraum vom Programm nicht so verteilt, wie ein Planer die Aufteilung organisieren würde (5-O-0), auch wenn sich das Programm nach wie vor an seine gesetzten Regeln hält und für alle Varianten eine für das Programm zufriedenstellende Lösung findet. Interessant ist hier auch der Einfluss der Adjazenzen (vergleiche 5-O-1 und 5-O-4 bzw. 5-Pav-1). Die geringe Flächenanzahl veranlasst das Programm dazu, sehr schnell mit einer Lösung zufrieden zu sein, was zu einem optisch hohen Anteil an Restfläche sowie kaum erkennbaren Typologieformen führt (siehe 5-H-1 und 5-H-4).

Um die Lösungen global und konkret miteinander zu vergleichen, werden die Bewertungen für Adjazenz und Einhalten der Typologie näher miteinander verglichen. Die Überschneidungsfreiheit ist bei allen Lösungen zu 100% gegeben, was mitunter auch den stets freibleibenden "Restflächen" zugute zu schreiben ist. Würde man eine vollflächige Anordnung wünschen und implementieren wollen, sei erneut auf die umfangreichen Erkenntnisse aus Reinhard Königs Studien zum Thema "Dichte Packung"²⁵ verwiesen.

Tabelle 2 bildet die positiven, nach der Gesamtanzahl im jeweiligen Grundriss gewichteten und skalierten Beziehungsdistanzen der Mittelpunkte der Grundriss-Skizzen vollständig ab. Ebenen mit ausschließlich neutralen Beziehungen sind in der Tabelle leer. Da nur die positiven Distanzen berücksichtigt werden, gilt ein kleinerer Wert als besseres Ergebnis. Im Mittel liegen die Typologie O und Pavillon am schlechtesten ab (16,26 und 17,03).

Knapp aneinander liegen die bestgeeignetsten Typologien mit T, Square und I. Klar geht hervor, dass kompakte Typologien in diesem Programm zu kürzeren gewünschten Verbindungswegen führen, auch wenn diese Wege auch außerhalb der Typologie verlaufen können. Auffallend ist, dass Lösung 3 mit den sich glücklich ergebenden Adjazenzen im Vergleich am besten abschneidet, obwohl aufgrund der geringen Flächen pro Ebene die Vermutung nahegelegen hätte, bei Lösung 5 das beste Ergebnis zu erzielen.

Eine große Fehleranfälligkeit im Bereich des Problemwissens ist jedoch in den Adjazenzen zu sehen: Wo können bei unerwünschter räumlicher Nähe nicht genauso gut bauliche und oder planerische

²⁵ Reinhard König, 2011, Generierung von Grundriss-Layouts mittels Hybrider Evolutionsstrategie, aus: Arbeitspapiere Nr. 4, März 2011, Bauhaus Universität Weimar, sowie Reinhard König, 2011, KREMLAS Entwicklung einer kreativen evolutionären Entwurfsmethode für Layoutprobleme in Architektur und Städtebau, 2011, Hrsg. Dirk Donath, Reinhard König und Frank Petzold, Bauhaus Universität Weimar

alternative Maßnahmen, wie zum Beispiel Schallschutzkonzepte oder Lüftungssysteme, die Trennung zufriedenstellend herstellen? Gegengleich sind inszenierte längere Wege oder das Erzeugen von Sichtachsen vielleicht für positive Nachbarschaften, die nicht auf reine wirtschaftliche Ablaufoptimierung abzielen, eine größere Aufwertung als eine serielle Anordnung.

Ein weiteres Problem ist direkt mit dem Anwendungsbeispiel verbunden: Die gesetzten Naheverhältnisse können nicht "ersetzt" werden, was sich bei der behandelten Raumstruktur, in der viele Bereiche gleichwertig und gegeneinander austauschbar sind, negativ auswirkt: Hier wäre eine Systematik, in der sich z.B. je zwei etwa gleich große Bildungsbereiche einen Kreativbereich teilen - ohne Einschränkungen, um welche Bildungsbereiche oder Kreativbereiche es sich im Speziellen handelt - viel wesentlicher als eine fixe Zuordnung. Eine solche Bedingungsstruktur lässt sich mit der hier verwendeten Matrix nicht abbilden. Diese müsste dynamisch während des Lösungsfindungsprozesses geändert werden oder ein vollkommene anderes Beziehungssystem gewählt werden.

| Lösung | Ebene | C | H | I | L | O | Pav | Squ | T | Z | Mittel | Mittel GESAMT |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|------------------|
| 2 | 0 | 19,62 | 18,55 | 18,18 | 17,31 | 18,74 | 19,17 | 17,14 | 16,93 | 16,03 | 18,21 | |
| | 1 | 14,16 | 16,36 | 15,25 | 15,19 | 16,08 | 16,93 | 14,45 | 15,86 | 13,72 | 15,54 | 16,87 |
| | Mittel | 16,89 | 17,46 | 16,72 | 16,25 | 17,41 | 18,05 | 15,80 | 16,40 | 14,88 | | |
| 3 | 0 | 13,49 | 19,39 | 16,92 | 16,41 | 24,38 | 14,11 | 14,51 | 17,60 | 18,20 | 17,10 | |
| | 1 | 16,92 | 13,36 | 14,46 | 13,30 | 15,77 | 14,99 | 13,44 | 15,65 | 14,59 | 14,74 | |
| | 2 | 10,87 | 10,40 | 10,23 | 8,83 | 10,49 | 21,00 | 14,00 | 7,77 | 8,31 | 11,70 | 14,51 |
| | Mittel | 13,76 | 14,39 | 13,87 | 12,85 | 16,88 | 16,70 | 13,98 | 13,67 | 13,70 | | |
| 4 | 0 | 14,92 | 15,54 | 16,86 | 14,86 | 15,97 | 15,00 | 16,64 | 17,86 | 15,08 | 15,96 | |
| | 1 | 19,38 | 15,21 | 15,56 | 15,56 | 18,99 | 18,58 | 14,21 | 14,41 | 20,04 | 16,49 | |
| | 2 | 17,17 | 15,85 | 7,65 | 10,82 | 11,88 | 10,84 | 11,15 | 12,17 | 12,58 | 12,19 | |
| | 3 | | | | | | | | | | | 14,88 |
| | Mittel | 17,16 | 15,54 | 13,36 | 13,75 | 15,61 | 14,81 | 14,00 | 14,81 | 15,90 | | |
| 5 | 0 | 14,58 | 20,92 | 14,27 | 24,96 | 17,98 | 18,34 | 14,57 | 15,43 | 17,50 | 17,63 | |
| | 1 | 16,01 | 18,93 | 19,36 | 16,18 | 17,29 | 23,87 | 17,13 | 15,17 | 15,65 | 17,99 | |
| | 2 | | | | | | | | | | | |
| | 3 | 11,07 | 10,50 | 11,74 | 12,14 | 11,26 | 14,52 | 12,59 | 10,73 | 13,81 | 11,82 | 15,81 |
| | 4 | | | | | | | | | | | |
| Mittel | 13,89 | 16,78 | 15,12 | 17,76 | 15,51 | 18,91 | 14,76 | 13,78 | 15,65 | | | |
| Mittel | 15,29 | 15,91 | 14,59 | 15,05 | 16,26 | 17,03 | 14,53 | 14,51 | 15,05 | | | |
| | | | | | | | | | | | | 3 längste |
| | | | | | | | | | | | | 3 kürzeste |
| | | | | | | | | | | | | krz. Mittel |

Tabelle 3 - gewichtete positive Adjazenzlängen, die Einheit der Länge entspricht einem Patch

Letztendlich gilt für eine Angabe zu Beziehungsnetzen in diesem Fall, die besten Ergebnisse mit einem ausgewogenen Maß an wohlüberlegten, bewusst gewählten Angaben zu erhalten. Einige neutrale Verbindungen tragen dazu bei, die wirklich wichtigen, primär organisatorisch begründeten Beziehungen auch umsetzen zu können.

In einem randomisierten Beispiel mit 20 Räumen, Proportionsmaximum 0.5 (Abb. 40) sind im ersten Viertel der Zeit zufällige Beziehungswerte gesetzt. Die Anzahl der Schnittstellen nimmt ab dem Zeitpunkt rapide ab, ab dem die Beziehungswerte nicht mehr berücksichtigt werden. Gleichfalls geben gezielte Angaben zu einigen (un-)erwünschten Naheverhältnissen dem Programm den Anreiz und die Anforderung, die Lösungen zu konkretisieren, ohne dabei den offenen, frei interpretierbaren Charakter aufgeben zu müssen.

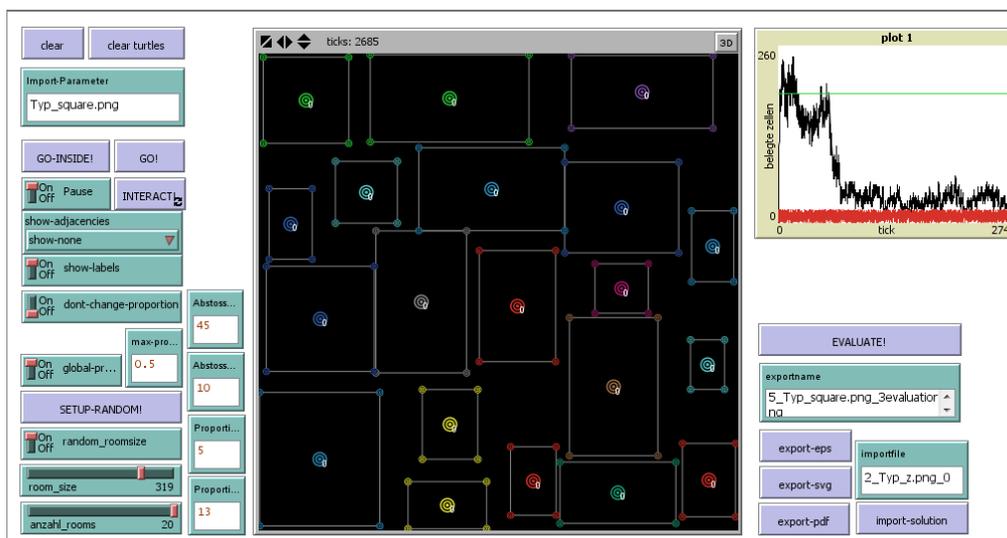


Abbildung 40 - Screenshot zur Performancebeeinflussung komplizierter Adjazenzen

Für die Größe der innerhalb einer Typologie gelegenen Bestandteile der Flächen werden die durch die Raumaufteilung von Daniel Elsner maximal belegten Gesamtpatches herangezogen, um ein gewichtetes prozentuales Verhältnis zu erhalten. Hierbei sei erwähnt, dass es aufgrund der Voxelisierung²⁶ in Netlog zu Abweichungen im Bereich von je +/- 1 Patchreihe kommen kann. Es muss auch berücksichtigt werden, dass ein niedriger Wert nicht gleichwertig mit einer schlechteren Lösung anzusehen ist, da das Programm darauf ausgelegt ist, nur bis zu einem Anteil von mindestens drei Flächenpunkten einen Drang nach einer Bewegung verspürt und damit auch ein Anreiz zur Erzeugung einer kreativen, unerwarteten Lösung geschaffen werden soll.

²⁶ Die Problematik der Voxelisierung ist ausführlich dokumentiert zu finden in W Lorenz, G. Wurzer, S. Swoboda (ed.), flying bricks: algorithmisches entwerfen, Institut für Architekturwissenschaften, Abteilung Digitale Architektur und Raumplanung 259.1, Wien, 2015

| Lö. | Eb. | Belegt | C | H | I | L | O | Pav | Squ | T | Z | Mittel | Mittel GESAMT |
|-----|-----------|-----------|------|------|------|------|------|------|------|------|------|--------|------------------|
| | | | 2100 | 2125 | 2080 | 2110 | 2130 | 2115 | 2116 | 2109 | 2107 | 2110 | |
| 2 | 0 | 89% | 1606 | 1362 | 1320 | 1610 | 1329 | 1296 | 1434 | 1268 | 1362 | 1399 | |
| | | gewichtet | 86% | 72% | 71% | 86% | 70% | 69% | 76% | 68% | 73% | 74% | |
| | 1 | 82% | 1326 | 1151 | 1186 | 1317 | 1388 | 1273 | 1461 | 1379 | 1376 | 1317 | 1358 |
| | | gewichtet | 77% | 66% | 70% | 76% | 79% | 73% | 84% | 80% | 80% | 76% | 75% |
| | | Mittel | 81% | 69% | 70% | 81% | 75% | 71% | 80% | 74% | 76% | | |
| 3 | 0 | 89% | 1411 | 1423 | 1383 | 1544 | 1205 | 1361 | 1520 | 1343 | 1369 | 1395 | |
| | | gewichtet | 75% | 75% | 75% | 82% | 64% | 72% | 81% | 72% | 73% | 74% | |
| | 1 | 89% | 1459 | 1343 | 1263 | 1500 | 1564 | 1462 | 1484 | 1338 | 1486 | 1433 | |
| | | gewichtet | 78% | 71% | 68% | 80% | 83% | 78% | 79% | 71% | 79% | 76% | |
| | 2 | 79% | 1400 | 1296 | 1269 | 1198 | 1196 | 1372 | 1427 | 1347 | 1289 | 1310 | 1380 |
| | | gewichtet | 84% | 77% | 77% | 72% | 71% | 82% | 85% | 81% | 77% | 79% | 76% |
| | | Mittel | 79% | 74% | 73% | 78% | 72% | 77% | 82% | 75% | 77% | | |
| 4 | 0 | 73% | 1316 | 1224 | 1346 | 1309 | 1313 | 1042 | 1236 | 1278 | 1151 | 1246 | |
| | | gewichtet | 86% | 79% | 89% | 85% | 84% | 67% | 80% | 83% | 75% | 81% | |
| | 1 | 88% | 1500 | 1073 | 1687 | 1341 | 1431 | 1294 | 1482 | 1479 | 1425 | 1412 | |
| | | gewichtet | 81% | 57% | 92% | 72% | 76% | 70% | 80% | 80% | 77% | 76% | |
| | 2 | 81% | 1337 | 1318 | 1257 | 1220 | 1311 | 1223 | 1466 | 1281 | 1283 | 1300 | |
| | | gewichtet | 79% | 77% | 75% | 71% | 76% | 71% | 86% | 75% | 75% | 76% | |
| | 3 | 88% | 1517 | 1315 | 1411 | 1454 | 1454 | 1495 | 1409 | 1552 | 1430 | 1449 | 1352 |
| | gewichtet | 82% | 70% | 77% | 78% | 78% | 80% | 76% | 84% | 77% | 78% | 78% | |
| | | Mittel | 82% | 71% | 83% | 77% | 79% | 72% | 80% | 80% | 76% | | |
| 5 | 0 | 89% | 1628 | 1191 | 1194 | 1453 | 1302 | 1326 | 1300 | 1489 | 1380 | 1363 | |
| | | gewichtet | 78% | 63% | 64% | 77% | 69% | 70% | 69% | 79% | 74% | 71% | |
| | 1 | 75% | 1161 | 1161 | 1149 | 1314 | 1116 | 1174 | 1291 | 1246 | 1399 | 1223 | |
| | | gewichtet | 74% | 73% | 74% | 83% | 70% | 74% | 81% | 79% | 89% | 77% | |
| | 2 | 70% | 1197 | 959 | 1179 | 1146 | 1114 | 1145 | 1128 | 1082 | 1340 | 1143 | |
| | | gewichtet | 81% | 64% | 81% | 78% | 75% | 77% | 76% | 73% | 91% | 77% | |
| | 3 | 83% | 1193 | 1093 | 1406 | 1469 | 1118 | 1474 | 1369 | 1070 | 1282 | 1275 | |
| | | gewichtet | 68% | 62% | 81% | 84% | 63% | 84% | 78% | 61% | 73% | 73% | |
| 4 | 84% | 1509 | 1140 | 1387 | 1302 | 1243 | 1379 | 1437 | 1475 | 1461 | 1370 | 1275 | |
| | gewichtet | 86% | 64% | 79% | 73% | 69% | 78% | 81% | 83% | 83% | 77% | 75% | |
| | | Mittel | 77% | 65% | 76% | 79% | 69% | 77% | 77% | 75% | 82% | | |
| | | Mittel | 1397 | 1194 | 1328 | 1347 | 1266 | 1293 | 1362 | 1324 | 1346 | | |
| | | | 80% | 69% | 78% | 78% | 73% | 75% | 79% | 77% | 78% | | |
| | | | | | | | | | | | | | 3 höchste |
| | | | | | | | | | | | | | 3 niedrigste |
| | | | | | | | | | | | | | höchst. Mit |

Tabelle 4 - Typologiebelegung in gewichteten Patches und Prozent

Auch in diesem Vergleich schneiden die kompakten Typologien wertmäßig besser ab, obwohl die Spannweite der oberen zwei Drittel insgesamt sehr nahe aneinanderliegt (77% bis 80%). Vergleichsweise weit schlagen hier die verbleibenden Typologien H (69%), O (73%) sowie Pavillon

(75%) ab. Bei genauerer Betrachtung der Daten liegt der Rückschluss nahe, dass hier wirklich die Typologie den Ausschlag gibt und die importierten Daten darauf weniger Einfluss nehmen: Auch wenn Lösung 5-3 zwei der Lösungen mit geringster Belegung aufbringt, liegt sie mit zwei weiteren Lösungen (L und Pavillon) mit 84% durchaus im oberen Bereich.

Ein Zusammenhang zwischen Belegtheitsgrad der Fläche und tatsächlicher Belegung schlägt sich in den Zahlen ebenfalls nieder. Der Durchschnitt der sieben besseren Ergebnisse beträgt 78% Patchbelegung bei 80% Geschoßbelegung, dem die sieben schlechteren Ergebnisse mit 70% Patchbelegung bei 86% Geschoßbelegung gegenüberstehen. Die Einzelergebnisse zeigen, dass dieser Effekt nicht immer auftritt und viele Lösungen dem Durchschnitt entgegen sprechen (4-I-1).

5.3 Layoutbandbreite

Um die Varianz der Lösungen festzustellen, wurde 4-I-2 mehrere Male durchlaufen. Neben dem Layout aus Kapitel 5.2 sind fünf weitere, fortlaufenden Layoutlösungen des Programms erzeugt worden (Abb. 41).



Abbildung 41 - Verschiedene Layouts für 4-I-2

| | Layout 0 | Layout 1 | Layout 2 | Layout 3 | Layout 4 | Layout 5 |
|------------------------|----------|----------|----------|----------|----------|----------|
| Adjazenz-Länge | 10,62 | 20,66 | 16,56 | 18 | 17,65 | 21,05 |
| belegte Patches | 1257 | 1141 | 1222 | 984 | 1084 | 885 |
| Ticks | 203 | 7 | 26 | 19 | 65 | 1043 (!) |

Tabelle 5 - nicht gewichtete, unskalierte Vergleichswerte der Layouts für 4-I-2

Es ist deutlich ersichtlich, dass die Lösungen sehr unterschiedlich gestaltet sind und die letztendlichen Konfigurationen weitestgehend dem Programm überlassen werden. Damit arbeitet der Algorithmus im Sinne von Reinhard König kreativ, da *"das Ergebnis dieses Prozesses Eigenschaften aufweist, die nicht als direkte Folge der Anforderungen bzw. nicht aus der genauen Beschreibung des Problems hervorgehen"*²⁷

Vergleicht man die Layouts mit den Werten, fällt auf, dass die Zeit bis eine Lösung gefunden ist (angegeben in Ticks) weder für bessere Gesamtwerte noch für besonders ausgefallene Lösungen sorgt. Je nachdem, wo die Flächen anfangs randomisiert initialisiert werden, fällt es dem Programm bei dieser Konstellation teilweise sehr leicht (Layout 1 mit 7 Ticks - praktisch im selben Atemzug mit dem Start des Programms) oder auch sehr schwer (Layout 5). Strukturell und auch hinsichtlich der Werte bietet Layout 5 jedoch keinen Mehrwert zum wesentlich schneller erzeugten und vergleichsweise ähnlichen Layout 3. Es empfiehlt sich also, nach einer gewissen Dauer ohne zufriedenstellendem Ergebnis das Programm von Neuem beginnen zu lassen. Hierzu könnte auch ein Automatismus implementiert werden, der die Komplexität der gerade gesuchten Lösung berücksichtigt und nach einer bestimmten Berechnungszeit den Neustart veranlasst bzw. vorschlägt.

5.4 Prüfung der Interaktionsmöglichkeit

Die implementierte Interaktionsmöglichkeit soll nun daraufhin getestet werden, unter welchen Aspekten Eingriffe vom Nutzer durch den Algorithmus angenommen werden, da die Erfahrung bei allen getesteten interaktiven Programmen gezeigt hat, dass oftmals einige wenige Iterationen oder Generationen später der Eingriff aufgrund von Widersprüchlichkeiten mit den Optimierungskriterien nicht mehr wahrnehmbar war.

Für das Layout 2-I-0 (Abb. 42-1) interagiert der Nutzer, um den Speisesaal weiter oben und den Gymnastikraum unten angeordnet haben (Abb. 42-2).

²⁷ Reinhard König, 2011, KREMLAS Entwicklung einer kreativen evolutionären Entwurfsmethode für Layoutprobleme in Architektur und Städtebau, 2011, Hrsg. Dirk Donath, Reinhard König und Frank Petzold, Bauhaus Universität Weimar, Seite 16



Abbildung 42 - Layoutentwicklung mit Interaktionen

Sichtbar ist, dass durch die negative Adjazenz der Speisesaal bis zur Lösungsfindung wieder von der Kollegialen Führung weg und damit nach unten bewegt wird (Abb. 42-3). Mit diesem Wissen kommt es zu einer neuerlichen Interaktion, in der die gewünschte Richtungsreihenfolge vollständig berücksichtigt (Abb. 42-4) und dann auch vom Algorithmus umgesetzt wird (Abb. 42-5).

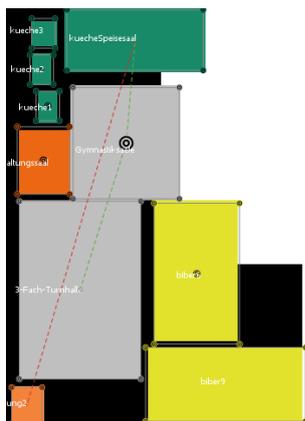


Abbildung 43 - Evaluierung von 3-L-0

Nun wird noch ein Fall getestet, wo eine Wunschlösung vom Nutzer nachgebaut werden soll. Es wird angenommen, dem Nutzer schwebt exakt die Lösung vor, die in Kapitel 5.2 unter 3-L-0 zu finden ist (Abb. 43). Ein neues Layout wird kreiert und während des Prozesses eine Fläche verschoben und daraufhin eine Lösung gesucht (Abb. 44). Im zweiten Beispiel (Abb. 45) werden daraufhin fast alle Räume in etwa dort angeordnet, wo sie in Ergebnis 3-L-0 liegen und erneut nach einer Lösung gesucht.



Abbildung 44 - Lösung, Interaktion mit 1 Fläche, neue Lösung

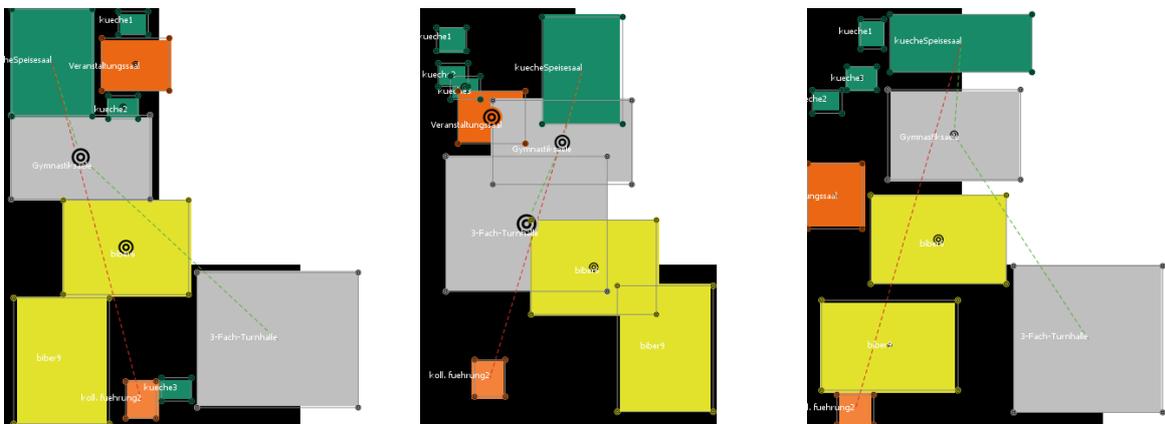


Abbildung 45 - neue Lösung, Interaktion der Flächen in der Art von 3-L-0, neue Lösung

Es geht klar hervor, dass zumindest gewisse Eigenschaften angenommen werden. Beibehalten wurde der Gymnastikbereich, der nach der einmaligen Verschiebung vom Fuß in den Hals auch dort verblieben ist. Jedoch hat das Programm genügend "eigenen Willen" aufgrund seiner gegebenen Anforderungen, um Eingriffe wieder rückgängig zu machen. Besonders fällt dies bei der Platzierung von der im Beispiel größten Fläche, der Dreifachturnhalle, auf.

Eine definierte Anordnung samt Fixierung durch Interaktion, wie sie in GAMES zu finden ist, müsste bei Bedarf jedenfalls ergänzend implementiert werden. Der Nutzer muss in dem Bewusstsein handeln, die Lösungsvielfalt damit gegebenenfalls auch ungünstig einzuschränken.

5.5 Vergleich zum Einfluss von Interaktion und Randomisierung

In der Gesamtaufstellung aus Kapitel 5.2 fällt auf, dass Layout 3-O-0 sowohl hinsichtlich der Nachbarschaftslängen als auch der Typologiepatches weit hinten liegt. Es wird mittels Interaktion der ursprünglichen Lösung (Abb. 46) und durch drei neu erzeugte Layouts (Abb. 47) ein Vergleich angestellt.

ausgeschlossen, jedoch ergeben sich daraus auch Vorteile: *"Erstens entsteht hinsichtlich der Nutzerinteraktion durch die Hierarchisierung eine bessere Übersicht über die zu bearbeitenden Elemente. [...] Zweitens können, verglichen mit NHLP [nicht-hierarchischen Layoutproblemen, Anm.], komplexe Probleme mit vielen Elementen und Relationen wesentlich schneller gelöst werden."*²⁸

Der betrachtete Bildungsbereich, der nun erzeugt werden soll, besteht aus sieben Bildungsräumen, denen Appendices, Abstell- und Sanitärräume zugeordnet sind. Sie sollen Sichtkontakt mit einer Multifunktionsfläche haben, dem Kernstück des Bildungsbereiches. Darüber hinaus gibt es einen Musikraum, Garderoben und für den Lehrkörper Aufenthaltsraum mit Teeküche.

| | | |
|------------------|------------------|-------------------------|
| Bildungsraum 1 | Abstellraum 2 | Sanitärbereich 4 |
| Appendix 1 | Sanitärbereich 2 | Multifunktionsfläche |
| Abstellraum 1 | Bildungsraum 3 | Bildungsraum Musik |
| Bildungsraum 2.1 | Abstellraum 3 | Garderobe |
| Bildungsraum 2.2 | Sanitärbereich 3 | Putzraum |
| Bildungsraum 2.3 | Bildungsraum 4 | Teamraum |
| Bildungsraum 2.4 | Appendix 4 | Teeküche |
| Appendix 2 | Abstellraum 4 | Sanitärbereich Personal |

Tabelle 7 - Raumprogramm eines Bildungsbereichs mit Farbschema

Für die bisherigen verlangten Anforderungen kommt der gewählte Algorithmus hinlänglich zurecht. Ein Versuch mit insgesamt 24 anzuordnenden Räumen überfordert den Algorithmus hinsichtlich der benötigten Zeit zur Lösungsfindung erstmals. Da sich die Lösungssuche exponentiell mit den Anforderungen verlängert, empfiehlt sich bei erhöhten Anforderungen die Umsetzung einer Evolutionsstrategie im Algorithmus. Hierzu bietet es sich an, auf die von Reinhard König für die "Dichte Packung" publizierten Parameter zurückzugreifen, *"da sich die verschiedenen Parameter gegenseitig beeinflussen und die günstigen Werte für einen Parameter hinsichtlich der Konvergenzgeschwindigkeit des Systems von der erreichten Qualität einer Lösung abhängig sein können."*²⁹ Dennoch macht es Freude, auf Basis von unausgereiften Layoutvorschlägen die Räume manuell nachzujustieren (Abb.48).

²⁸ Reinhard König, 2011, KREMLAS Entwicklung einer kreativen evolutionären Entwurfsmethode für Layoutprobleme in Architektur und Städtebau, 2011, Hrsg. Dirk Donath, Reinhard König und Frank Petzold, Bauhaus Universität Weimar, Seite 165 - 166

²⁹ Reinhard König, 2011, Generierung von Grundriss-Layouts mittels Hybrider Evolutionsstrategie, aus: Arbeitspapiere Nr. 4, März 2011, Bauhaus Universität Weimar, Seite 35



Abbildung 48 - Layout eines Bildungsbereichs mit 24 Räumen nach ca. 30 Sekunden (links) und Layout nach diversen unterstützenden Eingriffen, Gesamtdauer etwa 2 Minuten

Die Räume werden nun in einem weiteren Schritt in weiterhin sinnvolle Einheiten zusammengefasst (Appendices und Abstellräume als Teil des jeweiligen Bildungsraums), sowie eine Anpassung der jeweiligen Nachbarschaften vorgenommen. Für die verbleibenden 17 Räume werden ebenfalls nur sehr schwerfällig Lösungen gefunden (Abb. 49). Auffallend ist, dass scheinbare Anforderungen im Algorithmus das Programm dazu verleiten, die Multifunktionsfläche in der Typologie I quer anordnen zu wollen. Dieser Effekt tritt auch auf, wenn I um 90° gedreht wird, dann tendiert die Multifunktionsfläche zu einer Orientierung von oben nach unten. Dass die Anordnungen um diesen quergestellten Raum nicht sehr vielfältig sind, liegt in der Natur der Sache. Es ist unrealistisch, von einem Programm Kreativität zu erhoffen, wenn das gewünschte Ergebnis innerhalb eines bestimmten Rahmens erzeugt wird.

In beiden Fällen (24 und 17 Raumeinheiten) lassen sich die letztendlich erstellten Layouts jedoch immer noch gut und allenfalls dem eigenen Geschmack entsprechend interpretieren. Mit den nötigen Ergänzungen (Evolutionsstrategie) lässt sich auch die benötigte Performance herstellen, um damit letztendlich einen raschen interaktiven Entwurfsprozess simulieren zu können. Ein zusätzlicher Ansatz wäre auch, optional die erlaubten Schnittstellen an die Anzahl der Flächen bzw. Links zu koppeln, um die Lösungssuche mit vielen Räumen praktikabler gestalten zu können.

In dem oben beschriebenen Anwendungsbeispiel und den damit verbundenen, detailliert gegebenen Anforderungen und Umsetzungsspielräumen stellt sich die Frage, wie der Nutzer damit umgehen möchte. Eine Implementierung solcher fallspezifischer Eingrenzungen des Lösungsraums steht in der



Abbildung 49 - Layouts eines Bildungsbereichs mit 17 Räumen

Regel dem damit verbundenen Aufwand nicht entgegen und wurden deshalb auch in den obigen Layoutvarianten außer Acht gelassen. Als Beispiel sei hier die schriftliche Anmerkung des Putzraums zitiert: *"optional: Putzräume von 2 Bibern können in einem gemeinsamen Putzraum (16 m²) zusammengefasst werden"*³⁰. Hierbei muss sich der Nutzer einfach selbst zu helfen wissen, wenn er, zu gutem Recht, solche Informationen nicht an das Programm weitergeben möchte. Genauso steht es mit der Umsetzung der Angaben, die das Programm erhält. Teilen sich vier Bildungsräume einen Sanitärblock, wird das Programm tendenziell diesen Sanitärblock mittig mit einer wesentlichen zentralen Stellung positionieren - auch wenn dies dem Nutzer widerstrebt. *"Computer sind wirklich reizende Geschöpfe. Sofern man ihnen logische Anweisungen gibt, liefern sie logische Arbeit."*³¹ Will der Nutzer also daran etwas ändern, sollte er die Parameter abwandeln und den Sanitärblock etwa nur einem der Bildungsräume anhängen, wie im Beispiel mit 17 Räumen erfolgt.

³⁰ Magistrat der Stadt Wien, Magistratsabteilung 19, 2015, Neubau Bildungscampus Nordbahnhof 1020 Wien - Auslobung - Beilage C.03 (Detailanforderungen), Tabellenblatt Biber 1, Wien, 18.11.2015,

³¹ Haruki Murakami, Hard-boiled Wonderland und das Ende der Welt, Seite 346, Suhrkamp Verlag, 1. Auflage 2000

| Funktion\Typologie | C | H | I | L | O | T | Z | Squ. | Pav. | Mittelwert |
|---------------------------|------|------|-----|------|------|------|------|------|------|------------|
| 3-Fach-Turnhalle | 3,4 | 0,8 | 2,8 | 2,6 | 2,6 | 3,2 | 0 | 2,4 | 1,4 | 2,13 |
| Gymnastiksäle | 2,7 | 2,8 | 1,8 | 2,8 | 3,4 | 0,4 | 2,6 | 2,6 | 2,2 | 2,37 |
| Biber 9 | 0,2 | 2,8 | 1,6 | 0 | 1 | 0 | 1 | 0 | 0 | 0,73 |
| Biber 6 | 0 | 0 | 1,4 | 1,4 | 0,7 | 1 | 1 | 0 | 1 | 0,72 |
| Veranstaltungssaal | 0,2 | 1,2 | 0 | 1,5 | 0 | 1,2 | 0 | 0,6 | 1,4 | 0,68 |
| Küche Speisesaal | 0 | 0 | 0,2 | 0,4 | 0 | 0 | 0,6 | 0 | 0 | 0,13 |
| Küche 1 | 0 | 0,8 | 0,2 | 1,4 | 0 | 1 | 0 | 0 | 0,8 | 0,47 |
| Küche 2 | 0,4 | 0 | 1 | 0,8 | 0,2 | 0,1 | 0,8 | 1 | 0 | 0,48 |
| Küche 3 | 0,4 | 0,8 | 0 | 0 | 0,4 | 1 | 0,2 | 0,6 | 0,4 | 0,42 |
| Koll. Führung 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0,4 | 0 | 0 | 0,04 |
| Mittelwert | 0,73 | 0,92 | 0,9 | 1,09 | 0,83 | 0,79 | 0,66 | 0,72 | 0,72 | |

Tabelle 8 - Evaluierung zur Zentralität für 3-0

Durch die Berechnungsart der Evaluierung ist es logischerweise wahrscheinlicher, dass eine große Fläche eine hohe Zentralität aufweist, da diese eher von den Evaluierungsagenten passiert wird als eine kleine Fläche. Dennoch führt dies nicht zwangsweise zu einer vorhersehbaren Reihung der Werte. Die Gymnastiksäle, die in etwa gleich groß wie die Bildungsbereiche sind, liegen sogar noch eine Spur vor der beträchtlich größeren 3-Fach-Turnhalle, wohingegen die Bildungsbereiche wertmäßig nahe am kleineren Veranstaltungssaal liegen. Doch auch die großen Flächen können im Randbereich liegen (3-Z-0), was sich beispielsweise als Filter- oder auch Ausschlusskriterium bei weiteren Funktionen eignen würde.

Eingeschränkt vergleichbar sind auch die Layouts von 2-0 mit ebenfalls 10 Funktionsflächen sowie einigen gleichbleibenden Funktionen und Beziehungen.

| Funktion\Typologie | C | H | I | L | O | T | Z | Squ. | Pav. | Mittelwert |
|---------------------------|------|------|------|------|------|------|------|------|------|------------|
| 3-Fach-Turnhalle | 1,6 | 0,1 | 0,4 | 0,6 | 0 | 0,2 | 0,2 | 0 | 0 | 0,34 |
| Gymnastiksäle | 0,7 | 0,6 | 1,2 | 1,6 | 1 | 2,6 | 0,2 | 1,8 | 1,4 | 1,23 |
| Kreativbereich | 1,6 | 2,6 | 2,4 | 3,6 | 2 | 3,8 | 2,4 | 2,9 | 3,2 | 2,72 |
| Biber 7 | 2,6 | 0 | 2 | 0,4 | 0,6 | 1 | 1,8 | 1,4 | 1,4 | 1,24 |
| Biber 8 | 0,9 | 1,2 | 1,6 | 0 | 0,2 | 1,2 | 1,4 | 0,6 | 0,8 | 0,88 |
| Biber 9 | 0 | 1 | 2,4 | 1,2 | 1,6 | 1,1 | 0 | 0 | 0,2 | 0,83 |
| Biber 6 | 0 | 2,2 | 1,2 | 1,2 | 0,8 | 1 | 2,2 | 2,2 | 0,8 | 1,29 |
| KücheSpeisesaal | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0,00 |
| Koll. Führung 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0,00 |
| Veranstaltungssaal | 0,8 | 1 | 0 | 1,2 | 0,2 | 0 | 0 | 1,3 | 0 | 0,50 |
| Mittelwert | 0,82 | 0,87 | 1,12 | 0,98 | 0,64 | 1,09 | 0,82 | 1,02 | 0,78 | |

Tabelle 9 - Evaluierung zur Zentralität für 2-0

Die Auswirkungen der negativen Beziehung zwischen Speisesaal und Kollegialer Führung 2 sind auch in diesem Ergebnis klar abzulesen: durch die Abstoßung tendieren die Funktionen zu einer Randlage, wodurch sie weniger wahrscheinlich überquert werden.

Interessant ist die Auswirkung einer Sterntopologie um den Kreativbereich 2, der am stärksten positioniert und somit am zentralsten liegt. Dies hat Auswirkungen auf die 3-Fach-Turnhalle, die beziehungsbedingt durch die zwischengeschalteten Gymnastiksäle tendenziell eher an den Rand gedrängt wird.

Keine besonderen Erkenntnisse können zum Einfluss der Typologie auf die Intensität der Zentralitäten getroffen werden.

5.8 Anmerkungen zur Schnittstelle

Da das Dateiformat CSV zwar häufig eingesetzt wird, die allgemeinen Standards jedoch nicht systemübergreifend vereinheitlicht sind³², ist die Weiterverarbeitung der Daten sehr fallspezifisch zu betrachten. So kann z.B. die Kennzeichnung des letzten Datenzeichens in verschiedenen Betriebssystemen voneinander abweichen. Insbesondere durch den Einsatz von Excel durch die für den Nutzer optisch ansprechender wirkende Oberfläche im Vergleich zum regulären Texteditor entstehen letztendlich Probleme, z.B. bei der Verarbeitung von Sonderzeichen sowie dem Umgang mit string-definierenden Anführungszeichen, die teilweise ungewünscht als Teil vom Text interpretiert und deshalb vom Programm mit weiteren Anführungszeichen versehen werden. Für den weiteren Gebrauch würde sich die Entwicklung einer direkten geschloßweisen Parameterübergabe ohne Zwischenschritt im Excel lohnen.

³² Y. Shafranovich, Oktober 2015, verfügbar unter: <https://tools.ietf.org/html/rfc4180> Abschnitt 2 - Definition of the CSV Format [zuletzt abgerufen am 04.02.2016]

6. Zusammenfassung und Ausblick

Eine Inspiration in der frühen Planung; ein Loslösen von ersten Impulsen für einen gänzlich neuen Zugang zu einem Projekt oder auch das umfassend rasche Abwickeln umfangreicher Variationen bei unterschiedlichsten Ausgangsfaktoren bietet nach wie vor ein weitgehend ungenütztes, und dennoch sehr greifbares Potenzial. Ein fertiges Funktionsschema ist nicht die Lösung, die der Architekt in einem solchen Tool finden möchte. Die Lösungen in Abschnitt 5.2 sind für die rasche Erstellungszeit breiter gefächert, als es manuell in derselben Zeit erledigt werden könnte, da die Lösung auf eine Art berechnet wird, die der Nutzer selbst - unter anderem aufgrund von Erfahrung, Routine und Zeitaufwand - vermutlich nicht einschlagen würde. Der positive Effekt des Einsatzes eines Computers besteht in diesem Fall auch darin, dass er völlig losgelöst von vorherigen Ergebnissen jede Lösungssuche erfrischend neu angehen kann.

Der Umgang mit dem Programm und die Erforschung des Lösungsraums machen Spaß. Eine Verwendung für die Erstellung rascher, erster Lösungsansätze durch die Autorin ist angedacht.

Die Einführung der Typologien erweist sich mitunter nicht als Beschränkung des Lösungsraumes, sondern erweitert ihn in Hinblick auf Interpretierbarkeit der Layouts und Ausreizung der Lösungserstellung. Im Handumdrehen neue Typologien einführen zu können, stellt sich in der Praxis als besonders gelungene, unkompliziert handhabbare Funktion heraus (unvorhersehbar benötigt in Kapitel 5.6). Die Flächen könnten situationsbedingt einen höheren Impuls zur Bewegung in die Typologie erhalten bis beispielsweise nur noch ein frei wählbarer, vorgegebener prozentualer Wert der Fläche außerhalb der Typologie liegt. Die erzeugten Ergebnisse mit Mindestbelegung der Hälfte sind jedoch durchwegs als spannende Lösungsansätze zu betrachten.

Durch die Übernahme der zu verarbeitenden Daten des Schulcampus konnte bewiesen werden, dass ein sinnvoller Datenaustausch erfolgen und das Programm sehr gut an eine vorhandene Vorleistung anknüpfen und verfügbares Problemwissen abarbeiten kann, dieses aber nicht erforderlich ist, um eine Lösung zu erzeugen. Die hierarchische Struktur der Funktionsaufteilung aus den Auslobungsunterlagen bietet sich ganz besonders dafür an, die in Kapitel 5.6 besprochene Strategie der hierarchischen Layoutprobleme heranzuziehen. Dies sollte jedoch keine Vorgabe durch das Programm sein, sondern eine Entscheidung, die der Nutzer aufgrund der gegebenen Situation von Anwendungsfall zu Anwendungsfall neu treffen muss.

Die aus den Ablaufanalysen erzeugten Angaben zu Wunschnachbarschaften sind in einem ausgewogenen Maß vorhanden, um dem Programm Orientierung zu bieten, jedoch gleichzeitig Handlungsspielraum zu gewähren. Ab einem Knotengrad von vier ein- bzw. ausgehenden Knoten

wird die Anordnung der betroffenen Flächen relativ vorhersehbar. Wenn ein hierarchischer Ansatz gewählt ist kann überlegt werden, ob Einheiten mit besonders hohen Knotengraden untereinander nicht zu einer neuen, größeren Einheit gebündelt werden.

Das Setzen der Proportionsangaben kann nur unter Berücksichtigung des eben abzuwickelnden Maßstabs geschehen. Für die großflächigen Arrangements ergeben sich niedrigere mögliche Werte als bei einem Arbeiten auf Raumbasis. Dadurch lässt sich feststellen, dass ein individuelles Setzen der Proportionswerte primär dann sinnvoll ist, wenn eine Spanne der zu verteilenden Flächengrößen vorhanden ist. Bewegen sich die Flächen in etwa der gleichen Größe und Funktion, kann die Proportion global für alle Flächen überlegt und gesetzt werden.

Mit der Angabe von Problemwissen muss sehr sorgfältig und sehr bewusst umgegangen werden, um dem Programm keine fehlinterpretierbaren Botschaften zu übermitteln. Dies fängt bereits bei der Aufgliederung der Funktionsflächen an und zieht sich kontinuierlich über alle Informationen zu Beziehungen und Proportionen. Liefert das Programm nicht das, was erwartet wird, empfiehlt sich jedenfalls ein Überdenken der Angaben und ein erneuter Versuch.

Die optische Aufbereitung der Bewertungsfunktion hat sich als bequeme Übersicht zum erzeugten Layout erwiesen und ist hinsichtlich aller für einen Lösungssatz interessanten, errechenbaren Faktoren denkbar, die darstell- und messbar sind. Einen besonderen Reiz hätte es auch, eine solche Bewertung zum gezielten Filtern eines breitgefächerten erstellten Lösungsspektrums einzusetzen, um dadurch rasch Alternativen mit ähnlichen Eigenschaften zum Vergleich bündeln zu können.

Klar hervorzuheben sei an dieser Stelle nochmals, dass sich die Ergänzung bzw. der Nachbau einer Evolutionsstrategie auch bei einer bezüglich der Performance so angenehmen Programmier-Software wie NetLogo definitiv bezahlt machen würde. Damit könnte nicht nur die Arbeit mit dem Programm beschleunigt, sondern auch die generelle Leistungsfähigkeit betreffend Raumanzahl und verzwickter gestalteten Lösungsräumen verbessert werden.

Für einen Einsatz ist es vorstellbar und sinnvoll, das Programm noch um weitere Steuerungsmöglichkeiten der Layouts durch Priorisierung der verschiedenen Anforderungen, die das Programm erfüllen muss, zu ergänzen. Hierzu könnte der Wert von in Summe 100% auf die drei Faktoren

- a. Überlappungsfreiheit,
- b. Adjazenz
- c. Einhalten der Außengrenzen,

je nach Anforderung an die jeweilige Situation vom Nutzer aufgeteilt werden. So können Einflüsse wie z.B. die Grundstückssituation (Baulücke oder freistehend) oder Funktion/Konzeptionierung, die

zwingend getrennte Einheiten oder auch ineinander übergehende Bereiche erfordern kann bzw. deren Nachbarschaften von organisatorisch unterschiedlich intensiver Bedeutung sein mögen, individuell abgefangen werden.

Betrachtet man mögliche weitere Entwicklungen, stellen sich folgende zusätzliche Fragen als besonders nachgehenswert heraus.

- Für die Arbeit wurde ein klassischer Top-Down-Ansatz gewählt, der sich so mitunter auch aus den Auslobungsunterlagen angeboten hat. Wie sehen konträre Layouts auf Basis einer hinsichtlich der Angaben nahezu identen Bottom-Up-Variante im selben Programm aus?
- Darf die Summe der Flächengrößen $\sum A_i$ größer sein als die Fläche des zur Verfügung stehenden Raumes A_{ges} , indem die Überlappung bis zu einem gewissen prozentualen Wert erlaubt ist? Was bedeuten die Überlappungen der Zonen, welche Impulse können dadurch in der weiteren Planung entstehen?
- Darf eine vergleichsweise kleine Fläche innerhalb einer großen Fläche liegen und wird somit von der größeren Funktion aufgenommen? Welche Auswirkungen hat die Möglichkeit, dies für alle oder bestimmte Funktionen zu erlauben?
- Im Zuge der verfügbaren Arbeiten zur Layoutgenerierung wird die Orthogonalität als Einschränkung empfunden. Was passiert in einem konkreten Beispiel, würde ein Entwurf in so frühem Stadium mit Aufheben der Orthogonalität oder einer alternativen Darstellung der Flächen mittels Kreisen konfrontiert. Wird dadurch das Ergebnis interessanter und freier interpretierbar? Oder würde dies das Ergebnis in einem zu frühen Zeitpunkt so konkretisieren, dass die weitere Kreativität dadurch eher eingeschränkt würde? Läuft der Ansatz, polygonale Formen erzeugen zu wollen, mitunter Gefahr, Imitationen zu erzeugen, da eine komplexe Formensprache derzeit doch stark als Stilmittel und Handschrift eines Architekten sehr bewusst eingesetzt und dementsprechend gekonnt auf ein Zusammenspiel von Innen und Außen sowie Raumerfahrungen setzt?
- Welche Resultate würde das Programm im dreidimensionalen Raum bringen? Bringt der Ansatz, ein Geschoß nach dem anderen abzuarbeiten, die interessanteren Ergebnisse als eine parallele Anordnung der Flächen innerhalb der jeweiligen Geschoße? Eine dreidimensionale Ausweitung ist insbesondere an diesem Beispiel am ehesten mit fixen Geschoßhöhen denkbar. Gerade die Schule als öffentlicher Raum hat sich sowohl der Barrierefreiheit als auch einer wegetechnischen Effizienz zu beugen und eine bedachte Planung unterschiedlicher Höhen,

Verschnitte und dergleichen durch künstliche Intelligenz stünde nicht im Verhältnis zum programmiertechnischen Aufwand und würde derzeit noch kaum zu kreative(re)n Ergebnissen führen. Komplexe dreidimensionale Verschachtelungen sollten, ebenso wie das Verlassen der Orthogonalität als Stilmittel bewusst und gekonnt eingesetzt und nicht durch das Programm evoziert werden. Für eine vereinfachte Umsetzung im dreidimensionalen Raum bräuchte es jedoch lediglich eine Ergänzung der Adjazenzen in die angrenzenden Geschoße.

7. Schlussbetrachtung

In der hier vorliegenden Arbeit wurde eine Strategie zur computerunterstützten Layoutgenerierung mit interaktiver Funktion entwickelt und anschließend umgesetzt. Durch den Einsatz des Programms kann bereits mit geringem Problemwissen über eine Planungsaufgabe eine weite Bandbreite an Layouts innerhalb kürzester Zeit erzeugt und diese in Echtzeit manipuliert werden. Der Lösungsraum korreliert mit dem vorhandenen Problemwissen und kann nach verschiedenen, voneinander unabhängigen Aspekten eingeschränkt oder erweitert werden. Dadurch ist es dem Planer auch mit wechselnden Analyseansätzen und -fortschritten, sowie unterschiedlich intensiv ausgeprägten Anforderungen seitens Auftraggeber, bereits in der ersten Planungsphase möglich, die Lösungssuche durch das Programm als unterstützende Planungshilfe zu nutzen. Der Planer kommt mit dem Einsatz des Programms schneller zu einem breiten, fallweise unkonventionellen und jedenfalls unvoreingenommenen Lösungsspektrum als durch manuelle Planung.

Die Arbeit zeigt darüber hinaus anhand einer Fallstudie, wie eine solche algorithmische Grundrissgenerierung innerhalb von realen, umfangreicheren Projekten eingesetzt werden kann und zeigt Beispiele für die Resultate einer Anwendung bei frühem Problemwissen, welches in verschiedenen Varianten unterschiedlich stark ausgeprägt zu finden ist. Das exemplarisch herangezogene Beispiel behandelt eine Schule, die nach dem Konzept des Wiener Campusmodells strukturiert ist. Eine entsprechende Anwendung für das Programm bei der frühen Planung anderer komplexer Gebäude wie etwa Krankenhäuser, Flughäfen, Einkaufszentren, Firmenanlagen oder Justizanstalten ist gleichermaßen denkbar und möglich.

Ein Aufgreifen der verwendeten Methoden sowie die Weiterentwicklung der Funktionalitäten ist mit dem Quellcode im Anhang möglich und als share-alike gestattet.

Bibliographie

- [1] Georg Franck und Tomor Elezkurtaj, 2011, Design Methods und computergestützte Grundrissentwicklung, verfügbar unter: http://www.iemar.tuwien.ac.at/wp-content/uploads/2011/12/design_methods.pdf [zuletzt aufgerufen am 21.2.2016]
- [2] Riklef Rambow, Zur Bedeutung der Architektur für den Lernort Schule, 2006, leicht verändert in: Starke Eltern - Starke Kinder, Jahresheft 2006, Deutscher Kinderschutzbund, verfügbar unter <http://www.tu-cottbus.de/theoriederarchitektur/Lehrstuhl/deu/rambow/SESK.pdf> [zuletzt aufgerufen am: 09.03.2016]
- [3] Kaiman Lee, 1969, The Computer as an Architectural Design Tool, Iowa State University of Science and Technology
- [4] William J. Mitchell, Robert L. Dillon, 1972, A Polyomino Assembly Procedure for Architectural Floor Planning, aus: EDRA 3: Research and Practice: Proceedings of the edra3/ar8 Conference (1972), Seite 23-5-01 bis 23-5-12
- [5] Scott A. Arvin und Donald H. House, 1999, Making Designs Come Alive: Using Physically Based Modeling Techniques in Space Layout Planning, aus: Computers in Building (1999), Seite 245 bis 262
- [6] Benachir Medjdoub und Bernard Yannou, 2001, Dynamic Space ordering at a Topological Level in Space Planning, aus: Artificial Intelligence in Engineering Vol. 15 Issue 1, Januar 2001, Seite 47 bis 60
- [7] Tomor Elezkurtaj und Georg Franck, 2001, Evolutionary Algorithms in Urban Planning, aus: Computergestützte Raumplanung CORP 2001 Geo.Multimedia.01, 2001, Seite 269 bis 272, verfügbar unter http://www.corp.at/archive/CORP2001_Elezkurtaj_DO.pdf [zuletzt aufgerufen am: 12.02.2016]
- [8] Reinhard König, 2011, Generierung von Grundriss-Layouts mittels Hybrider Evolutionsstrategie, aus: Arbeitspapiere Nr. 4, März 2011, Bauhaus Universität Weimar, verfügbar unter: http://infar.architektur.uni-weimar.de/service/drupal-infar/sites/default/files/upload/Publikationen/Arbeitspapiere/InfAR_04_Grundriss-Layouts.pdf [zuletzt aufgerufen am 10.01.2016]
- [9] Dirk Donath, Reinhard König und Frank Petzold, 2011, KREMLAS Entwicklung einer kreativen evolutionären Entwurfsmethode für Layoutprobleme in Architektur und Städtebau, 2011, Bauhaus Universität Weimar
- [10] Wiener Schulen (Magistratsabteilung 56), Das Wiener Campusmodell, verfügbar unter: <https://www.wien.gv.at/bildung/schulen/schulbau/campus/> [zuletzt aufgerufen am 12.2.2016]
- [11] Uri Wilensky, NetLogo, 1999, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. Software unter: <http://ccl.northwestern.edu/netlogo/> [zuletzt abgerufen am: 13.02.2015]
- [12] Uri Wilensky, NetLogo User Manual version 5.3.1, 2016, verfügbar unter: <https://ccl.northwestern.edu/netlogo/docs/> [zuletzt abgerufen am: 14.03.2016]
- [13] Daniel Elsner, 2016 (in Druck), SimSchool - Optimierung der Abläufe in Schulbauten, Diplomarbeit, TU Wien, Fakultät für Architektur und Raumplanung
- [14] Magistrat der Stadt Wien, Magistratsabteilung 19, 2015, Neubau Bildungscampus Nordbahnhof 1020 Wien - Auslobung, Wien, 18.11.2015
- [15] W Lorenz, G. Wurzer, S. Swoboda (ed.), flying bricks: algorithmisches entwerfen, Institut für Architekturwissenschaften, Abteilung Digitale Architektur und Raumplanung 259.1, Wien, 2015
- [16] Y. Shafranovich, Oktober 2015, verfügbar unter: <https://tools.ietf.org/html/rfc4180> Abschnitt 2 - Definition of the CSV Format [zuletzt abgerufen am 04.02.2016]

Abbildungsverzeichnis

| | |
|---|----|
| Abbildung 1 - Ergebnis und Weiterverarbeitung in Matrix..... | 8 |
| Abbildung 2 - Ergebnisse von DOMINO in unterschiedlichen Maßstäben..... | 9 |
| Abbildung 3 - Ergebnisse von Physics in der geometrischen Ebene | 10 |
| Abbildung 4 - Bandbreite möglicher Topologien für ein zweigeschoßiges Einfamilienhaus (links) und eine geometrisch optimale Lösung der Topologie 3 (rechts) | 11 |
| Abbildung 5 - Schnittflächenberechnung graphisch aufbereitet (oben) | 11 |
| Abbildung 6.1 - Screenshot Flächen, Abbildung 6.2 - Screenshot Wände, Abbildung 6.3 - Screenshot Polygone | 12 |
| Abbildung 7 - Screenshot Topologie | 12 |
| Abbildung 8 - Screenshot GAMES | 13 |
| Abbildung 9 - Screenshot von Dichte Packung: Sterntopologie mit einer zentralen Fläche, 10 erstellte Flächen 13 | |
| Abbildung 10.1 - Schritt 1 Initialisierung, Abbildung 10.2 - Schritt 2 Analyse, Abbildung 10.3 - Schritt 3 Bewegung | 16 |
| Abbildung 11 - Von der Initialisierung zum Layout | 17 |
| Abbildung 12 - Optimierung Überlappung, Typologie und Adjazenzen..... | 18 |
| Abbildung 13.1 - Überlappung, Abbildung 13.2 - Verschiebung, Abbildung 13.2 - Proportionsänderung | 18 |
| Abbildung 15 - ausgewählte Gebäudetypologien | 19 |
| Abbildung 14 - Bewegung des Raums in den Grundriss..... | 19 |
| Abbildung 16 - Anziehung und Abstoßung durch positive und negative Nachbarschaftsbeziehungen | 20 |
| Abbildung 17- Lösungssuche mit Interaktion in Schritt 4 | 21 |
| Abbildung 18 - Screenshot der Benutzeroberfläche | 22 |
| Abbildung 19 - Screenshot der Schaltflächen zu Programmstart und Einstellungen..... | 23 |
| Abbildung 20 - Lösungsfenster..... | 24 |
| Abbildung 21 - Fenster für Nachverfolgungsgraph und Import/Exportbereich..... | 24 |
| Abbildung 22 - Überblick Programmablauf..... | 25 |
| Abbildung 23 - Diagramm Ablauf Import | 26 |
| Abbildung 24 - Screenshot der zusammengeführten Excel-Tabelle für eine Lösung mit 2 Geschoßen | 26 |
| Abbildung 25 - die aus Abbildung 24 erzeugte CSV für Geschoß 0 | 26 |
| Abbildung 26 - Skizze einer Fläche | 27 |
| Abbildung 27 - implementierte Grundrisstypologien. | 28 |
| Abbildung 28 - Screenshot zweier sich schneidender Flächen | 29 |
| Abbildung 29 - Screenshots Beziehungsanordnung randomisierter Beispiele | 29 |
| Abbildung 30 - Screenshot zweier Flächen auf dem Weg in eine I-Topologie | 30 |

| | |
|--|----|
| Abbildung 31 - Screenshot Verschiebung | 31 |
| Abbildung 32 - Screenshot einer Lösung von drei Flächen und ihre Evaluierung zur Zentralität | 32 |
| Abbildung 33 - Screenshot "SimSchool" Wegesimulationsprogramm..... | 33 |
| Abbildung 34 - Screenshot "SimSchool" Funktionsaufteilungsprogramm | 34 |
| Abbildung 35 - Diagramm zur hierarchischen Gliederung des Bildungscampus..... | 35 |
| Abbildung 36 - mögliche Aufteilung für zwei Geschoße von Daniel Elsner | 37 |
| Abbildung 37 - mögliche Aufteilung für drei Geschoße von Daniel Elsner | 40 |
| Abbildung 38 - mögliche Aufteilung für vier Geschoße von Daniel Elsner..... | 43 |
| Abbildung 39 - mögliche Aufteilung für fünf Geschoße von Daniel Elsner | 46 |
| Abbildung 40 - Screenshot zur Performancebeeinflussung komplizierter Adjazenzen | 50 |
| Abbildung 41 - Verschiedene Layouts für 4-I-2 | 52 |
| Abbildung 42 - Layoutentwicklung mit Interaktionen | 54 |
| Abbildung 43 - Evaluierung von 3-L-0 | 54 |
| Abbildung 44 - Lösung, Interaktion mit 1 Fläche, neue Lösung | 55 |
| Abbildung 45 - neue Lösung, Interaktion der Flächen in der Art von 3-L-0, neue Lösung..... | 55 |
| Abbildung 46 - Layout 3-O-0, Interaktion, Layout 3-O-0' | 56 |
| Abbildung 47 - drei weitere Layouts für 3-O-0..... | 56 |
| Abbildung 48 - Layout eines Bildungsbereichs mit 24 Räumen | 58 |
| Abbildung 49 - Layouts eines Bildungsbereichs mit 17 Räumen..... | 59 |
| Abbildung 50 - grafische Repräsentation der Evaluierung von 3-0..... | 60 |

Tabellenverzeichnis

| | |
|--|----|
| Tabelle 1 - Zusammenfassung und Vergleich der Programmeigenschaften..... | 14 |
| Tabelle 2 - verwendetes Funktionsflächenprogramm mit Farbkennzeichnung..... | 36 |
| Tabelle 3 - gewichtete positive Adjazenzlängen, die Einheit der Länge entspricht einem Patch | 49 |
| Tabelle 4 - Typologiebelegung in gewichteten Patches und Prozent | 51 |
| Tabelle 5 - nicht gewichtete, unskalierte Vergleichswerte der Layouts für 4-I-2 | 53 |
| Tabelle 6 - nicht gewichtete, unskalierte Vergleichswerte der Layouts für 3-O-0..... | 56 |
| Tabelle 7 - Raumprogramm eines Bildungsbereichs mit Farbschema | 57 |
| Tabelle 8 - Evaluierung zur Zentralität für 3-0 | 61 |
| Tabelle 9 - Evaluierung zur Zentralität für 2-0 | 61 |

Quellcode für NetLogo

```
__includes ["import.nls" "interact.nls" "intersecting-links.nls"
"beurteilung.nls"]

extensions [vectorview]

breed [students student]
undirected-link-breed [ umrahmungen umrahmung ]
undirected-link-breed [ beziehungen beziehung ]

patches-own[
  ;fuer layout
  _grundriss
  _belegungsgrad
  _boolbelegt
  ;fuer evaluierung
  _roomnumber
  _betretungsshaefueigkeit
]

turtles-own[
  ;fuer layout
  _xumriss
  _yumriss
  _x
  _y
  _realize
  _zahl ;Raumnummer
  _typ ;Eckenrichtung
  _doppelttest
  _edge-of-heading
  _proportion
  _maxproportion
  _my-moving-distance
  _function-name
  _outside-border
  _anziehung-x
  _anziehung-y
  _geschwindigkeit-x
  _geschwindigkeit-y
  _bezeichnung
  _beziehungsliste
  _inside-other
  ;fuer evaluierung
  _goal
  _passierende-tester
]

students-own [
  passing-list
]

links-own [
  _zahllink
]

beziehungen-own [
  _beziehungswert
  _endlindex
]
```

```

    _end2index
]

globals[
  weiter
  xumriss
  yumriss
  xumrissneu
  yumrissneu
  x
  y
  xold
  yold
  besetztflaeche
  gesamtgroesse
  doppelt
  doppelcount
  groesse
  zahl
  zahl2
  zahl3
  zahlfalse
  fehlerzaehler
  farbuebergabe
  realsize
  collision-ahead
  thisheading
  maxproportion
  gone_steps
  x_alt_manuell
  y_alt_manuell
  x_neu_manuell
  y_neu_manuell
  verschoben
  ersterlist
  zweiterlist
  erster
  zweiter
  can-move-edge
  can-move-all
  doppelt-intersection
  agent_index
  oldroom
  goal
  moving-distance
  verschieben-oder-proportion
  grundrisstest
  gesamtflwert
  beziehungsliste
  plot-patchesfalse
  grundhelpcount
]

to GO! ;Anordnung in Welt
  set moving-distance 0.5
  set pause false
  ask patches [set _grundriss true]
  set gesamtflwert count patches with [_grundriss = true]
  if count turtles = 0 [IMPORT!]
  move-it
end

```

```

to GO-INSIDE! ;Anordnung in Typologie
  set moving-distance 0.5
  set pause false
  load_parameters ;Typologie laden
  set gesamftlwert count patches with [_grundriss = true]
  if count turtles = 0 [IMPORT!]
  move-it
end

to IMPORT! ;CSV mit Flaechen laden
  let i_anzahl 0
  let file user-file
  set pause false
  carefully [file-open file] [print "Keine Datei geÃ¶ffnet" stop]
  let first-line? false
  carefully [
    while [not file-at-end?] [
      let csv_geschosseintrag file-read
      let bezeichnung file-read
      let farbe file-read
      let maxprop file-read
      let groessenangabe file-read
      set beziehungsliste file-read
      ;Erstellen des Zentrums-Turtles
      create-turtles 1 [setxy random-xcor random-ycor set _zahl i_anzahl
set _typ 0 set _x xcor set _y ycor set shape "target" set size 2 set label
bezeichnung set color farbe set _bezeichnung bezeichnung set _maxproportion
maxprop set _beziehungsliste beziehungsliste]
      ask turtles with [_zahl = i_anzahl] [
        set groessenangabe (groessenangabe / geschossgroesse) *
gesamtflwert
        set _xumriss sqrt(groessenangabe) / 2
        set _yumriss sqrt(groessenangabe) / 2
        if xcor <= _xumriss + 1 [
          set xcor xcor + _xumriss set _x xcor]
        if xcor >= (world-width - _xumriss - 1) [
          set xcor xcor - _xumriss set _x xcor]
        if ycor <= _yumriss + 1 [
          set ycor ycor + _yumriss set _y ycor]
        if ycor >= (world-height - _yumriss - 1) [
          set ycor ycor - _yumriss set _y ycor]
        set _realsize _xumriss * 2 * _yumriss * 2
        set realsize _realsize
        create-beziehungen-with other turtles
      ]
      set i_anzahl i_anzahl + 1
      set gesamtgroesse (gesamtgroesse + realsize)
      set anzahl_rooms i_anzahl
    ]
  ]
  [file-close print "csv verursacht Fehler!" stop]
  set doppelt true
  setup-everythingelse
  print "finished"
  file-close
end

to GET-READY! ;Randomisiertes Beispiel erzeugen
  clear
  set gesamtgroesse 0

```

```

ask patches with [_grundriss = true] [ set _boolbelegt false set
_belegungsgrad 0] ;set _belegt 0
let i_anzahl 0
while [i_anzahl < anzahl_rooms] [
  ifelse random_roomsize = true
  [set groesse random room_size]
  [set groesse room_size]
  if groesse > 35 [
    create-turtles 1 [setxy random-xcor random-ycor set _zahl i_anzahl
set _typ 0 set _x xcor set _y ycor set shape "target" set size 2]
ask turtles with [_zahl = i_anzahl] [
  set _xumriss sqrt(groesse) / 2
  set _yumriss sqrt(groesse) / 2
  if xcor <= _xumriss + 1 [
    set xcor xcor + _xumriss set _x xcor]
  if xcor >= (world-width - _xumriss - 1) [
    set xcor xcor - _xumriss set _x xcor]
  if ycor <= _yumriss + 1 [
    set ycor ycor + _yumriss set _y ycor]
  if ycor >= (world-height - _yumriss - 1) [
    set ycor ycor - _yumriss set _y ycor]
  set _realsize _xumriss * 2 * _yumriss * 2
  set realsize _realsize
  create-beziehungen-with other turtles
]
set i_anzahl i_anzahl + 1
set gesamtgroesse (gesamtgroesse + realsize)
]
]
set doppelt true
setup-everythingelse
end

to setup-everythingelse
ask beziehungen [set shape "gestrichelt" set _endlindex [who] of endl set
_endlindex [who] of endl
set hidden? false
]
let i_anzahl 0
while [i_anzahl < anzahl_rooms] [
  ask turtle i_anzahl [set beziehungsliste _beziehungsliste]
  carefully [ask beziehungen with [_endlindex = i_anzahl] [set
_beziehungswert item _endlindex beziehungsliste]] [ask beziehungen [set
_beziehungswert random 3]]
  set i_anzahl i_anzahl +1
]
ask beziehungen [if _beziehungswert = 0 [set color red] if
_beziehungswert = 2 [set color green]]
ask turtles with [_typ = 0] [
  hatch 1 [set xcor _x - _xumriss set ycor _y - _yumriss set _zahl _zahl
set _typ 1 set size 1 set label ""]
  hatch 1 [set xcor _x - _xumriss set ycor _y + _yumriss set _zahl _zahl
set _typ 2 set size 1 set label ""]
  hatch 1 [set xcor _x + _xumriss set ycor _y - _yumriss set _zahl _zahl
set _typ 3 set size 1 set label ""]
  hatch 1 [set xcor _x + _xumriss set ycor _y + _yumriss set _zahl _zahl
set _typ 4 set size 1 set label ""]
]
set i_anzahl 0
while [i_anzahl < anzahl_rooms] [
  ask turtles [

```

```

    set ersterlist [who] of turtles with [_typ = 1 and _zahl = i_anzahl]
    set zweiterlist [who] of turtles with [_typ = 2 and _zahl = i_anzahl]
    set erster item 0 ersterlist
    set zweiter item 0 zweiterlist
    ask turtle erster [create-umrahmung-with turtle zweiter]
    set ersterlist [who] of turtles with [_typ = 2 and _zahl = i_anzahl]
    set zweiterlist [who] of turtles with [_typ = 4 and _zahl = i_anzahl]
    set erster item 0 ersterlist
    set zweiter item 0 zweiterlist
    ask turtle erster [create-umrahmung-with turtle zweiter]
    set ersterlist [who] of turtles with [_typ = 4 and _zahl = i_anzahl]
    set zweiterlist [who] of turtles with [_typ = 3 and _zahl = i_anzahl]
    set erster item 0 ersterlist
    set zweiter item 0 zweiterlist
    ask turtle erster [create-umrahmung-with turtle zweiter]
    set ersterlist [who] of turtles with [_typ = 3 and _zahl = i_anzahl]
    set zweiterlist [who] of turtles with [_typ = 1 and _zahl = i_anzahl]
    set erster item 0 ersterlist
    set zweiter item 0 zweiterlist
    ask turtle erster [create-umrahmung-with turtle zweiter]
  ]
  set i_anzahl i_anzahl + 1
]
ask turtles [
  set _edge-of-heading false
  set zahl _zahl
  ask my-links [
    set _zahllink zahl]
]
if global-proportion = true [
  ask turtles [set _maxproportion max-proportion2]
]
reset-ticks
end

to outside
  let grundhelp false
  ask turtles with [_typ = 0] [
    set grundhelpcount 0
    let newzahl _zahl
    ask turtles with [_zahl = newzahl] [ask patch-here [if _grundriss =
false [set grundhelpcount grundhelpcount + 1]]]
    ask patch-here [ifelse _grundriss = false [set grundhelp false] [set
grundhelp true]]
    if grundhelp = true [set _outside-border false]
    if grundhelp = false [set _outside-border true]
    if grundhelpcount > 2 [set _outside-border true]
  ]
end

to-report inside-other?
  let inside-other false
  let other-zahl_1 []
  let other-zahl_2 []
  let other-zahl_3 []
  ask turtles with [xcor <= (xold - xumriss) and ycor <= (yold - yumriss)
and _typ = 1 and _zahl != zahl][
    set other-zahl_1 lput _zahl other-zahl_1
  ]
  ask turtles with [xcor <= (xold - xumriss) and ycor >= (yold + yumriss)
and _typ = 2 and member? _zahl other-zahl_1 and _zahl != zahl] [

```

```

    set other-zahl_2 lput _zahl other-zahl_2
  ]
  ask turtles with [xcor >= (xold + xumriss) and ycor >= (yold + yumriss)
and _typ = 4 and member? _zahl other-zahl_2 and _zahl != zahl] [
    set other-zahl_3 lput _zahl other-zahl_3
  ]
  ifelse empty? other-zahl_3
[report false]
[report true]
end

to-report doppelt-intersection?
  set doppelt-intersection 0
  ask patches with [pxcor >= (xold - xumriss - 0.5) and pxcor <= (xold +
xumriss + 0.5) and pycor >= (yold - yumriss - 0.5) and pycor <= (yold +
yumriss + 0.5) ] [
    set doppelt-intersection doppelt-intersection + count markers-here
  ]
  ifelse doppelt-intersection = 0
[report false]
[report true]
end

to-report outside-border?
  let outside-border 0
  ask patch-here [if _grundriss = false [
    set outside-border 1]
  ]
  ifelse outside-border = 1
[report true]
[report false]
end

to-report other-outside-border?
  let other-outside-border 0
  ask turtles with [_zahl = zahl] [
    ask patch-here [if _grundriss = false [
      set other-outside-border other-outside-border + 1]
    ]
  ]
  ifelse other-outside-border >= 3
[report true]
[report false]
end

to grundrisskontrolle
  set grundrisstest false
  ask patches with [_grundriss = false] [if any? turtles-here with [_typ =
0] [set grundrisstest true]]
end

to kontrolle
  set zahl count patches with [_boolbelegt = true]
end

to move-it
  EINGREIFEN_WILENSKY2
  outside
  if gesamtgroesse > (world-width * world-height) [print "Die Raeume
koennen nicht angeordnet werden da die Gesamtflaeche zu groß ist" stop]
  set besetztflaeche 0

```

```

go-intersect
while [count markers != 0 and pause = false or count turtles with [_typ =
0 and _outside-border = true] != 0 or inside-other? = true] [
  if show-adjacencies = false [ask beziehungen [set hidden? true]]
  if show-adjacencies = true [ask beziehungen [set hidden? false]]
  ask turtles with [_typ = 0 and shape = "target"]
  [set xold xcor
  set yold ycor
  set xumriss _xumriss
  set yumriss _yumriss
  set zahl _zahl
  if outside-border? = true or other-outside-border? = true [
    face min-one-of patches with [_grundriss = true] [distance myself]
    set thisheading heading
    ask turtles with [_zahl = zahl] [
      set heading thisheading
      set maxproportion _maxproportion
    ]
    set verschieben-oder-proportion random
(abstossungswahrscheinlichkeit + proportionswahrscheinlichkeit - 1)
    let drehversuch_typ 20
    let drehversuch 20
    ifelse verschieben-oder-proportion > abstossungswahrscheinlichkeit
    [
      if heading >= 90 and heading < 180 or heading >= 270 and heading
< 360 [
        set xumrissneu xumriss - 0.5
        set yumrissneu (_realize / (2 * _xumriss)) / 2
        get-proportion
      ]
      if heading >= 0 and heading < 90 or heading >= 180 and heading <
270
      [
        set yumrissneu yumriss - 0.5
        set xumrissneu (_realize / (2 * _yumriss)) / 2
        get-proportion
      ]
    ]
    get-my-variables
    while [not can-prop-all? and drehversuch_typ > 0] [
      set drehversuch_typ drehversuch_typ - 1
    ]
    if can-prop-all? [
      set _xumriss xumrissneu
      set xumriss _xumriss
      set _yumriss yumrissneu
      set yumriss _yumriss
      change-proportion-guys
    ]
  ]
  [
    while [not can-move-all? and drehversuch_typ > 0] [
      set moving-distance 1
      set heading random 360 set thisheading heading set
drehversuch_typ drehversuch_typ - 1
      ask turtles with [_zahl = zahl] [set heading thisheading]
      if can-move-all? [
        fd moving-distance
        come-on-guys-2
        get-my-variables
      ]
    ]
  ]
]

```

```

]

if can-move-all? [
  set moving-distance 1
  if pause = true [stop ]
  fd moving-distance
  come-on-guys-2
  get-my-variables
]
]
get-my-variables
]
if inside-other? [
  if can-move-all? [
    fd moving-distance
    come-on-guys-2
    set heading random 360 set thisheading heading
    ask turtles with [_zahl = zahl] [set heading thisheading]
  ]
  get-my-variables
]
ifelse doppelt-intersection? = true[
  set thisheading heading
  set moving-distance 64
  ask turtles with [_zahl = zahl and _typ > 0] [if (distance min-one-
of markers [distance myself]) / 2 < moving-distance [set moving-distance
(distance min-one-of markers [distance myself]) / 2 + 0.1
  face min-one-of markers [distance myself]
  ;set heading heading + 180
  set thisheading heading]]
  ask turtles with [_zahl = zahl and _typ > 0] [if (distance min-one-
of turtles with [_zahl != zahl and _typ = 0] [distance myself]) / 2 <
_xumriss [set moving-distance (distance min-one-of turtles with [_zahl !=
zahl and _typ = 0] [distance myself]) / 2 + 0.1
  face min-one-of turtles with [_zahl != zahl and _typ = 0]
[distance myself]
  set thisheading heading - 180]]
  ask turtles with [_zahl = zahl][
  set heading thisheading
  set maxproportion _maxproportion]
  let drehversuch 20
  set verschieben-oder-proportion random
(abstossungswahrscheinlichkeit + proportionswahrscheinlichkeit - 1)
  if dont-change-proportion = true [set verschieben-oder-proportion
0]
  ifelse verschieben-oder-proportion > abstossungswahrscheinlichkeit
  [
    if heading >= 90 and heading < 180 or heading >= 270 and
heading < 360 [
      set xumrissneu xumriss - 0.5
      set yumrissneu (_realsize / (2 * _xumriss)) / 2
      get-proportion
    ]
    if heading >= 0 and heading < 90 or heading >= 180 and heading
< 270
  [
    set yumrissneu yumriss - 0.5
    set xumrissneu (_realsize / (2 * _yumriss)) / 2
    get-proportion
  ]
  get-my-variables
]

```

```

while [not can-prop-all? and drehversuch > 0] [
  set drehversuch drehversuch - 1
]
if can-prop-all? [
  set _xumriss xumrissneu
  set xumriss _xumriss
  set _yumriss yumrissneu
  set yumriss _yumriss
  change-proportion-guys
]
]
[
while [not can-move-all? and drehversuch > 0] [
  set moving-distance 0.5
  set heading random 360 set thisheading heading set
drehversuch drehversuch - 1
  ask turtles with [_zahl = zahl] [set heading thisheading]
]
if can-move-all? [
  fd moving-distance
  come-on-guys-2
  get-my-variables
]
]
get-my-variables
]
[set fehlerzaehler fehlerzaehler - 1]
let thissize _realsize
if any? my-beziehungen with [_beziehungswert = 2 and link-length > 2
* sqrt thissize] [
  set thisheading 0
  let thisx xcor
  let thisy ycor
  ask my-beziehungen with [_beziehungswert = 2 and link-length > 2 *
sqrt thissize] [
    let xface [xcor] of other-end
    let yface [ycor] of other-end
    set thisheading thisheading + (atan (xface - thisx) (yface -
thisy))
  ]
  set thisheading thisheading / count my-links with [_beziehungswert
= 2 and link-length > 2 * sqrt thissize]
  set heading thisheading
  set moving-distance 0.5
  ask turtles with [_zahl = zahl] [set heading thisheading]
  if can-move-all? [
    fd moving-distance
    come-on-guys-2
    get-my-variables
  ]
]
if any? my-beziehungen with [_beziehungswert = 0 and link-length < 20
] [
  set thisheading 0
  let thisx xcor
  let thisy ycor
  ask my-beziehungen with [_beziehungswert = 0 and link-length < 20 ]
[
  let xface [xcor] of other-end
  let yface [ycor] of other-end

```

```

        set thisheading thisheading + (atan (thisx - xface) (thisy -
yface))
    ]
    set thisheading thisheading / count my-links with [_beziehungswert
= 0 and link-length < 20]
    ask turtles with [_zahl = zahl] [set heading thisheading]
    if can-move-all? [
        fd moving-distance
        come-on-guys-2
        get-my-variables
    ]
    ]
    ]
    go-intersect
    outside
    count-falsepatches
    tick
]
ask turtles with [_typ = 0] [set xold xcor set yold ycor get-my-variables
if inside-other? [show "You may want to interact with me"]]
print "Finished!"
stop
end

to count-falsepatches
    let i 0
    ask patches with [_grundriss = false] [if count turtles-here > 0 [set i i
+ 1]]
    set plot-patchesfalse i
end

to get-proportion
    ifelse xumrissneu <= yumrissneu
    [set _proportion (xumrissneu * 2) / (yumrissneu * 2)]
    [set _proportion (yumrissneu * 2) / (xumrissneu * 2)]
end

to get-my-variables
    set x xcor
    set y ycor
    set _x xcor
    set _y ycor
    set xumriss _xumriss
    set yumriss _yumriss
end

to-report can-move-all?
    set can-move-all 0
    ask turtles with [_zahl = zahl and _typ > 0] [
        if can-move? moving-distance = true [set can-move-all can-move-all + 1]
    ]
    ifelse can-move-all = 4
    [report true]
    [report false]
end

to-report can-prop-all?
    ifelse x - xumrissneu >= 0 and x + xumrissneu <= 64 and y - yumrissneu >=
0 and y + yumrissneu <= 64 and _proportion >= maxproportion
    [report true]
    [report false]

```

```

end

to get-edge-of-heading
  let edge-of-heading 0
  ask turtles with [_typ = 0] [
    let room _zahl
    if heading > 0 and heading < 90 [set edge-of-heading 4]
    if heading > 90 and heading < 180 [set edge-of-heading 3]
    if heading > 180 and heading < 270 [set edge-of-heading 1]
    if heading > 270 and heading < 360 [set edge-of-heading 2]
    if heading = 0 [set edge-of-heading 14]
    if heading = 90 [set edge-of-heading 12]
    if heading = 180 [set edge-of-heading 23]
    if heading = 270 [set edge-of-heading 34]
    ask turtles with [_zahl = room] [set _edge-of-heading false]
    ask turtles with [_typ = _edge-of-heading and _zahl = room] [set _edge-
of-heading true set color green]
  ]
end

to get-doppelt
  let doppeltest false
  let anzahl-turtles 0
  ask turtles with [_typ = 0] [
    set x _x
    set y _y
    set xumriss _xumriss
    set yumriss _yumriss
    set anzahl-turtles count turtles with [xcor >= x - xumriss and xcor <= x
+ xumriss and ycor >= y - yumriss and ycor <= y + yumriss]
    show anzahl-turtles
    ifelse anzahl-turtles > 5 [
      set doppeltest true
      set _doppeltest true
    ]
    [ set doppeltest false
      set _doppeltest false]
    show doppeltest
  ]
end

to come-on-guys
  ask turtles with [_zahl = zahl and _typ = 1 and shape = "target"] [
    set xcor x - xumriss set ycor y - yumriss]
  ask turtles with [_zahl = zahl and _typ = 2 and shape = "target"] [
    set xcor x - xumriss set ycor y + yumriss]
  ask turtles with [_zahl = zahl and _typ = 3 and shape = "target"] [
    set xcor x + xumriss set ycor y - yumriss]
  ask turtles with [_zahl = zahl and _typ = 4 and shape = "target"] [
    set xcor x + xumriss set ycor y + yumriss]
end

to come-on-guys-2
  ask turtles with [_zahl = zahl and _typ > 0 ] [
    fd moving-distance ]
end

to change-proportion-guys
  ask turtles with [_zahl = zahl and _typ = 1 and shape = "target"] [
    set xcor x - xumriss set ycor y - yumriss]
  ask turtles with [_zahl = zahl and _typ = 2 and shape = "target"] [

```

```

    set xcor x - xumriss set ycor y + yumriss]
ask turtles with [_zahl = zahl and _typ = 3 and shape = "target"] [
    set xcor x + xumriss set ycor y - yumriss]
ask turtles with [_zahl = zahl and _typ = 4 and shape = "target"] [
    set xcor x + xumriss set ycor y + yumriss]
end

to export-eps
set exportname word geschossanzahl "_"
set exportname word exportname Import-Parameter
set exportname word exportname "_"
set exportname word exportname geschoss
vectorview:export-view-vector "EPS" 2 exportname
export-world exportname
test_timetable_student
set exportname word exportname "evaluation"
vectorview:export-view-vector "EPS" 2 exportname
set exportname word exportname ".png"
export-view exportname
end

to import-solution
import-world importfile
end

to export-pdf
vectorview:export-view-vector "PDF" 2 exportname
end

to export-svg
vectorview:export-view-vector "SVG" 2 exportname
end

to print-a-csv
let file user-file
file-open file
while [not file-at-end?] [
    print file-read
]
print "finished"
file-close
end

to linklength
let ad 0
let anzbez count beziehungen with [_beziehungswert = 2]
ask beziehungen with [_beziehungswert = 2] [set ad ad + link-length]
set ad ad / anzbez
print ad
end

to bereinig
ask turtles with [_typ = 0] [set _passierende-tester 0]
ask turtles with [_typ = 0] [set size 2]
ask patches with [_grundriss = true] [set pcolor black]
ask patches with [_grundriss = false] [set pcolor white]
end

to belegt
print count patches with [_grundriss = true and pcolor != black]
end

```

```

to clear
  clear-all
end

to clear_pen
  ask turtles[die]
  clear-drawing
end

;intersecting-links.nls
breed [ markers marker ]
breed [ endpoints endpoint ]

links-own [
  _intersected
]

to setup
  clear-all
  set-default-shape markers "circle"
  create-endpoints 9
  [
    setxy random-xcor random-ycor
  ]
  ask endpoints
  [
    hatch-endpoints 1
    [
      setxy random-xcor random-ycor
      create-umrahmung-with myself
    ]
  ]
  place-markers
  reset-ticks
end

to go-intersect
  ;ask endpoints [ rt random-float 10 fd 0.1 ]
  place-markers
  ;tick
end

to place-markers
  ask markers [ die ]
  ;; each pair of segments checks for intersections once
  ask umrahmungen [
    let hilfzahl _zahllink;; performing this check on the who numbers
    keeps
    ;; us from checking every pair twice self > myself and
    ask umrahmungen with [_zahllink != hilfzahl] [
      let result intersection self myself
      if not empty? result [
        set _intersected true
        ask end1 [
          hatch-markers 1 [
            set color gray
            set size 1
            setxy (first result) (last result)
            set _typ -1
          ]
        ]
      ]
    ]
  ]

```

```

    ]
  ]
]

end

;; reports a two-item list of x and y coordinates, or an empty
;; list if no intersection is found
to-report intersection [t1 t2]
  let m1 [tan (90 - link-heading)] of t1
  let m2 [tan (90 - link-heading)] of t2
  ;; treat parallel/collinear lines as non-intersecting
  if m1 = m2 [ report [] ]
  ;; is t1 vertical? if so, swap the two turtles
  if abs m1 = tan 90
  [
    ifelse abs m2 = tan 90
      [ report [] ]
      [ report intersection t2 t1 ]
  ]
  ;; is t2 vertical? if so, handle specially
  if abs m2 = tan 90 [
    ;; represent t1 line in slope-intercept form (y=mx+c)
    let c1 [link-ycor - link-xcor * m1] of t1
    ;; t2 is vertical so we know x already
    let x-link [link-xcor] of t2
    ;; solve for y
    let y-link m1 * x + c1
    ;; check if intersection point lies on both segments
    if not [x-within? x-link] of t1 [ report [] ]
    if not [y-within? y-link] of t2 [ report [] ]
    report list x-link y-link
  ]
  ;; now handle the normal case where neither turtle is vertical;
  ;; start by representing lines in slope-intercept form (y=mx+c)
  let c1 [link-ycor - link-xcor * m1] of t1
  let c2 [link-ycor - link-xcor * m2] of t2
  ;; now solve for x
  let x-link (c2 - c1) / (m1 - m2)
  ;; check if intersection point lies on both segments
  if not [x-within? x-link] of t1 [ report [] ]
  if not [x-within? x-link] of t2 [ report [] ]
  report list x (m1 * x-link + c1)
end

to-report x-within? [x-link] ;; turtle procedure
  report abs (link-xcor - x-link) <= abs (link-length / 2 * sin link-
heading)
end

to-report y-within? [y-link] ;; turtle procedure
  report abs (link-ycor - y-link) <= abs (link-length / 2 * cos link-
heading)
end

to-report link-xcor
  report ([xcor] of end1 + [xcor] of end2) / 2
end

to-report link-ycor

```

```

    report ([ycor] of end1 + [ycor] of end2) / 2
end

; Based upon source by Uri Wilensky (which was placed under public domain)
; | die Funktion basiert auf Code von Uri Wilensky, veröffentlicht unter
; public domain

;interact.nls
to EINGREIFEN_WILENSKY2

    if mouse-down? [
        ask students [die]
        let candidate min-one-of turtles with [_typ = 0 and shape = "target"]
        [distancexy mouse-xcor mouse-ycor]
        if [distancexy mouse-xcor mouse-ycor] of candidate < 1 [
            watch candidate
            ask subject [
                set xold _x
                set yold _y
                set xumriss _xumriss
                set yumriss _yumriss ]
            ;; The WATCH primitive puts a "halo" around the watched turtle.

            while [mouse-down?] [
                ;; If we don't force the view to update, the user won't
                ;; be able to see the turtle moving around.
                display
                ;; The SUBJECT primitive reports the turtle being watched.
                ask subject [ setxy mouse-xcor mouse-ycor
                    set _x xcor
                    set _y ycor
                    set x _x
                    set y _y
                    set zahl _zahl
                ]
            ]
            come-on-guys
            reset-perspective
        ]
    ]
    if show-adjacencies = "show-none" [ask beziehungen [set hidden? true]]
    if show-adjacencies = "show-all" [ask beziehungen [set hidden? false]]
    if show-adjacencies = "show-positive-negative" [ask beziehungen with
    [color = red or color = green] [set hidden? false] ask beziehungen with
    [color = grey] [set hidden? true]]
    if show-adjacencies = "show-positive-adjacencies" [ask beziehungen with
    [color = green] [set hidden? false] ask beziehungen with [color = grey or
    color = red] [set hidden? true]]
    if show-adjacencies = "show-negative-adjacencies" [ask beziehungen with
    [color = red] [set hidden? false] ask beziehungen with [color = grey or
    color = green] [set hidden? true]]
    if show-labels = true [ask turtles with [_typ = 0] [set label
    _bezeichnung]]
    if show-labels = false [ask turtles with [_typ = 0] [set label ""]]
end

```

; Based upon source by Uri Wilensky (which was placed under public domain)
| die Funktion basiert auf Code von Uri Wilensky, veröffentlicht unter
public domain

```
;beurteilung.nls
```

```
globals [  
  passierende-tester  
  roomnumber  
]  
  
to test_timetable_student  
  ask patches [set _roomnumber 1000]  
  ask turtles with [shape = "target" and _typ = 0] [  
    get-my-variables  
    set _x xcor  
    set _y ycor  
    set x _x  
    set y _y  
    set xumriss _xumriss  
    set yumriss _yumriss  
    set farbuebergabe color  
    set zahl _zahl  
    ask patches with [pxcor >= (x - xumriss) and pxcor <= (x + xumriss) and  
pycor >= (y - yumriss) and pycor <= (y + yumriss) ];and _grundriss = true  
    [  
      set pcolor farbuebergabe  
      set _roomnumber zahl  
    ]  
  ]  
  ask students [die]  
  set agent_index 0  
  let roomscounter 0  
  let zielrichtung 0  
  while [agent_index <= anzahl_rooms - 1] [  
    ask turtles with [shape = "target" and _typ = 0 and _zahl =  
agent_index] [  
      while [roomscounter <= anzahl_rooms - 1] [  
        set _goal roomscounter  
        hatch-students 1  
        set roomscounter roomscounter + 1  
      ]  
    ]  
    set roomscounter 0  
    set agent_index agent_index + 1  
  ]  
  ask turtles with [shape = "target"] [set _goal ""]  
  ask students with [_zahl = _goal] [die]  
  ask students [set size 2  
    set goal _goal  
    ask turtles with [shape = "target" and _typ = 0 and _zahl = goal] [set  
zielrichtung who]  
    face turtle zielrichtung  
    set label zielrichtung  
    while [distance turtle zielrichtung > 1] [  
      ask patch-here [set oldroom _roomnumber]  
      fd 1  
      test-patches  
    ]  
  ]  
]
```

```

ask students [die]
ask turtles with [shape = "target" and _typ = 0] [
  set size _passierende-tester / anzahl_rooms ]
end

to test-patches
  let bo-test-patches false
  ask patch-here [if _roomnumber != oldroom and _roomnumber != goal and
pcolor != black and pcolor != white [set bo-test-patches true set
roomnumber _roomnumber]]
  if bo-test-patches = true
  [ask turtles with [shape = "target" and _typ = 0 and _zahl = roomnumber][
    set _passierende-tester _passierende-tester + 1]
  ]
end

;import.nls

to load_parameters
  ask patches [set _grundriss false]
  import-pcolors import-parameter
  ask patches[
    if pcolor != 0 [
      if pcolor = white [
        set _grundriss true
        set pcolor black
      ]
    ]
  ]
  ask patches with [_grundriss = false] [
    if pcolor = 0 [
      set pcolor white]
  ]
end

```