

Das tucseating-Paket

Version 0.2

Matthias Werner

13. Juli 2025

Das Paket `tucseating` ermöglicht es, Sitzpläne, wie sie z. B. für Prüfungen benötigt werden, einfach zu erstellen. Eine Reihe verschiedener automatischer Platzierungsschemata sind vordefiniert, aber man kann auch feingranular eigene Platzierungen vornehmen. Während das Paket zunächst für den internen Gebrauch an der TU Chemnitz gedacht ist und (einige) vordefinierte Raumpläne enthält, sind einerseits sowohl die Raumdaten leicht erweiter- oder ersetzbar, andererseits können Räume auch ad hoc erstellt werden.

Inhaltsverzeichnis

1	Einführung	2
2	Abhängigkeiten	2
3	Sitzlayout	2
3.1	Klassensoptionen	2
3.2	Späte Optionswahl	4
3.3	Modifikation des Sitzlayouts	4
3.4	Ausgabe	5
4	Sitzplatzbelegung	6
4.1	Parameter zur Konstruktion von Belegungsschemata	6
4.2	Vordefinierte Sitzschemata	7
4.3	Sitzbeschriftung	8
5	Beispiele	8
6	Deklaration neuer Räume	9
7	Fehler	9

1 Einführung

Für die Durchführung von Prüfungen benötigen wir mitunter Sitzpläne. So haben sich über die einige mit Pläne in Form von TikZ-unterstützten \LaTeX -Dateien angesammelt. Je nach Anzahl der Studierenden in einer Prüfung (und für wie groß wir die Gefahr eines Betrugsversuches bewerten) nutzen wir unterschiedliche Platzierungsschemata, so dass wir die Dateien in anpassen müssen. Außerdem wird uns von Zeit zu Zeit ein neuer Raum zugewiesen, für den wir noch keine Pläne haben.

Dies war die Motivation zur Erschaffung des `tucseating`-Pakets. Es...

- ermöglicht eine schnelle und einfach Erstellung von Sitzplänen;
- trennt das Raumlayout und das Platzierungsschema voneinander;
- bietet eine Reihe von Standardschemata für die Platzierung an;
- enthält bereits eine Anzahl vordefinierter Layouts der Sitze;
- erlaubt eine *Ad-hoc*-Erstellung neuer Räume und Platzierungsschemata.

2 Abhängigkeiten

Das `tucseating`-Paket arbeitet nur mit $\text{Lua}\text{\LaTeX}$ und benötigt eine hinreichend moderne \LaTeX -Version, mindestens vom Juli 2022. Es lädt folgende Pakete:

- `etoolbox`
- `luacode`
- `tikz`

Diese Pakete sind in allen gängigen \TeX -Distributionen vorhanden.

3 Sitzlayout

Das Paket wird auf dem üblichen Weg mit `\usepackage[optionen]{tucseating}` geladen. Dabei kann bereits weitgehend das Layout des Sitzplans festgelegt werden, also die Darstellung der Sitze im Raum.

3.1 Klassenoptionen

Durch einige der Optionen werden zwei grundsätzliche Anwendungsfälle unterschieden.

`room = \langle Raum \rangle`

Diesen Optionsschlüssel sollte man nutzen, wenn der gewünschte Raum ist bereits in der Datenbank von `tucseating` vorhanden ist.

3 Sitzlayout

Falls der Raum noch nicht bekannt ist, werden einige Schlüssel-Wert-Paar zur Beschreibung des Raumlaysout gebraucht.

`shape = rectangle|arc` Voreinstellung: `rectangle`
Hier wird beschrieben, ob das Layout der Sitze rechteckig (`rectangle`) oder bogenförmig (`arc`)¹ ist.

`rows = \langle Anzahl der Sitzreihen \rangle` (erforderlich)

`seats per row = \langle Sitze pro Sitzreihe \rangle` (erforderlich)

Anzahl der Sitzreihen und Sitze pro Reihe. Dabei muss man immer von der jeweils maximalen Anzahl ausgehen. Für unvollständige Reihen werden später Sitze entfernt, aber es können keine Sitze mehr zugefügt werden, die außerhalb des einmal festgelegten Layoutrahmens liegen. **Achtung:** Auch Gänge zwischen Blöcken von Sitzen müssen hier wie Sitze gezählt werden.

Wenn Sie bei den Klassenoptionen **sowohl** den Schlüssel `room`, **als auch** eine der Schlüssel `rows` oder `seats per row` angeben, ist das Ergebnis unbestimmt. Entsprechend wird eine Warnung ausgegeben.

Alle weiteren Klassenoptionen legen die Darstellung des Raumlaysouts fest. Sie können auch später mit `\tucsConfig` gesetzt werden.

`blackboard = true|false` Voreinstellung: `false`
Zeichnet eine Tafel ein.

`seat distance = \langle Abstand \rangle` Voreinstellung: `2pt`
Abstand der Sitze zueinander. Damit wird auch der Reihenabstand bestimmt. Möchte man beides unterschiedlich haben, so muss man die beiden folgenden Schlüssel nutzen:

`seat neighbor distance = \langle Abstand \rangle` Voreinstellung: `2pt`

`row distance = \langle Abstand \rangle` Voreinstellung: `2pt`

`rownumbers = none|left|right|both` Voreinstellung: `none`
Legt beim Rechteck-Layout fest, ob links und oder rechts der Reihen die Nummer der Sitzreihe angegeben wird. Dabei wird von vorn (Tafel) gezählt.

Anmerkung: Für das bogenförmige Layout ist diese Funktion derzeit nicht implementiert.

`rownumber distance = \langle Abstand \rangle` Voreinstellung: `2pt`
Abstand der Reihennummern zu den äußeren Sitzen, wenn `rownumbers` nicht `none` ist.

`empty seat background color = \langle Farbe \rangle` Voreinstellung: `lightgray!20`

`empty seat border color = \langle Farbe \rangle` Voreinstellung: `lightgray`

`assigned seat background color = \langle Farbe \rangle` Voreinstellung: `lightgray!30`

1. Z. B. an der TU Chemnitz der Raum A10.316.

`assigned seat border color = $\langle\text{Farbe}\rangle$` Voreinstellung: `black`

Legt die Farben für den Hintergrund und die Umrandung von leeren bzw. durch ein Sitzschema belegten Sitzen fest. Falls die Sitze nicht farblich unterschieden werden sollen, können auch die Schlüssel

`seat background color = $\langle\text{Farbe}\rangle$` (zunächst leer)

`seat border color = $\langle\text{Farbe}\rangle$` (zunächst leer)
verwendet werden.

`assigned seat label font = $\{\langle\text{Fontbefehl}\rangle\}$` Voreinstellung: `\small`

`assigned seat label color = $\langle\text{Farbe}\rangle$` Voreinstellung: `black`
Setzt die Größe und Farbe der Sitzbeschriftung.

3.2 Späte Optionswahl

`\tucsConfig`

Mit diesem Kommando können außer `room`, `shape`, `rows` und `seats per row` alle im Abschnitt 3.1 genannten Schlüssel auch außerhalb von `\documentclass`-Optionen gesetzt werden.

3.3 Modifikation des Sitzlayouts

Häufig ist sind nicht in jeder Reihe alle Sitze vorhanden. Bei Angabe eines Raums ist dies bereits berücksichtigt, aber auch hier kann es vorkommen, dass beispielsweise ein Klappsitz defekt ist und aus dem Layout genommen werden muss. Beim Erstellen eines eigenen Layouts ist die Modifikation die Regel. Dafür stellt folgende Kommandos zur Verfügung

`\tucsRemoveSeatAt $\{\langle\text{Reihe}\rangle\}\{\langle\text{Sitznummer}\rangle\}$`

Entfernt den Sitz $\langle\text{Sitznummer}\rangle$ in der Reihe $\langle\text{Reihe}\rangle$ aus dem Layout. Dabei beziehen sich $\langle\text{Reihe}\rangle$ und $\langle\text{Sitznummer}\rangle$ auf das vollständige Layout, ändern sich also nicht durch bereits entfernte Sitze. Wenn die $\langle\text{Sitznummer}\rangle$ negativ ist, wird von rechts aus gezählt

`\tucsRemoveSeats $\{\langle\text{Liste}\rangle\}$`

Entfernen alle in der Liste vorkommenden Sitze aus dem Layout. Die Liste enthält dabei kommaseparierte Einträge der Form $\{\langle\text{Reihe}\rangle,\langle\text{Sitznummer}\rangle\}$. Für $\langle\text{Reihe}\rangle$ und $\langle\text{Sitznummer}\rangle$ gilt wie bei `\tucsRemoveSeatAt` das ursprüngliche Layout. Im folgenden Beispiel werden die ersten drei Sitze links und der erste Sitz rechts in der ersten Reihe entfernt:

```
1 \tucsRemoveSeats $\{\{1,1\},\{1,2\},\{1,3\}\{1,-1\}\}$ 
```

`\tucsSetAisle[⟨Startreihe⟩-⟨Endreihe⟩]{⟨Sitznummer⟩}`

An der Stelle von $\langle \text{Sitznummer} \rangle$ wird durch alle Reihen ein Gang eingefügt. Nutzen Sie das optionale Argument, wenn der Gang nicht alle Reihen erfassen soll (Stichgang). Für horizontale Gänge (die also einen Sitzblock in einen vorderen und hinteren Teil zerlegen, anstatt einen rechten und einen linken) nutzen Sie bitte `\tucsRemoveSeats`. Das ist im Prinzip auch bei vertikalen Gängen möglich, jedoch können bei der automatischen Sitzschemazuweisung mit `\tucsSetAisle` erzeugte Gänge anders als die mit `\tucsRemoveSeats` erzeugten Gänge behandelt werden.

3.4 Ausgabe

`\tucsDrawSeating`

Gibt den Sitzplan aus. Dabei versucht `tucseating`, die Ausmaße der Sitz so zu berechnen, dass der zur Verfügung stehenden Platz des Gesamtplans vollständig ausgenutzt wird.

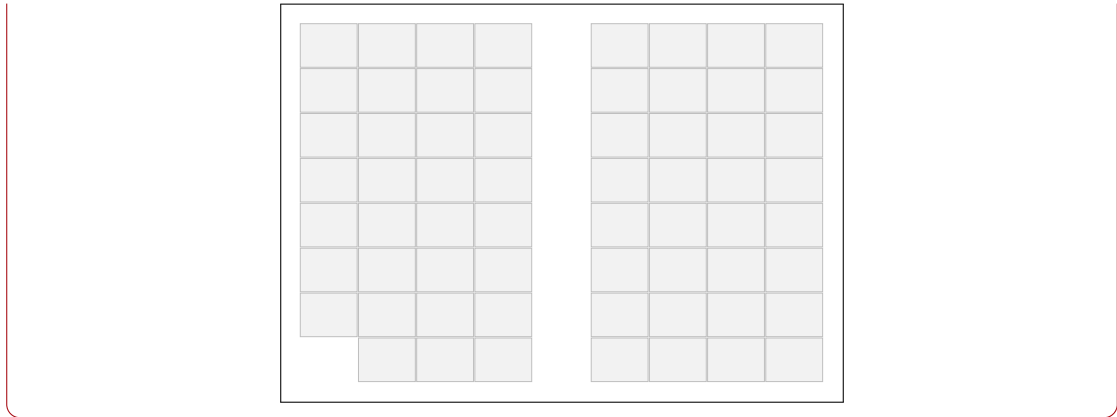
`\tucsDrawSeating*[seat width=⟨Breite⟩, seat height=⟨Höhe⟩]`

Da `\tucsDrawSeating` nicht immer alle Bedürfnisse trifft, können in der Sternvariante über das optionale Argument die Breite und Höhe der Sitze auf dem Sitzplan einzeln eingestellt werden.

```

1 \documentclass{scrartcl}
2 % Um das Papier gut auszunutzen, nutzen wir es in
3 % Landscape und reduzieren die Ränder
4 \usepackage[a4paper,landscape,inner=1cm,outer=1cm,top=1cm,bottom=1cm]{
  geometry}
5 \usepackage[
6   shape=rectangle,
7   rows = 8,
8   seats per row=9,
9 ]{tucseating}
10
11 \begin{document}
12   \tucsSetAisle{5}
13   \tucsRemoveSeatAt{1}{1}
14   \tucsDrawSeating
15 \end{document}

```



Der vorstehende Code zeigt Erzeugung Darstellung eines einfachen Sitzplanlayouts mit einem Mittelgang und einem fehlenden Sitz in der ersten Reihe.

4 Sitzplatzbelegung

4.1 Parameter zur Konstruktion von Belegungsschemata

Natürlich ist ein Sitzplan ohne die Markierung von belegten Plätzen etwas witzlos. Zur Erzeugung eines Sitzschemas gibt es den Befehl

```
\tucsSeatingScheme*[<Schlüsselliste>]{<Schema>}
```

Das Pflichtargument enthält in der Standardvariante des Befehls eine Bezeichnung für ein vorgegebenes Sitzschema, siehe 4.2. In der Sternvariante wird hier eine Zeichenkette von „X“ und „-“ übergeben, die (den Anfang) eines Sitzschemas für eine einzelne Reihe beschreibt, wobei „X“ für einen belegten und „-“ für einen leeren Platz steht. Ist die Zeichenkette kürzer als die Anzahl der Sitze in der Reihe, wird sie wiederholt angewendet. Reihe mehr Plätze als Beispielsweise wird mit

```
1 \tucsSeatingScheme{X--}
```

jeder dritte Sitz belegt.

Im optionalen Argument kann wieder eine Liste von Schlüssel-Wert-Paaren übergeben werden, die die restlichen Einstellung für das Sitzschema steuern. Es ist insbesondere in der Sternvariante wichtig, kann aber auch in der Standardvariante genutzt werden. werden:

row sep = *<Anzahl>*

Voreinstellung: 2

Legt fest, in jeder wievielten Reihe Sitze belegt werden.

start row = *<Nummer>*

Voreinstellung: 1

end row = $\langle \text{Nummer} \rangle$ Voreinstellung: $\langle \text{Anzahl Reihen} \rangle$
 Durch diese beiden Schlüssel wird gesteuert, in welchen Reihen ein Schema angewendet wird. Dadurch können unterschiedliche Schemata für verschiedene Reihen genutzt werden.

aisle counts = $\langle \text{Anzahl} \rangle$ Voreinstellung: 1
 Legt fest, wie viele Sitze ein Gang zählt. Dadurch kann berücksichtigt werden, dass ein Gang häufig breiter als ein Sitzplatz ist.

aisle restarts scheme = true|false Voreinstellung: false
 Startet das Schema nach einem Gang erneut. Der Schlüssel **aisle counts** wird damit wirkungslos.

ignore aisle = true|false Voreinstellung: false

ignore removed seats = true|false Voreinstellung: false
 Bei der Zählung von Sitzen, die im Platzlabel genutzt werden kann (vgl. Abschnitt 4.3) werden auch entfernte Sitze und Gänge mitgezählt. Dies ist sinnvoll, damit in Reihen (im Rechteck-Grundlayout) sich gleiche Sitznummern hintereinander befinden. Wird einer der Schlüssel gesetzt, werden entfernte Sitze bzw. Gänge nicht mitgezählt. Dies bietet sich insbesondere im Bogen-Grundlayout an.

assigned seat label = $\{\langle \text{Formatzeichenkette} \rangle\}$ Voreinstellung: $m\{\{\backslash, \}\}D$
 Über diesen Schlüssel kann festgelegt werden, wie die belegten Sitze beschriftet werden. Voreingestellt ist die Nummer der Reihe, gefolgt von einem schmalen Leerzeichen und einem Buchstaben für den laufenden belegten Sitz (z. B. „3 c“). Es sind hier eine Menge anderer Möglichkeiten einstellbar. Details dazu sind im Abschnitt 4.3 beschrieben.

4.2 Vordefinierte Sitzschemata

Das `tucseating`-Paket definiert eine Reihe vordefinierter Platzierungsschemata. Manche haben auch einen Alternativnamen.

1x1
 Jeder Platz ist markiert.
 Alternativname: **all**

2x2
 Zwischen zwei belegten Plätzen ist jeweils ein Platz bzw. eine Reihe Abstand.
 Alternativname: **simple**

2x2-
 Zwischen zwei belegten Plätzen ist jeweils ein freier Platz. Nach einer freien Reihe kommen *zwei* Reihen mit belegten Plätzen. Dies ist das Schema, mit dem in einer Prüfung die größtmögliche Zahl an Studierenden in einem Raum untergebracht werden können, aber trotzdem ein seitlicher Minimalabstand gegeben ist und sich die Aufsicht zu jedem Studierenden kommen kann (entweder von vorn, oder von hinten).
 Alternativname: **dense**

2x3

Belegte Sitze haben einen seitlichen Abstand von zwei Sitzen und belegte Reihen einen Abstand von einer Leerreihe. Dies wird in unser Gruppe als das anzustrebende Standardschem betrachtet.

Alternativname: **sixpack**

4.3 Sitzbeschriftung

Die Beschriftung der zugewiesenen Sitzplätze kann auf unterschiedliche Weise erfolgen, die durch Zuweisung einer Formatzeichenkette an den Schlüssel **assigned seat label** gesteuert wird. Es stehen dafür vier Zähler zur Verfügung:

- absolute Reihe: die Nummer der Reihe im Sitzlayout
- laufende Reihe: die Nummer der *belegten* Reihe. Reihen, in denen keine Sitz zugewiesen werden, werden hier übersprungen.
- absolute Sitznummer: die Nummer des Sitzes im Sitzlayout. Ob hier auch entfernte Sitze berücksichtigt werden, hängt vom Wert des Schlüssels **ignore removed seats** ab.
- laufende Sitznummer: die Nummer der *belegten* Sitze, unbelegte Sitze werden bei der Zählung übersprungen.

Jeder dieser Zähler kann unterschiedlich formatiert werden. Dafür werden in der Formatzeichenkette verschiedene Formatierungszeichen benutzt, die in der folgenden Tabelle gelistet sind:

Wert	Anmerkung	Darstellung als...				
		(arabische) Zahl	Kleinbuchstabe	Großbuchstabe	kleine römische Zahl	große römische Zahl
absolute Reihe	<i>bezieht sich auf das Sitzlayout</i>	m	a	A	y	Y
laufende Reihe	<i>bezieht sich auf das Belegungsschema</i>	r	b	B	i	I
absolute Sitznummer	<i>bezieht sich auf das Sitzlayout</i>	n	c	C	x	X
laufende Sitznummer	<i>bezieht sich auf das Belegungsschema</i>	s	d	D	j	J

Teile der Beschriftung, die nicht als Formatierungszeichen interpretiert werden sollen, werden in doppelte geschweifte Klammern gesetzt (**{{\geschützte Zeichenkette}}**).

5 Beispiele

To be done

6 Deklaration neuer Räume

7 Fehler