

Grid Library Dokumentation

1. Klasse Grid

Die zentrale Klasse, die ein Spielfeld als Canvas darstellt.

Konstruktor

<pre>new Grid(options, rows = 9, columns = 9, width = 400, height = 400)</pre>
--

Parameter	Typ	Beschreibung
options	Array<Class>	Liste von Klassen, die von <code>Cell</code> erben. Diese erscheinen in der <code>SelectBar</code> und können ins Grid gezogen werden.
rows	number	Anzahl der Reihen im Grid.
columns	number	Anzahl der Spalten im Grid.
width	number	Breite des Canvas in Pixeln.
height	number	Höhe des Canvas in Pixeln.

Eigenschaften

Name	Typ	Beschreibung
grid	2D-Array [rows][columns]	Speichert <code>Cell</code> -Instanzen oder <code>null</code> .
other_stuff	Array<OtherStuff>	Zusätzliche Elemente, die nicht ans Grid gebunden sind, aber gezeichnet werden können.
canvas	HTMLCanvasElement	Das eigentliche Spielfeld.
ctx	CanvasRenderingContext2D	Zeichenkontext.
select_bar	SelectBar	Auswahlleiste für neue Elemente.
mouse_pos	{x:number, y:number}	Letzte Mausposition.
dragging	null {element:Cell, from:{row,col}, x, y}	Enthält Infos, wenn gerade ein Element gezogen wird.

Methoden

<p>push(row, col, element): Fügt ein <code>Cell</code> an einer bestimmten Position ein.</p> <p>destroy(): Entfernt Eventlistener, wenn das Grid zerstört wird.</p> <p>_update(): Intern: Zeichnet das Grid und alle Objekte. Läuft über <code>requestAnimationFrame</code>.</p>

Interaktionen

- **Drag&Drop:** `Cell`-Klassen aus der `SelectBar` können per Drag&Drop auf das Grid gesetzt werden.
- **Drag im Grid:** Zellen können mit der Maus angefasst und innerhalb des Grids verschoben werden.
- **Click:** Klickt man auf eine belegte Zelle, wird ihre `onClick()`-Methode ausgeführt. Klickt man auf ein `OtherStuff`-Element, wird dessen `onClick()` aufgerufen.

2. Klasse SelectBar

Eine Auswahlleiste am unteren Bildschirmrand, die die verfügbaren `Cell`-Optionen darstellt.

Konstruktor

<pre>new SelectBar(options, width, height)</pre>
--

Parameter	Typ	Beschreibung
options	Array<Class>	Klassen, die von <code>Cell</code> erben.
width	number	Breite der Vorschaubilder in Pixel.
height	number	Höhe der Vorschaubilder in Pixel.

<p>- Für jede <code>Cell</code>-Klasse wird ein Vorschaubild (Image aus <code>Cell.image</code>) angezeigt.</p> <p>- Der Name (<code>Cell.name</code>) wird angezeigt.</p> <p>- Drag&Drop aktiviert: <code>dragstart</code> liefert Index in <code>options</code> zurück.</p>

3. Klasse Cell

Basisklasse für alle Zellen, die ins Grid eingesetzt werden können.

Konstruktor

<pre>new Cell(src, name)</pre>

Parameter	Typ	Beschreibung
src	string	Pfad zur Bilddatei.
name	string	Anzeigenname der Zelle.

Eigenschaften

Name	Typ	Beschreibung
name	string	Name der Zelle.
image	HTMLImageElement	Bild der Zelle.

Methoden

<p>onClick(): Standard: <code>console.log("test")</code>. Sollte in Subklassen überschrieben werden.</p> <p>draw(x, y, width, height, ctx): Zeichnet das Bild in den gegebenen Bereich des Canvas.</p>
--

4. Klasse OtherStuff

Basisklasse für Objekte, die unabhängig vom Grid existieren (z. B. Rechtecke).

Konstruktor

<pre>new OtherStuff(x, y, width, height)</pre>
--

Parameter	Typ	Beschreibung
x, y	number	Koordinaten (linke obere Ecke).
width, height	number	Größe.

Methoden

<p>draw(ctx): Zeichnet ein Standard-Rechteck (blau). Kann überschrieben werden.</p> <p>update(): Wird jedes Frame aufgerufen. Standard: leer.</p> <p>onClick(): Klick-Handler. Standard: leer.</p>

5. Klasse HTMLStuff

Hilfsklasse zum Erstellen zusätzlicher HTML-Elemente im Dokument.

Konstruktor

<pre>new HTMLStuff()</pre>

Methoden

<p>add(type, content, style = null): Erstellt ein neues HTML-Element. Rückgabe: <code>HTMLElement</code>.</p> <p>change_content(index, content): Ändert den Inhalt des Elements an <code>html[index]</code>.</p> <p>remove(index): Entfernt das Element aus der internen Liste. Achtung: entfernt es nicht automatisch aus dem DOM.</p>
--

6. Klasse Sidebar

Ermöglicht interaktive Sidebar-Elemente, die mit Zellen verbunden werden können.

Konstruktor

<pre>new Sidebar()</pre>

<ul style="list-style-type: none">• Erwartet ein <code><div id="sidebar"></code> im HTML.• Erstellt eine interne <code>div</code>-Box, die die Inhalte enthält.• Fügt automatisch einen „close“-Button hinzu.

Methoden

<p>open(): Öffnet die Sidebar, ersetzt vorhandene Inhalte durch die aktuelle <code>div</code>.</p> <p>close(): Schließt die Sidebar (leert den Inhalt).</p> <p>add(type, args): Fügt ein neues Element hinzu. Rückgabe: erstelltes DOM-Element.</p>
--

Unterstützte Typen bei add

- **div / h1 / h2 / h3:** `args.content` (String), `args.style` (optional)
- **button:** `args.content`, `args.callback`, `args.style` (optional)
- **input:** `args.type`, `args.placeholder`, `args.value`, `args.callback`, `args.style` (optional)
- **range:** `args.min`, `args.max`, `args.value`, `args.step` (Standard 1), `args.callback`, `args.style` (optional)

7. Helper Funktionen

weightedSum(values)

Berechnet den gewichteten Durchschnitt einer Liste von Werten.

- **Parameter:** `values` – Array von Objekten mit `{value:number, weight:number}`.
- **Rückgabewert:** `number` – gewichteter Durchschnitt.
- **Fehler:** Wirft `TypeError`, falls die Eingabe kein Array ist oder Elemente nicht die richtigen Eigenschaften haben.

weightedGroupSum(groups)

Berechnet den gewichteten Durchschnitt mehrerer Gruppen.

- **Parameter:** `groups` – Array von Objekten mit `{values:Array, weight:number}`.
- **Rückgabewert:** `number` – gewichteter Gesamtwert aller Gruppen.
- **Fehler:** Wirft `TypeError`, falls Gruppen oder Werte keine gültigen Arrays/Nummern sind.

buy(amount, balance)

Kauft einen bestimmten Betrag und reduziert das Guthaben.

- **amount:** `number` – Kaufbetrag.
- **balance:** `number` – aktuelles Guthaben.
- **Rückgabewert:** `number` – neues Guthaben.

sell(amount, balance)

Verkauft einen bestimmten Betrag und erhöht das Guthaben.

- **amount:** `number` – Verkaufsbetrag.
- **balance:** `number` – aktuelles Guthaben.
- **Rückgabewert:** `number` – neues Guthaben.

addPopulation(amount, percentages, populationGroups)

Fügt Population zu Gruppen hinzu basierend auf Prozentverteilung.

- **amount:** `number` – zu verteilende Menge.
- **percentages:** Objekt – Prozentwerte pro Gruppe (z. B. `{group1:40, group2:40, group3:20}`).
- **populationGroups:** Objekt – aktuelle Population jeder Gruppe.
- **Rückgabewert:** Objekt – aktualisierte Population.

removePopulation(amount, percentages, populationGroups)

Entfernt Population aus Gruppen basierend auf Prozentverteilung.

- **amount:** `number` – zu entfernende Menge.
- **percentages:** Objekt – Prozentwerte pro Gruppe.
- **populationGroups:** Objekt – aktuelle Population jeder Gruppe.
- **Rückgabewert:** Objekt – aktualisierte Population.