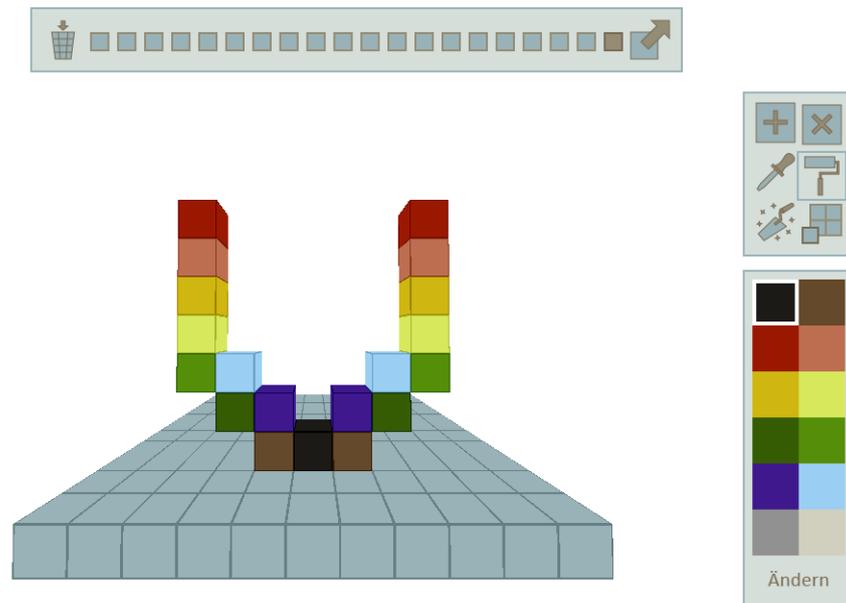


# Voxel Editor (MS Projekt B)



Der **Voxel Editor** ist ein browserbasiertes 3D-Modellierungstool zur einfachen Konstruktion von 3D-Modellen, die ausschließlich aus farbigen Würfeln (genannt Voxel) bestehen.

Neben grundlegenden Funktionen – Voxel setzen, Voxel löschen, Voxelfarbe bestimmen – gibt es zusätzliche Werkzeuge, die auch komplexere Voxel-Modelle effizient und schnell realisierbar machen. Dabei werden unterschiedliche Arbeitsweisen und individuelle Anpassungen der Anwender unterstützt. Zur aktuellen Version auf [Github](#).

## Inhalt

- Inhalt
- Vision
- Konzeption
  - Bedienoberfläche
  - Voxel erstellen und löschen
  - Farbwahl
  - Spezial-Werkzeuge
    - Voxel-Block erzeugen
    - Modell vergrößern
  - Kamera-Bewegung
  - Arbeitsschritte verwalten
  - Workflows
  - Export
- Gestaltung
- Entwicklung
- Inspiration

# Vision

Der Voxel Editor ist ursprünglich als Bestandteil des Terralea-Projekts geplant. Terralea wird ein Browser-MORPG mit einer prozedural generierten, minimalisiert gestalteten Welt. Um diese Welt mit Leben zu füllen, sollen sowohl die Terralea-Entwickler auch Künstler aus der Spieler-Community, im Voxel Editor ein Werkzeug finden, um Details und Charaktere im minimalisierten Stil zu modellieren. Da Terralea vollständig im Browser laufen wird, soll der Voxel Editor ebenfalls browserbasiert und unmittelbar verfügbar sein.

Doch auch andere Spieleentwickler sollen den Voxel Editor nutzen können, um Voxel-Modelle für ihr Spiel zu erstellen. Dazu muss Export in ein möglichst universelles Format angeboten werden.

Außerdem gibt es noch weitere Einsatzgebiete für den Voxel Editor:

- zur Visualisierung von Raum- und Gebäudeplanungen
- als Concept-Art-Tool für Level Designer und Character Designer, evtl. auch als Vorstufe für komplexere 3D-Modelle
- zur Veranschaulichung von Konzepten, die sich am besten räumlich erklären lassen
- im Gebiet des Rapid-Prototyping zum Erzeugen von 3D-Druck-Vorlagen

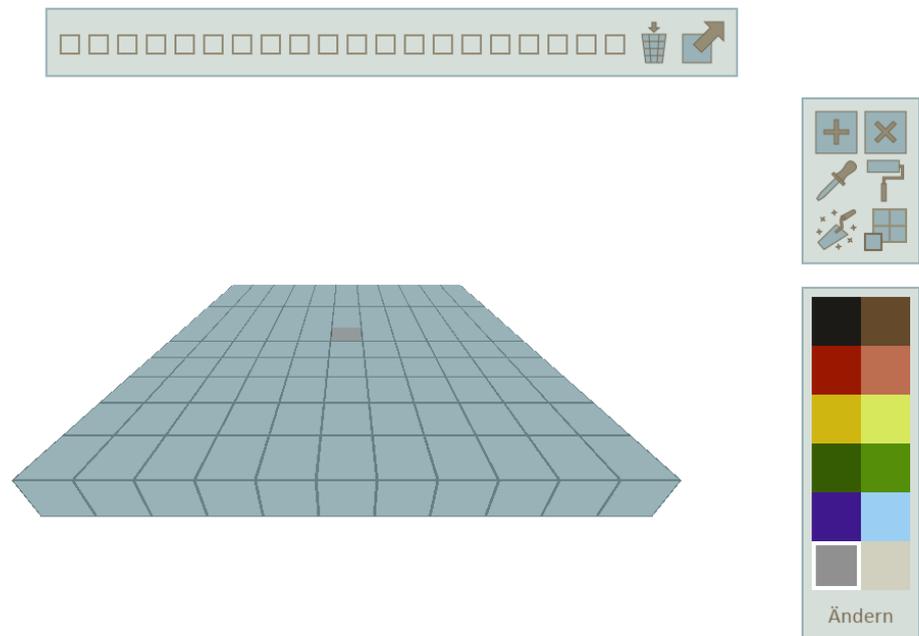
Natürlich lässt sich der Voxel Editor auch zu rein unterhalterischen Zwecken einsetzen. Wie früher beim Bauklötz-Spielen als Kind, kann der Anwender seiner Fantasie freien Lauf lassen und die entspannende Wirkung von vielen bunten Voxeln genießen.

Während der Voxel Editor im Rahmen des Projekt B nur für Notebooks und Standrechner mit Drei-Tasten-Maus entwickelt werden soll, wird eine Anpassung für Mobilgeräte zum schnellen "Voxeln" unterwegs als zielgruppenrelevant erachtet und ist für spätere Versionen geplant. Die Grundsteine für eine Touch-Bedienung - ein aufgeräumtes Interface, große Bedienelemente - sollen jedoch schon bei dieser Version gelegt und bei der Gestaltung berücksichtigt werden.

Nicht Teil des Projekts sind eine Cloud- und Community-Anbindung, die später das Speichern, Laden und Verwalten von Modellen ermöglichen wird.

# Konzeption

## Bedienoberfläche



### *Oberfläche des Voxel Editors beim ersten Start*

Die Bedienoberfläche des Voxel Editors besteht aus der Arbeitsfläche in der Mitte, der Projektleiste darüber sowie der Werkzeugleiste und der Farbpalette am rechten Rand.

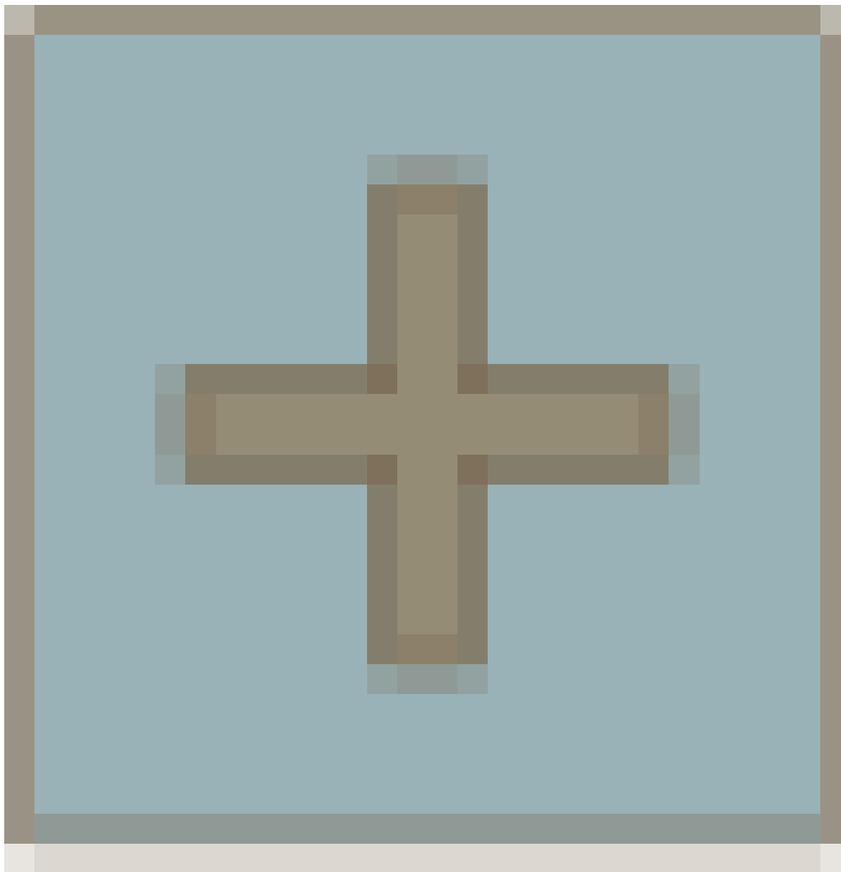
Die Arbeitsfläche dient als Fundament für die Voxel-Modelle, auf ihr werden die einzelnen Voxel angeordnet und bearbeitet.

Die Projektleiste bietet die Verwaltung von Arbeitsschritten mit den Möglichkeiten einzelne Schritte zu widerrufen, wiederherzustellen oder auch alle Schritte zu löschen. Außerdem beherbergt sie die Export-Funktion.

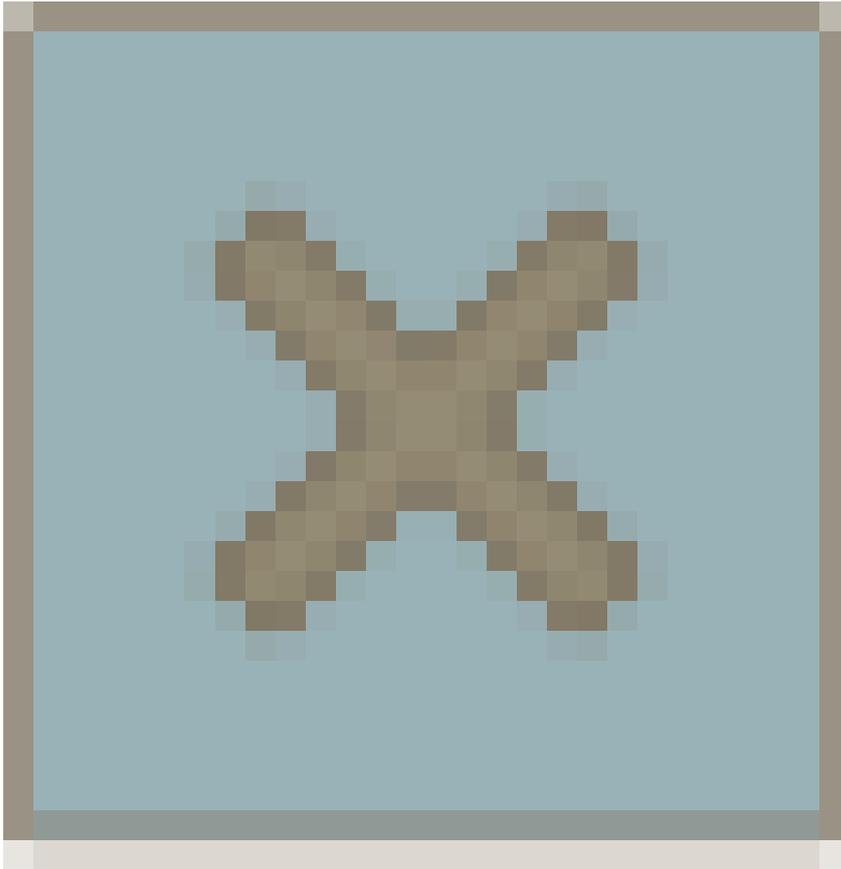
Die Werkzeugleiste stellt alle Werkzeuge zum Bearbeiten des Voxel-Modells zur Verfügung. Der Fokus wurde auf einige wenige ausgewählte Werkzeuge gelegt, um den Editor nicht zu überfrachten.

Es gibt vier Basiswerkzeuge:

- Voxel erzeugen



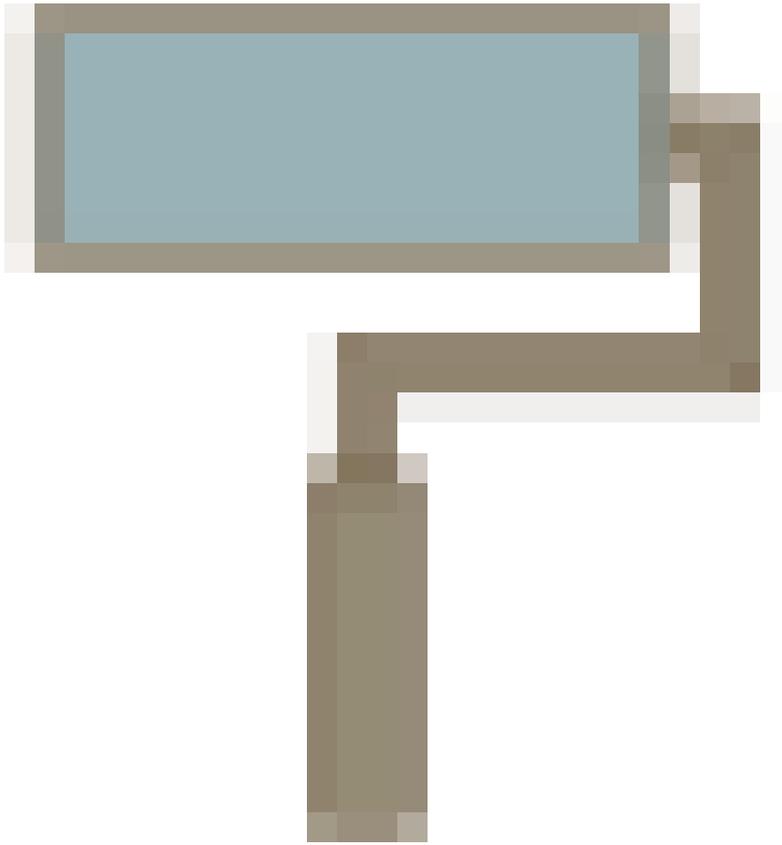
- Voxel löschen



- Voxelfarbe aufnehmen

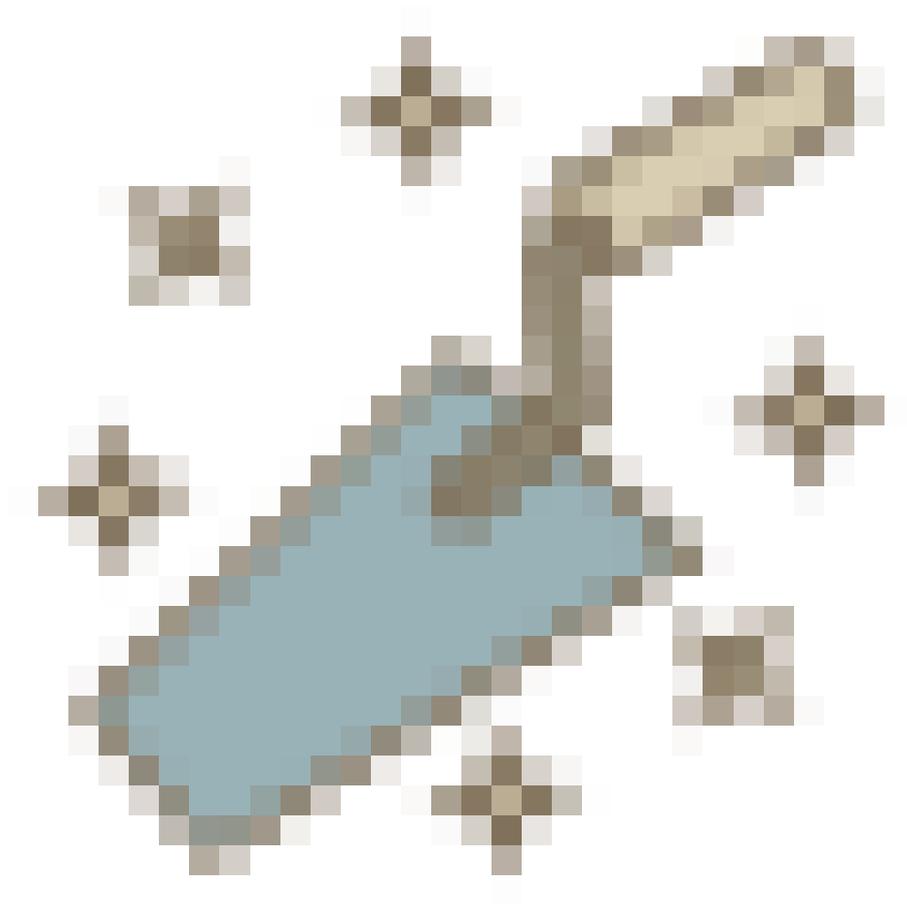


- Voxel neu einfärben

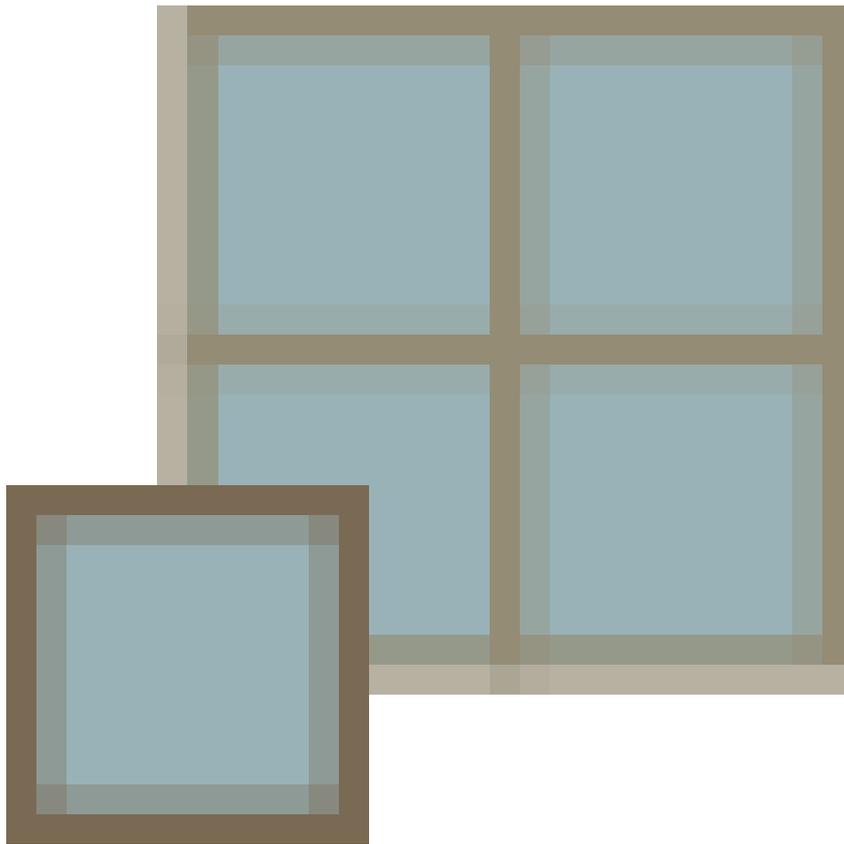


Zusätzlich gibt es Spezialwerkzeuge, die dem Anwender die Arbeit an komplexeren Würfelmodellen erleichtern und viel Zeit sparen helfen:

- Voxel-Block erzeugen



- Modell vergrößern



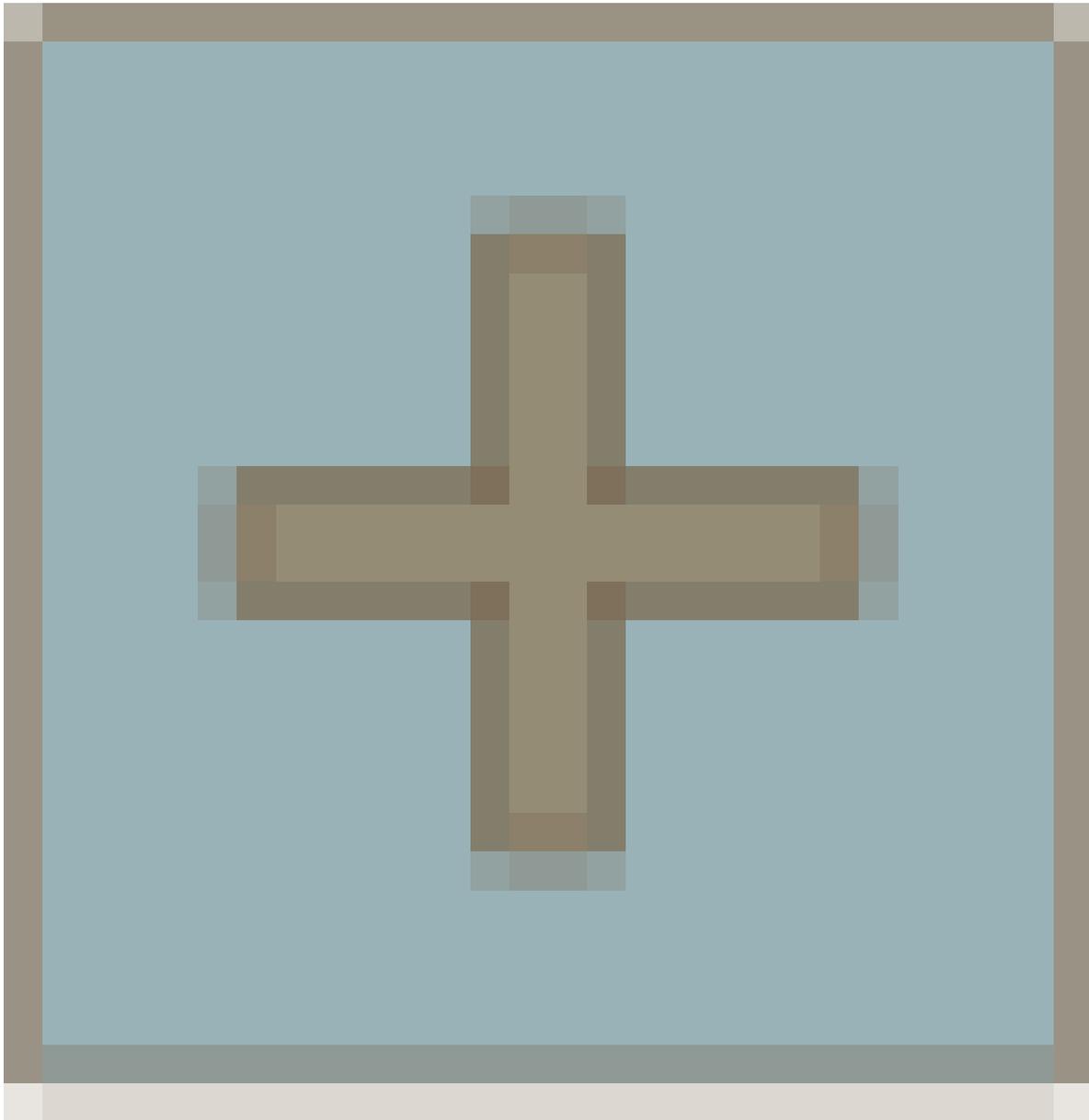
Ein Klick auf ein Werkzeug-Icon aktiviert den entsprechenden Bearbeitungsmodus. Dieser ist solange aktiv, bis ein anderes Werkzeug aktiviert wird. Eine Ausnahme bildet die Funktion "Voxelfarbe aufnehmen".

## Voxel erstellen und löschen

Das von Spielen wie Minecraft oder dem Spielzeug-Klassiker Lego inspirierte System des Voxel-Editors gibt vor, ausschließlich aus Würfeln (Voxel) bestehende Modelle zu konstruieren. Die Voxel werden auf einem vorgegebenen Raster angeordnet. Um die Bedienung zu vereinfachen, lassen sich zunächst keine schwebenden Voxel erzeugen. Stattdessen werden sie an den Flächen anderer Voxel angedockt.

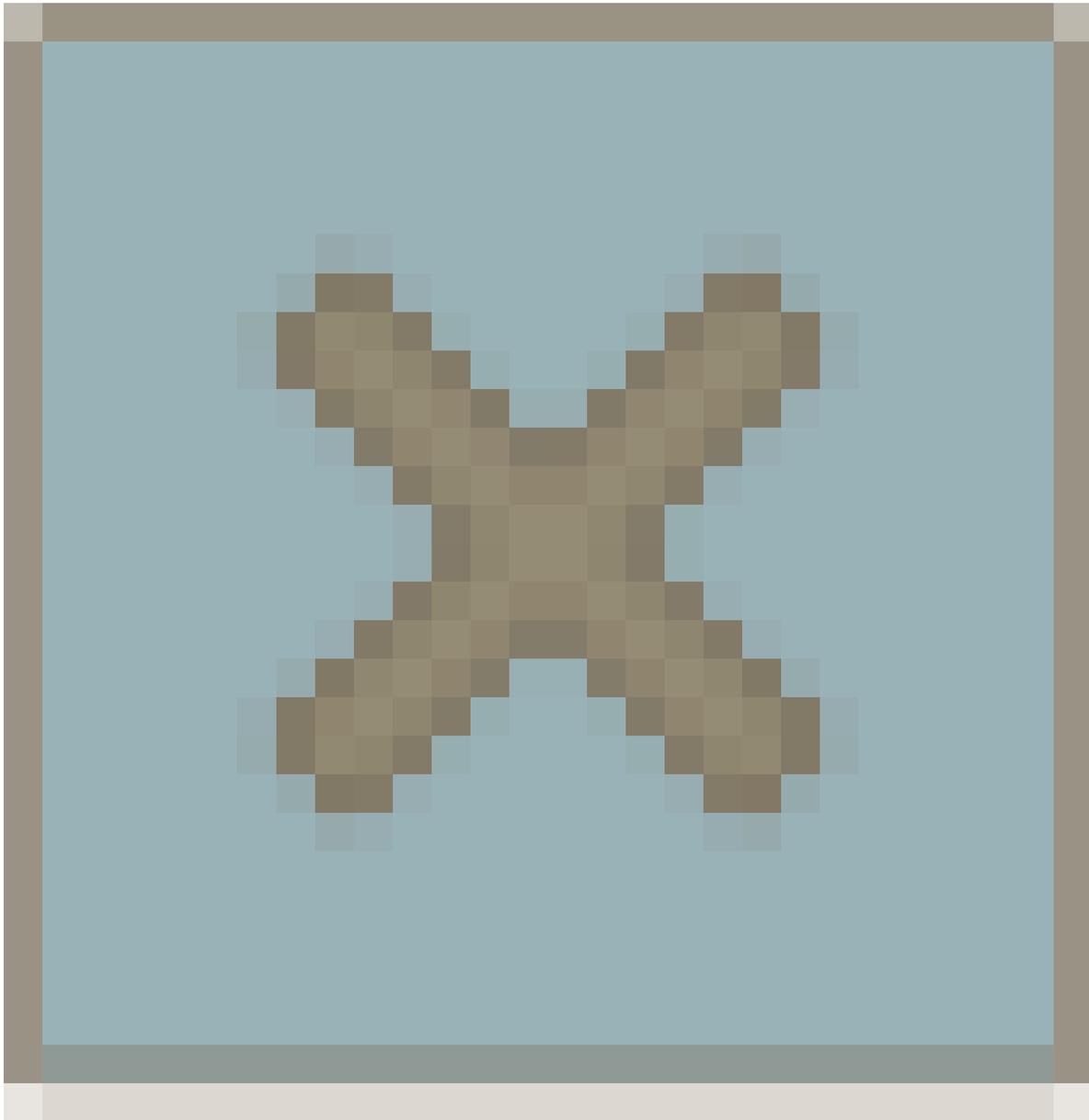
Das Anvisieren der gewünschten Fläche erleichtert eine Vorschau-Funktion, die die spätere Position eines neuen Voxels zunächst durch ein halbtransparentes Voxel anzeigt. Dies geschieht in Echtzeit, während die Maus über gültige Positionen bewegt wird.

Mit einem Linksklick materialisiert sich das Voxel an der ausgewählten Stelle. Dazu muss der "Voxel erzeugen"-Modus



aktiviert sein. Beim Starten eines neuen Projekts ist dies standardmäßig der Fall.

Sollte das Ergebnis doch nicht zufriedenstellend sein, lässt sich das Voxel jederzeit mit dem "Voxel löschen"-Tool



wieder entfernen. Diese Funktion ist entweder durch das entsprechende Icon in der Werkzeugleiste verfügbar oder auch jederzeit durch einen Rechtsklick anwendbar.

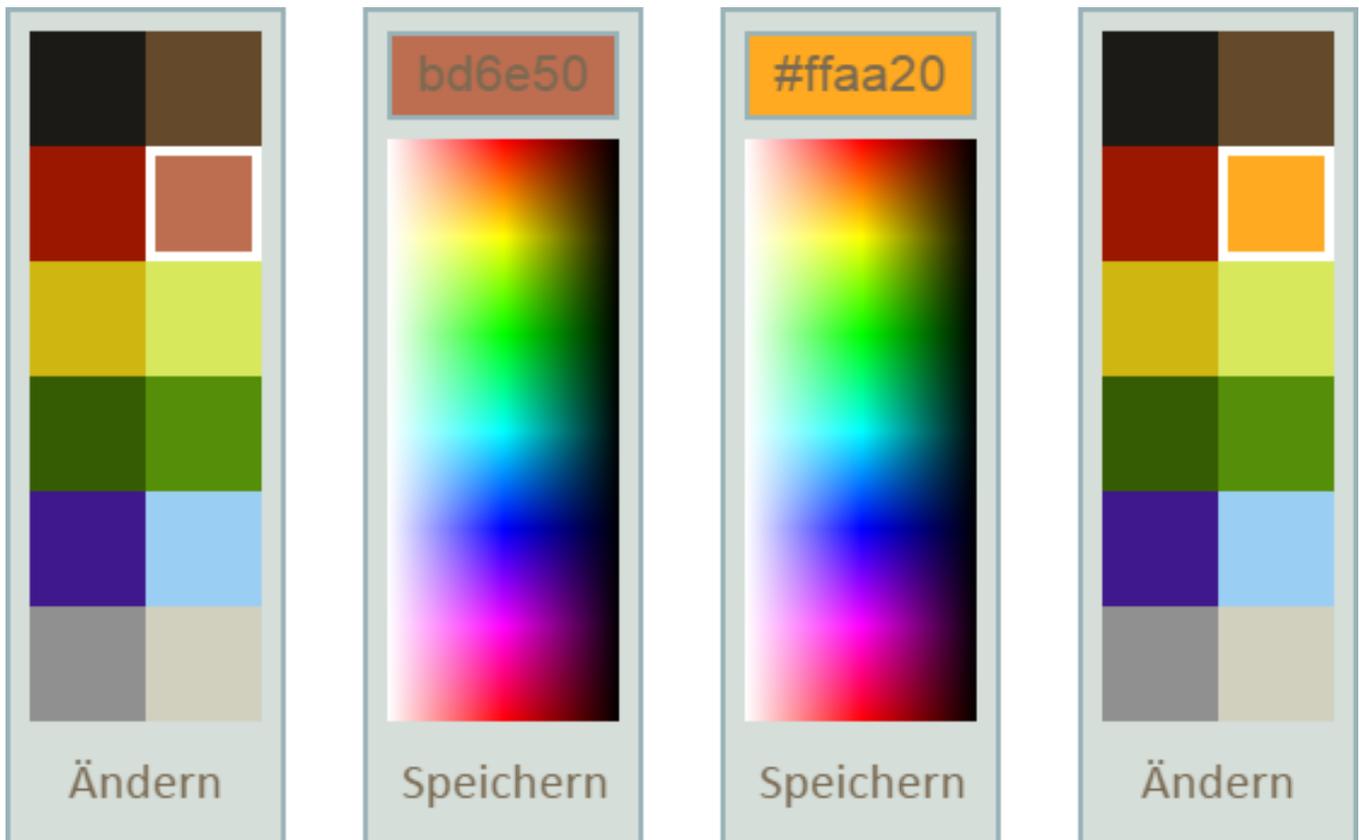
Zudem lassen sich mit dem Lösch-Werkzeug auch einzelne Voxel freistellen, sodass sie schweben. So kann die Einschränkung, nur angedockte Voxel zu erzeugen, umgangen werden.

## Farbwahl

Voxel lassen sich in jeder beliebigen Farbe erzeugen. Für den Anfang bietet eine Farbpalette eine Auswahl von 12 Farben, die nach den Anforderungen für die Terralea-Welt angemischt wurden.

Sollte eine Farbe fehlen, kann der Anwender eine nicht benötigte Farbe mit einer gewünschten neuen Farbe ersetzen.

Dazu wählt er eine Farbe aus – diese wird später ersetzt – und klickt auf "Ändern". Ein Farbwähler mit HSV-Farbraum wird angezeigt. Hier wählt der Anwender die gewünschte neue Farbe durch Ziehen mit gedrückter Linkstaste. Eine Hex-Wert-Eingabe ist ebenfalls möglich. Mit "Speichern" wird nun die vorherige Farbe mit der neu gewählten überschrieben und steht zur Verfügung.



#### ***Ändern einer Farbe in der Farbpalette***

Bereits im Modell verwendete Farben bleiben davon unangetastet, eine in der Farbpalette überschriebene Farbe bleibt also im Modell bestehen.

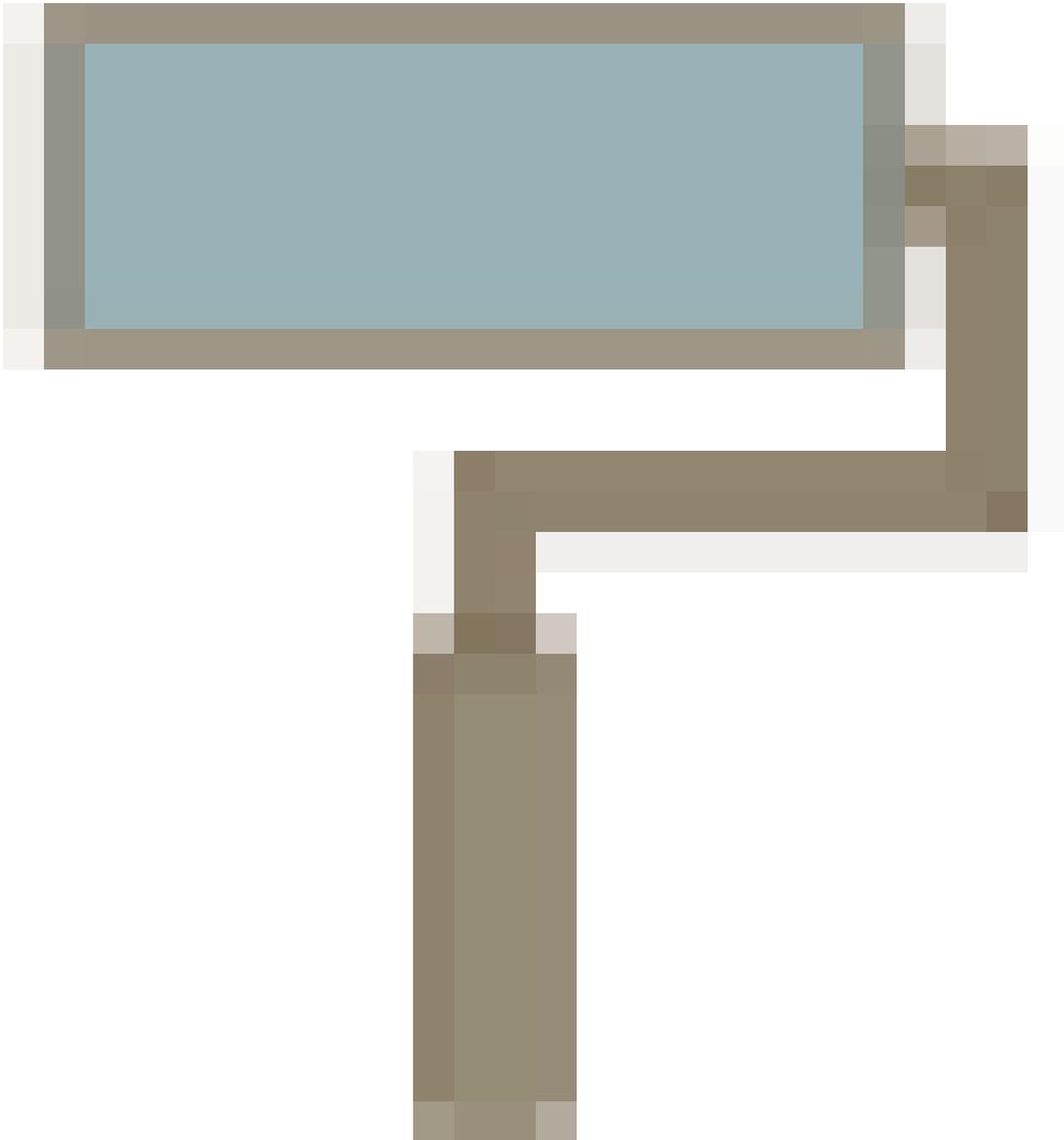
Falls eine solche Farbe dann doch noch benötigt wird, in der Farbpalette jedoch gelöscht wurde, ist dies ein Fall für das "Voxelfarbe aufnehmen"-Werkzeug



: Ist es aktiviert, so wird die Farbe angeklickter Voxel als aktuelle Farbe gesetzt.

Es erleichtert auch den Umgang mit einander sehr ähnlichen Farbtönen in der Farbpalette. Wenn zu einer Gruppe von Voxeln einer bestimmten Farbe weitere Voxel in derselben Farbe hinzugefügt werden, ist die richtige Farbe in der Farbpalette dann nur schwer ausfindig zu machen. Viel einfacher ist es mit dem "Voxelfarbe aufnehmen"-Werkzeug: mit ihm wird ein Voxel der Gruppe angeklickt, die Farbe wird für spätere Arbeiten gespeichert und anschließend wechselt der Editor automatisch in den Modus des zuletzt gewählten Werkzeugs. So kann die Arbeit direkt und mit der gewünschten Farbe fortgesetzt werden.

Um einem Voxel nachträglich eine andere Farbe zu geben, ist das Werkzeug "Voxel neu einfärben"



das Mittel der Wahl. Nach einem Klick auf das zugehörige Icon ist der Einfärbe-Modus eingeschaltet und angeklickten Voxeln wird derzeit aktive Farbe zugewiesen. Eine Vorschau-Funktion erleichtert das Anvisieren des gewünschten Voxels.

## Spezial-Werkzeuge

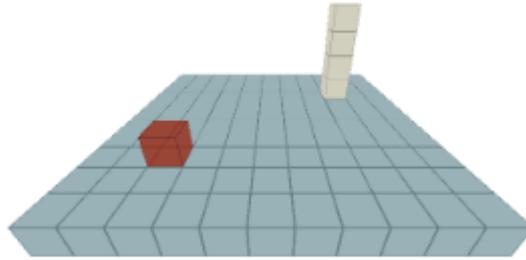
### Voxel-Block erzeugen

Das Voxel-Block-Werkzeug

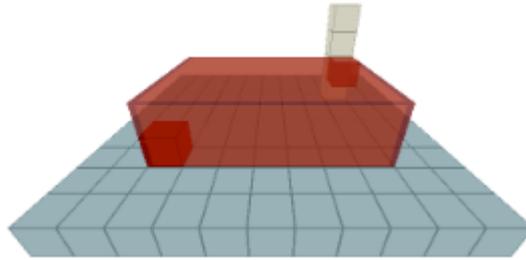


dient dem automatischen Erstellen von großen Voxel-Verbänden. Dazu wird wie folgt vorgegangen:

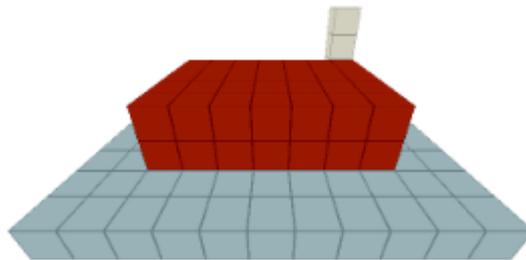
1. Durch den ersten Klick wird das Start-Voxel des Verbunds gesetzt – eine Referenz für die Position des finalen Verbunds



2. Anschließend wird die Größe definiert, indem der Anwender die Position des am weitesten vom Start-Voxel entfernten Voxels festlegt. Bereits beim Anvisieren geeigneter Positionen wird jeweils die Fläche, die der resultierende Voxel-Verbund einnehmen würde, visualisiert:
- a. Liegt das letzte Voxel auf einer Achse mit dem Start-Voxel ist das Resultat eine Voxel-Reihe.
  - b. Liegen beide Voxel auf derselben Höhe, jedoch diagonal entfernt, so ergibt dies eine Fläche von Voxeln.
  - c. Ist die Endposition höher und diagonal entfernt von der Startposition, so wird ein Voxel-Block erzeugt.

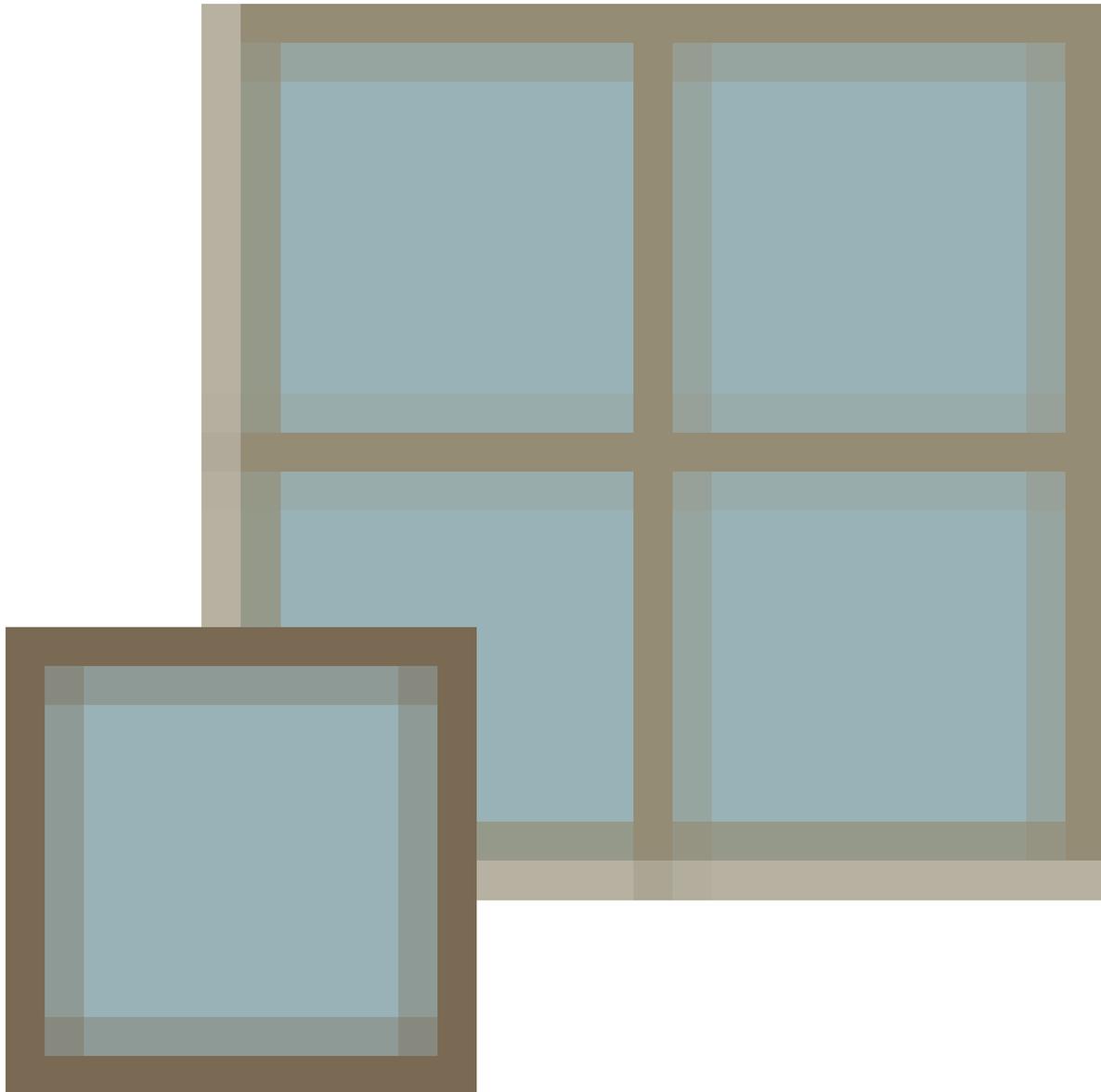


3. Ein Klick füllt schließlich die gezeigte Fläche mit einzelnen Voxeln auf und erspart damit mühevollen Handarbeit.



**Modell vergrößern**

Stellt der Anwender während der Arbeit am Modell fest, dass er einen anderen Maßstab benötigt, um beispielsweise zusätzliche Details einzufügen, kann das Modell mit einem Klick auf das "Modell vergrößern"-Werkzeug



auf die doppelte Größe skaliert werden. Ein Voxel entspricht also anschließend einem Würfel aus vier Voxeln.

## Kamera-Bewegung

Um auch jede gewünschte Voxel-Position erreichen und anvisieren zu können, lässt sich die virtuelle Kamera beliebig ausrichten: Durch Ziehen mit gedrückter linker Maustaste rotiert die Kamera, während das Ziehen mit gedrückter rechter Maustaste sie horizontal oder vertikal verschiebt. Mit dem Mausexplorer lässt sich schließlich noch der gewünschte Abstand zum Modell einstellen (Zoom).

## Arbeitsschritte verwalten

Viele Editoren bieten die klassischen Funktionen "Rückgängig" und "Wiederherstellen", die jeweils nur einen Arbeitsschritt betreffen. Wie z.B. beim Protokoll in Photoshop oder beim Journal in GIMP, sollen Nutzer beim Voxel Editor aber die Möglichkeiten haben, mit nur einem Klick gleich mehrere Arbeitsschritte zurückzusetzen bzw. wiederherzustellen.

Dazu gibt es das Bedienfeld "Speicherstände" in der Projektleiste, das die Resultate von bis zu 20 Arbeitsschritten zur Zeit aufnehmen kann. Jedes kleine Quadrat des Bedienfelds stellt einen Speicherplatz dar.



### **Projektleiste mit Speicherstände-Bedienfeld**

Jeder vom Nutzer durchgeführte Arbeitsschritt sorgt dafür, dass ein neuer Speicherstand angelegt und in Form eines farbig gefüllten Quadrates repräsentiert wird. Als Arbeitsschritte gelten: ein Voxel wurde gesetzt oder gelöscht, ein Voxel-Block wurde erzeugt oder das Modell wurde einmal vergrößert.

Der derzeit aktive Speicherstand wird als braun gefülltes Quadrat hervorgehoben. Alle derzeit inaktiven Speicherstände sind blau gefüllte Quadrate. Wenn der Nutzer nun auf einen inaktiven Speicherstand klickt, so wird dieser zum neuen aktiven Speicherstand.

Dadurch werden alle Arbeitsschritte, die zeitlich nach dem ausgewählten Speicherstand durchgeführt wurden, rückgängig gemacht. Alle Arbeitsschritte, die im Vorfeld durchgeführt wurden, werden ggf. wiederhergestellt.

Achtung: Sobald der Nutzer einen neuen Arbeitsschritt durchführt, obwohl noch inaktive Speicherstände auf den gerade aktiven folgen, werden diese überschrieben, also unwiderruflich gelöscht. Weiterhin wird der jeweils älteste Speicherstand gelöscht, wenn alle Speicherplätze belegt sind und ein neuer Arbeitsschritt gespeichert werden muss.

Um alle Arbeitsschritte zu löschen und damit das Projekt zurückzusetzen, genügt ein Klick auf das Mülleimer-Icon. Der gesamte Projektfortschritt geht damit unwiederbringlich verloren.

## **Workflows**

Der Voxel Editor unterstützt unterschiedliche Arbeitsweisen, die der Anwender beliebig kombinieren und so ein individuelles Vorgehen bei der Konstruktion von Voxel-Modellen entwickeln kann.

Eine naheliegende Herangehensweise ähnelt dem Bau eines Lego-Gebäudes: Der Anwender setzt zu Beginn die Voxel für den untersten Abschnitt des Modells auf die leere Arbeitsebene. Darauf werden wieder neue Voxel platziert. So arbeitet sich der Anwender nach und nach hoch, bis auch die höchsten Ebenen des Modells erreicht sind. Die gewünschte Farbe der einzelnen Voxel kann entweder schon gleich beim Setzen ausgewählt werden oder die Voxel werden nachträglich koloriert (Einfärben-Werkzeug).

Ein umgekehrtes Vorgehen ist die Bildhauer-Technik: Der Benutzer erstellt zunächst mit dem Voxel-Block-Werkzeug einen Block aus Voxel - so groß, dass das künftige Modell vollständig von ihm umschlossen wäre. Nun wird damit begonnen, einzelne Voxel mithilfe des Löschen-Werkzeugs "herauszumeißeln". Welcher Abschnitt wann bearbeitet wird, bleibt dem Benutzer überlassen. Nach Fertigstellen des Modells oder auch bereits zwischendurch können die Voxel individuell koloriert werden.

## **Export**

Damit der Benutzer sein Voxel-Werk auch außerhalb des Voxel Editors verwenden oder weiterverarbeiten kann, z.B. als Assets für ein eigenes Spiel, gibt es eine Export-Funktion



. Sie generiert eine einfache Textdatei, die alle relevanten Informationen des Modells enthält, und die sich von anderen 3D-Tools auslesen lässt. Für Unity gibt es bereits ein Plugin, welches den Import übernimmt.

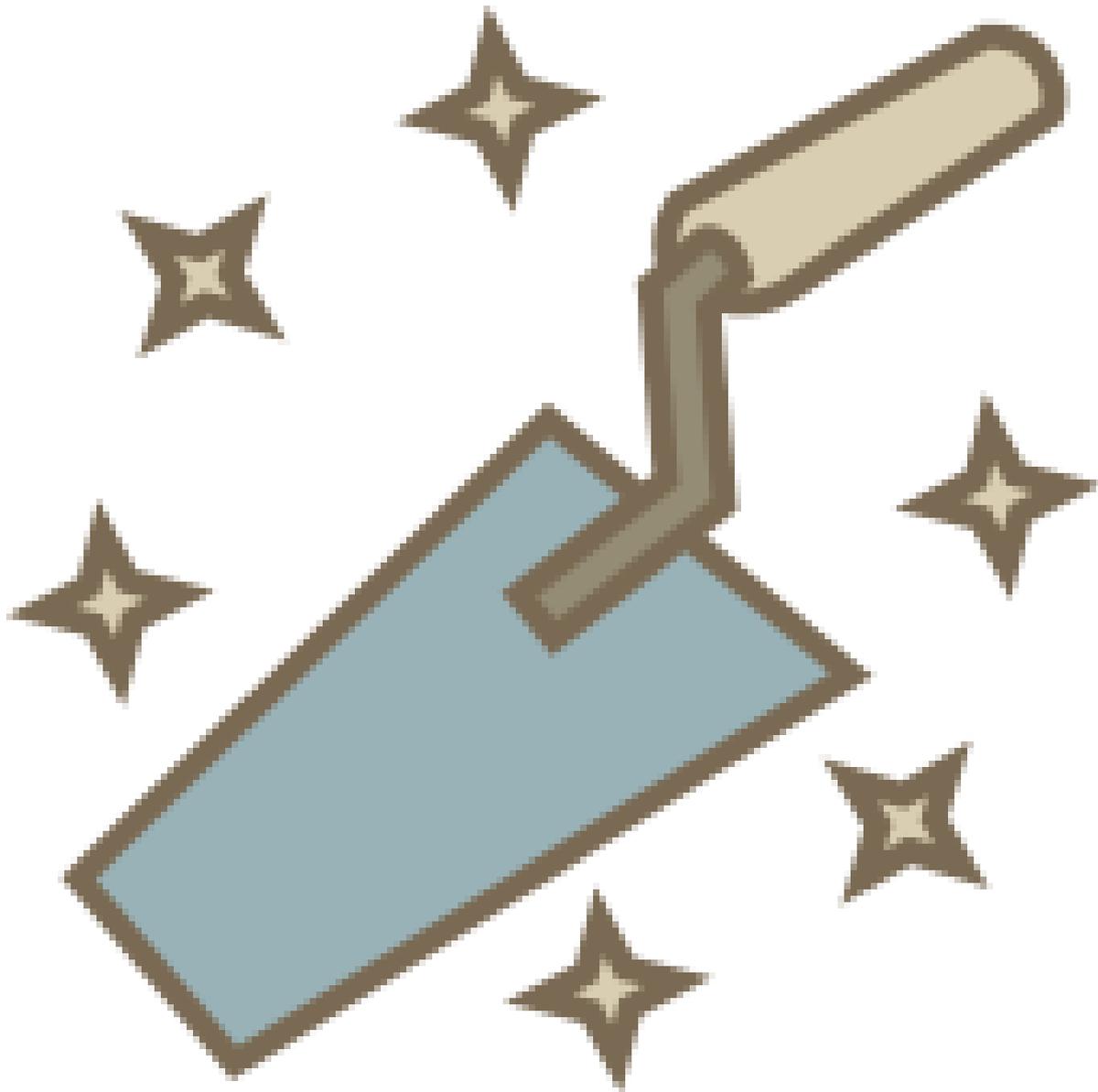
Eine Anleitung dazu und weitere Informationen finden sich im offiziellen Github-Repository.

# Gestaltung

Zentrum und Schauplatz des Geschehens ist die Arbeitsfläche des Voxel Editors. Layout und Design sollen diese Devise unterstützen. Alle weiteren Bedienelemente ordnen sich um die Arbeitsfläche herum an. Sie sind flach gehalten und zeichnen sich durch den Verzicht auf Schatten und Reflexionen aus. Die Farbgebung übt Zurückhaltung durch eine geringe Anzahl matter Blau-, Grau- und Brauntöne.

Um auch gleichfarbige Voxel gut differenzieren zu können, werden die Kanten jedes Voxels mit einer Kontur nachgezeichnet. Der Farbwert der Kontur ist dabei eine abgedunkelte Version der Flächenfarbe. Nur bei sehr dunklen Voxeln wird die Kontur aufgehehlt.

Dieser Stil wurde bei den meisten Bedienelementen aufgegriffen. Alle Icons folgen z.B. der Regel: Jede Fläche muss mit einer Kontur von mindestens 1 Pixel Dicke umgeben sein. Dies sorgt für ein Mockup-ähnliches Erscheinungsbild, welches im Einklang mit den Voxel Editor-Einsatzgebieten "Schnelles Visualisieren" und "Prototyping" steht.



### Icon des Voxel-Block-Tools

Eine weitere Richtlinie der Gestaltung war es, möglichst wenig Text einzusetzen. Dadurch soll eine intuitive und eher spielerische Bedienung erreicht werden. Ziel war es, dass ein Nutzer eher durch Ausprobieren, als durch Nachlesen, die Funktion eines Werkzeuges herausfindet. Ein Beispiel dafür ist das Voxel-Block-Werkzeug. Das zugehörige Icon mit der "magischen Maurerkelle" lädt den Nutzer dazu ein, zu ergründen, was sich hinter diesem Werkzeug verbirgt und wie man es einsetzt. Weder die Tooltips noch ein Tutorial oder eine Anleitung geben ihm darüber Auskunft. Allein das Ausprobieren mit Unterstützung der Vorschau-Funktion sollen ihn die Funktionsweise begreifen lassen.

## Entwicklung

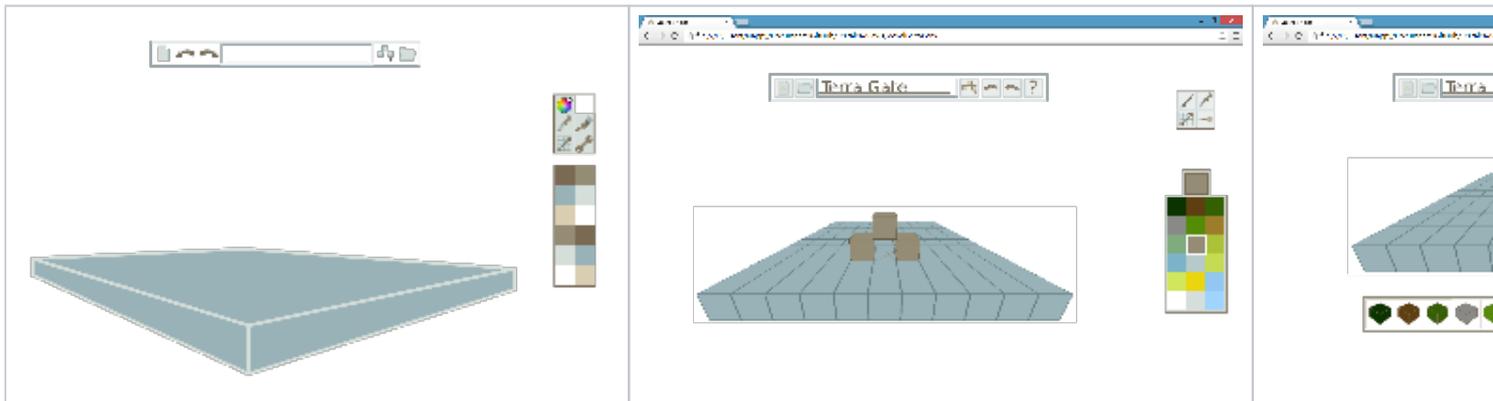
Die Entwicklung des Voxel Editors für das Projekt B nahm ca. 275 Stunden in Anspruch.

Das Team bestand aus:

- Lars Krafft: Programmierung, Konzeption
- Finn Schröder: Konzeption, Gestaltung, Management

Als Basis wurde ein bereits voll funktionsfähiger Prototyp entwickelt, dessen Bedienung und Oberfläche anschließend kontinuierlich überarbeitet wurde. Grundlage dafür waren zahlreiche Tests mit Personen unterschiedlicher Vorkenntnisse – vom Laien bis zum erfahrenen CAD-Nutzer.

Der Voxel Editor nutzt HTML5 und CSS3 für die Bedienelemente und das Layout. Die Programmlogik ist in Javascript implementiert. Die 3D-Darstellung der Arbeitsebene und Voxel übernimmt die Grafikkbibliothek WebGL, als Programmierschnittstelle wird Three.js verwendet. Die verwendeten Web-Techniken werden derzeit von allen aktuellen Browsern unterstützt – jedoch wurde das beste Nutzererlebnis unter Google Chrome erzielt.



### Interface-Evolution

Die Icons sowie im Vorfeld entstandene Mockups sind mit Inkscape angefertigt worden.

Als Plattform zur Kollaboration und Veröffentlichung dient Github. Im offiziellen Voxel Editor-Repository wird es auch zukünftig Updates und neue Funktionen geben. Zur Mitarbeit sind alle herzlich eingeladen!

## Inspiration

- CWME – Cube World Model Editor
- LEGO und der LEGO Digital Designer
- Minecraft
- Ralf Hebeckers Ideen für eine innovative Undo/Redo-Verwaltung

