

Kitschige Winterlandschaft

Eine Facharbeit in NwT von Tobias Russ, Janis Preikschas und Dominik Shrestha, Klasse 10b

Version 2.9 final 2 vom 05.03.2020



Inhalt

Aufgabe.....	1
Vorgehen	1
Die einzelnen Elemente.....	2
Kegel-Tannenbaum mit Stern.....	3
Der Vieleck-Tannenbaum	4
Der Zaun	5
Eisenbahnbrücke mit Fluss und Gleis	6
Der Testdruck	8
Grundplatte mit Haus.....	9
Das zusammengesetzte Modell.....	9
Die technische Zeichnung.....	9
Die LED-Arduino-Schaltung	12
Links.....	14

Aufgabe

Unsere Aufgabe war es, eine „kitschige Winterlandschaft“ zu erstellen. Es sollte mit dem Online-CAD¹-Programm Tinkercad ein 3D-Modell einer Winterlandschaft mit blinkenden LEDs, gesteuert durch eine Arduino-Schaltung geplant und gebaut werden. Aufgrund fehlender Zeit wurde auf den praktischen Aufbau mit Druck und Verlötung verzichtet.

Vorgehen

Wir haben geplant, das 3D-Modell der Winterlandschaft aus verschiedenen Elementen, die auf eine Grundplatte gebaut werden, zusammenzusetzen. Diese Modelle wurden unabhängig voneinander erstellt und dann in einem Modell zusammengefügt.

Wir haben für die Erstellung die Web-Programme BlocksCAD², Tinkercad „Codeblöcke“ und Tinkercad „3D-Entwürfe“³ genutzt. Mit den ersten beiden kann man einen visuellen Block-

Unter „**CAD**“ (computer-aided design) versteht man das technisch-geometrische Konstruieren von Objekten mit Hilfe des Computers. Anfangs in der Fertigungstechnik, dann auch zur Simulation und bei der Entwicklung und der Konstruktion eingesetzt, findet sich die CAx-Technologie heute in verschiedenen Bereichen der Industrie zur Steigerung der Produktivität und Genauigkeit.¹

¹ Quelle: S. Vajna, C. Weber, H. Bley, K. Zeman: „CAx für Ingenieure. Eine praxisbezogene Einführung.“, Springer-Verlag, 2. Auflage, 2009

² <https://www.blockscad3d.com/editor/>

³ <https://www.tinkercad.com/>

Programmcode erstellen, rendern lassen und als Datei exportieren. Mit BlocksCAD kann man sogar den gesamten Stand eines Entwurfes als XML-Datei speichern und damit wieder bearbeiten. Mit Tinkercad „3D-Entwürfe“ kann man Körper direkt nach dem WYSIWYG-Prinzip visuell modellieren und ebenfalls als Datei exportieren. Tinkercad ist ein kostenloses einfaches 3D-CAD-Programm der Firma Autodesk.⁴ Außerdem haben wir zum Transport und einer plattformunabhängigen Darstellung der Dateien den Onlinedienst GitHub verwendet.

Zur Erstellung der Vorschaubilder und der technischen Zeichnungen wurden die Programme Print 3D von Microsoft, die Open-Source Software OpenSCAD und das kommerzielle, in einer Testversion kostenlose, Fusion 360, ebenfalls von Autodesk, verwendet.

Die einzelnen Elemente

- Kegel-Tannenbaum mit Stern
- Vieleck-Tannenbaum
- Zaun
- Eisenbahnbrücke mit Fluss und Gleis
- Grundplatte mit Haus
- Testdruck

Nachfolgend werden die einzelnen Elemente und ihre Erstellung einzeln beschrieben:

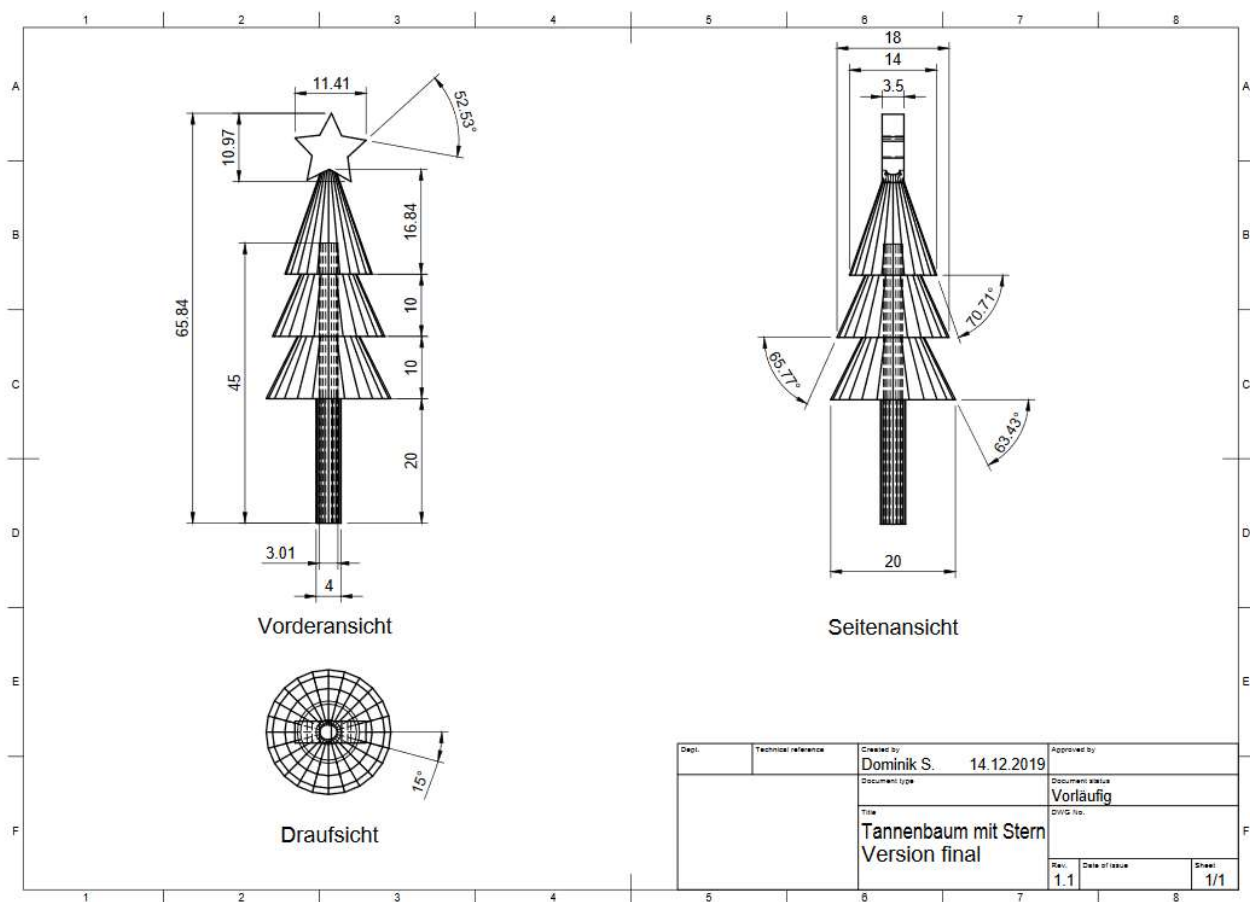
⁴ Quelle: Englische Wikipedia (<https://en.wikipedia.org/wiki/Tinkercad>)

Kegel-Tannenbaum mit Stern

Der Kegel-Tannenbaum (Arbeitsname "Brave Sango") besteht aus einem Zylinder-Stamm, drei aufeinandergesetzten Kegeln mit abnehmender Breite und zunehmendem Winkel und einem 5-zackigem, flachen, Stern. In der Mitte befindet sich ein ein zylinderförmiger Hohlraum für eventuelle Kabelverlegung, zur Verminderung des Gewichts und zum Einsparen von Druckmaterial.



1 Die Modulation des Kegel-Tannenbaums



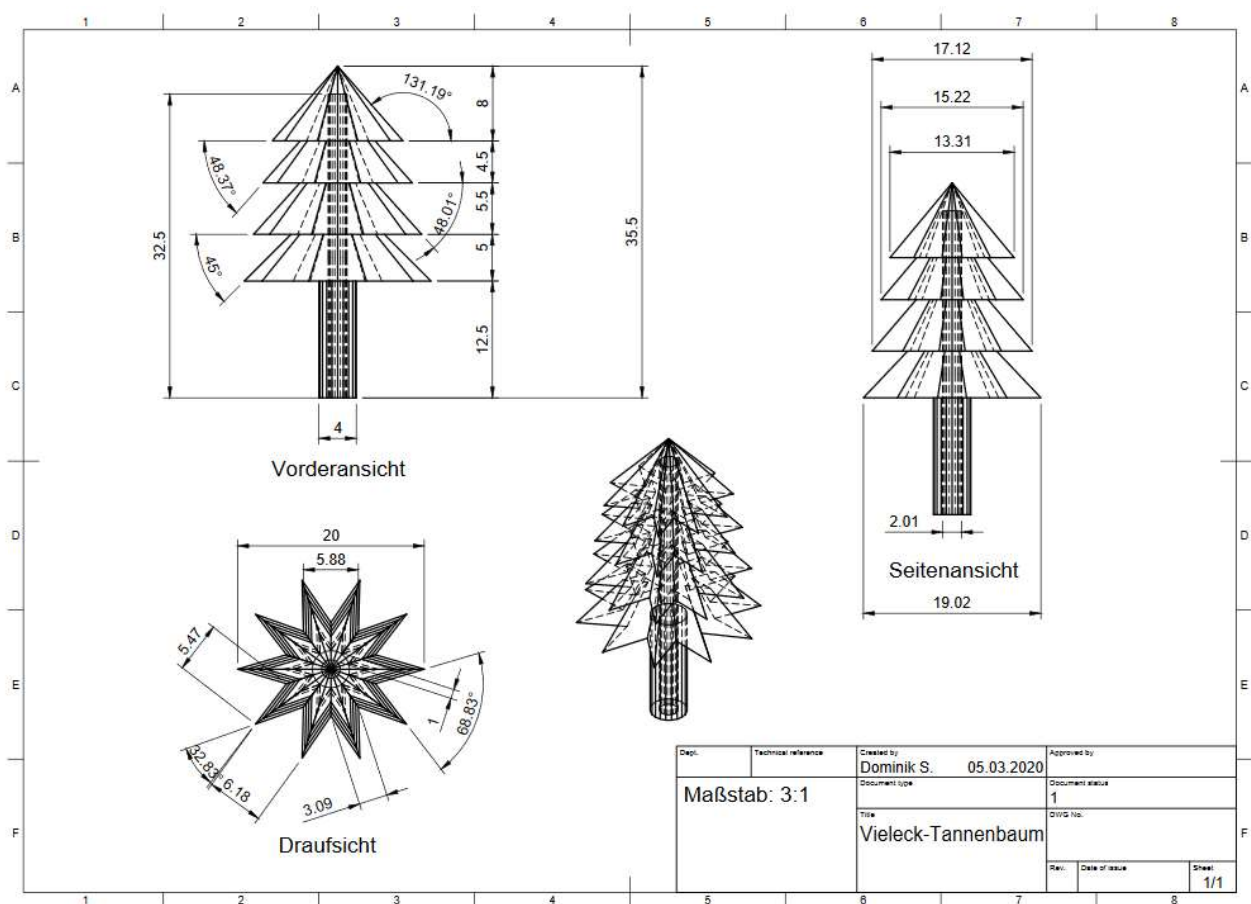
2 Die technische Zeichnung des Kegel-Tannenbaums

Der Vieleck-Tannenbaum

Die Vieleck-Tanne besteht aus einem Zylinderstamm und vier übereinandergesetzten 10-zackigen Sternen, die nach oben hin kleiner werden und abnehmende Winkel besitzen. In der Mitte befindet sich wie schon beim letzten Tannenbaum ein zylinderförmiger Hohlraum.

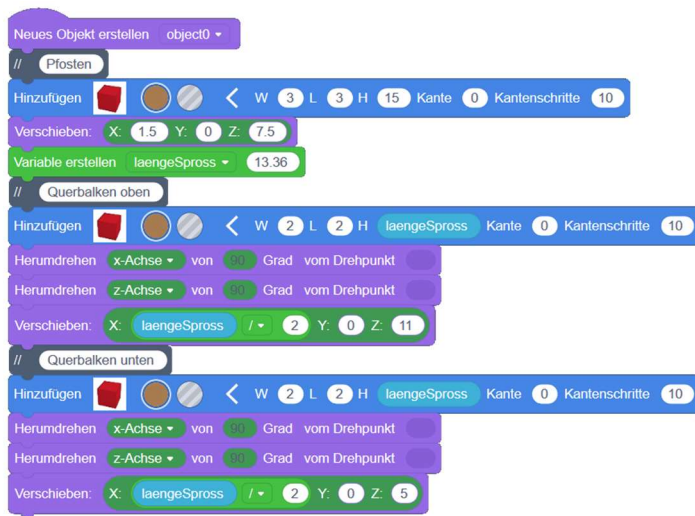


3 Die Modulation des Vieleck-Tannenbaums



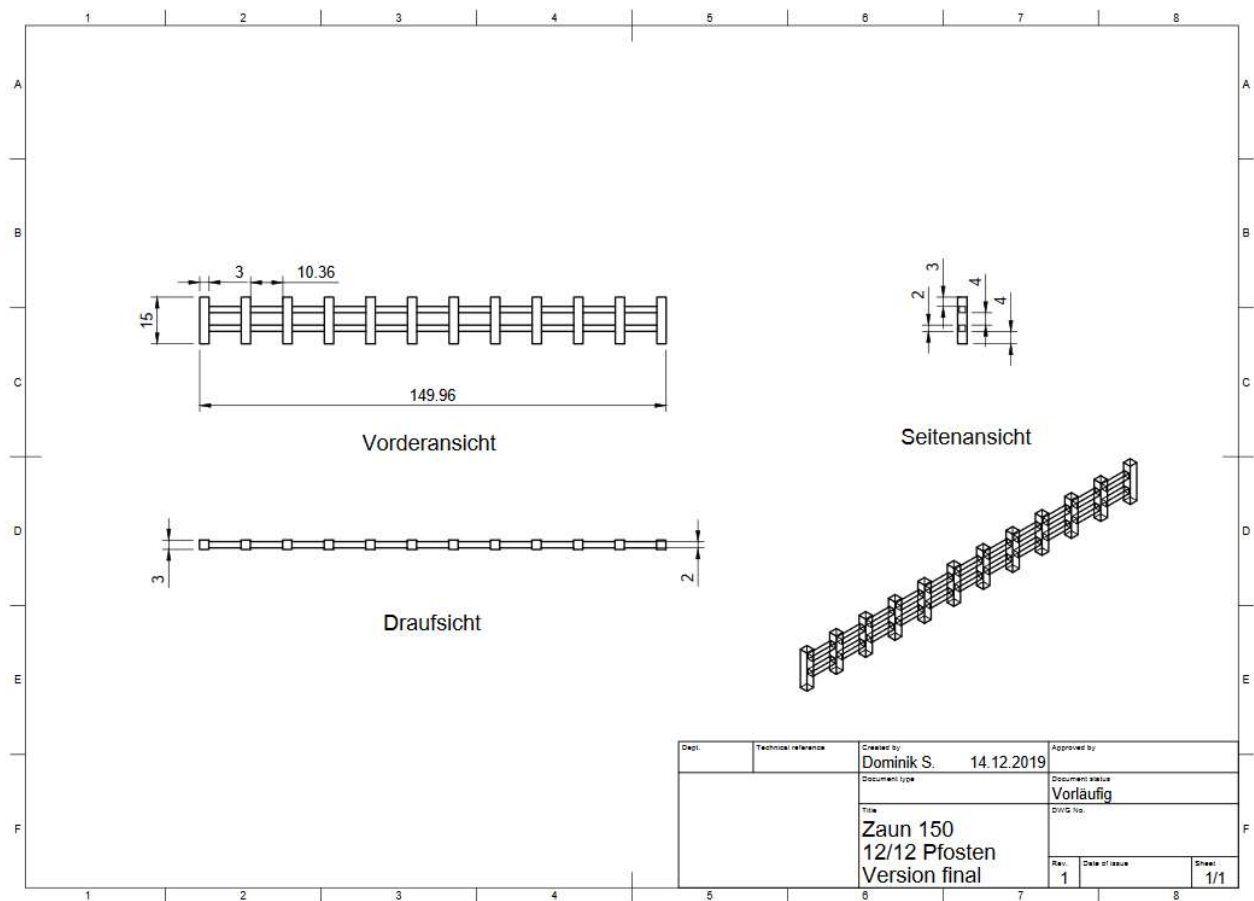
4 Die technische Zeichnung des Vieleck-Tannenbaums

Der Zaun



Der Zaun besteht aus mehreren Elementen von einem länglichen nach oben gerichteten Quader, der einen Pfosten bildet und zwei länglichen parallel quer dazu gerichteten Quadern, die die Sprossen bilden. Dieses Element wird 10 mal wiederholt und am Ende noch ein Schlusspfosten gesetzt. Die Maße wurden so gewählt, dass die gesamte Zaunreihe am Ende (etwa) 150 mm misst und damit genau auf die Grundplatte passt.

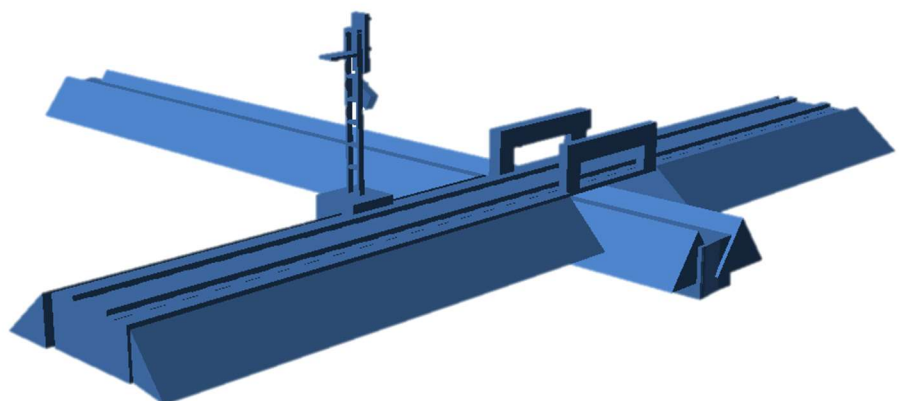
5 Der Block-Code des Zaunes in TinkerCAD

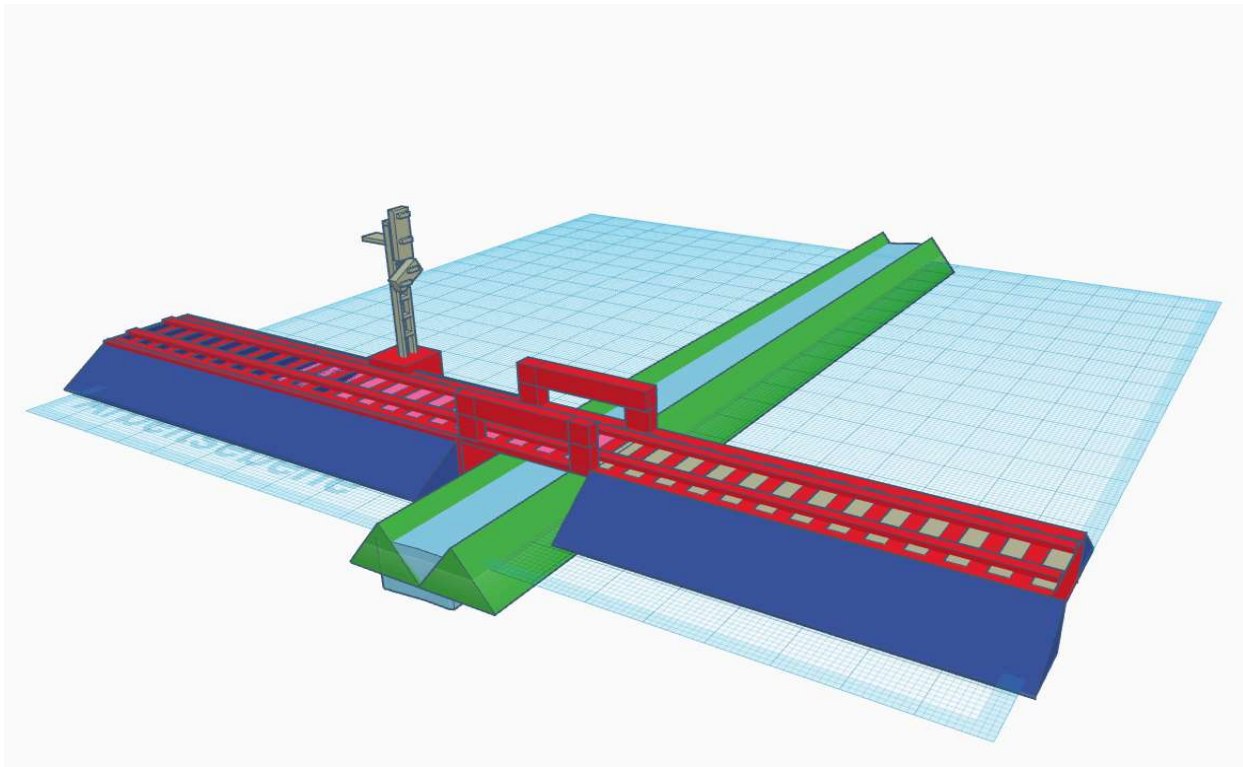


6 Die technische Zeichnung zu dem Zaun

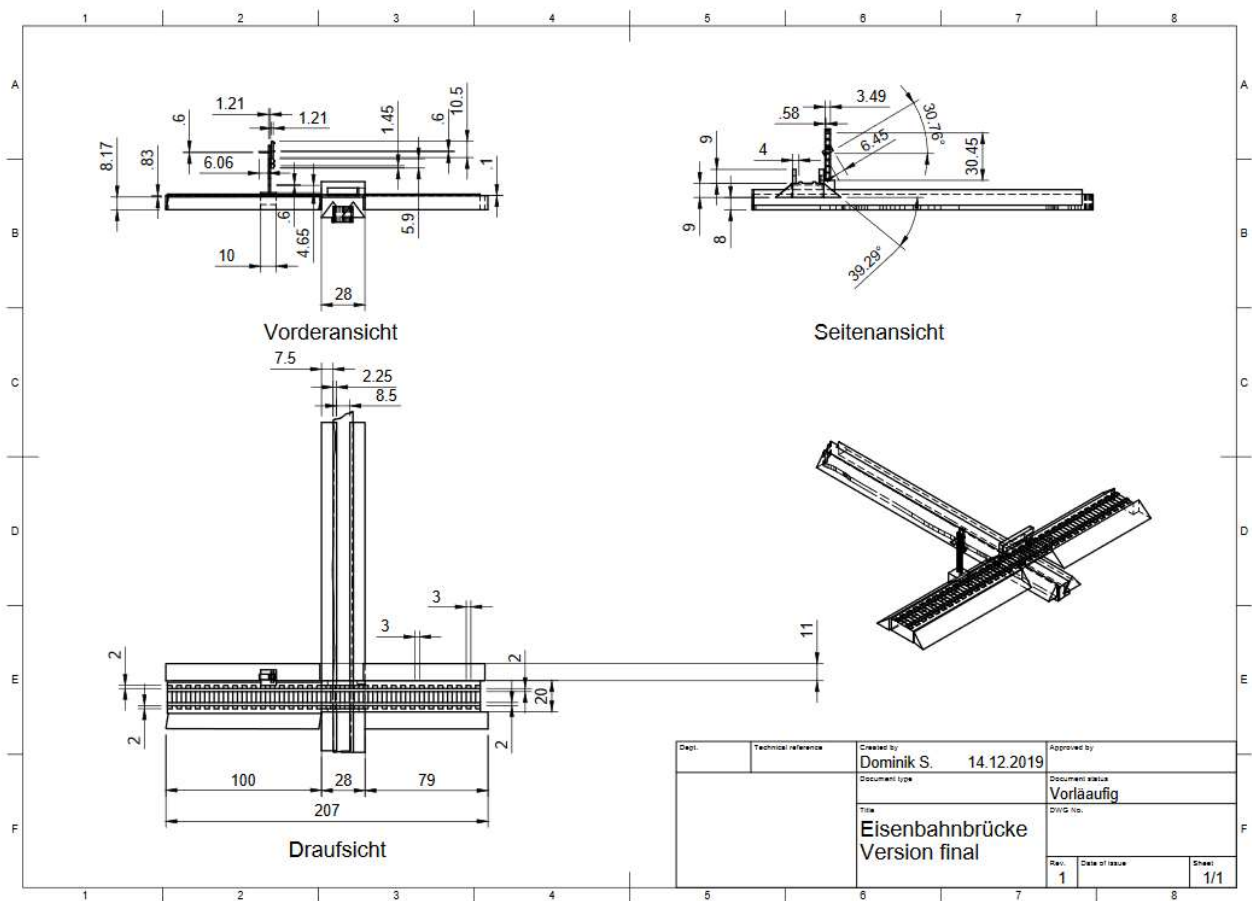
Eisenbahnbrücke mit Fluss und Gleis

Bei dem Aufbau der Eisenbahnbrücke wurden die Gleise aus Schwellen und Schienen zusammengesetzt. Die Schwellen wurden aus Quadern mit je 1 mm Länge zusammengesetzt. Die Schienen sind zwei weitere Quader, die in einem Winkel von 90° über die Schwellen gelegt wurden. Das Signal und die Brücke sind ebenfalls nur eine spezielle Anordnung von Quadern. Der Bahndamm besteht aus zwei Dreiecken sowie einem großen Quader in der Mitte. Das Flussbett sind ebenfalls zwei Dreiecke, die parallel angeordnet sind und in Quader dazwischen gelegt wurde.





7 Die Modulation der Eisenbahnbrücke



8 Die technische Zeichnung zu der Eisenbahnbrücke

Der Testdruck

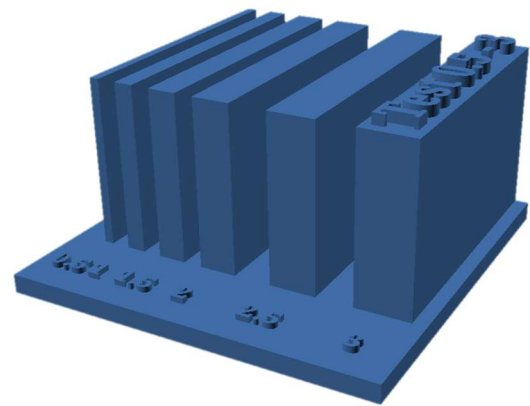
Wir hatten die Idee, mit dem 3D-Drucker etwas zu entwerfen, in das man eine LED hineinstecken kann und das an manchen Stellen eine so dünne Wand hat, dass das Licht der Diode hindurchscheint.

Das gleiche wollten wir auch möglicherweise bei den Fenstern des Hauses einsetzen und so gläserne Fensterscheiben simulieren.

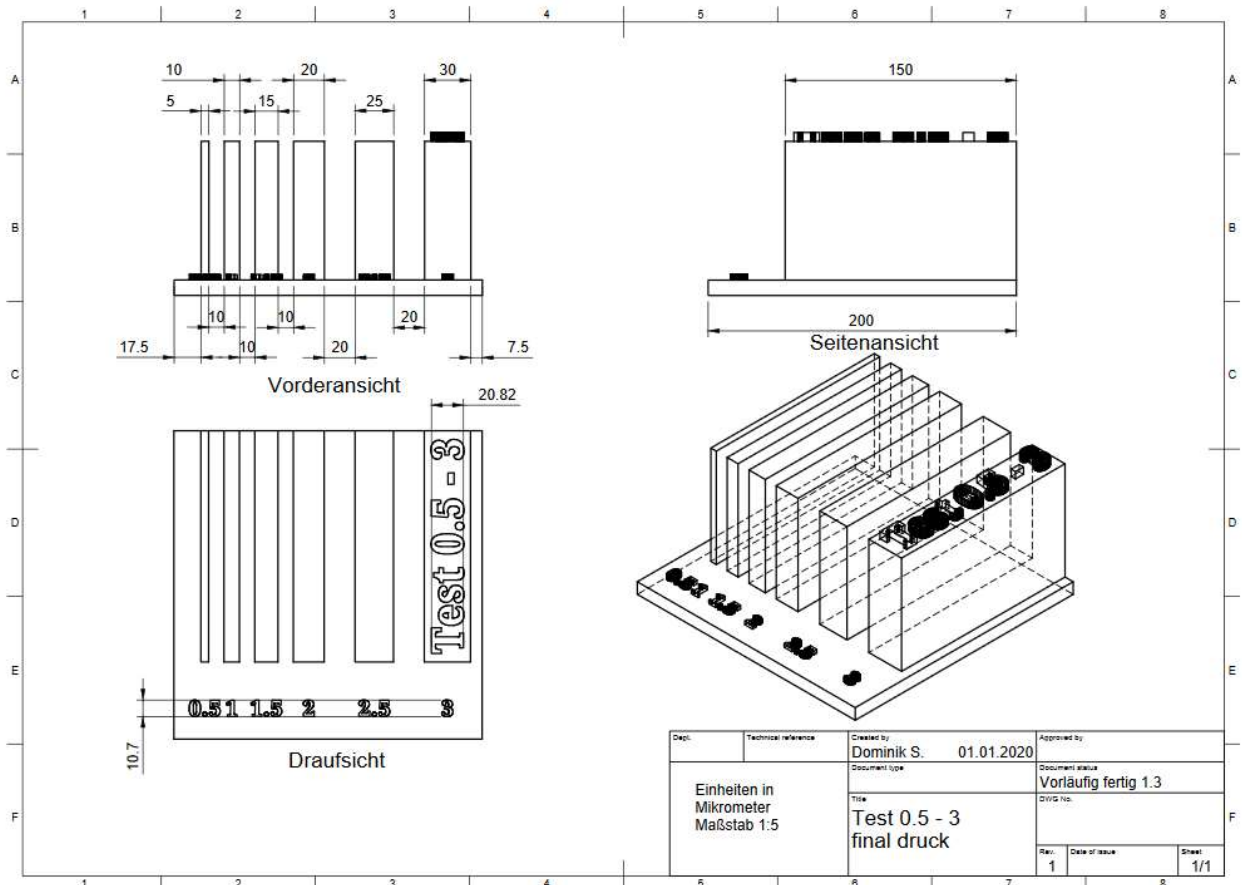
Allerdings hatten wir schon bei vorherigen Drucken festgestellt, dass der 3D-Drucker nur mit einer bestimmten Genauigkeit arbeitet und so besonders feine Modelle nicht ordentlich gedruckt wurden.

Um zu testen, wie dick eine Wand sein muss, damit sie hält und um zu testen, inwiefern der Kunststoff lichtdurchlässig ist, haben wir einen Testdruck entworfen. Dieser besteht aus verschiedenen Wänden mit unterschiedlicher Breite, die nebeneinander auf eine Platte gesetzt werden. Wir wählten für die Breite die Maße 0,5 mm; 1 mm; 1,5 mm; 2 mm; 2,5 mm und 3 mm.

Leider konnten wir aus Zeitgründen den Testdruck nicht mehr drucken, und damit die erhofften Erkenntnisse nicht in das Projekt einfließen lassen. Wir haben deswegen auf einen LED-Halter und Fensterscheiben verzichtet.



9 Die Modelation des Testdrucks



10 Die technische Zeichnung zum Testdruck

Grundplatte mit Haus

Aus Praktikabilitätsgründen haben wir die Grundplatte und das Haus in einem Element erstellt.

Die Grundplatte ist eine 15 mal 15 cm große, 0,5 cm dicke Platte, auf die alle anderen Elemente platziert werden. Sie bildet den Boden der Landschaft.

Darauf wurde das Haus gesetzt. Dieses besteht aus einem Quader, einen etwas kleineren Quader der diesen aushöhlt und so für Wände sorgt und drei weiteren aushöhlenden Quadern, die für Fenster sorgen. Außerdem wurden eine Tür, ein Türknauf und ein Dach hinzugefügt.

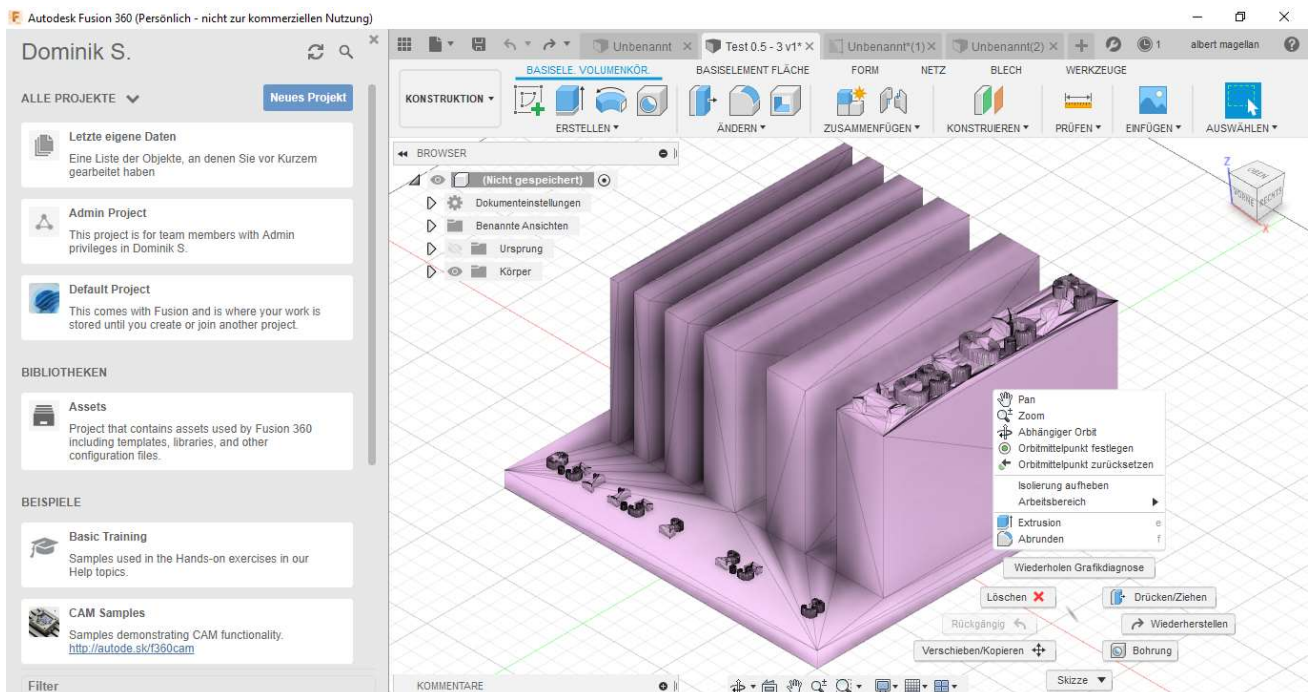
Das Haus ist so erstellt, dass sich darunter in der Grundplatte ein Loch befindet. Dies könnte dafür verwendet werden, um bestimmte Technik, zum Beispiel einen Arduino Nano zur Steuerung der LEDs in dem Haus zu verstauen.

Die Beschaffenheit der Grundplatte mit Haus kann man im Bild der Gesamtmodulation und der Gesamt-Zeichnung unten sehen.

Das zusammengesetzte Modell

Die verschiedenen Bestandteile des Modells wurden aus ihren entsprechenden Entwicklungsanwendungen als kompatible STL-Datei exportiert und in das Modell der Grundplatte mit Haus gesetzt.

Die technische Zeichnung



11 Die Benutzeroberfläche von Fusion 360 in der Ansicht "Konstruktion"

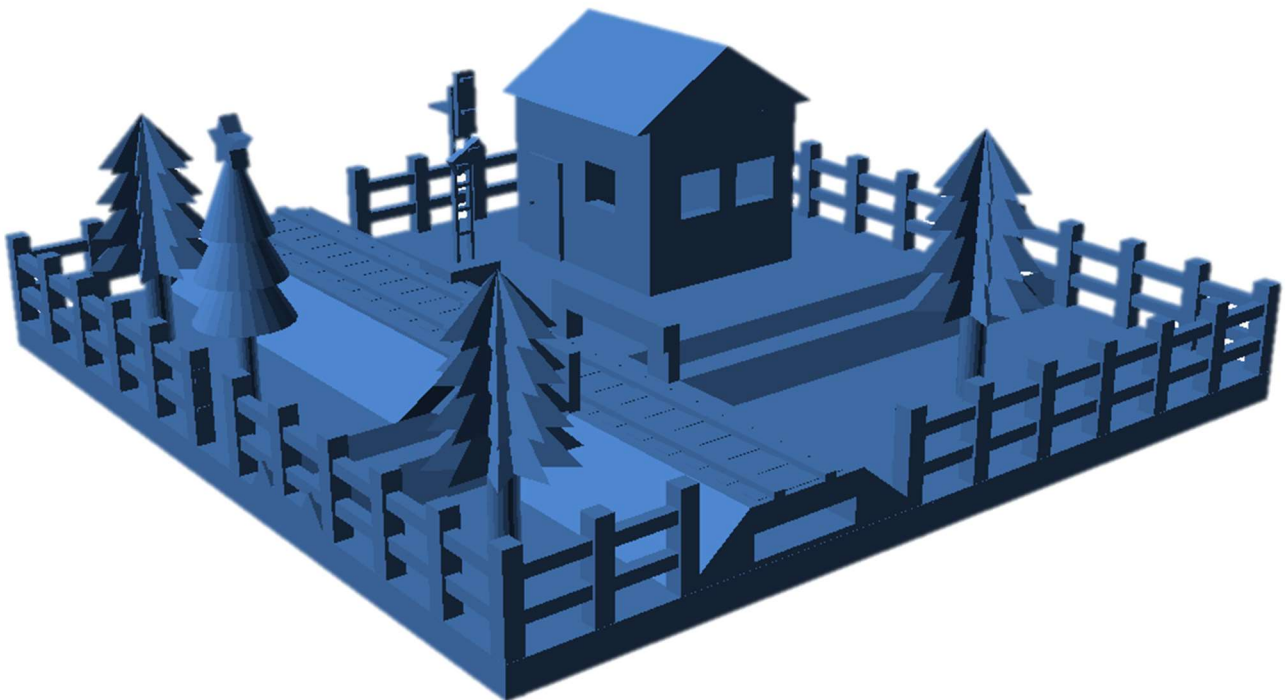
Aus dem Webprogramm wurde das gesamte Modell als eine STL-Datei exportiert und heruntergeladen.

Diese Datei haben wir dann in das CAD-Programm Fusion 360 von Autodesk, dem gleichen Hersteller von Tinkercad, geladen. Wenn man dort das Objekt in das richtige Format konvertiert (B-Rep), kann man davon technische Zeichnungen erstellen. Wir haben DIN-3 A Zeichnungen nach ISO mit dem Maßstab 1:1 gewählt.

Da es uns unmöglich erschien, in einer einzelnen technischen Zeichnung die Beschriftungen aller Teilelemente einzufügen, ohne dass dies stark an Übersichtlichkeit verlor, haben wir nur die die Bemaßungen der Grundplatte und des Hauses und lediglich die Positionen der anderen Teilelemente eingefügt.

Dafür haben wir dann mit dem gleichen Vorgehen wie oben für jedes Element noch einmal eine eigene technische Zeichnung mit genauer Bemaßung erstellt, welche sich oben bei den einzelnen Beschreibungen finden.

Da wir die ursprünglich DIN A3-formatigen Zeichnungen in diese Arbeit einfügen mussten, stimmen zwar die Maßstäbe innerhalb der Zeichnungen, allerdings in diesem Druck nicht mehr.



12 Das Gesamtmodell



Die LED-Arduino-Schaltung

Ursprünglich sollte das 3D-gedruckte CAD-Modell mit LEDs versehen werden, die von einem Arduino gesteuert werden. Aus Zeitgründen konnten wir dies nicht vollenden, wir haben aber die Technik einer kleinen LED-Blinckschaltung geplant

Mit folgendem Sketch kann man eine RGB-Led zufällig in den Farben grün, gelb, weiß und rot leuchten lassen:

```
int LEDblau=3;

int LEDrot=5;

int LEDgruen = 6;

int p=250; //p ist die Pausendauer von 0,25 Sekunden

int brightness1a=150; //Brightness definiert die Helligkeit, mit
der die RGB-Led leuchten soll.

int brightness1b=150;

int brightness1c=150;

int dunkel=0; //Mit dunkel ist eine Induktionsspannung von 0V
festgelegt, sodass die LED nicht mehr leuchtet.

int randomNumber; //hier wird ein Zufallsgenerator integriert

int y=250; //y hat die gleiche Bedeutung wie p

void setup()
{
    pinMode (LEDblau, OUTPUT); //hier wird festgelegt, dass die
Anschlüsse Outputs sind
    pinMode (LEDgruen, OUTPUT);
    pinMode (LEDrot, OUTPUT);
    randomSeed(analogRead(0));
}

void loop()
{
    randomNumber=random(4);

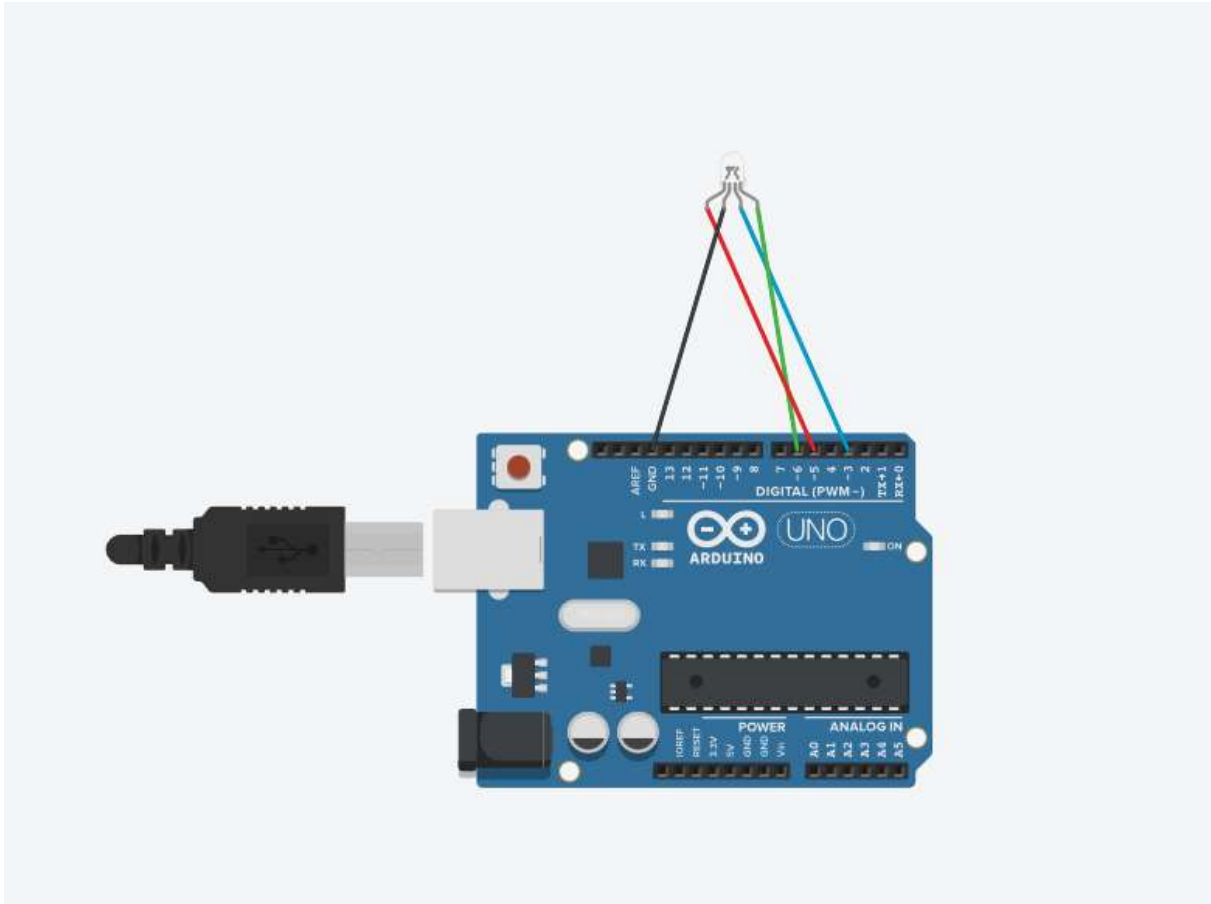
    delay (p);

    if(randomNumber<2)
    {
```

```
    analogWrite(LEDgruen, brightness1a);
    delay (y);
    analogWrite(LEDgruen, dunkel);
}
if (randNumber=2)
{
    analogWrite(LEDrot, brightness1a);
    delay (y);
    analogWrite(LEDrot, dunkel);
}
if (randNumber=3)
{
    analogWrite(LEDrot, brightness1a);
    analogWrite(LEDgruen, brightness1a);
    analogWrite(LEDblau,brightness1a),
    delay (y);
    analogWrite(LEDrot, dunkel);
    analogWrite(LEDgruen, dunkel);
    analogWrite(LEDblau, dunkel);
}
if (randNumber=4)
{
    analogWrite(LEDrot, brightness1a);
    analogWrite(LEDgruen, brightness1a),
    delay (y),
    analogWrite(LEDrot, dunkel);
    analogWrite(LEDgruen, dunkel);
}
}
```

In dem Sketch benutzt der Zufallsgenerator die Zahlen 0-4, wobei mit mehreren „if“-Schleifen die unterschiedlichen Farben geschaltet werden. Zum Beispiel wird bei allen Zahlen <2, also 1 und 0 die Farbe Grün geschaltet. Da die RGB-LED nur die Farben grün, blau und rot ansteuern kann, müssen Farben wie z. B. gelb gemischt werden.

In folgendem Bild ist noch die elektrische Schaltskizze des Arduino-Mikrocontrollers, der Verkabelung und der LED abgebildet:



Links

Diese Arbeit, die STL-Dateien des Modells und der einzelnen Elemente und die technischen Zeichnungen in hoher Auflösung finden sich zum online ansehen und Downloaden auf GitHub:

<https://github.com/AlbertMagellan/kitschy-winter-landscape/>