likely music
Probabilistische Musiknotation

Lukas Epple

post@lukasepple.de

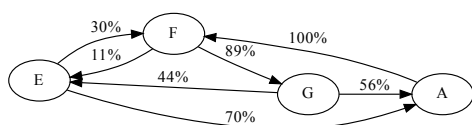27. September 2017

Zusammenfassung

likely music ist eine Software, um probabilistische Musik zu notieren und abzuspielen. Probabilistische Musik bedeutet in diesem Falle, dass die Interpretation der vorliegenden Notation deutlich freier ist als bei herkömmlicher Musik und auch die Reihenfolge der Noten betrifft. Um dies zu erreichen, wird ein eigenes Modell von Musiknotation verwendet. Anstelle von linearer Reihenfolge von Noten bzw. Akkorden tritt ein gerichteter Graph, in dem die Noten (bzw. Akkorde) die Knoten und die möglichen Übergange zwischen diesen die Kanten darstellen. Jeder Kante ist eine gewisse Wahrscheinlichkeit zugeordnet. Dieses Modell ist unter anderem sehr gut von einem Computer zu fassen, wodurch es möglich wird, solche Notationen automatisch zu „interpretieren" oder abzuspielen: Eine konkrete Notenabfolge wird gemäß der Notation ausgewürfelt.

Die Software *likely music* kann sowohl probabilistische Noten erstellen und editieren, als auch mittels MIDI diese abspielen oder als Audiodateien exportieren.

Idee

Der eigentlichen Idee ging ein mehr oder minder gescheitertes Projekt für diesen Wettbewerb voraus. Im Frühjahr diesen Jahres entschied ich mich, dieses – eine Demo [1] – abzubrechen, einfach weil ich befürchtete, es nicht bis zur Frist fertigstellen zu können. Die damalige Motivation für das Projekt speiste sich aus meiner Faszination für Demos an sich. Die Begeisterung für das neue speiste sich aus einer weiteren Faszination von mir, nämlich einer für Kunst, die durch Zufall entsteht. Ich erinnere mich besonders oft an Kunstinstallationen, die jeweils ihr gestaltendes Element aus Zufälligem, einen undurchschaubaren oder chaotischen Prozess bezieht. Beim Nachdenken über Zwölftonmusik, die – aus meiner Perspektive – ein wenig jenen Elements hat, kam mir die Grundidee für *likely music* auf dem Gang zwischen zwei Schulstunden: Nämlich ein Modell, um Musik zu beschreiben, die zufällig im Vortrag ist.
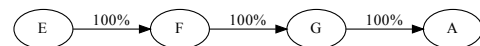
Das Modell, das ich aus Angst es zu vergessen, mehrmals aufschrieb, sieht Musik als gerichteten Graphen, wobei die Knoten Musiknoten einer bestimmten Länge und die Kanten zwischen ihnen die Wahrscheinlichkeit des Wechsel von der einen Note zu anderen sind. Vorstellen kann man sich es in etwa wie in der folgenden Grafik.
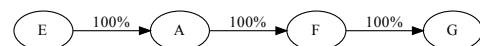

In diesem konkreten Graphen sind die Noten E, F, G und A als Knoten vertreten (der Einfachheit halber sind die Notenlängen weggelassen). Beispielsweise vom E führen zwei Kanten weg, eine zum F mit dreißigprozentiger Wahrscheinlichkeit und eine zum A mit siebzigprozentiger Wahrscheinlichkeit, d. h. nach dem E kommt in sieben von zehn Fällen das A und in den drei übrigen das F. Analog verhält es sich mit den anderen Noten.

Diese Darstellung ist in gewisser Weise auch nur eine ausdrucksstärkere Form einer normalen Notation, denn ein Weg durch den obigen Graphen könnte so aussehen:

Diese Interpretation, die eine Wahrscheinlichkeit von ca. 15% hat, entspricht einer einfachen, linearen Notation, wie sie in einem Gesangsbuch stehen könnte. Wir sehen also, dass solche probabilistische Noten (wie unser Graph von vorhin) durch ein Verfahren, das ich einfach in einer Erweiterung des Begriffs als Interpretieren bezeichne, auf eine lineare Notation reduziert werden können, die mit einem Instrument oder vom Computer gespielt werden können. Es ist sogar nicht nur eine lineare Notation, sondern – je nach vorgegebenem Graph – eine Vielzahl ihrer möglich. Beispielsweise wäre eine weitere:

Ähnlich enthält der ursprüngliche Graph weitere Möglichkeiten von klassischen Tonabfolgen. Insofern stellt eine probabilistische Notation eine ausdruckstärkere und mächtigere Notation dar, da sie beliebig viele klassische fassen kann.

Zu beachten ist bei den beiden Beispielinterpretationen noch: Sie sind nach vier No-

ten abgeschnitten, denn, da von jedem Knoten mindestens eine Kante ausgeht, könnte man den Graphen potentiell unendlich lang ablaufen und würde somit eine unendlich lange Interpretation generieren.

Was aus dieser Grundidee zu machen war, schien mir von Anfang an recht klar: Als Software implementieren, um ein graphisches Interface bereitzustellen, das es erlaubt, probabilistische Notation zu erstellen, zu editieren und abzuspielen.

Umsetzung

Gleich zu Beginn war klar, dass Haskell die Programmiersprache der Wahl werden sollte. Sie ist die Sprache, die ich in den letzten Jahren am aktivsten verwendet habe und mir einiges bietet: Statische Typisierung, um Fehler vorzubeugen, ein expressives Typsystem, das es erlaubt, Daten besser zu strukturieren, und funktionale Programmierparadigmen, die sich für mich sehr natürlich anfühlen und das Testen von Programmen erleichtern.

Zunächst konzentrierte ich mich darauf, den Graphen und den Interpretationsalgorithmus als Bibliothek zu implementieren. In der ersten Iteration dieser Bibliothek, noch *probable music* genannt, begann ich auch einen eigenen Softwaresynthesizer zu implementieren, der flexibel auf verschiedenen Plattformen und zu verschiedenen Zwecken verwendet werden kann. Der Synthesizer konnte jegliche Darstellungen von Klängen, Tönen oder Musik dank flexibler Architektur in tatsächliche Töne bzw. Audiowellen umwandeln. Dies ergab interessante Möglichkeiten, sich außerhalb des Zwölftonsystems zu bewegen. Die Tonerzeugung basierte dann auf einer freien Monade [2], die die Instruktionen ›Warten‹ und ›Abspielen‹ kannte. Indem man diese Instruktionen für verschiedene Audiosystem, wie SDL [4], Jack [3] oder auch Audiodateien wie WAV [5] implementierte, konnte man verschiedene Plattformen unterstützen. Allerdings gestaltete es sich schwierig, einen gut klingenden Synthesizer zu schreiben, denn die Messlatte ist im Vergleich zu realen Instrumenten hoch. Hinzu kamen noch einige Performance-Probleme mit meinem maschinennahen Audio-Code.

Also entschied ich mich, die Library vor allem auf den Graphen und die dazugehörigen Algorithmen zu fokusieren und zur Tonerzeugung eine geeignete Abstraktion zu verwenden, um diese zu vereinfachen. Ich habe hierfür MIDI gewählt, eine Technologie, die schon lang in allen Arten von Software und Hardware zur Musikproduktion verwendet wird. MIDI basiert auf einer Abfolge von zeitlich abgestimmten Nachrichten, wie zum Beispiel ›Note C an‹ oder ›Note C aus‹. Aufgrund dieser Nachrichten kann man die Erzeugung und das Abspielen von Musik zwischen mehreren Programmen aufteilen. Außerdem erlaubt es, die bereits existierende Infrastruktur für MIDI-Verarbeitung zu verwenden, die sehr beachtlich ist. Für MIDI verwendet *likely music* die Open-Source-Bibliothek Euterpea[1] [8], die unter anderem eine kleine Abstraktion über MIDI enthält. Sie erlaubt es, in einem internen Format Musik zu konstruieren und anschließend als MIDI zu exportieren bzw. an ein anderes Programm zur Weiterverarbeitung zu schicken.

Bei der Darstellung des Graphen habe ich mich vor allem darauf konzentriert, den Interpretationsalgorithmus, also das (zufällige) Ablaufen des Graphen, möglichst effizient zu gestalten. Da es sich um einen gerichteten Graphen handelt, ist es besonders wichtig zu wissen, wohin man von einem gegebenen Knoten aus gelangen kann bzw. welche Kanten von einem Knoten weggehen. So gelangt man in unserem Beispiel aus dem vorherigen Kapitel vom Knoten mit dem E zu den Knoten mit F und A. Es muss also möglichst effizient sein, die Kanten nachzuschlagen, die von einem Knoten *wegführen*. Mit der Datenstruktur *Map* [11] (im deutschen Sprachgebrauch typischerweise *assoziative Datenfeld*) kann genau das sehr leicht realisiert werden: Man verwendet die Knoten als Schlüssel und eine Liste von Kanten, die vom Schlüssel weggehen, als Elemente. Wenn der Algorithmus nun einen Knoten nachschlägt, erhält er direkt die Kanten, die von diesem Knoten weggehen und somit auch die nächsten möglichen Knoten. Dies ist die einzige Information, die in jedem Schritt benötigt wird. Die Operation des Nachschlagens hat in einem *Map* die Komplexität $O(\log n)$ [12], d. h. die Zeit, die benötigt wird, um ein Element nachzuschlagen, steigt mit dem Wachsen der Datenstruktur logarithmisch (d. h. weniger starkes Wachstum als linear!). Damit bleibt auch das Interpretieren großer Graphen ziemlich schnell. Der Code für die Datenstruktur findet sich im Abschnitt Library, Zeile 30 bis 43.

Der Interpretationsalgorithmus selbst ist rekursiv [15] gestaltet und findet sich in der Funktion `interpretation`, siehe Abschnitt Library, Zeile 52 bis 60. Diese Funktion benötigt einen initialisierten Pseudozufallszahlengenerator [13, 14], den zu interpretierenden Graphen in der eben besprochenen Datenstruktur und einen Startknoten. Nach Ablauf der Berechnung gibt die resultierende Interpretation im MIDI-Format von Euterpea [8] zurück. Zunächst wird der Startknoten im Graphen nachgeschlagen, so werden die Kanten bzw. die nächsten möglichen Knoten erhalten. Nun gibt es zwei Möglichkeiten für den weiteren Verlauf:

---

[1]Ich musste allerdings aufgrund von Inkompatibilitäten mit den aktuellen Haskell-Paketen diese selbst beheben [9]. Diese Änderung wartet [10] aktuell (Stand 23.09.2017) darauf, vom Hauptentwickler in den Code von Euterpea übernommen zu werden.

1. Es gibt keine Kanten, die von diesem Knoten ausgehen. Also wird die bisher generierte Interpretation einfach zurückgegeben, die Funktion terminiert.

2. Wenn es eine oder mehr Kanten vom Knoten aus gibt, wird eine (reelle) Zufallszahl zwischen 0 und 1 berechnet und mittels der Hilfsfunktion `edgeForRoll` (siehe Abschnitt Library, Zeile 62 - 67) die Kante erhalten, die gemäß des zufälligen Ergebnisses als nächstes abgelaufen werden soll. Nun ergibt sich das gleiche Problem wie zu Beginn der Interpretation: Man kennt einen Knoten und will wissen, wie es weitergeht. Also wird nach der Ermittlung des zweiten Knotens die MIDI-Nachrichten aus dem Startknoten extrahiert und dann der Interpretationsalgorithmus nochmal bzw. rekursiv aufgerufen – nur mit dem Folgeknoten als Startknoten. Dessen Ergebnis wird an die aktuellen MIDI-Nachrichten angehängt, was jener Aufruf auch seinerseits wieder macht. So entsteht rekursiv eine (potentiell unendliche) Verkettung von MIDI-Nachrichten, die letztlich die finale Interpretation ergeben.
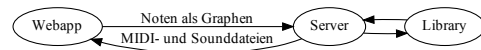
Da die meisten Graphen vermutlich vollständig untereinander verbunden sein werden, wie zum Beispiel der Beispielgraph im ersten Abschnitt, entstehen unendlich lange Interpretationen. Diese zu erstellen benötigt naturgemäß auch unendlich viel Zeit – der Interpretationsalgorithmus terminiert also nicht. Die einfache Antwort auf dieses Problem ist die Begrenzung der Länge der Interpretation auf eine gewisse Anzahl von Noten, was sich dank eines Sprachfeatures von Haskell – Lazy Evaluation [16] – leicht umsetzen lässt. Denn mit Lazy Evaluation wird nur das berechnet, was im Moment benötigt wird. Somit werden zum Beispiel nur die ersten vier benötigten Noten berechnet und nicht die unendlich vielen, die eigentlich noch darauf folgen würden – genau dies wird durch die Funktion `takeNotes` (siehe Abschnitt Library, Zeile 79 - 86) realisiert.

Nun können wir probabilistische Musik in Graphen darstellen, diese automatisch interpretieren und dank Euterpea nach MIDI exportieren. Was fehlt, ist eine angenehme Benutzerschnittstelle.

Zur Technologie für die Benutzerschnittstelle gab es für mich folgende Überlegungen: Zum einen sollte es leicht portabel bzw. auf jedem System laufen sowie außerdem einen begrenzten Entwicklungsaufwand mit sich bringen. Ich selbst entwickle meine Software auf GNU/Linux, aber zur Abgabe müsste es auf macOS und / oder Windows laufen. Alle größeren Frameworks für Graphische Interfaces für GNU/Linux, wie zum Beispiel Qt [21] oder GTK [22], laufen auch auf den anderen großen Betriebssystemen. Allerdings bin ich nicht besonders vertraut mit irgendeinem dieser Frameworks. Außerdem war ich mir nicht sicher, wie stressfrei die Verwendung dieser von Haskell aus sein würde (denn klassischerweise verwendet man C oder C++). Also entschied ich mich, *likely music* als Webapplikation, die einfach in gängigen Browsern läuft, zu implementieren. Das hat einige Vorteile für mich, unter anderem, dass es leicht zu testen ist, weil die Browser eigentlich überall gleich sind, und, dass ich schon einige Erfahrung in Webentwicklung hatte.

Ich hatte die Library allerdings in Haskell implementiert, in Browsern läuft jedoch nur JavaScript (ohne größeren Aufwand zumindest). Also musste ein Programm her, um die Kommunikation zwischen der Library und der Webapplikation zu realisieren. Ich entschied mich für eine Client-Server-Architektur [17], also einen Server, der die Interpretation und den Export von Sounddateien für den Client, also die Webapplikation, übernimmt. Der Client wiederum müsste sich ausschließlich um ein ansprechendes Interface kümmern. Die ungefähre Gesamtarchitektur sieht also nun so aus:

Der Server basiert auf den Libraries servant [18] als Webframework. Wie im Abschnitt Backend zu sehen, besteht das Serverbackend aus zwei Dateien Quelltext: In `Api.hs` wird die Struktur der REST-API [19] definiert, mittels der die Webapplikation mit dem Server kommuniziert. Die API bietet folgende Funktionalität an:
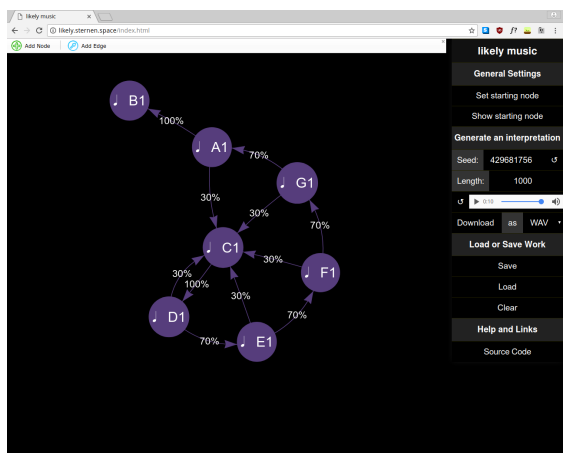
- `/interpretation/mid` An diesen Endpunkt schickt die Webapplikation einen Graphen plus einiger Parameter in Form von JSON [20] und erhält eine Interpretation auf Basis des Algorithmus als MIDI-Datei zurück.

- `/interpretation/wav` Gleich wie der obige Endpunkt, allerdings wird vorher noch das MIDI mittels des MIDI-Synthesizers fluidsynth [23] in eine WAV-Datei konvertiert, so dass man die Interpretation direkt anhören kann.

- Außerdem liefert der Server die statischen Dateien der Webapplikation, wie das nötige HTML, JavaScript und CSS.

Die erwähnten Parameter sind nur folgende drei:

- Der Anfangsknoten der Interpretation im Graphen, den der Algorithmus benötigt (wie oben besprochen).

- Die Länge der Interpretation als die maximale Anzahl an Noten in der Interpretation.

- Der Startwert für den Pseudozufallszahlengenerator [14], der für die Interpretation verwendet werden soll. Da derselbe Startwert in dieselbe Interpretation resultiert, erlaubt dies, sich interessante Interpretationen zu merken und zum Beispiel zu einer Interpretation noch die MIDI-Version zusätzlich herunterzuladen.

Dies ist auch schon alles, was das Serverbackend tut, denn es ist nur als minimaler Aufsatz auf die Library konzipiert. Das meiste für Benutzer*innen relevante passiert in der Webapplikation, die folgendermaßen aussieht:

Sie ist komplett in Englisch gehalten und sollte so in allen modernen Browsern laufen, getestet habe ich sie mit den aktuellen Versionen von Chrome [27] und Firefox [28].

Den Kern der Applikation bildet der Graph-Editor links, der auf der Library vis.js[2] [24] basiert. vis.js kümmert sich um einen sehr gut anpassbaren Graph-Editor, in dem der*die Benutzer*in Knoten und Kanten hinzufügen, löschen und ändern kann. Da die Library Callbacks [26] bereitstellt, ist es leicht, den Rest der Applikation mit dem Editor zu integrieren.
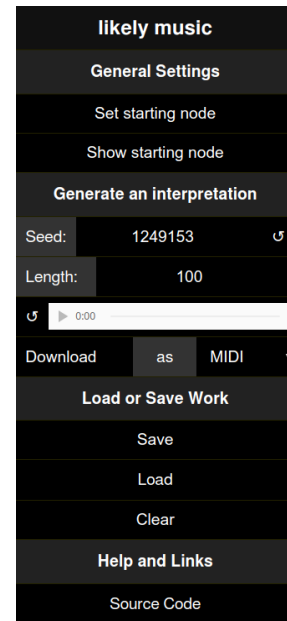
Wenn ein Knoten oder eine Kante geändert wird, wird diese Änderung in eine Zustandsvariable der Applikation mitübernommen und die Zusatzinformationen der Knoten und Kanten, also Notenlänge und Tonhöhe (Knoten) bzw. Wahrscheinlichkeit (Kante), von dem*der Benutzer*in in einer Einblendung abgefragt und ebenfalls abgespeichert. So gelingt es, den Graph-Editor so zu integrieren, dass der Graph zur Kommunikation mit dem Server und sonstiger Verarbeitung zur Verfügung steht. Die doppelte Speicherung der reinen Graphdaten kommt daher, dass vis.js es leider nicht erlaubt, die bereits im Editor vorhandenen Daten abzu-

___

[2]Eigentlich nur ein Teil von vis.js namens *network* [25], aber ich werde vis.js immer der Kürze halber synonym für *vis.js network* verwenden.

fragen. Daher büßt die Architektur der Applikation leider ein wenig an Eleganz ein.

In der Seitenspalte passiert dann alles, was relevant für die Verarbeitung der links entstehenden Notation ist. Zum einen kann der Notationsgraph abgespeichert oder ein gespeicherter geöffnet werden, zum anderen ist es möglich, Interpretationen generieren zu lassen, diese direkt im Browser abzuspielen oder als MIDI oder WAV herunterzuladen. Die Seitenspalte ist im Folgenden abgebildet.

Das Speichern und Öffnen von Notationen basiert auf JSON-Dateien [20] in bestimmtem Format, die als `<dateiname>.score.json` abgespeichert werden. Eine solche enthält eine Liste aller Knoten plus Zusatzinformationen und eine Liste aller Kanten plus Zusatzinformationen. Wie eine solche aussehen kann, sieht man im

Abschnitt Web (letzte Datei). Genau dieses Format wird übrigens auch zur Kommunikation mit dem Server verwendet, da es den Graphen verlustlos beschreiben kann.

Der Rest der Applikation kümmert sich vor allem um Interpretation und Export dieser. Oben in der Seitenleiste kann man die drei erwähnten Parameter setzen. Der Startknoten wird über Markieren desselben im Editor und klicken des entsprechenden Buttons gesetzt und kann durch Hervorhebung im Graphen auch angezeigt werden. Der Startwert kann manuell eingegeben (etwa, wenn man sich einen besonderen notiert hat) oder ein zufälliger durch Betätigung des Buttons neben dem Feld generiert werden. Die maximale Interpretationslänge ist dann darunter und wird ganz unspektakulär eingegeben.

Darunter befindet sich ein Audioplayer, mit dem erstellte Interpretationen direkt im Browser angehört werden können. Wenn man den Aktualisierungsbutton links betätigt, nimmt die Applikation alle Parameter sowie den aktuellen Graphen und sendet mithilfe der JavaScript Fetch API [29] den Graphen mitsamt der Parameter an den bereits erwähnten Endpunkt `/interpretation/wav`. Nach diesem Vorgang, der merklich Zeit benötigt, da fluidsynth [23] erst

das WAV generieren muss, wird die Audiodatei in den Player geladen[3] und kann direkt angehört werden.

Gleich unter dem Player kann man die Interpretation als MIDI oder WAV herunterladen. Dazu wählt man rechts eines der beiden Formate aus und klickt links auf „Download". Intern funktioniert dies genau gleich wie der Player, bloß dass jeweils der Endpunkte für das entsprechende Format verwendet und die Datei dann direkt heruntergeladen wird statt im Browser weiterverwendet wird.

Des weiteren werden der aktuelle Graph und die Parameter regelmäßig mittels LocalStorage [30] zwischengespeichert, die beim Öffnen der Webapplikation abgefragt wird. So ist gleich der letzte Stand vom letzten Mal geladen und man kann direkt weiterarbeiten.

Lizenzierung

Der gesamte Quelltext von *likely music* ist unter der *GNU Affero General Public License Version 3*, deren Text sich im Anhang im Abschnitt Lizenz findet, lizenziert. Die AGPL ist eine Freie-Software-Lizenz [32], das heißt, sie sichert dem*der Benutzer*in gegenüber dem Entwickler verschiedene Rechte (typischerweise nennt man vier) zu. Diese Rechte haben alle emanzipatorischen Charakter für den Nutzer: Das Recht die Software so auszuführen, wie der Nutzer es mag, natürlich offensichtlicherweise. Das Recht, den Quellcode zu erhalten und zu untersuchen. Das hilft vor allem dem*der Benutzer*in zu verstehen, was eigentlich auf seinem*ihrem Computer vor sich geht, und kann auch der Weiterbildung dienen. Die Freiheit, die Software frei und ohne Lizenzgebühren an andere weiterzugeben, ist mir besonders wichtig. Aufgrund diesen Umstandes kann freie Software unentgeltlich an jede*n weitergeben werden, was Zugang zu Software unabhängig des eigenen Geldbeutels erlaubt – vorausgesetzt man besitzt einen Computer. Diese Freiheit geht sogar noch weiter, dahingehend, dass auch die Modifikation ausdrücklich erlaubt (und erwünscht) ist. Somit kann nicht nur jede*r freie Software erhalten, sondern auch mitgestalten und verbessern. Auch andere freie Software kann profitieren, indem sie von anderen Projekten Code übernimmt. Dank der restriktiven Weitergabeklauseln kann aber nie freie Software verwendet oder verändert werden, ohne dass sie wieder freie Software wird. Freie Software erhält sozusagen ihre eigene Freiheit.

Mir ist dies an dieser Stelle ein besonderes Anliegen, weil ich – mit Sicherheit im Gegensatz zu den allermeisten anderen Wettbewerb-

teilnehmer*innen – mein Projekt komplett mit freier Software erstellen konnte. Ich war nicht auf eine von drei teuren Softwarelösungen großer Konzerne angewiesen, um meinen Beitrag anzufertigen, wie das zum Beispiel im Bereich Videoschnitt der Fall ist (auch weil es kaum ausgereifte freie Software in dem Bereich gibt).

Insofern sehe ich auch den emanzipatorischen Charakter von freier Software, denn Zugang zu Computern ist größtenteils auch dank von öffentlichen Bibliotheken selbstverständlich geworden, Zugang zu Software, die mehrere hundert Euro kostet, aber mit Sicherheit nicht. Der Preis von Software, die ein Konzern vielleicht auch irgendwann verwahrlosen lässt, ist sicher für viele eine Hürde, vielleicht sogar eine Hürde an diesem Wettbewerb teilzunehmen.

Ideen für die Zukunft

likely music als fertig zu bezeichnen wäre nicht ganz falsch und nicht ganz richtig. Es handelt sich zwar um eine voll funktionsfähige Software, aber dennoch ist noch einige Weiterentwicklung, für die ich keine Zeit mehr hatte, denkbar. Folgende Gedanken hatte ich bisher:

- **Unterstützung für Akkorde im Interface.** Zwar unterstützen Euterpea und die Library beide Akkorde, aber im Frontend gibt es keine Möglichkeit, solche hinzuzufügen, da ich die Euterpea-MIDI-Datenstruktur nicht vollständig in JavaScript nachgebaut habe. Dies zu beheben wäre für die Zukunft auf jeden Fall wünschenswert.

- **Mehrstimmige bzw. parallele probabilistische Musik.** Denkbar wäre es, eine Möglichkeit hinzuzufügen mehrere Startknoten auszuwählen, von denen dann zwei gleichzeitige Pfade durch den Graph ausgingen. Dies scheint mir die interessanteste Möglichkeit zu sein, Mehrstimmigkeit für *likely music* zu implementieren.

- **Import bereits durchkomponierter Musik.** Indem man die Möglichkeit schafft, bereits in konventionellen Notationsprogrammen erstellte Musik zu importieren, könnte man ein für den*die Benutzer*in angenehme Möglichkeit bieten, konventionell notierter Musik ein probabilistisches Element zu geben bzw. sie probabilistisch umzusetzen.

Diese Änderungen stehen nicht im Konflikt mit dem bisherigen Grundkonzept und -aufbau von *likely music*, dürften daher ohne größere Probleme umgesetzt werden können.

---

[3]Dabei muss man ein wenig Geduld haben, vor allem, wenn es über das Internet geschieht, da erst das WAV generiert und dann noch über das Internet geladen werden muss

Links

- Der gesamte Quelltext als Git-Repository
  `https://github.com/sternenseemann/likely-music`

- Eine laufende Instanz[4] von *likely music*
  `https://likely.sternen.space`

Literatur

[1] `https://de.wikipedia.org/wiki/Demoszene`

[2] `http://www.haskellforall.com/2012/07/purify-code-using-free-monads.html`

[3] `http://www.jackaudio.org/`

[4] `https://www.libsdl.org/index.php`

[5] `https://de.wikipedia.org/wiki/RIFF_WAVE`

[6] `https://www.midi.org/`

[7] `https://de.wikipedia.org/wiki/Musical_Instrument_Digital_Interface`

[8] `https://hackage.haskell.org/package/Euterpea`

[9] `https://github.com/sternenseemann/Euterpea2`

[10] `https://github.com/Euterpea/Euterpea2/issues/16`

[11] `https://hackage.haskell.org/package/containers-0.5.10.2/docs/Data-Map-Lazy.html#t:Map`

[12] `https://hackage.haskell.org/package/containers-0.5.10.2/docs/Data-Map-Lazy.html#v:lookup`

[13] `https://hackage.haskell.org/package/random-1.1/docs/System-Random.html#t:RandomGen`

[14] `https://en.wikipedia.org/wiki/Pseudorandom_number_generator`

[15] `https://de.wikipedia.org/wiki/Rekursion`

[16] `https://de.wikipedia.org/wiki/Lazy_Evaluation`

[17] `https://en.wikipedia.org/wiki/Client%E2%80%93server_model`

[18] `https://hackage.haskell.org/package/servant`

[19] `https://de.wikipedia.org/wiki/Representational_State_Transfer`

[20] `http://json.org/`

[21] `https://www.qt.io/`

[22] `https://www.gtk.org/`

[23] `http://www.fluidsynth.org/`

[24] `http://visjs.org/`

[25] `visjs.org/docs/network/`

[26] `https://en.wikipedia.org/wiki/Callback_(computer_programming)`

[27] `https://www.google.com/chrome/`

[28] `https://www.mozilla.org/en-US/firefox/`

[29] `https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API`

[30] `https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API`

[31] `https://www.gnu.org/licenses/agpl-3.0.html`

[32] `https://www.gnu.org/philosophy/free-sw.de.html`

---

[4] *likely music* ist bisher noch nicht auf Performance optimiert worden. Ich glaube nicht, dass genannte Server einen größeren Ansturm vor allem wegen des Exports zu WAV (fluidsynth [23] ist ziemlich langsam) aushalten würde. Daher möchte ich darum bitten, diesen Link nicht zu veröffentlichen, sondern, falls etwas in der Art gewünscht sein sollte, mit mir Rücksprache zu halten.
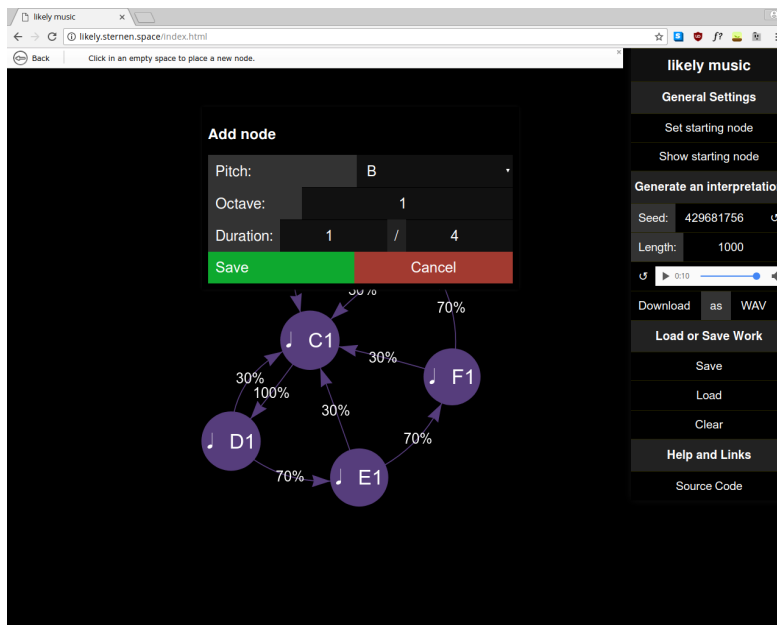
Anhang

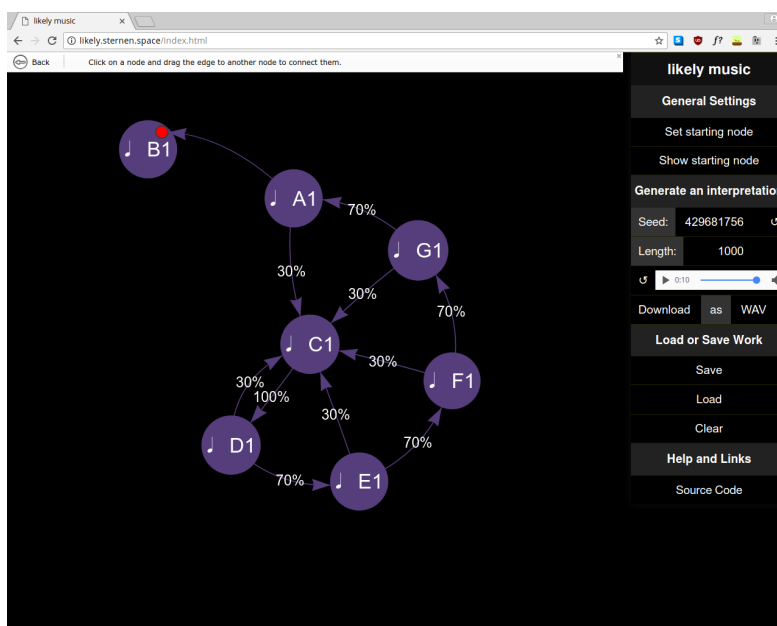## Screenshots

Abbildung 1: Hinzufügen eines Knotens

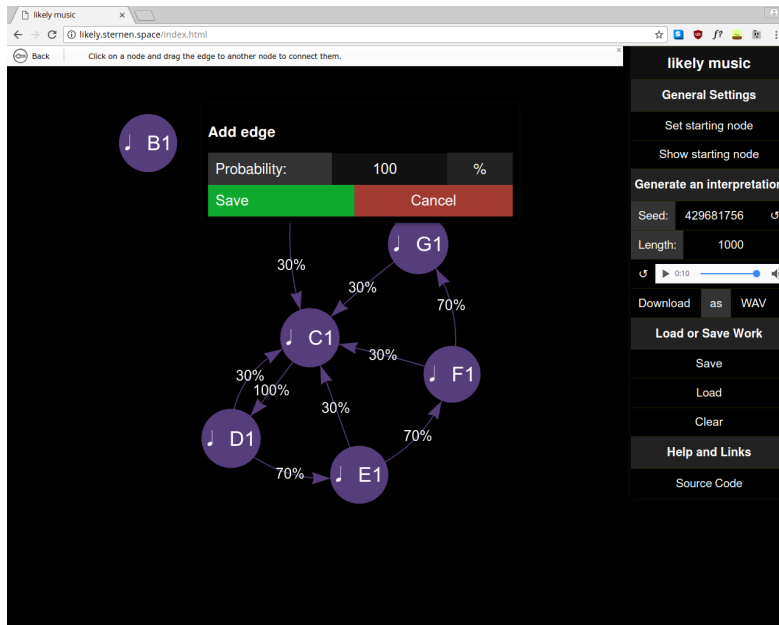Abbildung 2: Verbinden zweier Knoten mit einer Kante
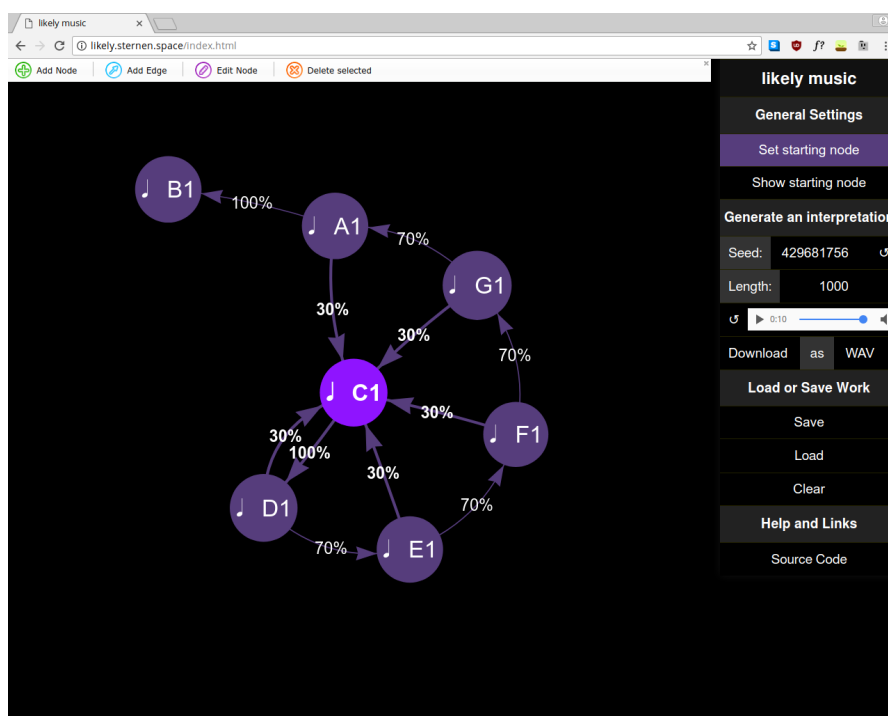
Abbildung 3: Setzen der Kanteneigenschaften

Abbildung 4: Setzen des Startknoten durch Auswählen desselben
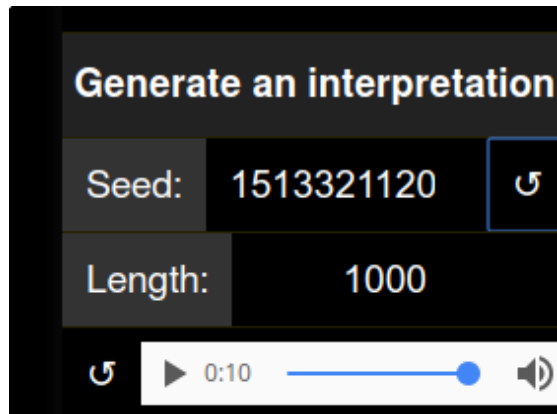
Abbildung 5: Auswürfeln eines neuen Startwerts per Knopfdruck
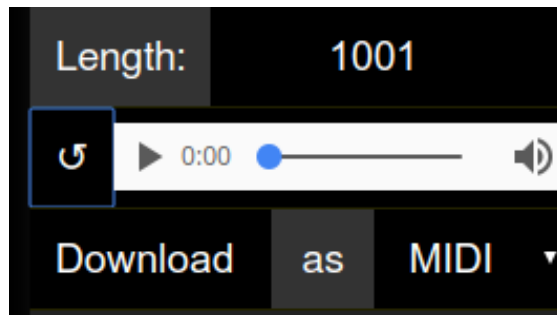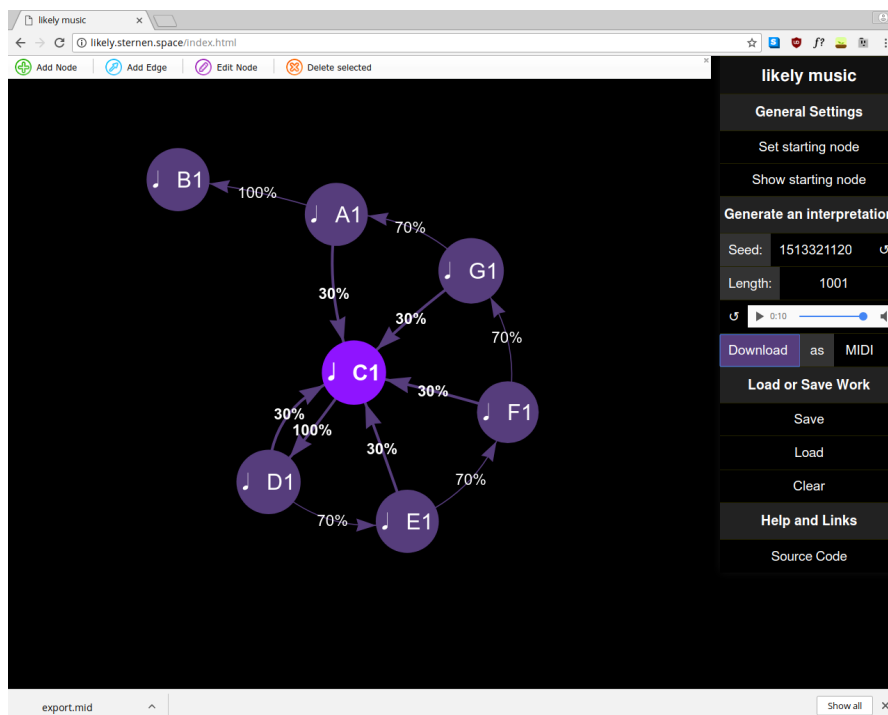

Abbildung 6: Laden der Interpretation in den Player

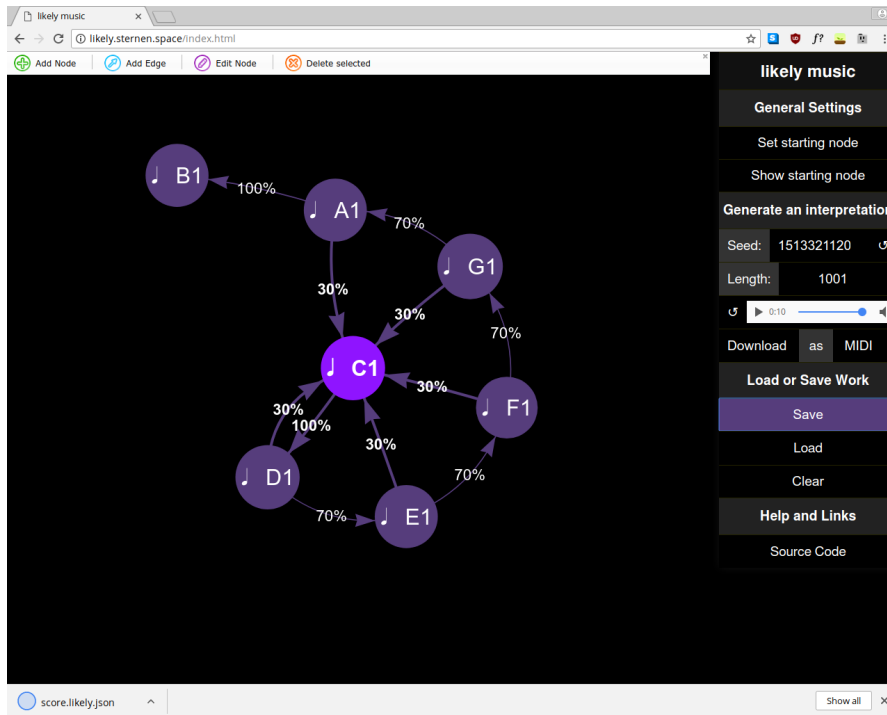Abbildung 7: Download der Interpretation als MIDI-Datei
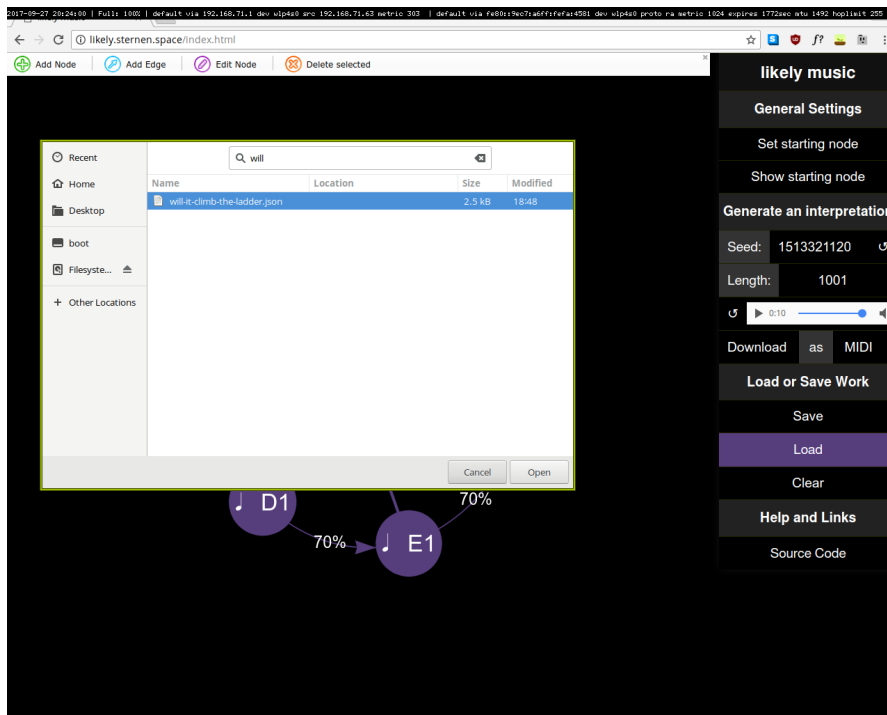
Abbildung 8: Speichern der Notation

Abbildung 9: Laden einer Notation

Quelltext

Library

lib/Sound/Likely.hs

```
1   -- Copyright 2017 Lukas Epple
2   --
3   -- This file is part of likely music.
4   --
5   -- likely music is free software: you can redistribute it and/or modify
6   -- it under the terms of the GNU Affero General Public License as published by
7   -- the Free Software Foundation, either version 3 of the License, or
8   -- (at your option) any later version.
9   --
10  -- likely music is distributed in the hope that it will be useful,
11  -- but WITHOUT ANY WARRANTY; without even the implied warranty of
12  -- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13  -- GNU Affero General Public License for more details.
14  --
15  -- You should have received a copy of the GNU Affero General Public License
16  -- along with likely music. If not, see <http://www.gnu.org/licenses/>.
17
18  {-# LANGUAGE OverloadedStrings #-}
19  {-# LANGUAGE FlexibleInstances #-}
20  module Sound.Likely
21    ( Probability
22    , ID
23    , Node (..)
24    , Edge (..)
25    , Graph (..)
26    , insertNode
27    , insertEdge
28    , interpretation
29    , takeNotes
30    , emptyMusic
31    , exampleGraph
32    ) where
33
34  import Control.Monad
35  import Data.Aeson
36  import Data.Aeson.Types (Parser ())
37  import Data.Maybe
38  import Data.Text (Text ())
39  import Euterpea
40  import System.Random
41  import qualified Data.Map as M
42  import qualified Data.Set as S
43
44  type Probability = Double
45  type ID = Text
46
47  data Node
48    = Node
49    { nId :: ID
50    , nMusic :: Music Pitch
51    } deriving (Show, Eq, Ord)
52
53  data Edge
54    = Edge
55    { eTo   :: Node
56    , eProb :: Probability
57    } deriving (Show, Eq, Ord)
58
59  newtype Graph = Graph { unGraph :: M.Map Node (S.Set Edge) }
60    deriving (Show, Eq, Ord)
61
62  insertNode :: Node -> Graph -> Graph
63  insertNode t = Graph . M.insertWith S.union t S.empty . unGraph
64
65  insertEdge :: Node -> Edge -> Graph -> Graph
66  insertEdge n e =
67    insertNode n . Graph . M.insertWith S.union n (S.singleton e) . unGraph
68
69  interpretation :: RandomGen g => g -> Graph -> Node -> Music Pitch
70  interpretation gen graph n = (nMusic n) :+:
71    recurse (fromMaybe S.empty (M.lookup n (unGraph graph)))
72    where (prob, gen') = randomR (0.0, 1.0) gen
73          recurse edges =
74            case edgeForRoll prob edges of
```

```
75              Nothing -> emptyMusic
76              Just nextEdge ->
77                interpretation gen' graph . eTo $ nextEdge
78
79   edgeForRoll :: Probability -> S.Set Edge -> Maybe Edge
80   edgeForRoll prob set =
81     if S.null set
82       then Nothing
83       else let curr = S.elemAt 0 set
84            in if prob <= eProb curr
85                  then Just curr
86                  else edgeForRoll (prob - eProb curr) (S.delete curr set)
87
88   emptyMusic :: Music a
89   emptyMusic = Prim (Rest 0)
90
91   exampleGraph :: Graph
92   exampleGraph = Graph $ M.fromList
93     [ (Node "bla" (c 4 qn), S.fromList [ Edge (Node "blub" (d 4 qn)) 1 ] )
94     , (Node "blub" (d 4 qn), S.fromList [ ])
95     ]
96
97   -- | Take the first @n@ notes of a 'Music'
98   takeNotes :: Integer -> Music a -> Music a
99   takeNotes _ m@(Prim _) = m
100  takeNotes n (Modify c m) = Modify c $ takeNotes n m
101  takeNotes _ m@(_ :=: _) = m
102  takeNotes n (m1 :+: m2)
103    | n < 1     = emptyMusic
104    | n == 1    = m1
105    | otherwise = m1 :+: takeNotes (n - 1) m2
106
107  instance FromJSON Node where
108    parseJSON = withObject "Node" $ \v ->
109      Node <$> v .: "id" <*> (Prim <$> v .: "music")
110
111  lookupNode :: Text -> [Object] -> Parser Node
112  lookupNode id nodes = do
113    matches <- filterM (fmap (== id) . (.: "id")) nodes
114    case matches of
115      [node] -> parseJSON (Object node)
116      _ -> fail "Couldn't␣match␣node␣by␣id"
117
118  buildMap :: [Object] -> [Object] -> Graph -> Parser Graph
119  buildMap _ [] m = pure m
120  buildMap nodes (e:es) m = do
121    toId <- e .: "to"
122    fromId <- e .: "from"
123    edge <- Edge <$> lookupNode toId nodes <*> e .: "prob"
124    from <- lookupNode fromId nodes
125    buildMap nodes es $ insertEdge from edge m
126
127  instance FromJSON Graph where
128    parseJSON = withObject "Graph" $ \v -> do
129      edges <- v .: "edges"
130      nodes <- v .: "nodes"
131      buildMap nodes edges $ Graph mempty
132
133  instance FromJSON (Primitive Pitch) where
134    parseJSON = withObject "Primitive" $ \v -> do
135      -- TODO Ratio _Integer_ is easy DOSable
136      -- RAM consumption
137      duration <- v .: "dur"
138      octave <- v .: "octave"
139      pitchClass <- v .: "pitch"
140      case pitchClass of
141        "Rest" -> pure $ Rest duration
142        p -> pure $ Note duration (read pitchClass, octave)
```

Backend

backend/Api.hs

```
1   -- Copyright 2017 Lukas Epple
2   --
3   -- This file is part of likely music.
4   --
5   -- likely music is free software: you can redistribute it and/or modify
6   -- it under the terms of the GNU Affero General Public License as published by
7   -- the Free Software Foundation, either version 3 of the License, or
8   -- (at your option) any later version.
9   --
10  -- likely music is distributed in the hope that it will be useful,
11  -- but WITHOUT ANY WARRANTY; without even the implied warranty of
12  -- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13  -- GNU Affero General Public License for more details.
14  --
15  -- You should have received a copy of the GNU Affero General Public License
16  -- along with likely music. If not, see <http://www.gnu.org/licenses/>.
17
18  {-# LANGUAGE OverloadedStrings #-}
19  {-# LANGUAGE FlexibleInstances #-}
20  {-# LANGUAGE DataKinds         #-}
21  {-# LANGUAGE TypeOperators     #-}
22  module Api where
23
24  import Data.Aeson
25  import Data.ByteString.Lazy (ByteString ())
26  import Data.Monoid ((<>))
27  import Data.Ratio
28  import Data.Text (Text ())
29  import GHC.Generics
30  import Servant.API
31  import Sound.Likely
32
33  type LikelyApi = "interpretation" :> Capture "format" OutputFormat
34                                    :> ReqBody '[JSON] GraphWithParams
35                                    :> Post '[OctetStream] ByteString
36            :<|> "seed" :> Get '[JSON] Int
37            :<|> Raw
38
39  data OutputFormat = Midi | Wav
40    deriving (Show, Eq, Ord)
41
42  instance FromHttpApiData OutputFormat where
43    parseUrlPiece "mid" = Right Midi
44    parseUrlPiece "wav" = Right Wav
45    parseUrlPiece x     = Left $ "Couldn't match " <> x <> " with {mid, wav}"
46
47  data GraphWithParams
48    = GraphWithParams
49    { gpParams :: Params
50    , gpGraph  :: Graph
51    } deriving (Show, Eq, Ord)
52
53  instance FromJSON GraphWithParams where
54    parseJSON = withObject "GraphWithParams" $ \v ->
55      GraphWithParams <$> v .: "params"
56                      <*> v .: "graph"
57
58  data Params
59    = Params
60    { pMaxHops      :: Int
61    , pStartingNode :: Node
62    , pSeed         :: Int
63    } deriving (Show, Eq, Ord)
64
65  instance FromJSON Params where
66    parseJSON = withObject "Params" $ \v ->
67      Params <$> v .: "maxhops"
68             <*> v .: "starting_node"
69             <*> v .: "seed"
```

backend/Main.hs

```
1   -- Copyright 2017 Lukas Epple
2   --
3   -- This file is part of likely music.
```

```haskell
   4  --
   5  --  likely music is free software: you can redistribute it and/or modify
   6  --  it under the terms of the GNU Affero General Public License as published by
   7  --  the Free Software Foundation, either version 3 of the License, or
   8  --  (at your option) any later version.
   9  --
  10  --  likely music is distributed in the hope that it will be useful,
  11  --  but WITHOUT ANY WARRANTY; without even the implied warranty of
  12  --  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
  13  --  GNU Affero General Public License for more details.
  14  --
  15  --  You should have received a copy of the GNU Affero General Public License
  16  --  along with likely music. If not, see <http://www.gnu.org/licenses/>.
  17
  18  {-# LANGUAGE OverloadedStrings #-}
  19  module Main where
  20
  21  import Api
  22
  23  import Codec.Midi (buildMidi)
  24  import Codec.ByteString.Builder
  25  import Control.Monad.IO.Class
  26  import Data.ByteString.Lazy (ByteString ())
  27  import qualified Data.ByteString.Lazy as B
  28  import Euterpea hiding (app)
  29  import GHC.IO.Handle
  30  import Network.Wai
  31  import Network.Wai.Handler.Warp
  32  import Servant
  33  import Sound.Likely
  34  import System.Directory
  35  import System.Exit
  36  import System.Environment
  37  import System.FilePath.Posix
  38  import System.IO
  39  import System.Process
  40  import System.Random
  41
  42  api :: Proxy LikelyApi
  43  api = Proxy
  44
  45  midiString :: ToMusic1 a => Music a -> ByteString
  46  midiString = toLazyByteString . buildMidi . toMidi . perform
  47
  48  server :: Server LikelyApi
  49  server = genInterpretation :<|> randomSeed :<|> serveDirectoryWebApp "web/dist"
  50
  51  randomSeed :: Handler Int
  52  randomSeed = liftIO newStdGen >>= return . fst . random
  53
  54  genInterpretation :: OutputFormat -> GraphWithParams -> Handler ByteString
  55  genInterpretation Midi g = do
  56    let params       = gpParams g
  57        maxHops      = fromIntegral . pMaxHops $ params
  58        randomGen    = mkStdGen $ pSeed params
  59        song         = interpretation randomGen (gpGraph g) (pStartingNode params)
  60    return . midiString $ takeNotes maxHops song
  61  genInterpretation Wav g = genInterpretation Midi g >>= synthWav
  62
  63  synthWav :: ByteString -> Handler ByteString
  64  synthWav midi = do
  65    inName <- tempFile "mid"
  66    liftIO $ B.writeFile inName midi
  67    outName <- tempFile "wav"
  68    (_, _, _, ph) <- liftIO $
  69      createProcess_ "fluidsynth"
  70        (proc "fluidsynth"
  71          [ "-a", "file"
  72          , "-F", outName
  73          , "-i"
  74  --        , "/usr/share/soundfonts/FluidR3_GM.sf2"
  75          , "/nix/store/591834mz365ccwyj3ah2d66ncsqvp8w9-Fluid-3/share/soundfonts/
  76                FluidR3_GM2-2.sf2"
  76          , inName ])
  77          { std_in = CreatePipe }
  78    code <- liftIO $ waitForProcess ph
  79    case code of
  80      ExitFailure _ -> throwError err500 { errBody = "fluidsynth␣failed" }
  81      ExitSuccess -> do
```

```
82        out <- liftIO $ B.readFile outName
83        liftIO $ removePathForcibly outName
84        return out
85
86  tempFile :: String -> Handler FilePath
87  tempFile ext = try 0
88    where maxtries = maxBound
89          try :: Int -> Handler FilePath
90          try n
91            | n < maxtries = do
92                progName <- liftIO $ getProgName
93                let path = "/tmp" </> addExtension (makeValid progName ++ "-" ++ show n)
                        ext
94                exists <- liftIO $ doesFileExist path
95                if exists
96                  then try (n + 1)
97                  else pure path
98            | otherwise = throwError err500 { errBody = "no␣temp␣files" }
99  app :: Application
100 app = serve api server
101
102 main :: IO ()
103 main = newStdGen >> run 8081 app
```

Web

web/source/index.html

```
 1  <!--
 2
 3    Copyright 2017 Lukas Epple
 4
 5    This file is part of likely music.
 6
 7    likely music is free software: you can redistribute it and/or modify
 8    it under the terms of the GNU Affero General Public License as published by
 9    the Free Software Foundation, either version 3 of the License, or
10    (at your option) any later version.
11
12    likely music is distributed in the hope that it will be useful,
13    but WITHOUT ANY WARRANTY; without even the implied warranty of
14    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
15    GNU Affero General Public License for more details.
16
17    You should have received a copy of the GNU Affero General Public License
18    along with likely music. If not, see <http://www.gnu.org/licenses/>.
19
20  -->
21  <!doctype html>
22  <html>
23      <head>
24          <meta charset="utf-8">
25          <meta http-equiv="x-ua-compatible" content="ie=edge" />
26          <meta name="viewport" content="width=device-width,␣initial-scale=1" />
27          <title>likely music</title>
28          <link rel="stylesheet" type="text/css" href="custom.css">
29          <link rel="stylesheet" type="text/css" href="vis.min.css">
30          <script src="main.js"></script>
31      </head>
32      <body>
33          <div id="network"></div>
34          <div id="sidebar">
35              <h1>likely music</h1>
36              <h2>General Settings</h2>
37              <button id="set-starting-node">Set starting node</button>
38              <button id="show-starting-node">Show starting node</button>
39              <h2>Generate an interpretation</h2>
40              <div class="multi-inputs">
41                  <label for="seed">Seed:</label>
42                  <input type="number" id="seed">
43                  <button id="random-seed">&#8634;</button>
44              </div>
45              <div class="multi-inputs">
46                  <label for="hop-count">Length:</label>
47                  <input type="number" min="0" id="hop-count" placeholder="Max.␣note␣
                      count">
48              </div>
49              <div id="player-container">
50                  <button id="reload-player">&#8634;</button>
51                  <audio id="player" controls></audio>
52              </div>
53              <div class="multi-inputs">
54                  <button id="download-audio">Download</button>
55                  <label for="format">
56                      as
57                  </label>
58                  <select id="format">
59                      <option value="mid">MIDI</option>
60                      <option value="wav">WAV</option>
61                  </select>
62              </div>
63              <h2>Load or Save Work</h2>
64              <button id="gen-score" class="save">Save</button>
65              <label for="upload-score" class="custom-file">
66                  <input type="file" id="upload-score" >
67                  <span>Load</span>
68              </label>
69              <button id="clear-score" class="cancel">Clear</button>
70              <h2>Help and Links</h2>
71              <a href="https://github.com/sternenseemann/likely-music">Source Code</a>
72          </div>
73          <div id="edge-overlay" class="hidden␣dialog">
74              <h2><span id="edge-operation"></span> edge</h2>
75              <div class="multi-inputs">
```

```
76          <label for="prob">Probability:</label>
77          <input id="prob" type="number" min="0.0" max="100">
78          <span>%</span>
79        </div>
80        <div class="multi-inputs">
81          <button class="save" id="edge-save">Save</button>
82          <button class="cancel" id="edge-cancel">Cancel</button>
83        </div>
84      </div>
85      <div id="node-overlay" class="hidden␣dialog">
86        <h2><span id="node-operation"></span> node</h2>
87        <div class="multi-inputs">
88          <label for="pitch">Pitch:</label>
89          <select id="pitch"></select>
90        </div>
91        <div class="multi-inputs">
92          <label for="octave">Octave:</label>
93          <input id="octave" type="number" step="1">
94        </div>
95        <div class="multi-inputs">
96          <label>Duration:</label>
97          <input min="0" id="numerator" type="number" step="1">
98          <span>/</span>
99          <input min="0" id="denominator" type="number" step="1">
100       </div>
101       <div class="multi-inputs">
102         <button class="save" id="node-save">Save</button>
103         <button class="cancel" id="node-cancel">Cancel</button>
104       </div>
105     </div>
106   </body>
107 </html>
```

web/source/custom.css

```
1   /*  Copyright 2017 Lukas Epple
2
3     This file is part of likely music.
4
5     likely music is free software: you can redistribute it and/or modify
6     it under the terms of the GNU Affero General Public License as published by
7     the Free Software Foundation, either version 3 of the License, or
8     (at your option) any later version.
9
10    likely music is distributed in the hope that it will be useful,
11    but WITHOUT ANY WARRANTY; without even the implied warranty of
12    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13    GNU Affero General Public License for more details.
14
15    You should have received a copy of the GNU Affero General Public License
16    along with likely music. If not, see <http://www.gnu.org/licenses/>.
17  */
18
19  body {
20      font-size:1em;
21      font-family: sans-serif;
22      margin: 0px;
23      background-color: black;
24  }
25
26  #network {
27      width: 79%;
28      float:left;
29      height: 100vh;
30  }
31
32  #sidebar {
33      width: 20%;
34      float:right;
35      color: white;
36      background-color: black;
37      box-shadow: 0px 0px 20px #111;
38      font-size: 1.2rem;
39  }
40
41  #sidebar > * {
42      width: 100%;
43      border-top: 1px solid #232200;
44      color: white;
45      padding-left: 0px;
46      padding-right: 0px;
47      margin: 0;
48  }
49
50  #sidebar button:hover, #sidebar input:hover,
51  #sidebar .custom-file:hover, #sidebar select:hover, #sidebar a:hover {
52      background-color: #563d7c;
53  }
54
55  #sidebar button, #sidebar input, #sidebar .custom-file, #sidebar select, #sidebar a {
56    background-color: #000;
57  }
58
59  #sidebar h1 {
60      font-size: 1.5rem;
61      padding-top: 0.75rem;
62      padding-bottom: 0.75rem;
63      text-align: center;
64      background-color: #111;
65  }
66
67  #sidebar h2 {
68      font-size: 1.2rem;
69      padding-top: 0.9rem;
70      padding-bottom: 0.9rem;
71      text-align: center;
72      background-color: #222;
73  }
74
75  #sidebar select {
76    color: white;
77    border: none;
78    padding: 0.75rem;
```

```
 79       font-size: 1.2rem;
 80       width: auto;
 81   }
 82
 83   #sidebar a {
 84       padding-bottom: 0.75rem;
 85       padding-top: 0.75rem;
 86       display: inline-block;
 87       text-decoration: none;
 88       color: white;
 89       text-align: center;
 90   }
 91
 92   button {
 93       border: none;
 94       color: white;
 95       background-color:black;
 96       font-size: 1.2rem;
 97       margin:0;
 98       padding:0.75rem;
 99   }
100
101   input[type="number"] {
102       background-color: #333;
103       color: white;
104       border: none;
105       text-align: center;
106       font-size: 1.2rem;
107       padding:0.75rem;
108   }
109
110   .custom-file {
111       top:0;
112       right:0;
113       position: relative;
114       display: inline-block;
115       height: 3rem;
116   }
117
118   .custom-file input[type="file"] {
119       position: relative;
120       top:0;
121       left:0;
122       right:0;
123       z-index:0;
124       opacity: 0;
125       width: 100%;
126       height: 100% !important;
127       margin:0;
128       padding:0;
129   }
130
131   .custom-file span {
132       text-align: center;
133       position: absolute;
134       top: 0;
135       left: 0;
136       right: 0;
137       z-index: 1;
138       width: 100%;
139       height: 3rem;
140       pointer-events: none;
141       background-color: transparent !important;
142       font-size: 1.2rem;
143       line-height: 1.5rem;
144       padding-top: 0.75rem;
145       padding-bottom: 0.75rem;
146   }
147
148   .dialog {
149       position: absolute;
150       top: 10%;
151       left: 25%;
152       width: 30%;
153       min-width:500px;
154       padding: 10px;
155       background-color: black;
156       color: white;
157       box-shadow: 0px 0px 10px #111;
```

```css
158  }
159
160  .dialog select {
161    padding: 0.75rem;
162    font-size: 1.5rem;
163    color: white;
164    background-color: #111;
165    border: none;
166  }
167
168  .hidden {
169      visibility:hidden;
170  }
171
172  .dialog > div {
173    width: 100%;
174  }
175
176  .dialog button {
177      padding: 0.75rem;
178      font-size: 1.5rem;
179  }
180
181  .dialog input {
182    font-size: 1.5rem;
183  }
184
185  button.cancel {
186      background-color: #a23a30;
187  }
188
189  button.save {
190      background-color: #0ea92f;
191  }
192
193  .dialog .multi-inputs {
194    font-size: 1.5rem;
195  }
196
197  .multi-inputs {
198    display: inline-flex;
199    flex-direction: row;
200    flex-wrap: nowrap;
201    justify-content: flex-start;
202    align-items: baseline;
203    width: 100%;
204  }
205
206  .multi-inputs > * {
207    flex-grow: 1;
208    flex-basis: auto;
209    transition: width 0.7s ease-out;
210    max-height: 100%;
211    text-align: center;
212  }
213
214  .multi-inputs :nth-child(1) {
215    text-align: left;
216  }
217
218  .multi-inputs label {
219    display: inline-block;
220    background-color: #333;
221    padding: 0.75rem;
222  }
223
224  .multi-inputs input {
225    display: inline-block;
226    color: white;
227    background-color: #111;
228    padding: 0.75rem;
229    border: none;
230    min-width: 0px;
231  }
232
233  .multi-inputs span {
234    display: inline-block;
235    padding: 0.75rem;
236    background-color: #222;
```

```
237  }
238
239  .multi-inputs button {
240      padding: 0.75rem;
241  }
242
243  #player-container {
244    display: inline-flex;
245    align-items: center;
246  }
247
248  #player-container > * {
249    flex: auto;
250  }
```

web/source/main.js

```
 1   //  Copyright 2017 Lukas Epple
 2   //
 3   //  This file is part of likely music.
 4   //
 5   //  likely music is free software: you can redistribute it and/or modify
 6   //  it under the terms of the GNU Affero General Public License as published by
 7   //  the Free Software Foundation, either version 3 of the License, or
 8   //  (at your option) any later version.
 9   //
10   //  likely music is distributed in the hope that it will be useful,
11   //  but WITHOUT ANY WARRANTY; without even the implied warranty of
12   //  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13   //  GNU Affero General Public License for more details.
14   //
15   //  You should have received a copy of the GNU Affero General Public License
16   //  along with likely music. If not, see <http://www.gnu.org/licenses/>.
17
18   import vis from 'vis';
19   import { Map } from 'immutable';
20   // types / internals
21
22   const valid_pitches = [
23       'Rest',
24       'Cf', 'C', 'Cs',
25       'Df', 'D', 'Ds',
26       'Ef', 'E', 'Es',
27       'Ff', 'F', 'Fs',
28       'Gf', 'G', 'Gs',
29       'Af', 'A', 'As',
30       'Bf','B', 'Bs'
31   ];
32
33   const display_pitches = [
34       'Rest',
35       'C', 'C', 'C',
36       'D', 'D', 'D',
37       'E', 'E', 'E',
38       'F', 'F', 'F',
39       'G', 'G', 'G',
40       'A', 'A', 'A',
41       'B', 'B', 'B'
42   ];
43
44   function displayPitch(pitch) {
45       var i = valid_pitches.indexOf(pitch);
46       if(i === -1) {
47           throw 'Invalid pitch';
48       } else {
49           return display_pitches[i];
50       }
51   }
52
53   function standard_rests(dur) {
54       if(dur.numerator === 1) {
55           switch(dur.denominator) {
56               case 1:
57                   return ' ';
58                   break;
59               case 2:
60                   return ' ';
61                   break;
62               case 4:
63                   return ' ';
64                   break;
65               case 8:
66                   return ' ';
67                   break;
68               case 16:
69                   return ' ';
70                   break;
71               case 32:
72                   return ' ';
73                   break;
74               case 64:
75                   return ' '
76                   break;
77               case 128:
78                   return ' '
```

```
 79                         break;
 80                 default:
 81                     return null;
 82                     break;
 83             }
 84         } else {
 85             return null;
 86         }
 87     }
 88
 89     function standard_notes(dur) {
 90         if(dur.numerator === 1) {
 91             switch(dur.denominator) {
 92                 case 1:
 93                     return ' ';
 94                     break;
 95                 case 2:
 96                     return ' ';
 97                     break;
 98                 case 4:
 99                     return ' ';
100                     break;
101                 case 8:
102                     return ' ';
103                     break;
104                 case 16:
105                     return ' ';
106                     break;
107                 case 32:
108                     return ' ';
109                     break;
110                 case 64:
111                     return ' '
112                     break;
113                 case 128:
114                     return ' '
115                     break;
116                 default:
117                     return null;
118                     break;
119             }
120         } else if(dur.numerator === 2 && dur.denominator === 1) {
121             return ' '
122         } else {
123             return null;
124         }
125     }
126
127     function compute_dot_times(dur, den) {
128         let term = den * ( (2 / den) - (dur.numerator / dur.denominator));
129         return [ den, -Math.log2(term) ];
130     }
131
132     function musical_symbol(lookup, dur) {
133         // unicode characters sometimes hide from you!
134         const dot = ' ';
135         let isNat = n => {
136             if (typeof n !== 'number')
137                 return false;
138             return (n >= 0.0) && (Math.floor(n) === n) && n !== Infinity;
139         };
140         var standard_symbol = lookup(dur);
141         var dots = [0, 1, 2, 3, 4, 5, 6, 7 ].map(compute_dot_times.bind(this, dur))
142             .filter(([den, dots]) => isNat(dots));
143
144         if(standard_symbol !== null) {
145             return standard_symbol;
146         } else if (dots.length !== 0) {
147             var symbol = lookup(new Rational(1, dots[0][0])) + ' ';
148             for(var i = dots[0][1]; i > 0; i--) {
149                 symbol = symbol + dot;
150             }
151             return symbol;
152         } else {
153             return dur.toString();
154         }
155     }
156
157     class Music {
```

```
158     constructor(dur, pitch_class, octave) {
159         this.dur = dur;
160         if(valid_pitches.indexOf(pitch_class) !== -1) {
161             this.pitch = pitch_class;
162         } else {
163             throw `Invalid pitch class '${pitch_class}'`;
164         }
165         this.octave = octave;
166     }
167
168     toString() {
169         if(this.pitch === 'Rest') {
170             return `${displayPitch(this.pitch)} for ${this.dur.toString()}`;
171         } else {
172             return `${displayPitch(this.pitch)}${this.octave} for ${this.dur.toString()
173                 }`;
174         }
175     }
176
177     nodeText() {
178         if(this.pitch === 'Rest') {
179             return `${musical_symbol(standard_rests, this.dur)} Rest`;
180         } else {
181             return `${musical_symbol(standard_notes, this.dur)}    ${displayPitch(this.
182                 pitch)}${this.octave}`
183         }
184     }
185
186     static fromObject(obj) {
187         return new Music(Rational.fromObject(obj.dur), obj.pitch, Number(obj.octave));
188     }
189 }
190
191 class Rational {
192     constructor(a, b) {
193         this.numerator = a;
194         this.denominator = b;
195         this.reduce();
196     }
197
198     reduce() {
199         let gcd = (a, b) => !b ? a : gcd(b, a % b);
200         let div = function(a, b) {
201             if(b === 0) {
202                 throw 'Divide by zero';
203             } else {
204                 return Math.floor(a / b);
205             }
206         };
207
208         var d = gcd(this.numerator, this.denominator);
209         this.numerator = div(this.numerator, d);
210         this.denominator = div(this.denominator, d);
211     }
212
213     toString() {
214         return `${this.numerator}/${this.denominator}`;
215     }
216
217     static fromObject(obj) {
218         return new Rational(obj.numerator, obj.denominator);
219     }
220 }
221
222 function collectGraphData(nodeDate, edgeData) {
223     return {
224         nodes: [... nodeData.values()].map(x => ({
225             id: x.nodeData.id,
226             music: x.music
227         })),
228         edges: [... edgeData.values()].map(x => ({
229             id: x.edgeData.id,
230             from: x.edgeData.from,
231             to: x.edgeData.to,
232             prob: x.prob
233         }))
234     };
235 }
```

```
235
236  function importGraphData(g) {
237      nodeData = new Map();
238      edgeData = new Map();
239      var nodeSet = new vis.DataSet({});
240      var edgeSet = new vis.DataSet({});
241      for(let node of g.nodes) {
242          var music = Music.fromObject(node.music);
243          var data = { id: node.id, label: music.nodeText() };
244          nodeData = nodeData.set(node.id, { nodeData: data, music: node.music });
245          nodeSet.add(data);
246      }
247
248      for(let edge of g.edges) {
249          var data = {
250              id: edge.id,
251              from: edge.from,
252              to: edge.to,
253              label: `${edge.prob * 100}%`
254          };
255          edgeData = edgeData.set(edge.id, { edgeData: data, prob: edge.prob });
256          edgeSet.add(data);
257      }
258
259      network.setData({ nodes: nodeSet, edges: edgeSet });
260  }
261
262  // helper
263
264  function download(url, filename) {
265      var link = document.createElement('a');
266      link.setAttribute('href', url);
267      link.setAttribute('download', filename);
268      link.style.display = 'none';
269      document.body.appendChild(link);
270      link.click();
271      document.body.removeChild(link);
272  }
273
274  function downloadFile(content_type, filename, content) {
275      var data = `data:${content_type},${encodeURIComponent(content)}`;
276      download(data, filename);
277  }
278
279
280  // graph code
281
282  var nodeData = Map();
283  var edgeData = Map();
284  var network = null;
285  var starting_node_id = null;
286
287
288  function showOverlay(id) {
289      document.getElementById(id).classList.remove('hidden');
290  }
291
292  function genericEditNode(data, callback) {
293      function clearOverlay() {
294          document.getElementById('node-save').onclick = null;
295          document.getElementById('node-cancel').onclick = null;
296          hideOverlay('node-overlay');
297      }
298
299      function saveNode(data, callback) {
300          var duration = new Rational(document.getElementById('numerator').value,
301              document.getElementById('denominator').value);
302          var music = new Music(duration, document.getElementById('pitch').value,
303              Number(document.getElementById('octave').value));
304          data.label = music.nodeText();
305          clearOverlay();
306          callback(data);
307          nodeData = nodeData.set(data.id, { music: music, nodeData: data });
308      }
309
310      function discardNode(callback) {
311          clearOverlay();
312          callback(null);
313      }
```

```
314
315        showOverlay('node-overlay');
316        var node = nodeData.get(data.id);
317        if(node !== undefined) {
318            var music = node.music;
319            document.getElementById('pitch').value = music.pitch;
320            document.getElementById('octave').value = music.octave;
321            document.getElementById('numerator').value = music.dur.numerator;
322            document.getElementById('denominator').value = music.dur.denominator;
323        }
324        document.getElementById('node-save').onclick = saveNode.bind(this, data, callback);
325        document.getElementById('node-cancel').onclick = discardNode.bind(this, callback);
326    }
327
328    function genericEditEdge(data, callback) {
329        function clearOverlay() {
330            document.getElementById('edge-save').onclick = saveEdge.bind(this, data,
                    callback);
331            document.getElementById('edge-cancel').onclick = discardEdge.bind(this,
                    callback);
332            hideOverlay('edge-overlay');
333        }
334
335        function saveEdge(data, callback) {
336            // for some reason, editWithoutDrag
337            // sets from & to the node respective
338            // node objects, which results in the edge
339            // disappearing.
340            if (typeof data.to === 'object')
341                data.to = data.to.id
342            if (typeof data.from === 'object')
343                data.from = data.from.id
344
345            var prob = document.getElementById('prob').value / 100;
346            data.label = `${prob * 100}%`;
347            clearOverlay();
348            callback(data);
349            edgeData = edgeData.set(data.id, { prob: prob, edgeData: data } );
350        }
351
352        function discardEdge(callback) {
353            clearOverlay();
354            callback(null);
355        }
356
357        showOverlay('edge-overlay');
358        var edge = edgeData.get(data.id);
359        if(edge !== undefined) {
360            document.getElementById('prob').value = edge.prob * 100;
361        }
362        document.getElementById('edge-save').onclick = saveEdge.bind(this, data, callback);
363        document.getElementById('edge-cancel').onclick = discardEdge.bind(this, callback);
364    }
365
366    function deleteFromMap(data, callback) {
367        for(let node of data.nodes) {
368            nodeData = nodeData.delete(node);
369        }
370
371        for(let edge of data.edges) {
372            edgeData = edgeData.delete(edge);
373        }
374
375        callback(data);
376    }
377
378
379    function hideOverlay(id) {
380        document.getElementById(id).classList.add('hidden');
381    }
382
383    function handleImport() {
384        var files = document.getElementById('upload-score').files;
385        if(files.length === 0) {
386            alert('Select a file first!');
387        } else {
388            var file = files[0];
389            var reader = new FileReader();
390            reader.addEventListener('loadend', function() {
```

```
391             var parsed = JSON.parse(this.result);
392             if(parsed === undefined) {
393                 alert('Could not parse likely score');
394             } else {
395                 var confirmation = window.confirm('Proceeding will overwrite the
                        current graph. Are you sure?');
396                 if(confirmation) {
397                     try {
398                         importGraphData(parsed);
399                     } catch(e) {
400                         alert(`Could not import likely score, probably the file was
                                malformed. Error: ${e}`);
401                     }
402                 }
403             }
404         });
405         reader.readAsText(file);
406     }
407 }
408
409 function saveDataToLocalStorage() {
410     const json = JSON.stringify(collectGraphData(nodeData, edgeData));
411     const params = JSON.stringify(gatherParams());
412     localStorage.setItem("score", json)
413     localStorage.setItem("params", params)
414 }
415
416 function showStartingNode() {
417     if(typeof starting_node_id === 'string') {
418         network.selectNodes([starting_node_id], false);
419     } else {
420         alert('No starting node selected yet!');
421     }
422 }
423
424 function setStartingNode() {
425     var selected = network.getSelectedNodes();
426     if(selected.length > 1) {
427         alert('Only select one node!');
428     } else if(selected.length === 0) {
429         alert('Select a node first!');
430     } else {
431         starting_node_id = selected[0];
432     }
433 }
434
435 function fetchInterpretation(params, format) {
436     var jsonRequest = JSON.stringify({
437         graph: collectGraphData(nodeData, edgeData),
438         params: params
439     });
440
441     var myHeaders = new Headers();
442     myHeaders.set('Content-Type', 'application/json');
443
444     var myInit = {
445         method: 'POST',
446         headers: myHeaders,
447         mode: 'cors',
448         body: jsonRequest
449     };
450
451     var myRequest = new Request(`/interpretation/${format}`, myInit);
452
453     return fetch(myRequest).then(res => res.blob());
454 }
455
456 function gatherParams() {
457     var starting_node_entry = nodeData.get(starting_node_id);
458     if(starting_node_entry !== undefined && starting_node_entry !== null) {
459         var starting_node = {
460             id: starting_node_entry.nodeData.id,
461             music: starting_node_entry.music
462         };
463     } else {
464         var starting_node = null
465     }
466
467     var maxhops = document.getElementById('hop-count').value;
```

```
468        if(maxhops === "" || Number(maxhops) === NaN) {
469            maxhops = null;
470        } else {
471            maxhops = Number(maxhops);
472        }
473
474        var seed = document.getElementById('seed').value;
475        if(seed === "" || Number(seed) === NaN) {
476            seed = null;
477        } else {
478            seed = Number(seed);
479        }
480
481        return {
482            maxhops:  maxhops,
483            starting_node: starting_node,
484            seed: seed
485        };
486    }
487
488    function completeGatherParams() {
489        var p = gatherParams();
490        if(p.starting_node === null) {
491            alert('Set a starting node first!');
492            return null;
493        }
494
495        if(p.maxhops === null) {
496            alert('Set the maximum amount of hops to a valid number');
497            return null;
498        }
499
500        if(p.seed === null) {
501            // TODO auto generate a random one, let the user confirm before
502            alert('Set the seed to a valid number!');
503            return null;
504        }
505
506        return p;
507    }
508
509    function importParams(p) {
510        if(p.starting_node !== null) {
511            starting_node_id = p.starting_node.id;
512        }
513        if(p.seed !== null) {
514            document.getElementById('seed').value = p.seed;
515        }
516        if(p.maxhops !== null) {
517            document.getElementById('hop-count').value = p.maxhops;
518        }
519    }
520
521    function randomSeed() {
522        if(window.crypto) {
523            var array = new Int32Array(1);
524            window.crypto.getRandomValues(array);
525            document.getElementById('seed').value = array[0];
526        }
527    }
528
529    function downloadInterpretation(format) {
530        var params = completeGatherParams();
531        if(params != null) {
532            try {
533                fetchInterpretation(params, format).then(file => {
534                    var url = URL.createObjectURL(file);
535                    download(url, `export.${format}`);
536                    URL.revokeObjectURL(url);
537                });
538            } catch(e) {
539                alert('An error occured while contacting the API: ' + e);
540            }
541        }
542    }
543
544    function reloadPlayer() {
545        var params = completeGatherParams();
546        if(params !== null) {
```

```
547          if(document.getElementById('player').src) {
548              URL.revokeObjectURL(document.getElementById('player').src);
549          }
550
551          document.getElementById('player').src = null;
552
553          try {
554              fetchInterpretation(params, 'wav').then(file => {
555                  var url = URL.createObjectURL(file);
556                  document.getElementById('player').src = url;
557              });
558          } catch(e) {
559              alert('An error occured while contacting the API: ' + e);
560          }
561      }
562 }
563
564 function init() {
565      var container = document.getElementById('network');
566
567      var options = {
568          manipulation: {
569              addNode: function(nodeData, callback) {
570                  document.getElementById('node-operation').innerHTML = 'Add';
571                  genericEditNode(nodeData, callback);
572              },
573              addEdge: function(edgeData, callback) {
574                  document.getElementById('edge-operation').innerHTML = 'Add';
575                  genericEditEdge(edgeData, callback);
576              },
577              editNode: function(nodeData, callback) {
578                  document.getElementById('node-operation').innerHTML = 'Edit';
579                  genericEditNode(nodeData, callback);
580              },
581              editEdge: {
582                  editWithoutDrag: function(edgeData, callback) {
583                      document.getElementById('edge-operation').innerHTML = 'Edit';
584                      genericEditEdge(edgeData, callback);
585                  }
586              },
587              deleteNode: deleteFromMap,
588              deleteEdge: deleteFromMap,
589              controlNodeStyle: {
590              }
591          },
592          nodes: {
593              borderWidth: 0,
594              color: {
595                  background: '#563d7c',
596                  hover: {
597                      background: '#8f14ff'
598                  },
599                  highlight: {
600                      background: '#8f14ff'
601                  }
602              },
603              chosen: true,
604              font: {
605                  color: 'white',
606                  size: 20,
607                  align: 'center'
608              },
609              shape: 'circle',
610          },
611          edges: {
612              arrows: {
613                  to: { enabled: true }
614              },
615              color: {
616                  color: '#563d7c',
617                  hover: '#563d7c',
618                  highlight: '#563d7c',
619              },
620              font: {
621                  color: '#ffffff',
622                  strokeWidth: 0
623              }
624          }
625      };
```

```
626
627        network = new vis.Network(container, {}, options);
628
629        try {
630            const score = localStorage.getItem('score');
631            if(score !== null) {
632                importGraphData(JSON.parse(score));
633            }
634        } catch(e) {
635            localStorage.removeItem('score');
636        }
637
638        try {
639            const params = localStorage.getItem('params')
640            if(params !== null) {
641                importParams(JSON.parse(params));
642            }
643        } catch(e) {
644            localStorage.removeItem('params');
645        }
646
647        const pitch_selector = valid_pitches.map((p, i) =>
648            `<option value="${p}">${display_pitches[i]}</option>`)
649            .reduce((acc, v) =>
650                acc + v, '');
651        document.getElementById('pitch').innerHTML = pitch_selector;
652
653        /* event handling, order as in sidebar */
654        document.getElementById('set-starting-node').onclick = setStartingNode;
655        document.getElementById('show-starting-node').onclick = showStartingNode;
656
657        document.getElementById('random-seed').onclick = randomSeed;
658
659        document.getElementById('reload-player').onclick = reloadPlayer;
660        document.getElementById('download-audio').onclick = () => {
661            var format = document.getElementById('format').value;
662            downloadInterpretation(format);
663        };
664
665        document.getElementById('gen-score').onclick = () =>
666            downloadFile('application/json', 'score.likely.json',
667                JSON.stringify(collectGraphData(nodeData, edgeData)));
668        document.getElementById('upload-score').addEventListener('change',handleImport);
669        document.getElementById('clear-score').onclick = () =>
670            importGraphData({ nodes: [], edges: []});
671
672        window.setInterval(saveDataToLocalStorage, 5000);
673 }
674
675 document.addEventListener('DOMContentLoaded', () => init());
```

Graph im JSON Format der Webapplikation

```json
1  {
2    "nodes": [
3      {
4        "id": "8639d9e3-570d-47e1-b18b-0389cfd36693",
5        "music": {
6          "dur": {
7            "numerator": 1,
8            "denominator": 4
9          },
10         "pitch": "C",
11         "octave": 1
12       }
13     },
14     {
15       "id": "0073dfd0-1d9c-49ac-b59c-db8282fd7fe2",
16       "music": {
17         "dur": {
18           "numerator": 1,
19           "denominator": 4
20         },
21         "pitch": "D",
22         "octave": 1
23       }
24     },
25     {
26       "id": "67f468d4-6d6a-4e78-b003-9dd9ebc21558",
27       "music": {
28         "dur": {
29           "numerator": 1,
30           "denominator": 4
31         },
32         "pitch": "E",
33         "octave": 1
34       }
35     },
36     {
37       "id": "569ed8c7-f6d1-427a-bf32-e2378f1fc56d",
38       "music": {
39         "dur": {
40           "numerator": 1,
41           "denominator": 4
42         },
43         "pitch": "F",
44         "octave": 1
45       }
46     },
47     {
48       "id": "e9b032b8-a7bf-4b6b-aab8-f87c03651a1c",
49       "music": {
50         "dur": {
51           "numerator": 1,
52           "denominator": 4
53         },
54         "pitch": "G",
55         "octave": 1
56       }
57     },
58     {
59       "id": "3558118c-0872-49dd-ac60-72c12603e1bd",
60       "music": {
61         "dur": {
62           "numerator": 1,
63           "denominator": 4
64         },
65         "pitch": "A",
66         "octave": 1
67       }
68     },
69     {
70       "id": "6a58ff77-d7f5-439a-b220-a68ae8f70e8a",
71       "music": {
72         "dur": {
73           "numerator": 1,
74           "denominator": 4
75         },
76         "pitch": "B",
77         "octave": 1
78       }
```

```
 79           }
 80        ],
 81        "edges": [
 82           {
 83              "id": "dbc6dc78-f0b9-47eb-bae1-ab4936c3839c",
 84              "from": "3558118c-0872-49dd-ac60-72c12603e1bd",
 85              "to": "8639d9e3-570d-47e1-b18b-0389cfd36693",
 86              "prob": 0.3
 87           },
 88           {
 89              "id": "a6e82604-6021-4d72-b85b-c33fb097d549",
 90              "from": "e9b032b8-a7bf-4b6b-aab8-f87c03651a1c",
 91              "to": "3558118c-0872-49dd-ac60-72c12603e1bd",
 92              "prob": 0.7
 93           },
 94           {
 95              "id": "fceba52d-2381-4760-83bb-893428bfb3ba",
 96              "from": "569ed8c7-f6d1-427a-bf32-e2378f1fc56d",
 97              "to": "e9b032b8-a7bf-4b6b-aab8-f87c03651a1c",
 98              "prob": 0.7
 99           },
100           {
101              "id": "8f8677c3-1d8b-45b2-8170-9b9dce55af5b",
102              "from": "e9b032b8-a7bf-4b6b-aab8-f87c03651a1c",
103              "to": "8639d9e3-570d-47e1-b18b-0389cfd36693",
104              "prob": 0.3
105           },
106           {
107              "id": "85e95c65-41fc-447f-be0d-7a573fe49ca8",
108              "from": "67f468d4-6d6a-4e78-b003-9dd9ebc21558",
109              "to": "8639d9e3-570d-47e1-b18b-0389cfd36693",
110              "prob": 0.3
111           },
112           {
113              "id": "3923f713-5ccd-4d31-9f4a-977376129018",
114              "from": "0073dfd0-1d9c-49ac-b59c-db8282fd7fe2",
115              "to": "67f468d4-6d6a-4e78-b003-9dd9ebc21558",
116              "prob": 0.7
117           },
118           {
119              "id": "ceb83b3d-7fbc-4c8c-9274-be54e3e0694f",
120              "from": "67f468d4-6d6a-4e78-b003-9dd9ebc21558",
121              "to": "569ed8c7-f6d1-427a-bf32-e2378f1fc56d",
122              "prob": 0.7
123           },
124           {
125              "id": "3b157032-e4e3-4972-8602-37bf0d4fe97c",
126              "from": "3558118c-0872-49dd-ac60-72c12603e1bd",
127              "to": "6a58ff77-d7f5-439a-b220-a68ae8f70e8a",
128              "prob": 0.7
129           },
130           {
131              "id": "8e9d7cdd-5bf9-4152-b496-ac9e7e7e13ef",
132              "from": "569ed8c7-f6d1-427a-bf32-e2378f1fc56d",
133              "to": "8639d9e3-570d-47e1-b18b-0389cfd36693",
134              "prob": 0.3
135           },
136           {
137              "id": "92196538-8307-49a6-b086-7569b071de13",
138              "from": "0073dfd0-1d9c-49ac-b59c-db8282fd7fe2",
139              "to": "8639d9e3-570d-47e1-b18b-0389cfd36693",
140              "prob": 0.3
141           },
142           {
143              "id": "14fcf488-3af0-4f6a-95af-7005543cfa04",
144              "from": "8639d9e3-570d-47e1-b18b-0389cfd36693",
145              "to": "0073dfd0-1d9c-49ac-b59c-db8282fd7fe2",
146              "prob": 1
147           }
148        ]
149     }
```

Lizenz

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

   You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

   (a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

   (b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

   (c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

   (d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

   A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

   You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

   (a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

   (b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

   (c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

   (d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

   (e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

(a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

(b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

(c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

(d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

(e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

(f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

   You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

   However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

   Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

   Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

   You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

    Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

    An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

    You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Remote Network Interaction; Use with the GNU General Public License.

Notwithstanding any other provision of this License, if you modify the Program, your modified version must prominently offer all users interacting with it remotely through a computer

network (if your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software. This Corresponding Source shall include the Corresponding Source for any work covered by version 3 of the GNU General Public License that is incorporated pursuant to the following paragraph.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the work with which it is combined will remain governed by version 3 of the GNU General Public License.

14. Revised Versions of this License.

    The Free Software Foundation may publish revised and/or new versions of the GNU Affero General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

    Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU Affero General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU Affero General Public License, you may choose any version ever published by the Free Software Foundation.

    If the Program specifies that a proxy can decide which future versions of the GNU Affero General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

    Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

    THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

    IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

    If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <textyear>  <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU Affero General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License
along with this program.  If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If your software can interact with users remotely through a computer network, you should also make sure that it provides a way for users to get its source. For example, if your program is a web application, its interface could display a "Source" link that leads users to an archive of the code. There are many ways you could offer source, and different solutions will be better for different programs; see section 13 for the specific requirements.

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU AGPL, see `http://www.gnu.org/licenses/`.