The Minuet Handbook

**Sandro S. Andrade**

Contents

Abstract

Minuet is an application for music education. It features a set of ear training exercises regarding intervals, chords, and scales.

Chapter 1

Introduction

Welcome to Minuet: the KDE software for music education. Minuet aims at supporting students and teachers in many aspects of music education, such as ear training, first-sight reading, solfa, scales, rhythm, harmony, and improvisation. Minuet makes extensive use of MIDI capabilities to provide a full-fledged set of features regarding volume, tempo, and pitch changes, which makes Minuet a valuable tool for both novice and experienced musicians.

Minuet features a rich set of ear training's exercises and new ones can be seamlessly added in order to extend its functionalities and adapt it to several music education contexts.
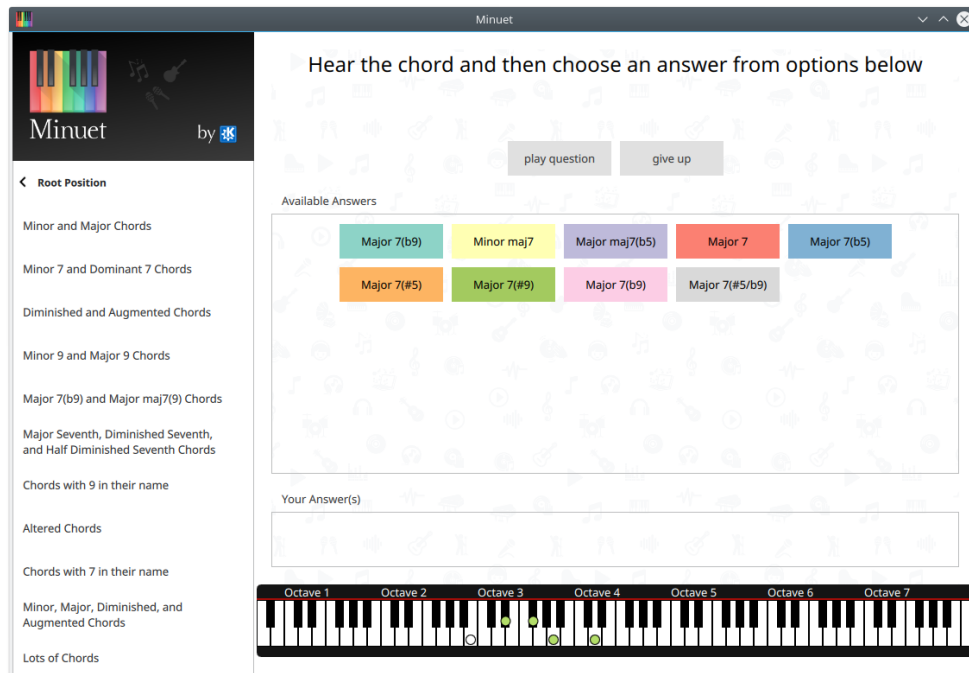
Minuet's ear training chord exercises

Chapter 2

Using Minuet

In the next two sections - Starting Minuet and Minuet Exercises - we will provide you the required steps to get Minuet up and running.

2.1 Starting Minuet

You can start Minuet from the application launcher. Open the KDE program menu by clicking on the application launcher icon on the toolbar at the bottom left of your screen. This will raise a menu. Move your cursor up the menu to the **Applications** → **Education** → **Miscellaneous** → **Minuet (Music Education Software)** menu item.

2.2 Minuet Exercises and Workflow

Minuet's user interface entails three major components:
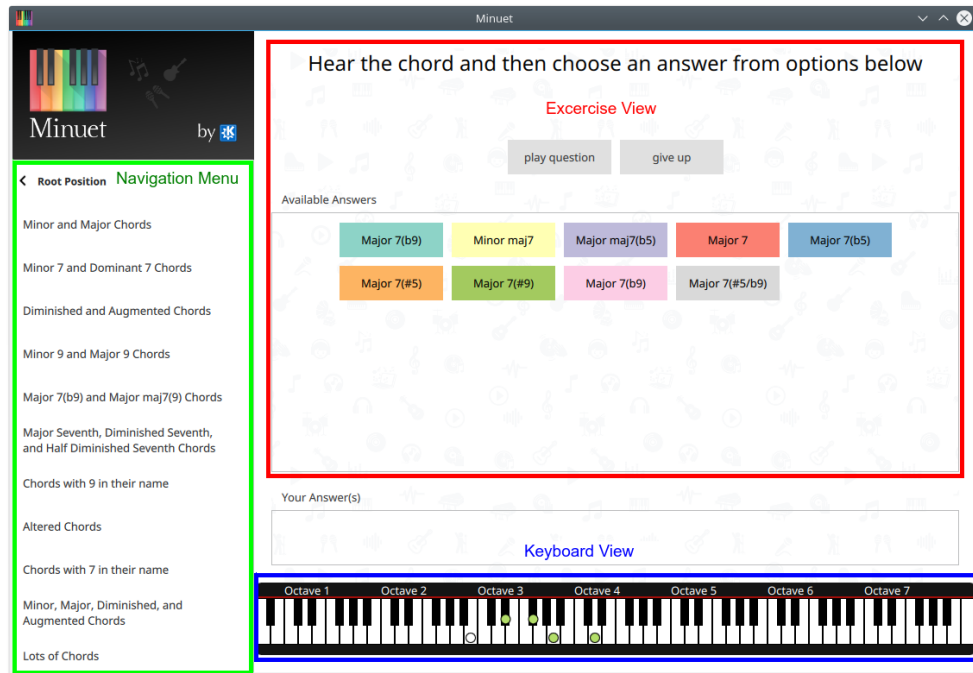
Minuet's UI components

Navigation Menu

Allows for navigating in Minuet's exercise categories and selecting a particular exercise. The Navigation Menu is dynamically created based upon exercises specification files as described in Creating Exercises. Minuet's exercises are grouped according to classes such as intervals, scales, and chords.

Keyboard View

Exhibits MIDI `note on` events being sequenced by a MIDI file or by an exercise execution.

Exercise View

Presents, for a given exercise, buttons for controlling exercise presentation and a set of exercise's possible answers. An exercise run begins by clicking the `new question` button. A randomly selected interval/chord/scale is played by Minuet and then the student is expected to pick up an answer among the ones shown in the answer's grid. Minuet always present - as a white small circle in Keyboard View - the first note of selected interval/scale or the root note of selected chord. By hovering a possible answer, Minuet highlights its corresponding visual representation in Keyboard View. The remaining notes of the hovered answer are presented as small circles whose colors are the same of hovered answer's button. The student can hear the exercise again by clicking the `play question` button, click the chosen answer button, or get the right answer by clicking the `give up` button. A new run of the same exercise can be started by clicking again the `new question` button.

Chapter 3

Creating new Minuet's exercises

Minuet's exercises are defined in exercise specification files, written in JSON format:

```json
{
  "exercises": [
    {
      "name": "Intervals",
      "root": "21..104",
      "playMode": "scale",
      "children": [
        {
          "name": "Ascending Melodic Intervals",
          "children": [
            {
              "name": "Seconds",
              "options": [
                {
                  "name": "Minor Second",
                  "sequenceFromRoot": "1"
                },
                {
                  "name": "Major Second",
                  "sequenceFromRoot": "2"
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

Minuet's exercise specification files contain one top-level JSON object featuring the *exercises* array. Such an array defines a hierarchical structure of exercises, grouped by categories. Every category/exercise has a name. Category JSON objects contain a property named *children*, which describes the subcategories/exercises entailed by such a category. Exercise JSON objects contain a property named *options*, which defines the possible answers for such an exercise. In each exercise run, Minuet randomly selects one answer among the possible ones and the student is expected to click the answer's button which corresponds to the selected answer.

Any (sub)category may define a *root* parameter to specify the range from which the initial interval/chord/scale's note will be randomly chosen for all exercises in this category. Such range

corresponds to standards MIDI note numbers and follows the format `<min-value>..<max-value>`. The example presented above uses all keyboard range as possible root notes (21..104). The `playMode` parameter indicates how possible answers should be played: as a `scale` (one note after the other) or as a `chord` (all notes ringing out simultaneously).

Each exercise's option defines a name and the sequence of notes which should be played from the root note randomly selected in each exercise run. Such sequence of notes is defined as relative distances from the root note, describing the interval each note forms in conjunction with the root note. For example, for a major scale, the sequence of notes is ˝2 4 5 7 9 11 12˝, which respectively denotes the ˝whole whole half whole whole whole half˝ major scale structure. The `sequenceFromRoot` parameter may contain any notes in length. Also, Minuet's core ensures that only answers whose all notes lies within keyboard range are randomly selected.

To provide a better infrastructure for organizing a large set of exercise specification files, Minuet's core supports the use of several specification files, which are automatically merged to compose the final exercise hierarchy presented in the Navigation Menu. Exercises are correctly merged as long as different specification files use the same (sub)category name when defining exercises. For now, Minuet's provides no GUI for creating exercise specifications so that you must manually create such JSON files. Minuet's exercise specification files may be installed system-wide or locally in the `minuet/exercises/` directory located in **qtpaths --path GenericDataLocation**

Chapter 4

Credits and License

Minuet

Program copyright 2016 Sandro Andrade sandroandrade@kde.org

Documentation Copyright (c) 2016 Sandro Andrade sandroandrade@kde.org

This documentation is licensed under the terms of the GNU Free Documentation License.
This program is licensed under the terms of the GNU General Public License.