

DENZO GUITAR SOFTWARE III

© DenzoSoft 1999

DENZO GUITAR SOFTWARE is not just another scale/chord engine. You probably have seen others before, but you have to ask yourself if those programs have really helped you in using the many styles and modes of notes or chords. In other words what have you really learned?

DENZO GUITAR SOFTWARE is a totally different software. I designed it for my own personal use, but now it has grown into this software program, with so much functionality, that it will be useful for a lot of other guitarists.

Anyhow, please remember that this is not in any shape or form, an attempt at professional software, and some things may appear a little strange, and complex at first sight. In the task section I'll introduce you to a new interface paradigm that characterizes this application.

I believe that this software can change radically your way of playing guitar, by suggesting interesting positions and movements, but one thing to remember about music: its inspirational and personal and no software can make music, only sounds.

1. Introduction

Welcome to DENZO GUITAR SOFTWARE,

This is a new guitar fretboard mapping software: a program that tries to reproduce, on a guitar fretboard, chords, scales, arpeggios, etc. Unlike other software, it doesn't rely on any positional database, so it tries to find all positions using your own parameters.

The main problem to this approach is the possibility of finding impossible positions: configurations of notes on fretboard that no-one can play. DENZO GUITAR SOFTWARE uses some very complex algorithms to avoid these problems, and in doing so, it can find some really novel positions that you can introduce in your music.

The great adaptability of this program means, the more experience gathered by you the user, step by step, you'll learn the basics of music theory, and be able to truly understand the real power of this software.

You cannot find 70% of DENZO GUITAR SOFTWARE features in other programs. This is its greatness!

?? Who Needs DENZO GUITAR SOFTWARE III

Every guitarist can use this software to find a lot of new ideas: the beginner will discover basic positions, and the experienced musician, may learn new ways to link chords or arpeggios.

The idea is always the same: you have to know what you're looking for, and this software, will map this in the best way it knows on the fretboard.

The starting point is always music; all you have to know is the principles of linking notes. And this software will help you to interpret your ideas with guitar, in a better way.

?? Music Theory

Music theory is necessary, but not essential, to use this software in the right way. Music theory will be mentioned during this documentation, so some music knowledge will be needed. Try to pay attention if it all seems new to you, the information is vital if you wish to use this software to it's full potential.

Every riff, and every song, is built using these 3 objects, and if you know how to use them properly, you can play practically everything. Obviously there are so many different positions that you would probably never use all of them, but you can find, just about any position you fancy!

This program will help give you new ideas, but then you have to learn to play them on an instrument.

1. Tones

In music there are 12 notes or tones:

A, A#, B, C, C#, D, D#, E, F, F#, G, G#,

There are some enharmonic substitutions for altered tones:

A# = Bb, C# = Db, D# = Eb, F# = Gb, G# = Ab

Therefore, the 12 notes or notes would look like:

A, Bb, B, C, Db, D, Eb, E, F, Gb, G, Ab,

A # increase a tone of an half step

A b decrease a tone of an half step

NB: DENZO GUITAR SOFTWARE uses only # as the enharmonic alteration, to simplify things, so you have to remember this using the software.

?? Intervals

An interval is the distance between 2 notes:

2nd Minor/Major

3rd Minor/Major

4th Perfect/Augmented

5th Diminished/Perfect/Augmented

6th Minor/Major

7th Minor/Major

In particular:

0 unison

1 minor 2nd

2 Major 2nd

3 Augmented 2nd; minor 3rd

4 Major 3rd

5 Perfect 4th

6 Augmented 4th; Diminished 5th

7 Perfect 5th

8 Augmented 5th; minor 6th

9 Major 6th

10 minor 7th

11 Major 7th

12 octave

13 minor 9th

14 Major 9th

15 Augmented 9th

16 Major 10th

17 Perfect 11th

18 Augmented 11th

19 Perfect 12th

20 Major 13th

The naming convention is as follows:

When a major interval is decreased by a half step, it becomes minor.

When a major interval is increased by a half step, it becomes augmented.

When a minor interval is decreased by a half step, it becomes diminished.

When a minor interval is increased by a half step, it becomes major.

When a perfect interval is decreased by a half step, it becomes diminished.

When a perfect interval is increased by a half step, it becomes augmented.

Example:

(We will use C as root)

C-A: 6th or 13th

C-F#: augmented 4th or diminished 5th (C-Gb)

You have to memorize all this convention, and then practice learning to find these intervals on your guitar fretboard automatically. DGT will help you in this work!

?? **Whats a chord**

The first brick of music that we see is the chord: a chord is formed by 3 or more notes played at the same time. Some consider chords 2 note structures (a bichord), but for the purists its 3. Anyhow the main idea is

that in a chord, some notes are played contemporary. This implies that you have to find a lot of notes in different strings of your guitar and in the same time you have to press the selected fret.

Normally this isn't simple, which means you'll never be able to play the same chord structure used by a pianist, because the mapping process is different, due to the different note positions.

1. Formula

A chord is defined by its degrees: for example a major chord contains a root, a major third and a fifth, so for C major you have C E G.

If you change only the root you have another major chord. So its possible to define a chord with a root, and a pattern or formula by its degrees.

DENZO GUITAR SOFTWARE defines this formula using relative distance in half-steps, so a major chord formula is 043: so 0 is the root (C), 4 is the major third (E) 4 half steps after the root, and 3 is the fifth (G) 3 half steps after the major third.

In this way you can define any interval and chord!

Example:

A min7 chord is defined as 0343

?? Inversions

When you have a chord you have only a group of notes, and you have to put them on your fretboard. Normally a guitarist doesn't respect the perfect order of tones that he obtains from a harmonization, because he could obtain impossible positions. So he tries to rearrange the notes in the best way (musically or physically) he can.

Sometimes root note isn't the lowest note of a chord: this is called an inversion chord.

Theoretically we have an inversion when the lowest note is raised an octave:

C(1)E(1)G(1) -> E(1)G(1)C(2)

In parenthesis you have the octave!

?? Slash Chord

Normally a chord is played with the root as the lowest note because in this way its the most important note in the chord and defines its root as the main characteristic.

Sometimes for melodic purposes, or a walking bass riff, you may prefer to play another tone other than the root: this is a slash chord!

Example:

C/E: in a C major chord with E as the lowest note

Cmin/A: is a Cmin6 chord with A as the lowest note

?? Chords Description

A guitar chord is described using different conventions, but the information are always the same:

Text format: D m9 [10 12 10 10 10 12]

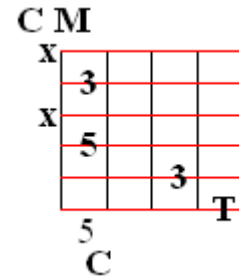
Where each number represents the fret of its own string (From low to high)

And a graphical format:

Which is described as the chord frame!

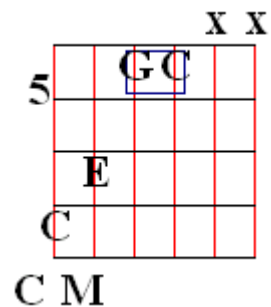
High string is the first from top!

NB Here you can see degrees (as above), notes, or balls.



Its also possible to use a vertical representation:

Bass string is the first from left!



?? Whats a scale

A scale is a sequence of notes according to a given interval structure. The major scale is built by a root, a note a step after the root, a note 2 steps after the root, etc. So when you change the root you have a new major scale.

The main difference between scales and chords, is that in a scale, notes are played sequentially and not at the same time.

Normally scales have 6, 7 or 8 notes!

1. Degrees

For a seven note scale it is very important to learn the name and specific quality of each degree, because when you master this, you have another instrument to give a sense of feeling to your music.

For a C major scale we have:

C tonic

D sopratonica

E mediante

F subdominat

G dominant

A sopradominante

B sensible

?? Modes

Modes for a scale, are the same thing as inversions, are for chords: you have to maintain the same order, but can start with a different tone. So, if you used the notes used in a C major scale, but started at D, you would have a D doric scale: D E F G A B C.

Remember that harmonization of modes is the same as the original scale, but you have to 'underline' that D is the root and not a second degree of a C major scale so you have to use a specific melody and a good harmony: for example an A dominate before the root chord Dmin.

?? **Harmonization**

If you choose some notes from a scale you can obtain chords, that together, describe the same quality of the original scale. This is the principle of harmonization.

In western music there is basically one method of harmonization, known as harmonization per third: in this process you harmonize a scale choosing a root and selecting a sequence of notes at a distance of a third each other.

For example:

If you have a C major scale: C D E F G A B

You'll find:

C

D **E** F **G** A B = C

C **D** E **F** G **A** B = Dmin

C D **E** F **G** A **B** = Emin

In this case we harmonize a C major scale per third, and with 3 notes for 3 different chords.

Its possible to harmonize, with more or less notes for chords.

Its also possible to harmonize a scale per fourths or seconds, (the other intervals are only inversions of these and don't produce nothing more that duplicates!).

In modal jazz and classical music these different types of harmonizations are very frequent.

Its also possible to look for all types of chords, that harmonize with the same pattern of degrees as a scale: for example you may want all 4/6 chords of C major scale:

C 4/6 (C E F G A)

Dmin4/6 (D F G A B)

Emin4/6b (E G A B C)

This is an harmonization by mask, and DENZO GUITAR SOFTWARE allows it very simply.

Another type of harmonization is obtained, by looking for all chords contained in a scale, even if not produced by overlapping thirds. For example C6 (C E G A) couldn't be obtained per third, but its contained in C major scale and so it could be a harmonization of first degree of that scale.

?? **Whats an arpeggio**

An arpeggio is the scalisation of a chord: all the notes of the chord, are played one by one sequentially. In other words, its a little scale or a subscale.

The approach for arpeggios is very important, because you can obtain a sequence of positions, very hard to play normally, plus lots of possibilities. Its also very important to practice, and try and try again.

?? Guitar Approach

When you know all there is to know about music theory you're not yet a guitarist, you have also to learn how to physically play what you have studied.

This process could be really frustrating and many times you find pieces of music really difficult to play. Sometimes you'll find another way of playing it differently, or you'll just play a different song altogether.

This software is designed to help you in this process of searching for the way to play something, and you'll see that it will avoid you doing a lot of useless work!

?? New Version

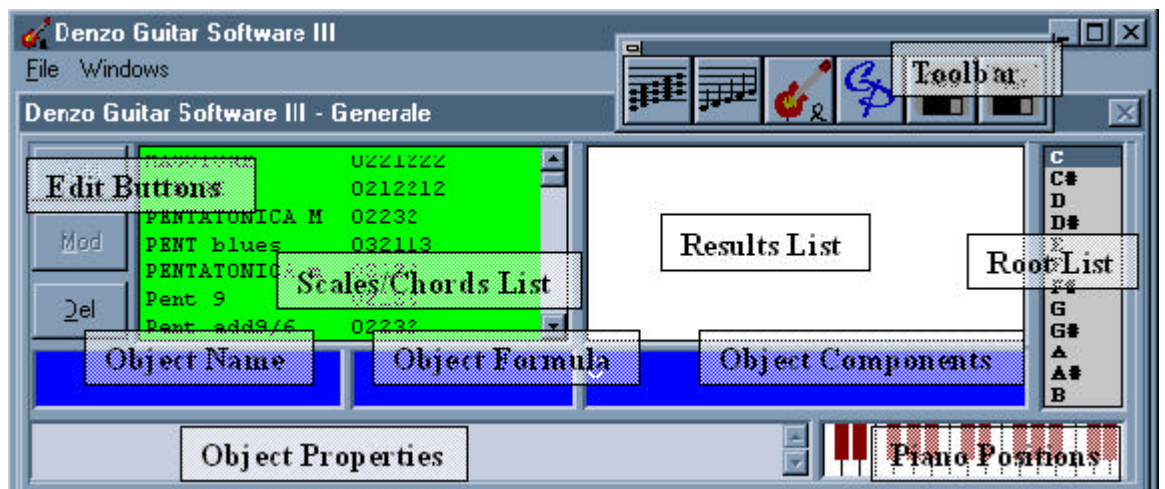
The new version of DENZO GUITAR SOFTWARE has a totally new interface and a lot of new functions for mapping on guitar. This would probably look like a whole new application for the ones who know Version 2.0

1. Main Window

DENZO GUITAR SOFTWARE starts with an MDI window that contains the main (Generale) window.

Normally you use this window to calculate Objects that you want to map on fretboard, or to analyze relationships, between scales and chord harmonization.

All operations are recorded with their results in the journals, and you can see them whenever you want.



This window is the interface to a lot of general music related functions, that can be accessed using the toolbar or context menus.

To understand this window properly, you have to learn the differences between a 'musical object', or simply 'OBJECT' for DENZO GUITAR SOFTWARE.

Using the toolbar you can choose alternatively, from the scales list/chords list, a scale/chord by selecting an object, then select a 'root' from the root list so that, finally, you can apply them together as a function.

Every function that produces results, will be written in the Results List so that it can be cleaned, memorised, exported, sorted, etc. using context menu.

NB: Every graphical object has a defined context menu, that normally can be shown using a right mouse click, sometimes you have to use right click (as in blue boxes).

1. Edit Buttons

Add, Mod, Del buttons, allows user to modify chords and scales lists.

Add

: add current object (defined by current value of Object Name and Object Formula) in the position selected in the visible list

Mod

: substitute the selected element in list with value of current object (defined by current value of Object Name and Object Formula)

Del

: delete selected element in visible list

NB: after this, you have to save new lists using **Save Chords/Scales Information** in the toolbar, so the next time you start the program you'll see your modification.

Scales/Chords list are loaded from denzogtr.INI file at startup.

?? Main Toolbar

This toolbar is used to access the most frequently used functions of the main window.



First button shows chords list

Second button shows scales list

Third button activate guitar window with current object

Fourth button exports chords in Chord Play format

Fifth button save Chords/Scale Information

Sixth button save journals so you can see them at a later date

?? Scales/Chords Lists

These lists contain all defined Object: you can select an item from them and then elaborate on it, or you can modify lists using the edit buttons.

A single left click on an element of a list defines that element as Selected Object.

A double left click on an element of a list defines that element as Current Object.

A right click defines that element as Current Object and show context menu.

?? Results List

In this list you can collect all results of your elaboration to use them in other operations.

This list is filled with functions and can be handled using its context menu.

If DENZO GUITAR SOFTWARE has a lot of objects to show in this list its possible that it will use small fonts.

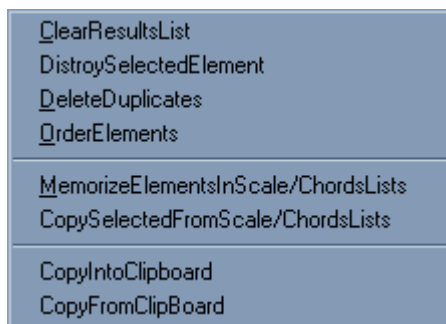
With a single click over an item in the Results List it becomes the Selected Object, with a double click it becomes the Current Object and the blue boxes are filled with its values. At this point you can work on it as usual Current Object.

Its very important to keep good administration of Results List content. Some functions that produce a lot of results clear automatically from the list, others not, so its important to remember to clear it.

At the beginning of an elaboration of a function, write a line to describe its actions. These lines mustn't be selected because they are irregular objects.

1. Results List Context Menu

As in other cases, a right click shows the context menu. This menu allows user to elaborate on the Results List, ordinate or reduce.



ClearResultsList

: delete all items in the list

DestroySelectedElement

: delete selected item

DeleteDuplicates

: delete duplicate items

OrderElements

: order elements alphabetically

MemoriseElementsInScale/ChordsLists

: look for all items in the Results List that are not in Scales/Chords List (the visible one!) and automatically add them.

CopySelectedFromScale/ChordsLists

: put selected item in Scales/Chords list (The visible one!) in Results List.

CopyIntoClipboard

: memorize all items in Results list in the clipboard so they can be used in group elaborations or to paste outside.

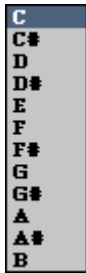
CopyFromClipboard

: this feature is useful to import from outside programs.

?? Root List

This list shows all roots.

After you have selected an object, you have to define its root, so DENZO GUITAR SOFTWARE can elaborate on its composition.



?? Objects

Objects are the simplest structure used by DENZO GUITAR SOFTWARE to organize notes.

A note, a chord and a scale are all Objects and DENZO GUITAR SOFTWARE Main window only functions to manipulate Objects.

An object defines a structure and with root information allows the window to represent every musical brick!

Fundamentally Objects are divided in 2 categories:

The first collects Objects formed by note played in sequence (Scales)

The second collects Object formed by note played together (Chords)

But you can simply decide to mix these 2 groups: this division is only useful for users who know where to find or how to insert some objects.

Another possible classification regards properties of containers (scales) of contained objects (Chord) using a harmonization view.

1. Selected Object

With a single click on Scale/Chords List, you select an Object (**Selected Object**) and, Object Components shows you which are its notes, while Piano Positions shows you the fingerings of this Object on a keyboard.

?? Current Object

After you have selected an Object and see its components you may want to apply a function. So if you double-click an item in Scales/Chords List this Objects becomes the current object and you can see its name and formula in the blue box below the list.

NB: Remember that functions always use current object as first parameter.

When you have a current object you can handle it with functions.

?? Object Boxes

All characteristics of Current Object are stored in the Object Boxes.

The first contains Object name, then Object formula and finally Object components.



All other characteristics are stored in the Object properties box.

?? Object Context Menu

Most of the functions of this window are accessible using the Object Context Menu, that you can activate with a right click of mouse, when pointer is on Chords/Scales list, or with a right click when pointer is on one of the Object definition box (blue box).

Using this menu you are able to start functions that save their results in the Results List.



Object Context Menu is divided in 3 main groups of functions:

Harmonization

HarmonicUtilities

Play

1. Harmonization

This drop-down menu shows the Harmonization window that can be used to harmonize Current Object.



Using this window you can set parameters of harmonization.

Here you can find 3 different algorithm of harmonization that allow you to analyse, with great detail the relationships from scales to chords.

The results of harmonization are placed in the Results List and can be added to Chords List to further extend library of Objects.

NB: In the harmonic utilities you can find functions for studying relationship from chords to scales.

1. **C-Harmonize**

C-Harmonize is the simplest way to harmonize a scale: an algorithm will try to find all chords with at most **NotesForChords** notes that contains only scale degrees.

This kind of harmonization gives you a lot of new Objects that describe completely your scale.

For instance the C-Harmonization of a C major scale with at most 4 notes is:

A m5#/7 0352

A m7 0343

B m5#/7 0352

B ø 0334

C 6 0432

C maj7 0434

D m6 0342

D m7 0343

E m5#/7 0352

E m7 0343

F 5b/6 0423

F 5b/maj7 0425

F 6 0432

F maj7 0434

G 6 0432

G 7 0433

?? **G-Harmonize**

G-Harmonize is another kind of algorithm: here you're looking for all chords with **NotesForChord** number of notes selected at a distance defined by **HarmonizingInterval** in terms of degrees.

This is the real scale harmonization as defined in harmony books.

The harmonization for thirds (HarmonizingInterval=2) is the standard harmonization, but is also possible to harmonize for seconds or for fourth (all other harmonization produce only inversion of these).

For instance the harmonization for third of a C major scale with 4 notes is:

B \emptyset 0334

A m7 0343

G 7 0433

F maj7 0434

E m7 0343

D m7 0343

C maj7 0434

?? **M-Harmonize**

M-Harmonize is an algorithm that allows you to harmonize an Object with a mask: for instance you want to obtain all 6/9 chords from major scale, you are not interested if they are major or minor chords, but you simply want chords defined by Tonic, Third, Fifth, Sixth and Ninth.

You can do this, by defining the right mask and harmonizing with this mask.

Harm. Mask field defines the mask the algorithm will use: in this mask you don't find 1 (Tonic) because its implicit, but all others degrees. In the process of harmonization **NotesForChord** parameter is used so if you have the mask "357246" and 4 NotesForChords you'll have a normal harmonization in 7th chords (4 notes: 1st, 3rd, 5th, 7th).

Then pushing button in the results list you'll find what you want.

For instance, if you want to harmonize a C major scale with the mask "356427" and searching 5 notes chords you'll have (1st, 3rd, 5th, 6th, 4th):

B m5b6b/4 03212

A m6b/4 03221

G 4/6 04122

F 4#/6 04212

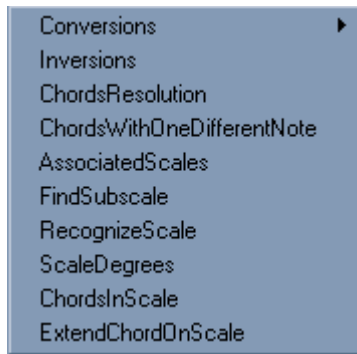
E m6b/4 03221

D m4/6 03222

C 4/6 04122

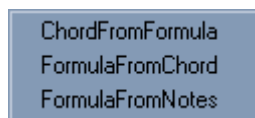
?? **HarmonicUtilities**

This menu allows the user to select some interesting functions of Object elaboration. As with harmonization, these functions work with object and produce other objects or analytical information on them.



1. **Conversions**

These functions allow users to work with Object definitions like Formula, Composition or Name.



You have to enter in the right box (Formulas in the Formula Object box, and so on)

ChordFromFormula

: if you enter a formula the algorithm will try to obtain the chord name

FormulaFromChord

: if you enter ChordName the algorithm calculates its formula

FormulaFormNotes

: if you enter the list of notes the algorithm calculates formula

?? **Inversions**

This function calculates all inversions of Current Object.

For instance if you're looking for all inversions of a C maj7/9:

D no5 sus13 02341

B m5#/add9b/4 01223

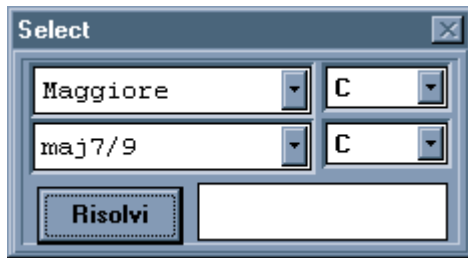
G 4/6 04122

E m6b/7 03412

C maj7/9 02234

?? **ChordsResolution**

This function calculates the resolution chords of a given chord in a specific tonal environment described by a scale.



The window allows you to select a scale and a chord the pressing 'resolve' you can see in the Results List the resolution chords.

For instance a C maj7/9 in a C major tonality will resolve on:

C add9 0223

C add9 0223

Usually this function returns 2 results (sometimes identical).

?? ***ChordsWithOneDifferentNote***

This function calculates all Objects that differ of one note from Current Object and their inversions. This function is useful for searching chords or scales substitutions.

Remember that for a good substitution you have to verify constraints like chords function and leading notes (new chord has to have all the most important note of original).

For instance you can look for substitutions of a C maj7:

A m7 0343

A madd9 0214

A sus9 0253

A#sus5b/add9/6 0243

A#sus5b/add9b/6 0153

A#sus5b/add9b/add 0114

B aug4 0413

B augadd9b 0134

B m5#/4 0323

B m5#/add9b 0125

B sus5#/4 053

B sus5#/4/6 0531

B sus5#/7/4 0532

B sus5#/add9/4 0233

B sus5#/add9b 017

B sus5#/add9b/6 0171
B sus5#/add9b/add 0116
B sus5#/maj7/4 0533
B sus5#/maj7/9b 0173
B sus5b/6b/4 0512
B sus5b/6b/add9b 0152
B sus5b/add9b/4 0141
B sus6b/4 0521
B sus6b/add9b 0161
B sus9b/5# 0172
B susadd9b/4 0142
C 043
C 4 0412
C 4# 0421
C 5b/maj7 0425
C 6 0432
C 6b 0431
C 7 0433
C add9 0223
C add9# 0313
C add9b 0133
C augmaj7 0443
C mmaj7 0344
C sus4#/maj7 0614
C sus6b/maj7 0713
C sus7/maj7 0731
C susmaj7 074
C susmaj7/4 0524
C susmaj7/6 0722
C susmaj7/9 0254

C susmaj7/9b 0164

C#m5bmaj7 0335

C#ø 0334

C#sus5b/7/maj7 0641

D#aug6 0441

D#augadd9b 0134

D#sus5#/add9b/6 0171

E 6b 0431

E add9# 0313

E augadd9# 0314

E m 034

E m4 0322

E m4# 0331

E m5# 035

E m5#/4 0323

E m5#/6 0351

E m5#/7 0352

E m5#/add9 0215

E m5#/add9b 0125

E m5#/maj7 0353

E m5b6b 0332

E m6 0342

E m7 0343

E madd9 0214

E madd9b 0124

E mmaj7 0344

E sus4#/6b 0611

E sus6b 071

E sus6b/4 0521

E sus6b/6 0711

E sus6b/7 0712
E sus6b/add9 0251
E sus6b/add9b 0161
E sus6b/maj7 0713
F sus4#/add9 0241
F sus4#/maj7 0614
F sus5b/maj7/9 0245
F susmaj7/9 0254
F#sus5b/7/4 0514
F#sus5b/add9b/4 0141
F#sus9b/5b 0154
G 4 0412
G 5b/4 0411
G 5b/6 0423
G 6 0432
G aug4 0413
G aug6 0441
G sus4/6 0522
G sus5#/4/6 0531
G sus5b/4/6 0513
G#augadd9# 0314
G#augmaj7 0443
G#m5#/maj7 0353

?? **AssociatedScales**

This function calculates all scale in Scales List that can contain Current Object in their harmonization. This function is really important for improvisation because it presents a great variety of scale that you can use to play over a chord.

You have to remember that the list of scale is the visible list, not the effective scale list

. So you can use this function to find chords that contain Current Object as scales.

For instance if you want to know all scales that you can use to play over a C maj7/9:

A BLUES I 02121112
A BLUES II 0212113
A Dorico 0212221
A dorico -major 02111221
A Eolio 0212212
A Etiope I 0212212
A min_NATURALE 0212212
A MINORE 0212212
A Spagnola II 0212212
B Frigio 0122212
B Locrio 0122122
B napoletana m 0122213
B napoletana m II 012221
B phrigio -major 01211212
B Spagnola 01211122
C Armonic maj 0221213
C BLUES III 022111121
C Ionico 0221222
C Lidio 0222122
C MAGGIORE 0221222
D Dorico 0212221
D dorico -major 02111221
D Etiope II 0221221
D lidio -misolidi 02211121
D Misolidio 0221221
E BLUES I 02121112
E Eolio 0212212
E Etiope I 0212212
E Frigio 0122212
E min_NATURALE 0212212

E MINORE 0212212

E phrigio -major 01211212

E Spagnola II 0212212

E Unghe-gypsy 0213112

F Lidio 0222122

F#Locrio 0122122

F#Spagnola 01211122

G dorico-major 02111221

G Etiope II 0221221

G Ionico 0221222

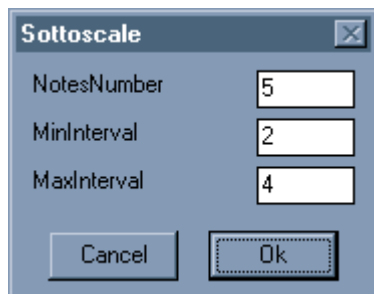
G lidio-misolidi 02211121

G MAGGIORE 0221222

G Misolidio 0221221

?? **FindSubscale**

This function is used to find all subscale contained in Current Object that verify some constraints. For example you may want to know all pentatonics contained in a given scale.



NotesNumber

: defines the number of notes of scale you're looking for.

MinInterval

: defines the minimum number of half steps between two consecutive notes in found subscale

MaxInterval

: defines the maximum number of half steps between two consecutive notes in found subscale.

?? **RecognizeScale**

This function tries to recognize a scale from a given formula and write in the Object Properties box.

If object has less than 7 notes it is described as a chord.

Is automatically activated when you select a new object.

?? **ScaleDegrees**

This function writes in the Object Properties degrees contained in the Current Object.

Is automatically activated when you select a new object.

?? **ChordsInScale**

This function considers the Current Object as a scale and harmonize it using the C-Harmonize algorithm.

?? **ExtendsChordsOnScale**

This function allows user to analyze the relationship between a chord and a scale looking for all chord extensions allowed by selected scale and their inversions.

The scale used is the selected one in the scale list.

This function is very useful in re-harmonization or substitutions of chords.

For instance if you try to extend a C over a C major scale you'll obtain:

A m7 0343

B sus5#/add9b/4 0143

C 043

C 4 0412

C 6 0432

C add9 0223

C maj7 0434

D no5 sus11 0235

E m5# 035

E m5#/4 0323

E m5#/7 0352

E m5#/add9b 0125

E m6b 0341

F susmaj7/9 0254

G no5 4/6 0414

G no5 sus4/6 054

G no5 sus7/4/6 0541

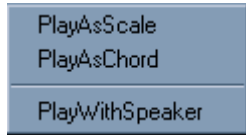
G no5 susadd9/4/6 0234

G sus4/6 0522

?? **Play**

This menu allows user to hear Current Object played.

You can play Object using your sound card or using your speaker.



There are 3 possibilities:

PlayAsScale

: Object is played as scale or arpeggio using sound card

PlayAsChord

: all notes of Object are played together as chord using sound card

PlayWithSpeaker

: Object is played as scale or arpeggio using speaker

NB: its impossible to play chords using speaker

All settings of sound card are described in MIDI Settings window.

?? **Piano Positions**

This box is used to describe on a virtual keyboard all selected objects.

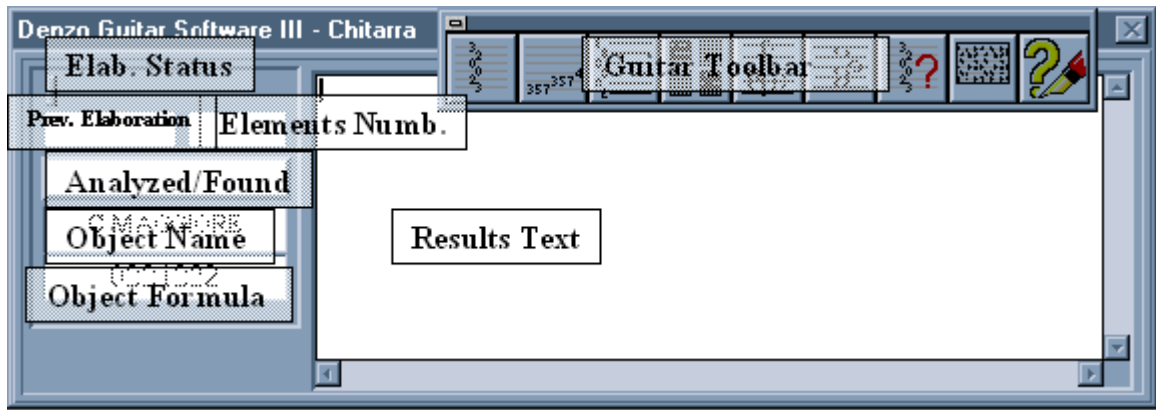
The map process of an Object on a keyboard is very simple but its helpful for whoever is learning guitar and is already able to play piano.



?? **Guitar Window**

This window allows user to map in the better way Objects onto a guitar fretboard. As we said before, Object mapping process isn't simple like other instruments (for instance onto piano), so you have to use very intelligent algorithms to help you.

This window gives you the interface to call all these functions and map all Object calculated in Main Window.



This window is organized so that when you select a Current Object in the main window and press button on main toolbar to show Guitar Window, object characteristics are loaded in Object Name and Object Formula and that is what you want to elaborate.

If you want you can decide to change it directly without returning in the Main Window editing its definitions.

When you're ready for elaboration you have to decide what to do.

You can search for chords or scales positions, defining a new tuning, view graphically archives and print, analyze musics, looking for chords from their notes, chords from their positions, view fretboard positions and build exercises from what you have found.

All these functions are accessed using toolbar.

Using menu you can build harmonic progression using archives.

When you look for chords or scales the results are show in archive lists. You can save them or use as you prefer for other elaborations.

Elab. Status

shows the status of current elaboration.

Prev. Elaboration

shows statistics of previous elaboration (Analyzed/Found)

Elements Numb.

Shows the number of the current archive

Analyzed/Found

gives statistics about current elaboration

Object Name

is the name of Object to elaborate

Object Formula

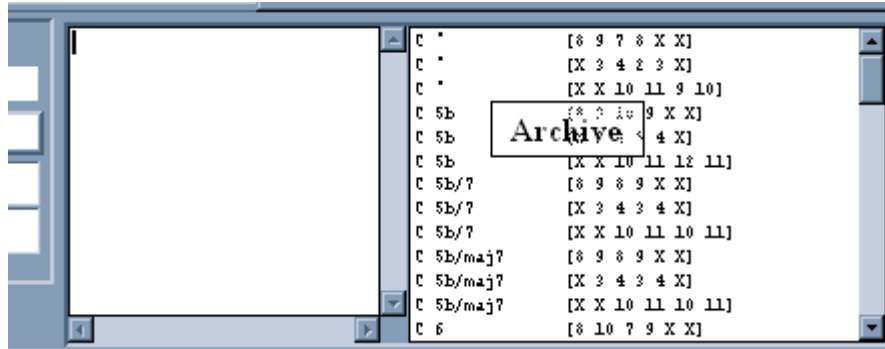
is the formula of Object to elaborate

All functions related to this window are accessible using guitar toolbar, context menu or standard menu.

1. Archives

Every elaboration writes their results in Result Text or in the archive lists.

You can see archives by right-clicking Results Text.



There are 2 main archive types:

temporary archive

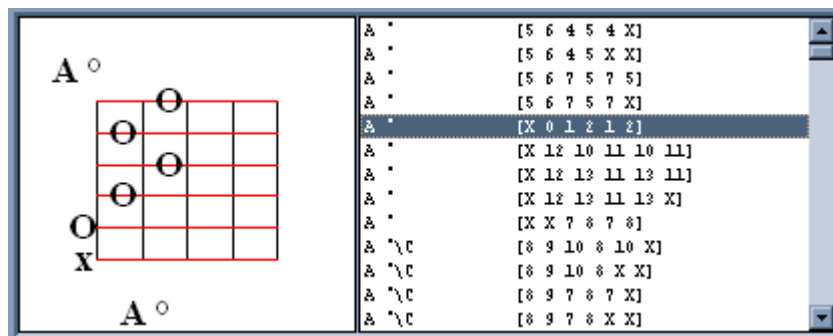
that is used to save some results of current elaboration

saved archive

(or simply archive) that is used to load previous results

You can switch from one to other with the Archive Context Menu. By default, after an elaboration temporary archive is showed.

When you select an element in the list you can see automatically a graphic representation on the left side:



Scales, chords and arpeggios are saved in these lists with a particular format:

Chord Format

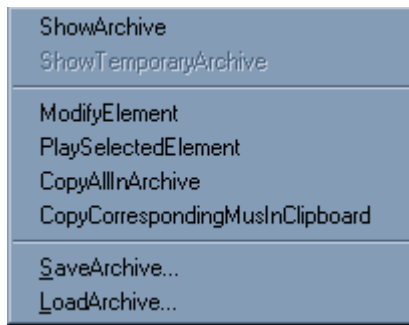
: is represent by number inside [and]. They represent fret from low to high string, so [X 3 2 0 1 0] is the C major normal position.

Scale Format

: to represent scale that normally have more that a fret for each string is used another convention: fret positions are listed sequentially with 2 digit for each and strings are separated by a . . So 040507:030506:030406:020405:030406:020405 is a diminished scale with 4,5,7 fret on the lowest string and 3,5,6 on the higher and so on.

1. Archive Context Menu

This menu is used to access all archive-related functions as: switching between archives, load or save archives, editing archive items and so on.



ShowArchive

: show saved archive, enabled only over temporary archive

ShowTemporaryArchive

: shows temporary archive, enabled only over saved archive

ModifyElement

: activate a window for item editing (only for saved archive!)

PlaySelectedElement

: plays selected item as a chord or scale (auto-recognition)

CopyAllInArchive

: copy all selected items in current saved archive (enabled only in temporary archive)

CopyCorrespondingMusInClipboard

: memorize .Mus description in clipboard for other elaborations

SaveArchive

: for saving current visible archive

LoadArchive

: load in saved archive with or without replacing

?? Guitar Toolbar

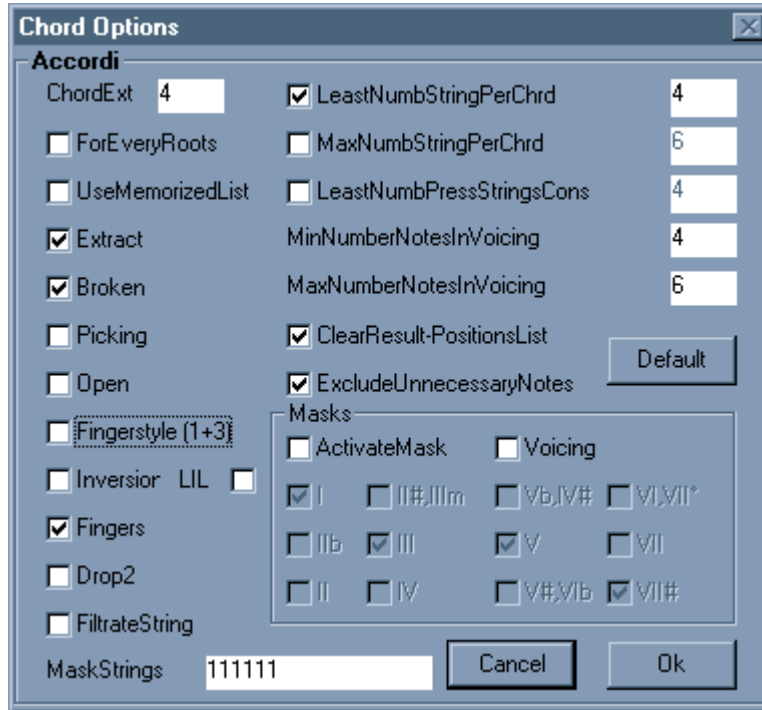
Using toolbar you access the most functions of this window.



You can try to map Current Object as a chord, as a scale, or define a new tuning, view and print graphically archives, see in tablature various music files, study chords by their notes or by their positions, see all possible positions on fretboard and finally build your exercises.

1. Chord Search

Using this window you can set all parameters for chord search. This is a very important window and you have to spend a lot of time to understand all possibilities you have.



This window starts with some default that you can change for your own purpose in searching of chord positions.

ChordExt

: this parameter describes maximum number of fret of chord position you're going to search for.

ForEveryRoot

: using the same formula describing a chord (es. 0344 min7) you can find all positions for all roots with the same settings

UseMemorizedList

: in this case you can select formula to analyze not from Current Object, but from memorized elements (Results context menu in the main window), in this case the same research is repeated for all memorized chords.

Extract

: this setting allows engine to manage those positions that don't select all strings, for instance a position like [X 3 2 0 1 0] contains a not picked string and you can find it only by setting this parameter.

Broken

: this parameter is quite similar to Extract, but is used to define as acceptable, positions that contain not picked strings inside of other picked strings, for instance a position like [0 3 X 0 1 0] is a broken position as [0 3 X X 1 0], but [X 3 2 0 1 X] is only an extract position.

NB: Its impossible to have a Broken position that isn't an extract position, but its possible to have an extract position that isn't a broken position (see example in Extract).

Picking

: this parameter states that you want only positions good for a picking style like funky or reggae. In this case you can have broken position, but X strings must be stopped by some fingers or mustn't be inside other picked strings: [0 3 X 0 1 0] is a pickable position because the X string is stoppable by finger that press 5th string on 3rd fret, but [0 3 2 0 X 0] is not pickable because none can stop X string because it has ove an open string, this is a simply broken position.

Open

: this parameter tells engine to not consider open string in calculation of fret extension so you can find positions everywhere in your fretboard that use open string, for instance [0 7 5 0 0 0] is a good open position for E min.

NB: Pay attention using this parameter because it can produce a very large number of solutions, but very good for New Age and acoustic styles.

FingerStyle(1+3)

: this parameter filters out postions that can't be played as a pattern of right hand thumb +3 fingers, very useful for jazz and walking bass styles. For instance a position like [0 3 X 0 1 0] isn't good, but [X 3 X 0 1 0] and [X 3 2 0 1 X] are good, here you can play them with your right hand thumb over 5th string and i,m,a over other 3 strings.

Inversions

: this parameter considers good inversion chords, so [0 3 X 0 1 0] is a good C major chord (C/E), very useful for walking bass style.

LIL

: Low Interval Level, this parameter, if enables, filtrates all positions that don't respect LIL contraits that defines some voicing positions of chords like interval inversions, root positions and so on. Good for calculating equilibrated chords.

Fingers

: this parameter filtrates positions that can't be played with 4 fingers or with 3 fingers and a barre. Usually always enabled!

Drop2

: this parameter restricts research only to drop 2 positions for seventh chords. Drop 2 position is the standard way used in jazz to find the best fingering of seveth chords.

FiltrateStrings

: this check enables string filtering with MaskString for escluding some strings in position research. With a mask of 011111 you can obtain only extract positions without the sixth string as [X 0 2 2 2 X] for A major.

LeastNumbStringPerChrd

: this parameter describes the least number of string playing in an acceptable chord position. For instance if this parameter is enabled and equal to 4, a position like [X 0 4 4 4 0] is good (5 strings), but [X 0 2 2 X X] isn't good. If its not enabled this parameter is ignored.

MaxNumbStringPerChrd

: as previous, but describes the maximum number of strings, so if its enabled and equal to 4, a position like [X 0 2 2 X X] is good (if no one parameters said different!)

LeastNumbPressStringsCons

: this parameter describes the least number of strings consecutively played in a chord. For instance [X 3 2 0 1 X] has 4 strings played, but [0 3 2 X 1 0] has only 3 string consecutively played. This parameter can be used to find compacted positions (normally of 3 or 4 note).

MinNumberNotesInVoicing: with this parameter you can state to the engine the number of voicing (note) that a chord must have. As you'll see after a while there are some functions that allow you to reduce number of notes in a chord (not necessary for chord sound as perfect 5th). But you can state to have at least 2 different notes.

MaxNumberNotesInVoicing

: as above you can state to the engine the maximum number of notes (pay attention to constraints defined by MinNumberNotesInVoicing <= than MaxNumberNotesInVoicing). This parameter can be used to find very compact positions (3 notes for 5 notes chord, or for mandulin).

ClearResult-PositionList: with this check you can clear temporary result list before every elaboration.

ExcludeUnnecessaryNotes

: this parameter is very useful for guitar mapping process, there is an algorithm that reduce the number of notes starting from fifth, then root, then other rules very complex. In this way its possible to map complex position in a few strings. Pay attention to enable inversions check if you decide to work with a little number of strings.

In the Mask section you can define which degrees to require and which degrees to ignore.

ActivateMask

: all degrees checked below are considered unnecessary

Voicing

: all degrees checked below are considered necessary in chord.

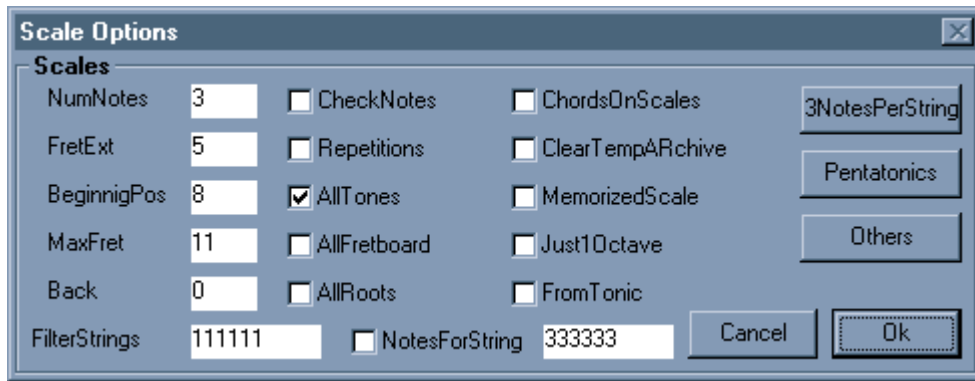
These parameters are useful especially for single chord searching!

After selecting your parameter values, you can press OK to start search or Cancel to avoid. During research a button labeled Stop appears to stop searching engine, and an incremental bar, tells you the status of elaboration!

NB: Pay attention to avoid settings parameters in contradiction each other, so you can't obtain anything.

?? Scale Search

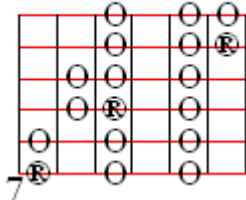
Using this window you can set all parameters for Scales and Arpeggios search. This is a very important window and you have to spend a lot of time to understand all possibilities you have.



NumNotes

: this parameter describes how many notes you want for each string. This parameter becomes a constraint only if you select CheckNotes.

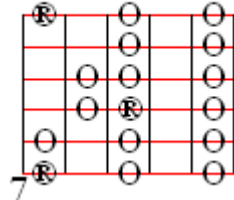
C MAGGIORE



FretExt

: this parameter defines the maximum extension in fret on each string, normally this is the default guide parameter for engine: is ignored if you activate CheckNotes.

C MAGGIORE



BeginningPos

: describes the starting fret in position analysis. This parameter is ignored if AllFretboard is checked.

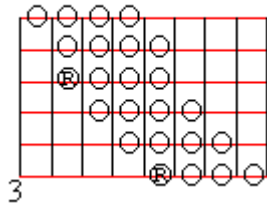
MaxFret

: this is the maximum fret used as starting position in AllFretboard modality

Back

: describes to engine the number of fret to subtract to StartingPos for each string, useful if you're looking for scale positions that shift on the left when you go from bass string to high as:

C Cromatica



NB: with the new engine released in v 3.0 this parameter is obsolete.

FilterStrings

: is the same as in ChordSearch window and allows you to select the string that you want to play.

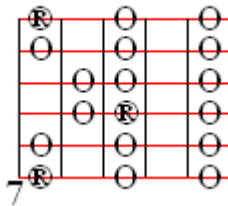
CheckNotes

: this parameter transforms NumNotes in a constraints for engine. Anyhow the research is based on FretExt parameter.

Repetitions

: allows engine to consider also repeated notes as

C MAGGIORE

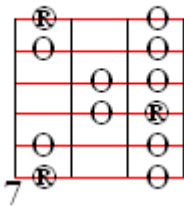


NB: The G on 3rd and 2nd string is repeated!

AllTones

: this parameter enforces engine to consider only positions that contain all notes without missing tones inside.

C MAGGIORE



NB: Here you can see an example of scale with missing notes (guided only by FretExt).

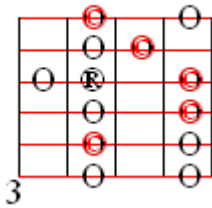
AllFretboard

: this parameter enables the analysis of all positions of a given scale on the fretboard increasing of a fret the parameter StartingPos at time. This is the simplest way to find a group of scales on your fertboard.

AllRoots: this parameter enforces engine to repeat every elaboration for every root. Another kind of group elaboration.

ChordsOnScales: you can search for scales, and see them in relationship with a chord for example a C Maggiore and a Dmin chord you'll have:

C MAGGIORE



In this case I've selected a Dmin chord in Saved Archive and then I've looked for a C Major scale with scale on chord parameter checked.

ClearTempArchive

: clear temporary archive before every elaboration.

MemorizedScales

: repeat elaboration for every memorized scale (another group elaboration).

Just1Octave

: this parameter enforces engine to search positions of only an extension of one octave.

FromTonic

: enforce engine to search for positions starting from root.

NoteForStrings: this list of parameter enforces the number of notes for each string (its an extension of NumNotes). This parameter has to be checked if you want engine to use it.

3NotesPerString

: this button sets the default for searching scale with 3 notes for each string.

Pentatonics

: this button sets automatically the parameters for pentatonics searching.

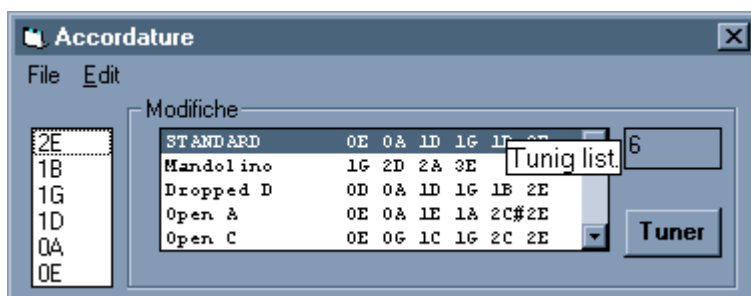
Other

: set parameters for generic research.

Then if you press OK you'll start engine, if you press cancel you'll abort the research.

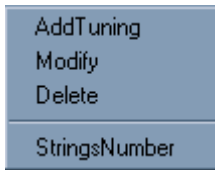
?? Tuning Window

This window is quite simple and allows you to select new tunings or to define and save your own.



On the left side you can see notes in your tuning in the form of Octave-Note. You can select a saved tuning from tuning list. On the right you can see the number of notes and a button for tuning your guitar accordly to selected tuning.

Using menus you can modify or add new tunings:



AddTuning: ask you for tuning name and then using the current value in the left list add under the selected item in the Tuning list the new tuning.

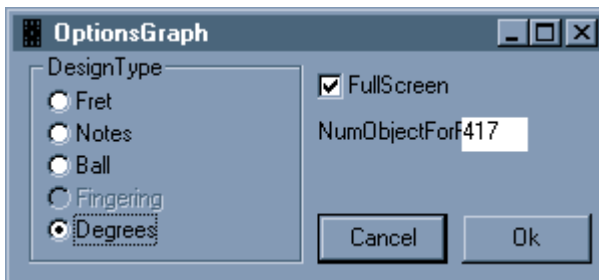
Modify: rebuild the selected tuning using the values in the left list

Delete: delete selected tuning

NB: Remember that if you want to make permanent your tuning modifications you have to go to main wondow and save Chords/Scales Information

?? Design Archive

In this window you can see a graphical and printable representation of your current archive. You can see chords or scales.



This window can be used to define properties of representation.

You can see:

Fret fret number

Notes

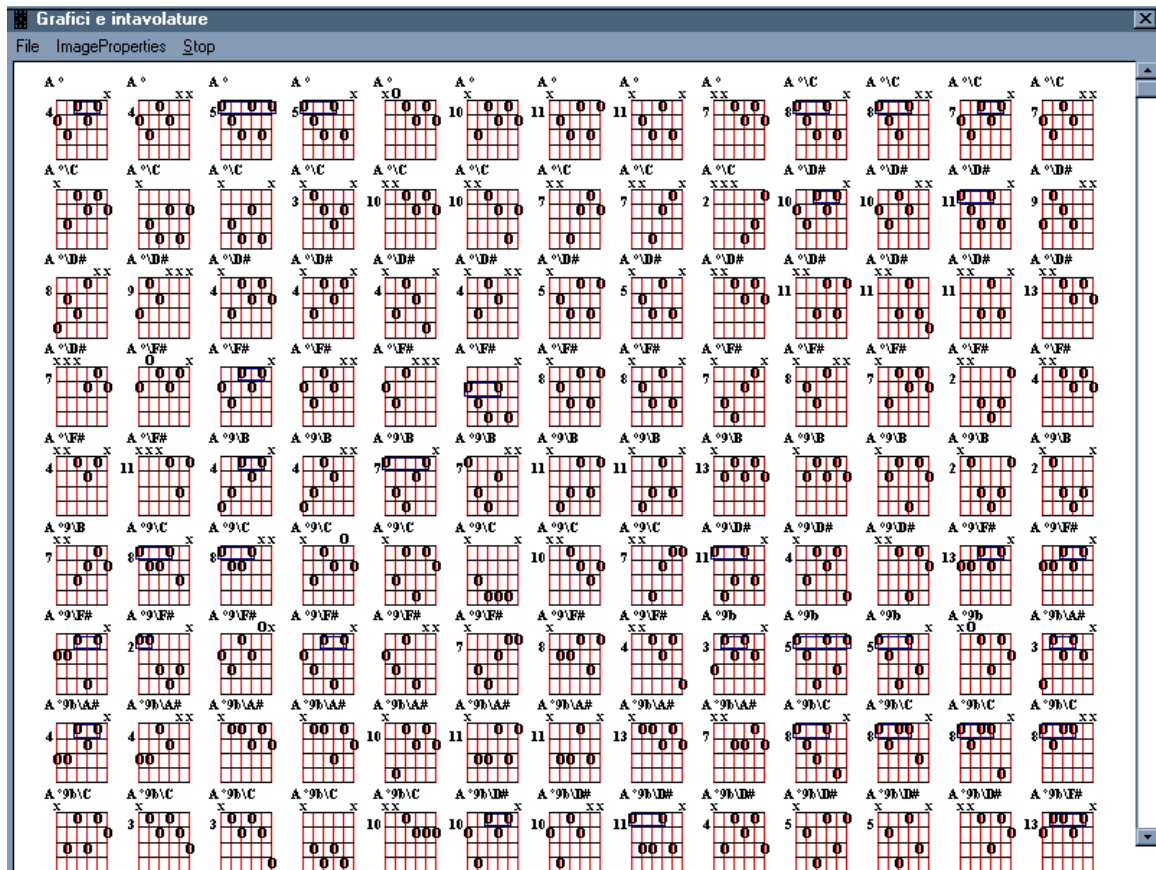
Ball

Degrees

NB: Now I have not developed an algorithm for fingering calculation! If you have suggestions

Then you can define the number of objects for each page, and if you want to see it at full page!

If you press ok you can see the result!



In this window you can see all positions you have found, you can print them with a postscript resolution. You can use menu to save image as bmp (NB: you have to work at 256 or less colors) or copy into clipboard, you can zoom, you can go to next graphic page or previous one.

The right scrollbar allows you to see all hidden chords below (think you're watching an A4 paper).

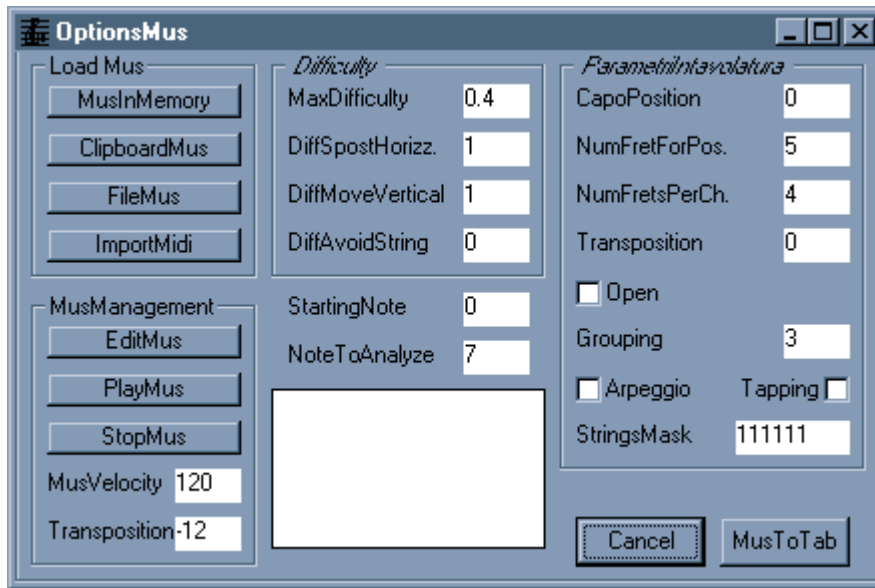
?? Music Analysis

This window is very important: its principal function is to map melodies over fretboard.

Melodies are described in MUS format (see Data Formats chapter) and a very complex algorithm finds the best way to play them mapping over fretboard.

This algorithm is used also in Find Chord From Note windows!

Using this window you have to define the MUS source (Last elaborated Object, Clipboard, File .MUS or MIDI track) then you can decide to play this MUS or to map over fretboard so you have to choose parameter values for determining difficulties or mapping method and finally you can see a tablature with positions.



As you can see there are a lot of parameters and its very important to understand their meanings before starting with MusToTab conversion.

Load Mus

MusInMemory

: every time an object becomes Current Object DGS memorize its mus description and with this button you can select this for MusToTab mapping.

ClipboardMus

: as you have seen before for archives there are some functions to memorize a MUS of a given positions so that you can analyze all possibilities to play the same note.

FileMus

: you can load an external Mus file.

ImportMidi

: you can import a MIDI track and convert it to Mus, so you can tab that track.

Mus Management

EditMus

: if you have the standard Mus editor you can start it to modify current Mus

PlayMus

: you can hear Mus

StopMus

: stop playing

MusVelocity

: you can define the velocity of playing

Transposition

: you can transpose of a given number of half-steps reproduction

Difficulty

You have to understand, at this point, some strategies used by mapping algorithm: it is based on the idea that a good position is a position with low level of difficulty, every movement over fretboard produce a difficulty so if you can reduce this parameter you can find a good position.

Now I define 3 specific parameters and one general parameter to describe difficulty: the algorithm use 3 specific parameters to define a difficulty of each position of a note and then make an average of all difficulty and if the value is less than MaxDifficulty than the position is good. The average is made after every note and if it becomes more than 2 times of MaxDifficulty the position is scraped and the algorithm tries with another one.

To calculate single note difficulty algorithm, define a frame (using previous notes) of five fret and 3 adjacent strings (one below and one above of the current string) and here every movement is without difficulty. Outside from the frame you can calculate horizontal and vertical distance from frame and define so the difficulty.

MaxDifficulty

: this parameter define the maximum level of the average difficulty of all notes in melody

DiffSpotHoriz.

: this parameter defines the difficulty of playing a note that is horizontally faraway from defined frame

DiffSpotVertical

: this parameter defines the difficulty of playing a note that is vertically faraway from defined frame

DiffAvoidString

: this parameter can be used to force to find positions in which notes are only in adjacent string, but for string skipping mapping or arpeggios its important to avoid this (this parameter is used only if string of previous note and current one are non consecutive)

StartingNote: you can define a range of Mus file to analyze, this is the starting note

NoteToAnalyze: this defines how many notes must be analyzed

TablatureParameters

Here you can specify what kind of tablature output you want:

CapoPosition

: here you can define the minimum fret is acceptable for you (if you decide to play melody with mobile capo)

NumFretForPos.

: this defines the horizontal extension of difficulty frame

NumFretPerChord

: this defines the horizontal extension of difficulty frame in case of a chord (more that a note played at the same time and over different strings)

Transposition

: number of half steps to transpose before mapping

Open

: if checked open strings are not used to calculate difficulty

Grouping

: you can decide to consider some notes as they will be played at the same time (for arpeggio) and so they are mapped over different strings

Arpeggio

: consider grouping for arpeggio mapping

Tapping

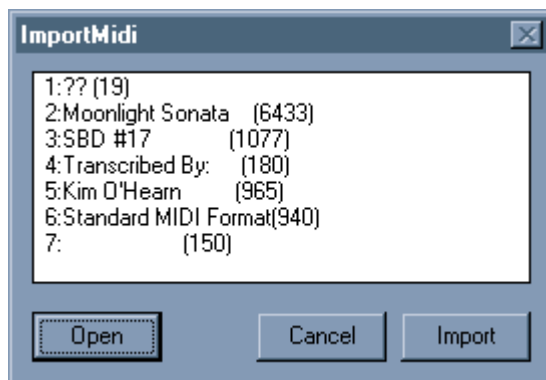
: consider grouping for tapping mapping (all on the same strings)

StringMask

: select strings to use

1. Import Midi

You can use this window to load a Midi track and to convert it to Mus format.



Open

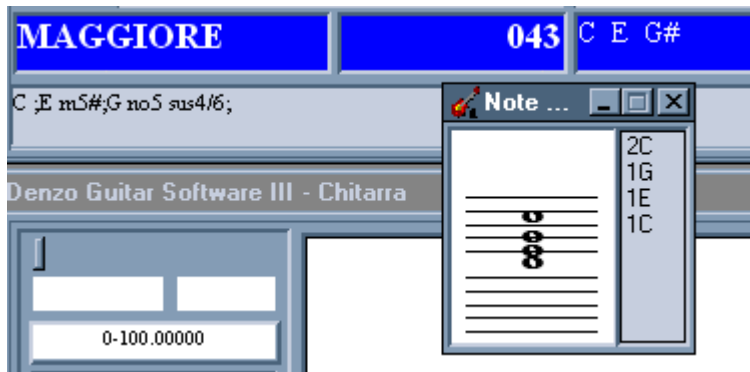
: shows dialog for selecting files, when you open a Midi file you can see the list with all tracks, (in parenthesis you can see how many events they have!), then you can select one, and import it

Import

: starts MidiToMus conversion

?? Find Chord From Notes

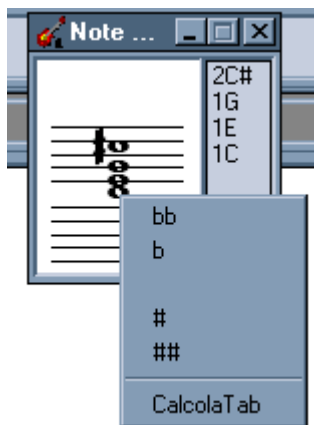
This is a very useful function, used to obtain a standard description and intavolature of a chord represented by its notes.



In this window you can simply drag-and-drop notes to the right position and in the Object Properties box you can read the description, all inversions, the formula and composition.

You can introduce alterations using a context menu that you can activate with a right click over the note you want to alter.

Finally you can find its intavolature with the voice CalcolaTab in context menu.



With NewChord menu you can define a new chord, number of notes and so on!

?? Analysis of fretboard positions

You can use this window to analyze a chord position on fretboard.



You can shift the sliders with your cursor to describe finger positions.

On the left side of scrollbars you can read the fret, on the right, the note.

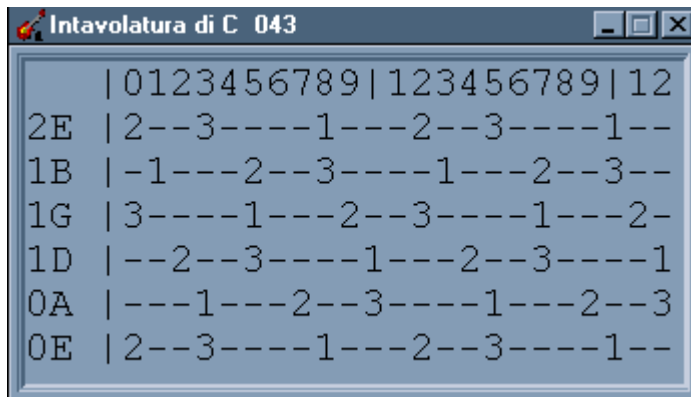
In the above box you can see all inversions of designed chords and its formula.

As you can see its very simple!

?? **Positions on freeboard**

This is a useful window to see all notes of current object mapped over freeboard, to study what you want to find. Its important to decide some parameters for scale research (for instance Back values) and other.

This window gives you a complete view of all possible positions.



In this example you can see all positions of C major chord notes.

NB Archeological note: this is the first result of .5 version! How many time!!!

?? **Build Exercise**

To be completed!

?? **Progressions**

?? **Data Formats**

In DENZO GUITAR SOFTWARE are used a lot of conventions to represent Objects, chords, scales and arpeggios. In this section they will be presented and described if you want to interface another application to DENZO GUITAR SOFTWARE.

1. **.MUS format**

This is the internal representation of music. In the same way as MIDI Mus files describe notes and tempos. They have a very simple structure and you can do by yourself a .Mus editor or require it to me.

A Mus file is based on 5 chars chunk that is repeated until the end of your song.

An example is:

```
f 008g 008a 008A 008c 108d 108D 108f 108g 108a 108A 108c 208d 208D 208f 208g 208
```

that represent a scale!

The chunk is so structured:

```
# Byte Meaning
```

```
1 Op Code
```

1 Point/Chord extension

1 Octave

2 Length

OpCode is one of these character: "pcCdDefFgGaAb*\\"

p pause

c c note (low case)

C c# note (up case = #)

* repeat from beginning

\ end of bar

Point/Chord extension:

- If there is a blank then you dont have any alteration

- If you have a . you have a pointed length (note length is increased by a half!)

- If you have an a this means that this note has to be played simultaneously to previous (a chord!)

Octave is represented by a digit:

The low E in standard guitar tuning is assumed in octave 0

Length: is represented by a number that describes how many notes you need for 2 quarters.

For example a quarter is represented by 02

An eighth is represented by 04

A triplet of sixteenth is represented by 12 and so on

NB Mus files are used to study Midi-to-tab mapping and other mapping algorithms.

?? **String Mask**

Normally you can use this parameter to select which strings to use:

You can use a 1 to enable a string and a 0 to disable it, so you obtain something like this: 111011 which means that 4th string (D in standard tuning) is disabled.

?? **Midi Settings Window**

With this window you can do all Midi settings for application.



SoundActive

: enables Midi output

NotesTranspositions

: default transposition for playing

Instruments

: defines instruments sound to use

Output

: define your Midi output support

?? **Task**

DENZO GUITAR SOFTWARE is task-oriented software. In other words its interface is organized by functions: you have general music functions, harmonizing functions, guitar mapping functions and so on.

The high complexity of functions and parameters needed for any goal, don't allow a design of a normal interface, so its necessary to use a new communication paradigm: task-oriented paradigm!

When you begin to use this interface you have to ask yourself what you want to do, then you can decide the right path of function to obtain your goal or task!

For example you want to analyze all mandolin position of a Dmin chord:

First of all select root in main window, and then select from chord list the chord you want. When Dmin is the current object you can open tuning window and selection mandolin tuning. Finally open guitar window, select the chords search function, and define parameters as you want, and run searching. All results will be listed in temporary result list on the right side of guitar window.

Now you can analyze chords one by one or all together using graphic window.

If you are careful you can see that we have described a path of functions: this is a task-oriented interface.

If you understand this you can do everything you want building the right functions path.