Software DIAOULEK 1.2

User's book

Table of contents

Introduction

Software « Diaoulek » is a software to help you learn languages, any language, but as this is a new software, it does not have already made lessons, except for the couple : language to be learned = Breton, language of reference = French.

In software « Diaoulek », the vocabulary to be learned is divided into small lessons, each of them

having about twenty entries (words, expressions, sayings...).The exact number of entries may vary but it must remain inferior to 100 on one given lesson. We can, however have as many lessons as we want. The set of available lessons is making the data-base of « Diaoulek ». All the words (more exactly : entries) of the data-base can be reused into your personal lessons. You will then take advantage of the audio files that may come together with the published lessons. Your own lessons will complete the data-base.

When you are studying a language, it is necessary to learn the vocabulary in the two directions : from the language to be learned to the reference language and from the reference language to the language to be learned. By convention, software « Diaoulek » is naming « Question » the text written into the language to be learned and « Response » the text written into the reference language. Thus, it is necessary to study the language in both directions : Question → Response and Response → Question. For software « Diaoulek », both directions are independent but you can change direction very easily by a simple click of the mouse.

Software « Diaoulek » is registering (after confirmation) your results, if you knew or not the vocabulary. For each entry, it registers the number of interrogations with a positive response, the number of interrogations with a negative response, the date of the last interrogation with a positive response and the date of the last interrogation with a negative response. As both directions Q → R and R → Q are independents, we have 8 values for each entry.

With its registered values, software « Diaoulek » is able to estimate how well you know each vocabulary entry that was studied. For each entry, it will allocate two errors (one on each direction). The software can compute for you heterogeneous lessons, that is to say lessons whose entries are taken into various lessons and more specially the most badly known entries in the data-base.

It is also possible to attribute to each entry one or more tags, this will allow you to selectively study the vocabulary of the land, of the farm, the bird or tree names, etc... For that, it is enough to ask software « Diaoulek » to compute an heterogeneous lesson with entries having all the same specific tag.

The separation of the vocabulary into short lessons is well suited to the learning phase but heterogeneous lessons (many kinds of them exist), are more suited to later reviewing.

Even if learning vocabulary is the software's main objective, « Diaoulek » may also display what is called complete lessons, like in a book, but a book the would be able to speak. This allows you to hear and hear again the lesson's sentences. The study of the vocabulary can also be made an integral part of such lessons. As the software is new, no method is using this possibility yet (at the end of 2010). However, you will find some 10 complete Breton → French lessons that were written to devise and tune the software. They are published with no guaranty at all ! Send me your corrections...

Software « Diaoulek » is written in C and should work on Linux and MS-Windows. It is only making use of the graphic library GTK and of the audio library « libsndfile ».

How the software looks like.

General Presentation.

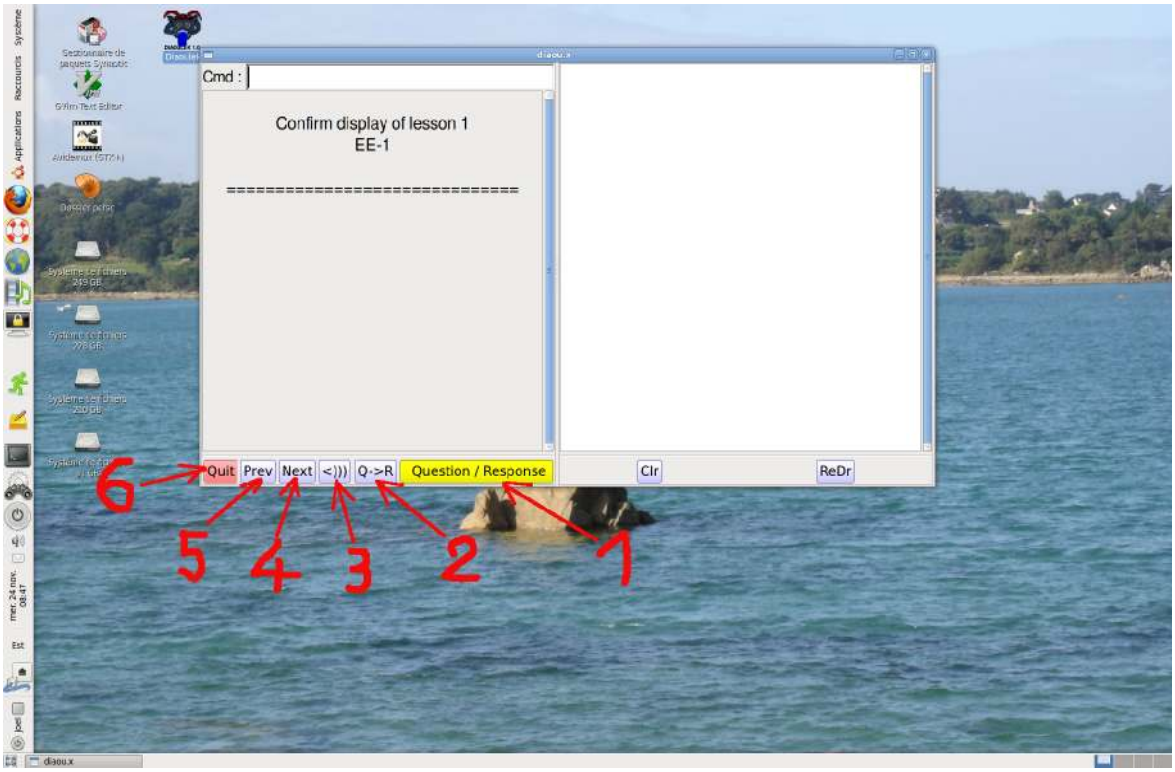

Figure 1 (Diaoulek-0.png) *Whole screen capture showing the different buttons.*

The main window of software « Diaoulek » is divided in two parts : Left part is the "vocabulary window" where you will learn or review the vocabulary. The right part is the “lesson window” where the few complete lessons will be displayed. The button labeled 1 is the main button of « Diaoulek ». If you can succeed in mastering this button, you will have understood almost everything in software « Diaoulek ».

This button has two states : the yellow state and the red state. When in the yellow state, it is simply waiting for a mouse click to make the software continue and in its red state, it is ready to register the meaning of your next click and make the software pass to the next step.

This button is able to discriminate between the different clicks generated by the mouse, the left click (the usual one) means : yes, it is good, OK... The right click means : no, it is not good, KO... The middle click means : I pass, forget the previous recording, let us continue without any recording...

All the other buttons are good and honest buttons which are doing only one single thing when pressed.

Here is on the following table the various actions of the main button according to its state :

State :	*Left click*	*Middle click*	*Right click*
Yellow state	Continue	Erase the recording of the previous word	Continue
Red state	Record as good and continue	Pass the recording of the present word and continue	Record as false and continue

Remark 1 : The records (« good » or « false » results) are only provisional, they will become definitive only when leaving the lesson and after a demand of confirmation by the software.

In fact, the main button is into its yellow state during the first part of the interrogation and then it toggles to the red state when we are in the second part which is the verification phase. The red means that the software is ready to register your response. The middle click erases the recording of the previous word in the yellow state but it does not make the program to continue, the software is still waiting for a left or right click. In the red state, we don't register anything for the present word but we continue.

Remark 2 : in the interrogation phase (yellow state of the main button), the left or right clicks of the mouse have the same effect, to make the program continue, but it is better to make a left click if you think that you know the meaning or the translation of the word and you will do a right click in the opposite case if you are sure you don't know. If you do that, during the next phase, the verification one, where the recording takes place, the click will become an almost instinctive reaction and you will have fewer errors.

As it has already been said, by definition, software « Diaoulek » is calling "Question" the words or expressions in the language to be learned and "Response" their translation into the reference language. One must learn the vocabulary in both directions : Question -> Response and Response -> Question. The button labeled 2 allows you to toggle between those two directions Q->R and R->Q.

The learning of a living language without audio records is almost impossible. The default behavior of « Diaoulek » is to always search for audio files. However, it is sometimes better to cut off sound, for example to accelerate the learning of a lesson that was already studied. The button labeled as 3 allows you to toggle easily from one state to the other. The " <))) " logo symbolize a loudspeaker in activity and the logo " <XX " symbolize (even more badly) a blocked loudspeaker.

My advice is to divide the learning of a language into small lessons, each with some twenty words into them. Button 4 (Next), allows you to go to the next lesson and button 5 (Prev, put here for « Previous ») allows you to go back to the previous lesson.

The button labeled as 6 (Quit) makes you leave the application while registering the present state (last studied lesson, position and size of the window...). You will be back to that state next time you use « Diaoulek ».

Remark 3 : You can study any language with « Diaoulek ». However, as the software is very young, it is only for the pair Breton → French that ready-made lessons are available at present. You will be able to download 10 full lessons. Though they were mainly written for technical reasons, the code's adjustment, they can also be used as an introduction to the study of Breton language. Moreover, other vocabulary lessons are also available, together they constitute a data-base of about 3500 words or expressions (end of 2011) which can be used into your own lessons. All the lessons, the full ones and those of the data-base are given with their audio files.

Remark 4 : Figure 1 was taken on a computer with an Ubuntu 10.04 operating system, with a MS-Windows system the color of the buttons may be slightly different as the versions of the graphic library GTK are also different.

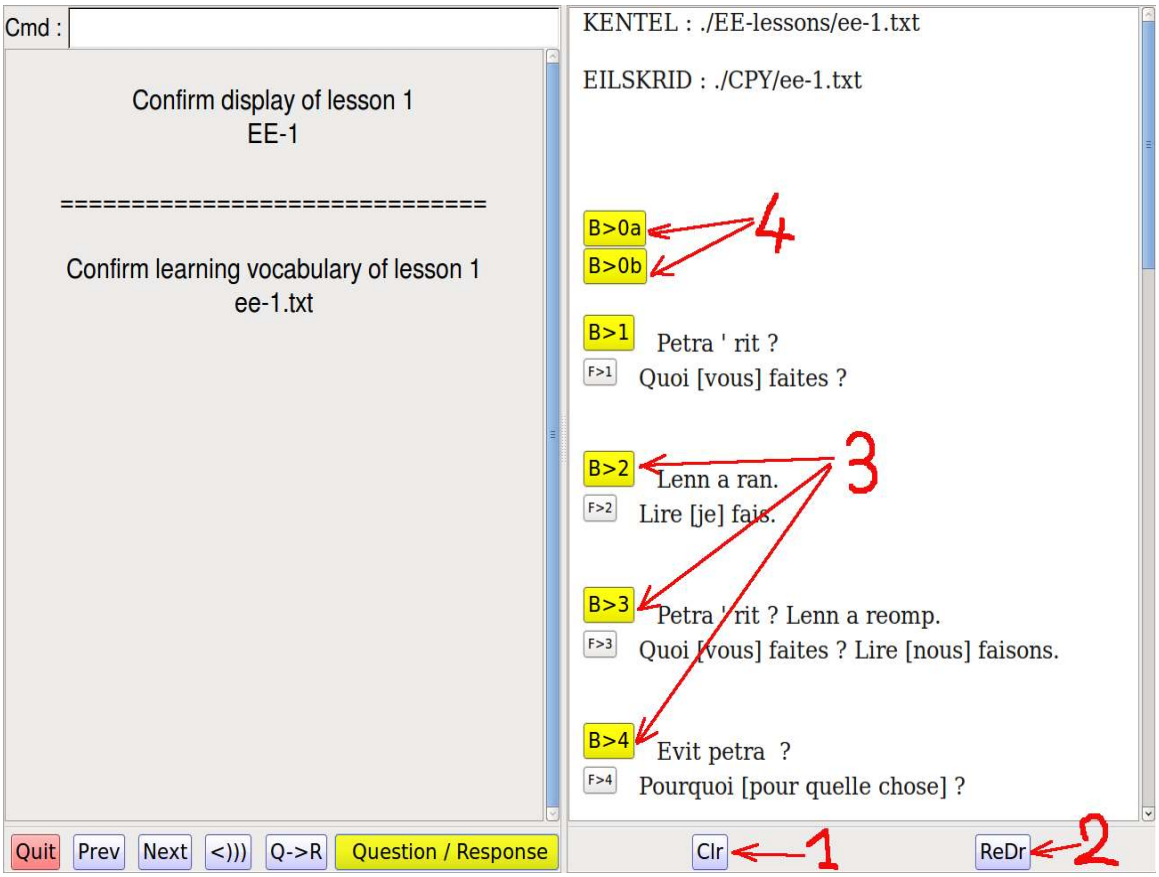

Figure 2 (Diaoulek-1.png) *Display of a full lesson.*

Even if the main objective of software « Diaoulek » is the learning and review of the vocabulary, a secondary objective is the display of classical lessons, like in a book, but a book that is able to speak.

The button with label 1 (Clr pour "Clear") allows you to erase the lesson window, this can be useful when you are learning the vocabulary of that lesson.

The button with label 2 (ReDr pour "ReDraw") reestablishes the contents of the lesson window.

The buttons with label 3 in the lesson windows are audio tags that allow you to hear and hear again each of the sentences of the lesson.

The buttons with label 4 allow you to hear and hear again the full lesson. That is however specific to that lesson and may not exist on other lessons.

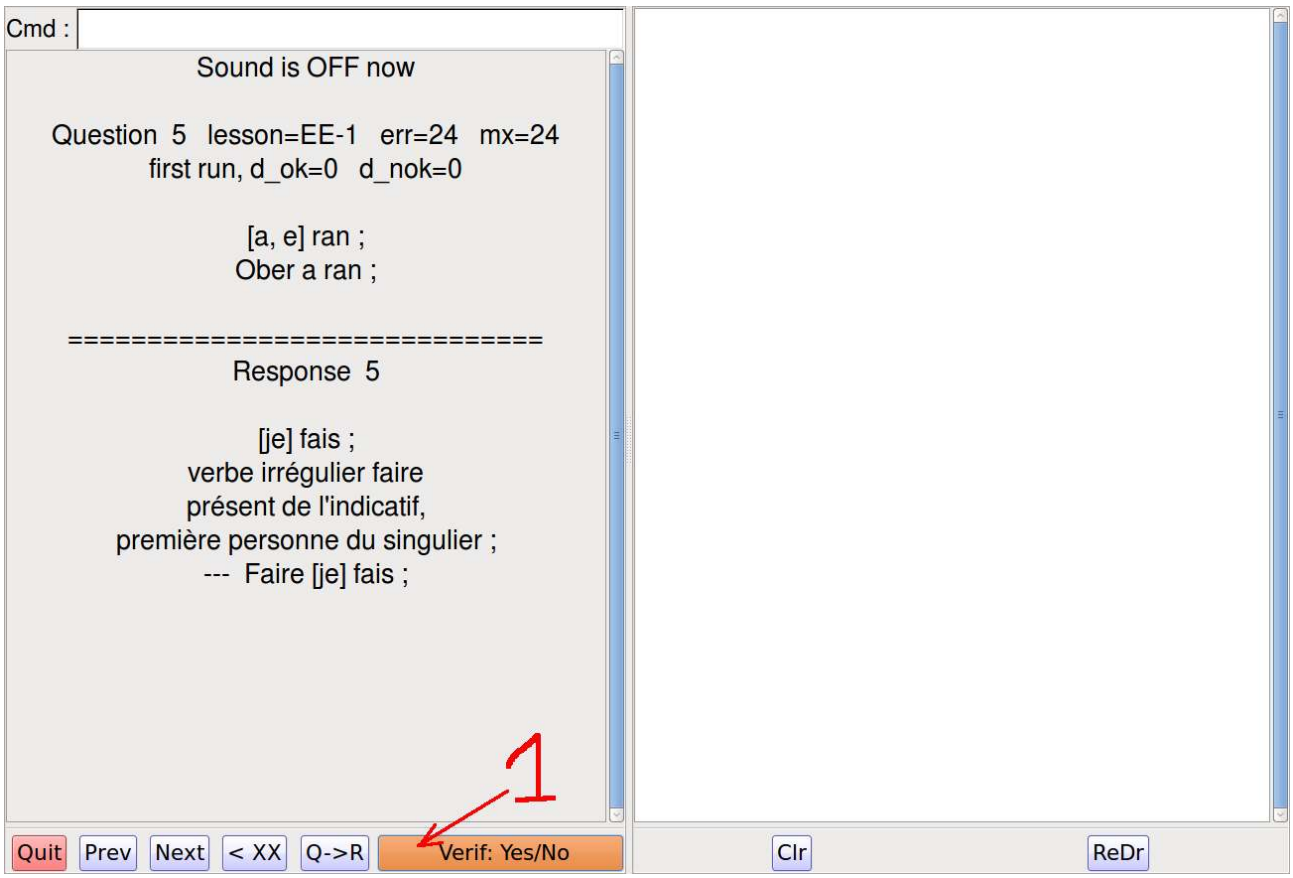

Figure 3 (Diaoulek-5.png) *Learning of a lesson's vocabulary.*

We are now for this full lesson in a phase of vocabulary learning. The lesson window has been erased, the sound is off as indicated by the button « <XX ». The learning is done in the direction Question → Response as indicated by the button« Q->R ».

As the sound is off, it is a writing interrogation (before the line "=========="). When the sound is active, the interrogation is an oral one, at least when an audio file is available. The Breton writings would have then appear only in the verification phase, that is to say under the line "==========".

On figure 3, we are already in the verification phase and the main button (labeled 1) is in its red state. It is ready to register your click. A left click to say "it is good", a right click to say "I don't know", or a middle click to say "jump, forget that question, and don't record"

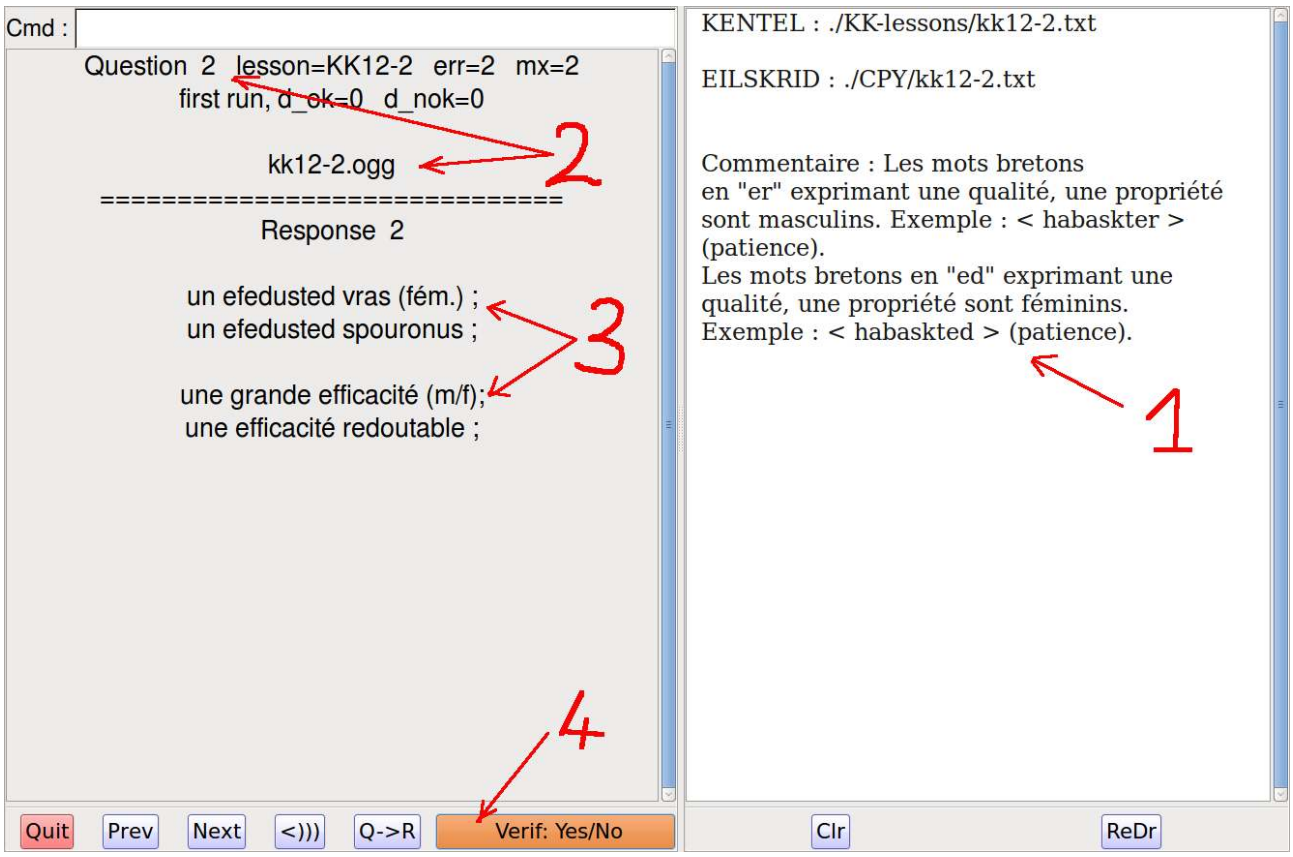

Figure 4 (Diaoulek-6.png) *Display of an almost ordinary lesson.*

Even if we have only very few full lessons, some lessons can still include useful comments like here on the window labeled as 1.

In the case of figure 4 the sound is active as indicated by the button "<)))". In the interrogation phase, before the line "====", we find, as indicated in 2, the number of the question, the name of the lesson (here : KK12-2) and the name of the sound file (here : kk12-2.ogg). Most often several words are on the same sound file. There are also other indications that will be explained on the following figure.

In the verification phase (labeled as 3), under the line "====", the words or expressions in Breton with their translations in French are given. It is then possible to verify that the Breton speech was well understood and that the translation is known... or the contrary.

As in the case of the previous figure, the main button (marked here as 4) is waiting for your click to make the recording.

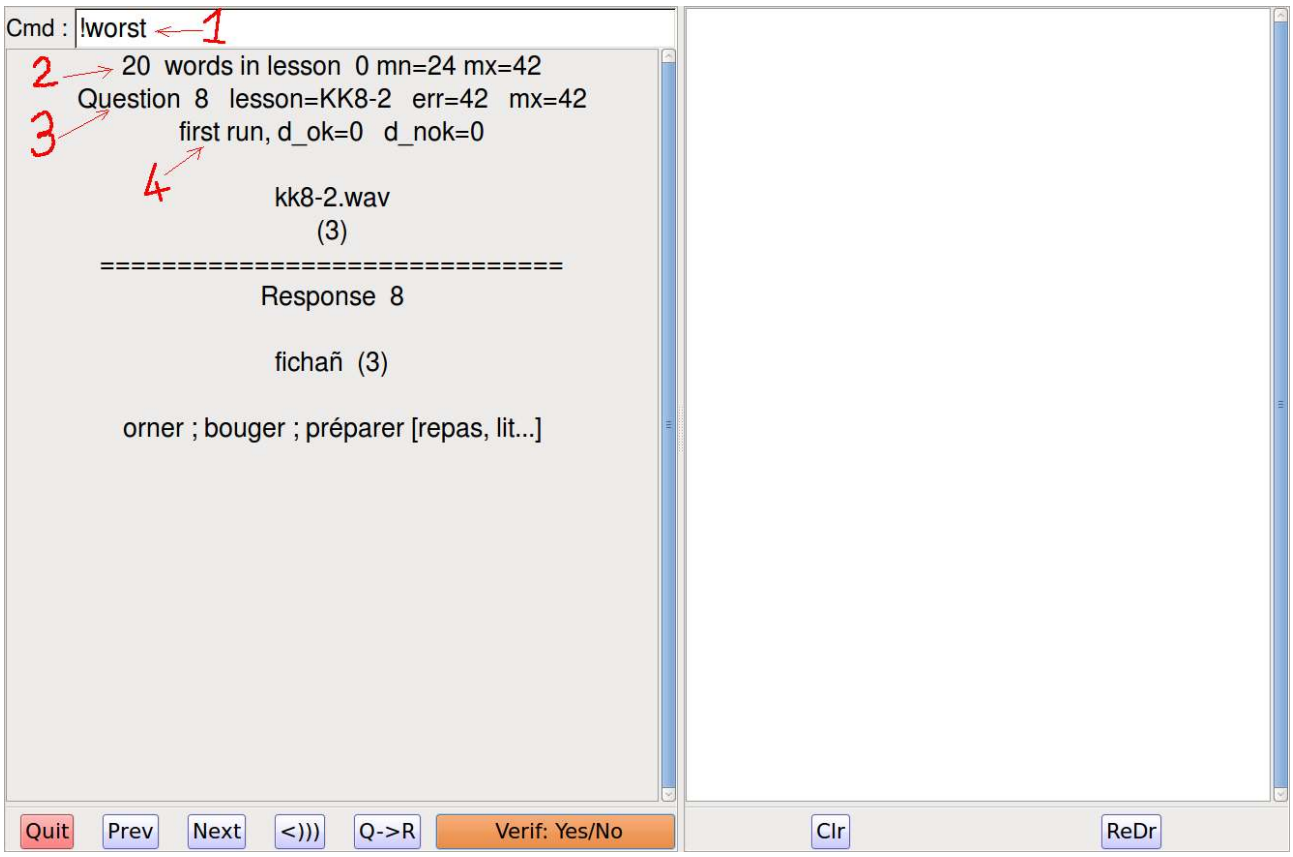

Figure 5 (Diaoulek-7.png) *Making use of the command line*.

The command line (labeled as 1), at the top of the vocabulary window, allows you to reach a given lesson by its number or by its name without having to use the "Next" and "Prev" buttons. But a lot of other commands for different objectives are also at your disposal. All these commands are beginning by an exclamation mark.

It is possible to make the software compute for you composite lessons where words are chosen according to various criteria. For example a selection of the most badly known words, words that were not already studied, words with a given tag (vocabulary of the sea, of the farm, of the house, bird names...). Obviously we make the difference between the directions Q->R and R->Q. A list of all the available commands can be obtained by typing "help" in the line with label 1.

In the case of figure 5, we have, by the command "!worst", made software Diaoulek to compute a lesson with the 20 most badly known words in the present direction Q->R. These words are taken among those already studied. As the software is registering for each word and each direction the number of interrogations with a positive result , the number of interrogations with a negative result (badly known word...) but also the dates of the positive or negative interrogations, it is able to find the most badly known entries among those already studied. To do that, the software takes into account your previous results as well as a forgetfulness factor. Computation time is negligible even on an old computer.

The software Diaoulek gives to each word an error. A large error indicates a badly known word and an error equal to 0, a well known word.

In the interrogation phase (over the line "===="), various informations are given. The line with label 2 indicates that there are 20 words in lesson 0. It is the usual number and all the

computed lessons are said to be "lesson 0". But line 2 also indicates that in the present lesson the smallest error is **mn**=24 and the largest is **mx**=42.

On the line with number 3, it is also indicated that we are displaying question 8 which is sending you to lesson KK8-2. This word, in the present direction Q->R does have an error "**err**=42" and the maximal error in lesson is also "**mx**=42". We then are dealing with the most badly known word or at least one of them if several have the same error. Software « Diaoulek » makes its interrogations in random order but it will insist on the badly known words. It does have some inclination to be slightly "diabolic".

The game's objective is to decrease **mx** towards zero. At that point the lesson can be considered as having been learned.

The lesson's learning is made in two steps. During the first step (as indicated here by "**first run**" on the line with tag 4), we will see successively all the words in random order. When all the words have been seen at least once, we pass to the "**second run**" during which the software will strongly bias it random order towards the most badly known words.

On the line with tag 4, "**d_ok**" and "**d_nok**" are the numbers of good and bad results for the learning of that word in the present lesson (it is not the whole history record of that word). When we are in "**first run**" these numbers are both equal to zero. During the "**second run**", at least one of these numbers is not equal to zero.

If we don't have enough time, we will stop at the end of "**first run**". When we will enter into the « **second run** », we will be sure that every word was seen at least once. If we have a little more time, we will continue to make "**mx**" decrease and reach a zero value. At that point, the lesson will be considered as known, at least for the day...

Redaction of a simplified lesson.

At present, only a few complete lessons have been added to the release of software « Diaoulek », they were written for technical reasons, code writing and tuning. Possibly, if nothing better is available, they can be used as an introduction to the study of the Breton language starting from French. It is however better to study a language with the help of a classical and popular method and limit the use of software "Diaoulek" to the learning of the vocabulary found in your method's lessons. For that purpose, you will write your own lessons, limiting them, it is an advice, to something like twenty words each. We will see now how to write a simplified lesson.

Composition of a simplified lesson.

Name of the file.

A lesson is a text file, you can choose any name for the file but it is better to make it simple and still end those names by the extension « .txt », in order to show explicitly that we are dealing with simple text files. If you also intent to write more than one file of the same series, it may be good to use file names with a number. For example, you can name your personal files : perso1.txt, perso2.txt, … , perso10.txt. With that convention, you can, in the configuration file « diaou.conf » introduce the list of the lessons under the form perso1->10.txt and even if you want to begin by the last lesson : perso10->1.txt. That is much more easy to type than the full set : perso10.txt perso9.txt … perso1.txt.

The name or alias of the lesson.

Each lesson must receive an alias (in fact a name). This alias must be composed of a few well chosen letters or numbers. For example in the case of the personal lessons that were mentioned above the aliases P1, P2, … , P10 would be well suited. Another possibility could have been PP-1, …, PP-10, as the sign « - » is allowed. The aliases have to be short and easy to remember. They will be used in the command line of the vocabulary window to call the lessons. If you have 200 lessons, it is easier than having to press 199 times on the « Next » button in order to reach the last lesson.

Composition of the lesson.

The lesson's first line must begin by the characters « **!#** » and then a space and the alias of the lesson. On the following lines, the couples questions and responses will be introduced by the signs « # », « **Q>** » and « **R>** ». One example will be more explicit. Here is your first personal lesson :

```
!# P1
 !  Here is my first personal lesson
#
Q>   ti
R>   house

#
Q>   heol
R>   sun
```

The lines beginning by the sign « **!** » are commentaries and they are ignored with exception of the first one when the exclamation mark is followed by the sign « # ». In that case, the software looks for the lesson's alias.
We can put anything we want after the « **Q>** » and « **R>** » marks, even a few text lines. We only have to know that the lines will be displayed as they are and centered. So, you must not write lines that are too long.
A slightly more elaborate example is given in the file « ex_simple.txt ».
In that lesson I am using some personal conventions which are useful only for the study of the Breton language. Here is one example :

```
#
Q>  ur gleizenn ; kleizennoù ; div gleizenn ;
R>  une cicatrice ; pluriel ; deux cicatrices ;
```

In my lessons, a Breton word is generally introduced by an article and this may induce a mutation, the word is followed by its plural (without any mutation) and the gender is indicated by « div » (pronounce « diou », for feminine words) or « daou » (for masculine words). These two words « div » and « daou » which mean “two” for feminine and masculine nouns are inducing softening mutations. For the translation in French, everything can be translated like in the above example but most often the plural and the gender indication are replaced by three dots like in :

```
R>  une cicatrice ; …
```

Another of my personal conventions is the indication of the number of expected translations like below :

```
#
Q>  strafuilhañ (2)
R>  Troubler, Inquiéter
```

Obviously, you can use or not that kind of conventions or some others of your own.

Reuse of entries belonging to other lessons.

It is very useful to be able to reuse the words or expressions belonging to other lessons. If these lessons have been published on Internet and moreover with their audio files, that will make, together with your own lessons, a very useful data-base. We will see now how to reuse the previous lessons, that they be your own or downloaded ones.

All your lessons must be referred to in your configuration file « diaou.conf », otherwise they

will be ignored by software « Diaoulek ». We will see later the contains of that file.

Then, you have also to create a dictionary of all your entries in alphabetic order. This is easily done by typing in the command line of the vocabulary window the order « **!dico** ». This order generates the file « diaou_dico ». Here are some lines of such a file :

```
##anal *KE-5 * corps santé
##anaout anavezout *KE-36 * R2 verbe société
##andon *KK7-2 * paysage eau
##anez ma *KK8-2 * expression
##anken *KK12-3 * Q3 sentiment
##Ankou skrignet *KE-5 * R2 expression
##ankounac'h *KE-33 * Q2 esprit
##ankounac'haat *KE-36 * verbe société
##annoar *KK3-2 * animal ferme
```

In order to reuse the entry « anken » in one of your lessons, it is enough to make a copy and paste of the line :

```
##anken *KK12-3 * Q3 sentiment
```

into your lesson.

In fact, only the beginning of the line is necessary, you have there the entry you are looking for, here « anken », followed by the lesson name (alias) where it can be found, here « KK12-3 ». The remaining of the line is not necessary though the list of the tags attached to the data-base entry, here « sentiment » , may be of some use to select a particular word or improve the data-base.

With what has been said in the chapter « Composition of a simplified lesson », at present you know everything necessary for the redaction of your own lessons. However, it is possible to make more complete lessons and that will be detailed in the following chapter.

A slightly more complete lesson.

The separation line between entries.

For the simplified lessons of the later chapter, the separation line between the Question/Response couples, was reduced to its most simple form. It was composed of a single character : « # ». We will see now how we can complete that line and add some more information useful to the management of the data-base or to the display of the entry.

Formation of the entries dictionary.

We have seen that it was very useful to make a dictionary of all the words or expressions of the data base in order to reuse them in other lessons. How is that dictionary made ? In the case of a simplified lesson, the beginning of the « Question » is used as reference. Thus for :

```
#
Q>  ur gleizenn ; kleizennoù ; div gleizenn ;
R>  une cicatrice ; pluriel ; deux cicatrices ;
```

the words « ur gleizenn » would be retained for the formation of the dictionary. The defect of that method is evident, many dictionary entries will begin by an article ! It is however possible to impose the word or expression that will be used for the formation of the dictionary. For that purpose, the separation line will be completed by the chosen reference

followed by the character « * ». For example, in the preceding case we can write :

```
# kleizenn *
Q>  ur gleizenn ; kleizennoù ; div gleizenn ;
R>  une cicatrice ; pluriel ; deux cicatrices ;
```

This couple of « Question/Response » will then be referenced under the word « kleizenn » in the data-base dictionary. We could have put « Ref 35 » or anything else but it is wise to use a reference that is in close relationship with the contains of the « Question/Response ».

Use of commas and semi-colons.

We can use any character for the referencing of the « Question/Response » couple except obviously the « # » and « * » characters. We can also use commas and semi-colons but those characters have special meanings in versions 1.1 and later of the software. In the the case of multi words, verbs, etc... we can separate these words or verbs by commas as in the following example :

```
 #gwennañ, gwennaat *  couleur verbe

<))  aln-k87.ogg  2092031  2253823
Q>   [1] gwennañ ;
     [2] gwennaat ;
R>   [1] blanchir, justifier [blanchir qq en justice] ;
     [2] blanchir, mûrir [pour la moisson] ;
```

This couple of Question-Response will be referenced in the data base dictionary as both « gwennañ » and « gwennaat ». We will have in the file « diaou_dico » :

```
##gwennaat #gwennañ, gwennaat *K87 * couleur verbe
```

##gwennañ #gwennañ, gwennaat *K87 * couleur verbe

The use of the semi-colon will now be explained. When we are referencing a couple of « Question/Response », the separation line is scanned starting from the first « # » and up to the encounter of a character « * » or « ; »

Let us suppose that we want to reference an expression like « I am hungry like a bear » that we have in a lesson whose name is « P7 ». We also want that expression to be referenced under the words « hungry » and « bear » but still let the user know that these words are used in the expression « hungry like a bear ». We will then write :

```
 #hungry, bear ; hungry like a bear *  expression
```

In the file « diaou_dico », we will have the two references :

```
##bear #hungry, bear ; hungry like a bear *P7 * expression

##hungry #hungry, bear ; hungry like a bear *P7 * expression
```

If we had replaced in the lesson the semi-colon by a comma, we would have had the three following references in the file « diaou_dico » :

```
##bear #hungry, bear, hungry like a bear *P7 * expression

##hungry #hungry, bear, hungry like a bear *P7 * expression

##hungry like a bear #hungry, bear, hungry like a bear *P7 * expression
```

The tags.

It was already said that the couples « Question/Response » could receive tags in order to study and review vocabulary with a given tag. These tags are put on the separation line after the character « ***** ». We can have several tags. For example, in the case of « kleizenn » that was given above, we can have :

```
# kleizenn * corps visage médecine
Q>  ur gleizenn ; kleizennoù ; div gleizenn ;
R>  une cicatrice ; pluriel ; deux cicatrices ;
```

Here, three tags « corps » (body), « visage » (face) and « médecine » (medicine) are attached to the entry « kleizenn » (scar). If a tag is used for the first time, it will be added to the tag dictionary.
It is possible to obtain the list, in alphabetic order, of the tags already in use by typing the order « **!ltag** » in the command line of the « vocabulary window ». This command will display the tag list with, for each of them, the number of times it was used in the data-base and the number of words or expressions really studied in both directions QR and RQ. For example for the tag « école » (school), at one time I had the following indication :
* école * nb=33 QR=4 RQ=26
That is to say, in the data-base 33 words or expressions have the tag « école » but only 4 were really studied in the direction Question-Response (Breton to French) and 26 in the direction Response-Question (French to Breton)
The order « **!ltag** » creates also the file « diaou_ltag » with the same indications. That file can be copied into your lessons if you want to have under your eyes the list of tags already in use. That is convenient when you compose a lesson of your own but this is not mandatory.

The very special tags : « Qi » and « Ri ».

The tags « Qi » and « Ri » where « i » is an integer are very special. We have already seen, for example into :

```
#
Q>  strafuilhañ (2)
R>  Troubler, Inquiéter
```

that I find convenient to indicate between parentheses the number of translations expected for the word. In the case of a written interrogation, the indication « (2) » will appear because everything written before the marker « R> » is displayed on the screen. Obviously, that is not true during an oral interrogation and we have such an interrogation as soon as an audio file is available (and the sound is not off). In order to correct that flaw, it is enough to introduce the tag « Q2 » which means that the « Question » is waiting for two responses. We will then write something like :

strafuilhañ * Q2 verbe
Q> strafuilhañ (2)
R> Troubler, Inquiéter

In the above example, we have two tags « Q2 » and « verbe ». In the case of an oral interrogation, the indication « (2) » will appear below the name of the audio file. This means that the question wants two responses (translations).

An example of such a behavior is visible on figure 5 where the Breton word « fichañ » is waiting for 3 translations. In the interrogation part (over the line "====") the indication « (3) » appears below the name of the audio file « kk8-2.wav ». In the verification part (below the line "====") we can find the text in Breton with the indication of the number of expected translations (again !). Then we have the text in French with the three translations.

We must take note that the indication « (3) » of the Breton text and the tag « Q3 » are independent. It is up to the author of the lesson to make them be the same.

The tag « Ri », where « i » is an integer, was introduced by reason of symmetry when a word in the language of reference has several translations in the language to be learned. In the present state of the software that kind of tag is of no use. However, it could be employed if we had audio files for the language of reference.

The audio files.

It is unlikely that a lay person using software « Diaoulek » will create its own audio files. However, it is useful to know that it is possible to point to an audio file in a given lesson by having after the separation line another line such as :

```
<))  ee-2.ogg   2848768   2928128
```

We begin that line by « **<))** », then we have an audio file name, then two numbers indicating the beginning and ending records numbers (frame numbers). As we have generally 40000 frames by second, the beginning and endings numbers are often very large. As a convention of the software, if 0 and 0 are indicated as beginning and endings frames, this means : all the file . So :

```
<))  ee-2.ogg   0   0
```

means all the audio file « ee-2.ogg » from beginning to end.

We can have several audio records, all will be played successively one after the other. Perhaps it will be possible in a future version of « Diaoulek », to make a selection among these files according to the name they have.

Remark : Software « Diaoulek » is using the sound library « Libsndfile » and so it can read any format recognized by the sound library. For all practical porpoises I am using only « .wav » and « .ogg » files. The « .wav » files are for the originals and the « .ogg » files (a free compressed format analogous to mp3) are used for the distribution. Later I may use the « speex » format if it is introduced into the audio library.

The configuration files of software « Diaoulek ».

Software « Diaoulek » has two configuration files : « diaou.rc » and « diaou.conf ». The first file is in fact a configuration file of the graphic library GTK and the second is specific to the « Diaoulek » software.

The configuration file « diaou.rc ».

This file is a configuration file of the graphic library GTK, it allows you to adjust the color of the buttons in their various states and the font of the button labels. At least for the first times you will be using this software, it is unlikely that you will need much changes to this file. You can note that the colors are defined by three numbers varying between 0.0 and 1.0. These numbers represent the red, green and blue colors. We can also adjust the fonts by choosing their names and size. Nevertheless the fonts have to exist on your computer and in the selected size or we will fall back on the default font.

The configuration file « diaou.conf ».

This file is a configuration file specific to software « Diaoulek ». It gives to the software the list of the lessons making its data-base and set the values of some parameters like the size of the display fonts or the windows default sizes.

The list of the lessons.

The components of the lessons' list must be given between the two flag marks « **Which_lesson :>** » and « **<:** ». It is better to give only one lesson by line and indicate the absolute or relative path like as follow :

```
Which_lesson :>
 ./K-lessons/k86.txt
 ./K-lessons/ks53.txt
 <:
```

However, in order to simplify the writing, when a full set of lessons is available, for example for all the lessons « ee-i.txt », with « i » varying between 1 and 10, one can use the notation « ee-1->10.txt ». In that case, we have an increase from 1 to 10, but a decrease is also possible by writing : « ee-10->1.txt ». It is even possible to have in a file name several variable numbers as below for the « kk... » lessons :

```
Which_lesson :>
 ./EE-lessons/ee-1->10.txt
 ./KK-lessons/kk13->1-3->1.txt
 ./K-lessons/ks53.txt
 ./K-lessons/k79->86.txt
 ./KE-lessons/ke-1->76.txt
 <:
```

In that example, the notation « `kk13->1-3->1.txt` » replaces the full series :

kk13-3.txt, kk13-2.txt, kk13-1.txt, kk12-3.txt, kk12-2.txt, … , kk1-1.txt

Adjustment of the parameters specific to your own memory.

It seems that the memory of the human being can be subdivided between long-term and short-term memories. As users are not alike, a first group of adjustable parameters has been provided for the long-term memory and a second group for the short-term memory. The default parameters can be changed in the configuration file « diaou.conf ».

Parameters in use for the computation of the long-term error.

Software « Diaoulek » is working in the following way : Based on your past performances, the software allocates to each entry an error which is supposed to represent the probability for you to ignore that word meaning. A large error should be given to a badly known word (or expression). On the contrary a small error is for a well known word (in the long-term sense). During the lesson's on-going learning of the day, this error should normally decrease and reach a zero value. When the error for every word of the lesson has reached the zero level, that lesson can be considered as known — provisionally — (this is the short-term memory). It would be of no utility to keep repeating the same words. It is better to switch to something else and return later to the same lesson. If you do that way, the errors in the lesson will not restart from zero because you probably don't know perfectly every entry of the lesson but the errors would, hopefully, have decreased.

Roughly, the software is computing a starting error (longterm error) according to the following formula :

err = cof1*nb_nok – cof2*nb_ok + (date – date0)/cof3

where the coefficients **cof1, cof2, cof3** are adjustable decimal numbers (« float » numbers) and the numbers **nb_nok** and **nb_ok** represent respectively the numbers of false and correct responses since you have begun to study that word. The factor **(date - date0)** represents the number of days since the last interrogation of the world. That formula only gives an approximate idea of the error computation. The actual implementation in the software introduces various thresholds and secondary effects, the details of which are of no use here.

The default values of the adjustable coefficients are :

cof1 = 9.0 ; **cof2** = 3.0 ; **cof3** = 2.0

At least when you begin to use software « Diaoulek », it is better not to change coefficients **cof1** and **cof2**. On the contrary, you can slightly decrease **cof3** if you tend to easily forget things or increase it if you have an excellent memory. In any kind, you can only make **cof1** and **cof2** to vary between 0 to 10 and **cof3** to vary between 0.1 to 10.

The parameters in use for the computation of the short-term error.

The coefficients **cof4** and **cof5** are only used for the decrease toward zero of the error during the lesson current learning. They must represent the learning process of the short-term memory. They are not used for the computation of the starting error which is supposed to represent your long-term knowledge of that word or expression. Each time you have a good response, the current error will decrease by **cof4** points and each time you have a bad response, the current error will increase by **cof5** points. In order to avoid endless wandering, the error must not increase too rapidly or decrease too slowly. The default values are : **cof4** = 6.0 and **cof5** = 10.0. Moreover, the ratio **cof1/cof2** will be kept close to **1. + cof5/cof4**. Various procedures have also been added to the software in order to avoid boring the student and decrease his feeling of getting nowhere. These procedures may bypass the effect of the two coefficients **cof4** and **cof5** which must take values between 1. and 10.

As a conclusion to this long study on the memory and the adjustable coefficients :

It is better to only change slightly the coefficient **cof3** to adapt the software to how fast your long-term memory is forgetting vocabulary !

Choice of the fonts.

The fonts in use in the « lesson » and « vocabulary » windows can be changed by the orders « Font_lesson » and « Font_vocab ». For example :

```
#
Font_lesson :>   Serif 18   <:
Font_vocab  :>   Helvetica normal 20   <:
#
```

The fonts you want to use must have been set on your computer, otherwise the default fonts will be used instead.

The windows' default sizes.

The default sizes of the various windows can be changed in the configuration file « diaou.conf ». You can set the dimension in pixels of the whole frame, vocabulary window + lesson window by the orders « **Horiz_pix_total** » and « **Verti_pix_total** » as follow :

```
#
  Horiz_pix_total :>   1200  <:
  Verti_pix_total :>   850  <:
#
```

The width of the vocabulary window can be given in pixels by the order « **Width_pix_vocab** ». For example :

```
# #  Number of pixels for the width of the window displaying
#   the vocabulary (left part of main window) ; example : Width_pix_total/2
#
    Width_pix_vocab :> 580 <:
#
```

Obviously, subtracting « **Width_pix_vocab** » from « **Horiz_pix_total** », you get the width in pixels of the lesson window.

These sizes are only default values, they are used only at the first start of software « Diaoulek » or after an emergency stop due to a software crash or by a click on the decoration button « × » of the total window. When using the software, you can adjust the size and position of the windows by dragging the edges or corners. When you leave the software normally by using the « **Quit** » button, the positions and sizes of the various windows are recorded and they will be used at the next start of « Diaoulek ».

The command line of the « vocabulary window ».

Many actions are possible thanks to the command line. A list of all the possible actions can be obtained by writing the order « **help** » in that command line. We can for example review the most badly known words, learn new ones, etc... For example the order « **!worst qr** » allows you to get the 20 most badly known words and imposes the QR direction. Obviously « **!worst rq** » would have set the learning direction to be RQ. Another example : « **!tag oiseau** » looks for the 20 most badly known words or expressions with the tag « **oiseau** » (bird). The order « **!ntag oiseau** » looks for at most 20 words with the tag

« **oiseau** » that were not studied yet. I let you discover by yourself all the possibilities of the command line by typing the orders « **help** » or « **!help** » on it. Only the action of two particular orders that let you check, correct and copy the data-base will be explained with some details in the present document.

The order « !ccdb » to check and correct the data-base.

Your data-base is a dynamic element, which has to maintain the list of all the words and expressions that are part of it. We have also to known in what lesson and where in the lesson we can find the words and expressions. In the data-base, we must also have the recording of all your learning results. The data-base must adapt when you modify, introduce or remove lessons. As a consequence, the data-base is always varying and, as in Nature with the genetic code, any error in the data-base will propagate and be amplified. The software « Diaoulek » contains elaborate mechanisms to maintain, verify and correct the data-base. These mechanisms are automatic and transparent for the user. However, in some cases, it can be useful to force a verification/correction to be made. That will be the case, for example if you observe some mixing between the « Questions » and the « Responses » when you study a lesson. Then, to correct the problem you will type, in the command line of the vocabulary window, the order « **!ccdb** » (which means « Check and Correct Data-Base »). That order forces a correction of the complete data-base. Normally, you should not lose any element of the previous records. Only some rare entries that are truly irrecoverable should be reseted to zero.

The order « **!ccdb** » is fast and efficient. My advice is to use it systematically in some somewhat peculiar cases :

1) If you share the same data-base between two not very compatible operating systems like MS-Windows and Linux. In that case the dates of modification of the files are not always transmitted well.
2) If you transmit lesson files with an USB key. In that case the transmission of the dates is often erroneous. This induces errors when the lessons are not new ones. You have to force the data-base actualization.
3) If you are using the order « !synchro » in order to synchronize the data-base between two computers. That can be useful, for example after a vacation if you have continued to use « Diaoulek » on a portable...

The order « !import » to import lessons, sound files and the data-base.

The order « !import » has been introduced in version 1.2 to ease the software upgrading by making automatic the copy of the lessons and data-base of a previous version. When you upgrade your software, the quickest way is to copy the executable into the same directory where your previous executable was and use it in lieu of the older one. Then a « !ccdb » order will check and perhaps adapt your data-base to your new « Diaoulek » version. However, it is also possible to import your lessons and their sound files as well as the former data-base into your new « Diaoulek » directory. For that purpose, you will use the order « !import » which must be followed by the absolute path towards your older configuration file. For example :

```
!import /home/user/D1-0/diaou.conf
```

or on a Windows OS :

```
!import C:\Program Files\D1-0\diaou.conf
```

The « Diaoulek » configuration file will be checked, the name of the lessons will be extracted and these lessons and their associated sound files will be copied into your new « Diaoulek » directory. Your former data-base will also be copied. However, if a lesson already exists into your new « Diaoulek » directory, it will not be imported. That is why the order « !import » should be used before any other action as soon as you have obtained your new executable.

The order « !import », once it has imported all your lessons and the data-base, closes your « Diaoulek » session. You will then start another « Diaoulek » session and execute a « !ccdb » order. This will check, correct and adapt the copy of your data-base. You may then continue to use software « Diaoulek » as usual.

Remark : The order « !import » is also useful if you want to make a copy of your lessons and data-base into another directory for example on a USB key.

Known « bugs » in software « Diaoulek ».

Any software released in open nature goes with its swarm of « bugs ». The present project makes no exception to the rule. Among the known « bugs », we can quote :

1) It is not possible to internationalize the project.
2) Under Linux/Ubuntu, with and even without Alsa, a long enough record makes the screen to darken.
3) The main button can freeze. That may not be due to « Diaoulek », however. In any case, it is enough to click two times on one of the two buttons at the left of the main one and the software will continue its job.
4) The installation under Linux must be done by trial and error. That is the common problem for softwares on Linux when there are no ready made packages for the specific distribution.
5) There are many mistakes in the lessons and the audio files are not perfect. That can be improved with the participation of everybody.

Some advices on how to use software « Diaoulek ».

As a conclusion, here are some advices on how to use software « Diaoulek ». As it is true for any software, it is only a tool and you have to learn how to use it best. Because of the numerous possibility of software « Diaoulek », everybody can use it in a way that will be best suited to his needs and his remembering possibilities. Some general advices may however be useful. The key to success when learning a language is everyday repetition. A little every day allows you to learn, much from time to time is of no value. In order to learn a language, you have first to listen to it. The ear have a memory of its own, young children are able to understand before they speak. When we are a grown up person, it is possible for us to speak as we have studied in books with grammar and we can chose our words. To do mistakes when speaking is not too unforgivable (except at school !), but you still have to understand the reply. For that, your ear would have to be used to various kinds of voices, masculine, feminine, young children ones, and also to various accents and speech rates.

Perhaps, one day, software « Diaoulek » will offer you that possibility if various persons are releasing audio files. For the moment, you will have to do with only one voice, it is anyway better than nothing...

When learning a language, my advice is to use a good method, a not too difficult and very progressive one. Many are available, but anyway, you will have to use the one selected by your professor or by yourself if your are an isolated student. Then, you will use software « Diaoulek » to learn and review the vocabulary of your method's lessons. You will have to write your own personal lessons for software « Diaoulek ». They will be, as explained earlier, plain text files that may be very simple to write. In the Breton → French case, you will use the data-base made from all the published lessons and the writing of your own lessons will, most often, be reduced to simple copy and paste actions, at least if you are a beginner. The simplest and most frequently encountered vocabulary is already in the data-base. The first few times you learn your personal lessons, you will call them in the command line by their ordering numbers or by their aliases. Don't forget to learn your lessons in both directions QR and RQ.

The review of the vocabulary that you have already well studied will be done by the use of the orders « !worst qr » and « !worst rq ». You can have as many « !worst » lessons you may want, one following the other. Of course, you will be going successively towards better known words, at least in theory. You can stop the learning of a lesson after a first pass over each word (first run), or you can continue for a better learning. You have many kinds of compound lessons computed according to different criteria, they will allow you to criss-cross through your previously learned vocabulary. The main drawback of the methods used at school is that we keep learning new words and let the review of that same, so difficultly (and provisionally) learned, vocabulary to be done by chance encounters in texts. Thanks to its records, software « Diaoulek » allows you to review your vocabulary in an optimal fashion. Forgetting is not bad, I even believe that this is an integral part in the learning process. Too bad, however that it is the easiest part of the job, but it is not difficult to learn again what was already well studied. If you use it regularly, software « Diaoulek » will help you to do that !