

DESIGNING A LANGUAGE PACK FOR SLOVA

Slova allows you to create your own language packs. A language pack contains a font – in the language of your choice – and English words with their foreign translations.

Creating a language pack is simple, but you need a Ti-83+ assembler and some way to turn programs into application variables. For Windows users, I have included “spasm,” which is a popular tool to create ASM games for the Ti-83+. I have also included “8xp28xv”, a tool by Jacobly which lets you turn programs into application variables. Finally you will need the file “relocate.inc.” It works for spasm, and it should work for all other assemblers.

I've included two files – Rhymes.asm and RussianS.asm – which are the files I used to make the language packs that come with the game. You can look at them as examples. I highly recommend that you look at Rhymes.asm first, because it's more straight-forward and doesn't require you to know Russian ☺

Language packs are simply lists of data. Your language **MUST** start with the following:

```
.org 0000
```

```
#include "relocate.inc"
```

```
tempswaparea = $82A5
```

```
.db "Slova"
```

After this, you need 1 byte telling the calculator how many “set of words” your language pack has. Every level in Slova contains seven English words with seven translations, so each set of words you include has to have seven words. So take the number of English words you included and divide by 7. If your language pack has 56 English words and 56 Spanish translations, your language pack has **8** sets of words.

```
.db 8
```

A language pack supports up to 127 sets of words. Make sure that each set contains 7 English words with 7 translations, or you'll experience bugs!

Now insert these four bytes of data:

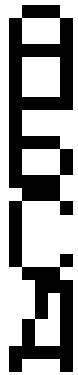
```
.dw LargeFont
```

```
.dw WordLists
```

Now include the line “.db 255” We will come back to this later.

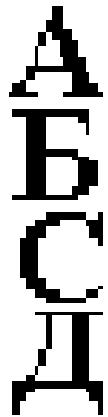
Next, you will need a font of the foreign language you want to include. (Slova provides the 5x7 font used for English words. However, in Rhymes.asm, I had to include that font again, because I have to consider the rhyming words as a “foreign language.”) Your font can have up to 159 characters, and you will use ASCII 97 and above (lowercase ‘a’ and above) to enter words using this font.

Each character in your font needs to be 5 pixels wide by 7 pixels high. As an example, here are the first four characters of my Russian Font:

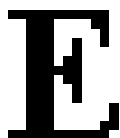


Spasm allows you to include a monochrome bitmap of your font. Or you can compile binary sprite data if you wish. Include the file after the label “SmallFont:”

Now create a 16x16-sized font in your foreign language. Again, you can include a monochrome bitmap or compile binary sprite data. Include the file after the label “LargeFont:”



When you make your 16x16 font, make sure that the four corners of each 16x16 character are WHITE. This is because of Slova’s special collision detection—you want the player to COLLECT these letters, not bump into them!



This is fine.



This is not.

Now to create the English words and their translations! Remember to create words and their translations in groups of seven. For each set of seven you will need some kind of label, maybe “WordSet1:” You will also need the following line at the beginning of each set:

```
relocate(tempswaparea)
```

And include the following line at the end of each set:

```
endrelocate()
```

Now create seven English words, each with some kind of label. Each word must be in CAPITAL letters and end with a zero. (Use an underscore, “_”, for spaces.) Also each word can have up to 16 characters (not including the ‘0’). If you have more than 16 characters the game might crash.

EnglishWord1:

```
.db “CAT”,0
```

EnglishWord2:

```
.db “DOG”,0
```

EnglishWord3:

```
.db “FROG”,0
```

EnglishWord4:

```
.db “GOAT”,0
```

EnglishWord5:

```
.db “COW_FOOD”,0
```

EnglishWord6:

```
.db “PIG”,0
```

EnglishWord7:

```
.db “HORSE”,0
```

So how do you add data for words in a foreign language? Well, the fonts you created are mapped to lowercase “a.” In Slova, the word “aaaaa” will display the first character of your foreign language five times. The word “abcde” will display the first five characters of your font.

.db “acbd”,0 **ACEA**

So, right after the English words, type in the 7 foreign translations, each with a label, each with LOWERCASE LETTERS (OR symbols that come after lowercase ‘z’), each with a 0 afterwards, each no more than sixteen characters.

When your lists are done, you need to tell the calculator where in the appvar these words are. Immediately after your 16x16 font you should have a label called “WordLists.” For each set of seven words you have, you will need 30 bytes of data. For the first two bytes, tell the calculator the label of the first word set:

.dw WordSet1

Then, type in the label for each English word in the set, immediately followed by the label for each translation.

.dw EnglishWord1
.dw SpanishWord1
.dw EnglishWord2
.dw SpanishWord2
.dw EnglishWord3
.dw SpanishWord3
.dw EnglishWord4
.dw SpanishWord4
.dw EnglishWord5
.dw SpanishWord5
.dw EnglishWord6
.dw SpanishWord6
.dw EnglishWord7
.dw SpanishWord7

Do you want to include an “accent mark” in your font? An accent mark will help players know how to pronounce the foreign words in your language pack, because they will know which syllable needs to be stressed. If you want to include accent marks in your language pack, you need to do three things:

1. In the language pack, place the character ‘@’ in any foreign words, at the place you want the accent mark displayed. For example, “g@ato” or “McD@onalds”
2. Slova will draw the small 5x7 accent mark for you, but you will need to create your own 16x16 accent mark.
3. Remember the line “.db 255?” You need to tell the calculator where in your 16x16 font the accent mark is. Count the number of characters **before** your accent mark. Then replace the number 255 with this number. (If there is no accent mark in your font, **leave this alone!**)

That’s it! You’ve created a language pack! Be sure and double check / test your work, because Slova does not include any form of error-handling.

Now compile your data. For spasm, you will need to create a program:

Spasm LanguagePack.asm LanguagePack.8xp

Then, if you are using Windows, drag this program onto 8xp28xv, and it will become an application variable. PRESTO!

By the way, language packs cannot be bigger than 16 KB. Furthermore, remember that a language pack on the calculator **MUST BE ARCHIVED**.