

DialogED

For 3D Game Studio

By Padmalcom / Blue Boons Interactive 2007



www.jofre.de

Introduction

Within the last years the NPC's AI got more and more complex. Mostly this growth took place in shooting games making it possible to let NPCs act like humans. But looking beyond one's own nose, the gaming community offers much more possibilities to let their figures appear more realistic. So, for instance, in dialogs.

But compared to a human to human conversation, dialogs in games were characterised by straightness and by a missing flexibility. Decisions during the virtual verbal interaction raised problems for programmers because of the difficulty of constantly having an overview. Since Conitec's great 3D Game Studio does unfortunately not support a tree structure it was not easy to find a pattern allowing to visualize a dialog while programming.

This condition ameliorated with the Delphi SDK written by LazyDog for the 3D Game Studio community. He - as well as his steady help and calmness ;-) – helped us to develop a WYSIWYG dialog editor allowing to click your dialog parts easily together. This dialog file will be saved in one single file and started with one single line of WDL code.

What began as a small tool for our first adventure thus developed to a (hopefully) easy to use program helping not only us but also you to let your NPCs seem more real.

Features

- Unlimited dialogs in your game
- Unlimited decisions in a dialog with up to 5 possibilities
- Easy to use WYSIWYG PanelED to place your dialog panel etc.
- Use variable operation and function calls in your dialog
- Start your dialog with just one line of code

Version

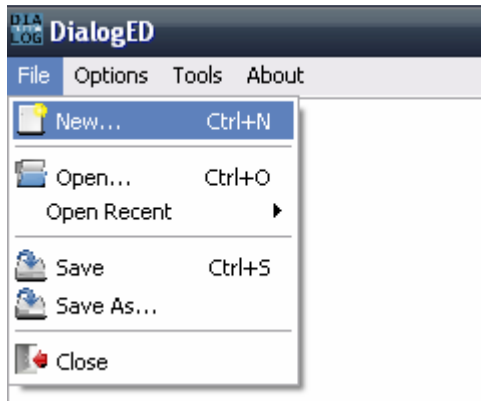
This is the first version 0.9.7 offered by us. If you find bugs or if you have ideas for possible improvements tell us via xjupasx@web.de or via the contact formulary on

<http://www.jofre.de/kontakt.html>

Disclaimer

We do not take any responsibility for damages on your system. Since this program does not create files / folders in any other directory but in its installation directory it is very unlikely that any problems occur.

How to create a new dialog



Creating a new dialog is pretty easy. Click on **File** → **New...** and a new file is opened. If a project is opened, changed and not saved you will be asked for a confirmation. You see a single entry in the tree view called “Root”. This is the beginning of our dialog file. To tell the dll that will later on start our dialog what our windows, fonts, and buttons will look like we need now to create a project configuration file. This happens by starting the **PanelED**.

Illustration 1: Create a new dialog

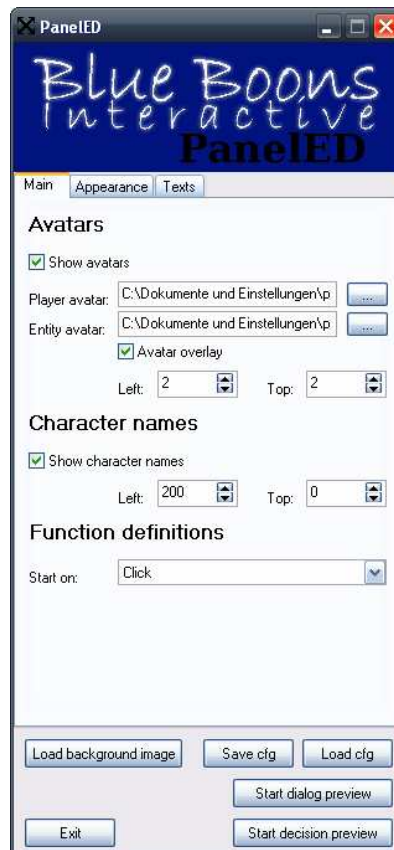


Illustration 2: Setting the project options

This is the most exhausting part. I will not explain every single option since it is more or less self explaining.

If you do not want to spend so much time on creating your own configuration file you can use the files created by us. Click on load will directly lead to the **configs** folder where you can find them. When you are done, save your file and click **Options** → **Change config file...** in the DialogED to choose the configuration file you have just created.

The first dialog

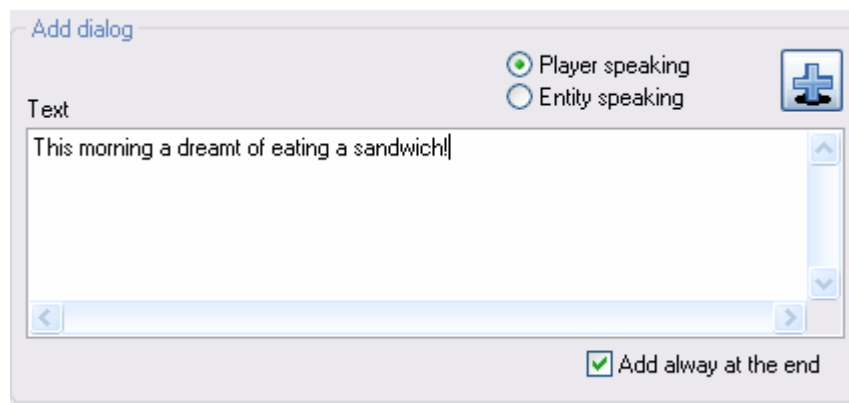


Illustration 3: Adding a dialog

Type your selected text in the text box, select who is talking - entity or player - and click on the blue plus. There we have our first sentence! Easy, isn't it? If you add another dialog you will see that it is placed right behind the first one. In this way one could tell a long long story. For example about our player eating a sandwich. But what if we want him to decide if he wants tuna on his sandwich or bacon?

The first decision

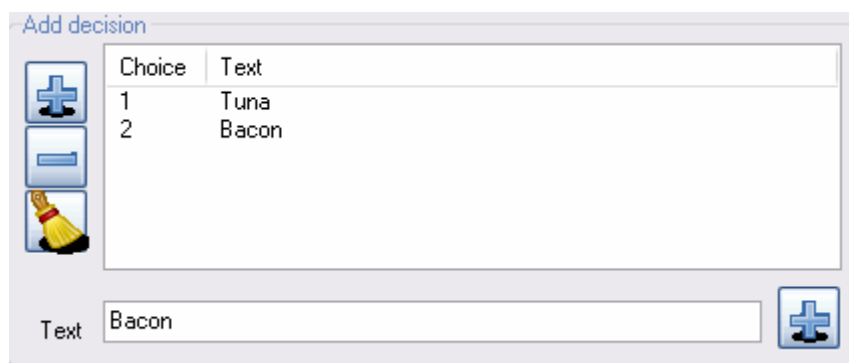
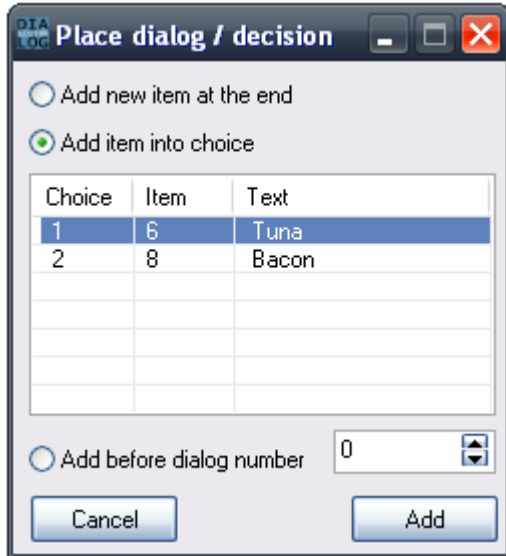


Illustration 4: Adding a decision

Enter the possible choice in the text field on the bottom and press enter or click the left blue plus. To delete an item select it and click the minus. To delete all items click on the broom. To add the entire decision to our dialog file click the plus on the left side.

Adding a dialog to a decision

If we now want to display that our entity is happy about hearing that we decided to take a tuna sandwich we do the following: Type your dialog as usual but uncheck “Add always at the end”. If you now click on the blue plus a new dialog will appear:



Here you can choose if you simply want to add your object at the end of the dialog, if you want to add it into a choice or before a already created dialog.

We want ours to be created in the *Tuna* choice. As you can see we now have a dialog being displayed when we click on Tuna in our game.

Not add a second dialog in the choice bacon and one at the end saying “Bye” or whatever.

Illustration 5: Placing a dialog

Our structure should look like that now:

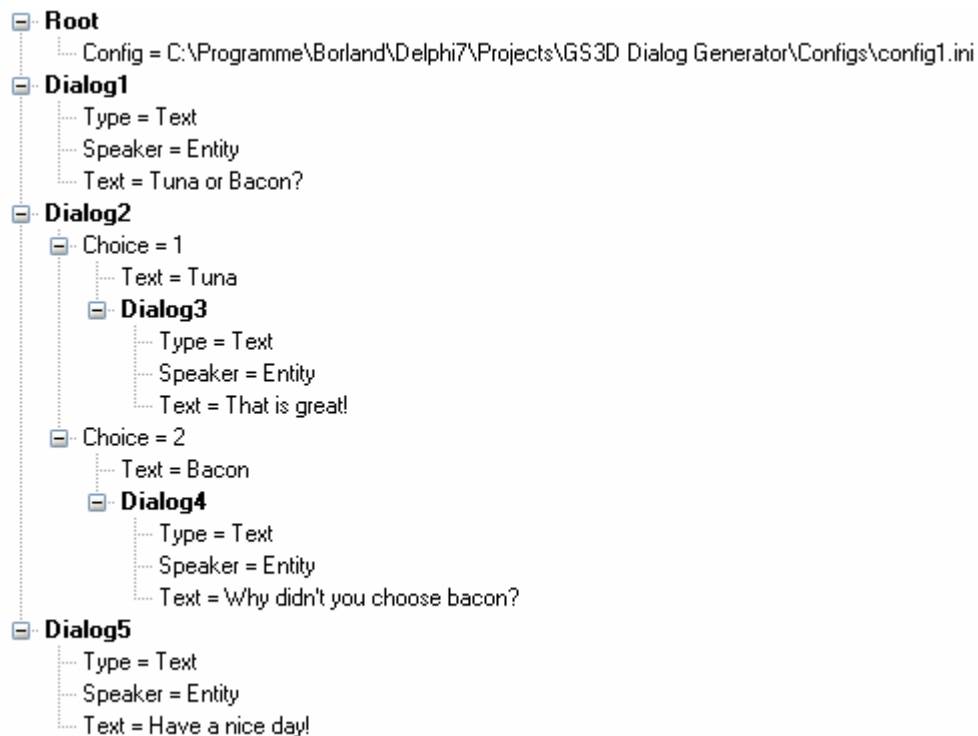


Illustration 6: Our dialog structure

Now save your project using **File → Save as...** and be prepared for the more advanced options.

Variable operations

Just imagine our NPC gives us our sandwich and we want it to our inventory. For that kind of operations the DialogED offers you the **Add variable operation**.

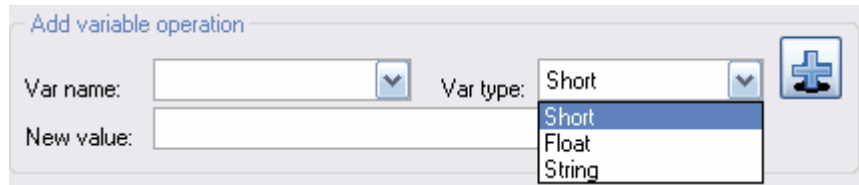


Illustration 7: Variable operations

If we have a variable *food* in our WDL code and we want to set it from 1 to 2 use the following parameters.

- **Var name:** food
- **Var type:** short
- **New value:** 2

Now you have to select the dialog in your dialog structure where this operation should be added to and click the blue plus.

To change a real number, not a natural number, you have to choose *Float* as Var type. If you want to change a string, select string. **Attention:** If the string has a predefined length the new string will be truncated if it is longer. If you want to avoid that, create a string in your WDL code without any start value.

Call a function

To call a function from you dialog use this ability of DialogED:

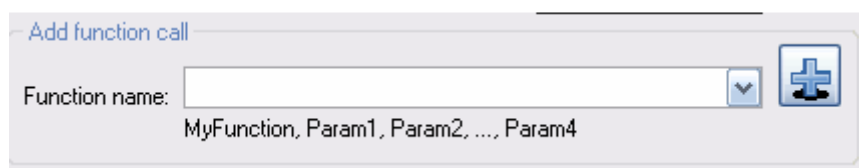


Illustration 8: Function calls

First write the function in the textbox then add up to four parameters separated by commas.

Example

```
add(a,b) {  
    var c;  
    c = a+b;  
}
```

To call this function use the following line: **add,1,2**

Unfortunately strings are not supported yet.

To add the function call, select a bold dialog again and click the blue plus.

Go to a dialog

If you want to skip a few dialogs or go back to a dialog to repeat a decision for a puzzle or whatever you can use the goto function.

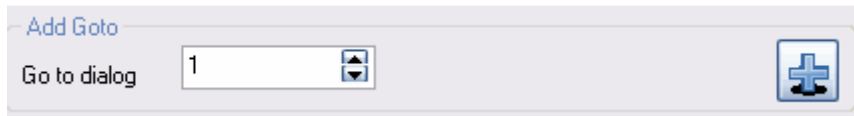


Illustration 9: Goto function

Enter the dialog number you want to goto in the number box, select a bold dialog and click the blue plus. **If you want to end the dialog instantly go to a dialog that does not exist!**

Delete a choice / dialog

If you want to delete a dialog or a choice than use this function.

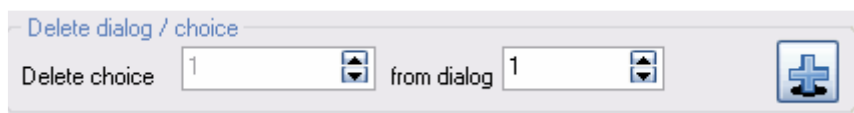


Illustration 10: Deleting dialogs / choices

Pick the dialog number by entering its number. If the selected object is a decision, the first number box is enabled, otherwise it is disabled. **If you end a dialog it will not be saved if objects have been deleted! So use this function with care.**

Search

If you have a very complex dialog and you search an item enter the string you want to search for and click the search button.

Creating a project

To create a project your dialog file must be opened and a config file must exist. Then click the green arrow in the left bottom corner of the main window. A window appears asking for a project name. All necessary files will be put in this folder. You should find there:

- Button images
- Avatars
- Font images
- The configuration file
- The dialogDLL
- The WDLAction.wdl
- The PanelEvent.wdl
- The dialog file

Additional dialogs can be used in your project by saving them in your project folder and including them in your wdl code.

Using the dialog

Copy all files in your game directory then include the PanelEvent.wdl in your WDL file and add the dllfunctions from the WDLAction.wdl into your code as well as the myDialog action and the startMyDialog function. Then assign the myDialog action to an entity in your game. That's it ;-)

Exiting a dialog

Calling *CleanDialog()* in your WDL code will close the active dialog. Be sure that you close every dialog before you end your game otherwise you will receive an error. Closing the window by clicking the x will automatically clean the dialog.

The Font Generator

To create coloured fonts you need to use a font image. The font generator offers a function to create these font images.

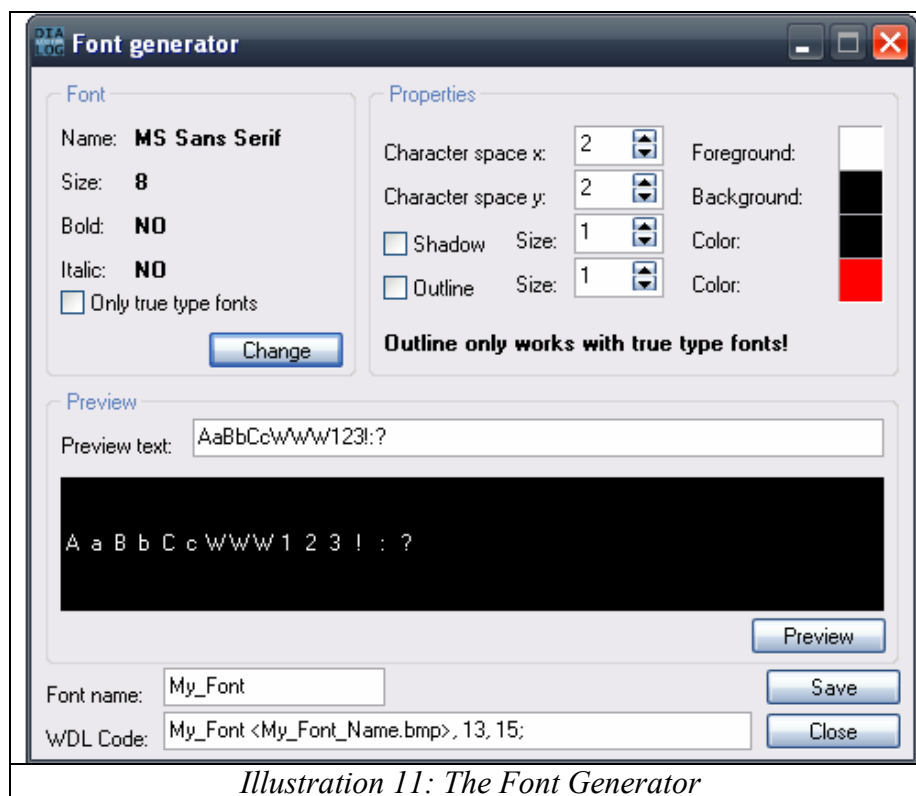


Illustration 11: The Font Generator

Font: Specifies the font with all its properties.

Properties: Space x and space y sets the space between the single letters.

Shadow: Sets if the font should have a shadow and if yes, its size.

Outline: This works only with true type fonts and creates a border around the letters if checked.

Preview: Previews the generated font in you standard image view application.

WDL Code: Generates the wdl code if you want to use your font in C-Script.

FAQ

What happens if a dialog is already running while I call another?

Long story short: The first dialog will be ‘cleaned’ and the second one will be started.

I always want to open the *.dlg files in the DialogED. How can I do that?

Easy! Click on **Options → Program options...** and choose **register dlg extention**.