Commandline Parameters for your game or application

The orxParam module is provided to make handling commandline parameters easier in your game or app. In this tutorial, we'll add a commandline parameter, and a handler for it.

To begin, `init` up a new project that will be used as the basis for this tutorial with: Creating your own Orx-based Project using 'init'

Now that you have a project which compiles, and runs, let's start by looking at what parameters already exist on the demo program. Go to your compiled application's bin folder. You'll see something like:

```
rwxr-xr-x   MyGamed.exe
rwxr-xr-x   orx.dll
rwxr-xr-x   orxp.dll
rwxr-xr-x   orxd.dll
```

If you execute your game with the `-h` parameter, ie, `MyGame.exe -h`, you will receive the following output:

```
Options:
-c        --config                  Loads the specified configuration file.
-v        --version                 Prints orx's version.
-h        --help                    Prints this help. A parameter can be
specified to print its complete description (-h <param>).
```

These are three built-in commandline parameters from orx.

You could also have specified `MyGame.exe --help`. This is the long format version of the parameter.

Adding your own

In this example, we want the user to be able to specify a different screen width on the commandline, rather than run in the default resolution set by the application.

Add the following to the top of your `Init()` function:

```
  orxPARAM stParam = {orxPARAM_KU32_FLAG_NONE, "w", "width", "Set the window
width", "Set the width of your application window",
&SetWindowWidthParameter};

  orxParam_Register(&stParam);
```

A parameter struct is created and then registered for use. The values above are:

orxPARAM_KU32_FLAG_NONE - No special behaviour for our param. See orxParam.h for options.

"w" - the short format parameter letter chosen. This means `-w` on the commandline.

"width" - this is the long format commandline parameter. The user can say: `--width` instead of `-w`.

"Set the window width" - Short format help. This is shown with `-h`.

"Set the width of your application window" - Long format help shown with: `-h width`.

&SetWindowWidthParameter - address of the handler function that will run if the parameter is used on the commandline.

The `SetWindowWidthParameter` function hasn't been created yet, so that's next:

```
static orxSTATUS orxFASTCALL SetWindowWidthParameter(orxU32 _u32ParamCount,
const orxSTRING _azParams[]){

    //At least two params expected for width: the -w and the value
    if (_u32ParamCount > 1) {
        orxLOG("At least two params found.");
    }

    int count;
    for (count = 0; count < _u32ParamCount; count++){
        orxLOG("Parameter: %s", _azParams[count]);
    }

    return orxSTATUS_SUCCESS;
}
```

The above function will only be called if either a long or short formatted parameter is passed to the application, `MyGame.exe -w 640` or `MyGame.exe --width 640`.

In the function above, at least two params are expected `-w` and `640` or whatever value the user wishes to pass in to change the screen size.

And finally, the params are looped over and printed out. The actual changing of the screen size can be left as an exercise for the reader.

Compile the project and then run it again with `MyGame.exe -h`. Hopefully you should see the new parameter in the available list:

```
Options:
-c        --config                  Loads the specified configuration file.
-w        --width                   Set the window wdth
```

```
-v        --version                  Prints orx's version.
-h        --help                     Prints this help. A parameter can be
specified to print its complete description (-h <param>).
```

That's pretty cool. The short and long parameter names are shown, and the short help text. To get the long help, you can type: `MyGame.exe -h width`:

```
Options:
-w --width              Set the width of your application window
```

The short and long parameter names are shown and the long format help is displayed.

More information

You can see more examples of orxParam in use in the orxCrypt and orxFontGen tools.