

Game Engine Programming

Lab 9: Technical Documentation

Karsten Pedersen
Department of Creative Technology

October 29, 2018

In this lab we are going to be looking into documenting your code in a consistent and formal manner. Lets begin by downloading and extracting Doxygen to your home directory (H:). Run the following command from a POSIX terminal emulator:

```
$ cd /h
$ wget http://ftp.stack.nl/pub/users/dimitri/doxygen-1.8.14.<arch>.bin.zip
$ unzip doxygen-1.8.14.<arch>.bin.zip
```

The most important step is to add *doxygen* from the extracted archive to your path. Run the following command.

```
$ echo $PATH
```

This should output a colon seperated list of directories that the system shell looks within for programs. We need to add the folder containing *doxygen* to it. We do this by creating and editing your shell profile:

```
$ vi ~/.bashrc
```

or:

```
$ notepad ~/.bashrc
```

What we need to do now is append to your *PATH* variable. We append because we want to keep what it already has but we just want to add an additional folder to search in. The following listing shows this as an example:

```
export PATH=$PATH:/h/doxygen
```

Note:

You do not want spaces in filenames. It causes issues with many things in the world of computers.

Now restart the terminal emulator and try running *doxygen*. You should see the help output appear as text. With this in place we are now ready to process our project. Change into your project directory, run *doxygen* to generate a template config file and run doxygen:

```
$ cd /h/Projects/myengine
$ doxygen -g Doxyfile
$ doxygen
```

You should notice that *latex* and *html* folders have appeared which contain the documentation output. You should not check these into your repository because they can be regenerated. If you use L^AT_EX as the typesetter for your report you can just type *make* in the latex folder or include it within your document, otherwise ignore it.

Next edit the *Doxyfile* file and in particular have a look at the following options:

- PROJECT_NAME
- EXTRACT_ALL
- INPUT
- RECURSIVE

From now on, you just need to rerun the *doxygen* command.

You can comment your code using the following format and changes should be reflected in the generated documentation:

```
/**
 * Represent a Player state
 */
class Player
{
    int health; ///< The amount of health remaining.
};
```

Note:

With this in place, have a look at the options *HAVE_DOT* and *CALL_GRAPH* to improve the diagrams generated by *Doxygen*. You will need to install the *dot* tool from the *GraphViz* package in a very similar way to *Doxygen*.