Windows debugging

Conda only ships the release libraries, therefore windows debug builds are not possible.

For a windows debug build we could utilise conan to source debug versions of our c++ dependencies.

As a basic conan file consider:

```
conanfile.txt
 [requires]
 poco/1.9.4
 boost/1.7.5
```

To create a debug instance of this environment we can install the dependencies with:

```
conan install -s build_type=Debug
```

Which means building our project would amount to:

```
$ conan install .. -s build_type=Debug
$ cmake .. -G "Visual Studio 16 2019"
$ cmake --build . --config Debug
```

To find the debug libraries we would make use of Conan's inbuilt cmake integration. In the conanfile.txt we may speicify the generators:

```
[generators]
cmake_find_package
cmake_paths
```

This will generate configuration files for us that ensure CMake find_package calls will look in the conan directories. To use these files, we must include them in our CMake files with:

```
include(${CMAKE_BINARY_DIR}/conan_paths.cmake)
```

As a full example, consider requiring zlib as a dependency for a simple c++ project:

conanfile.txt

```
[requires]
zlib/1.2.11
...

[generators]
cmake_find_package
cmake_paths
```

CMakeLists.txt

```
cmake_minimum_required(VERSION 3.16)
project(helloworld)
include(${CMAKE_BINARY_DIR}/conan_paths.cmake)
add_executable(helloworld hello.cpp)
```

```
find_package(ZLIB)
target_link_libraries(helloworld ZLIB::ZLIB)
```