res_path.h

```
#ifndef RES_PATH_H
#define RES_PATH_H

#include <iostream>
#include <string>   // <> search in path specified by properties
#include <SDL.h>    // <> search in current directory first

/*
* Get the resource path for resources located in res/subDir
* It's assumed the project directory is structured like:
* bin/
*  the executable
* res/
*  Lesson1/
*  Lesson2/
*
* Paths returned will be Lessons/res/subDir
*/
std::string getResourcePath(const std::string &subDir =
""){
```

```
//We need to choose the path separator properly based on which
    //platform we're running on, since Windows uses a different
    //separator than most systems
#ifdef _WIN32
    const char PATH_SEP = '\\';
#else
    const char PATH_SEP = '/';
#endif

    //This will hold the base resource path: Lessons/res/
    //We give it static lifetime so that we'll only need to
call
    //SDL_GetBasePath once to get the executable path
    static std::string baseRes;
    if (baseRes.empty()){
        //SDL_GetBasePath will return NULL if something went
wrong in retrieving the path
        char *basePath = SDL_GetBasePath();
        if (basePath){
            baseRes = basePath;
            SDL_free(basePath);
        }
        else {
```

```
            std::cerr << "Error getting resource path: " <<
SDL_GetError() << std::endl;
            return "";
        }
        //We replace the last bin/ with res/ to get the the resource path
        //size_t pos = baseRes.rfind("bin");
        //baseRes = baseRes.substr(0, pos) + "res" + PATH_SEP;
        size_t pos = baseRes.rfind("Debug");
        if (pos == std::string::npos)  // no "Debug", assume
"Release"
            pos = baseRes.rfind("Release");
        size_t prevSlash = baseRes.rfind(PATH_SEP, pos - 2);
// find slash before project name
        baseRes = baseRes.substr(0, prevSlash) + PATH_SEP +
"res" + PATH_SEP;

    }
    //If we want a specific subdirectory path in the
resource directory
    //append it to the base path. This would be something
like Lessons/res/Lesson0
    return subDir.empty() ? baseRes : baseRes + subDir +
PATH_SEP;
```

```
}

#endif
```