Program:	**GDP1**
Course:	**INFO6025 – Configuration and Deployment – Fall 2023**
Professor:	**James Lucas**
Project # 3:	**MSBUILD: AfterBuild Packaging**
Weight:	**25% of "project" mark**
	(or an even weight, given less or more than 4 total projects)
Due Date:	**Thursday, November 23, 10pm**

Note: This project must be done independently. No group submissions are allowed.

Description and Purpose

Create an MSBuild project that implements items, properties, tasks, and targets. This should be built from an existing project you have created that renders some scene.

This project is a packaging process that creates a single zip folder containing your executable and all required dependencies. The packaging process will be triggered after a "Release" mode build in Visual Studio.

Your Submission

- **Submit a single .zip containing your <u>entire</u> solution** (PLEASE remove 'extra' files from it, eg if you are using source control, I really don't want the *.git* folder or the *.vs* folder)
- No video is required for this project.

"Show-Stopper" Marks

While these aren't "worth" any marks, they could deliver large penalties, or in some cases a mark of zero.

- Does not compile (mark=0)
- Build fails, doesn't run, or runs with crashes/errors (I may investigate for a simple fix to something like a last-minute typo, but if it's not a super simple fix for me then… mark=0)
- No/Awful/Nonsensical documentation (-30%)
- No/Terrible/Painful to look at conventions/style (-50%)

Marking Scheme

This is how the marks are divvied up:

Item	Marks
ReadMe Include a ReadMe file that: • Lists the properties, items, tasks, targets you've created. • Describe where your "DependsOnTargets" and "Conditional Target" marks are found.	REQUIRED Or grade=0
Properties Create at least 3 properties to use in your MSBuild project.	10
Items Create at least 3 items to use in your MSBuild project.	10
Tasks Create at least 3 tasks to use in your MSBuild project. Your tasks must be grouped into a single target named "AfterBuild"	20
DependsOnTargets Implement at least one "DependsOnTargets" condition.	10
Conditional Target Your AfterBuild target must run only when building in "Release" mode, otherwise it should not be executed – even when requested on the command line.	10
Packaged Software The above requirements must all be implemented to create an output folder named "project3.zip" which contains your project executable and dependencies ONLY (no additional files allowed). Unzipping the contents to any location and running your executable must render a scene.	10
TOTAL	70

Additional Requirements

- While you may freely "borrow" mine (or anyone other) code **_but_** your code should be "sufficiently" different from mine. See the "plagiarism" test, later in this document, for more details.

- Further, you _cannot_ simply use an existing game engine (or part of a game engine), even if it's a "from source" engine (i.e. you have the entire source) to complete this assignment; it should be either completely new of significantly modified. This includes, but is _not_ limited to: Unity, Unreal, Cry, Anarchy, XNA, Cocos, Ogre, the framework from the OpenGL text, etc. In other words, you are expected to have made the vast majority (essentially all) of the engine _in this term by yourselves, from "scratch"_ - i.e. starting from something a rudimentary as the "OpenGL Book" code or the GLFW starter code (we started with that in class).

- You also may not use any "3rd party" physics code/libraries like bullet, havok, etc. The exception to this is code taken from the text book, of course, but you also can't just use Ian Millington's complete "cyclone" engine code – you can use code inspired from that, but it has to be almost completely "yours" and created this term.

- The most "engine" type code you can use is limited to GLFW, glad, and OpenGL Math (glm); anything more is almost certainly "too much engine" code.

75/10-year old "squinty eye" plagiarism test:

(Credit: Feeney)

I have very little tolerance for plagiarism, but many students are unclear about what it is.

Basically, it's submitting somebody else's work as your own.

There is sometimes some confusion over this because you could argue nothing is actually "unique" (see: http://everythingisaremix.info/ for a fascinating overview of this).

The whole point of assignments/tests/projects in this course (or any course, really) is to try to see if you are actually able to **_do_** the coding that's asked of you. In other words: How competent are you? Handing me someone else's code and/or making a trivial change isn't good enough.

Also, it's illegal:

- http://www.plagiarism.org/ask-the-experts/faq/

- http://definitions.uslegal.com/p/plagiarism/

- http://en.wikipedia.org/wiki/Plagiarism

-

In other words, I'm not going to be drawn into a giant debate over how "different" your code is from mine or anyone else's, if any sensible person (including me) would conclude that the code/application is pretty much the same thing, then it is. It is up to my discretion to decide this.

- While you may freely "borrow" mine (or anyone other) code **_but_** your code should be "sufficiently" different from mine (you might want to replace the word "sufficiently" with "significantly").

- In other words, you _cannot_ simply use an existing game engine (or part of a game engine) to complete this assignment; it should be either completely new of **significantly** modified.

- How will I determine this?

  o If I showed your application and/or your source code to either a pragmatic 75-year-old mother, or a typical 10-year-old, or even some random person walking down the hallway (i.e. a non-expert), and they looked at it, tilted their heads, squinted their eyes, and said "you know, they look the same," then they **_are_** the same.

  o Another test would: How much time it would take for a competent programmer (for example, me) to make the changes you are submitting? The point here is that I don't "care" if you tell me "But it took me _weeks_ to make the changes!" Fine, but if I can make those same changes in 10 minutes, then not a lot of work has been done (certainly **not** sufficient work – these projects should show take **days** of work having been done).