

PG5200 Final assessment

Solution structure

This solution contains the projects for the two assessments as projects (**CardCreator** and **LevelEditor** respectively). In addition, it contains several smaller projects that serve as utilities. These smaller projects were extracted because they contain functionality not directly related to the main projects and it also allows for sharing across the projects (as in the case of **I0** and **Result**).

Documentation

Please see the `documentation_card-creator` and `documentation_level-editor` for more details about the respective projects.

On the solution hierarchy, or: why is there only one solution?

While the two projects originally had their separate solutions, as they started to get more involved it became clear that they had certain functionality in common. There were a few possible solutions to this:

- I could compile separate library projects and use their `dll` files as references and bundle them with my project. The downside is I would have to re-build and re-add the dependencies whenever something changed. Not ideal.
- Alternatively I could publish them as packages to NuGet and use them as dependencies that way, but this would come with the additional overhead of learning how NuGet works *and* I would still have to republish and then update my dependencies for every change.
- I could just copy the code between the solutions. Quick and dirty, but if I change the code in one place, I'd have to remember to update the other copy as well. This would get messy quick.
- Add a project from the other solution. This would solve many of the problems but would likely cause breakage when set up on another machine as the paths probably wouldn't match if indeed both projects were there at all.
- Or I could simply add the one project from the Level Editor solution to the Card Creator solution and everything would work as intended. This also comes with the added benefit of having to deal with only one solution, rather than two separate ones.

Thus, I opted for putting both projects in the same solution. The main projects are both very well contained, but they both depend on **I0** and **Result** for handling saving and loading.