

## Project structure

It looks like the entire “maanetorn” folder is the root of an imported asset package. If there are plans to update it to a new version later, changing the structure now could cause problems. However, if the package will be used as-is, I’d still suggest making some adjustments for clarity and simplicity.

Create a “\_Project” folder (or similar) as the root for all project-specific assets. The leading underscore ensures it stays at the top of the Project view for better clarity and organization.

Rename the “Farm Game UI - Simple 2D UI” folder to simple “UI” since it only contains UI-related assets. This will simplify around a-lot of the project’s paths.

Folder names like Script should follow plural naming - Scripts.

Move FGDemoScene to Scenes and delete the default SampleScene.

I’d place the CustomButton script and its related README file into a CustomButton subfolder inside Scripts, since they belong together.

The Third-Party Notices file should go into the Fonts folder, following the same logic as the Button README.

There’s an error in one of the fonts - the main object name 'LilitaOne~ (for ‘greater than’ symbol)’ doesn’t match the asset filename. I would just rename it to 'LilitaOne' to fix the issue.

## UI

Move all UI panels to position (0, 0) to center them on the canvas.

Set the CanvasScaler component to UI Scale Mode = Scale With Screen Size, with a reference resolution of 1024×1024. Set Screen Match Mode to Expand so all panels fit the window across all aspect ratios.

## Performance optimization

Installed the 2D Sprite package to create texture atlases for rendering optimization. Currently, the number of sprites is small, so we can pack everything into a single atlas. However, in the future, we should separate atlases based on which sprites appear on screen simultaneously and for memory optimization purposes.

To preserve visual quality, source sprite compression must be disabled before atlas packing.

If we still experience a high number of draw calls, we can separate text and images in the UI hierarchy, ensuring no images overlap text elements. This can reduce draw calls to just two: one for images and one for text.

We should also monitor script performance, any low-performance areas can likely be improved by minimizing memory allocations and refactoring the code.

## **Pipeline optimization**

If it's a 2D project, set **Project Settings > Editor > Default Behavior Mode** to **2D** so that images are automatically imported as sprites. If there are any special requirements, we can always write a custom importer script to automate anything needed.