

Game folder structure

This is the guide on how to name things, where to put them, and over-all how we intend to organize the project files.

Considerations

The game project's file structure should achieve several key things:

- Organize multiple repositories into folder structures that can co-exist.
 - One directory for the Godot project, a separate directory for the art source files that are created outside of the engine.
 - These directories will each be a Git repository and should conform to the hyphenated naming convention.
 - These names should make sense in and out of context, the top-level folder name should make sense when viewed as a tab in a Git GUI e.g:
 - **example-game**
 - **example-art**
 - **example-audio**
 - Splitting the repositories avoids having to sync large amounts of irrelevant content for your role on the project:
 - Programmers / Dev Ops / Build Bots should only need to sync the **example-game** repository.
 - Artists / Animators can sync **example-art** along side the **example-game** repository.
 - Musicians / Foley Artists can sync **example-audio** along side the **example-game** repository.
 - The folder structures should work without using Git sub-modules, since sub-modules add a lot of complexity and confusion without really adding value.
- The folder structures in these directories should mimic each other 1:1
- Tools and pipelines should be able to export / import content by simply swapping the root, e.g:
 - **example-art**/art_assets/characters/main_character/main_character_diffuse.png
 - |--->**example-game**
/art_assets/characters/main_character/main_character_diffuse.png
- Ship-able game content should be the only thing going into the Godot project resource folder in the game repo.
- Art source content from tools like Blender, Gimp, Krita, etc... should be kept in the art repo.
- Music production files for a DAW should be kept in the audio repo.

Naming Conventions

- The repositories should use **lowercase-hyphenated-case**
- Files and folders should use **snake_case** unless otherwise required by third-party tools or conventions for the specific file e.g. README.md
- No spaces in file names.
- All directories should be plural:
 - "characters" not "character"
 - "environments" not "environment"

Compromises

No folder structure is perfect, but this structure should lead to the fewest problems. Here is a list of known compromises we've decided to accept:

Top level folder names repeat the game's name even though they are already inside of a top-level folder named after the game:

- **example**
 - **example-game**
 - **example-art**
 - **example-audio**

While this introduces redundancy, folders don't cost anything, so this is okay.

Some intermediate sub-folders are empty and seem to be useless, but they keep the hierarchies between directories in perfect sync. For instance, **example-art** may never end up with content directly inside of it, most if not all content will be inside of its first sub-folder **art_assets** i.e. `example-art/art_assets/` This is on purpose, since it ensures the game content is at the same hierarchical indentation level to the source content, making pipelines and paths extremely easy to work with; just swap the root and you're done!

You ought to have a directory on your computer for "**BUGJam**" projects. You can place this anywhere on your computer with a few considerations:

- Do not sync this directory with OneDrive, iCloud, Google Drive, or any other backup tool, this will break the Git repositories inside the folder since these services interfere with Git.
- Don't nest the folder too deep so you avoid file name length issues:
 - Good (Windows style path): `C:\Users\xgreer\Projects\BUGJam`
 - Good (MacOS / Linux): `/home/xgreer/Projects/BUGJam`
 - Bad: `/home/xgreer/Projects/Extra Projects/Seattle Blender User Group/Game Projects/BUGJam`

Within the BUGJam directory there will be folders for each BUGJam project. For a project called '**Genesis**' we will have a project folder called 'genesis' that serves as the root for all content necessary to develop the game.

File Tree Structure Example

```
BUGJam <!-- Top-level folder for the entire studio. -->
├── genesis <!-- Top-level folder that contains all repositories for the game. -->
│   ├── genesis-art <!-- Repository for large art content, stored outside of the Godot project. -->
│   │   ├── .forgejo <!-- Hidden content used for resources on Forgejo -->
│   │   │   └── banner
│   │   │       └── genesis_art_banner.png
│   │   ├── art_assets <!-- Source files for creating art -->
│   │   │   ├── characters
│   │   │   │   ├── main_character
│   │   │   │   │   ├── animations
│   │   │   │   │   │   ├── idle.blend
│   │   │   │   │   │   ├── main_character_diffuse.png
│   │   │   │   │   │   └── main_character_rig.blend
│   │   │   │   └── monster
│   │   │   │       ├── animations
│   │   │   │       │   └── walk.blend
│   │   │   │       └── monster_diffuse.png
│   │   │   └── monster_rig.blend
│   │   ├── audio_assets <!-- Source files for music and sound effects -->
│   │   │   ├── dialogue
│   │   │   ├── music
│   │   │   └── sound_effects
│   │   └── pipeline <!-- Pipeline tools for producing content outside of the game -->
│   │       ├── addons
│   │       ├── extensions
│   │       └── scripts
│   └── genesis-game <!-- This is the Godot root for resources: 'res://' -->
│       ├── .forgejo <!-- Hidden content used for resources on Forgejo -->
│       │   └── banner
│       │       └── genesis_game_banner.png
│       ├── addons <!-- Godot plugins -->
│       ├── art_assets <!-- A 1:1 mirrored file structure with the genesis-art repository. -->
│       └── characters
```

```
| | | └─ main_character
| | | | └─ main_character_mat.tres <!-- Material -->
| | | | └─ main_character_diffuse.png <!-- Texture -->
| | | | └─ main_character_diffuse.png.import <!-- Import settings and UID -->
| | | | └─ main_character_rig.bin <!-- Binary data for the model file -->
| | | | └─ main_character_rig.gltf <!-- Model file -->
| | | | └─ main_character_rig.gltf.import <!-- Import settings and UID -->
| | | └─ monster
| | | | └─ monster_mat.tres <!-- Material -->
| | | | └─ monster_diffuse.png <!-- Texture -->
| | | | └─ monster_rig.bin <!-- Binary data for the model file -->
| | | | └─ monster_rig.gltf <!-- Model file -->
| | | | └─ monster_rig.gltf.import <!-- Import settings and UID -->
| | └─ common
| | | └─ textures
| | | | └─ blue_noise_64.png
| | └─ environments
| | | └─ rock
| | | | └─ [...]
| | | └─ wall
| | | | └─ [...]
| | └─ interactables <!-- Core puzzle elements for designing levels. -->
| | | └─ coin
| | | | └─ [...]
| | | └─ door
| | | | └─ [...]
| └─ audio_assets <!-- A 1:1 mirrored file structure with the genesis-audio repository. -->
| | └─ dialogue
| | └─ music
| | └─ sound_effects
| └─ entities <!-- Complete assets that are ready to drag and drop into a level. -->
| | └─ characters
| | | └─ main_character.tscn
| | | └─ monster.tscn
| | └─ environments <!-- Stuff without direct gameplay mechanics. -->
| | | └─ rock.tscn
| | | └─ wall.tscn
| | └─ puzzle_elements <!-- Core puzzle elements for designing levels. -->
| | | └─ button.tscn
| | | └─ pusher.tscn
| └─ levels <!-- Playable levels. -->
```

```
| | └─ level_packs <!-- Data assets that organize levels into packs. -->
| | | └─ world1.gd
| | | └─ world2.gd
| | └─ epilogue.tscn
| | └─ intro.tscn
| | └─ open_world.tscn
| └─ scripts
| | └─ player.gd
| └─ shaders
| | └─ iris.gdshader
| └─ ui
| | └─ icons.png
| └─ .gitignore
| └─ icon.png
| └─ project.godot <!-- Main Godot project file -->
| └─ README.md
└─ logo <!-- Example of other stuff for the studio that is not directly related to the game.
-->
```

Revision #17

Created 2025-12-22 21:48:44 PST by xury

Updated 2026-02-14 09:43:22 PST by xury