# File Structure Explained:

**Key Points**

* **Assets stay separated** → Never mix raw images, sounds, or fonts directly with scripts/scenes.
* **Scenes stay modular** → Each object (player, enemies, UI, etc.) gets its own folder.
* **Scripts match scenes** → If player.tscn exists, its code goes in scripts/player/player.gd.
* **Data folder** → Use for save files, configs, or balancing tables (e.g. enemy stats).
* **Docs folder** → Perfect for design notes, sketches, and gameplay ideas.
* **.gitignore setup** → Ensures you don’t commit Godot’s .import cache files or temp files to GitHub.

## File Structure:

/project-root

│ project.godot # Godot project file

│ README.md # (Optional) Short project info

│

├── assets/ # Raw imported assets (never edit directly)

│ ├── audio/

│ │ ├── music/

│ │ └── sfx/

│ ├── fonts/

│ ├── sprites/

│ │ ├── characters/

│ │ ├── enemies/

│ │ └── environment/

│ ├── textures/ # Non-sprite textures (e.g. UI backgrounds)

│ ├── tilesets/

│ ├── ui/

│ └── shaders/

│

├── scenes/ # All .tscn files (levels, UI, etc.)

│ ├── player/

│ ├── enemies/

│ ├── environment/

│ ├── ui/

│ └── world/

│

├── scripts/ # All GDScript or C# files

│ ├── player/

│ ├── enemies/

│ ├── systems/ # Game managers, spawners, etc.

│ ├── ui/

│ └── utils/ # Helper functions, reusable code

│

├── data/ # Non-asset data

│ ├── configs/

│ ├── saves/

│ └── tables/

│

├── docs/ # Design docs, notes, references

│

└── .gitignore # Ignore imports, temp files, etc.