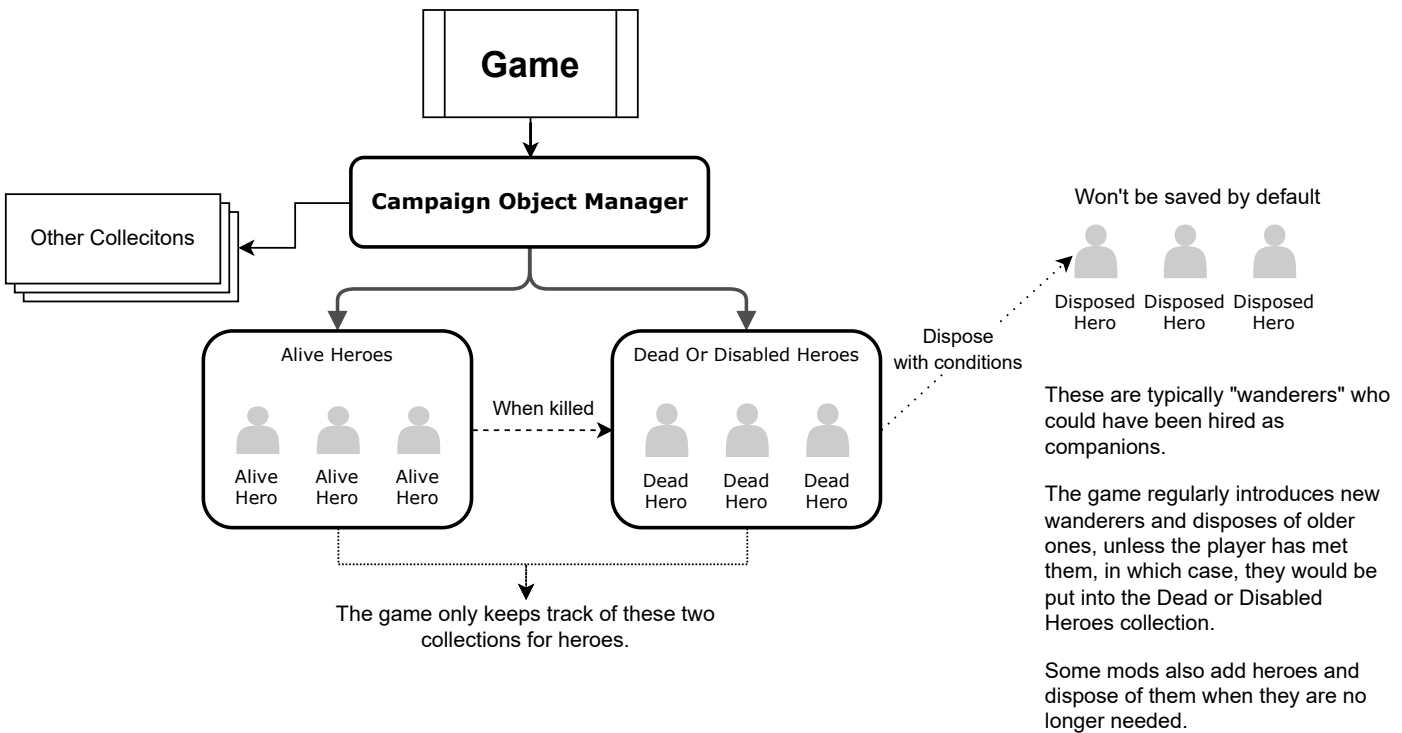


How The Save System Works

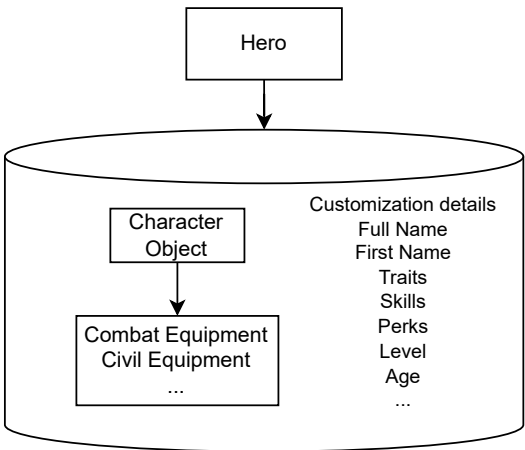
The save system tracks objects starting from the "Game" object, following links to the **Object Manager** and all connected objects.

Unlinked objects, such as disposed heroes, are not saved because the system cannot trace the connection from the "Game" object to them.

Example: The Heroes



Data that a hero would have



Why Save Bloat Happens

Save bloat occurs when mods create persistent links to game objects. For example:

- Mods often track heroes when they are spawned to manage complex relationships or features. However, they may fail to untrack these heroes after they are disposed of. As a result, these disposed objects remain linked to the "Game" object through the mod, causing them to be saved along with all their associated data.
- Some mods generate new heroes and dispose of them when they are outdated. While the game may unregister these heroes and remove certain data from them, it never cleans up links created by mods. If another mod tries to access removed data from these disposed heroes through lingering references, it can cause errors like **NullReferenceExceptions**, potentially crashing the game.

Disposed objects can remain linked through mods or other references, preventing the game from fully removing them during the save process. The **Save Cleaner** helps by identifying these disposed objects and cutting off their links from their sources. While it can assist with this issue, mod authors may also find this document helpful to ensure their mods properly clear references to disposed objects to prevent save bloat, improve performance, and reduce the risk of errors related to invalid references.

