Amnesia-styled Inventory

The Inventory System comes with two built-in modules: The Inventory Module and the Journal Module. Each of these modules comes with its own corresponding file with helper functions.

The Inventory System can be downloaded from the Steam Workshop or from ModDB. With the download comes a sample map that demonstrates the most common usages of the Inventory System's capabilities.

Inventory Module

```
void Inventory_UpdateToolDescription(const tString &in asTool,
    const tString &in asToolName,
    const tString &in asDescription)
```

The Inventory Module works with the PlayerTool module to see which tools are in the player's possession, but Tools must also be registered with the Inventory Module to give them a corresponding title and description. Use this function to accomplish this.

tString asTool - The name of the Tool entity you wish to register.

tString asToolName - The title of the Tool entity that will be visible in the Inventory interface.

tString asDescription - The description of the Tool entity that will be visible in the Inventory interface.

```
void Inventory_SetTransCategory(const tString &in asCategory)
```

The Inventory Module supports the use of the ImGui lang file TransCategories. Use this function to set the TransCategory you wish to use.

tString asCategory - The name of the TransCategory you wish to utilize.

```
bool Inventory_IsToolEquipped(const tString &in asTool)
```

The Inventory Module works in tandem with the PlayerTool module when equipping tools through the Inventory interface. Use this function if you wish to check if the given Tool is the one that is equipped.

tString asTool - The name of the Tool entity to check if it is equipped.

returns - Whether the given tool is currently equipped.

```
void Inventory_UnequipTool()
```

If you wish to manually unequip the currently equipped tool, use this function.

Journal Module

```
void Journal_SetTransCategory(const tString &in asCategory)
```

The Journal Module supports the use of the ImGui lang file TransCategories. Use this function to set the TransCategory you wish to use.

tString asCategory - The name of the TransCategory you wish to utilize.

```
void Journal_SetCondensedLangFile(const tString &in asLangfile)
```

Because of certain technical limitations, the Journal Module requires that the entries for condensed categories be stored in their own separate lang file (using the same schema and category name).

tString asLangfile - The relative path to the lang file.

```
bool Journal_AddCategory(const tString &in asCategoryName,
    const tString &in asCategoryTitle,
    int alCategoryWeight = ,
    bool abCollapseEntries = false)
```

This function allows the creation of journal categories. The categories are groups that any entries will be sorted into. If a category is marked as “collapsed”, all entries within that category will be treated as a non-interactable list rather than a set of individual entries. (This is useful to achieve things like current objectives.)

tString asCategoryName - The unique name of the given category.

tString asCategoryTitle - The title of the category that will be visible in the Journal interface.

int alCategoryWeight - The weight of this category in the category list.

bool abCollapseEntries - If true, this category will collapse its entries into a list.

returns - Whether the addition was successful.

```
bool Journal_AddEntry(const tString &in asEntryName,
    const tString &in asEntryTitle,
    const tString &in asEntryDate,
    const tString &in asEntryJournalCategory,
    const tString &in asEntryOnReadCallback,
    const tString &in asEntryContents,
    int alEntryWeight = ,
    bool abContentIsImage = false)
```

This is the function that allows the addition of specific journal entries. The entries will be sorted into their corresponding categories and by weight. (Note: If the entry is intended for a collapsed journal category, only the entryName and entryTitle will be used.)

tString asEntryName - The unique name of the entry given.

tString asEntryTitle - The title of the entry given that will be visible in the Journal interface. (In a condensed category, this field will serve as the entry body.)

tString asEntryDate - The date of the entry. Can be left blank. (Ignored in a condensed category.)

tString asEntryJournalCategory - The journal category this entry will be assigned to.

tString asEntryOnReadCallback - The name of the callback function to be called when this entry is read. Can be left blank.

tString asEntryContents - The content body of this entry. Can be left blank. (Ignored in a condensed category.)

int alEntryWeight - The weight of this entry in the entry list.

bool abContentIsImage - If true, the entry body will be replaced by an image defined by the asEntryContents field.

returns - Whether the addition was successful.

```
bool Journal_UpdateCategory(const tString &in asCategoryName,
    const tString &in asCategoryTitle,
    int alCategoryWeight = ,
    bool abCollapseEntries = false)
```

This function allows the modification of existing journal categories.

tString asCategoryName - The unique name of the category to be modified.

tString asCategoryTitle - The title of the category that will be visible in the Journal interface.

int alCategoryWeight - The weight of this category in the category list.

returns - Whether the update was successful.

```
bool Journal_UpdateEntry(const tString &in asEntryName,
    const tString &in asEntryTitle,
    const tString &in asEntryDate,
    const tString &in asEntryJournalCategory,
    const tString &in asEntryOnReadCallback,
    const tString &in asEntryContents,
    int alEntryWeight = )
```

This function updates the contents of existing journal entries.

tString asEntryName - The unique name of the entry to be updated.

tString asEntryTitle - The title of the entry given that will be visible in the Journal Module. (In a condensed category, this field will serve as the entry body.)

tString asEntryDate - The date of the entry. Can be left blank. (Ignored in a condensed category.)

tString asEntryJournalCategory - The journal category this entry will be assigned to.

tString asEntryOnReadCallback - The name of the callback function to be called when this entry is read. Can be left blank.

tString asEntryContents - The content body of this entry. Can be left blank. (Ignored in a condensed category.)

int alEntryWeight - The weight of this entry in the entry list.

returns - Whether the update was successful.

```
bool Journal_RemoveCategory(const tString &in asCategoryName)
```

This function removes a journal category from the list. (This action will remove all entries under the category as well. This cannot be reversed.)

tString asCategoryName - The unique name of the entry to be removed.

returns - Whether the removal was successful.

```
bool Journal_RemoveEntry(const tString &in asEntryName)
```

This function removes a journal entry from the list. (This action cannot be reversed.)

tString asEntryName - The unique name of the entry to be removed.

returns - Whether the removal was successful.

```
int Journal_GetEntryCount()
```

This function returns the total number of entries in the journal.

returns - The number of entries in the journal.

```
void Journal_ReadEntry(const tString &in asEntryName)
```

An entry's OnReadCallback function will automatically fire when the entry is read. To manually fire this callback, use this function. (Does nothing if the entry has no assigned OnReadCallback function.)

tString asEntryName - The unique name of the entry whose OnReadCallback will be fired.

```
tString Journal_GetLastEntry()
```

This function will return the unique name of the last entry that was added or updated.

returns - The unique name of the last added or updated entry.

```
void Journal_ShowMessage(const tString &in asMessage,
    const tString &in asIcon = "")
```

This function will toggle a notification message to appear in the bottom left of the screen, accompanied by an optional image. This notification will automatically disappear after a few seconds.

tString asMessage - The message to be displayed.

tString asIcon - The image path to be displayed.